Frontiers in Robotics, Automation and Control

Edited by
Alexander Zemliak

Frontiers in Robotics, Automation and Control

Frontiers in Robotics, Automation and Control

Edited by
Alexander Zemliak

In-Tech

Published by In-Tech

http://intechweb.org/
Additional copies can be obtained from:
publication@ars-journal.com

First published October 2008
Printed in Croatia

A catalogue record for this book is available from the University Library Rijeka under no. 120101002
Frontiers in Robotics, Automation and Control, Edited by Alexander Zemliak
p. cm.
ISBN 978-953-7619-17-6
1. Robotics. 2. Automation I. Alexander Zemliak

Preface

This book contains some new results in automation, control and robotics as well as new mathematical methods and computational techniques relating to the control theory application in physics and mechanical engineering. It contains the latest developments and reflects the experience of many researchers working in different environments (universities, research centers or even industries), publishing new theories and solving new problems in various branches of automation, control, robotics and adjacent areas. The main objective of the book is the interconnection of diverse scientific fields, the cultivation of possible scientific collaboration, the exchange of views and the promotion of new research targets as well as the future dissemination and diffusion of the scientific knowledge.

This book includes 23 chapters introducing basic research, advanced developments and applications. The book covers topics such us modeling and practical realization of robotic control for different applications, researching of the problems of stability and robustness, automation in algorithm and program developments with application in speech signal processing and linguistic research, system's applied control, computations, and control theory application in mechanics and electronics.

The authors and editor of this book hope that the efforts of the authors to provide high-level contributions will be appreciated by the relevant scientific and engineering community. We are convinced that the book will be a source of knowledge and inspiration for students, academic members, researchers and practitioners working on the topics covered by the book. We cordially thank I-Tech Education and Publishing for their efforts to maintain a high quality book.

Editor
Alexander Zemliak
Puebla Autonomous University
Mexico
National Technical University of Ukraine "KPI"
Ukraine

Contents

1

Evaluation of Robotic Force Control Strategies using an Open Architecture Test Facility

Michael Short
University of Leicester
United Kingdom

1. Introduction

Industrial robots are currently employed in a large number of applications and are available with a wide range of configurations, drive systems, physical sizes and payloads. However, the numbers in service throughout the world are much less than predicted over twenty years ago (Engelberger 1980). This is despite major technological advances in related areas of computing and electronics, and the availability of fast, reliable and low-cost microprocessors and memory. This situation is mainly a result of historical and economic circumstances, rather than technical considerations. Industrial robots have traditionally performed a narrow but well-defined range of tasks to a specified degree of accuracy and whilst new robot arm designs are specified for many years of continuous operation, the technological development of their controllers has been slow in comparison with other computer-based systems.

Traditionally, most industrial robots are designed to allow accurate and repeatable control of the position and velocity of the tooling at the device's end effector. Increasingly, these systems are often also required to perform complex tasks requiring robust and stable force control strategies. In addition, task constraints sometimes require position or velocity control in some Degrees-Of-Freedom (DOF), and force control in others. Thus, to fulfil these extra demands, an important area of robotics research is the implementation of stable and accurate force control. However this is often difficult to achieve in practice, due to the technological limitations of current controllers, coupled with the demanding requirements placed upon them by the advanced control schemes that are needed in cases where robots are operating in unpredictable or disordered environments.

This chapter describes a research project that has been undertaken to partly address these issues, by investigating algorithms and controller architectures for the implementation of stable robotic force control. The chapter is organised as follows. In Section 2, the fundamental concepts of robotic force control are introduced, and the problems inherent in the design of stable, robust controllers are described. This Section also describes some of the difficulties that are faced by developers when implementing force control strategies using traditional robot controllers. It is shown that linear, fixed-gain feedback controllers designed using conventional techniques can only provide adequate performance when they are tuned to specific task requirements. In practice the environmental stiffness at the robot/task

interface may be unknown and bounded, and may even vary significantly during the course of a specific task. In such cases, performance can be significantly degraded and is often exacerbated further by the sampling and processing limitations of traditional robot controllers.

In Section 3, a brief summary of previous work in the area of force control is given. Several strategies designed to help ameliorate the stability problems described in Section 2 are covered; two of these novel force control strategies are then discussed in greater depth. The first of these two techniques is based around an adaptive PD controller implemented using fuzzy inference techniques. The second technique centres on a model-following force controller that is robust to bounded uncertainty in the environmental stiffness. General design principles for both types of controller are discussed; the remainder of the chapter seeks to further investigate the performance of these two strategies. Section 4 describes a prototype open architecture robot controller that has been developed to overcome some of the fundamental restrictions of traditional controllers; this facility allows the direct real-time implementation of the force controllers.

Section 5 provides comparative results from a series of experiments that were undertaken to evaluate the performance of the controllers. Several additional measures of real-time performance and design complexity are also discussed. In Section 6, it is concluded that although both controllers display comparable performance, the model-based controller is favourable due to its reduced implementation overheads and reduced design effort, coupled with the fact that it lends itself to a simpler stability analysis.

2. Robotic Force Control

A typical conventional force control scheme is shown in Figure 1 (Zhang & Hemami 1997; Whitney 1985; Bicker et al. 1994). In the figure, f_r is the reference force, f_m is the measured (processed) force, f_e is the force feedback error and f_a is the actual applied force. The 'Position Controlled Robot' block consists of a robot and its host (proprietary) controller. The force sensor and related control elements are typically implemented as a physically separate system from the host controller. A control signal u is generated by the force controller, and effectively passed to the host controller as a vector of reference positions to be tracked. The end effector generates the forces and torques through interaction with the current contact dynamics. When implementing such a strategy, it is common for the external outer loop controller to pass the position commands to the proprietary joint controller over some form of communications link; such a feature has been common in most industrial robot controllers for many years. For example the ALTER command with the PUMA range of robots allows position setpoints to be sent from an external device over an RS-232 serial link, using a simple messaging protocol (Bicker et al. 1994).

The contact dynamics are represented by the combined stiffness at the end effector/task interface in the direction of the applied force (K_e). There is quite often a very short lag in these dynamics; however this is often neglected as it is many orders of magnitude smaller than the dominating lags elsewhere in the system. The environmental stiffness gain typically varies between a minimum value, determined by the objects in the environment with which the robot is in contact, and a maximum value, limited by the stiffness of the arm and torque sensor. The latter is dominant when the robot is touching a surface of very high stiffness, i.e. in a hard contact situation. Designing a fixed-gain conventional controller to meet a chosen

specification for a specific value of K_e is, in principle, a relatively straightforward task. A problem arises when K_e is unknown or variable; for example, consider the case where the system is tuned to achieve a specified performance at an upper limit of K_e. At low K_e the system will be overdamped, with a relatively high settling time. Conversely, if the system has been tuned for the desired performance at the lower limit of K_e, significant overshoot and oscillatory behaviour would occur at higher stiffness values. Figure 2 shows such a situation, using data recorded for the robotic system described in Section 4. In this figure, two plots of contact force for a fixed-gain controller tuned for low K_e are displayed. The low K_e contact situation is as expected; however oscillatory behaviour for the high K_e situation can clearly be seen. In practical robotic systems, this kind of 'chattering' behaviour can have serious consequences, potentially causing serious damage to the robot and its environment.

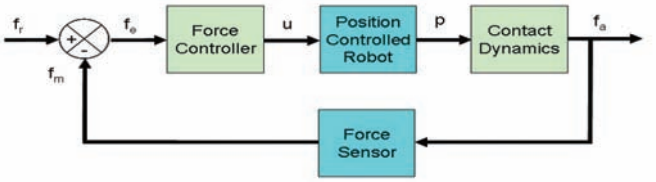


Fig. 1. Typical conventional robotic force control scheme

Other major factors contributing to poor, unstable performance include the finite and relatively low sampling rates of many industrial robot control systems. These problems are often considerably worsened by the presence of noise, non-linearities and other factors. For this reason, force controllers of the type described usually require some form of environment stiffness detection technique to enable the controller gains to be switched accordingly. The main problem with this process is that it is time consuming, often involving 'guarded moves' to contact in order to enable sufficient data to be collected for the algorithm to work. Such methods are also vulnerable to the presence of transducer noise, and are not very effective in situations where K_e is variable or rapidly changing - for example during a deburring task (Ow 1997). This also has the effect of slowing down task execution significantly. Problems such as these have motivated much research into designing efficient force control schemes, and this is the subject of the next Section.

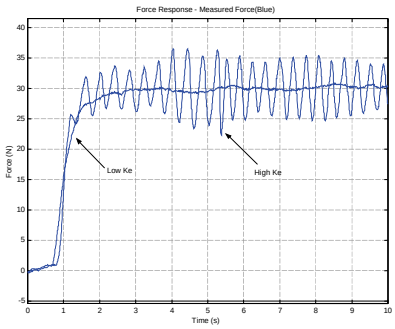


Fig. 2. Environmental stiffness effects on the performance of a fixed-gain force controller

3. Advanced Force Control Schemes

A large number of force control techniques of varying complexity have been proposed over the last twenty years (Zhang & Hemami 1997; Whitney 1985). The most basic direct methods simply transform joint-space torques into a Cartesian-space wrench, either in an open-loop fashion (which does not require the explicit measurement of forces and torques) or using inner and outer closed loops for accurate control of joint torques and Cartesian forces, respectively. However, since most industrial robots have position control loops that are not easily modified, indirect methods such as those described in the previous Section are often preferred. As mentioned, these involve modifying either joint or Cartesian position setpoints in order to control forces by deliberately introducing position control errors and using the inherent stiffness of the manipulator in different Cartesian directions.

As mentioned, stable force control is particularly difficult to achieve in 'hard' or 'stiff' contact situations, where the control loop sampling rate may be a limiting factor. In an attempt to improve stability various methods have been proposed, the simplest being the addition of compliant devices at the robot wrist (Whitney & Nevins 1979). Another solution is to employ 'active compliance' filters, where force feedback data is digitally filtered to emulate a passive spring/damper arrangement (Kim et al. 1992). However, both methods introduce a potentially unacceptable lag. Recent increases in processing power of low-cost computers has led to an increased interest in 'intelligent control' techniques such as those employing fuzzy logic, artificial neural networks and genetic algorithms (Linkens & Nyongsa 1996). Where attempts have been made to employ these techniques (specifically fuzzy logic) in explicit robot force controllers, simulation studies have demonstrated good tracking performance despite wide variations in environment stiffness, e.g. (Tarokh & Bailey 1997; Seraji 1998), and for specific contact situations, e.g. deburring (Kiguchi & Fukuda 1997). Improved performance using a hierarchical fuzzy force control strategy has also been demonstrated for various contact situations, such as peg-in-hole insertion (Lin & Huang 1998). A highly successful and generically applicable force control strategy based upon a Sugeno-style Fuzzy Inference System (FIS) was proposed by Burn et al. (2003), and will be described in more detail in Section 3.1.

However, these fuzzy techniques are not without problems. In addition to problems associated with the 'curse of dimensionality', i.e. large numbers of rules that must be evaluated in the inference process, the performance and stability of fuzzy systems are often difficult to validate analytically (Cao et al. 1998; Wolkenhauer & Edmunds 1997). Additionally, when compared to more 'traditional' control methods such as LQR (Frankin et al. 1994), the resulting fuzzy designs are more complex, have larger memory requirements and larger execution times (Bautista & Pont 2006). Such a technique which has proved to be popular in recent years has been the use of Model Following Control (MFC). Due to its conceptually simple design and powerful robustness properties, this type of controller has been found to be particularly suited to industrial applications such as robotics and motion control (e.g. Li et al. 1998; Osypiuk et al. 2004). Recent investigations have also shown that MFC-based techniques can be successfully applied in the force control domain (Short & Burn 2007). The MFC-based force control technique will be investigated further in Section 3.2.

3.1 Fuzzy Approach To Force Control

A method of designing Sugeno-style fuzzy controllers has previously been developed that

effectively produces a Proportional + Velocity (PV) controller with variable gains, capable of maintaining acceptable performance irrespective of K_e (Burn et al. 2003). A block diagram of the arrangement is shown in figure 3. To design a controller using this method, firstly a Sugeno-style FIS is created to emulate a conventional PV controller tuned for a high K_e environment. The FIS is assigned three inputs (f_e, Δf_e and Δp), and one output (u), where the input ranges are measured from conventional system data. The output from the FIS is a velocity demand. In order to create a linear system, initially only a single Membership Function (MF) for each input and output is required. By assigning names normal to the input MF's, and u_1 to the output MF, a rule of the following form produces the desired linear control surface: IF (f_e, Δf_e, Δ_p) are 'normal' then u is u_1. Note that a consequence of employing only one rule is that no defuzzification algorithm is required. By employing a first-order, Sugeno-style FIS, output u_1 is then defined by:

$$u_1 = K_1 \cdot f_e + K_2 \cdot \Delta f_e + K_3 \cdot \Delta p + K_4 \tag{1}$$

where K1 is a positive constant (equal to the forward gain K_p of a PV controller), K3 a negative constant (equal to the velocity feedback gain K_v), and K2 and K4 are - in this case - set to zero.

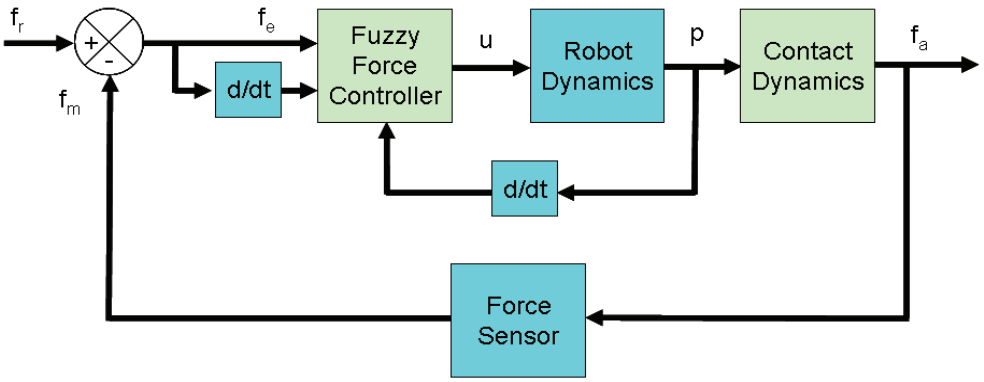


Fig. 3. Fuzzy force controller

The choice of MF type is influenced by the concept of data 'spread', and the measurement or calculation of standard deviation data σi from step response tests. For the single rule system each input is assigned a single Gaussian MF centered at zero, each with a σ_{normal} parameter equal to that of data obtained from tuned step responses at high K_e. Since the single rule system emulates a conventional PV controller it suffers the same disadvantages in the face of variable K_e. However, having created the initial FIS, it is now possible to modify the controller using a combination of analytical and intuitive methods.

With the system tuned for high K_e, during soft contact the maximum value of Δf_e is reduced. This reflects an overdamped response, an undesirable effect that can be minimized by increasing the proportional gain component of the controller output given by equation (1) if

lower Δf_e is 'detected' by the fuzzy controller. This is achieved initially by adding a second Gaussian MF to the Δf_e input set (low), with a smaller standard deviation $\sigma_{\Delta felow}$. In addition, during a dynamic response of a tuned system to a step input, the maximum value of Δp is inversely proportional to K_e. In other words, Δp increases during soft contact. A second rule is thus added to take into account the decrease in Δf_e relative to the 'normal' (desired) profile, and the relative increase in Δp. By adding a second output of the same form as equation (1) it is possible to vary the effective gains. Therefore, a rule is added of the form: IF (Δf_e is low) AND (Δp is high) then u is u_2, where u_2 has the same form as u_1 in equation (1), but with a modified forward gain component $K1_a$, equivalent to K_p tuned for soft contact such that $K1_a > K1$, and $\sigma\Delta p_{high} > \sigma\Delta p_{normal}$.
The advantage of the method lies in its apparent simplicity, although its success relies upon the correct determination of the MF parameters, particularly $\sigma\Delta p_{high}$ and $\sigma\Delta p_{normal}$. Due to the structured and well-defined methodology utilized in creating the controller design, as a related work a software design tool was created that automates the process of designing a fuzzy force controller. The tool includes an iterative method to tune these MF parameters until acceptable performance is achieved (Burn et al. 2004).

3.2 Model-Based Approach To Force Control

The robust model-based force controller previously described by Short & Burn (2007) is loosely based around a robust PID strategy discussed in detail by Scokzowski et al. (2005). The original strategy is based upon a two-loop MFC, containing a nominal model of the controlled plant and two PID controllers. The block diagram of a basic MFC controller is shown in figure 4.

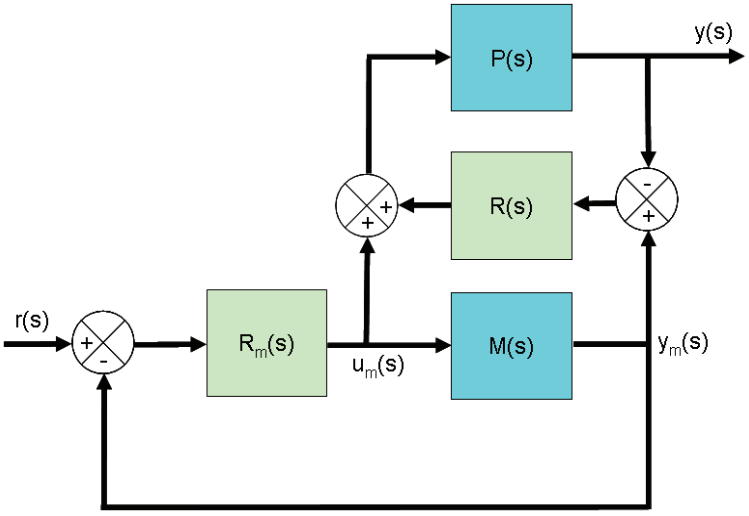


Fig. 4. Robust PID based on MFC

In this type of control, the model compensator $R_m(s)$ is tuned to a nominal model of the plant M(s); the actual plant P(s) contains bounded uncertainties. The auxiliary controller

R(s) acts on the difference between the actual process output and the model process output to modify the model control signal $u_m(s)$, which is also fed to the plant. In the case of robotic force control, the model M(s) is simply the second order motion control loop dynamics, augmented by a free integrator, and a known (base) value of environment stiffness.
Assuming that model is of reasonable quality, the bounded uncertainty in the plant is then dominated by the environment stiffness K_e, varying between K_{emax} and K_{emin}.
If the two loop controllers R(s) and $R_m(s)$ are simple proportional gains, as shown in Figure 5, then the MFC structure is considerably simplified. The model loop gain K_p can be tuned for K_{emax} - a relatively trivial task - whilst the auxiliary loop gain K_p' can be tuned to provide an additional control signal should the actual value of K_e be less than K_{emax}. However, with this type of controller structure it is important to consider the stability criteria, and provide a bound on the maximum value for K_p'.

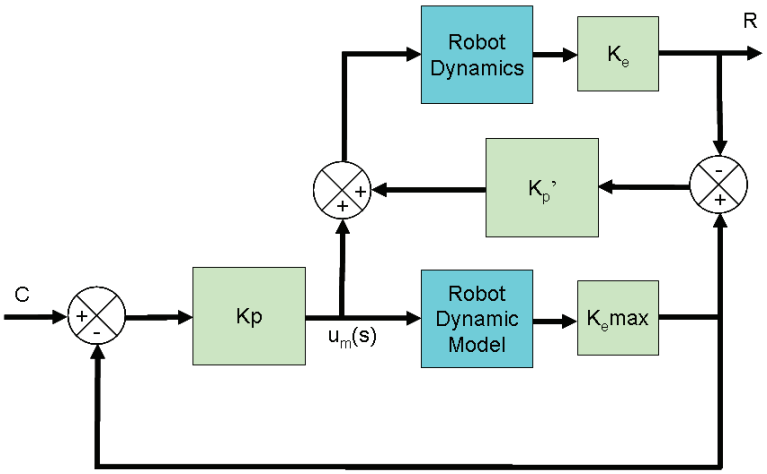


Fig. 5. Robust force controller

If the 'model loop' controller $R_m(s)$ is tuned for stability using a nominal design method on the plant P(s) augmented by the maximum environmental stiffness gain K_{emax}, then the stability of the overall control strategy is restricted by the roots of the equation:

$$1 + R(s)M(s)[1 + \Delta(s)] = 0 \tag{2}$$

Where $\Delta(s)$ denotes the model perturbations (uncertainty). The objective is to find for a given plant and bounded uncertainty in the stiffness gain a maximum bound on $|R(s)|$ that will maintain stability. In the case where the uncertainty exclusively resides in the environment stiffness gain K_e, then if the original loop is tuned for K_{emax} then $M(s)[1+\Delta(s)]$ in (2) reduces to:

$$M(s)[1 + \Delta(s)] = P(s) = G(s)K_{e\max} \tag{3}$$

Where G(s) represents the nominal robot dynamics and has the form (due to the free integrator in the forward path):

$$G(s) = \frac{\omega_n{}^2}{s^3 + 2\xi\omega_n s^2 + \omega_n{}^2 s} \tag{4}$$

Since the controller R(s) in this case is a single gain, K_p', using (3) and (4), equation (2) can be re-written as follows:

$$s^3 + 2\xi\omega_n s^2 + \omega_n{}^2 s + \omega_n{}^2 Kp' Ke_{\max} = 0 \tag{5}$$

Applying the Routh-Hurwitz stability criterion (Pippard 1997) for a cubic equation, the system will be stable if all the co-efficients in the left of (5) are positive, and the following criterion is satisfied:

$$2\xi\omega_n\omega_n{}^2 \geq \omega_n{}^2 Kp' Ke_{\max} \tag{6}$$

Re-arranging (6) gives a stability limit for the controller gain $K_p'_{max}$ as follows:

$$Kp'_{\max} = \frac{2\xi\omega_n}{Ke_{\max}} \tag{7}$$

Thus if the gain K_p' is chosen between the limits:

$$Kp < Kp' < Kp'_{\max} \tag{8}$$

The controller will be stable for unknown environment gains in the range $0 < K_e \leq K_{emax}$; as for all gains below K_{emax}, the stability criteria of (6) holds. Clearly, the formulation of these two controllers follows two distinct paths. The first is mainly based on an intuitive, heuristic formulation, while the second is based on a more thorough analytical approach. In Section 5, experimental results are presented for both controllers applied to an experimental test facility, which is described in the following Section.

4. Experimental Test Facility

4.1 Description

A research facility, previously described in detail (Burn & Short 2000; Short 2003), has been developed in the form of a planar robot arm and PC-based open architecture controller. The

robots joints are manufactured from toughened ABS plastic, and are actuated by brushless servomotors (with digital servoamplifiers). The control loop for each axis is closed via a multitasking DSP embedded in a Delta Tau® Programmable Multi-Axis Controller (PMAC) motion control card, installed into the PC. Each axis has an individual PID controller with feed forward control to enable accurate velocity and position profile following. A six-axis force/torque sensor was developed in-house for the project, and is employed in the current study. The robot arm is shown photographically in figure 6, and schematically in figure 7.

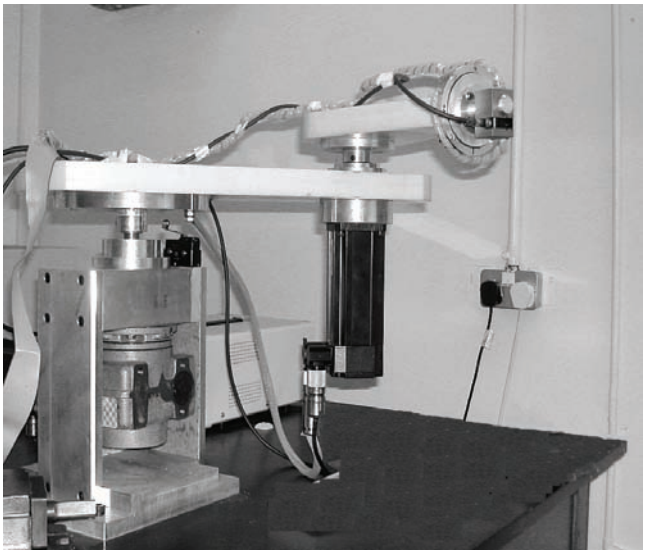


Fig. 6. Robotic test facility

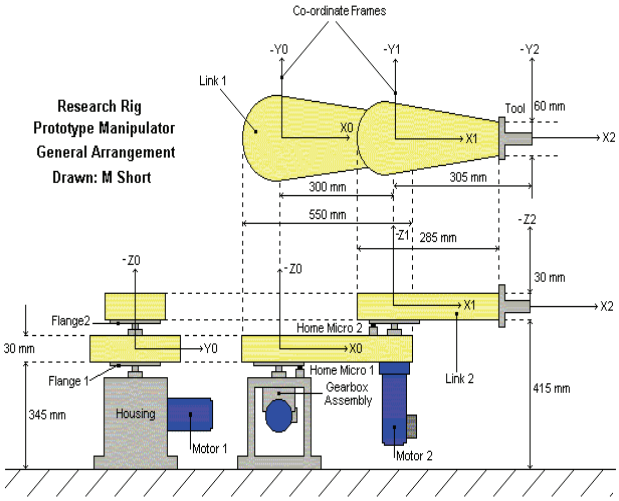


Fig. 7. Schematic of test facility

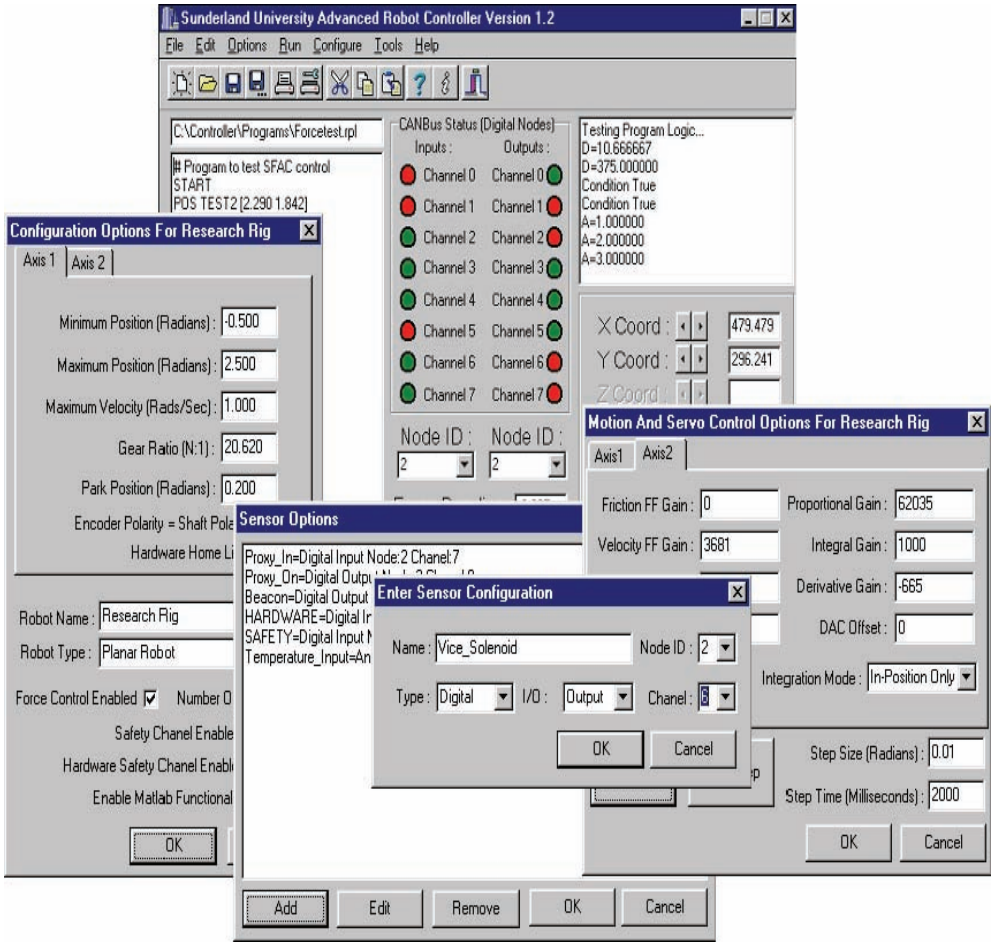


Fig. 8. Application software screenshots

The controller for this robot was developed with a completely 'open architecture' in mind (Ford 1984), with a view to the integration and implementation of novel sensor-based control strategies. The software is based on a three-layered open architecture, as described by Short (2000 & 2003). It features the ability to design and integrate advanced control strategies into the controller, and via the use of an ActiveX® link allows the full functionality of software packages such as Matlab® to be embedded within the controller software. The sensors required to perform these control strategies in real-time are integrated into the system using flexible fieldbus technologies. The underlying kinematic and dynamic models of the robot can be changed to suit the current configuration, allowing the controller to be tailored to any particular arm configuration or drive system. A modular robot programming language - named Sunderland ARm Language (SARL) - was developed for

the controller. Screenshots of the Windows® XP version of the application software are shown in figure 8.

The force controllers described in the previous Section were coded in the C programming language, compiled and added into the controller's modular software component library. Several experiments were then performed using the resulting software. Each experiment involved a specific contact situation, where the robot first approached the contact surface with constant velocity, and subsequently applied a force of 30 N. The contact surface was varied in each experiment, and two surfaces were used; hard (steel) and soft (plastic). In order to reliably detect the contact surface, the end effector was fitted with a Baumer Electric® photoswitch which was calibrated to signal with high accuracy when a solid object was within a distance of 5mm. The sensor was integrated into the controller architecture using a Controller Area Network connection (CAN) fieldbus. The robot was programmed approached the contact surface at a slow jog speed until this signal was made, then switched to force control mode. The sample rate employed was 200 Hz in each experiment, and the measured force was prefiltered using a first-order low pass filter with cut off frequency of 0.01 Hz. In the following section, the parameters that were used to design the controllers are described.

4.2 Controller Design

From a previous identification exercise, the model parameters of each joint of the robot arm and the environment stiffness limits were determined to be as follows (Short 2003):

$$\omega_n = 244\, rad/s,\ \xi = 1, \qquad K_{e\max} = 168\ N/mm,\quad K_{e\min} = 11\ N/mm \tag{9}$$

Using these parameters, the controllers were designed as follows. In both cases, the nominal loop gain K_p was tuned to a value of 0.02 to give the desired transient performance – a 95% rise time of approximately 2 seconds with minimal overshoot. For the fuzzy method, the nominal gain for contact at low K_e could then simply be calculated as follows:

$$K_{plow} = \left(\frac{K_{e\max}}{K_{e\min}} \right) \cdot K_p = 0.3 \tag{10}$$

This information was then used as input to the controller design software (Burn et al. 2004), and the controller tuning algorithm was run for 200 iterations to produce the final fuzzy controller that was utilized in the experiment. The highly non-linear I/O surface of the controller is illustrated by figure 9, which shows a plot of F_e and ΔF_e versus controller output u. In the MFC controller, the value of the nominal loop gain K_p was then used to design the value of $K_p'_{max}$ calculated as given by (7) to have a value of 2.9. A value of K_p' = 1.5 was therefore chosen for the experiments, as this gave good performance and remained well below the stability limit. The experimental comparison of the controllers is discussed in the following Section.

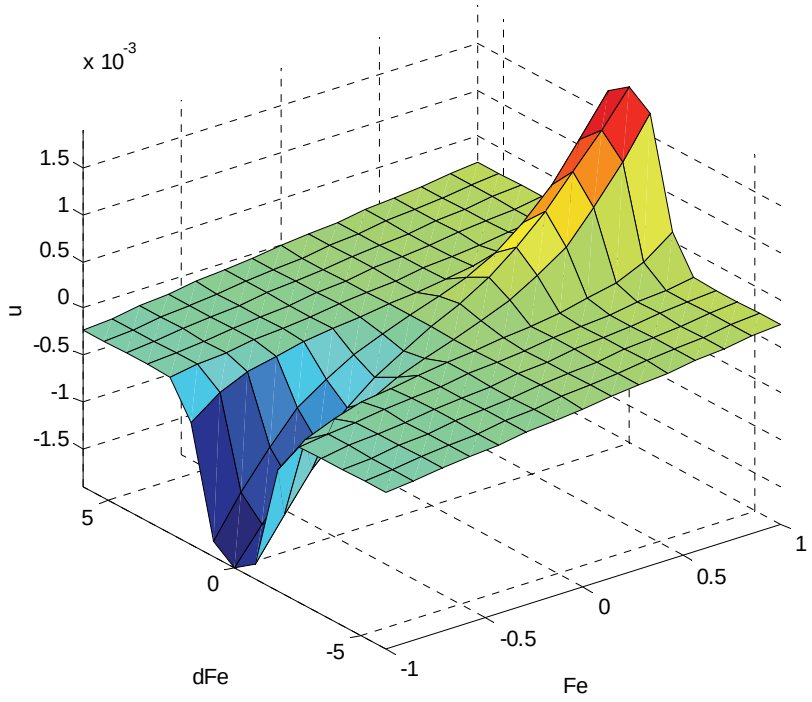


Fig. 9. Fuzzy controller I/O surface

5. Experimental Results and Analysis

This section begins by presenting the results of the contact experiments described in the previous Section, beginning with the FIS-based controller. Figure 10 shows the responses of this controller when applying a force to the hard (steel) and soft (plastic) surfaces. Considering now the MFC-based controller, figure 11 shows the responses of this controller when applying a force to the same surfaces. The very small negative force indicated before contact with the surface was made (at approx 1s) was due to a small drift in the calibration of the force sensor whilst moving in free space.

These figures demonstrate the effectiveness of both approaches; comparing these figures with the responses shown in figure 2, the fixed gain controller, it can be seen that the responses display little sign of instability. The compensation added by the adaptation of loop gains in the FIS controller, and the extra loop and forcing gain in the MFC controller can clearly be seen; in all cases, a very similar transient response is seen. There is a slight overshoot in the response of the fuzzy method when contacting the hard surface, and it can be seen that the steady-state behaviour seems to be slightly less stable than the MFC controller.

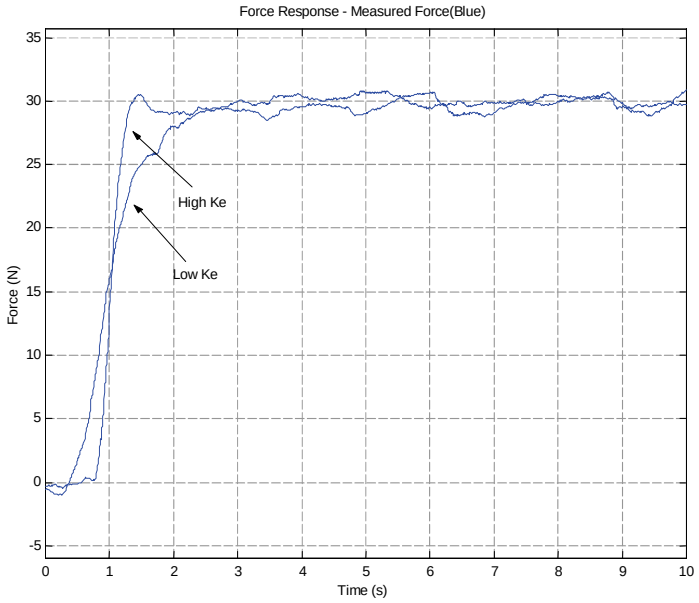


Fig. 10. Contact force profile for the FIS-based controller

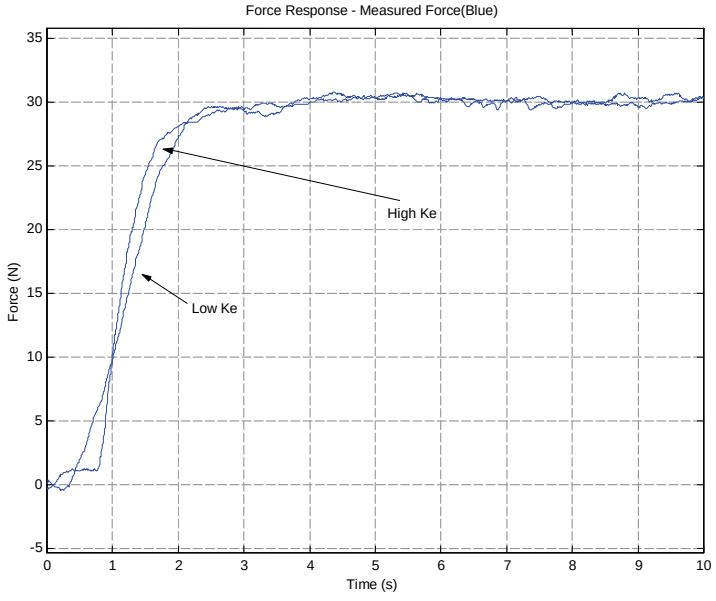


Fig. 11. Contact force profile for the MFC-based controller

In addition to these response measurements, the Integral of Time by Absolute Error (ITAE) for each of the responses was calculated and is shown in table 1. The ITAE is a useful measure of system performance in the time-domain and is given by equation (11) (Franklin et al. 1994). This table includes additional information pertaining to each method, including the approximate Source-Lines-Of-Code (SLOC) needed for each implementation, the measured run-time overheads (i.e. CPU execution time per iteration, in milliseconds) and a relative measure of the 'design effort' needed for each controller. The latter was classified subjectively into either a LOW or HIGH category.

$$ITAE = \int |e|.t.dt \tag{11}$$

	Performance Measures				
Controller	**ITAE (low)**	**ITAE (high)**	**SLOC**	**CPU (ms)**	**Effort**
FIS	227.1	247.1	1000+	2.05	HIGH
MFC	215.6	220.1	<100	0.15	LOW

Table 1. Summary of controller comparative data

The ITAE data further illustrates the performance of both force control methods. As expected, the FIS controller had a higher overall ITAE value for both contact situations, and there was a larger measurable difference between the two response values. This reflects the observation that the FIS controller demonstrated a faster response with a slight overshoot at the higher stiffness gain value. Although a slight difference exists between the responses for the MFC controller, as can be seen from the responses its performance seems more predictable and it does not 'hunt' around the setpoint to the same level as the FIS. This is best attributed to the fact that close to the setpoint, the model error – and hence the forcing control signal – are close to zero. The FIS controller, however, seems to be sensitive to noise in the error signal; especially through the Δf_e input, thus allowing small gain adaptations to take place in the region of the setpoint. These small gain adaptations are magnified by the resulting changes in the measured Δ_p, exacerbating the problem slightly. Despite these observations, it can be argued that the overall performance of both controllers is acceptable for most practical situations.

The SLOC measure gives a rough indication of the complexity of the code required for the implementation of the controllers. In this respect it can be seen that the MFC has a huge advantage over the FIS. As well as the main implementation code, e.g. the calculation of error signals for the controller inputs, the FIS method requires extensive code to implement the fuzzification, logical inference and de-fuzzification subroutines. These will – generally – cause a large increase in the code size, either from the inclusion of specialised library files or from a direct implementation of the underlying mathematical equations. The MFC controller, however, simply requires the updating of a third-order equation and simple addition/subtraction and multiplication to generate the control effort. The increased complexity of the FIS method is reflected by an almost 14-fold increase in the CPU

overheads required at each sample iteration to generate the control signal. In addition, although the design methodology proposed for the MFC controller guarantees its stability at present, no such guarantee can be placed on the FIS-based controller. To summarize, it was reflected that the amount of effort required to implement the FIS method was significantly higher than the MFC, from both the control and software design perspectives.

6. Conclusion

This chapter has been concerned with the practical realisation of robotic force control. It has been shown that many potential difficulties arise when implementing a force control method, including stability and robustness problems associated with applications where environmental uncertainty exists, and with sampling and control limitations related to the basic operation of many tradition robot controllers.

Force control remains an ongoing area of research; however, in recent years a variety of efficient solutions to many of these problems have been proposed. This chapter has considered two novel methods in depth, and has described a series of experiments to perform a direct real-time comparison of the controllers using an open-architecture test facility. Whilst the results generated indicate slightly better behaviour for the MFC-based method, in practical situations both methods were deemed acceptable and a considerable improvement over a fixed-gain controller. Based on the analysis of design effort, system stability and incurred CPU overheads, however, this chapter concludes that the MFC-based controller has significant advantages over the FIS-based design.

Future work in this area will include analysis of situations where PD controllers are used as the MFC loop compensators, and also consider the effects of model mismatch (which is inevitable if the methodology is to be applied to larger-scale industrial robots). Further work will also consider implementation on both controllers on a 6-DOF manipulator to further investigate and contrast the two approaches.

7. References

Bautista, R. & Pont, M.J. (2006). Is fuzzy logic a practical choice in resource-constrained embedded control systems implemented using general-purpose microcontrollers? In *Proceedings of the 9th IEEE International Workshop on Advanced Motion Control*, Istanbul, Vol. 2, pp.692-697.

Bicker, R., Burn, K., Glennie, D. & Ow, S.M. (1994). Application of force control in telerobotics, *Proc Int Conf EURISCON '94*, Malaga, Spain.

Burn, K, Home, G, Short, M. & Bicker, R. (2004). A software tool for automating the design of robot fuzzy force controllers, *Robotica*, Vol. 23(2), pp. 247-256.

Burn, K, Short, M. & Bicker, R. (2003). Adaptive and nonlinear fuzzy force control techniques applied to robots operating in uncertain environments, *Journal of Robotic Systems*, Vol. 20(7), pp. 391-400.

Burn, K. & Short, M. (2000). Development of a generic robot controller architecture for advanced and intelligent robots, In: *Proc. 14th Int. Conf. On Systems Eng. (ICSE 2000)*, Vol. 1, pp. 92-97.

Cao, S.G., Rees, N.W. & Feng, G. (1998). Lyapunov-like stability theorems for continuous-time fuzzy control systems. *Int J Control*, Vol. 69(1), pp. 49-64.

Engelberger, JF. (1980). *Robots in practice*, Kogan Page; London, UK.

Ford, WE. (1984). What is an open architecture robot controller? In: *Proc 9th IEEE Int. Symp. on Intelligent Control*, pp. 27-32.

Franklin, G.F., Powell, J.D. & Emani-Naeini, A. (1994). *Feedback Control Of Dynamic Systems*, Addison-Wesley Publishing, Reading Massachusetts, third edition.

Kiguchi, K. & Fukuda, T. (1997). Intelligent position/force controller for industrial robot manipulators – application of fuzzy neural networks, *IEEE Trans Industrial Electronics*, Vol. 44(6), pp. 753-761.

Kim, W.S., Hannaford, B. & Bejczy, A.K. (1992). Force Reflection and Shared Compliant Control in Operating Telemanipulators with Time Delay, *IEEE Trans on Robotics and Automation*, Vol. 8(2), pp. 176-185.

Li, G., Tsang, K.M. & Ho, S.L. (1998). A novel model following scheme with simple structure for electrical position servo systems, *Int. J. Syst. Sci.*, Vol. 29(9), pp. 959–969.

Lin, S.T. & Huang, A.K. (1998). Hierarchical Fuzzy Force Control for Industrial Robots, *IEEE Trans on Industrial Electronics*, Vol. 45(4), pp. 646-653.

Linkens, D.H. & Nyongesa, H.O. (1996). Learning systems in intelligent control: an appraisal of fuzzy, neural and genetic algorithm control applications, *IEE Proc Control Theory Appl*, Vol. 143(4), pp. 367-386.

Osypiuk, R., Finkemeyer, B. & Wahl, F.M. (2004). Forward-model based control system for robot manipulators, *Robotica*, Vol. 22(2), pp. 155–161.

Ow, S.M. (1997). *Force Control in Telerobotics*, PhD Thesis, University of Newcastle upon Tyne, UK.

Pippard, A.B. (1997). *Response & Stability: An Introduction to the Physical Theory*, Cambridge University Press.

Seraji, H. (1998). Nonlinear and Adaptive Control of Force and Compliance in Manipulators, *Int J Robotics Research*, Vol. 17(5) pp. 467-484.

Short, M. (2003). *A Generic Controller Architecture for Advanced and Intelligent Robots*, PhD. Thesis, University of Sunderland, UK.

Short, M. & Burn, K. (2007). Robust and Stable Robotic Force Control, In: *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics (ICINCO'07)*, Angers, France, May 09-12.

Skoczowski, S., Domek, S., Pietrusewicz, K. & Broel-Plater, B. (2005). A Method for Improving the Robustness of PID Control, *IEEE Trans Industrial Electronics*, Vol. 52(6), pp. 1669-1676.

Tarokh, M. & Bailey, S. (1997). Adaptive fuzzy force control of manipulators with unknown environment parameters, *J Robotic Sys*, Vol. 14(5), pp. 341-353.

Whitney, D.E. (1985). Historical Perspective and State of the Art in Robot Force Control, *Int J Robotics Res*, Vol. 6(1), pp. 3-14.

Whitney, D.E. & Nevins, J.L. (1979). What is the Remote Centre Compliance (RCC) and what can it do? In: *Proc Int Symp on Industrial Robots*, Washington DC, pp. 135-152.

Wolkenhauer, O. & Edmunds, J.M. (1997). A critique of fuzzy logic in control, *Int J Electrical Engineering Education*, Vol. 34(3), pp. 235-242.

Zhang, G. & Hemami, A. (1997). An Overview of Robot Force Control, *Robotica*, Vol. 15, pp. 473-482.

2

Towards a Roadmap for Effective Handset Network Test Automation

Clauirton A. Siebra, Andre L. M. Santos & Fabio Q. B. Silva
CIN/Samsung Laboratory of Research and Development[1] - UFPE
Brazil

1. Introduction

In recent years, wireless networks have presented a significant evolution in their technology. While first-generation networks, based on analog signalling, were targeted primarily at voice and data communications occurring at low data rates, we have recently seen the evolution of second- and third-generation wireless systems that incorporate the features provided by broadband networks (Garg, 2001). In addition to supporting mobility, broadband networks also aims to support multimedia traffic, with quality of service (QoS) assurance. Therefore the evolution from 2G to 3G wireless technology bears the promise of offering a wide range of new multimedia services to mobile subscribers (De Vriendt et al., 2002).

In this context, handset devices are following this ongoing network evolution and taking advantage of this technological update to offer a broad variety of resources and applications to their users. In fact, handset development has evolved into a complex engineering process, mainly because of the recent network capabilities supported by new mobile communication and computer technology advances (memory speed and size, processing power, better resources for information delivery, etc.). This scenario has increased the demand on the test phase of handset development, which is required to apply more extensive and efficient evaluation procedures so that the final product meets the fast time-to-market goals and can compete in the global marketplace.

While the number and complexity of tests are increasing, test centers need to decrease their test execution time. The quicker a specific handset is evaluated and delivered to the market, the better will be its chances to compete with other models. Therefore we have a contradiction: we need to increase the number of tests and decrease the test time. Furthermore, this contradiction can lead us to reduce the quality of our test processes.

Test automation is one alternative to this emerging scenario, because it enables tests to be launched and executed without the need for user intervention. Thus, common delays and errors associated with the manipulation of test parameters by humans can be avoided.

[1] The results presented in this chapter have been developed as part of a collaborative project between Samsung Institute for Development of Informatics (Samsung/SIDI) and the Centre of Informatics at the Federal University of Pernambuco (CIn/UFPE), financed by Samsung Eletronica da Amazonia Ltda., under the auspices of the Brazilian Federal Law of Informatics no. 8248/91.

Furthermore, tests can be continuously run during hours or days, ensuring maximum test coverage.
Considering such issues, this chapter discusses the steps associated with the specification of an automation architecture for network tests of handsets. We begin in Section 2 by presenting the handset test domain and an initial test architecture to support simulation of real network conditions in laboratory. We then look at techniques for dealing with the automation requirements for this initial architecture. Section 3 details the *semantic definition* and standardization of test cases, so that their content can be used in a common way by decision-making processes. Section 4 discusses the use of Intelligent Planning as an option to create appropriate sequences of test cases, based on specific devices, environment features and semantics. Section 5 introduces the concept of *autonomic computing* and shows how *knowledge-based systems* can be used to create a self-managing test process that can deal with unexpected situations. During the test performance, it is important to monitor the test indicators to ensure the process quality and to identify failures and opportunities for improvements. Section 6 considers this issue and discusses our experience in applying DMAIC, a *Six Sigma* framework for statistical monitoring and control of processes, which is an important part of our automation and control architecture. Section 7 presents the idea of using test management tools and how they can aggregate value to the process of automation. Finally, Section 8 concludes this chapter, summarising its main objectives.

2. The Handset Test Domain

Network tests for handsets intend to verify if new devices are able to perform a set of operations pre-defined by carriers over the network. The majority of these operations, such as handovers and reselection of cells, are default operations of any network carrier. However, specific operations can be offered for just one carrier. Note that this kind of evaluation does not test the user experience features of the device, user interface, performance of data applications, battery life and others. Network tests are mainly focused on conformance (protocol performance, application enabler and radio components) and interoperability testing.
Each of the individual layers (Fig. 1) of a handset implementation requires a different battery of tests, which are performed in specific scenarios. A scenario is a collection of cell configurations and other options, which can represent parts of artificial or "real-life" networks. There are specific tools (e.g., Sagem, Nemo and Tems) for network information capture, which are able to read and save network configurations so that we can simulate scenarios of these networks in laboratory.

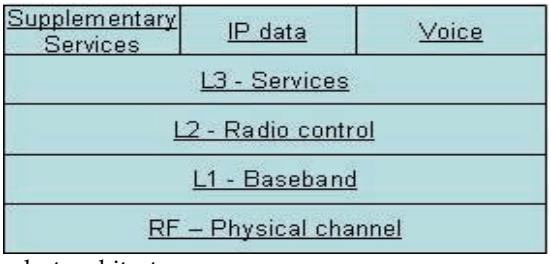


Fig. 1. Layers of a handset architecture

The scenarios that we are working with are based on the GSM standard. GSM is a cellular network, which means that handsets connect to it by searching for cells in their immediate vicinity. The main feature of this network is its operation frequency ranges. Most GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas (including the United States and Canada) use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated.

Handsets usually have to be evaluated in more than one scenario. However, some test cases are not appropriate to every scenario. For example, consider a scenario with two cells: $cell_1$ and $cell_2$. Using only the frequency ranges of 900 MHz and 1800 MHz, we could set three different scenarios: ($cell_1$= $cell_2$= 900MHz), ($cell_1$ = $cell_2$ = 1800MHz) and ($cell_1$ = 900MHz, $cell_2$ = 1800MHz). However frequency range is only one of the parameters that can be set to cells, therefore a considerable variation of scenarios can be specified.

Considering that we have hundreds of test cases to be performed in several different scenarios, the use of *field-testing*, which uses real carriers' networks, can be expensive, especially in terms of travel cost, time and potential disruption of real networks. The use of *field-testing* also makes it impossible to guarantee the reproducibility of the same test environment in different runs of the same test. Such facts have motivated the development of simulation environments for wireless networks, which support the performance of a significant variety of network tests. This section, in particular, describes our experience with the Anite/SAS network simulator (Anite, 1999).

The Anite/SAS environment enables the handset communication within any network configuration of up to eight base stations under laboratory conditions. A PC controller runs the network simulation software and controls the base station modules (in our case, two Agilent 8960 test sets). The test sets interact with the handset being tested via a normal RF interface. To test end-to-end communication, the PC can be connected to the Internet, allowing data to be transmitted to the handset. Using this configuration, for example (Fig. 2), we are able to emulate a two-cell scenario and perform tests that involve handovers, cell selection and cell reselection for circuit-switched, packet-based and dual transfer mode (DTM) services over GSM.

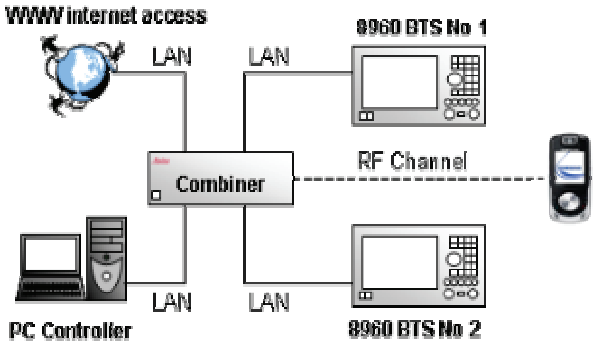


Fig. 2. Test simulation environment

An important feature of network simulation environments is their flexibility in configuring cells. Currently we are creating artificial scenarios via the simulator's cells editor. Using this

editor, for example, we can set the cell identity, channel descriptions and frequency list. The collection of all these parameters defines a cell and the collection of all cells defines a scenario. Scenarios can be saved, loaded and several different scenarios can be used to test a handset. In an extension of this project, we intend to use "real-life" scenarios, creating a database with real network information of several Brazilian cities.,

3. A Semantic Standardization for Tests Cases

3.1 A Classification for Network Tests

The main sources of information to create a suite of tests are the carrier's requirements. Indeed, if carriers account for creating and maintaining wireless network services, it is natural that they also provide the requirements that a handset must implement to operate over their networks. However, the tests provided by different carriers are not uniform and present several variations related to their procedures. Thus, we needed to carry out a process of information extraction and consolidation to create a test suite proposal. There are some efforts in the direction of unifying the performance of network tests via the creation of certifications centers. These efforts vary with different technologies and geographies. For example, the procedure for evaluating the GSM technology (Rahnema, 1993) is managed by the PCS Type Certification Review Board in the U.S., while the Global Certification Forum does this in other countries. Furthermore, groups like that also elaborate documents, such as the 3GPPTS 51.010-1, which contains collections of test cases for handset validation. We have also used similar documents to support and aggregate content to our test suite.

In order to consolidate all the information from carriers and certification documents we have elaborated a classification that currently has 25 test batteries. The test cases were mainly classified according to technological aspects and kind of services that they are supposed to evaluate. Examples of test batteries are: Air Signaling Tests (RF), Enhanced Messaging Service, Supplementary Services, Data Connection, Multimedia Messaging Service and Wireless Application Protocol.

3.2 The Test Case Specification

The next step, after the test cases classification, is to identify and organize the common concepts related to test cases, so that they can be represented in a unified way. From our analysis, we have defined the following concepts (Fig. 3):

- Test Case Identifier – a unique key that identifies a test case (TC) among the entire test case collection. The pattern proposed in our project is <xx_TCyyy>, where "xx" represents the battery identifier and "yyy" represents the TC number. For example, SM_TC003 is the third test case from the Short Message (SM) battery;
- Description – a short description of the main function of a test case;
- Preconditions – required initial conditions to the execution of a test case;
- References – indicate the files, such as the test scripts, associated with a test case;
- Estimated time – average time required to perform the test case;
- List of test case steps – steps are represented by the 4-tuple <step-number, technique, procedure, expected-result>, where step-number is a sequential integer value; technique specifies if the step is manual or automatic; procedure is the description of the step; and expected-result is any verification to be performed on the handset display or on the simulator execution log.

3.3 Mapping Test Cases to Simulator Scripts

Before running the test in our simulation environment, we need to map each test case to a script, using the simulator language. An example of a script, which represents the mapping of the RF_TC005 test case (Fig 3), is illustrated in follow (Fig. 4).

IDENTIFIER	RF_TC005
DESCRIPTION	Handover between two 1800 MHz cells.
PRECONDITION	Handset with SIMCard and connected to Anite Simulator. Both 1800 MHz cells must be created in the Anite Simulator and activated.
REFERENCES	PI_SCP001.psc
ESTIMATED TIME	00:02:20

STEPS	TECHNIQUE	PROCEDURE	EXPECTED RESULT
1	Manual	Switch handset on.	Verify the following message in the Anite Simulator log: "G-Cell B MS->SS RACH CHANNEL REQUEST"
2	Automatic	Set the LOCATION UPDATE to cell B.	Verify the correct exhibition of the SPN at handset's display and verify the following message in the Anite Simulator log: "G-Cell B SS->MS SDCCH/4 LOCATION UPDATING ACCEPT"
4	Manual	Make a voice call.	Verify the following message in the Anite Simulator log: "G-Cell B SS->MS SDCCH/4 CALL PROCEEDING"
5	Automatic	Answer the voice call.	Verify the following message in the Anite Simulator log: "G-Cell B SS->MS FACCH CALL CONNECT"
6	Automatic	Execute a HANDOVER to the cell A.	Verify the following message in the Anite Simulator log: "G-Cell A SS <- MS FACCH HANDOVER COMPLETE"
8	Manual	Deactivate the voice call.	Verify the following message in the Anite Simulator log: "G-Cell A MS->SS FACCH DISCONNECT"
9	Manual	Deactivate the handset.	Verify the following message in the Anite Simulator log: "G-Cell A MS->SS SDCCH/4 IMSI DETACH INDICATION"
10	Automatic	Deactivate cell A and cell B.	Verify the following message in the playback window in the Anite Simulator: "Verdict : PASS"

Fig. 3. Example of test case specification. This one refers to test case number 5 from the Radio Frequency battery.

● □	SS ↔ SS	00:00.00		LOAD SCENARIO DATA
	SS → MS	00:00.00	G-Cell A	ACTIVATE CELL [C0ARFCN = 700 | PWR = -75 | FREQ = 0 | SYNC = 0 | BER = 0.00]
	SS → MS	00:00.00	G-Cell B	ACTIVATE CELL [C0ARFCN = 705 | PWR = -50 | FREQ = 0 | SYNC = 0 | BER = 0.00]
	SS ↔ SS	00:00.00		USER PROMPT [Switch the handset ON]
□	SS ← MS	00:00.00	G-Cell A	LOCATION UPDATE
	SS ↔ SS	00:00.00		USER PROMPT [Please initiate voice speech]
□	SS ← MS	00:00.00	G-Cell A	SPEECH CALL [TI = 0] [123] [Speech]
□	SS → MS	00:00.00	G-Cell A	ANSWER CALL [TI = 0]
□	SS → MS	00:05.00	G-Cell B	HANDOVER
	SS ↔ SS	00:00.00		USER PROMPT [Please clear the voice call]
□	SS ← MS	00:00.00	G-Cell B	CLEAR CALL [TI = 0]
	SS ↔ SS	00:00.00		USER PROMPT [Please switch the handset OFF]
	SS → MS	00:00.00	G-Cell B	DEACTIVATE CELL
	SS → MS	00:00.00	G-Cell A	DEACTIVATE CELL

Fig. 4. Simulator playback of the RF_TC005.

Let's comment some details about this script and its relation to the test case (Fig. 3). First, the initial command in the script (LOAD SCENARIO DATA) loads a network scenario, which must be previously defined in the simulator. Specific parameters of this scenario can be changed, according to preconditions of the test case (Fig. 3). For example, this script sets the power value of Cell A to -75dBm and Cell B to -50dBm. Second, automatic steps of the test case can be directly mapped to simulator commands. For example, step 2 (Set the LOCATION UPDATE to cell A) is mapped to the LOCATION UPDATE command. Commands like that are pre-defined in the simulator language. Finally, manual steps are mapped to USER PROMPT commands. These commands are, in fact, requests to testers so that they execute some specific command in the handset, such as *Switch the handset ON*.

3.4 Using a Formal Description

Whereas test cases documentation (Fig. 3) is understandable and simple to human readers, since they use natural language, scripts (Fig. 4) are defined in a proprietary language, whose syntax is defined to increase the performance of the simulator. We need a new description that considers such features together. This means that the description needs to have a good semantic level and be easily manipulated by computational components. Consider, for example, the concept of *ontology* (Gruber, 1995). An ontology defines a set of representational primitives with which to model a domain of knowledge. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In this way, an ontology can also be viewed as an abstraction of data models, analogous to hierarchical and relational models, but intended for modelling knowledge about individuals, their attributes, and their relationships to other individuals. For this reason, an ontology is said to be at the semantic level of a system description.
We do not intend to develop a complete ontology for handsets' network test cases. However we can use some fundaments to design a test case specification, which support the execution of a decision process by the test system. One of the main ideas of an ontology is to define the classes, attributes and relationships of a domain (Guarino, 1995). For that end, consider the BNF metasyntax, used to express context-free grammars and, thus, a formal way to describe formal languages. Using BNF we can create the following specification:

<test-case> ::= <tc-head> <list-of-steps>
<tc-head> ::= <identifier> <description> <list-of-preconditions>
<list-of-references> <estimated-time>
<list-of-steps> ::= <list-of-steps> | <step>
<step> ::= <step-number> <technique> <procedure> <expected-result>

This is just the first level of a specification. The complete specification must decompose the classes to primitive concepts. Note that descriptions like these allow us to abstract away from data structures and implementation strategies. However we can approximate such a description to computational structures if we apply a direct mapping from BNF to a marked

language, such as XML. For example, consider the XML version of the above BNF description:

```
TEST-CASE :: = <test-case>
                    <tc-head > TC-HEAD </tc-head>
                    <list> <step> STEP </step> </list>
               </test-case>
TC-HEAD ::=    <tc-head id=”SYMBOL” description=”NAME” estimated-time = “TIME”>
                    <list> <precondition> PRECONDITION </precondition> </list>
                    <list> <reference> REFERENCE </reference> </list>
               </tc-head>
STEP ::=       <step step-number=”NUMBER” technique=“SYMBOL”>
                    <procedure> PROCEDURE </procedure> </list>
                    <expected-result> EXPECTED-RESULT </ expected-result >
               </step>
```

Several ontological concepts are represented in such a description. For example, TEXT-CASE, TC-HEAD and STEP are examples of classes. TC-HEAD has three attributes: *id*, *description* and *estimated-time*. Finally, the definition structure itself defines the relations among the classes. For example, TEST-CASEs contain one TC-HEAD and a non empty list of STEPs. Modern programming languages, like Java, provide a strong support for the manipulation of XML-based descriptions via, for example, packages to create parsers and objects that represent the classes and attributes defined in the description. The next section shows how this kind of description can be used during the test process.

4. Planning the Test Suite

As discussed before, test automation is an appropriate technique to deal with the current handset test scenario. We have observed that the common approach used in test automation is to employ pre-defined recorded test cases, which can be created and edited in tools provided by simulation environments. Such tools also enable the building of different and more complex tests from previous ones. However, this entire process is carried out manually. Thus, we have an automatic execution of tests, but using fixed sequences of test cases, so that the sequence is not automatically adapted to specific scenarios.

The idea explored in this section is to develop a mechanism that autonomously creates test cases based on devices and environment features. This mechanism must specify the most appropriate sequence of test cases for a specific handset, also minimizing the total test time. For that end we have investigated the *Artificial Intelligence* (AI) *Planning* technique (Ghallab et al., 2004), which is used to create optimal sequences of actions to be performed in some specific situation. One of the advantages of AI planning is its modelling language, which has a similar syntax to the language used for test case description. Furthermore, the language is open, providing a direct way to represent states (situations of tests), goals (desired results of tests) and actions (steps of each test case). This section details these issues, starting with a brief introduction to AI planning and its fundamentals.

4.1 Fundamentals of AI Planning

AI Planning can be viewed as a type of problem solving in which a system (Planner) uses beliefs about actions and their consequences to search for a solution over a space of plans or states. The key idea behind planning is its open representation of states (e.g., test case scenarios), goals (e.g., expected results of test cases) and actions (e.g., steps of a test case). States and goals are represented by sets of sentences, and actions are represented by logical descriptions of preconditions and effects. This enables the planner to make direct connections between states and actions. The classical approach to describe plans is via the STRIPS language (Fikes & Nilsson, 1971). This language represents states by conjunctions of predicates applied to constant symbols. For example, we can have the following predicates to indicate the initial state of a handset network: FreqRange($cell_a$,900) $\wedge$ FreqRange($cell_b$,1800). Goals are also described by conjunctions of predicates, however they can contain variables rather than only constants. Actions, also called operators, consist of three components: action descriptions, conditions and effects. Conditions are the (partial) mandatory states to the application of the operator and effects are the (partial) final state after its application. Using these basic definitions, the representation of plans can be specified as a data structure consisting of the following four elements (Wilkins, 1994):

- A set of plan steps, each of them representing one of the operators;
- A set of tasks ordering constraints. Each ordering constraint is of the form S_i » S_j, which is read as "S_i before S_j" and means that step S_i must occur sometime before step S_j;
- A set of variable binding constraints. Each variable constraint is of the form v = x, where v is a variable in some task and x is either a constraint or another variable; and
- A set of *causal links*. A causal link is written as $\{T_i \rightarrow T_j\}^c$ and read as "Ti achieves c for T_j". Causal links serve to record the purpose(s) of steps in the plan. In this description, a purpose of T_i is to achieve the precondition c of T_j.

Using these elements, we can apply a *Partial-Order Planning* (POP) algorithm, which is able to search through the space of plans to find one that is guaranteed to succeed. More details about the algorithm are given later.

4.2 Test Case as Planning Operators

Let us consider now the process of mapping a test case to a planning method. One of the parts of this research is to codify each of the user cases in a plan operator. As discussed before, a plan operator has three components: the action description, the conditions and the effects. Therefore we need to find information inside user tests to create each of these components. Starting by the action description, this component is just a simple and short identifier that does not play any active role during the decision process of the planner. For its representation we are issuing identifiers that relate each action description with a unique test case TC. Second, we need to specify the operator conditions. The test case definition has an attribute called precondition that brings exactly the semantic that we intend to use for operator conditions. Note, however, that this description is a natural language sentence that must be translated to a logic predicate before being used by the planner. Finally, the effects do not have a direct mapping from some component of the test case descriptor. However, each step of the test case (e.g., Table 1) is one action that can change the current state of the domain. Consequently, the effects can be defined by the conjunction of the expected results of each step, if this step has the feature of changing the domain status. Another observation is that sequential steps can change a given status. Thus, only the last change of a status must

be considered as an effect. As discussed for preconditions, this conjunction of plan steps results also must be codified in logic predicates.

Step	Procedure	Expected result
01	Switch handset on.	"G-Cell B MS->SS RACH CHANNEL REQUEST"
02	Set the LOCATION UPDATE to cell B.	"G-Cell B SS->MS SDCCH/4 LOCATION UPDATING ACCEPT"
03	Make a voice call.	"G-Cell B SS->MS SDCCH/4 CALL PROCEEDING"
04	Execute HANDOVER to the cell A.	"G-Cell B SS->MS FACCH CALL CONNECT"
05	Execute HANDOVER to the cell A.	"G-Cell A SS <- MS FACCH HANDOVER COMPLETE"
06	Deactivate voice call.	"G-Cell A MS->SS FACCH DISCONNECT"
07	Deactivate the handset.	"G-Cell A MS->SS SDCCH/4 IMSI DETACH INDICATION"
08	Deactivate cells and B.	"Verdict : PASS"

Table 1. Partial specification of a test case.

A close investigation into the test cases shows that they are not dealing with operations related to changes in some of the network domain parameters, such as frequency band or number of transceivers in each cell. To have a complete automation of the handsets' network tests, the planner needs special operators that are able to change such parameters. Considering this fact, the set of automation operators can be classified into two groups: the *Test Case Operators* (TCO) and the *Domain Modify Operators* (DMO). The use of both operators is exemplified in the figure below (Fig. 5).

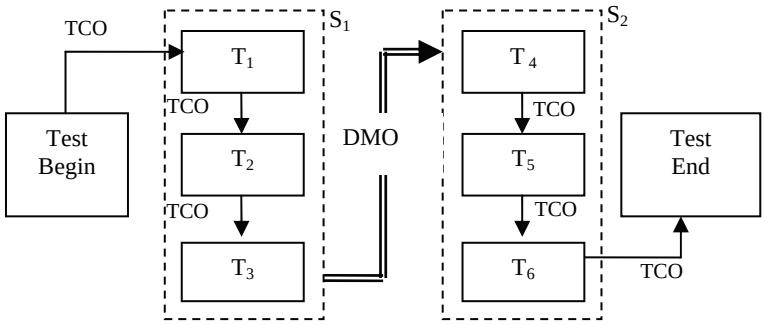


Fig. 5. Test case for a handover between two cells of 1800 MHz.

During the planning process, a planner should normally consider all the test case operators (TCO) of a scenario S_1 before applying a DMO and change to S_2. Actually the number of changes between scenarios (or use of DMO's) can be a measure of the planner performance.

A complete test case plan is a sequence of these operators in which every precondition, of every operator, is achieved by some other operator.

4.3 Planning Algorithm

The reasoning method investigated during this project is described via a *Partial-Order Planning* (POP) algorithm (Nguyen & Kambhampati, 2001). A POP planner has the ability of representing plans in which some steps are ordered with respect to each other while other steps are unordered. This is possible because they implement the principle of least commitment (Weld, 1994), which says that one should only make choices about things that it currently cares about, leaving the other choices to be worked out later.
The pseudocode below (1 to 4) describes the concept of a POP algorithm, showing how it can be applied to the handsets' network test domain. This pseudocode was adapted from the original POP algorithm (Russel & P. Norvig, 2002).

```
function POP(testBegint,testEnd,operators) return plan
  plan ← Make-Basic-Plan(testStart, testFinish)
  loop do
    if Solution?(plan) then return plan                          (1)
    Ti, c ← SELECT-SUBGOAL(plan)
    CHOOSE-OPERATOR(plan,operators, Ti,c)
    RESOLVE-THREATS(plan)
  end
```

Code (1) shows that the POP is a loop that must create a plan from a *testBegin* state (no tests performed) to a *testEnd* state (all tests performed). For that end, the loop extends the plan by achieving a precondition *c* of a test case T_i, which was selected as a subgoal of *plan*.
Code (2) accounts for the selections of this subgoal:

```
function SELECT-SUBGOAL(plan) returns Ti,c
  pick a test stateTi from TEST_STEPS(plan)                      (2)
    with a precondition c that has not been achieved
 return step Ti,c
```

Code (3) details the choice of an operator T_{add} (TCO or DMO), which achieves *c*, either from the existing steps of the plan or from the pool of operators. Note that the causal link for *c* is recorded together with an ordering constraint. If T_{add} is not in TEST_STEPS (T_i), it needs to be added to this collection.
We can improve the SELECT-SUBGOAL function by adding a heuristic to lead the choice of a test goal. During the test process, if one of the tests fails, the problem must be fixed and all the test collection carried out again. Imagine an extreme scenario where a problem is detected in the last test. In this case, the test process will take about twice the normal time to be performed if any other problem is found. Considering this fact, the SELECT-SUBGOAL function could identify and keep track of the tests where errors are more commons. This could be implemented via a module of learning (Langley & Allen, 1993), for example. Based

on this knowledge, the function should give preference to tests with a bigger probability to fail because then the test process will be interrupted earlier.

```
procedure CHOOSE-OPERATOR(plan,operators, Ti,c)
  choose a step Tadd from operators or TEST_STEPS (plan) that has  c as an effect
  if there is no such step then fail
  add the causal link {Tadd→ Ti }c to LINKS(plan)
  add ordering constraint Tadd » Ti to ORDERINGS(plan)                    (3)
  if Tadd is a newly added step from operators then
    add Tadd to TEST_STEPS (Ti)
    add Start » Tadd » Finnish to ORDERINGS(plan)
  end
```

The last procedure, Code (4), accounts for resolving any threats to causal links. The new step T_{add} may threaten an existing causal link or an existing step may threaten the new causal link. If at any point the algorithm fails to find a relevant operator or fails to resolve a threat, it backtracks to a previous choice point.

```
procedure RESOLVE-THREATS(plan)
  for each Tthreat that threatens a link  {Ti→Tj}c in LINKS(plan) do
    choose either                                                          (4)
      Promotion: Add Sthreat » Si to ORDERINGS(plan)
      Demotion: Add Sj » Sthreat to ORDERINGS(plan)
    if not CONSISTENT(plan) then fail
  end
```

POP implements a regression approach. This means that it starts with all the handset network tests that need to be achieved and works backwards to find a sequence of operators that will achieve them. In our domain, the final state "Test_End" will have a condition in the form: $Done(TC_1) \wedge Done(TC_2) \wedge \ldots \wedge Done(TC_n)$, where n is the total number of test cases (TC). Thus, the "Test_Begin" has a effect in the form: $\neg Done(TC_1) \wedge \neg Done(TC_2) \wedge \ldots \wedge \neg Done(TC_n)$. In this way, all TCOs must have an effect in the form $Done(TC_i)$, where i is an integer between 1 and n. The other effects of each TCO change the current plan state, restricting the operations that can be used. If there is a fail, this could indicate that a DMO must be applied to change the scenario (network parameters).

According to [VanBrunt, 1993], the test methodology to handset network, which is based on GSM, are focused on conformance evaluations used to validate the underlying components of the air interface technology. However, the launch of new technologies, such as the WCDMA (*Wideband Code Division Multiple Access*), could change the current way that tests are performed, requiring, for example, a more progressive and integrated approach to evaluation of user equipments. Considering this fact, features like maintenance and extensibility must be considered during the development of test automation. In our approach, any new requirement can be easily contemplated via the creation of new operators. New network configurations could also be defined via the definition of new scenarios and DMOs that change the conditions of test applications.

5. Automation and Autonomic Architectures and Autonomic Computing

Automation test architectures intend to support the execution of tests by computational processes, independently from human interference. This automation considers a pre-defined and correct execution, so that concepts such as adaptation and self-correction are not generally contemplated. The second and more complex level of automation is characterised by systems that present some level of autonomy to take decisions by themselves. This kind of autonomic computing brings several advantages when compared with traditional automation and several approaches can be employed to implement its fundaments. All these issues are detailed in this section.

5.1 The CInMobile Automation Tool

We have implemented an automation environment by means of CInMobile (*Conformance Instrument for Mobiles*). This tool aims to improve the efficiency of the test process that is being carried out over the network simulation environment. The CInMobile architecture is illustrated in follow (Fig. 6), where its main components and their communication are presented. As discussed before (see Fig. 4), simulator scripts can have prompt commands that request some intervention from human testers. The first step of our approach was to lead such commands to the serial port so that they could be captured by an external process called *Handset Automator* (HA), running in a second computer. In this way, the HA receives prompt commands, as operations requests, and analyzes the string content to generate an appropriate operation, which is sent to the handset in evaluation. The HA can use pre-defined scripts from the *Handset Script Base* during this operation. The HA also accounts for sending synchronization messages to the SAS software, indicating that the requested handset operation was already carried out.

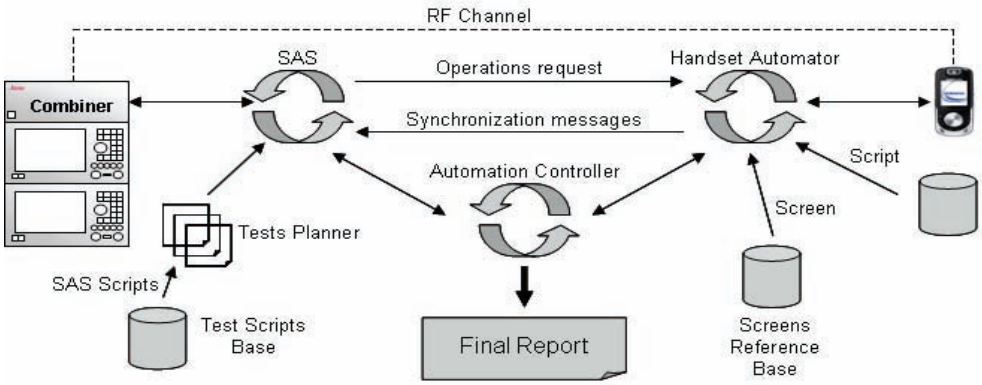


Fig. 6. CinMobile Architecture.

Both SAS and HA generate logs of the test process. However, results of several handset operations, over the simulated wireless network, can be identified via comparisons of resulting handset screens to reference images. Our architecture also supports this kind of evaluation and its results are considered together with the SAS and HA logs. In fact, each of these results (images comparison, SAS and HA logs) brings a different kind of information that must be analyzed to provide useful information about a particular test case.

One of the functions of the *Automation Controller* (AC) process is to consolidate all the test process resulting information, performing a unified analysis so that a unique final report can be generated (Fig. 7). The advantage of this approach is that we are passing the responsibility of reporting the test results from the human testers to an automatic process. Furthermore, the *Final Report Generator* (see Fig. 7) can be used to customize the final report appearance (e.g., using templates) according to user needs.

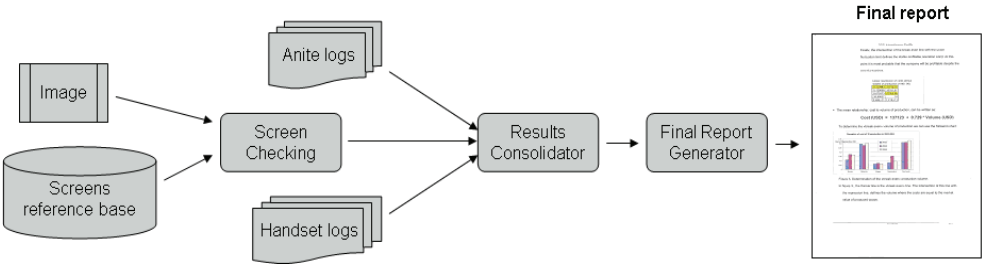


Fig. 7. Test report generation process.

A second important AC function is associated with the control of script loading in simulator. This is useful, for example, because some of the tests must be repeated several times and there is an associated approval percentage. For instance, consider that each test is represented by the 3-tuple $<t,\eta,\varphi>$, where t is the test identifier, η is the number of test repetitions for each device, and φ is the approval percentage. Then a 3-tuple specified as $<t_1,12,75\%>$ means that the t_1 must be performed 12 times and the device will only be approved if the result is correct at least in 9 of them. However, if the 9 first tests are correct, then the other 3 do not need to be executed, avoiding waste of time.

A last AC core function is to manage the performance of several handsets automators. Some tests cases, mainly found in the Bluetooth battery, require the use of two of more handsets to evaluate, for example, operations of voice conference (several handsets sharing the same *Voice Traffic Channel*) over the network. In this scenario, the AC accounts for the tasks of synchronization, among simulator and handset automators, and consolidation of multiple handset logs during the generation of test reports.

5.2 Autonomic Computing

The test process, carried out by our Test Center team, currently provides a percentage of about 53% of automation. However the level of automation provided to such tests is not enough to support a total *autonomic execution*, that is, without the presence of human testers. Considering this fact, our aim is to implement new methods that enable a total autonomic execution, so that tests can be executed at night, for example, increasing the use time of the simulation environment per day and, consequently, decreasing the total test time and operational cost. Our fist experiment was carried out with the E-mail battery. This battery was chosen because it is currently 100% automatic and represents about nine hours of test time. Thus, we could decrease in about one day the test process if such a battery could be performed at night. These new sets of tests, that can run without the human presence, we call *autonomic tests*.

The study of autonomic computing was mainly leaded by IBM Research and a clear definition is (Ganek & Corbi2003): "Autonomic computing is the ability of systems to be more self-managing. The term autonomic comes from the autonomic nervous system, which controls many organs and muscles in the human body. Usually, we are unaware of its workings because it functions in an involuntary, reflexive manner -- for example, we do not notice when our heart beats faster or our blood vessels change size in response to temperature, posture, food intake, stressful experiences and other changes to which we're exposed. And, by the way, our autonomic nervous system is always working".
An autonomic computing paradigm must have a mechanism whereby changes in its essential variables can trigger changes in the behavior of the computing system such that the system is brought back into equilibrium with respect to the environment. This state of stable equilibrium is a necessary condition for the survivability of a system. We can think of survivability as the system's ability to protect itself, recover from faults, reconfigure as required by changes in the environment, and always to maintain its operations at a near optimal performance. Its equilibrium is impacted by both the internal and external environment.
An autonomic computing system (Fig. 8) requires: (a) sensor channels to sense the changes in the internal and external environment, and (b) return channels to react to and counter the effects of the changes in the environment by changing the system and maintaining equilibrium. The changes sensed by the sensor channels have to be analyzed to determine if any of the essential variables has gone out of their viability limits. If so, it has to trigger some kind of planning to determine what changes to inject into the current behavior of the system such that it returns to the equilibrium state within the new environment. This planning would require knowledge to select the right behavior from a large set of possible behaviors to counter the change. Finally, the manager, via return channels, executes the selected change. Thus, we can undersdand the operation of an aunomic system as a continuous cycle of sensing, analyzing, planning, and executing; all of these processes supported by knowledge (Kephart & Chess, 2003).

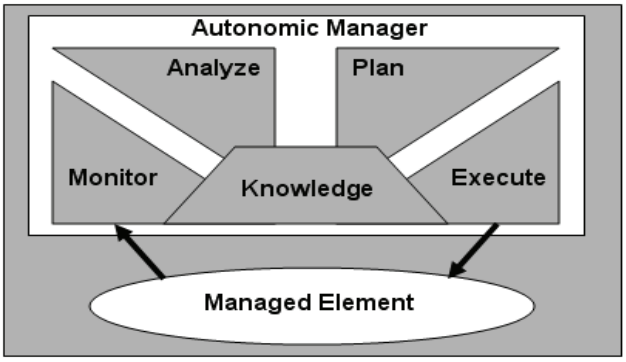


Fig. 8 – The classical autonomic computing architecture.

This classical autonomic architecture (Fig. 8) acts in accordance with high-level policies and it is aimed at supporting the principles that govern all such systems. Such principles have been summarized as eight defining characteristics (Hariri et al, 2006):

- Self-Awareness: an autonomic system knows itself and is aware of its state and its behaviour;
- Self-Protecting: an autonomic system is equally prone to attacks and hence it should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity;
- Self-Optimizing: an autonomic system should be able to detect performance degradation in system behaviour and intelligently perform self-optimization functions;
- Self-Healing: an autonomic system must be aware of potential problems and should have the ability to reconfigure itself to continue to function smoothly;
- Self-Configuring: an autonomic system must have the ability to dynamically adjust its resources based on its state and the state of its execution environment;
- Contextually Aware: an autonomic system must be aware of its execution environment and be able to react to changes in the environment;
- Open: an autonomic system must be portable across multiple hardware and software architectures, and consequently it must be built on standard and open protocols and interfaces;
- Anticipatory: an autonomic system must be able to anticipate, to the extent that it can, its needs and behaviours and those of its context, and to be able to manage itself proactively.

An autonomic manager component does not need to implement all these principles and the choice for one or more depends on the kind of managed element that we are working with. For example, the implementation of the Self-Protecting principle only makes sense if we are working with systems that require a high level of security, such as Internet or network systems. In our case, in particular, we are initially focusing our investigation on three of these principles: Self-Awareness, Self-Healing and Contextual Awareness.

We can find several similarities if we compare the classical autonomic computing architecture to the structure (Fig. 9) of an intelligent utility-based agent (Russel & Norvig, 2002). First, both systems present specific components to sense (sensors channels) the environment (or managed element) and to execute operations (return channels) on such an environment. Second, the knowledge of autonomic computing architectures can be compared to the knowledge of agents (knowledge about its state, how the world evolves, what its actions do and utility of decisions). Finally, in both cases, such knowledge supports the process of analysis and planning of actions, which are going to be executed on the environment.

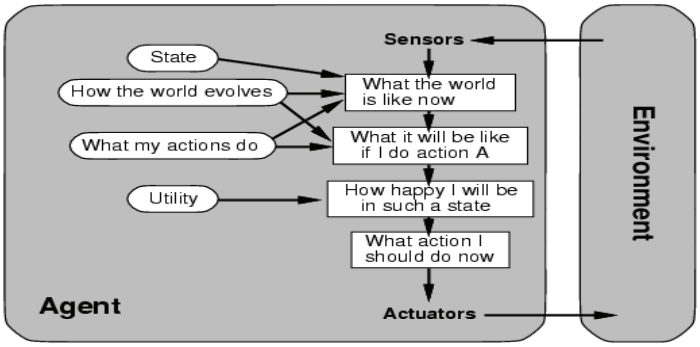


Fig. 9. Utility-based agent structure.

The autonomic manager module, which is in development by our team, is based on this agent approach. For that end, we are specifying the five components of the autonomic architecture (monitor, knowledge base, analyzer, planner and executor) in the following way:

- Monitor – this component is a listener of exceptional events (e.g., Java exceptions), variable content updating (e.g., serial port identifiers) and temporal references (when specific no-exceptional start events are flagged, a timer is started to check if such events finish within pre-defined intervals) ;
- Knowledge base – set of objects that represent both a collection of facts and rules. Facts abstract the current status of the managed element, whereas rules mainly indicate what the systems must do if some event is flagged;
- Analyzer – events received by the monitor need to be analysed in a holistic way. This means, rather than analysing some event individually, we must perform such analysis considering the current status of the knowledge base. For this analysis we use a set of analysis rules, which are part of the knowledge base;
- Planner – analysis rules insert new facts into the knowledge basis, which are used by the planner to decide the actions that are going to be performed. The planner is in fact a set of rules, which we call decision rules;
- Executor – this component has a set of methods that are able to modify the state of the managed element. This set of methods is limited and must be predefined, according to a previous study of the system.

The knowledge base, analyzer and planner are being specified as a production system and for that we are using JEOPS - *Java Embedded Object Production System* (Filho & Ramalho, 2000). The main reason for its use is its first-order forward-chaining inference approach, which starts with the available facts and uses inference rules to extract more facts until an appropriate action is reached. Another reason is its complete integration with Java, which is used in the development of our software. Such components together provide the mechanisms to support the principles of Self-Awareness (knowledge base represents an internal state and updates it), Self-Healing (rules identify problems and trigger recovering methods) and Context Awareness (rules consider the current context once events are analysed in a holistic way).

6. Automation Monitoring and Control via DMAIC Concepts

This section discusses the use of DMAIC (Define, Measure, Analyze, Improve, and Control) (Simon, 2007), a Six Sigma (Harry, 1998) framework based on measures and statistical analysis, which has commonly been applied during several stages of software development (Biehl, 2004). We show that, using DMAIC, we could be able to both find out automation process failures and identify potential points in a test process that require a review. Furthermore, we could monitor the quality of the test process, as discussed below.

6.1 Six Sigma and DMAIC Concepts

Six Sigma is a set of practices to systematically improve processes by eliminating its defects. For that, Six Sigma stresses two main points: processes can always be measured, analyzed, improved and controlled; and continuous efforts to reduce variation in process outputs are essential to business success.

In this discussion we are interested in the statistical fundaments of Six Sigma. In fact, one of the features of its frameworks is the use of measurements and statistical analysis. However, it is a mistake to view the core of Six Sigma frameworks as statistics; an acceptable Six Sigma project can be started with only rudimentary statistical tools. The Six Sigma idea is very similar to SPC (Statistical Process Control) analysis (Florac & Carleton, 1999), which identifies both the location of problems (producing defectives) and whether or not you can cost-effectively fix them.
Six Sigma provides two main frameworks, which can be better applied according to the scenario that we have. Such scenarios are: (1) there is no process at all (note that a bad process is as good as no process); and (2) there is already existing process(es) that is working reasonably well. In the first scenario, whose focus is on process design, Six Sigma suggests the use of the DMADV framework. DMADV is summarized by the following ideas:

- **Define** the project goals and customer (internal and external) deliverables;
- **Measure** and determine customer needs and specifications;
- **Analyze** the process options to meet the customer needs;
- **Design** (detailed) the process to meet the customer needs;
- **Verify** the design performance and ability to meet customer needs.

In the second scenario, whose focus is on significant process improvements, Six Sigma suggests the use of the DMAIC framework. DMAIC stands for:

- **Define** process goals in terms of key critical parameters (i.e. critical to quality or critical to production) on the basis of customer requirements;
- **Measure** the current process performance in context of goals;
- **Analyze** the current scenario in terms of causes of variations and defects;
- **Improve** the process by systematically reducing variation and eliminating defects;
- **Control** future performance of the process.

Sometimes a DMAIC application may turn into a DMADV application because the process in question requires complete re-design to bring about the desired degree of improvement. Such a discovery usually occurs during the improvement phase of DMAIC. In our case we have decided for DMAIC because we already have a process and our intention is the improvement of this process and detection of its problems.

6.2 DMAIC Phases Application

As discussed before, Six Sigma specifies more than one problem-solving framework, as processes and situations can vary on their nature. We have decided for DMAIC because it is more appropriate for existing processes. In this way, this section summarizes the role of each DMAIC phase and details how we have specified each of these phases to work during the measure and analysis of our test process.

Define Phase

The main role of the Define phase is to formally specify the DMAIC project, its elements, context, importance and purpose. For our test process, in particular, the most relevant output from this phase is the definition of what is the issue that we intend to improve. This issue is the prediction of total test time or test effort for a handset.
To better understand this problem, consider a partial list of handset features (MMS, Bluetooth, EDGE, etc.). Each handset that is going to be evaluated supports a subset of such

features, which are used to compose its test suite. For example, if a handset does not support Streaming, all the tests related to this features are removed from the evaluation set. Thus, we can conclude that the total test time is not the same for all handset models.
When the development unit sends a handset to our test team, they need to know the total test time so that they can plan their next actions. To deal with unpredictable problems, we can add an error limit to our estimations. For example, if we estimate that a battery is performed in 120 minutes, we can add 20% of error and say that it performs in 144 minutes. This approach can generate delays in the development unit process and, consequently, in the process as a whole. To exemplify the problem of simple estimations, i.e. estimations without a real statistical investigation, observe the graph below (Fig. 10) where the Y-axis represents time and X-axis represent the test batteries. This graph represents the results of our first evaluation running and it brings information about the estimated time for each test battery and the real time of its execution[2].

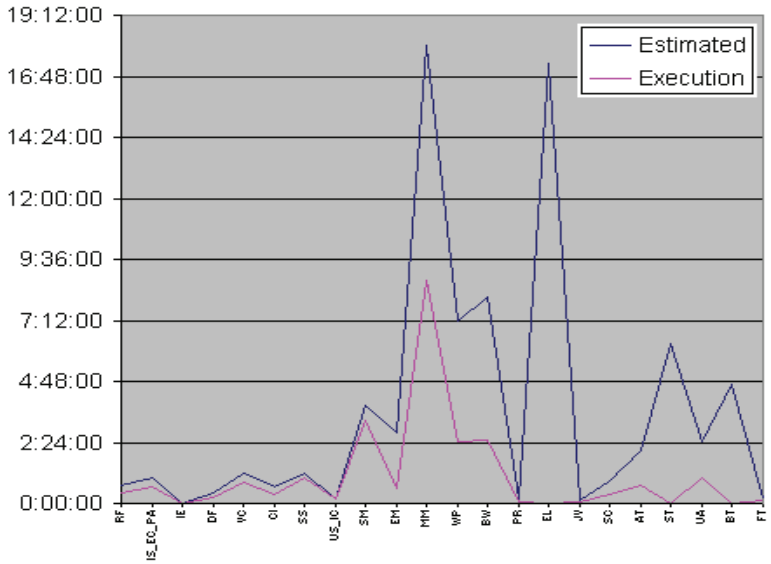


Fig 10. Relation between estimated and execution time.

All our estimates were too high and in some cases (e.g., MM - *Multimedia Messaging Service* - and WP – *Wireless Application Protocol*) the estimates were very deficient. All these estimation problems raise collateral effects to the development unit, which could have defined a better operational plan if they had more effective test time estimates.

Measure Phase

The most important output of the *Measure* phase is the *Baseline*, a historical measurement of indicators chosen to determine the performance before changes made by the DMAIC

[2] Zero Execution time means that the related battery was not applied to the handset in test (e.g., ST – Streaming).

application. These indicators are also measured to assess the progress during the *Improve* phase and to ensure that such an improvement is kept after the *Control* phase. The lead indicator, in our case the prediction of the total test time, is used to track variables that affect its value. Our investigation in this phase is separately performed on each test battery, so that we can perform a more granular investigation on the results. This approach is justified because the batteries have very particular features, which can be better identified and understood if they are analysed in this way. For simplifications, let us consider that there exists only one battery and the total test time is the time to perform this battery. Considering this premise, The *Shewhart Graph* (Florac & Carleton, 1999) (Fig. 12) shows the results related to initial test executions, which were manually carried out in the simulator environment.

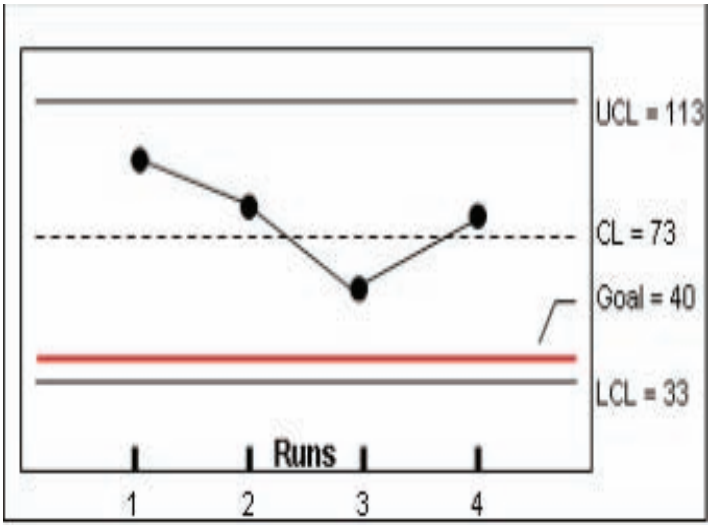


Fig. 12. Shewhart Graph for total time of test runs.

The Shewhart graph contains a Central Line (CL), a Lower Control Line (LCL), an Upper Control Line (UCL) and values related to the issue that we intend to control or improve. In our case, these values are related to the total test time of initial test runs. These data is sequentially plotted along the time, so that we have a historical registry of this information. This is a continuous process and the more data we have, the better will be our Baseline. The CL represents a central value or average of the measures performed on the issue. Both control limits, which are estimates of the process bounds based on measures of the issue values, indicate the limits to separate and identify exceptional points (Humphrey, 1988). The control limits are placed at a distance of 3-sigma (or 3σ) from the central line (sigma σ is the standard deviation).

This graph is an appropriate resource to clarify our objectives in using DMAIC. These objectives and their relation with the graph are:

- More accurate prediction of total test time: this is indicated by the distance between LCL and UCL. The shorter this distance is, the more accurate our prediction will be;

- Test process improvement: the central line (CL) represents our current prevision for the performance of a test run, considering the complete set of test batteries. Test process improvements mean to reduce such a prevision. Our goal is indicated by the red bound line in the graph (goal line);
- Better control: if all total test times are between the control limits, then the test presents only common causes of variation and we can say that the test is in a statistically controlled state (stable test). Differently, if a total test time is out of the control limits, the test presents special causes of variation and we can say that the test is out of statistical control (unstable test).

Before continuing to the next phase, it is important to understand the meaning of common and special causes of variation. Common causes of variation are problems inherent in the system itself. They are always present and affect the output of the process. Examples are poor training and inappropriate production methods. Special causes of variation are problems that arise in a periodic fashion and they are somewhat unpredictable. Examples of special causes are operator error and broken tools. This type of variation is not critical and only represents a small fraction of the variation found in a process (Deming, 1975).

Analysis Phase

The Analysis phase accounts for raising and validating main causes of problems during the handsets network test process. Table 2 summarizes examples of such problems for our domain.

#	**Cause**	**(I)**	**(C)**	**(P)**
1	New testers/operators	L	M	L
2	Operational errors	H	L	H
3	Later found incompatibilities	M	L	M
4	Infra-structure support	M	H	M
5	Bugs in network simulator	H	H	L
6	Test version no longer compatible	M	L	H

Table 2. Summary of problems for the handset test domain

Three parameters are associated with each cause of problems: influence on lead indicator (I), theoretical cost to fix this cause (C), and priority to apply some solution (P). These parameters can assume three qualitative values: low (L), medium (M) and high (H). Such values were set based on our first test experiments and discussions with the technical team. For example, for the first cause we have concluded that the insertion of new testers (externals and trainers) has a low impact on the test process, despite the fact that they only have an initial experience in this process. The cost to carry out a special training is medium, once that we need to allocate a tester engineer to this task. Thus this task is not a priority.

Improve Phase

The Improve phase accounts for selecting and implementing solutions to reduce or eliminate the causes of problems discovered during the Analysis phase. During the Analysis phase, we have already started the discussion about potential solutions for these causes. The following actions are examples of solution that could be implemented in our process: specification of a more granular and formal test process, which focuses mainly on avoiding loss of data (e.g., extensive use of backups) and finding the points where we can carry out tasks in parallel to eliminate dependences and producer-consumer like errors. Solutions are not fixed and they must be adapted to new causes that may appear. This is also one of the reasons to monitor the process even after the application of solutions.

Control Phase

The Control Phase accounts for maintaining the improvement after each new cycle. In our case, this cycle represents the execution of a complete test run. This phase is also related to the process of monitoring the execution of each test, so that new problems can be detected. According to DMAIC, a new problem is raised when the lead indicator, in our case the total test time, is out of the control limits. The graph below (Fig. 13) shows an example of new limit controls that could appear after the application of some solutions. We can observe that the central line is not reached. However we can see some improvement in terms of a new central line (shorter total test time) and narrower control limits. From now on, all the measures must respect such limits.

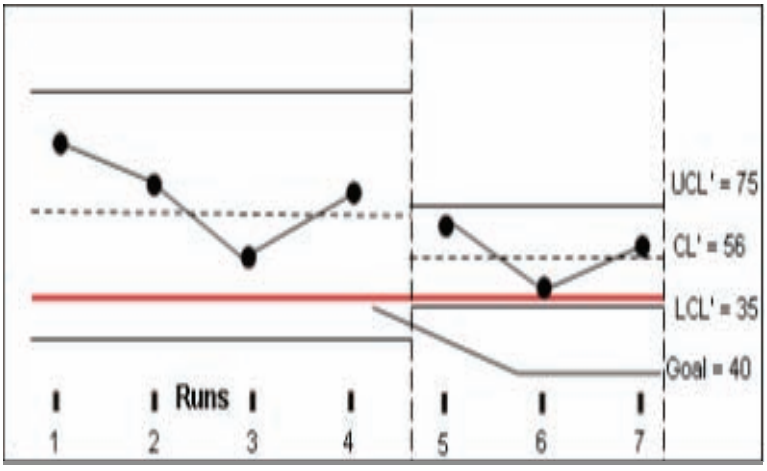


Fig. 13. New values after the application of solutions

Our test suit is not complete. Currently we are executing 60% of the total number of tests, already specified by our telecom test team. Furthermore, the evolution of new technologies, such as 3G, will certainly bring the need for new test cases. All these facts also contribute to a continuous and dynamic change of the indicator values, which can be monitored via the use of statistic methods such as the ones used in this work.

6. Test Management Tools

A last point to be discussed in this chapter is the use of test management tools as an alternative to complete some functions required by an automation test architecture. Current tools[3] can provide one or more from the following features:

- Specification for plans of tests - considers definition of test cases to be applied, test priority, test strategies, test schedule and resources to be allocated;
- Test evaluation support - considers specification and customization of test evaluation documents and summary of tests (test results, test cases coverage, process test environment);
- Extraction of statistical indicators - considers the capture of statistical indicators, specification and customization of statistical reports;
- Test cases creation - considers the edition and maintenance of test cases;
- Integration support - considers the existence of an API to enable the integration of a tool platform with external applications;

We can observe that several of these features are already considered by our proprietary solution. For example, the specification for test plans is covered by the test suit planner (Section 4). The main advantage in using external test management tools is the quality provided by several existing commercial systems. However, we must consider if such systems are flexible enough so that we can make adaptations and customizations in some of their functions.

7. Conclusion

The purpose of this chapter was to discuss several aspects related to the process of test automation. With this objective, we have introduced our domain and initial simulation test environment. Using this environment we have discussed several solutions and concepts that are currently under investigation by our research team. Furthermore, we have also discussed a statistical technique for controlling the process during its continuous evolution.

8. Acknowledgements

The authors would like to thank all the test engineers (Amanda Araujo, Karine Santos, Rivaldo Oliveira and Paulo Costa), software engineers (Angela Freitas and Kleber Carneiro), telecom internals (Anniele Costa and Ronaldo Bitu) and product manager (Fernando Buononato) of the CIn/SIDI-Samsung Test Center, which provided the technical details about the handsets' network test domain, simulation environment and GSM principles. The team is also very grateful for the support received from Samsung/SIDI team, in particular from Ariston Carvalho, Miguel Lizarraga, Ildeu Fantini and Vera Bier. The National Council for Scientific and Technological Development (CNPq) has provided valuable support to the project through the Brazilian Federal Law no. 8010/90.

[3] TestLink (http://testlink.org), QATraq (http://www.testmanagement.com), RHT (http://sourceforge.net/projects/rth/), Salome-TMF (https://wiki.objectweb.org/salome-tmf) and so on.

conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

8. References

Anite (1999). SAS 12447D/UMOOI GSM-9OO/DCSI800 /PCS-1900. *Stand Alone Simulator User Manual*, Rel. 3.0, Anite Telecoms Ltd, Fleet, Hampshire, UK.

Biehl, R. (2004). Six Sigma for Software, *IEEE Software*, 21, 2, 68-70, USA.

Deming, W. (1975). On probability as a basis for action, *The American Statistician*, 29, 4, 146-152.

De Vriendt, J. ; Laine, P. ; Lerouge, C. & Xiaofeng, X. (2002). Mobile network evolution: a revolution on the move, *IEEE Communications Magazine*, 40(4):104-111.

Fikes, R. & Nilsson, N. (1971). STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, *Proceedings of Second International Joint Conference in Artificial Intelligence*, London, UK.

Filho, C & Ramalho, G. (2000). JEOPS – The Java Embedded Object Production System, *Springer Verlag's Lecture Notes in Artificial Intelligence*, 1952, 52-61, Heidelberg Germany.

Florac, W. & Carleton, A. (1999). Measuring the software process: statistical process control for software process improvement. *The SEI Series in Software Engineering*, Addison-Wesley.

Ganek, A. & Corbi, C. (2003). The Dawning of the autonomic computing era, *IBM Systems Journal*, 42, 1, 5-18.

Garg, V. (2001). *Wireless Network Evolution 2G to 3G*. Prentice Hall, 0-13028-077-1, USA.

Ghallab, M.; Nau, D. & Traverso, P. (2004). *Automated Planning: theory and practice*, Morgan Kaufmann Publishers, 1-55860-856-7, USA.

Gruber, R. (1995). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal Human-Computer Studies*, 43, 5/6, 907-928.

Guarino, N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation, *International Journal of Human-Computer Studies*, 43, 5/6, 625–640.

Hariri, S. et al. (2006). The Autonomic Computing Paradigm, *Cluster Computing: The Journal of Networks, Software Tools, and Applications*, 9, 1, 5-17, Kluwer Academic Publishers.

Harry, M. (1998). Six Sigma: A Breakthrough Strategy for Profitability, *Quality Progress Publications*, 31, 5, 60-64.

Humphrey, W. (1988). Characterizing the Software Process: A Maturity Framework. *IEEE Software*, 5, 2, 73-79.

Kephart, J. & Chess, D. (2003). The Vision of Autonomic Computing, *IEEE Computer*, 36, 1, 41-50.

Langley, P. & Allen, J. (1993). A unified framework for planning and learning. In S. Minton (Ed.), *Machine learning methods for planning*. San Mateo, CA: Morgan Kaufmann.

Nguyen, X. & Kambhampati, S. (2001). Reviving partial order planning, *Proceedings of Seventeenth International Joint Conference in Artificial Intelligence*, 459-466, Seattle, WA, USA.

Russel, S. & Norvig, P. (2002). Artificial Intelligence: A Modern Approach, 2nd Edition, Prentice Hall, 0-13790-395-2, USA.

Rahnema, M. (1993). Overview of the GSM system and protocol architecture, *IEEE Communications Magazine*, 42, 4, 493-502.

Simon, K. (2007). DMAIC versus DMADV, *Six Sigma WebPage*. Available in: http://www.isixsigma.com/library/con tent/c001211a.asp

VanBrunt, R. (2003). WCDMA versus GSM: handset performance testing, *RF Design*, 26, 9, 14-23, Cardiff Publishing Company Inc, USA.

Weld, D. (1994). An introduction to least-commitment planning, *AI Magazine*, 15, 4, 27-61.

Wilkins, D. (1984). Domain-independent planning: representation and plan generation, *Artificial Intelligence*, 11, 3, 269- 301.

3

Automatic Speaker Recognition by Speech Signal

Milan Sigmund
Brno University of Technology
Czech Republic

1. Introduction

Acoustical communication is one of the fundamental prerequisites for the existence of human society. Textual language has become extremely important in modern life, but speech has dimensions of richness that text cannot approximate. From speech alone, fairly accurate guesses can be made as to whether the speaker is male or female, adult or child. In addition, experts can extract from speech information regarding e.g. the speaker's state of mind. As computer power increased and knowledge about speech signals improved, research of speech processing became aimed at automated systems for many purposes.

Speaker recognition is the complement of speech recognition. Both techniques use similar methods of speech signal processing. In automatic speech recognition, the speech processing approach tries to extract linguistic information from the speech signal to the exclusion of personal information. Conversely, speaker recognition is focused on the characteristics unique to the individual, disregarding the current word spoken. The uniqueness of an individual's voice is a consequence of both the physical features of the person vocal tract and the person mental ability to control the muscles in the vocal tract. An ideal speaker recognition system would use only physical features to characterize speakers, since these features cannot be easily changed. However, it is obvious that the physical features as vocal tract dimensions of an unknown speaker cannot be simply measured. Thus, numerical values for physical features or parameters would have to be derived from digital signal processing parameters extracted from the speech signal. Suppose that vocal tracts could be effectively represented by 10 independent physical features, with each feature taking on one of 10 discrete values. In this case, 10^{10} individuals in the population (i.e., 10 billion) could be distinguished whereas today's world population amounts to approximately 7 billion individuals.

People can reliably identify familiar voices. About 2-3 seconds of speech is sufficient to identify a voice, although performance decreases for unfamiliar voices. One review of human speaker recognition (Lancker et al., 1985) notes that many studies of 8-10 speakers (work colleagues) yield in excess of 97% accuracy if a sentence or more of the test speech is heard. Performance falls to about 54% when duration is shorter than 1 second and/or distorted e.g., severely highpass or lowpass filtered. Performance also falls significantly if training and test utterances are processed through different transmission systems. A study

using voices of 45 famous people in 2 seconds test utterances found only 27% recognition in an open-choice test, but 70% recognition if listeners could select from six choices (Lancker et al., 1985). If the utterances were increased to 4 seconds, but played backward (which distorts timing and articulatory cues), the accuracy resulted to 57%. Widely varying performance on this backward task suggested that cues to voice recognition vary from voice to voice and that voice patterns may consist of a set of acoustic cues from which listeners select a subset to use in identifying individual voices. Recognition often falls sharply when speakers attempt to disguise their voices e.g., 59-81% accuracy depending on the disguise vs. 92% for normal voices (Reich & Duke, 1979). This is reflected in machines, where accuracy decreases when mimics act as impostors. Humans appear to handle mimics better than machines do, easily perceiving when a voice is being mimicked. If the target (intended) voice is familiar to the listener, he often associates the mimic voice with it. Certain voices are more easily mimicked than others, which lends further evidence to the theory that different acoustic cues are used to distinguish different voices.

From the performance point of view, automatic speaker recognition by speech signal can be seen as an application of artificial intelligence, in which machine performance can exceed human performance e.g., using short test utterances and a large number of speakers. This is especially true for unfamiliar speakers, where the training time for humans to learn a new voice well is very long compared with that for machines. Constraints on how many unfamiliar voices a person can retain in short-term memory usually limit studies of speaker recognition by humans to about 10 speakers.

2. Verification and Identification of Speakers

Speaker recognition covers two main areas: speaker verification and speaker identification. Speaker verification is concerned with the classification into two classes, genuine person and impostor. In verification, an identity claim is made by an unknown speaker, and an utterance of the unknown speaker is compared with the model for the speaker whose identity is claimed. If the match is above a certain threshold, the identity claim is verified. Figure 1 shows the basic structure of a speaker verification system. A high threshold makes it difficult for impostors to be accepted by the system, but at the risk of rejecting the genuine person. Conversely, a low threshold ensures that the genuine person is accepted consistently, but at the risk of accepting impostors. In order to set a threshold at a desired level of user acceptance and impostor rejection, it is necessary to know the distribution of customer and impostor scores.

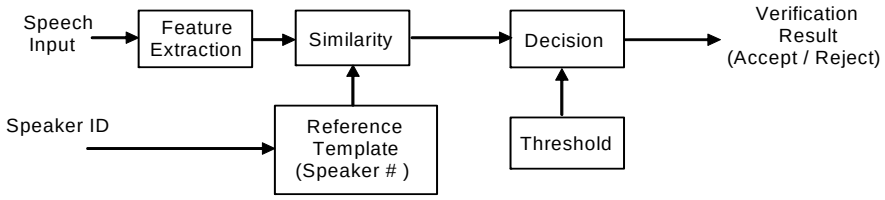


Fig. 1. Basic structure of speaker verification system.

There are two corresponding types of errors, namely the rejection of genuine speakers, often called false rejection, and the acceptance of impostors, often called false acceptance. The most common performance measure used for comparing speaker verification systems is the equal error rate. The equal error rate is found by adjusting the threshold value until the false acceptance rate is equal to the false rejection rate. In most cases, this value must be determined experimentally by collecting the recognition scores for a large number of both accepting and rejecting comparisons. This involves applying an a-posteriory threshold. An illustration of an error rate graph is shown in Figure 2. The use of an equal error rate implies a perfect choice of threshold, which is not possible in a real application since the threshold would have to be determined a-priory. This problem can be solved using probability theory. The threshold for speaker verification must be updated with long-term voice variability (Matsui et al., 1996).

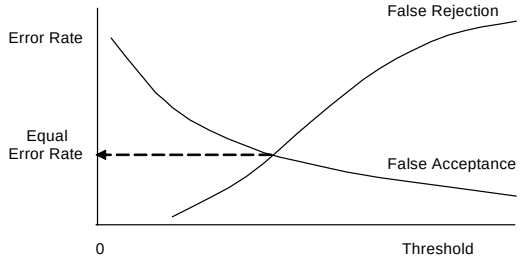


Fig. 2. False rejection rate and false acceptance rate as a function of the decision threshold.

In speaker identification, a speech utterance from an unknown speaker is analysed and compared with models of known speakers. The unknown speaker is identified as the speaker whose model best matches the input utterance. Figure 3 shows the basic structure of a speaker identification system.

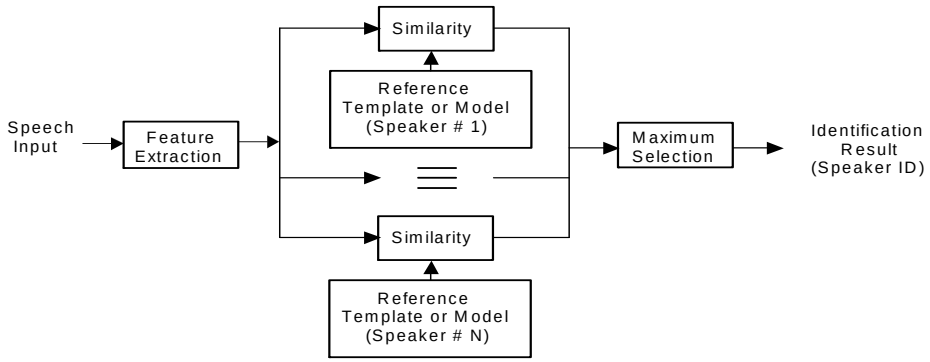


Fig. 3. Basic structure of speaker identification system.

There is also the case called "open set" identification, in which a model for the unknown speaker may not exist. In this case, an additional decision alternative, "the speaker does not

match any of the models", is required. The fundamental difference between identification and verification is the number of decision alternatives. In identification, the number of decision alternatives is equal to the size of the population, whereas in verification there are only two decision alternatives (accept or reject).

3. Text-Dependent Speaker Recognition

Speaker recognition methods can also be divided into text-dependent and text-independent methods. The former require the speaker to provide utterances of the key words or sentences having the same text for both training and recognition trials, whereas the latter do not rely on a specific text being spoken. The text-dependent methods are usually based on template matching techniques in which the time axes of an input speech sample and each reference template or reference model of registered speakers are aligned, and the similarity between them accumulated from the beginning to the end of the utterance is calculated. The structure of text-dependent recognition systems is, therefore, rather simple. Since this method can directly exploit the voice individuality associated with each phoneme or syllable, it generally achieves higher recognition performance than the text-independent method.

3.1 Effectiveness of Various Phonemes for Speaker Recognition

The speaker-specific information contained in short-term spectra was used in the initial experiments. Twelve male speakers read the same text twice. The signal was sampled at 22 kHz with 16-bit linear coding. The speech signals were labeled using own tool (Sigmund & Jelinek, 2005), and a log-power spectrum (128 point FFT) was calculated in the centre of continuant sounds. The spectral channel containing maximum intensity was then set at 0 dB. The reference samples were created by averaging three spectra. Finally, the spectra were compared by a distance measure derived from a correlation based similarity measure

$$d = 1 - \frac{< \mathbf{x}, \mathbf{y} >}{\|\mathbf{x}\| \, \|\mathbf{y}\|} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent the spectral vectors. Each phoneme in the test was compared with each of the corresponding reference phonemes. The reference sample with the minimal distance was considered to be identified. The identification rate varies from 11% to 72%. The results obtained indicate that an individual analysis of each phoneme is impossible but that the data can be reasonably grouped into phonetically defined classes. Table 1 gives average identification rates. Thus, in terms of speaker-recognition power, the following ranking of phoneme classes results:

vowels, nasals > liquids > fricatives, plosives.

As expected, vowels and nasals are the best phonemes for speaker identification. They are relatively easy to identify in speech signal and their spectra contain features that reliably distinguish speakers. Nasals are of particular interest because the nasal cavities of different speakers are distinctive and are not easily modified (except when nasal congestion). For our purposes, Table 1 gives a preliminary general overview of the results.

Phoneme Class	Identification Rate (in %)
Vowels (a, e, i, o, u)	68
Nasals (m, n)	67
Liquids (l, r)	53
Fricatives (f, s, sh, z)	46
Plosives (p, t, b, d, g)	32

Table 1. Speaker identification rate by phoneme classes.

Two experiments were then performed on the data set within the vowel class. Because of the formant (i.e. local maxima) structure of vowel spectra the identification rate for each vowel can be estimated. In Table 2, individual vowels are compared in terms of speaker-recognition power.

Vowel	No. of Vowels in Test	Identification Rate (in %)
i	117	52.7
o	106	61.4
u	85	68.2
a	121	74.8
e	122	76.2

Table 2. Speaker identification rate by individual vowels.

It is known that different speakers show not only different formant values but exhibit different arrangements in their vowel systems. The general distribution patterns of vowels in formant planes can be used to build up a feature matrix for the vowel system of individual speakers. Table 3 shows the identification rate for various numbers of different vowels. The test started with only one vowel (the most effective) and successively other vowels were added one by one according to their individual effectiveness as ranked in Table 2. The identification rate increased almost logarithmically from 76.2% using the one individually best vowel "e" up to 97.4% using all the five vowels simultaneously.

No. of Vowels	Vowels	Identification Rate (in %)
1	e	76.2
2	e, a	88.7
3	e, a, u	93.8
4	e, a, u, o	95.6
5	e, a, u, o, i	97.4

Table 3. Speaker identification rate depending on number of vowels used.

3.2 Effectiveness of Speech Features in Speaker Recognition

In order to see which features are effective for speaker recognition, we studied here the following six parametric representations: 1) autocorrelation coefficients; 2) linear prediction (LP) coefficients; 3) log area ratios; 4) cepstral coefficients; 5) mel-cepstral coefficients; 6) line spectral-pair (LSP) frequencies. More details how to compute these parameters could be found in (Rabiner & Juang, 1993). Although all of these representations provide equivalent information about the LP power spectrum, it is only the LSP representation that has the localized spectral sensitivity property. As can be seen in Section 3.1, the vowel phonemes result the best in recognition performance regarding the speaker identification rate. Thus, the vowels as speech data used for this purpose were derived from utterances spoken by nine male speakers. These utterances were low-pass filtered at 4 kHz and sampled at 10 kHz. The steady-state part of the vowel segment was located manually.

For each speaker and for each feature set the first ten coefficients were used. The Euclidean distance was obtained by comparing a test vector against a template. A match was detected based on the minimum distance criterion, if the intra-speaker distance was shorter than all the inter-speaker distances. Otherwise a mismatch was declared. These matches and mismatches were registered in the confusion matrices for each parametric representation. Table 4 shows recognition rates for all six parametric representations mentioned above. From these results, it can be seen that for text-dependent speaker recognition the autocorrelation coefficients are not very effective, the log area ratio coefficients set generally surpasses any other feature sets, and mel-cepstral coefficients are comparable with LSP frequencies in recognition performance. In order to compute the text-dependent speaker recognition performance for each feature set, the following procedure was used. For each vowel, five repeats were used as the training set and about thirty randomly chosen vowels were used as the test set; all this for a given speaker. The training set and the test set were disjunct.

Parameters	Test Patterns	Recognition Rate (in %)
Autocor. coeffs.	270	61.3
LP coeffs.	268	83.7
Log area ratios	254	94.1
Cepstral coeffs.	262	87.5
Mel-Cepstal coeffs.	249	91.2
LSP frequencies	241	90.8

Table 4. Performance of vowel-dependent speaker recognizer using various parametric representations.

4. Text-Independent Speaker Recognition

There are several applications in which predetermined key words cannot be used. In addition, human beings can recognize speakers irrespective of the content of the utterance. Therefore, text-independent methods have recently been actively investigated. Another advantage of text-independent recognition is that it can be done sequentially, until a desired

significance level is reached, without the annoyance of repeating the key words again and again. In text-independent speaker recognition, the words or sentences used in recognition trials cannot generally be predicted. For this recognition, it is important to remove silence/noise frames from both the training and testing signal to avoid modeling and detecting the environment rather than the speaker.

4.1 Long-Term Based Methods

As text-independent features, long-term sample statistics of various spectral features, such as the mean and variance of spectral features over a series of utterances, are used (see Fig. 4). However, long-term spectral averages are extreme condensations of the spectral characteristics of a speaker's utterances and, as such, lack the discriminating power included in the sequences of short-term spectral features used as models in text-dependent methods. The accuracy of the long-term averaging methods is highly dependent on the duration of the training and test utterances, which must be sufficiently long and varied.

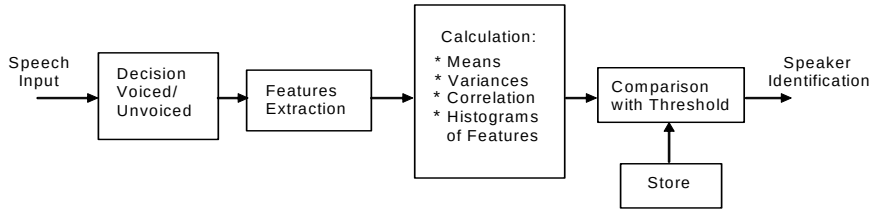


Fig. 4. Typical structure of the long-term averaging system.

4.2 Average Vocal Tract Spectrum

In a long-time average spectrum of a speech signal the linguistic information (coded as frequency variation with time) is lost while the speaker specific information is retained. In this study, a speaker analysis approach based on linear predictive coding (LPC) is presented. The basic idea of the approach is to evaluate an average long-time spectrum corresponding to the anatomy of the speaker's vocal tract independent of the actually pronounced phoneme. First, we compute the short-time autocorrelation coefficients R_j (k), k=1,...,K for the j-th frame (20 msec) of speech signal s (n)

$$R_j(k) = \sum_{n=1}^{N-k} s(n)\, s(n+k) \tag{2}$$

where N is the number of samples in each frame, and then we compute the K average autocorrelation coefficients

$$\overline{R}(k) = \frac{1}{J} \sum_{j=1}^{J} R_j(k) \tag{3}$$

corresponding to the whole utterance formed by J frames. Thus, from the average autocorrelation coefficients, we get the average predictor coefficients $\overline{a}_m$ e. g., via the Durbin algorithm (Rabiner & Juang, 1993) and finally the normalised average LPC-based spectrum using

$$S(f) = \left| \frac{1}{1 - \sum_m a_m z^{-m}} \right|^2 \qquad \text{for } m = 1,...,M \tag{4}$$

where $z = \exp(j\, 2\pi f / f_s)$, f_s is the sampling frequency and M is order of the LPC model equal to the highest autocorrelation order K. More details how to compute the LPC coefficients and corresponding spectra on short frame of speech signal can be found in (Rabiner & Juang, 1993). The speech signal was sampled at 22 kHz using a 16-bit A/D converter under laboratory conditions over a period of five months. A group of 26 speakers (19 male, 7 female) aged 20 to 25 years took part in the tests.

A comparison between intra- and inter-speaker variability in long-time spectrum is shown in Figures 5 and 6. Figure 5 illustrates two vocal tract spectra of the same speaker corresponding to two different texts. The difference between both curves is 12%.

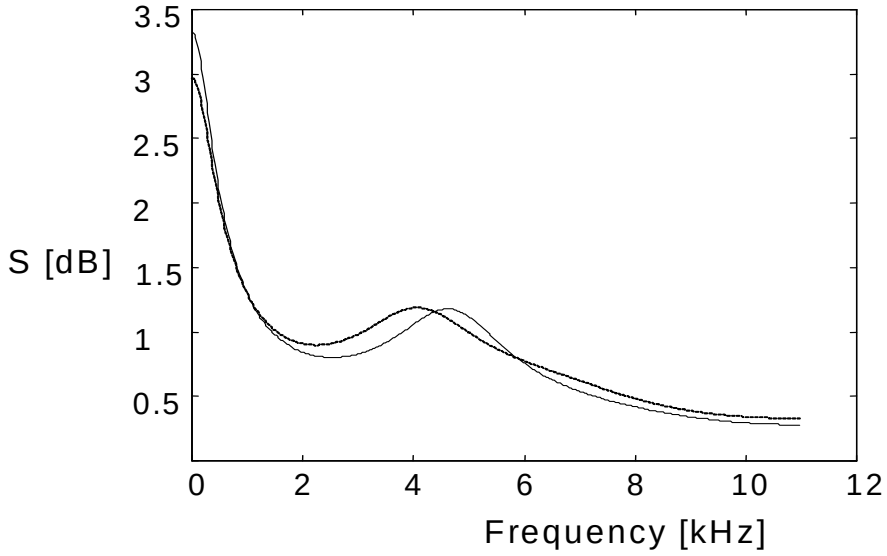


Fig. 5. Long-time spectrum difference of one and the same speaker (LPC order 6, speech duration 100 sec).

Vocal tract spectra obtained from two different speakers saying the same text is shown in Figure 6. The difference between both curves increased to 22% in this case. The average

intra-speaker difference over all speakers was 12.6%, while the average inter-speaker difference (gender-specific) reached 23.4%. In accordance with the inter-gender differences, the estimated difference between the two groups of speakers (male and female) was more apparent (29.6%) than within the groups (Sigmund & Mensik, 1998).

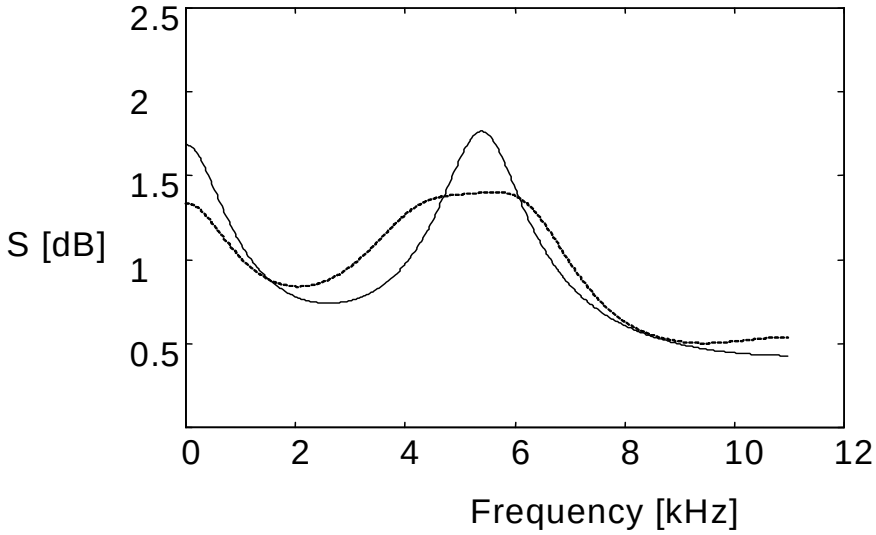


Fig. 6. Long-time spectrum variability between speakers (speech duration 100 sec).

5. Automatic Gender Recognition

Some studies show that speaker identification as well as speech recognition would be simpler, if we could automatically recognize a speaker's gender (sex). For example, in the "cocktail party effect", the voices of two or more speakers may be mixed. If the speakers are of opposite sex and if sex identification can be made on short segments of speech, the voices can be at least partially separated. Sex identification was used primarily as a means to improve recognition performance and to reduce the needed computation. Accurate sex identification has different uses in spoken language systems, where it can permit the synthesis module of a system to respond appropriately to an unknown speaker. In languages like French, where formalities are often used, the system acceptance may be easier if greetings such as "Bonjour Madame" are foreseen. In the past, automatic gender identification has been investigated for clean speech by Wu and Childers (Wu & Childers, 1991). Clean speech and speech affected by adverse conditions are evaluated for a variety of gender identification schemes in (Slomka & Sridharan, 1997). Using speech segments with an average duration of 890 msec (after silence removal), the best mentioned accuracy is 98.5% averaged over all clean and adverse conditions. There is some evidence that sex-related speech characteristics are only partly due to physiological and anatomical differences between the sexes; cultural factors and sex-role stereotypes also play an important part.

The main feature which can speaker's sex distinguish is fundamental frequency F_0 with typical values of 110 Hz for male speech and 200 Hz for female speech. The pitch of children is so different that they are often treated as "the third sex". Most values of F_0 among people aged 20 to 70 years lie between 80-170 Hz for men, 150-260 Hz for women while 300-500 Hz for children (Baken & Orlikoff, 2000). There are Gaussian distributions of these ranges, so that dispersion is wide and we often could not categorize the acoustic signal reliably by using this criterion only. Figure 7 (Titze, 1989) illustrates the inverse relationship between fundamental frequency of speech F_0 and length of glottal membrane L_m.

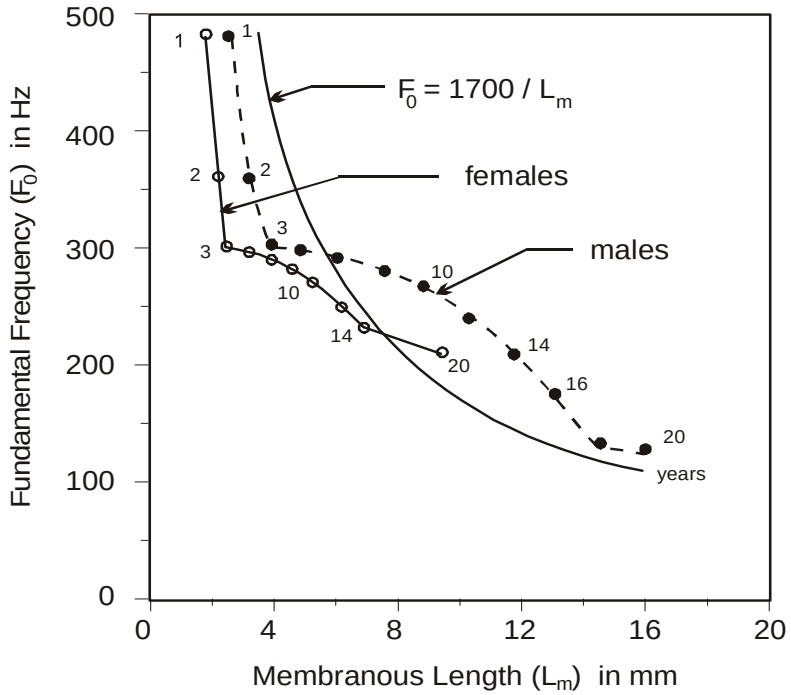


Fig. 7. Mean speaking fundamental frequency F_0 as a function of membranous length L_m.

5.1 Cepstral Analysis

The most commonly used short-term features in speech signal processing are cepstral coefficients and their frequency-warped alternative coefficients. Thus, the mel-frequency warped cepstral coefficients were taken for our experiment to identify the sex of a speaker. First, a Hamming window was applied for each speech frame (20 msec) of the recorded vowels and the FFT spectrum was computed. Then, the spectrum was mel-warped and the inverse Fourier transform of the logarithm of the warped spectrum produced the vector of cepstral coefficients. The mel-frequency scale is linear below 1 kHz and logarithmic above 1 kHz (Rabiner & Juang, 1993). Using a set of 41 mel-cepstral coefficients c_0 through c_{40} and their various differences, the performance of these individual features as identifiers of the sex of a speaker was measured. Table 5 summarizes the selected suitable coefficients which had the lowest variation calculated individually for Czech vowel phonemes and then

averaged for both genders separately. The best individual feature seems to be the coefficient c_{24} followed by c_{26} and c_{25}, respectively. On the other hand, the differences of cepstral coefficient pairs are not reliable for sex identification (Kepesi & Sigmund, 1998).

		c_0	c_1	c_2	c_6	c_9	c_{17}	c_{18}
Male	μ	-828,0	-326,0	338,0	28,0	94,0	21,0	31,0
	σ	181,0	122,0	141,0	73,0	52,0	65,0	39,0
Female	μ	-1150,0	-597,0	164,0	182,0	-20,0	129,0	112,0
	σ	125,0	88,0	137,0	70,0	53,0	45,0	47,0
		c_{22}	c_{23}	c_{24}	c_{25}	c_{26}	c_{35}	c_{36}
Male	μ	20,0	-35,0	-1,2	4,6	-42,0	20,9	60,1
	σ	46,0	51,0	39,0	57,0	47,0	41,0	45,0
Female	μ	109,0	37,0	98,0	158,0	101,0	-119,0	-84,0
	σ	57,0	55,0	29,0	28,0	34,0	80,0	91,0

Table 5. Mean μ and standard deviation σ of selected mel-frequency cepstral coefficients c_i.

5.2 Gender Identifiers

Two sex recognition approaches were used in our test. The first approach was based on an individual cepstral coefficient. Applying an empirical formula to the coefficient c_{24} we get the gender identifier D_{24} in the form

$$D_{24} = |c_{24} - 80| - |c_{24} - 40| - 120 + 2c_{24} \tag{5}$$

This indicator gives a negative value for male and a positive value for female speakers. The second approach used a set of selected cepstral coefficients according to the Table 5. For both sex classes the reference mean vectors were formed as follows:

Male reference: **M** = [-326, 338, 28, 94, 21, 31, 20, -35, -1, 5,…]
Female reference: **F** = [-597, 164, 182, -20, 129, 112, 109, 37,…]

and the Euclidean distances d_1 and d_2 were calculated in each test

$$d_1(\mathbf{X},\mathbf{M}) = [(\mathbf{X}-\mathbf{M})^T (\mathbf{X}-\mathbf{M})]^{1/2} \tag{6}$$

$$d_2(\mathbf{X},\mathbf{F}) = [(\mathbf{X}-\mathbf{F})^T (\mathbf{X}-\mathbf{F})]^{1/2} \quad (7)$$

where **M** and **F** denote the reference vectors mentioned above, **X** is the tested vector formed by the same coefficients c_i as the reference vectors, and T denotes transpose. Computing the difference of the two distances

$$D = d_1(\mathbf{X},\mathbf{M}) - d_2(\mathbf{X},\mathbf{F}) \quad (8)$$

we get a measure which gives similar polarity result as the identifier D_{24} (negative for male, positive for female).
Both procedures described above were evaluated for Czech vowel phonemes, which provided an identification accuracy of more than 90%. Especially for vowel "a" almost no error occurs. Table 6 shows the recognition rate obtained for all individual vowels cut out from a normally spoken speech.

Identifier	Test Vowel				
	a	e	i	o	u
D_{24}	99	92	97	93	91
D	99	94	98	92	94

Table 6. Gender recognition rate in percent testing all individual vowels.

The used speech data consisted of 420 sentences in total, 5 sentences by each of the 84 speakers (53 male and 31 female). All speakers in the database were subjective in good physical and psychical condition and have no speech, language or hearing difficulties. Most of the speakers are student aged 20 to 25 years. All speakers are Czech natives speaking with standard accent. The speakers were not informed of the objectives of the study before the experiment. The speech signal was sampled at 22 kHz using a 16-bit A/D converter under usual conditions in an office room.

6. Applications of Automatic Speaker Recognition

Law enforcement and military security authorities were among the first to make use of speaker recognition technology. The first type of machine speaker recognition using spectrograms of their voices, called voiceprint analysis or visible speech, was begun in the 1960s. The term voiceprint was derived from the more familiar term fingerprint. Voiceprint analysis was only a semiautomatic process. First, a graphical representation of each speaker's voice was created. Then, human experts manually determined whether two graphs represented utterances spoken by the same person. The graphical representations

took one of two forms: a speech spectrogram or a contour voiceprint (Baken & Orlikoff, 2000). The more commonly used form consists of a representation of a spoken utterance in which time is displayed on the horizontal axis, frequency on the vertical axis and spectral energy as the darkness at a given point.
At present, the increase in commercial application opportunities has resulted in increased interest in speaker recognition research. The main commercial application for speaker recognition seems to be speaker verification used to the physical entry of a person into a secured area, or the electronic access to a secured computer file or licensed databases. Such voice-based authorization is often a part of a security system that also includes the use of PIN number, password, and other more conventional means. The most immediate challenge in voice-based authorization is a caller authentication over the telephone network that will be accurate enough so that financial transactions could take place under it aegis. Car access is yet another popular area where voice-based security systems are gained ground. Some automobile manufactures are testing a speaker identification system to control door locks and ignition switches. An interesting twist to this application is that the ignition switch can be programmed not to work if the driver is under the influence of drugs or alcohol, since intoxication is detectable in the speech signal.

7. Acknowledgement

This work was supported by the Czech Ministry of Education in the frame of the Research Plan No. MSM 0021630513 "Advanced Electronic Communication Systems and Technologies".

8. References

Baken, R. J. & Orlikoff, R. F. (2000). *Clinical Measurement of Speech and Voice,* Singular Publishing Group, ISBN 1-56593-869-0, San Diego

Kepesi, M. & Sigmund, M. (1998). Automatic recognition of gender by voice, *Proceedings of Radioelektronika'98*, pp. 200-203, ISBN 80-214-0983-5, Brno, April 1998, CERM, Brno

Lancker, D.; Kreiman, J. & Emmorey, K. (1985). Familiar Voice Recognition: Patterns and Parameters - Recognition of Backward Voices. *Journal of Phonetics*, Vol. 13, No. 1, (January 1985), pp. 19-38, ISSN 0095-4470

Matsui, T.; Nishitani, T. & Furui, S. (1996). Robust methods of updating model and a-priori threshold in speaker verification, *Proceedings of IEEE Internat. Conf. on Acoustics, Speech and Signal Processing*, pp. 97-100, ISBN 0-7803-3192-3, Atlanta, May 1996, IEEE Computer Society, Washington, DC

Rabiner, L. R. & Juang, B. H. (1993). *Fundamentals of Speech Recognition*, Englewood Cliffs, ISBN 0-13-015157-2, New Jersey

Reich, A. & Duke, J. (1979). Effects of selected vocal disguises upon speaker identification by listening. *Journal of the Acoustical Society of America*, Vol. 66, No. 4, (April 1979), pp. 1023-1028, ISSN 0162-1459

Sigmund, M. & Jelinek, P. (2005). Searching for phoneme boundaries in speech signal, *Proceedings of Radioelektronika 2005*, pp. 471-473, ISBN 80-214-2904-6, Brno, April 2005, MJ Servis, Brno

Sigmund, M. & Mensik, R. (1998). Estimation of vocal tract long-time spectrum, *Proceedings of Elektronische Sprachsignalverarbeitung*, pp. 69-71, ISSN 0940-6832, Dresden, September 1998, w.e.b. Universitätsverlag, Dresden

Slomka, S. & Sridharan, S. (1997). Automatic gender identification under adverse conditions, *Proceedings of Eurospeech'97*, pp. 2307-2310, ISSN 1018-4074, Rhodes, September 1997, Typoffset, Patras

Titze, I. R. (1989). Physiologic and acoustic differences between male and female voices. *Journal of the Acoustical Society of America*, Vol. 85, No. 4, (April 1989), pp. 1699-1707, ISSN 0162-1459

Wu, K. & Childers, D. G. (1991). Gender recognition from speech. *Journal of the Acoustical Society of America*, Vol. 90, No. 4, (April 1991), pp. 1828-1840, ISSN 0162-1459

4

Verification Based Model Localizes Faults from Procedural Programs

Safeeullah Soomro
Department of Computer Science,
National University of Computer Sciences and Emerging Sciences
ssoomro@ist.tugraz.at, safeeullah.soomro@nu.edu.pk
NU-FAST, Karachi
Pakistan

1. Overview

Detecting and locating faults is one of the most important issues of current research community. Many efforts have been taken to improve software development and to prevent faults. But still software faults pose the most challenging problems to software engineers. Fault localization is a most challengeable task during the debugging process. Fault localization is the next step after detecting faults in programs. Use of fault localization in control engineering where engineers often employ the procedural programming paradigm. Often controls software is safety-critical and thus detection but also localization of bugs is uttermost important.

This chpater makes use of abstract dependences between program variables for locating and localizing faults in procedural programs. The so called verification-based Model (VBM) for debugging is an extension of dependence model from Jackson's Aspect System, which has been used for verification of C programs. The Aspect system analyzes the dependences between variables of a given program and compares with specified dependences. Otherwise, the program fulfills the specification. In case of mismatch the program is said to violate the specification. Unfortunately, the Aspect system does not allow locating the source of mismatch. The VBM extends Jackson's idea towards not only detecting behavior but also localizing malfunctioning real cause. The VBM performs fix-point computation for recursive invocation (in all cases where we obtain a cyclic call graph). We presented algorithm and proof for fix point computation which ensures that no dependences loss during iteration and we always reached fix-point after finite number of iterations. Furthermore we present novel results obtained from our most recent case studies. Notably, whenever our novel model detects a structural fault, it also appears to be capable of localizing the detected misbehavior's real cause.

Key Words
Model Based Reasoning, Software Verification, Software Debugging, Fault Detection and Localization.

2. Introduction

Software verification is an important phase of software development. In the last decade the software verification and the debugging communities have made considerable progress. In this chapter we focus on fault localization which is based on the abstract dependencies that are used by the Aspect system (Jackson 1995) for detecting faults. The verification-based model for debugging is an extension of the dependence model from Jackson's Aspect system (Jackson 1995) which has been used for dependency based verification of C programs. The Aspect system analysis the dependences between variables of a given program and compares them with the specified dependences. In case of a mismatch the program is said to violate the specification. Otherwise, the program fulfils the specification. Unfortunately, the Aspect system does not allow locating the source of a mismatch. In the following we extend Jackson's idea towards not only detecting misbehaviour but also localizing the malfunctioning real cause.

Although program slicing, as a lightweight technique, has seen successful application in fault localization (Agrawal et al., 1993, Fritzson et al., 1999, Lyle and M. Weiser 1987, its discrimination like MBSD (Wotawa 2002). In (Kuper 1989, Wieland 2001) the authors employ the notion of dependences for fault localization. In contrast to latter approach we do not employ detected difference in variable values at a certain line in code but use of differences between specified and computed dependencies and thus also incorporate the structural properties of program and specification. Thus, the models introduced in (Kuper 1989, Wieland 2001) can not deal with assertions or pre- and post conditions in a straightforward way.

In this chapter we focus on localizing faults in procedural programs and dealing with global variables. Procedural programs are generally more computationally efficient that object oriented programs, because there is less overhead to handle abstractions and the data structures more closely resemble the hardware that must manipulate them.

The chapter is organized as follows. In Section 3 we introduce our verification based model by using motivating example. The results and discussion given in Section 4 reveal the verification based model provides a useful means for detecting and localizing common errors for procedural programs and particular in contest of global variables. In Section 5 we present related research. Further more we resent limitation of our model in section 6. Finally we summarize the chapter.

3. Motivating Example

In this section we explain the basic idea of localizing the fault by checking whether the post condition is satisfying or not using the verification based model.

In this chapter we focus on fault localization which is based on abstract dependencies that are used by the Aspect system (Jackson 1995) for detecting faults. Abstract dependencies are relations between variables of a program. We say that a variable x depends on a variable $y$iff

a new value for *y* may causes a new value for *x*. For example, the assignment statement *x = y + 1* implies such a dependency relation. Every time we change the value of *y* the value of *x* is changed after executing the statement. Another example which leads to the same dependency is the following program fragment (Fig. 1):

```
If (y < 10) then
    X = 1;
Else
    X = 0;
```

Fig. 1. Simple conditional block

In this fragment not all changes applied to *y* cause a change on the value of *x*, although *x* definitely depends on *y*. The Aspect system now takes a program, computes the dependencies and compares them with the specified dependencies. If there is a mismatch the system detects a bug and notifies the user. However, the Aspect system does not pinpoint the root-cause of the detected misbehavior to the user.

In the following program fragment (Fig. 2) we explain the basic ideas using the following small program which implements the computation of the circumference and area of a circle. The program contains one fault in line 2 where a multiplication by π is missing.

```
// pre true
1. d = r * 2;
2. c = d; // BUG! a = d * pi;
3. c = r * r * pi;
// post c = r². π ^ a = 2. r. π
```

Fig. 2. Calculation of circumference and area

These dependences solely are given by a statement whenever we assume that the statement is correct (w.r.t. the dependences). If a statement is assumed to be incorrect, the dependences are not known. We express the latter fact by introducing a new type of variable, the so called model variables. Model variables are variables that work as place-holder for program variables. For example, if we assume statement 2 to be incorrect, we introduce a model that says that program variable *a* depends on model variable ε^2 (where ε^2 is unique).

The idea behind our approach is to find assumptions about the correctness and incorrectness of statements which do not contradict a given specification. In our running example, the specification is given in terms of a post-condition. From this post-condition we derive that *c* has to depend on *r* and *pi*. However, when assuming statement 1 and 2 to be correct, we derive that *a* depends on *d* and *d* in turn depends on *r* which leads to *c* depends on *r* but not on *pi*. Hence, the computed dependence contradicts the specified one.

To get rid of this inconsistency, we might assume line 2 to be faulty. Hence, we can compute that *c* depends on model variable ε^2. When now comparing the specification with the

computed dependence we substitute ε^2 by *r* and *pi* and we can not derive an inconsistency anymore.

The authors of (Wotawa and Soomro 2005, Peischl and Soomro 2006, Soomro 2007) present a detailed formalization of this idea and also present rules for most important language artifacts like an assignment statement, the if-then-else statement, while loop and procedures. In this chapter we focus on method invocation, parameter substitution, return statement and global variables.

```
0.   // Pre conditions of class: true
1.    public class sumpowers {
2.      int i, start, sum;
3.      int stop, f;
4.      …
5.      …
6.      i = start;
7.      while (i < stop)
8.      {
9.        sum = sum + power(x,f);
9.        // post (sum, x), (sum,f), (sum, power)
10.       i = i + 1;
11.     }
12.
13.     //Pre-condition of method: true
15.     int power(xf,ef) {
1.        int power = 1;
2.        while (ef > 0)
2.        {
3.            power = power * 10;
3.            //instead of power = power * xf
4.            ef = ef - 1;
4.         }
5.         return power;
16         // post {(power,power),(power,xf), (power, ef)}
16.    }
17.      //Equation power = xf^ef
17.   }
```

Fig. 3. Sum and power of integer values

In Fig. 3 we show an example that use a method to compute the sum and power of integer numbers.The program contains one method which computes the n^{th} power of an integer number.

In computing the dependences for procedures and their invocations we first compute the dependences of the procedure being invoked. Afterwards we substitute the procedure's formal parameters by the actual ones. We capture recursive invocations by computing the transitive and reflexive closure of the procedure's body and subsequently get rid of

dependences induced by local variables. Finally, we add those dependences caused by the procedure's return values.
In Fig. 3., line number 9 we call the method *power* with some parameters. The specification of method is ***{ (power, power), (power, ef), (power, xf) }***. When computing the dependences from the method we derive these dependences ***{ (power, power), (power, ef), (ef, ef) }.***

In these dependences the variable pair *(ef, ef)* are not impacting overall dependences.
So we have final dependences of method are ***{ (power, power), (power, ef) }***.
We have to map formal to actual parameters from these derived dependences.

Definition 1 (Parameter Substitution)
Let *d* be the dependences of the a method *m* and let *f* be a formal parameter and *a* the corresponding actual parameter. The dependences after method invocation are given by
***d'*= { (a, y) | (f, y) ∈ d } ∪ { (x, a) | (x, f) ∈ d }.**

After mapping parameters we derived dependences ***(sum, power), (sum, f)*** , here we find a contradiction with the post conditions ***{ (sum, x), (sum, f), (sum, power) }***. If post conditions are consistent with computed ones then we introduce model variables. We used unique model variable for every assumption. The Definition 2 states that how to establish the relationship between the return variables and the target variable of calling context.

Definition 2 (Return Values of a Method)

$$\begin{gathered}\neg Ab(x = proc(a_1, a_2, \ldots, a_n)) \rightarrow \\ D(t = proc(a_1, a_2, \ldots, a_n)) = \\ \{t\} \times \{v \mid (x, v) \in D(proc(a_1, a_2, \ldots, a_n)), x \in return(proc)\}\end{gathered} \tag{1}$$

$$\begin{gathered}Ab(t = proc(a_1, a_2, \ldots, a_n)) \rightarrow \\ D(t = proc(a_1, a_2, \ldots, a_n)) = \{(t, \xi_l), (g, \xi'_l) \mid g \in G\}\end{gathered} \tag{2}$$

where t denotes the target variable, g global variable in proc(body) and return(proc) is a function returning the return values of the procedure proc. Where Ab shows abnormal and ¬Ab shows not abnormal conditions.

The definition 3 states how to establish dependences of the two consecutive statements of the program.

Definition 3 (Composition)
Given two dependency relations R_1, $R_2 \in D$ on V and M. The composition of R_1 and R_2 is defined as follows:

$$\begin{aligned}R_1 \circ R_2 = \quad & \{ (x,y) \mid \exists(x,z) \in R_2 \ \& \ \exists(z,y) \in R_1\} \cup \\ & \{ (x,y) \mid \exists(x,y) \in R_1 \ \& \sim\exists(x,z) \in R_2\} \cup \\ & \{ (x,y) \mid \exists(x,y) \in R_2 \ \& \sim\exists(y,z) \in R_1\}\end{aligned} \tag{3}$$

This definition ensures that no information is lost during computing the overall dependency relation for a procedure or method. Hence, the first line of the definition of composition handles the case where there is a transitive dependency. The second line states that all dependencies that are not re-defined in R_2 are still valid. In the third line all dependencies that are defined in R_2 are in the new dependency set provided that there is no transitivity relation.

For combining the dependencies of two consecutive statements we define the following composition operator as given in definition 3 for dependency relations to obtain the following dependences ***D (sum = power(x , f)) = { (power, power), (power, ef) }***. After substituting formal to actual parameter derived dependences of line number 9 are ***{ (sum, power), (sum, f) }*** but the post conditions are ***{ (sum, x), (sum, f), (sum, power) }***. Here we find contradiction between both dependences, derived ones and specified ones.

In order to compare a computed dependence set with the specification we have to find a substitution that makes the computed dependence set equivalent to the specified one. If there is no such substitution the sets are said to be inconsistent.

A substitution σ is a function which maps model variables to a set of program variables, i.e., σ: M → 2^V. The result of the application of the substitution σ on a dependence relation ***R*** is a dependence relation where all model variables ***x*** in ***R*** have been replaced by ***σ (x)***.

We assume that statement 9 is abnormal, we take the target variable from the assignment statement and introduce a model variable (sum, ξ_9) . In order to compute dependences
we derive ***{ (sum, ξ_9), (i, i), (i, stop) }*** and the substitution variables are ***{ (x, f, power) }***. We now compare the specification with the computed dependences obtained by substituting ξ_9 with ***{ (x, f, power) }***. Since, we can not derive an inconsistency anymore, so line number **9** is a bug candidate.
Some assumptions for other lines of method call:
If we assume statement **3** to be incorrect then we can take the left variable from the assignment statement and introduce a model variable to arrive at ***(power, ξ_3)***. After computing dependences with this model variable we derive ***{(power, ξ_3), (ef, ef)}*** and substitution variables are ***{ (power, xf, ef) }***. If we now compare the specification with the computed dependences obtained by substituting ξ_3 by ***{ (power, xf, ef) }***, we can no longer derive inconsistency. So line number **3** from method is a bug candidate.

Assuming line **2** to be incorrect, the dependences derived with model variable are ***{((power, power) , (power, ξ_2), (ef , ef)) }***. Now we substitute the model variable from set of program variables, i.e, ξ_2 ***= { ef, xf, power }***. After substituting we derived ***{ (power, power) ,(power, xf), (power, ef), (ef , ef) }***. In order to allow the direct comparison of specified dependences with the computed ones, we introduce a projection operation which deletes all dependences for variables which have no impact on the overall dependences, like an internal variable pair ***(ef, ef)*** from Fig. 3.

A projection is defined on dependence relations ***R ϵ D*** and a set of variables ***A → M U V***. The projection of ***R*** on ***A*** written as $\prod_A$ (R) is defined as follows in an equation 3:

$$\prod_A (R) = \{ (x,y) \mid (x,y) \in R \wedge x \in A \tag{5}$$

However when assuming statement numbers from method *1,2,3,4* we obtain three diagnoses. Line numbers *1,2,3* are said to be faulty, but line number 4 did get substitution so line number ***4*** is not faulty.

Definition 4 (Treatment of Global Variables)
To obtain dependences form global variables we are dealing with the following features

1. Global variables impact global variables

If a global variable depends upon global variable in a program then we use similar rules to derive dependences from simple statements.

For an assignment statement $g = a + g$ the dependences are (g, a), (g, g).

2. Local variable impact global variables

If a global variable depends on a local variable of a method and the return variable also depends upon the same local variable then we can compute dependences as: Let d be the dependences of the method m and let l be a local variable, g the corresponding global variable and x be the returning variable . The dependences after method invocation is given by $\{ (g, x) \mid (g,l) \}$, where x is $(x \in return)$.

3. Formal variables impact global variables dependences

Let d be the dependences of a method m and let f be a formal parameter, a be an actual parameter and g the corresponding global variable. The dependences after method invocation is given by $\{ (g, a) \mid (g, f) \in d \}$.

If we assume an invocation to be abnormal we introduce a single variable for every occurrence of a certain procedure. For recursive invocations (in all cases where we obtain an cyclic call graph) we have to perform a fix-point computation. In order to guarantee that the computed dependences increase monotonically w.r.t. the subset relation like $d_n = d_n + 1$. Computing fixed point we add these dependences to overall dependences.

Fig. 4 shows the fixed point computation.

```
INPUT : while C do {B} end
OUTPUT: D (Dependences)
initialize d = d0
initialize d_Prev = null
do
{
    d_Prev = d
    d = d_Prev ∪ (d_Prev ∘ d_0)
}
while (!d.equals(d_Prev))
alldep = (alldep ∘ d)
```

Fig. 4. Algorithm of Fixed Point Computation

In

Fig. 4 d_0, d, d_{Prev} are variables which are storing pairs of dependences, where d computes new dependences, d_{Prev} stores previous dependences. d comprises of block dependencies from while loop **while C do { B } end**. The return statement stores overall dependences into d after finding fix-point. Function *union* adds both dependences $d_{Prev} \cup (d_{Prev} \circ d_0)$. The composition operator ∘ ensures that no information is lost during computing the overall dependency relation. Union operator in the algorithm shows that dependencies are adding after every iteration of loop. The condition of the *do {} while* loop ensures that whenever previous and new dependences are same we reached fix-point with finite number of iterations. After reaching fix-point we have to terminate loop and add computed dependence pairs into overall dependences d.

Theorem 1 (Fixed Point Computation Theorem)
The fixed-point computation algorithm computes a fix-point from repetitive invocation within a finite number of iterations.

Proof. We prove this theorem in two steps: First we prove that the dependencies are increasing monotonically. Second these dependencies should become equal within finite number of iterations at one point which is a fix-point.

- Dependencies are increasing monotonically i.e $\forall i\ d_{i+1} \supseteq d_i$.
  From the above algorithm, we know that $d_{i+1} = d_i \cup \{d_i \circ đ\}$. The computed dependences of block statements are stored in d. Because of the union operator it is obvious that $d_{i+1} \supseteq d_i$. This $(d_i \circ đ)$ are the new dependences which are added to the old dependency set. This leads to a monotonically increasing amount of dependences.
- Fix Point Computation i.e. $\exists i\ d_{i+1} = d_i$.
  We know that set of variable v is finite. Hence, $d_i = V \times V$ is finite and the upper bound of the dependency computation. From this follows that iteration an i exists when $d_{i+1} = d_i$.

(Wieland's 2001 (Fix-point Computation)) Complexity of computing dependences from while loop to reached fix-point in a finite number of iterations.

Theorem 1 shows the complexity of while loop for computing the dependences using the above algorithm. In [Wieland 2001], the author proves the theorem depending upon the number of variables used in dependences set. This example represents the worst case.

In Fig. 5 we call method ***foo*** recursively. Here we show that the dependences of recursively methods are in this fashion with using fixed point computation. The dependences of calling method ***foo*** has following dependences ***{ (y,x), (z, y), (z, x), (res, z), (res, y), (res, x) }***. The definition 4 ensures that we substitution of local, global variables are derived correctly.We use fixed point algorithm to find $d_n = d_n + 1$. In the *fooexample* method line number 5 has an assignment statement *t = foo(a, b)* that calls a method. Now we have to subsitute formal into actual parameters from computed dependences of calling method. After substitution we derived following dependences *(t, a),(t, b),(t, res)*.

```
1.    public int fooexample {
2.           int t, a, b;
3.           …
4.           …
5.           t = foo(a, b)
5.           //{(t,a),(t,b),(t,res)}
      private int foo(int x, int y) {
6.           int res=0;
7.           int z=1;
8.           if ( x < 0 )
9.              y = x;
             else
10.             z = foo(x-1, y);
11.          res = z + y;
12.          return res;
12.    {// (y, x), (z, y), (z, x), (res, z), (res, y), (res, x)}
12.  }
```

Fig. 5. Recursively call foo function

4. Example applying on fixed point computation

We show the small example to find fixed point over transitive relations. In an example program in Fig.6 we compute dependences step by step fashion to show that how we reach fixed point by using the algorithm in

Fig. 4. The body of while loop iterates i times.

In particular we are interested to compute dependences from the body of loop and summarized the loop dependences. Obviously all variables are used in the the body of loop *(a, b, c, d and i)*, which depends on variable *i*, because *i* appears in the condition of while loop. We did not show the computed dependences of i variable in Fig.6 and the graph's Fig. 7. The dependences of i are explicitly computable and do not make any changes during iteration of loop. This is the reason we are not presenting these dependences. The dependences of other variables are presented in the below. The algorithm presented presented earlier outlines the method of computing dependences from recursive invocation.

The dependences of the variables a; b; c and d are depicted in the following cases:

- (i = 0): and d_0 = {(a, b), (b, c), (c, d), (d, e)}
- (i = 1): and d_1 = {(a, b), (b, c), (c, d), (d, e), (c, e), (a, c), (b, d)}
- (i = 2): and d_2 = {(a, b), (b, c), (c, d), (d, e), (c, e), (a, c), (b, d), (a, d), (b, e)}
- (i = 3): and d_3 = {(a, b), (b, c), (c, d), (d, e), (c, e), (a, c), (b, d), (a, d), (b, e), (a, e)}
- (i = 4): and d_4 = {(a, b), (b, c), (c, d), (d, e), (c, e), (a, c), (b, d), (a, d), (b, e), (a, e)}

- The given specification from Fig. 6 and computed dependences are equal. So we reached fix point with finite number of iteration. We find $d_3 = d_4$ is a fix-point where both dependences are equal. Which ensures $d_n = d_{n+1}$.

All variables of assignment statements are depend upon variable i $\leftarrow$ (a, b, c, d).

SPEC {(a, b),(b, c),(c, d), (d, e), (c, e), (a, c),(b, d), (a, d), (b, e),(a, e)}

```
1. public int WhileExample {
2.     int a, b, c, d, e;
3.     int i = 0;
4.     ....
5.     ....
6.     while (i ≤ 5)
7.     {
8.           a = b;
9.           b = c;
10.          c = d;
11.          d = e;
12.          i = i + 1;
       }
    }
```

Fig. 6. Example program showing transitive dependencies in a loop

The graph shows the dependences of the while loop. There are five nodes in Graph including incoming and outgoing edges. Based on Theorem 1 we can prove that after a finite number of iterations we reached fixed point. we explain as follows:

Dependences are increasing monotonically i.e. $\forall i\ d_{i+1} \supseteq d_i$. We present the algorithm in

Fig. 4 for computing dependences of block statements. Because of the union operator it is obvious that $d_{i+1} \supseteq d_i$. This leads to a monotonically increasing amount of dependences. which is proven in Theorem 1. After calculating n we find fix-point within four iterationsn = 4. Therefore, we reached at fixed point after a finite number of iterations which is described in the algorithm. The graph shows the graphically representation of finding fixed point.

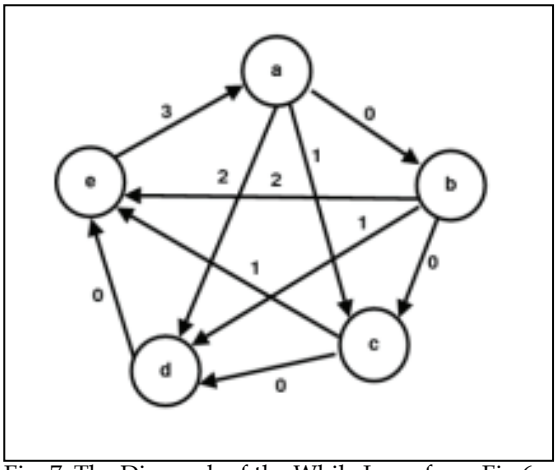


Fig. 7: The Diagraph of the While Loop from Fig.6

5. Experimental Results and Discussions

The proposed model has been implemented in Java using the Eclipse platform. In this section, we present the experiments that evaluate the result using dependencies in Java programs without using object-oriented features. Experiments were performed on a Intel Pentium 4 Workstation (3 GHZ,512 MB Memory) running Gentoo Linux (Gentoo Base System Version 1.4.9, Kernel version 2.6.5). The results are reported in Table 1.

For various examples programs, we introduced a single fault, and afterwards computed all single-fault diagnoses. Table 1 presents empirical results of programs with methods. We considered medium sized programs. The second column shows the lines of code from 26 to 509. The third column counts the number methods in the programs. The fourth column reports the number of diagnosis candidates. The 5th column gives the number of input variables and the last column shows the number of output variables.

In Table 1 the tested programs consisting of simple and multiple statements, loops, methods and global variables. Program specification consists of all variables i.e., input variables and output variables. In Fig. we show all these programs with minimum, maximum and average of diagnoses.

In Fig. 8 we presented a graph of programs with the number of faults depending on output variables (presented in Table 1). We used hundred iteration for every possible combination of output variables. It shows the minimum, maximum and mean of diagnoses in respect to

the output variables and the faults. In all graphs full line represent the minimum, dash line the maximum and dotted line the number of diagnoses candidates.

Programs	LOC	Methods	Diag.No	Input-VAR	Output-VAR
Adder	51	3	6	18	15
AddulseTime	378	21	10	98	54
Equation	26	4	4	13	5
MathFunctions	509	22	3	80	57
MethodTest1	42	3	12	14	11
MethodTest2	75	5	2	22	16
MethodTest3	46	5	3	20	14
Programs	**LOC**	**Methods**	**Diag.No**	**Input-VAR**	**Output-VAR**
MethodTest4	218	15	3	53	44

Table 1. Diagnosing candidates obtained by an introducing a single fault

Note, in the graph we consider only those diagnoses which has contradiction. This means that we never pick values which lead to no contradiction. Full line indicates that when we increase the number of output variables used in the specification, and then the number of diagnosis increases. The results indicate that our approach is feasible for detecting and localizing real cause of misbehaviour. The results presented there solely stem from procedural programs.

6. Limitation

Unlike previous approaches (Wieland 2001, Mayer. W & Stumptner M 2003) , the debugging approach introduced in this chapter specially intends localizing structural faults. In an account of this, we focus this discussion our model's weaknesses in detecting and localizing these faults.

First, the proposed model is not localizing faults caused by a erroneous target variable. For example, the following code snippet assigns the value of variable z to two different variables, namely, x and y.

1. **y ← z // should be x ← z**
2. **y ← z**

Due to (structural) error in line 1, we obtain *(y, z)* as a single dependence, but our specification enumerates *{ (x, z), (y, z) }* as dependences. Since *{(y, z)}* is not super set or equal to *{(x, z), (y, z) }*, we know some thing must be wrong. Thus, obviously, we can detect this bug. In localizing the detected misbehaviour's cause, we assume statement 1 to be abnormal and obtain $R_1 = \{ (x, \varepsilon_1)\}$. $R_2 = \{(x, z)\}$, *and* $R_1 \circ R_2 = \{(x, z)\}$. This results shows that we can not find any substitution σ that removes the given contradiction. Virtually, we can not correct our program by modifying solely the right-hand side of statement 1.

Allowing for model variable left hand side we can get rid of this deficiency: R1 = { (ε_1, z)} R_2, = { (y, z) }, which yields to $R_1 \circ R_2$ = { (ε_1, z), (x, z) }. Apparently, the substitution σ(ε_1) = x fulfils {(ε_1, z), (y, z)} is super set or equal to *{(x, z), (y, z)}* and thus we can localize this fault. However, this is approved solely in the rather are case that the substituted variable does not appear in any of the statements (on the right hand side) subsequent to variables substitution. Thus, this approach is not applicable in a practical setting.

Furthermore our model is only applicable to alias-free programs. This issue is further discussed in (Jackson 1995). In summary we conclude that, provided the fault appears on the right hand side of an assignment, our model allows for correcting these kind of structural faults as long as we can also detect this fault in terms of the specified dependences.

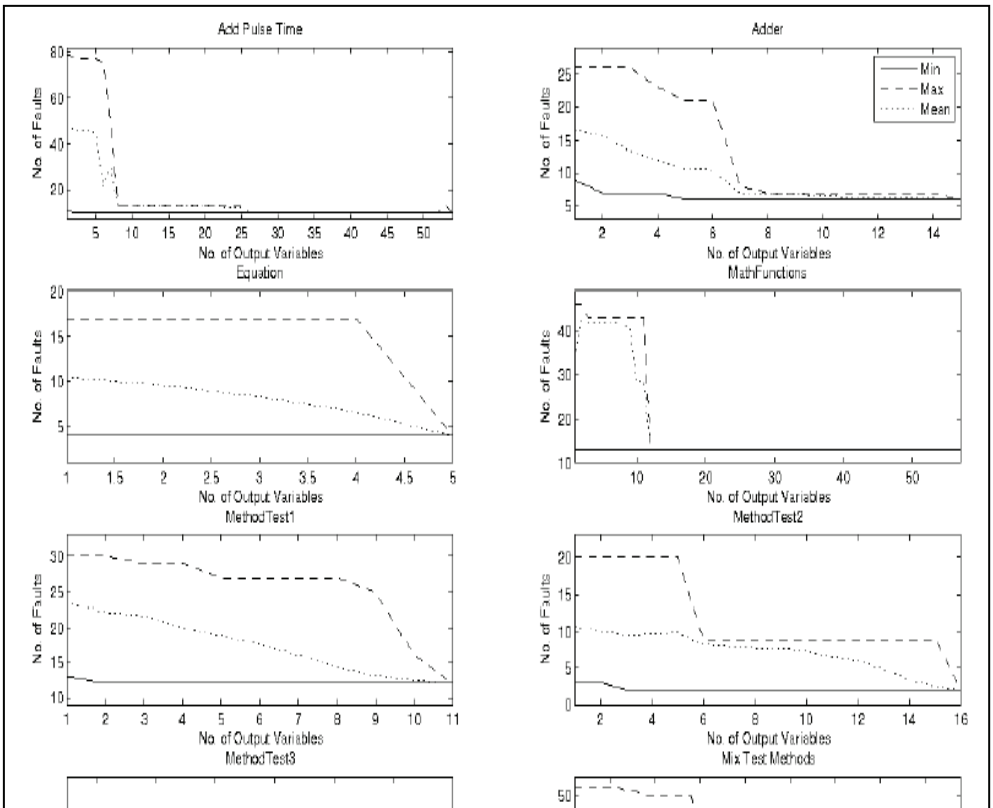


Fig. 8. Sensitivity Analysis of All Programs of Table 1.

7. Related Research

The author of (Jackson 1995) presents work which is closest to the work presented herein. This work employs abstract dependences for detecting rather than for localizing a fault. Furthermore in (Kuper 1989, Wieland 2001) the authors employ the notion of dependences for fault localization. In contrast to the latter approach we do not employ detected differences in variable values at a certain line in code but make use of differences between specified and computed dependences and thus also incorporate the structural properties of program and specification.

In the recent past the authors (Wotawa 2000, Wotawa et, al. 1999, Koeb and Wotawa 2004, Mayer and Stumptner 2003) developed models for different languages at various abstraction levels in the model-based context. In general, abstract modelling approaches sacrifice detail in favour of computational complexity whereas more detailed value-level models (Wotawa 2002, Peischl and Wotawa 2003) provide accurate fault localization capabilities but on the other hand require considerable computational resources in terms of space and computing power.

Although program slicing, as a lightweight technique, has seen successful application in fault localization (Agrawal et, al., 1993, Fritzson et, al., 1999, Lyle and Weiser 1987) its discrimination like MBSD (Wotawa 2000). In (Kuper 1989, Wieland 2001) the authors employ the notion of dependences for fault localization. In contrast to latter approach we do not employ detected difference in variable values at a certain line in code but use of differences between specified and computed dependencies and thus also incorporate the structural properties of program and specification. Thus, the models introduced in (Kuper 1989, Wieland 2001) can not deal with assertions or pre- and post conditions in a straightforward way.

The authors of (Wotawa 2000, Friedrich et, al., 1999, Wieland 2001) solely make use of concrete values in incorporating correctness information. These models do not allow taking advantage of arbitrary relationships between several variables or variables and constants. The author (Stumptner 2001) shows that localizing structural faults requires exploiting design information like assertions, and pre and post conditions.

Other approaches like (Johnson 1986) focus on novice programmers and make use of methods that help to find faults in the code by comparing the code with pre-specified problem formulations.

7. Conclusion

In this chapter we extend and present the novel model which detect and localize real faults from programs, comprising methods invocations and global variables. The computation of dependences from recursive invocation we found that every iteration has added new dependences and the number of dependences increasing monotonically. In order to guarantee that the computed dependences are increasing monotonically w.r.t. iterations, we find fixed-point where all dependences are equal with finite number of iterations. We presented an algorithm and proof for fixed point computation which ensures that no dependences loss during iteration and we always reaches fix-point after a finite number of iterations. Moreover, the approach is different to other available dependency-models and provides better results for medium sized programs.

A future research challenge is the formal and empirical evaluation of the modelling approaches when apply it to real object-oriented programs.

8. Acknowledgments

This work was supported by the Higher Education Commission, Islamabad, Pakistan under its research and development funding. The author would like to thank Prof. Franz Wotawa and Bernhard Peischl for their valuable comments and advice during my stay at Graz University of Technology, Austria.

8. References

Peischl. B & Wotawa F. (2003). Model-Based Diagnosis or Reasoning from First Principles, *IEEE Intelligent Systems*, 18(3), (May-June 2003) page numbers (32—37), 1541-1672.

Peischl B.; Soomro S. & Wotawa F. (2006). Abstract Dependence Model in Software Debugging. *Proceedings of the 17th International Workshop on Principles of Diagnosis (DX-06)* , France.

Jackson D. (1995). Aspect: Detecting Bugs with Abstract Dependences, *ACM Transactions on Software Engineering and Methodology*, 4(2), (April 1995) page numbers (109-145), 1049-331x.

Wieland D. (2001). Model Based Debugging of Java Programs Using Dependences. *PhD*

Thesis, Vienna University of Technology, Computer Science Department, Institute of Information Systems (184)}, Database and Artificial Intelligence Group (184/2), Vienna, Austria.

Koeb D. & Wotawa F. (2004). Introducing Alias Information into Model-Based Debugging. *Proceedings of 16th European Conference on Artificial Intelligence (ECAI)*, IOS Press, pp 833--837, Valencia, Spain.

Wotawa F.(2000). On the Relationship between Model-Based Debugging and Program Slicing, *Artificial Intelligence*, 135(1-2) (February 2002), page numbers (124—143), 0004-3702.

Wotawa. F(2002). Debugging Hardware Designs using a Value-based Model, *Applied Intelligence*, 16(1) , (January-Fabruary 2002) page numbers (71—92), 0924-669x.

Wotawa F. & Soomro S (2005). Using abstract dependencies in debugging. *Proceedings of 19th International Workshop on Qualitative Reasoning QR-05*, pp. 23--28, Austria.

Wotawa F. & Soomro S (2005). Fault Localization Based on Abstract Dependencies. *Proceedings of the 18th Conference on International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE 2005)*, Lecture Notes in Artificial Intelligence (LNAI),pp. 357-359, Springer Verlag, Italy.

Friedrich G.; Stumptner M. & Wotawa F.(1999). Model-based Diagnosis of Hardware Designs, *Artificial Intelligence*, 111(1-2), page numbers (3—39), 0004-3702.

Agrawal, H.; A. Demillo; Richard & H. Spafford; Eugene (1993). Debugging with dynamic slicing and backtracking Softw. *Practice and Experience*, 23(6), (June 1993), page numbers (589—616) , 0038-0644.

Lyle J.R. & Weiser. M (1987). Automatic Program Bug Location by Program Slicing. *Proceedings of the 2nd International Conference on Computers and Applications*, pp. 877--882, Beiging (Peking), China.

Johnson W. Lewis (1986). Intention-Based Diagnosis of Novice Programming Errors, In: *Morgan Koffman*,(Ed.), page numbers (333) Los Altos.

Weiser M. (1984). Program Slicing, *IEEE Transactions on Software Engineering*, 10(4), (July 1984) , (439-449) 0-89791-146-6.

Stumptner M. (2001). Using Design Information to Identify Structural Software Faults, *Proceedings of the 14th Australian Joint Conference on Artificial Intelligence*, pp. 473-486, London, Springer-Verlag, UK.

Fritzson P.; Gyimothy T.; Kamkar M. & Shahmehri N. (1999). Generalized Algorithmic Debugging and Testing, *Proceedings of the ACM SIGPLAN'91 Conference on Programming Language Design and Implementation,* pp 317—326, Toronto, Canada,

Kuper Ron. I. (1989). Dependency Directed localization of software bugs. *Technical Report* AI-TR 1053, MIT AI Lab, May USA.

Mayer W. & Stumptner M. (2003). Extending Diagnosis to Debug Programs with Exceptions. *Proceedings of the 18th IEEE International Conference on Automated Software Engineering (ASE), Montreal, IEEE Conferences on Automated Software Engineering*, pp 240--244, Montreal, Canada.

Soomro. S. (2007). Using Abstract Dependences to Localize Faults from Procedural Programs. *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications (AIA),* pp-180-185, Innsbruck, Austria.

5

Neural Networks Applied to Thermal Damage Classification in Grinding Process

Marcelo M. Spadotto, Paulo Roberto de Aguiar, Carlos C. P. Sousa, Eduardo C. Bianchi
Sao Paulo State University – Unesp – Bauru Campus
Brazil

1. Introduction

Throughout the course of human progress from prehistoric times until now the technological world was characterized by the development and improvement of new methods to control the environment. One of the most obstacle to overcome in order to reach the complete automation of machining process within the integrated and flexible manufacturing systems is the development that can be named a non-human-assisted machining, that is, a process in which the moment for tool change, the tool change itself and the change of the grinding conditions no longer need the human being assistance. Thus, the development of monitoring and control systems in real time is of great importance. High temperatures in grinding process are the main source of thermal damages to the ground surface, which is a visible manifestation in steels known as grinding burn. Depending on the temperature reached in the grinding zone a burn degree on the part surface can be observed which is due to temper color from very thin oxide layers. One of the challenges found in the implementation of intelligent grinding process is the automatic detection of surface burn of the parts. Several systems of monitoring have been assessed by researchers in order to control the grinding process and guarantee the quality of the ground parts. However, monitoring techniques still fails in certain situations where the phenomenon changes are not completely obtained by the employed signals.

Several monitoring systems which use force or power and acoustic emission sensors have been assessed by researchers to control surface burn in grinding (Aguiar et al.,2002; Aguiar et al., 1998; Kwak & Song, 2001; Wang et al., 2001; Kwak & Ha, 2004; Dotto et al., 2006; Aguiar et al., 2006a; Aguiar et al., 2006b). However, those techniques still need improvements where the phenomenon variations are not entirely acquired by the signals used.

High temperatures in grinding are the main source of thermal damages to the ground surface. A visible manifestation of this damage in steels is grinding burn – a discoloration of the ground surface often visible directly to the naked eye or brought out by etching of the surface. Depending on the temperature reached in the grinding zone a burn degree on the part surface can be observed which is due to temper color from very thin oxide layers on the part surface. This layer of ferrous material is composed of Fe_2O_3, Fe_3O_4, and FeO membranes

from the free surface. At the onset of a grinding burn, the grinding force and the rate of wheel wear increase sharply, and the surface roughness deteriorates (Kwak & Song, 2001; Badger & Torrance, 2000). Other type of thermal damage is referred to as rehardening burn, which is caused by a metallurgical phase change in the material when the grinding temperature exceeds the austenizing temperature, creating a thin layer of hard, brittle, untempered martensite. To further exacerbate the problem, rehardening burn is also accompanied by secondary residual stress, because the newly formed material has a greater density than the original material (Badger & Torrance, 2000)

In this study, the neural network has been applied to classifying the burn degrees obtained on the surfaces of the parts in grinding. The parameters of acoustic emission (AE), power signals and others derived from these signals have been used as the inputs of the neural network. The characterization of the surface quality of the ground parts was done by visual analyses with the naked eye and also by the software developed (Dotto, 2004).

What makes this work distinguished from others is the use of grinding parameters as input to the neural network, which have not been tested yet in burn classification by neural networks. Besides, a high sampling rate data acquisition system was employed to acquire the raw acoustic emission and cutting power.

2. Grinding Burn Monitoring

High temperatures generated in the grinding zone can cause several types of thermal damage to the part, for instance grinding burn in the case of steels, excessive tempering of the superficial layer with possible rehardening and increase of the brittleness, undesirable residual stress of tension, decrease of fatigue-life performance and micro-cracks. The decrease of grinding power is needed in order to minimize the restriction of thermal damages. This can be obtained by utilizing a softer grinding wheel or a rougher dressing operation but both present disadvantages (Malkin, 1989)

Grinding burn occurs during the cutting process when the amount of energy generated in the contact zone produces an increase of temperature enough to provoke a localized phase change in the material of the part. Such occurrence can be visually observed by the discoloration of the part surface (Malkin, 1989; Kwak & Song, 2001; Kwak & Ha, 2004). The burning is expected to occur when a critical temperature is exceeded in the grinding zone. He estimated a temperature rise of 720°C for burning to occur (Malkin, 1989).

Burn in steels is characterized by a visible bluish temper color on the ground surface. In steel, due to the burning phenomenon, the temper color changes from light brown to dark brown to violet to blue, in that order, depending on the severity of burn (Malkin, 1984; Nathan et al., 1999; Badger & Torrance, 2000; Liu et al., 2005).

The root mean square value (RMS) of the acoustic emission signal has been the main parameter studied in previous grinding researches over a frequency band carefully selected. This signal has been a good parameter because it is rich in sound waves carrying a lot of useful information (Lee et al., 2006; Liu et al., 2006).

Aguiar et al. (2002) has demonstrated in their investigation that the combination of the acoustic emission (RMS) signal and the electric power signal of the electric motor that drives the wheel can provide meaningful parameters to indicate when grinding burn takes place. From the combination of these signals they obtained the parameter referred to as DPO for burn detection in grinding. This parameter consists of the relationship between the standard

deviation of the RMS acoustic emission and the maximum value of the electric power in the grinding pass. The equation (1) describes the mentioned parameter.

$$DPO = S_{EA} P_{\max} \tag{1}$$

Where S_{EA} is the standard deviation of the RMS acoustic emission; and P_{max} is the maximum value of the electric power.
In the hope of more sensitivity to detect grinding burn, Dotto et al. (2006) has proposed a new parameter referred to as DPKS, which also utilizes the RMS acoustic emission and the power signals. This parameter is defined according to equation (2).

$$DPKS = \left(\sum_{i=1}^{i=m} (POT(i) - S(POT))^4 \right) \cdot S(EA) \tag{2}$$

Where i is the power index that varies from 1 to m samples per grinding pass; m is the number of samples of the pass; POT(i) is the instantaneous value of the power; S(POT) is the standard deviation of the power in the pass; S(EA) is the standard deviation of the RMS acoustic emission in the pass.
The statistics known as Constant False Alarm Rate (CFAR) and Mean Value Deviance (MVD) were employed successfully for detection of grinding burn (Wang et al. 2001; Aguiar et al., 2006b). The equation 3 represents the CFAR and the equation 4 the MVD.

$$T_{cpl}(X) = \frac{\sum_{k=0}^{M-1} X_k^{\upsilon}}{\left(\sum_{k=0}^{M-1} X_k \right)^{\upsilon}} \tag{3}$$

Where X_k is the *k-th* is the magnitude-squared FFT bin, υ is a changeable exponent and 2M is the total number of FFT bins (due to conjugate symmetry, only half of the magnitude-squared FFT bins need be interrogated). Respectively υ =1 and $\upsilon=\infty$ correspond to the energy detector and $max\{X_k\}$; $2<\upsilon<3$ provides good performance in a wide range.

$$T_{mvd}(X) = \frac{1}{M} \sum_{k=0}^{M-1} \log \left[\frac{\overline{X}}{X_k} \right] \tag{4}$$

Where $\overline{X}$ is the mean value of $\{X_k\}$; M and X_k have the same meanings as in the CFAR statistic.

3. Neural Network and its Application to Grinding

Neural networks are composed of many non-linear computational elements operating in

parallel fashion. Neural networks, because of their massive nature, can perform computations at a higher rate. Because of their adaptive nature in using the learning process, neural networks can adapt to changes in the data and learn the characteristics of input signals. Learning in a neural network means finding an appropriate set of weights that are connection strengths from the elements of this layer to the elements of a next layer (Kwak & Ha, 2001).

There are three layers in a network, namely the input layer (which receives input from the outside world), the hidden layer (between the input and the output layers) and the output layer (the response given to the outside world) (Nathan et al. 1999). The neurons of different layers are interconnected through weights. Thus, a neural network is constituted by processing elements at different layers, interconnections between them, and the learning rules that define the way in which inputs are mapped on to the outputs. The usefulness of an ANN comes from its ability to respond to an input pattern in a desirable fashion, after the learning phase. As such, the processing units receive inputs and perform a weighted sum of its input values using the connection weights given initially by the user. This weighted sum is termed the activation value of the neuron, given by:

$$u = \sum_{i=1}^{N} w_i x_i + \theta \tag{5}$$

where w_{ij} is the weight interconnecting two nodes i and j; x_i is the input variable; and u is the threshold value. During the forward pass through the network, each neuron evaluate an equation that expresses the output as a function of the inputs. Using the right kind of transfer function is therefore essential. A sigmoidal function can be used for this purpose, and is given by:

$$f(x) = 1 \Big/ \left(1 + e^{-u}\right) \tag{6}$$

Depending on the mismatch of the predicted output with the desired output, the weights are adjusted by back-propagation of error, so that the current mean square error (MSE) given by the following equation is reduced:

$$MSE = \frac{1}{2NK} \sum_{n=1}^{N} \sum_{k=1}^{K} \left(b_{nk} - S_{nk}\right)^2 \tag{7}$$

where N is the number of patterns in the training data, K is the number of nodes in the network, b_{nk} is the target output for the *n-th* pattern and s_{nk} is the actual output for the *n-th* pattern.

Still, it should be noted that the MSE itself is a function of the weights, as the computation of the output uses them. During this learning phase of the network the weights and the threshold values are adapted in order to develop the knowledge stored in the network. The weights are adjusted so as to obtain the desired output. The problem of finding the best set of weights in order to minimize the discrepancy between the desired and the actual

response of the network is considered as a non-linear optimization problem (Nathan, 1999). The most popularly used learning algorithm, namely the back-propagation algorithm, uses an interactive gradient-descent heuristic approach to solve this problem. Once the learning process is completed, the final set of weight values is stored, this constituting the long term memory of the network, which is used later during the prediction process.
Previous investigations have proved the efficiency of the artificial neural networks in the prediction of grinding burn (Wang et al., 2005; Kwak & Song, 2001; Kwak & Ha, 2001; Nathan et al., 1999; Aguiar et al., 2005, Spadotto et al., 2006). Thus, this technique is very promising and can also be applied successfully to industrial automation in a flexible and integrated fashion.

4. Methodology and Results

A surface grinding machine from Sulmecânica manufacturer, Brazil, model RAPH-1055 was used in the grinding tests. The grinder was equipped with an aluminum oxide grinding wheel, from Norton Manufacturer, Model ART-FE-38A80PVH. A fixed acoustic emission sensor from Sensis manufacturer, model DM-42, placed near the workpiece and an electrical power transducer for measuring the electrical power consumed from the three-phase induction motor that drives the wheel were employed.
The workpieces for the grinding tests consisted of laminated bars of steel SAE 1020 ground in the shape of a prism with 150mm length, 10mm width and 60mm height. The grinding process took place along the workpiece length.
The power transducer consists of a Hall sensor to measure the electric current and a Hall voltage sensor to measure the voltage at the electric motor terminals. Both signals are processed internally in the power transducer module by an integrated circuit, which delivers a voltage signal proportional to the electrical power consumed by the electric motor. The acoustic emission and the power signal are further sent to the data acquisition board from National Instrument, model PCI-6011, which is installed onto a personal computer. The LabVIEW software was utilized for acquiring the signals and storing them into binary files for further processing and analysis. The acoustic emission sensor used has a broad-band sensitivity of 1.0 MHz. Its amplifier also filtered the signal outside the range of 50 kHz to 1.0 MHz. Figure 1 shows the schematic diagram of the grinding machine and instrumentation used.

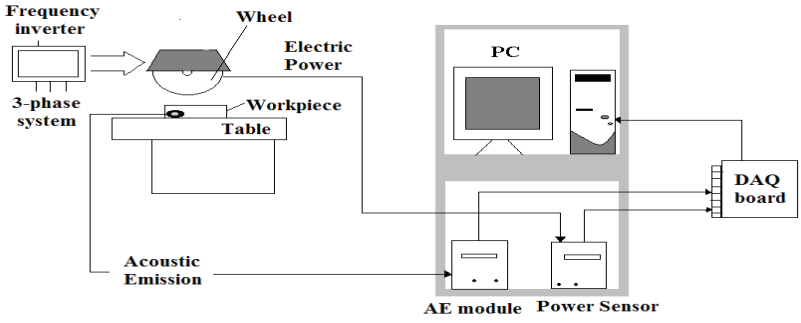


Fig. 1. Experimental setup.

The tests were carried out for 12 different grinding conditions, and subsequently the burn degrees (no-burn, slight burn, medium burn, and severe burn) could be visually assessed for each workpiece surface. Dressing parameters, lubrication and peripheral wheel speed were adequately controlled in order to ensure the same grinding condition for each test. The workpiece speed was set up at 0.033 m/s and the wheel speed at 30 m/s. The latter was maintained constant by adjusting the frequency of the induction motor on the frequency inverter, as the grinding wheel had its diameter decreased along the tests. The G-ratio, which is the volume of material removed per unit volume of wheel wear (Malkin, 1989), was set to 1, maintaining the dressing condition the same for all the tests. A water-based fluid was used with 4% concentration. Each run consisted of a single grinding pass of the grinding wheel along the workpiece length at a given grinding condition to be analyzed. The acoustic emission and cutting power signals were measured in real time at 2.0 millions of samples per second rate, and then stored onto binary data files for further processing. It is important to mention that the raw acoustic emission signal was acquired instead of the root mean square generally used.
The digital signal processing phase started after all the 12 tests were carried out and the data files stored. The digital signal processing of acoustic emission and power generated 7 new statistics as previously described, that is, the parameters DPO and DPKS, and the statistics CFAR and MVD. Seven structures were used for the neural network implementation as shown in Table 1. It can be noted in this table that besides the signals and statistics aforementioned the depth of cut a was also used as input of every structure.

Structure	Inputs	Structure	Inputs
I	Pot, AE, a	V	CFAR, a
II	DPO, a	VI	AE, a
III	DPKS, a	VII	Pot, a
IV	MVD, a		

Table 1. Neural network structures.

In this work, the back-propagation algorithm of neural networks, which is one of the learning models, was used. The following parameters were also found more suitable: downward gradient training algorithm; all data in the neural networks were normalized; training for 1000 epochs; square mean error value of 10-5. Cross-validation was used to estimate the generalization error of the model.
The outputs of the neural network was configured in a binary way according to the degree of burn obtained, that is, 0001 for no-burn, 0010 for slight burn, 0100 for medium burn, and 1000 for severe burn.
Each statistic was represented by a vector of 3000 samples for each test subsequently the digital processing of the acoustic emission and power signals. Initially, a visual analysis was carried out by naked eye on the part surfaces. Then, a quantification of the grinding burn on every part surface was done by a specific software for that purpose, which assessed the surface of a given part regarding the burn level through its digitalized picture (Dotto, 2004). Thus, a precise characterization of the burn levels occurred on the part surfaces was achieved. From the results of this characterization, input vectors were separated and

assigned to the corresponding type of burn. The input vectors were again divided into training, validation and test vectors. Then, the process of optimization for the neural network was carried out.

The architecture of the neural networks was determined according to the tests of number of neurons of the hidden layer, learning rate and momentum. As the problem dealt in this work consists of pattern classification, only a simple hidden layer was chosen (Haykin, 1994). With learning rate equal to 1 and momentum fixed to zero the neurons of the hidden layer were varied at steps of 5 up to 50 neurons. The optimum number of neurons of the hidden layer was estimated according to the mean square error of the validation set of each structure calculated in the training phase. Figure 2 illustrates the comparison of errors obtained for each number of neurons considered for the structure 2 (DPO and depth of cut).

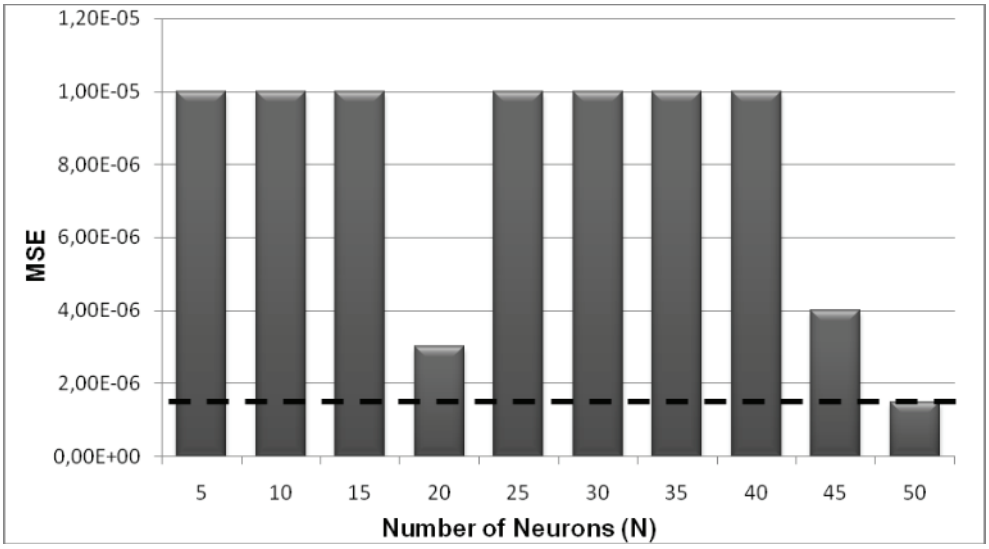


Fig. 2. Mean square error versus number of neurons of the hidden layer for structure 2.

In order to obtain the best values for learning rate and momentum the training was carried out for all aforementioned structures by varying these parameters. The proceeding for choosing the best pair momentum and learning rate was performed by fixing initially a value for momentum, varying the learning rate from 0.1 to 0.7 with step of 0.1. After the error curves were obtained, the pair momentum and learning rate was chosen based on the curve which presented the smallest oscillation with the smallest number of epochs. This process was repeated for values of momentum from 0.2 to 0.7, with step of 0.1. Thus, the best six pairs were obtained in the end of this process, choosing among these pairs the one which presented the smallest oscillation with the smallest number of epochs.

Figure 3 illustrates the process of choice of the pair momentum and learning rate from comparisons between two curves of the mean square error in function of the number of epochs. These curves were obtained following the training of the structure 4.

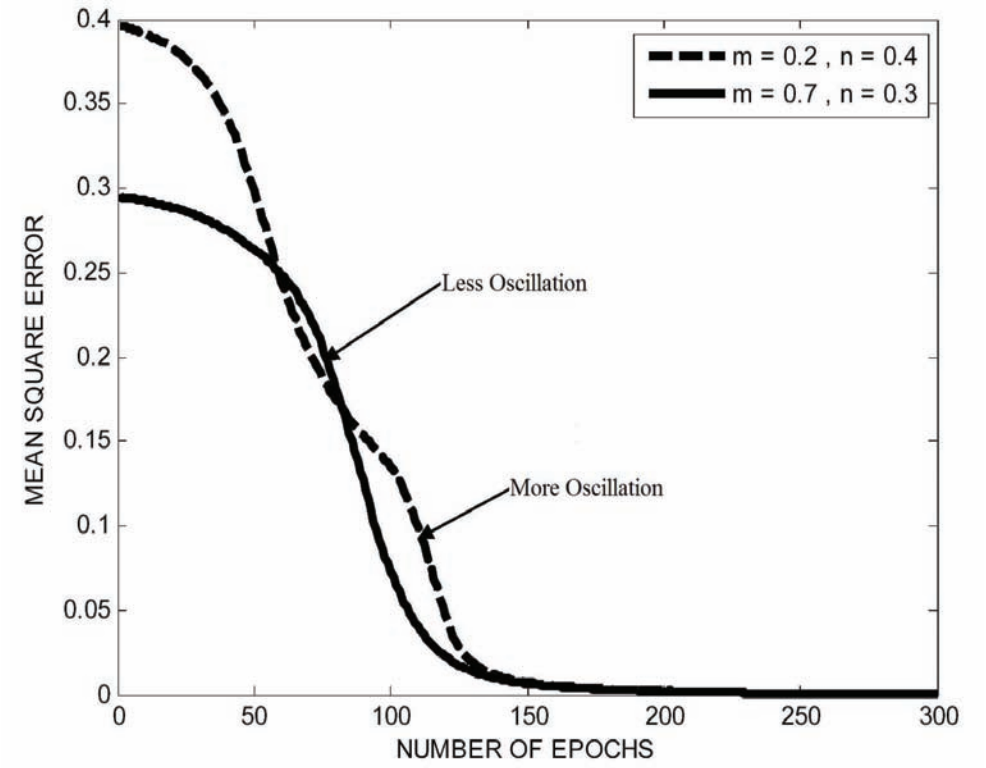


Fig. 3. Comparisons between two curves of the mean square error for a given pair momentum and learning rate for structure 4.

It can be observed in Figure 3 two curves of mean square error in function of number of epochs showing different behaviours, that is, the curve with momentum 0.2 and learning rate 0.4 presented more oscillation than the curve with momentum 0.7 and learning rate 0.3. Thus, the latter curve was chosen because it provided the smallest oscillation. This criterion is suggested by Haykin (1994) which states the learning rate and momentum lead to a local minimum in the error surface of the network with the smallest number of epochs.

On the other hand, the hidden neurons make the network to learn complex tasks by progressively extracting more significant characteristics of the input patterns (vectors). Moreover, because the change of the synaptic weight of the network is proportional to derivative of the activation function, it turns out that for the sigmoid activation function the synaptic weights are changed more intensely for those neurons of the network where the functional signal is in the middle of its interval. Based on these remarks, and also due to the previous experiences gotten from trial and error, the logsig activation function was chosen in this work for the neurons of the hidden and output layers.

On the basis of the optimization process of the neural network previously described in addition to the tests carried out for each structure, the best results for the all structures were obtained and presented in Table 2.

Structure	Neurons	Learning rate	momentum
I	3 – 35 – 4	0.7	0.6
II	2 – 50 – 4	0.7	0.3
III	2 – 45– 4	0.5	0.7
IV	2– 30 – 4	0.3	0.7
V	2 – 50– 4	0.7	0.3
VI	2 – 40 – 4	0.7	0.7
VII	2 – 20 – 4	0.5	0.3

Table 2. Configuration for the 7 neural network structures studied.

The neural networks of each structure were adjusted to have the correct parameters in the test phase for obtaining the number of neurons of the hidden layer, learning rate and momentum. The cross-validation method was employed with the training afterwards. The results for each structure were generated by inputting the corresponding data (signals and/or parameters) along with the depth of cut information, and the network output was interpreted in a bar graph fitting the form of the ground workpiece according to each burn level obtained. The digital picture of the workpiece with the corresponding bar graph for each structure was put all together for comparisons.

Figure 4 shows the results obtained when the signal vectors of test 2 not used in the training were inputted to the neural network. Thus, the data given to the neural network are different from those it was used to, testing this way its ability of classifying the burn levels.

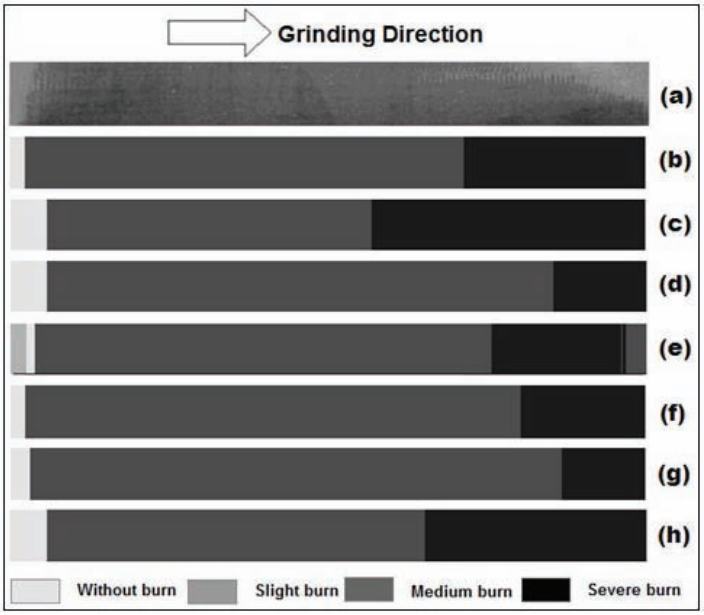


Fig. 4. Results obtained for Test 2; (a) Workpiece picture; (b) Structure 1; (c) Structure 2; (d) Structure 3; (e) Structure 4; (f) Structure 5; (g) Structure 6; (h) Structure 7.

It can be observed in Figure 4 that the structures were able to detecting well the changes in the burn levels occurred in the test 2. Some minor errors of classification were also observed as is the case of the structure 4, Figure 4(e), which has failed in classifying severe burn in the end of the workpiece, and the medium burn has been classified instead.
Following the classification performed for all structures, the percentage of success was calculated. Based on the comparisons of success percentages found for each structure, it was possible to determine the structure among all studied the one that presented more efficient at classifying the degrees of burn in the surface grinding process. Figure 5 shows the success rates in percentage illustrated by a bar graph as well as the grading of each structure. This ranking was carried out based on the general success rates for each structure, that is, considering the success of classifying workpieces with slight burn, medium burn, severe burn and no-burn.

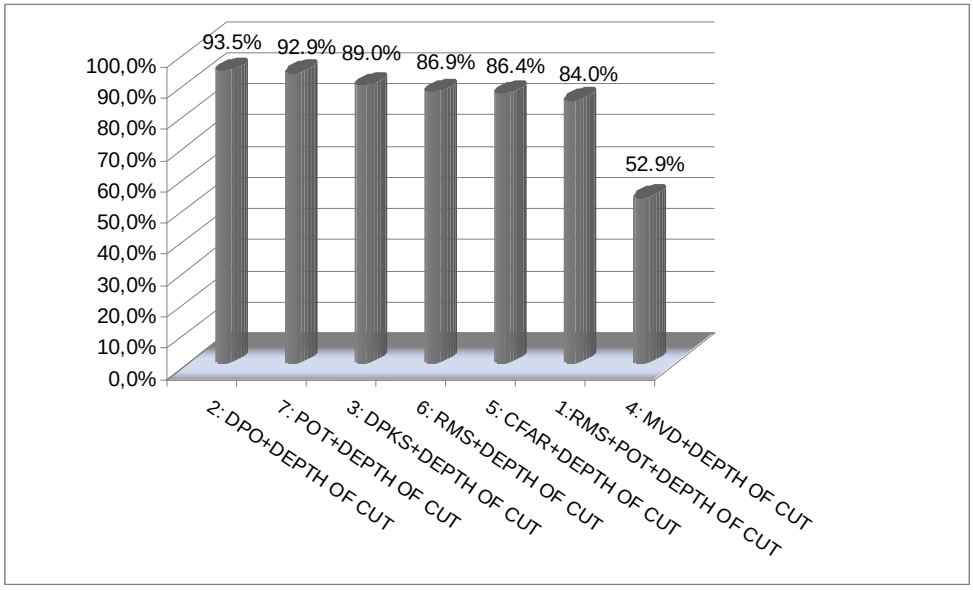


Fig. 5. Rate of success for each structure and the ranking obtained.

It can be observed in Figure 5 that all structures have presented a success rate quite good, with the exception of the structure 4 which has presented a success rate of only 52.9%. The structure having acoustic emission, power and depth of cut was supposed to own a better position in the grading since these signals are widely employed in the grinding process monitoring. On the other hand, structure 2 composed by DPO parameter and depth of cut has presented the best result. This fact demonstrates the parameter DPO proposed by Aguiar et al. (2002) actually have a great sensitivity when thermal damage takes place in grinding, and therefore it shows up an excellent input to the neural network for solving classification problems of burn degrees. It can be emphasized that all structures detected slight burn quite well, and the grading showed in Figure 5 was built based on the success rate for all degrees of burn studied.

5. Conclusion

The utilization of neural network of type multi-layer perceptron using the back-propagation algorithm guaranteed very good results. Tests carried out in order to optimize the learning capacity of neural networks were of utmost importance in the training phase, where the optimum values for the number of neurons of the hidden layer, learning rate and momentum for each structure were determined. Once the architecture of the neural network was established with those optimum values, the mean square error obtained during the training phase for the validation set proved that the neural network configuration was optimized, reaching values as lows as 1.0e-6. Moreover, the usage of cross-validation in the training phase was very important to right stop the training without overfitting occurrence, and then assuring a better generalization of the problem.

As all structures have detected correctly the degree of slight burn which is the first stage of change on thermal damage, it can be concluded that all structures worked well for classification of burn or non-burn occurrence.

The structure 2 with DPO and depth of cut as inputs presented the best results in comparison with the others studied. This can also be explained due to the parameter DPO combines the variations of the RMS acoustic emission and the maximum amplitude of the electric power during the grinding pass, resulting in an excellent tool for detection of burn degrees.

The differences of errors found among the structures 2, 6 and 7 are quite small, that is, less than 1% for the set of input #7, and 6.6% for the set of input #6 with respect to structure 2. Therefore, the acoustic emission and electric power signals can also be employed successfully as inputs to the artificial neural networks for classification of burn degrees in grinding.

7. Acknowledgments

This work was supported by FAPESP -The State of Sao Paulo Research Foundation, IFM - The Institute Factory of Millennium, and CNPq - National Council for Scientific and Technological Development (Grant # 304842/2006-0). Also, thanks go to the Laboratory of Grinding and the Laboratory of Data Acquisition and Signal Processing at Unesp, campus Bauru, for the assets given to this work.

8. References

Aguiar, P. R.; Willett, P. & Webster, J. (1998), Acoustic Emission Applied to Detect Workpiece Burn during Grinding. *International Symposium on Acoustic Emission: Standards and Technology Update*, S. Vahaviolos, Ed., Fort Lauderdale, FL. pp. 107-124.

Aguiar, P. R.; Bianchi, E. C. & Oliveira, J. F. G. (2002). A method for burning detection in grinding process using acoustic emission and effective electrical power signal, *CIRP Journal of Manufacturing Systems*, Vol. 31, Paris, 253-257.

Aguiar, P. R.; Bianchi, E. C.; Dotto, F. R. L.; Flauzino, R. A. & Spatti, D. H. (2005). Neural network applied to detect burn in grinding. *The IASTED International Conference on Artificial Intelligence and Applications*, Innsbruck, Austria, 831-836.

Aguiar, P. R.; França, T. V. & Bianchi, E. C. (2006a). Roughness and roundness prediction in grinding, *Proceedings of the 5th CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME '06)*, Italy, 183-188.

Aguiar, P. R.; Serni, P. J. A.; Dotto, F. R. L. & Bianchi, E. C. (2006b). In-Process Grinding Monitoring Through Acoustic Emission. *Journal of the Brazilian Society of Mechanical Sciences*, Vol. XXVIII, No. 1, 118-124.

Badger, J. A. & Torrance, A. (2000). Burn awareness - Understanding the causes of grinding burn helps alleviate the problem, *Cutting Tool Engineering Magazine*, Vol. 52, 12.

Dotto, F. R. L.; Aguiar, P. R.; Bianchi, E. C.; Serni, P. J. A. & Thomazella, R. (2006). Automatic system for thermal damage detection in manufacturing process with internet monitoring, *Journal of Brazilian Society of Mechanical Science & Engineering*, Vol XXVIII, No. 2, 153-160.

Dotto, F. R. L. (2004). Online automatic detection system of thermal damages in the grinding process via Internet (in Portuguese). *Master of Science Dissertation*, Unesp, Bauru-SP. 220p.

Haykin, S. (1994). *Neural Networks – A Comprehensive Foundation*, Macmillan, New York.

Kwak, J. S. & Song, J. B. (2001). Trouble diagnosis of the grinding process by using acoustic emission signals, *International Journal of Machine Tools & Manufacture*, Vol. 41, 899–913.

Kwak, J. S. & Ha, M. K. (2004). Neural network approach for diagnosis of grinding operation by acoustic emission and power signals, *Journal of Materials Processing Technology*, Vol. 147, 65-71.

Lee, D. E.; Hwang, I.; Valente, C. M. O.; Oliveira, J. F. G. & Dornfeld, D. A. (2006). Precision manufacturing process monitoring with acoustic emission, *International Journal of Machine Tools & Manufacture*, Vol. 46, 176-188.

Liu, Q.; Xun C. & Gindy, N. (2005). Fuzzy pattern recognition of AE signals for grinding burn, *International Journal of Machine Tools & Manufacture*, Vol. 45, 811-818.

Liu, Q.; Xun C. & Gindy, N. (2006). Investigation of acoustic emission signals under a simulative environment of grinding burn, *International Journal of Machine Tools & Manufacture*, Vol. 46, 284-292.

Malkin, S. (1989). *Grinding Technology - Theory and Applications of Machining with Abrasives*, Ellis Horwood Limited Publishers, Chichester, Halsted Press: a division of John Wiley & Sons.

Nathan, R. D.; Vijayaraghavan, L. & Krishnamurthy, R. (1999). In-process monitoring of grinding burn in the cylindrical grinding of steel, *Journal of Materials Processing Technology*, Vol. 91, 37–42.

Spadotto, M. M. ; Aguiar, P. R.; Prazeres, R. C. ; Souza, C. C. P. & Bianchi, E. C. (2006). Artificial neural network applied to detection of burning in grinding process. 5th International Conference on Mechanics and Materials in Design, Porto, Portugal.

Wang, Z.; Willet, P.; Aguiar, P. R. & Webster, J. (2001). Neural network detection of grinding burn from acoustic emission, *International Journal of Machine Tools & Manufacture*, Vol. 41, 283-309.

Wang, J. Z.; Wang, L. S.; Li, G. F.; & Zhou, G. H. (2005). Prediction of surface roughness in cylindrical traverse grinding based on ALS algorithm. *Proceedings of the Fourth Internacional Conference on Machine Learning and Cybernetics*, Guangzhou, 549-554.

6

Motivation in Embodied Intelligence

Janusz A. Starzyk
Ohio University
U.S.A.

1. Introduction

Before artificial intelligence set its mind on developing abstract intelligent agents which can think, Alan Turing suggested training embodied machines equipped with sensors and actuators to accomplish intelligent tasks like understanding spoken English (Turing, 1950). Looking at intelligence from a different perspective, philosopher, and neuroscientist Francisco Varela (Maturana & Varela, 1980), (Varela et al., 1992) proposed the embodied philosophy of living systems which argues that human cognition can only be understood in terms of the human body and the physical environment with which it interacts. What may seem to be a revelation from a historical perspective, early robots built on cybernetic principles demonstrated goal-seeking behavior, homeostasis (the ability to keep parameters within prescribed ranges), and learning abilities (Walter, 1951), (Walter, 1953). These were precursors for embodied intelligence. Perhaps the most influential figure in developing embodied intelligence as a methodology to design intelligent machines is Rodney Brooks. He suggested the design of intelligent machines through interaction with the environment driven by perception and action, rather than by a prespecified algorithm (Brooks, 1991a).
Like Hans Moravec before him (Moravec, 1984), Brooks suggested that locomotion and vision are fundamental for natural intelligence. He also observed that the environment is its best model and that representation is the wrong "unit of abstraction". These simple observations revolutionized the way people think about intelligent machines and created a field of research called "embodied intelligence". The growth of interest in embodied intelligence that followed Brooks' works can be compared to the increase in research activities in artificial intelligence that followed the famous Dartmouth Conference of 1956 (McCarthy et al., 1955) or the revival of neural network research in the 1980s. His approach revived the field of autonomous robots, but as robotics thrived, research on embodied intelligence started to concentrate on the commercial aspects of robots with a lot of effort spent on embodiment and a little on intelligence.
The open question remains: how to continue on the path to machine intelligence? Today, once again, artificial intelligence research is focused on specialized problems, such as ways to represent knowledge, natural language and scene understanding, semantic cognition, question answering, associative memories or various forms or reinforcement learning. In recent years, the term "general artificial intelligence" was coined as something new, incorrectly implying that the original idea of AI was something less than to develop a natural intelligence.

Brooks decided to build intelligent autonomous creatures that work in a dynamically changing environment. He pointed out that he is not interested in finding how humans work, nor in philosophical implications of creatures he creates. He let them find their own niche to operate in. Although he would like humans to perceive these creatures as intelligent, he does not define what this would mean. He would like these creatures to be able to adapt to changes in the environment by gradual changes in their capabilities. Each creature should have a purpose of being; it should maintain and pursue multiple goals, choosing which goal to implement based on the environmental conditions. In addition, the complexity of a creature's behavior would reflect the complexity of the environment in which it operates rather than its own.

Proposed by Brooks, subsumption architecture leads to independent sensory-motor control structures that work concurrently and are designed such that lower level skills are subsumed by the higher levels. Thus multiple parallel sensory-motor paths must be implemented to control the creature's behaviour. He argues that no central control or representation is needed. Instead individual robot skills are built layer after layer each one composed of a simple data driven finite state machine with no central control.

Brooks seems to reject the connectionist (and implicitly neural network) approach. The finite state machines he uses to control his creatures must be explicitly programmed to perform certain actions. However, this explicit engineering approach that works successfully on very low levels of subsumption architecture does not have a natural mechanism for self-organization from which higher level skills could evolve. Machine learning, which may be a critical element of intelligence, is almost left out of consideration. Indeed, the only learning that takes place in embodied agents is based on simple neural network structures. But years of development of classical neural networks failed to deliver acceptable forms of learning due to the catastrophic interference observed in generic neural networks (McCloskey & Cohen, 1989). Yet, in my opinion, learning distinguishes the intelligent from the unintelligent. Thus, subsumption architecture may be a clever way to design autonomous robots with reactive control, but it is not a mechanism that may scale up to human level intelligence. I claim that many years after Moravec's article, subsumption architecture has still failed to solve fundamental problems of embodied intelligence and needs a major revamp.

Brooks requires that machine uses multiple, data driven, parallel processing mechanisms that control machine's behavior. Yet, he clearly differentiates his approach from this of neural networks. He claims that there is no obvious way to assign the credit or blame in neural networks for a correct or incorrect action. He pointed out that the most successful learning techniques for embodied robots use reinforcement learning algorithms (like Q-learning) rather than parallel processing neural networks. He stressed dense connectivity of neural networks that are in striking contrast to his system of loosely connected processes.

By rejecting the connectionist approach and self-organization of machine architecture, Brooks denied his subsumption architecture the flexibility to integrate evolved lower level functions into more complex levels without explicit interference of a human designer. From a system engineering point of view, each subsequent step in system complexity requires exponentially harder design effort and understanding of what the creature can do and how it does it. Yet as Brooks observed, this was not the case in nature. It took nature over 3 billion years to create insects from the primordial soup, but it took only 200 million more years to create mammals, and only 15 million years for the transformation of great apes to

modern man about 3 million years ago, with all major developments of the civilized world within the last 10,000 years. It seems that in nature it is easier to append a primitive brain to create a complex brain capable of abstract thought, than it is to learn locomotion and survival skills in primitive brains. While this may justify an approach in which a machine's reflexes are developed first, the lack of a mechanism to add complexity at a low design cost is a major problem that cannot be left to chance.
Brooks rightfully indicated that development of intelligence should proceed in a bottom-up fashion from simpler to more complex skills, and that the skills should be tested in the real environment. He rightfully criticized the symbolic manipulation approach for requiring that a complete world model is built before it can be used. He also rejected knowledge representation as ungrounded. However, instead of proposing an approach that bridges the gap between processing raw sensory and motor signals, symbolic knowledge representation and higher level manipulation of symbols, he assumed a constructionist approach with no hint of how to develop natural learning. This denying the need for representation was criticized by Steels (Steels, 2003), who pointed out that representations are internal conceptualizations of the real world and thus ought to be acceptable to the embodied intelligence idea. So, in spite of its great success in building creatures that can move in a changing environment, subsumption architecture failed to create foundations for intelligence. To paraphrase Brooks' own words - the last seventeen years have seen the discipline coasting on inertia more than anything else.

In this chapter, I will present a path for further development of the embodied intelligence idea. First, I will directly address the issue of intelligence. The problem with Brooks' approach is not that he did not define intelligence, leaving it to philosophers, but that he accepted any autonomous behaviour in a natural environment as intelligent. While it is true that survival-related tasks form a necessary basis for development of intelligence, they alone do not constitute one. Is an amoeba intelligent? How about virus or bacteria? If we expect an intelligent behaviour, we need to define one. Instead of defining embodied intelligence, Brooks wants to design creatures that are seen as intelligent by humans. Still, he knows very well that a complex behaviour may result from a very simple control process. So how will he decide if an agent is intelligent? In fact, he is not interested in designing intelligent agents but instead in building working autonomous robots. Yet he claims that those reactive machines are intelligent.

Why might this be important? For a number of years in embodied intelligence, process efficiency dominated over generality. The principle of cheap design in building autonomous agents promoted by Pfeifer (Pfeifer & Bongard, 2007) supports this philosophy. It is cheaper and more cost effective to design a robot for a specific task than it is to design one that can do several tasks, and even more so than to design one that can actually make its own decisions. A computer can compute many times faster and more accurately than man, but it takes a human to understand the results. A machine can translate foreign speech, but it takes a human to make sense out of it. Thus there is a danger of using the principle of chip design to design a robot with no intelligence and call it intelligent as long as it does its job. This must not happen if we want to continue on the path to build more and more intelligent machines. So the question is what traits of embodied intelligence development must really be stressed, and where must the design effort concentrate?

2. Design Principles for Embodied Intelligence

The principles of designing robots based on the embodied intelligence idea were first described by Brooks (Brooks, 1991b) and were characterized through several assumptions that would facilitate development of embodied agents**.**

The first assumption was that the agents develop in a changing environment which they can manipulate through their actions and perceive through their senses. An important assumption was that there was no need to build a model for the environment; instead we could use the environment the way it is. These assumptions constrain the dynamics of agent-environment interaction. Based on Wehner's work (Wehner, 1987), Brooks suggested that evolutionary development led to the right form of interaction between sensory inputs and motor control provided by the nervous system. This led him to a design principle based on an ecological balance that must exist between the amount of information received, the processing capability and the complexity of the motor control.

Brooks rejects the need for explicit representations of environment or goals within the machine. Instead he uses active-constructive representations that permit manipulation of the environment based on graphically represented maps of environments. His statement that he does not represent the environment may be misleading. Just saying that this representation is different than traditional AI representation is not enough – a robot builds and maintains representations of the world. The fact that instead of planning ahead what to do next, an iterative map is used does not change the fact that some form of environment representation is needed. A local marker telling the robot where he is with respect to the map is also a form of environment representation.

Additional principles of designing embodied intelligence were characterized by Rolf Pfeifer (Pfeifer & Bongard, 2007) and include:

1) Principles of cheap design and redundancy. According to these principles design must be parsimonious and redundant. This means that by exploiting an ecological niche design can be simplified, while redundancy requires functionality overlap between different subsystems. Although these principles were not explicitly stated in Brooks' work, he stipulated them in his description of the design process.

2) Principle of parallel, loosely-coupled processes. This requires that intelligence emerges from interactions of lower level processes with the environment. This principle was in fact a foundation of internal organization of subsumption architecture based on Brooks' ideas and led to implementations of embodied agents that integrated many reactive sensory-motor coordination circuits using finite state machine architectures.

3) The value principle. This principle stands out among those adopted by Pfeifer as the one that tells a robot what is good for it. The agent may use this principle to decide what to do in a particular situation. In Brooks' work this is decided by competing goals but the goals are predetermined by a designer, and deciding which goal to pursue is also preset.

It was demonstrated that subsumption systems based on embodied intelligence ideas can anticipate changes in the environment and have expectations, can plan (Agre & Chapman, 1990), can have goals (Maes, 1990) and can do all of this without central control or symbolic representations.

In Brooks' article (Brooks, 1991b), an important issue related to learning in the subsumption architecture remains unsolved: how to develop methods for new behaviors, and change the existing behaviors. Brooks indicated that the performance of a robot might be improved by

adapting the ways in which behaviors change as a result of experience, however he does not say how this might be accomplished. He claims that thought and consciousness will emerge from interaction with the environment. While such a general statement is definitely true, based on nature's success in creating people who think and are conscious, there is no indication of how these may emerge in the subsumption architecture.

Pfeifer indicated that by allowing an agent to develop its own behaviors rather than having them programmed, additional properties may emerge (Pfeifer & Bongard, 2007). Although, unlike Brooks, Pfeifer admits that learning is an essential part of intelligence, he dismisses successes of machine learning fields as "almost entirely disembodied" and therefore not interesting. In addition he seems to deny the possibility of building embodied intelligence in the virtual world, and instead points out the necessity to bring it up entirely in mechanical robots. Yet there is nothing in the concept of embodied intelligence that precludes existence of a virtual embodied agent, as long as it has well-defined sensors and actuators. A virtual agent will be situated in a dynamically changing environment. Such an agent will perceive its environment through its sensors and act on it in a way similar to a robot that acts in the real world, and such an agent may do this in an intelligent way. In fact, considering the significant cost and design effort of building and maintaining robots, virtual agents should be the first rather than the last choice to develop ideas of embodied intelligence. And yes, development of good ideas and structural organization principles of signal processing elements in intelligent machines are what we need to solve the intelligence puzzle.

One of the motivations that Pfeifer uses in support of a developmental approach to cognition is the ontogenetic development of humans from children to adults, and he would like to see some form of implementation of the physical growth process. I see no such need, as a child may fully develop psychologically without the physical growth of its body. It's the brain of a child that needs to develop by experiencing the world, and the brain development is accomplished by learning proper behaviors rather than by a physical growth. In fact, the opposite may be true regarding topological complexity of the networks of neurons in the brain, as the brain of a young child has many more neural connections and therefore may have a higher ability to learn than the brain of an adult.

Pfeifer is right when he suggests that representing lower level attractor states as symbols provides a grounded way of bottom-up building of cognitive systems. This is in contrast to earlier views by Brooks, who denied that symbol manipulation may play a useful role in development of embodied intelligence. The symbols used in this bottom-up representation building are known only to the machine that holds them and cannot be explicitly defined and entered from outside (for instance by a programmer). Thus they are grounded in the machine's way to perceive and history of interactions with the environment.

Pfeifer acknowledges that the value system in embodied intelligence is murky to a similar degree as it is in biology, psychology or artificial intelligence. However, he states that the value is in the head of the designer rather than in the head of an agent. This approach to value learning is acceptable only for simple reactive systems that require external reinforcement to learn values and may not be sufficient for intelligent systems.

In reinforcement learning (Sutton, 1984), values are either associated with the machine's states or with activation of neurons in neural network implementation. However, state-based value learning is useful only for the simplest systems with a small number of states. The learning effort does not scale well with the number of states. If a system uses neurons

to learn and control its operation, then its number of states grows exponentially with the number of neurons and learning the values associated with all these states is difficult. In addition, a system that uses only external reinforcement to learn its values suffers from the credit assignment problem where credit or blame must be assigned to various parts of the system for an action that resulted in a reward or punishment (Sutton, 1984) , (Fu & Anderson, 2006).

Optimal decision making of human activities in a complex environment was rendered intractable by reinforcement learning. To remedy this deficiency of reinforcement learning, a hierarchical organization of complex activities was proposed (Currie, 1991). Expecting that a hierarchical system will improve reinforcement, Singh analyzed the case in which a manager selects its own sub-managers (Singh, 1992) who are responsible for their subtasks. Sub-managers had to learn their operation and their system of values. In a similar effort, Dayan (Dayan & Hinton, 1993) developed a system in which a hierarchy of managers was used to improve the reinforcement learning rate. It was demonstrated (Parr R. & Russell, 1998) that dividing a task into simpler tasks in reinforcement learning significantly improves learning efficiency. Based on these ideas, Dietterich used decomposition of the Markov decision process and developed a hierarchical approach to reinforcement learning (Dietterich, 2000). This divide and conquer approach requires evaluation of internal states of the machine and close supervision by a designer. In its extreme case of controlling each step, it will converge toward a supervised learning. Such a system is incapable of setting its own system of values.

A fundamental question that Pfeifer asked in his book (Pfeifer & Bongard, 2007) is what motivates an agent to do anything, and in particular, to enhance its own complexity. What drives an agent to explore the environment and learn ways to effectively interact with it? According to Pfeifer, an agent's motivation should emerge from the developmental process. He called this the "motivated complexity" principle. But isn't this like the chicken and egg problem? An agent must have a motivation to learn (and therefore to develop into a complex being), while at the same time, its motivation must emerge from this same development. Another idea for handling the motivation problem was presented by Steels (Steels, 2004), where he suggested equipping an agent with self-motivation that he calls the "autotelic principle". According to this principle the idea of "flow" experienced by some people when they perform their expert activity well would be used as motivation to accomplish even more complex tasks. However, no mechanism was proposed to identify "flow" in a machine or to implement the flow as a driving force for learning.

Many people in the embodied intelligence area ask (Steels, 2007) – where do we go now? In spite of many successes of embodied intelligence, fundamental problems of intelligence still remain unanswered. So it is quite surprising that the suggestion put forth by Pfeifer and Bongard (Pfeifer & Bongard, 2007) is to concentrate on advancements of robotic technology like building artificial skin or muscles. While this may be important for development of robots, it diverts attention from developing intelligence.

I hope that this discussion will help to bring focus back to the critical issues for understanding and developing intelligence. In the next few sections I will show how an agent may develop and maintain its system of values that controls its behavior. Such values are directly related to higher level goals and are only partially controlled by the environment. Higher level goals are established and their values learned by the machine. The machine is motivated to accomplish goals by the way it interacts with the environment.

3. Intelligence

In his seminal paper (Brooks, 1991b), Brooks pointed out that it does not matter what is intelligence and what is environmental interaction. Instead he stressed the utility of an agent's interaction with the environment and determined intelligence through the dynamics of this interaction. While this assumption helped to simplify the design of intelligent robots and justified a bottom-up approach to building intelligent machines, it also introduced a dangerous possibility of confusing a complex behavior with synonyms of intelligence. The question of intelligence is an important one if one wants to design an intelligent machine.

There is no universal agreement about how to define intelligence. However, there is a good understanding of what an intelligent agent (biological, mechanical or virtual) must be capable of. Scientists list such capabilities as abstract thinking, reasoning, planning, problem solving, intuition, creativity, consciousness, emotion, learning, comprehension, memory and motor skills as traits of intelligence. They use various tests and intelligence measures to compare levels of intelligence and differentiate between the intelligence of humans and nonhuman animals. In fact, passing various tests for (human level) intelligence was used as a substitute for its definition. Complex skills and behaviors were used to define how intelligence manifests itself. This skill based approach was inconsistent, because once a machine that was obviously not intelligent satisfied one test, another test was used in its place. This was a result of poor understanding of what is needed to create intelligence.

3.1 Definition of embodied intelligence

Existing definitions of intelligence focus on describing the properties of the mind rather than describing the mind itself. It is like defining a TV set not by how it is built and how it works but by what it does. Yet in order to design a mind we must agree on what we are designing. Perhaps driven by similar needs John Steward defined cognitive systems as follows (Stewart, 1993):

Definition: A system is cognitive if and only if sensory inputs serve to trigger actions in a specific way, so as to satisfy a viability constraint.

In a similar effort I propose an arbitrary and utilitarian definition of intelligence with the aim to present a set of principles and mechanisms from which necessary traits of intelligence can be derived. I hope that this definition is general enough to characterize agents of various levels of intelligence including human. To avoid a general discussion on intelligence I will utilize this definition to design embodied agents suggested by Brooks (Brooks, 1991b) and described in more detail by Pfeifer (Pfeifer, 1999).

Definition: Embodied intelligence (EI) is defined as a mechanism that learns how to survive in a hostile environment.

A mechanism in this definition applies to all forms of embodied intelligence, including biological, mechanical or virtual agents with fixed or variable embodiment, and fixed or variable sensors and actuators. Implied in this definition is that EI interacts with the environment and that the results of actions are perceived by its sensors. Also implied is that the environment is hostile to EI so that EI has to learn how to survive. This hostility of environment symbolizes all forms of pains that EI may suffer – whether it is an act of open hostility or simply scarcity of resources needed for the survival of EI. The important fact is that the hostility is persistent. For example, battery power level is a persistent threat for an agent requiring it. Gradually the energy level goes down, and unless the EI replenishes its energy, the perceived discomfort from its energy level sensor will increase.

Hostile stimulation that comes from the environment towards EI is necessary for it to acquire knowledge, develop environment related skills, build models of the environment and its embodiment, explore and learn successful actions, create its value system and goals, and grow in sophistication. Thus perpetual hostility of environment will be the foundation for learning, goal creation, planning, thinking, and problem solving. In advanced forms of EI it will also lead to intuition, consciousness, and emotions. Eventually all forms and levels of intelligence can be considered under the proposed definition of EI.

A critical element of the EI definition is learning. Thus an agent that knows how to survive in a hostile environment but cannot learn new skills is not intelligent. This will help to draw the line between developmental systems that learn from those that do not and perhaps will help to differentiate intelligent and non-intelligent animals. In this definition, purely reactive systems that do not learn are not intelligent, even if they exhibit complex behavior. A system must maintain its learning capability for us to continue calling it an intelligent system.

Notice that this definition of EI clearly differentiates knowledge from intelligence. While knowledge represents the acquired set of skills and information about the environment, intelligence requires the ability to acquire knowledge. Knowledge is a byproduct of learning, thus it is not necessary to include a pre-existing knowledge base in the machine memory. In turn, learning requires associative memories capable of storing spatio-temporal information acquired over various time scales. Learning to survive requires not only memory but its management, so that only the important memories are retained. Learning also requires the ability to associate the sensory and motor signals, so that action outcomes can be linked with causes.

3.2 Embodiment and intelligence

Intelligence cannot develop without an embodiment or interaction with the environment. Through embodiment, intelligent agents carry out motor actions and affect the environment. The response of the environment is registered through sensors implanted in the embodiment. At the same time the embodiment is a part of the environment that can be perceived, modelled and learned by intelligence. Properties of the motors and sensors, their status and limitations can also be studied, modelled and understood by intelligent agents.

The intelligence core interacts with the environment through its embodiment, as shown in Fig. 1. This interaction can be viewed as closed-loop sensory-motor coordination. The embodiment does not have to be constant nor physically attached to the rest of a body that contains the intelligence core (brain). The boundaries between embodiment and the environment change during the interaction which modifies the intelligent agents' self-determination. Because of the dynamically changing boundaries, the definition of embodiment contains elements of indetermination.

Definition: Embodiment of EI is a mechanism under the control of the intelligence core that contains sensors and actuators connected to the core through communication channels.

A first consequence of this definition is that the mechanism under control may change. When the embodiment changes, the way that the embodiment works and the intelligent agent interacts with the environment will be affected. Second, embodiment does not have to be permanently attached to the intelligence core in order to play its role of facilitating sensory-motor interaction with the rest of the environment. For instance, if we operate a

machine (drive a car, use a keyboard, play tennis), our embodiment dynamics can be learned and associated with our actions to an extent that reduces the distinction between the dynamics of our own body and the dynamics of our body operating in tandem with the machine. Likewise, artificially enhanced senses can be perceived and characterized as our own senses (e.g. glasses that improve our vision or a hearing aid that improves our hearing). Another example of sensory extension could be an electronic implant stimulating the brain of a blind person to provide visual information. Third, not all sensors and actuators have to probe and act on the environment external to the body. While those that do, allow the EI to interact with the external environment, internal sensors and actuators support the embodiment. When its body temperature rises, a machine may activate an internal cooling mechanism. When an animal is threatened, its heart beats faster in preparation for a fight or an escape. The body experiences internal pain that communicates a potential threat. Thus a flow of signals though embodiment is as shown in Fig. 1.

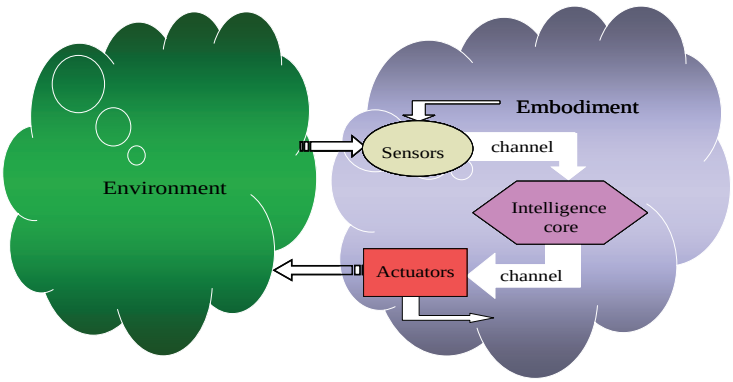


Fig. 1. Intelligence core with its embodiment and environment

Extended embodiment does not have to be of a physical (mechanical) nature. It could be in the form of remote control of tools in a distant surgery procedure or monitoring the Martian landscape through mobile cameras. It could also be our remote presence at the soccer game through received TV images or our voice message delivered through a speakerphone to a group of people at a teleconference.

An extended embodiment of intelligence also may come in the form of organizations and their internal working mechanisms and procedures. A general directing troops on a battle field feels a similar power of moving armies as a crane operator feels the mechanical power of the machine that he operates. In a similar way, a president feels the power of his address to his nation and the large impact it makes on people's lives.

This extended embodiment enhances EI's ability to interact with the environment and thus its ability to grow in complexity, skills and effectiveness. If the President learns how to address the nation, his abilities and skills to affect the environment grow differently than that of a woman in Darfur trying to save her child from violence.

Our knowledge of embodiment properties is a key to its proper use in interaction with the environment. We rely on this knowledge to plan our actions and predict the responses from

the environment. A change in the way our embodiment implements desired actions or perceives responses from the environment introduces uncertainty into our behavior and may lead to confusion and less than optimal decision making. If a car's controls were suddenly reversed during operation, a user would require some adaptation time to adjust to the new situation and might not be able to do it before crashing. Therefore, what we learn about our environment and our ability to change this environment is affected not only by our intelligence (ability to learn, understand, represent, analyze and plan) but by correct perception of our embodiment as well.

3.3 Designing an embodied intelligence

Learning is an active process. EI acquires information about its environment through sensors and interacts with it by sensory-motor coordination. The motor neurons fire in response to excitations according to desired actions associated with the perceived situation.
Learning which actions are desirable and which are not makes the learning agent more fit to survive in a hostile environment. There are several means of adapting to the environment that an agent can use to survive: evolutionary - by using the natural selection of those agents that are most fit, developing new motor skills like sweating in the hot weather or new sensors like cell sensitivity to light; and cognitive - by learning, using memory and associations, performing pattern recognition, representation building, and implementing goals. Here we address only the latter form of adaptation for the development of EI as the one we associate with an agent's intelligence. Another important form of intelligence - group intelligence - is left for future consideration, as it depends on the individual intelligence of the group members.
All spatio-temporal patterns that we experience during a lifetime underlie our knowledge, and lead to internal models of the environment. The patterns have features on various abstraction levels, and relations between these features are learned and remembered. Abstract representations are also built to represent motor actions and skills. The perceptual objects that a person can recognize, the relations among the objects, and the skills that he has are all represented in his memory. The memory is episodic and associative. It is distributed, redundant, and parallel, short term or long term. Various parts of memory are interconnected and interact in real time.
Another critical aspect of human brain development is self-organization. By self-organizing their interconnections, neurons quickly create representations of stored patterns, learn how to interact with the environment, and build expectations regarding future events. A six year old child has many redundant and plastic connections ready to learn almost anything. After years of learning, the connection density among neurons is reduced, as only the most useful information is retained, and related memories and skills are refined.
Although existing neural network models assume full or almost full connectivity among neurons, the human cerebral cortex is a sparsely connected network of neurons. For example, a neuron projecting through the mossy pathway (of a rat) from the dentate gyrus to subregion CA3 of the hippocampus has been estimated to synapse on 0.0078% of CA3 pyramidal cells (Rolls, 1989). Sparse connections can, at the same time, improve the storage capacity per synapse and reduce the energy consumption of a network.
For the purpose of building intelligent machines, it seems useful to develop a neural network memory that allows the machine to perceive and learn in a manner similar to that of humans. The memory should use a uniform, hierarchical, and sparsely-connected

structure with the capability to self-organize. EI with this type of memory will learn predominantly in an unsupervised manner by responding to stimuli from the environment. The learning process is deliberate, perpetual, and closely related to the machine's goals in the environment.

Having the general purpose of surviving and certain more specific goals, the machine can efficiently organize its resources to process the incoming information and learn the important skills. The creation of goals should result from the machine's interaction with its environment. Therefore, an intelligent machine must have a built-in mechanism to create goals for its behavior and such a mechanism will be called the goal creation system (GCS). The main role of GCS is to develop sensory-motor coordination, goal-oriented learning of perceptions and actions, and to act as stimuli for interaction with the environment. Like the machine's memory, GCS is based on a uniform hierarchical and self-organizing structure. The structure grows in complexity as goal hierarchy evolves. Meanwhile, the goal creation stimulates the growth of the hetero hierarchy representing sensory inputs and a similar hetero hierarchy representing actions and skills.

3.4 Pain signals as motivation

In embodied intelligence research a fundamental question is what motivates a machine to develop into an intelligent, knowledgeable being (Pfeifer & Bongard, 2007). It is an important question since a machine with intelligence is different from a robot that does only the tasks it was designed to do. An intelligent machine must be able to learn and execute various tasks, but the question is what makes it do any of them and in particular what motivates it to strive for excellence in executing these tasks?

To answer this question we may want to ask ourselves what motivates us to get up every morning and go to work. An attempt to formulate an answer to this question was suggested by Csikszentmihalyi (Csikszentmihalyi, 1996), who introduced "flow" theory which states that humans get internal reward for activities that are slightly above their level of development. Stimulated by the "flow" idea, Oudeyer et al. (Oudeyer, et al., 2007) developed an intrinsic motivation system for autonomous development in robots. A robot explores the environment and activates learning when its predictions do not match the observed environmental response. This leads to exploratory learning of the environment and basic sensory-motor coordination. The motivation in such systems comes from the desire to minimize the prediction error and is related to "artificial curiosity" presented by Schmidhuber (Schmidhuber, 1991). A variant of this type of learning was proposed by Barto et al. (Barto et al., 2004). Although artificial curiosity helps to explore the environment, it leads to learning without a specific purpose. It may be compared to the exploratory phase in reinforcement learning - internal reward motivates the machine to perform exploration. It is obvious that exploration is needed in order to learn and model the environment. But is this mechanism the only motivation we need to develop intelligence? Can "flow" ideas explain goal oriented learning? Can we find another more efficient mechanism for learning?

I suggest a goal-driven mechanism to motivate the machine to act, learn, and develop. I suggest that it is the hostility of the environment, as expressed in the definition of EI adopted here, that is the most effective motivational factor for learning. It is the pain we receive that moves us. And it is our intelligence determined to reduce this pain, which responds to the pain and motivates us to act, learn, and develop. The two conditions are

needed together - hostility of the environment and intelligence that learns how to "survive" by reducing the pain signal. Thus pain is good. Without pain there would be no intelligence, and without pain we would not be motivated to develop.
Thus in some strange step in the process of designing a foundation for intelligence, we come back to great philosophers like Plato who stated "if a pain is good, it is because it prevents a greater pain, or leads to a grater pleasure." (Moore, 1993). In philosophy, pain and pleasure are related to motivation for our own actions, as was eloquently stated by Robert Audi (Audi, 2001) - "There are general standards of rationality, including widely held standard of pleasure and pain as generating good prima facie reasons both for action and desire".
The same view that pain is good is shared by medical doctors (Yellon et al., 1996). Brand stated that pain is one of the ways that your body tries to tell you that something is wrong (Brand & Yancey, 1993). Pain can serve as a safety guard and action trigger; for example when exposed to a danger like fire or electric shock, if pain is felt, the body's immediate response is to pull away - a pain action trigger. Many people would die from the infection of a ruptured organ if they did not feel the pain! Leprosy patients lost their body parts not due to leprosy, but their inability to feel any pain at all. Pain also is a great tool for instruction, from the toddler learning to avoid the hot stove, to weight lifters working out and straining muscles, etc. In life, pain serves as a protector against danger or triggers a person to grow spiritually or intellectually after experiencing a cognitive pain.

4. Goal Creation for Embodied Intelligence

In human intelligence, the perception and the actions are intentional processes. They are built, learned and carried out attempting to meet certain goals or needs. Based on primitive needs, people first create simple goals and learn simple actions. Subsequently, by using the learned perception and skills, they build complex perceptions and actions to meet complex goals. It is postulated that this bottom-up process enables a human to find relevant subtasks for a complex task, dividing it into procedures that can be finished step-by-step. The process also generates human needs and these needs or expectations affect human attention to sensory information. In human learning, the rewards are more subjective than objective, and are given by the environment as well as being internally generated.
Pain, as a term for all types of discomforts and pressures, is a common experience to all people. On the most primitive level, people feel discomfort when they are hungry so that they learn to eat and to search for food. They feel pain when they touch burning charcoal so that they learn to stay away from extreme heat. Although, on more abstract levels, individuals experience different motives and higher-level goals, the primitive pains essentially help them to build this complex system of values in order to survive in the environment and to develop skills useful for successful operation.
Neurobiological study facilitated by neuro-imaging techniques, such as positron emission tomography (PET) and functional magnetic resonance imaging (fMRI) etc, supports the suggestion that there are multiple regions of the brain involved in the pain system which form the neomatrix, also called the "pain matrix" (Melzack, 1990). Experiments using fMRI have identified that such a matrix includes a number of cortical structures, the anterior insula, cingulate and dorsolateral prefrontal cortices (Peyron, et al., 2000), and subcortical structures including the amygdala (Derbyshire, et al., 1997) and the thalamus and hypothalamus (Hsieh, et al., 2001).

Two concurrent systems are identified in the pain matrix - the lateral pain system, which processes the physical pains, and the medial pain system, which processes the emotional aspects of pain, including fear, stress, dread and anxiety (Tölle, et al., 1999). The physically harmful stimuli activate neurons in the lateral pain system, and the anticipation of the pains can induce stress and anxiety, which activates the medial pain system. It has also been demonstrated experimentally that the anticipation of a painful stimulus can activate both pain systems (Porro, 2002).

It has been widely accepted for decades that pain has sensory-discriminative, affective, motivational, and evaluative components (Melzack, 1968). The work presented by (Mesulam, 1990) on a neurocognitive network model suggests that the cingulate cortex is the main contributor to a motivational network that interacts with a perceptual network in the posterior parietal cortex. In this work it is proposed that the pain network is responsible for the goal creation process and affects motivation, attention and sensory perception.

In the proposed learning paradigm, the EI machine will use neuronal structures to self-organize the proposed goal creation system (GCS). GCS stimulates the creation of goals on various abstraction levels, starting from the given primitive goals. It is responsible for evaluating actions in relation to EI goals, stimulating learning of useful associations and representations for sensory inputs and motor outputs. It finds the ontology among sensory objects, associates actions and input stimuli, creates needs and affects the agent's attention.

Accordingly, instead of computing a global value system by a typical reinforcement learning (RL) of the embodied machine, the value system is essentially embedded in the hierarchical GCS. In a classical actor-critic RL paradigm, the action is chosen by the action network based on the present sensory (state) input. The critic network evaluates the state-action pair to determine how the action network may improve the selection of actions. However, learning values of state-action pairs in RL is a long and slowly converging process.

Using the GCS, the machine's learning through interaction with its environment becomes an active process since the machine finds the optimum actions according to its internal goals and pain signals. The machine uses internal reinforcement signals, which make learning of state-action pairs' values more efficient. Since internal rewards depend on accomplishing goals set internally by the machine, learning is organized without reinforcement input from the teacher. Once the machine learns how to accomplish lower level goals, it develops a need for sensory inputs required to perform a beneficial action, and this need is used to define higher level goals. Thus the EI agent evaluates and chooses its actions through an integrated system of goals and values that have only loose relations to the primitive goals and external rewards.

In the following sections the concept and structures for the goal creation system will be further developed.

4.1 Goal Creation System

The built-in goal creation and value system triggers learning of intentional representations and associations between the sensory and motor pathways. When the EI machine realizes that a specific action resulted in a desirable effect related to a current goal, it stores the representation of the perceived object involved in such action and learns associations between the representations in the sensory pathway and the active action neurons in the motor pathway. If the produced results are not relevant to the current goal, no intentional learning takes place. Since this usually happens during the exploration stage, such a deliberate learning process protects the machine's memory from overloading with less

important information. This is not to say that a machine cannot learn during the exploratory phase. However, learning in this phase is less intensive and can be based on finding novelty in perceived environment response to EI actions.
Neurons in the goal creation pathway form a hierarchy of pain centers. They receive the pain signals and trigger creation of goals, which represent the needs of the machine and the means to solve its pains. Lower level pains and associated goals are externally stimulated through primitive sensory inputs. Neurons' activation on these inputs may represent a large number of situations that the EI encounters while interacting with the environment. Higher level pains and goals are developed through associations between neuron activities in the sensory-motor pathways that reduce lower level pains. Goals on the lower levels correspond to simple, externally driven objectives, while those on the higher levels correspond to complex objectives that are learned through the machine's actions and are related to finding the best ways to accomplish the lower level goals.

4.2 Fundamental Characteristics of the Goal Creation System

In the proposed goal creation system for intelligent machines, the advancement of EI value and action systems is stimulated by a simple built-in mechanism rooted in dedicated sensory inputs, called "primitive pain". Since the pain signal comes from the hostile environment (including the embodiment of the EI machine), it is inevitable and gradually increases unless the machine figures out how to reduce and avoid it. Pain reduction is desirable while pain increase is not. Thus, the agent has a desire to reduce the pain or equivalently to pursue pleasure/comfort. EI is forced by the "primitive pain" to explore the environment seeking solutions to achieve its goal - reduction of the pain. In this process, the machine will accumulate knowledge about the environment and its own embodiment, and will develop its cognitive skills.
The EI machine may have several primitive pains, and each one of them has its own varying intensity, and requires a distinct solution. At any given time, the machine suffers from the combination of different pains with different intensities. Pains vary over time and the agent needs to set reduction of the strongest pain as its current goal.
We can make references to human learning systems where a similar mechanism is used to induce activity-based exploration and learning. The "primitive pain" inputs for a human include pain, hunger, urge, fear, stress, anxiety, and other types of physical discomfort. The pain usually happens when something is missing. For instance, we feel hungry when we lack the sufficient sugar level in our blood. We feel anxious when we lack enough food or money. We feel fear when we have no protection, etc... This postulation of deficiency in satisfying our goals as a trigger for action and learning makes the proposed goal creation mechanism biologically plausible even at the level of human intelligence. For example, in a newborn baby, a hierarchical goal creation system and value system has not yet been developed. If the baby is exposed to a primitive pain and it suffers, it will not be satisfied until some action can result in the pain reduction. When the pain is reduced, the baby learns to represent objects and actions that helped to lower that pain.
We also need to find and eat food to sustain our activities. A gradually increasing discomfort coming from the low "sugar level" tells us that we must eat. The pain gets stronger and forces us to search for solutions. Similar urges pressure us to go to the bathroom, put on clothes when we feel cold, or not touch a burning coal. The pain warns us against incoming threats, but also forces us to take an action. We feel relief if we take an

action that reduces this pain. Thus pleasure and comfort can be perceived as a reduction of pain and discomfort.

The intensities of the perceived pains prioritize our actions and are responsible for goal creation. For example, the urgent need to go to the bathroom may easily overtake our desire to eat, or even more so to sit through an interesting lecture. In general, the strongest pains will determine the most pressing goals. Thus the pain-based GCS yields a natural goal management scheme.

A primitive pain leads the machine to find a solution and then the solution is set as the primitive goal. Afterwards, the primitive pain will also trigger development of higher level pain centers and create higher level goals. This is based on a fundamental mechanism for the need to act in response to pain and a simple measure for satisfying such a need. I would argue that this simple need to act motivates machine development and may lead to creation of complex goals and means of their implementation. The mechanism of goal creation in a human, and specifically how the human brain controls human behaviors, is not yet fully established in the field of behavioral science or psychology. It is quite likely that the proposed mechanism is different from the way people create their goals. However, it is feasible, simple, and it satisfies our need to establish the goal creation and to formulate the emergence of a goal hierarchy for machine learning. In addition, this goal creation system stimulates the machine to interact with its environment and to develop its skills.

4.3 Basic Unit of GCS

The proposed goal creation mechanism is based on evolving uniform, basic goal creation units. A GCS unit contains three groups of neurons that interact with each other, including the pain center neurons, reinforcement neuro-transmitter neurons and the corresponding connected neurons in the sensory and motor pathways. The basic goal creation unit (GCU) structure is shown in Fig.2. Although as demonstrated in (Starzyk, et al al., 2008), representations of sensory objects or motor actions are best built using distributed groups of neurons in sensory and motor pathways, they are illustrated here as a single neuron for simplicity.

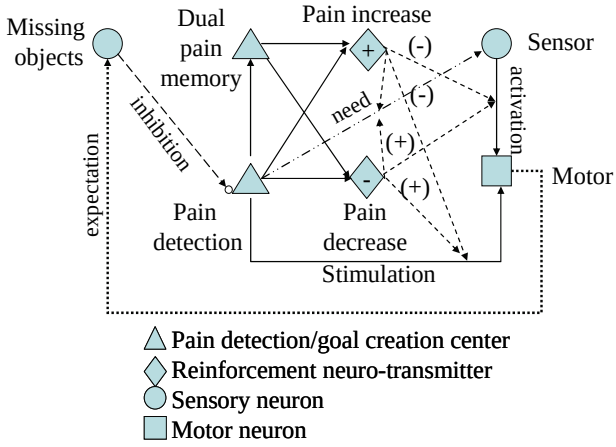


Fig. 2. Basic goal creation unit

The pain detection center is stimulated by the pain signal I_P and represents the negative stimulus, such as pain, discomfort, or displeasure. Since the pain exists due to the absence of certain objects, denoted as "missing objects" in Fig. 2, the perceived object can inhibit the pain signal through the "inhibition" link. Thus the pain detection center is activated by the silence of this sensory neuron. A dual pain memory center stores the delayed pain level, I_{Pd}. The currently detected pain signal and the previous pain signal (in the previously completed event) are compared in the second group, which contains reinforcement neuro-transmitter neurons.

Neuro-transmitter neurons register a decrease or increase in the pain level by comparing signals from the pain detection center and the dual pain memory center. They do not physically connect to any neuron but send positive or negative reinforcement signals to build the associations. The "pain decrease" neuro-transmitter neuron gives a positive reinforcement while the "pain increase" neuron gives a negative reinforcement. The reinforcement signal is calculated in (1).

$$r = I_P - I_{Pd} \tag{1}$$

The third group of neurons in Fig. 2 contains the corresponding active neurons in the sensory and motor pathways that these pain center neurons connect to.

In a GCU, initially, the pain detection center directly stimulates multiple motor neurons. A gradually increasing pain level forces the machine to explore various motor actions by stimulating the motor neurons through initially random connection weights W_{MP}. The machine explores starting from the action with the strongest activation (strongest weights connecting to given pain stimuli). To carry out such action, certain objects which will be involved in this action must be available. Initially, a motor neuron may be associated with multiple sensory neurons by activation weights W_{MS}. The available (active) sensory neurons send activations to the motor neurons so that a certain sensory-motor action pair can be implemented. The direct links from the pain center to the motor neurons force exploration or the implementation of certain motor actions as long as the selected pain persists.

After the action is taken, once the pain reduction or increase is detected by the second group of neurons, a learning signal r is produced to reinforce or weaken the value of an action and the value of the sensory-action pair by strengthening or weakening the stimulation links from the pain detection center to the motor neurons and the activation links from the sensory to the corresponding motor neurons. Pain increase will make the links more inhibitory, while pain decrease will make links more excitatory, as shown in (2)

$$\begin{aligned} W_{MP} &= W_{MP} + r \cdot \beta^n \\ W_{MS} &= W_{MS} + r \cdot \beta^n \end{aligned} \tag{2}$$

where β denotes a smaller than 1 learning rate and n denotes how many times the link has been adjusted.

Meanwhile, since the active sensory neuron representing the object which was involved in the action helps reduce the pain, a "need" link, with weight W_{SP} will be created to connect the active pain detection center to the active sensory neuron using Hebbian Learning. On

the other hand, the object, which was missing and produced the abstract pain signal, becomes available and the neuron representing the object becomes active after the motor action; an "expectation" link with weight W_{SM} will be created to connect the motor neuron and the missing object.
The "need" link and the "expectation" link will be updated by reinforcement learning. The stronger the change in the pain level is, the stronger are the reinforcement signal and the weight adjustment on the involved links. The described interaction of various groups of neurons in the goal creation mechanism and the "stimulation", "activation", "expectation" and "need" links are illustrated in Fig. 2.
This simple mechanism is easy to expand and generalize. In order to generate abstract and complex goals, we will incorporate basic goal creation units into a hierarchy of the goal creation pathway as discussed next.

4.4 Building Goal Hierarchy

A primitive pain is a signal received from the primitive pain sensors. It stimulates the primitive pain detection center. In solving the pain on the primitive level, the machine is stimulated to explore for actions or to exploit the action that relieves the primitive pain. The exploration at first is based on the random stimulation and activation links, or links that were initially (genetically) set to help reduce the primitive pains. Such genetically set links facilitate learning of higher level skills and correspond to built-in skills. Genetic setting of lower level skills may be useful in designing machines that need to develop complex skills.
Genetically set associations between the primitive pain centers and actions also exist in animals. We have genetically wired sensory-motor circuits to sustain basic bodily functions like heart beating, breathing, digestion, etc. A baby cries when it is wet or hungry; it also has well developed skills to eat.
A burning pain from touching a hot plate triggers an automatic pull back reflex. These sensations and actions become gradually associated with the circumstances under which they occurred, leading an intelligent agent to learn basic skills or improve upon them.
To solve the primitive pain from low sugar level, after several random trials, the action "eat", connected with perception of "food", will be rewarded. As a result, the strength of the stimulation link from primitive pain detection center to "eat" and the activation link from "food" to "eat" will be increased. The "need" link is connected from the pain center to "food" and reinforced when such successful action is exploited and rewarded several times. In addition, the "eat" action will trigger expectation of sufficient "sugar level" on the sensory input. Thus, whenever the "low sugar level" pain center sends out pain signals, the "eat" will be excited, prompting the machine for this action.
Since "food" is needed for solving this primitive pain, its absence will lead to anxiety or stress for the machine. A second level pain center representing such stress is created and is called an abstract pain center. An abstract pain center is not stimulated from a physical pain sensor; it only symbolizes a real pain or represents the discomfort of not having the object that can prevent the primitive pain.
When "food" is available and the agent "eats", the primitive pain is relieved. The pain signal disappears and the agent goes back to its normal painless state. As a result, an inhibitory link is developed between the sensory signal representing presence of "food" and the abstract pain center, which means that the existence of the "food" can inhibit the abstract pain.

When "food" is not available, the agent cannot reduce the physical primitive pain. Then, he tries to find a solution to reduce the "abstract pain". Although reduction of the abstract pain does not directly reduce the primitive pain on the lower level, it may be a prerequisite for such reduction.

As specified in the example of a primitive pain center, the agent is forced to explore to reduce the abstract pain. Again, exploration is done based on the initial associations between the abstract pain center and motor actions, and associations between sensory representations and motor actions. The reinforcement neuro-transmitters connected with this abstract pain center update the interconnection weights. Eventually, the reduction in the abstract pain resulting from the action "open" combined with sensory object "refrigerator" indicates that the pain from absence of "food" will be associated with the sensory-motor pair "refrigerator"-"open". It does not matter whether such action (opening refrigerator) was found by pure exploration or by instruction from a teacher. Since once the machine opens the refrigerator, it sees the food and the abstract pain is suppressed, the action will be reinforced. In addition, an expectation link from the motor action "open" to the sensory neuron "food" is built; thus "food" will be expected as the result of the action "open". This expectation link will be used for planning future actions in which a certain action's result can be expected. This process can be illustrated using Fig. 3.

This goal hierarchy can be further expanded vertically. If the agent "opens" the "refrigerator", but the "food" is not found, the machine needs other options to suppress the abstract pain, and subsequently the primitive pain. It may explore the environment by random search, or use instruction. Once it "spends" some "money" (in a store), food is available and then the abstract pain (no food) is reduced. Such action is rewarded and in the future will be more strongly stimulated by the abstract pain center. The "food" is eaten, the primitive pain is suppressed, and the pain signals are reduced. However, when "money" is not available, an abstract pain center on level III is activated with an inhibitory link from "money". Subsequently, the machine needs to learn how to solve the abstract pain on level III related to lack of "money".

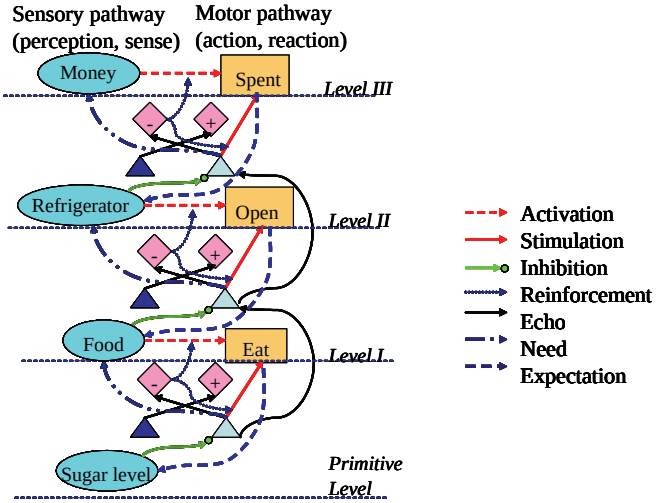


Fig. 3. Creating the abstract pain signals

In the proposed GCS, at every step, the machine finds an action that satisfies its goals, and this action and the involved representations may result in creating further goals. Therefore, in this mechanism, the machine simultaneously learns to associate the goals with deliberate actions, the expected results of actions, the means to represent and obtain objects, and relations among objects. It learns which objects are related to its goals. This helps to establish higher level goals and the means of their implementation. The machine governs execution of actions to satisfy its goals and manages the goal priorities at any given time.
We need both – artificial curiosity and goal creation - to create intelligent systems. We need the first one to explore, and the second one to learn efficiently with a purpose. They complement each other in motivated systems as exploration and exploitation complement each other in reinforcement learning. However, unlike in reinforcement learning, they develop a complex structure of internal goals and rewards that makes learning more efficient, and gives the machine freedom to decide how to approach a given problem. This brings the machine's self-organizing structures a step forward towards general intelligence.
In the proposed model EI uses a hierarchical self-organizing learning memory (HSOLM) for representation building, goal creation, and learning. HSOLM is organized as a hetero hierarchical system of sparsely connected processing units (neurons and their connections). Neurons on different levels of hierarchy handle different tasks. Lower-level neurons are either activated by the sensory neurons and extracted features, or activate motor neurons that control the machine's behaviour. Subsequent level neurons combine the extracted features and represent elements of more complex entities, goals and skills. Information is gathered, associated and abstracted (in an invariant form) as it flows up the memory hierarchy. Top-level neurons represent perceived entities, ideas, goals, skills, and actions.
HSOLM uses three basic pathways – a sensory pathway responsible for perception, a motor pathway responsible for actions, and a goal creation pathway responsible for goal creation, evaluation of actions in relation to goals, learning of useful associations and motivation to act and learn. These three pathways interact on various abstraction levels.

5. The Goal Creation Experiment

The purpose of this experiment is to compare the effectiveness of learning based on the goal creation (GC) system, and reinforcement learning (RL). In order to have a fair comparison, a similar initial neural network topology is used in both learning methods. Networks have similar complexity expressed by the number of input, output and hidden neurons; they also have a similar number of links between neurons and a similar depth of the network. The difference is in how the weight adjustment takes place. While in the RL system, weights are adjusted based only on the external reward/punishment signal, in the GC system, learning is triggered by the local pain centers associated with sensory inputs. In this experiment we assume that both sensory inputs and motor outputs are symbolic, which means that they represent a source (sensory) or action triggered by this output (motor).
The goal creation system built in this experiment is a simplified implementation of the embodied intelligence idea. Thus the system interacts with the environment and is informed about the quality of its actions by an external pain signal. This pain signal is a foundation for setting internal abstract pains and goals to remove these pains.

5.1 Network Organization

Let's assume that the network has to learn how not to go hungry in an environment in which there are limited food resources and advanced skills are required to get them. The machine has six sensory inputs that represent: toys, food, grocery store, bank, office and school. In addition, six motor outputs represent: play, eat, buy, withdraw (money), work, and study. The current state of the environment is determined by the availability of the resources.

A single primitive pain input r(t) that represents hunger is the sugar level in the blood. This primitive pain automatically increases if the machine does not eat for some time. Initially there are plenty of resources around, indicated by a high probability of firing of all the sensory inputs. This allows the machine to learn by exploration how to satisfy its primitive goal (reduce hunger). However, as the machine uses them, the original resources are gradually depleted and need to be replaced. Thus the machine needs to learn how to do so. It will use the GC system to set higher level goals and learn how to accomplish them. Desired sensory-motor pairs and their effect on the environment are listed in the following Table 1. All other sensory-motor pairs are either undesirable or have no effect on the machine's perceived success.

PAIR #	SENSORY	MOTOR	INCREASES	DECREASES
1	Food	Eat	sugar level	food supplies
8	Grocery	Buy	food supplies	money at hand
15	Bank	Withdraw	money at hand	spending limits
22	Office	Work	spending limits	job opportunities
29	School	Study	job opportunities	-

Table 1. Meaningful sensory-motor pairs and their effect on the environment

At each time step the primitive pain increases by 0.1 from its previous level unless it is reset by eating the food. A sensory input probability of various items of category c_i (availability of resource) is described by the function:

$$f_{ci}(k_{ci}) = 1 / [\, 1+(k_c/\tau_c)] \qquad (3)$$

where: τ_c is a scaling factor (resource decline rate), k_c stands for the number of the times a resource was used (initially set to zero). In this experiment τ_c=10. Each time a specific action is taken by the machine (for instance the machine withdraws money from bank account) the corresponding k_c is increased by 1 decreasing the likelihood of this particular resource. This resource can be renewed by invoking a higher level goal after which k_c at a particular level is reset to 0. For instance if grocery was bought, the counter for food was rest to zero and at the same time the counter for money increased by 1.

First, the goal creation experiment is implemented using the reinforcement learning (RL) scheme. The machine must not only learn proper actions, but must adjust its actions to changing environment. Then the same environment is used to test the goal creation system.

5.2 Reinforcement Learning Results

The algorithm of RL used is Actor-critic architecture as shown in Fig. 4. In this scheme, the optimal mapping from states to actions is to be learned during interaction with the external environment based on the primary reinforcement.

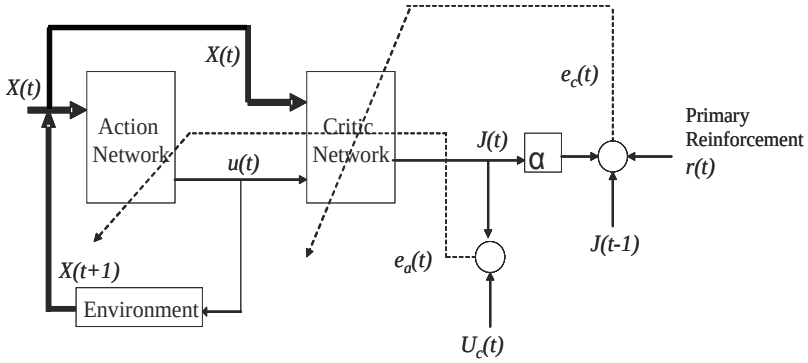


Fig. 4. Actor-critic architecture for RL

The actor-critic architecture contains two components: the action network and the critic network, and both networks are implemented by a multi-layer perceptron (MLP) in this experiment. The action network determines the action u(t) based on current states X(t). The critic network evaluates the state-action value J according to {X(t), u(t)}. The J(X(t), u(t)), also denoted as J(t) value is defined as

$$J(X(t),u(t)) = R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}, \tag{4}$$

where $\gamma, 0 \le \gamma \le 1$, is the discount rate.

The critic network is trained using the temporal difference method (Sutton & Barto, 1998). The critic network directs the action network to produce better action so that J(t) is maximized; thus the action network is trained. A more detailed implementation of this algorithm can be found in (Si & Wang, 2001).

Since the agent has no prior knowledge, the reasonable associations among different objects and actions are not yet built; therefore all possible object- action combinations are considered. The algorithm is described as follows.

Reinforcement learning using AC method:

Step 1). The agent receives input information X(t) and the reward signal r(t) from the environment. X(t) shows the availability of the resources as a binary vector, with "1" indicating being available. The reward signal r(t) is related to the state-action pair taken.

Step 2). The action network (AN) determines the action u(t) from 36 possible action-object combinations based on current input vector X(t). The u(t) is in the form of a binary

vector as well, with "1" indicating the selected action and "0" for all other actions.
Step 3). The critic network (CN) determines the value of this state-action pair J(t).
Step 4). Using the reward signal r(t), CN is trained by TD method. The error function of CN is show in (5).

$$\begin{aligned} e_c(t) &= \gamma J(t) - [J(t-1) - r(t)] \\ E_c(t) &= \frac{1}{2} e_c{}^2(t) \end{aligned} \tag{5}$$

Step 5). The weights in CN are updated according to gradient descent,

$$\begin{aligned} w_c(t+1) &= w_c(t) + \Delta w_c(t) \\ \Delta w_c^{(2)}(t) &= l_c(t)\left[-\frac{\partial E_c(t)}{\partial w_c^{(2)}(t)}\right] \\ \Delta w_c^{(1)}(t) &= l_c(t)\left[-\frac{\partial E_c(t)}{\partial w_c^{(1)}(t)}\right] \end{aligned} \tag{6}$$

where $l_c(t)$ is the learning rate of CN.
Step 6). The CN reevaluates the value of J(t).
Step 7). The AN is trained in order to produce action u(t) which has a desired value

$$\begin{aligned} e_a(t) &= J(t) - U_c(t) \\ E_a(t) &= \frac{1}{2} e_a{}^2(t) \end{aligned} \tag{7}$$

where $U_c(t)$ is the desired value. The updating algorithm is similar to that in CN and is based on a gradient descent rule,

$$\begin{aligned} \Delta w_a(t) &= l_a(t)\left[-\frac{\partial E_a(t)}{\partial w_a(t)}\right] \\ \frac{\partial E_a(t)}{\partial w_a(t)} &= \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial w_a(t)} \end{aligned} \tag{8}$$

where $l_a(t)$ is the learning rate of AN.
Step 8). Apply the determined action u(t) to the environment, and the environment will give new states X(t+1). Repeat steps 1) to 8).

In a simulated trial, the agent runs for 600 time steps. The pain input (hunger) during this trial is shown in Fig.5.

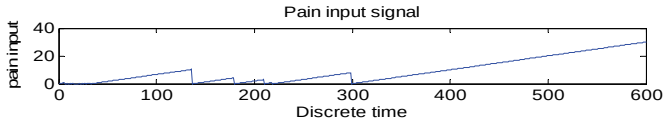


Fig. 5. Results from RL experiment

Initially the RL agent learns how to control the input pain by performing the "eat food" action, but as the food resource is depleted the agent needs to learn another action ("buy food"). The agent learns this action in 103 iterations, and restores the food supply resetting the food counter to 0. After that, it can find food more easily and perform the "eat food" action that reduces the input pain. It repeats the "buy food" action again in iterations number 309 and 407. However, after iteration 299 it never returns to the "eat food" action even though the food is available and the primitive pain increases.

The overall performance by the RL scheme on this experiment can be evaluated from multiple trials. As is evident from the simulation results, it takes a long time for the RL mechanism to adjust to the changing environment and to learn efficient actions. As the system learns, the environment changes requiring higher skills. If the system is not capable of learning these higher level skills in the limited time, it may fail to perform well, since the environment conditions will change and the model initially learned will no longer be satisfactory. In several runs, the changes in the environment happened too quickly for the RL system to keep up with them. In addition to the large number of iterations required in RL, each iteration was more time consuming than the corresponding iteration in the GCS.

5.3 Goal Creation System Results

A goal creation system can be implemented in a general learning scheme similar to the AC method, as shown in Fig 6. In addition to the blocks used in the AC method, it contains a pain network that affects the operation of the critic network, produces abstract pains, and internal reward based on detected inhibition of the abstract pains. These pain detection, goal creation and learning mechanisms were previously illustrated in Fig. 2. The pain network receives sensory input from the environment, including the primitive pain signals r'(t).

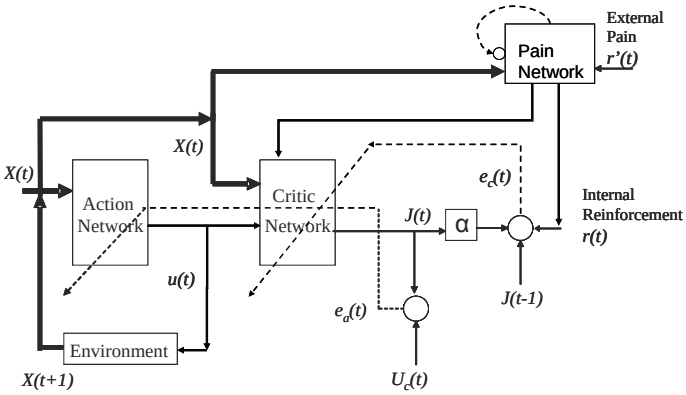


Fig. 6. Goal creation system architecture.

As soon as the machine learns how to satisfy a lower level pain, it identifies the environmental condition that helps it to remove the pain (for instance by supplying food), and creates an abstract pain that activates when these conditions are not met (for instance no food is present). These abstract pain centers have their own bias signals that indicate the importance of this pain compared to other pains. Each time an abstract pain is reduced, it increases the bias weight of the pain center associated with the sensory input that was required to reduce this abstract pain.

A very small decrease is applied simultaneously to all of the bias weights. This small reduction of biases allows the machine to forgo some abstract goals, in the case in which either the environment changed in a significant way and the machine needs to adapt, or the machine learned more effective ways to interact with the environment and replaced former less effective goals with new ones. The machine uses its goal creation approach to learn what to do, and how to adjust to changing environmental conditions. It does so by adjusting pain biases and weights between the pain signals and the critic network.

A typical result of CGS simulation is shown in Fig. 7. This figure shows dynamic changes in the pain signals (including the primitive pain) over several hundred iterations. At first the only pain that the machine responds to is the primitive pain. Once the machine learns that eating food reduces the primitive pain, the lack of food (as observed in the sensory input) becomes an abstract pain. As there is less and less food in the environment, the primitive pain increases again (since the machine cannot get food) and the machine must learn how to get food (buy at the grocery). Once it learns this, a new pain source is created and so on. Notice that the primitive pain is kept under control in spite of changing environmental conditions. On an average run, the machine can learn to develop and solve all abstract pains in this experiment within 200-300 iterations. In all the experiments, the pain threshold was set to 0.1. Below this threshold level no pain is detected.

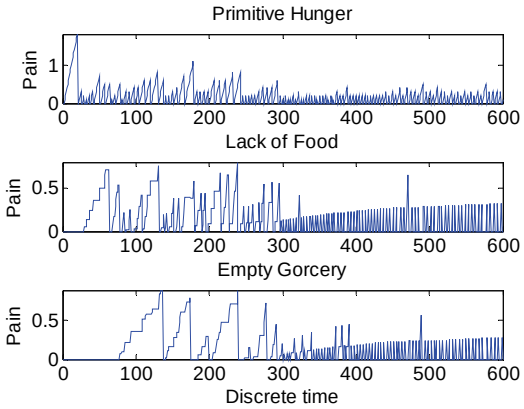


Fig. 7. Pain signals in CGS simulation

Fig. 8 shows a scatter plot that illustrates the selection of a specific action (specified by the sensory-motor pair) in 5 different runs of the GCS. As seen in the attached figure, the machine learns to select useful actions while exploring the environment and selecting the most useful actions repeatedly after it learns a proper interaction scheme.

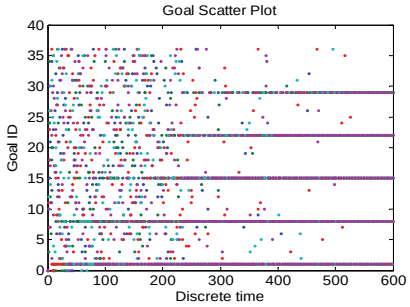


Fig. 8. Action scatters in 5 CGS simulations

The average of all abstract pain signals obtained on the basis of 100 such experiments is shown in Fig. 9. As can be observed, the machine learns to contain all abstract pains and maintain the primitive pain signal at a low level in demanding environmental conditions.

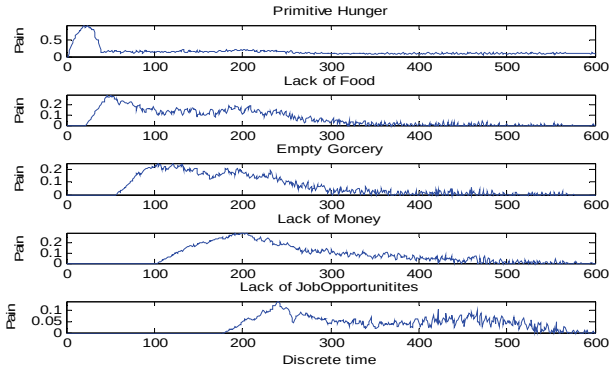


Fig. 9. The average pain signals in 100 CGS simulations

7. Conclusions

This chapter presented a goal creation system that motivates embodied intelligence to learn how to efficiently interact with the environment. The system uses artificial curiosity to explore, and the creation of abstract goals to learn efficiently and purposefully. It develops higher level abstract goals and increases the internal complexity of representations and skills that it stores in its memory. It was demonstrated that this type of system learns better and faster than traditional reinforcement learning systems.

In a striking contrast to classical reinforcement learning, where the reinforcement signals come from the outside environment, GCS generates an internal reward associated with the abstract goal that the machine was able to accomplish. This makes the reinforcement process not observable, and to some degree makes the machine less controllable than one whose operation is based on classical reinforcement learning. The machine's actions are more difficult to understand and explain by an external observer, thus the machine behaves

more like an independent intelligent being rather than a well trained robot.
Instead of the computational-based value system used in typical reinforcement learning, the value system is essentially embedded in the GCS. Initially, the agent acts on very primitive goals and learns the sub-goals or higher level goals through interaction with its environment. At every step, the agent finds an action that satisfies its goals and this action may result in creating further goals. Gradually, the agent learns values of various states for implementing goals. It also learns to associate the primitive goals with its internal states and learns to create higher level goals. At more advanced levels, the agent will be able to understand external instructions and use them as its primitive goals to find the intermediate subgoals or a sequence of abstract goals.
In this scheme, learning that occurs on a higher level of the memory structure to satisfy a higher level goal may have no direct relationship with actions that may satisfy the lower level goals. For instance, an abstract goal of earning money might have been stipulated by a need to buy food or by a need to pay the heating bill or both. In a similar way, accomplishing this higher level goal does not necessarily remove hunger or heat the living space. The only externally observable "primitive pain" may remain unaffected.
This differentiates an externally administered reward related to a "primitive pain" and a reward that satisfies a higher level goal. While we can measure (and thus optimize) the total external reward received to satisfy the primitive goals, it is impossible to measure (and therefore to optimize) the total amount of reward received for higher level goals, as these goals remain mostly unknown to the external observer. This fact differentiates the proposed GCS from the RL scheme, in which the total amount of reward is assumed to be measurable, so that a system can be optimized to maximize the reward. Legg & Hutter demonstrated (Legg & Hutter, 2006) that a mathematically optimal intelligent agent can be built by optimizing the total amount of the rewards that a RL receives. Since the total amount of reward in the proposed GCS cannot be measured from outside, the mathematical proof of optimality presented in (Legg & Hutter, 2006) does not apply. An important question can be formulated as follows – is it an internal reward system and motivation to act, that differentiates an intelligent being from an unintelligent one? I claim that this is true.
In summary, the presented goal creation system motivates a machine to act and develop its cognitive skills in response to externally applied pain signals. It also helps the machine to perceive its environment, learn with a purpose, and respond to changes in environment.

8. References

Agre, P. E. & Chapman, D. (1990). What are plans for?, *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, Pattie Maes (ed), pp.17-34, MIT Press, Cambridge, MA.

Audi, R. (2001). *The Architecture of Reason: The Structure and Substance of Rationality*, Oxford University Press, ISBN 0-195-14112-1.

Barto, A. Singh, S. & Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills, *Proc. 3rd Int. Conf. Development Learn*., pp. 112–119, San Diego, CA.

Brand, P. & Yancey, P. (1993). *The Gift of Pain: Why We Hurt and What We Can Do About It.* Zondervan Publishing House, ISBN: 0310221447.

Brooks, R.A., (1991a). Intelligence without representation, *Artificial Intelligence 47*, pp. 139–159.

Brooks, R.A. (1991b). Intelligence without reason, *Proc. 12th Int. Joint Conf. on Artificial Intelligence*, pp. 569-595, Sydney, Australia.

Csikszentmihalyi, M. (1996). *Creativity-Flow and the Psychology of Discovery and Invention*. New York: Harper Perennial, ISBN-10: 0060928204.

Currie, K.W. & Tate, A. (1991). O-Plan: the Open Planning Architecture. *Artificial Intelligence*, 52(1). pp. 49-86, ISSN: 0004-3702.

Dayan, P. & Hinton, G.E. (1993). Feudal Reinforcement Learning, In: *Advances in Neural Information Processing Systems 5*. S. J. Hanson, J. D. Cowan and C. L. Giles (Eds.), Morgan Kaufmann: San Mateo, CA.

Derbyshire, S.W. G., Jones, A. K. P, Gyulai, F., et al. (1997). Pain processing during three levels of noxious stimulation produces differential patterns of central activity, *Pain*, vol. 73, pp. 431–445, ISSN 0304-3959, Elsevier, Amsterdam.

Dietterich, T.G. (2000). Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition, *Journal of Artificial Intelligence Research*, 13, 227-303.

Fu, W.-T. & Anderson, J. R. (2006). Solving the Credit Assignment Problem: Explicit and Implicit Learning with Internal and External State Information. *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: LEA.

Hsieh, J.C., Tu, C. H., Chen, F. P. et al. (2001). Activation of the hypothalamus characterizes the acupuncture stimulation at the analgesic point in human: a positron emission tomography study, *Neurosci Lett.* vol. 307, pp. 105–108.

Legg, S. & Hutter, M. (2006). A Formal Measure of Machine Intelligence," *Proc. 15th Annual Machine Learning Conference of Belgium and The Netherlands,* pp.73-80.

Maes P. (1990). Situated agents can have goals, pp. 49-70, *Robotics and Autonomous Systems, 6*.

Maturana, H.R. & Varela, F.J. (1980). Autopoiesis and Cognition - The Realization of the Living: *Boston Studies in the Philosophy of Science* , vol. 42.

McCarthy, J.; Minsky, M. L. & Shannon, C.E. (1955). *A proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. Harvard University, N. Rochester, Aug. 31.

Melzack, R. & Casey, K. L. (1968). Sensory, motivational, and central control determinants of pain, In: Kenshalo DR (Ed). *The Skin Senses.* Pp. 423–439, Springfield: C.C. Thomas.

Melzack, R. (1990). Phantom limbs and the concept of a neuromatrix," *Trends Neurosci*, vol. 13, pp. 88-92.

Mesulam, M.M. (1990). Large-scale neurocognitive networks and distributed processing for attention, language, and memory, *Ann Neurol,* vol. 28, pp. 597–613.

Moore, G.E. (1993), *Principia Ethica*. Cambridge: Cambridge University Press, 1903; Revised edition with "Preface to the second edition" and other papers, ed. T. Baldwin, Cambridge: Cambridge University Press.

Moravec, H.P. (1984). Locomotion, vision and intelligence, In: *Robotics Research 1*, M. Brady and R. Paul, eds., MIT Press, Cambridge, MA, pp. 215-224.

Oudeyer, P-Y., Kaplan, F., Hafner, V. (2007). Intrinsic Motivation Systems for Autonomous Mental Development, *IEEE Transactions on Evolutionary Computation*, 11(2).

Parr, R. & Russell S. (1998). Reinforcement Learning with Hierarchies of Machines, In *Advances in Neural Information Processing Systems 10*, MIT Press.

Peyron, R.; Laurent, B. & Garcia-Larrea, L. (2000). Functional imaging of brain responses to pain. A review and meta-analysis," *Neurophysiol Clin.* vol. 30, pp.263–288.

Pfeifer, R. & Scheier, C. (1999). *Understanding Intelligence*, MIT Press, Cambridge, MA, 1999.

Pfeifer, R. & Bongard J.C. (2007). *How the Body Shapes the Way We Think: A New View of Intelligence,* The MIT Press (Bradford Books), 2007.

Porro, C.A., Baraldi, P., Pagnoni, G., et al. (2002). Does anticipation of pain affect cortical nociceptive systems? *J. Neurosci,* vol. 22. pp. 3206–3214.

Rolls, E.T. (1989). Functions of neuronal networks in the hippocampus and neocortex in memory, In: *Neural models of plasticity: experimental and theoretical approaches*, pp. 240–265, San Diego: Academic Press, 1989.

Schmidhuber, J. (1991). Curious model-building control systems, *Proceedings Int. Joint Conf. Neural Networks*, pp. 1458–1463, Singapore, vol. 2.

Si, J. & Wang, Y. (2001). On-Line Learning Control by Association and Reinforcement, *IEEE Trans. Neural Networks*, Vol. 12, No. 2, pp. 264-276.

Singh, S.P. (1992). Transfer of learning by composing solutions for elemental sequential tasks. *Machine Learning,* 8, pp. 323-340.

Starzyk, J. A.; Liu, Y. & Vogel, D., (2008). Sparse coding in a hierarchical self-organizing memory with sparse connectivity, submitted to *IEEE Trans. Neural Networks*.

Steels, L. (2003) Intelligence with Representation. *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 361, no. 1811, pp. 2381- 2395.

Steels, L. (2004) The Autotelic Principle, in Embodied Artificial Intelligence, *Lecture Notes in AI*, vol. 3139, pp. 231-242.

Steels, L. (2007) The symbol grounding problem is solved, so what's next? In *Symbols, embodiment and meaning*, De Vega, M. and G. Glennberg and G. Graesser, editors, Academic Press, New Haven.

Stewart, J. (1993). Cognition without neurons: Adaptation, learning and memory in the immune system. CC-AL 11, pp. 7-30.

Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning.* PhD thesis, University of Massachusetts, Amherst, MA.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An introduction*. Cambridge, MA: MIT Press.

Tölle, T. R.; Kaufmann, T.; Siessmeier, T. et al. (1999). Region-specific encoding of sensory and affective components of pain in the human brain: a positron emission tomography correlation analysis, *Ann. Neurol.*, vol. 45, pp. 40–47.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 59(236): 433-460.

Varela, F. J.; Thompson, E. T. & Rosch, E. (1992). *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge, MA: The MIT Press.

Walter W.G. (1951). A Machine That Learns, *Scientific American,* 185(2), pp. 60-63.

Walter W.G. (1961). *The Living Brain,* Duckworth, London, 1953, republished by Penguin, Harmondsworth, UK.

Wehner R. (1987). Matched Filters - Neural Models of the External World, *J. Comp. Physiol. A*, vol. 161, pp. 511-531.

Yellon, D.M.; Baxter G.F. & Marber M.S. (1996). Angina reassessed; pain or protector? *Lancet;* vol. 347: pp. 1159–62.

7

Robot Control by Fuzzy Logic

Viorel Stoian, Mircea Ivanescu
University of Craiova
Romania

1. Introduction

Fuzzy set theory, originally developed by Lotfi Zadeh in the 1960's, has become a popular tool for control applications in recent years (Zadeh, 1965).
Fuzzy control has been used extensively in applications such as servomotor and process control. One of its main benefits is that it can incorporate a human being's expert knowledge about how to control a system, without that a person need to have a mathematical description of the problem.
Many robots in the literature have used fuzzy logic (Song & Tay, 1992), (Khatib, 1986), (Yan et al., 1994) etc. Computer simulations by Ishikawa feature a mobile robot that navigates using a planned path and fuzzy logic. Fuzzy logic is used to keep the robot on the path, except when the danger of collision arises. In this case, a fuzzy controller for obstacle avoidance takes over.
Konolige, et al. use fuzzy control in conjunction with modeling and planning techniques to provide reactive guidance of their robot. Sonar is used by robot to construct a cellular map of its environment.
Sugeno developed a fuzzy control system for a model car capable of driving inside a fenced-in track. Ultrasonic sensors mounted on a pivoting frame measured the car's orientation and distance to the fences. Fuzzy rules were used to guide the car parallel to the fence and turn corners (Sugeno et al., 1989).
The most known fuzzy models in the literature are Mamdani fuzzy model and Takagi-Sugeno-Kang (TSK) fuzzy model. The control strategy based on Mamdani model has the linguistic expression (Mamdani, 1981):

Rule k: IF condition C1 AND condition C2 ..... $\Leftarrow$ Fuzzy sets
THEN decision D_k $\Leftarrow$ Fuzzy sets

The TSK models are formed by logical rules that have a fuzzy antecedent part and functional consequent (Sugeno, 1985):

Rule i: IF x_1 is C_{1i} AND x_2 is C_{2i} AND ..... $\Leftarrow$ Fuzzy sets
THEN $u_i = f_i(x_1, x_2, ... , x_n)$ $\Leftarrow$ Non fuzzy sets

where C_{ij}, j = (1, p), i = (1, n) are linguistic labels defined as reference fuzzy sets over the imput spaces (X_1, X_2, ...), x_1, x_2, ... are the values of imput variables and u_i is the crisp output inferred by the fuzzy model as a nonlinear functional.
The advantage of the TSK model lies in the possibility to decompose a complex sistem into simpler subsystems. The TSK model allows to use a fuzzy decomposition and an interpolative reasoning mechanism. In some cases this method can use a decomposition in linear subsystems.

2. Robot control system by fuzzy logic

2.1 Control methodology

Consider the conventional control system of a robot (Fig. 2. 1) which is based on the control of the error by using standard controllers like PI, PID.

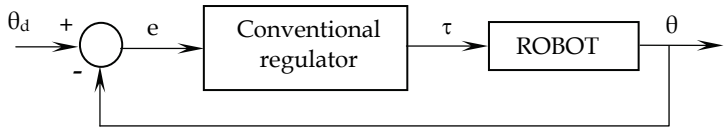


Fig. 2. 1. Conventional control system

$$e(t) = \theta_d(t) - \theta(t) \tag{2.1}$$

The control strategy determines the torque of the robot arm so that the steady error converges to zero

$$e_s = \lim_{t \to \infty} e(t) = 0 \tag{2.2}$$

We can conclude that in the classical approach, the basic decisions imply the use of simple feedback control loops, loop interactions, internal feedbacks by cascade controllers and multimode controllers.
The basic idea of Fuzzy Logic Control (FLC) centre on the labelling process in which the reading of a sensor is translated into a label as performed by human expert controllers (Yan et al., 1994), (Van der Rhee, 1990), (Gupta et al., 1979). The general structure of a fuzzy logic control is presented in Fig. 2. 2.

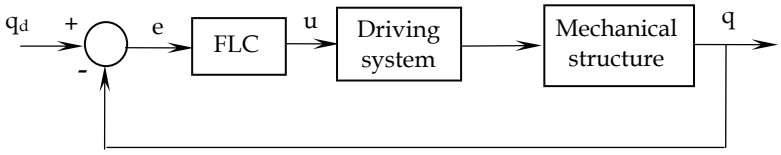


Fig. 2. 2. General structure of a fuzzy logic control

The main component is represented by the Fuzzy Logic Controller (FLC) that generates the control law by a knowledge-based system consisting of IF … THEN rules with vague predicates and a fuzzy logic inference mechanism (Jager & Filev, 1994), (Yan et al., 1994), (Gupta et al., 1979), (Dubois & Prade, 1979). A FLC will implement a control law as an error function in order to secure the desired performances of the system. It contains three main components: the fuzzifier, the inference system and the defuzzifier.

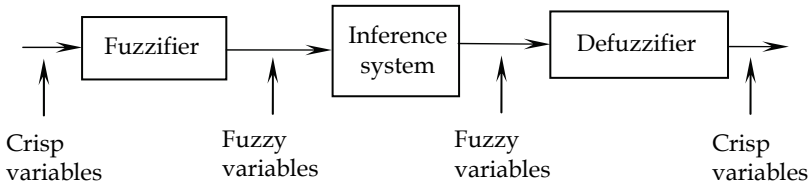


Fig. 2. 3. The structure of the fuzzy logic control

The fuzzifier has the role to convert the measurements of the error into fuzzy data.
In the inference system, linguistic and physical variables are defined. For the each physical variable, the universe of discourse, the set of linguistic variables, the membership functions and parameters are specified. One option giving more resolution to the current value of the physical variable is to normalize the universe of discourse. The rules express the relation between linguistic variables and derive from human experience-based relations, generalization of algorithmic non fully satisfactory control laws, training and learning (Gupta et al., 1979), (Dubois & Prade, 1979). The typical rules are the state evaluation rules where one or more antecedent facts imply a consequent fact.
Defuzzifier combines the reasoning process conclusions into a final control action. Different models may be applied, such as: the most significant value of the greatest membership function, the computation of the averaging the membership function peak values or the weighted average of all the concluded membership functions.
The FLC generates a control law in a general form:

$$u(k) = F(e(k), e(k-1), \dots e(k-p), u(k-1), u(k-2), \dots, u(k-p)) \tag{2.3}$$

Technical constraints limit the dimension of vectors. Also, the typical FLC uses the error change

$$\Delta e(k) = e(k) - e(k-1) \tag{2.4}$$

and for the control

$$\Delta u(k) = u(k) - u(k-1) \tag{2.5}$$

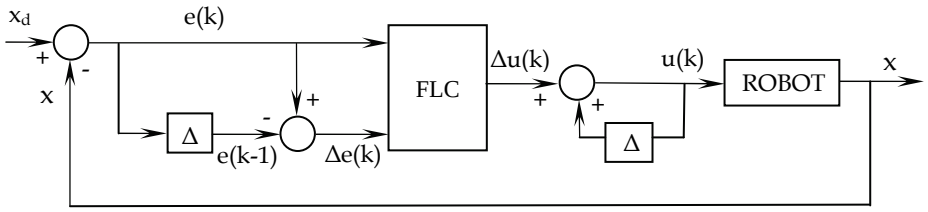


Fig. 2. 4. The structure of the robot control by fuzzy logic

Such a control law can be written as (2.6) and (2.7) (Gupta et al., 1979), (Dubois & Prade, 1979) and it is represented in Fig. 2. 4.

$$\Delta u(k) = F(e(k), \Delta e(k)) \tag{2.6}$$

$$u(k) = u(k-1) + \Delta u(k) \tag{2.7}$$

The error e(k) and its change Δe(k) define the inputs included in the antecedents of the rules and the change of the control Δu(k) represents the output included in the consequents.

The methodology which will be applied for the control system of the robot arm is:
- Convert from numeric data to linguistic data by fuzzification techniques
- Form a knowledge-based system composed by a data base and a knowledge-base.
- Calculate the firing levels of the rules for crisp inputs.
- Generate the membership function of the output fuzzy set for the rule base.
- Calculate the crisp output by defuzzification

2.2 Control System

Consider the dynamic model of the arm defined by the equation

$$\dot{x} = f(x) + b(x)u \tag{2.8}$$

where x represents the state variable, a (n x 1) vector, and u is control variable. The desired state of the motion is defined as:

$$x_d = \left[x_d, \dot{x}_d, \ldots, d_d^{(n-1)}\right]^T \tag{2.9}$$

and the error will be

$$e^* = \left[x - x_d, \dot{x} - \dot{x}_d, \ldots, x^{(n)} - x_d^{(n)}\right]^T \tag{2.10}$$

consider the surface given by the relation

$$s = e^* + \sigma \dot{e}^* \tag{2.11}$$

where

$$\sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots \sigma_n) \tag{2.12}$$

is a diagonal positive definite matrix. The surface

$$S(x) = 0 \tag{2.13}$$

defines the switching surface of the system. For n = 1, the switching surface becomes a switching line (Fig. 2.5)

$$s = e + \sigma \dot{e} \tag{2.14}$$

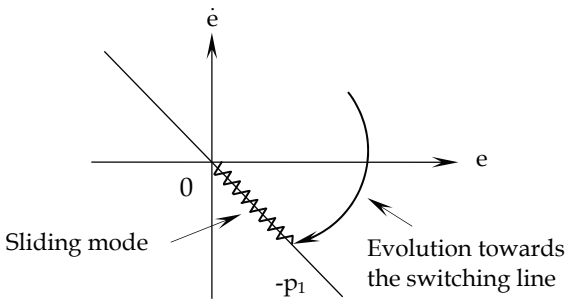


Fig. 2.5. Trajectory in a variable structure control

The control strategy is given by (Dubois & Prade, 1979).

$$u = -k\,\mathrm{sgn}(s) \tag{2.15}$$

Assuming a simplified form of the equation (2.8) as

$$m\ddot{x} + k\dot{x} = u \tag{2.16}$$

from (2.14) one obtains

$$\ddot{e} = \dot{s} - \sigma \dot{e} \tag{2.17}$$

For a desired position $x_d, \dot{x}_d, \ddot{x}_d$ this relation can be written as

$$\dot{s} = -\frac{k}{m}s + H - \frac{1}{m}u \tag{2.18}$$

where

$$H(e, x_d, \dot{e}, \dot{x}_d, \ddot{x}_d) = \sigma e + \sigma\frac{k}{m}e + \ddot{x}_d - \frac{k}{m}\dot{x}_d \tag{2.19}$$

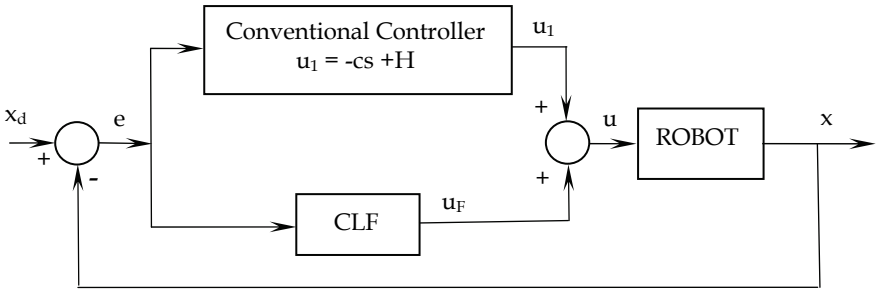


Fig. 2.6. Control system of the robot

We shall consider the control law of the form

$$u = -cs + m(H + u_F) \tag{2.20}$$

where c is a positive constant, c > 0, the second component mH compensates the terms determined by the error and desired position (2.19) and the last component is given by a FLC (Fig. 2.6). The stability analysis of the control system is discussed following Lyapunov's direct method. The Lyapunov function is selected as

$$V = \frac{1}{2}s^2 \tag{2.21}$$

hence

$$\dot{V} = s\dot{s} \tag{2.22}$$

and, from the relation (2.18) one has

$$\dot{V} = \frac{s^2}{m}(-k + c) + su_F \tag{2.23}$$

Thus, the dynamic system (2.16), (2.20) is globally asymptotical stable if

$$\dot{V} < 0 \tag{2.24}$$

One finds that

$$c < k \tag{2.25}$$

$$u_F = -\alpha \operatorname{sgn} s \tag{2.26}$$

The last relation (2.26) determines the control law of FLC. Consider the membership functions for e, $\dot{e}$ and u represented in Fig. 2.7 and Fig. 2.8 where the linguistic labels NB, NM, Z, PM, PB denote: NEGATIVE BIG, NEGATIVE MEDIUM, ZERO, POSITIVE MEDIUM and POSITIVE BIG, respectively.

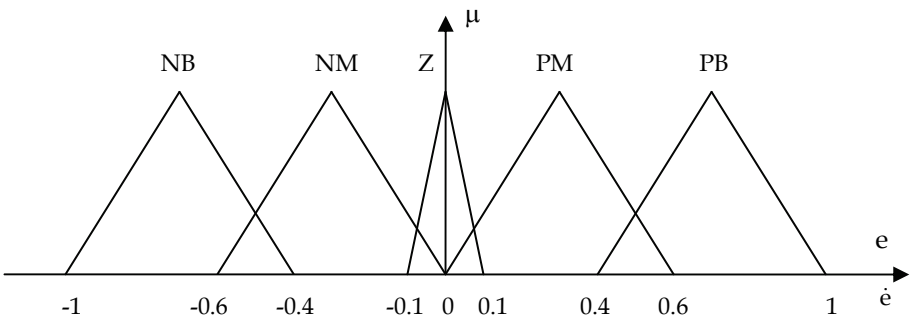


Fig. 2.7. Membership functions for e and $\dot{e}$

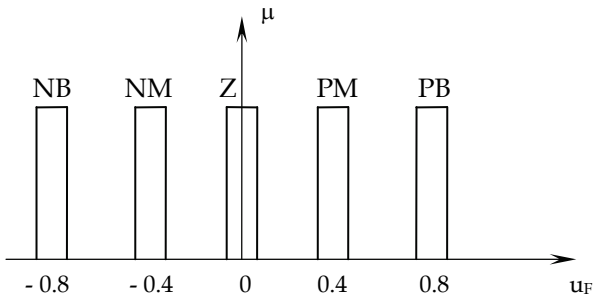


Fig. 2.8. Membership functions for u_F.

The rule base, represented in Table 2.1 is obtained from the relation (2.26).

ė \ e	NB	NM	Z	PM	PB
PB	Z	NM	NM	NB	NB
PM	PM	Z	NM	NM	NB
Z	PM	PM	Z	NM	NM
NM	PB	PM	PM	Z	NM
NB	PB	PB	PM	PM	Z

Table 2.1. Rule base for u_F

The rule base for u_F is the following:

Rule 1: IF e is NB AND ė is PB
THEN u_F is Z
Rule 2: IF e is NB AND ė is PM
THEN u_F is PM
.....
Rule 25: IF e is NB AND ė is PB
THEN u_F is Z

3. Mobile robot control system based on artificial potential field method and fuzzy logic

3.1 Artificial potential field approach

Potential field was originally developed as on-line collision avoidance approach, applicable when the robot does not have a prior model of the obstacles, but senses them during motion execution (Khatib, 1986). Using a prior model of the workspace, it can be turned into a systematic motion planning approach. Potential field methods are often referred to as "local methods". This comes from the fact that most potential functions are defined in such a way that their values at any configuration do not depend on the distribution and shapes of the obstacles beyond some limited neighborhood around the configuration. The potential functions are based upon the following general idea: the robot should be attracted toward its goal configuration, while being repulsed by the obstacles. Let us consider the following dynamic linear system with can derive from a simplified model of the mobile robot:

$$\dot{x} = A\,x + B\,F \tag{3.1}$$

where $\mathbf{x} = \left[x_1, \ldots x_n, \dot{x}_1, \ldots \dot{x}_n\right]^T \in R^{2n}$ is the state variable vector

$\mathbf{F = u} \in \mathbf{R}^{2n}$ is the input vector

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}_{nxn} & \mathbf{I}_{nxn} \\ \mathbf{0}_{nxn} & \mathbf{0}_{nxn} \end{bmatrix}; \mathbf{B} = \begin{bmatrix} \mathbf{0}_{nxn} \\ \mathbf{I}_{nxn} \end{bmatrix} \tag{3.2}$$

$\mathbf{0}_{n\,x\,n} \in \mathbf{R}^{n\,x\,n}$ is the zero matrix

$\mathbf{I}_{n\,x\,n} \in \mathbf{R}^{n\,x\,n}$ is the unit matrix

We can stabilize the system (3.1) toward the equilibrium point $[x_1 \,..\, x_n]^T = [x_{T1} \ldots y_{Tn}]^T$ by using the artificial potential field (artificial potential $\prod$ which generates artificial force system **F**).

$$\mathbf{F}(t) = \frac{\partial W_P(\mathbf{x})}{\partial \mathbf{x}} - \mathbf{F}_d - \frac{\partial \Pi(\mathbf{x})}{\partial \mathbf{x}} \tag{3.3}$$

where the first term compensates the gravitational potential, the second term assures the damping control and the last component defines the new artificial potential introduced in order to assure the motion to the desired position.

$$\frac{\partial \Pi(\mathbf{x})}{\partial \mathbf{x}} = \left[\frac{\partial \Pi(\mathbf{x})}{\partial x_1}, \frac{\partial \Pi(\mathbf{x})}{\partial x_2}, \ldots \frac{\partial \Pi(\mathbf{x})}{\partial x_n} \right]^T \tag{3.4}$$

In order to make the robot be attracted toward its goal configuration, while being repulsed from the obstacles, $\prod$ is constructed as the sum of two elementary potential functions:

$$\Pi(\mathbf{x}) = \Pi_A(\mathbf{x}) + \Pi_R(\mathbf{x}) \tag{3.5}$$

where: $\Pi_A(\mathbf{x})$ is the *attractor potential* and it is associated with the goal coordinates and it isn't dependent of the obstacle regions.
$\Pi_R(\mathbf{x})$ is the *repulsive potential* and it is associated with the obstacle regions and it isn't dependent of the goal coordinates.

In this case, the force **F**(t) is a sum of two components: the *attractive force* and the *repulsive force*:

$$\mathbf{F}(t) = \mathbf{F}_A(t) + \mathbf{F}_R(t) \tag{3.6}$$

3.2 Attractor potential artificial field

The artificial potential is a potential function whose points of minimum are attractors for a controlled system. It was shown (Takegaki & Arimoto, 1981), (Douskaia, 1998), (Masoud & Masoud, 2000), (Tsugi et al., 2002) that the control of robot motion to a desired point is possible if the function has a minimum in the desired point. The attractor potential $\prod_A$ can be defined as a functional of position coordinates **x** in this mode:

$$\Pi_A : \Omega \rightarrow \mathbf{R};\ \Omega = \mathbf{R}^n \tag{3.7}$$

$$\Pi_A(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{n} \left[k_i (x_i - x_{Ti})^2 + k_{n+i} \dot{x}_i^2 \right] \Sigma = \frac{1}{2} \mathbf{x}^T \mathbf{K} \mathbf{x} \tag{3.8}$$

where

$$\mathbf{K} = \text{diag}\,(k_1, k_2, \ldots, k_{2n}), \quad k_i > 0\ (i = 1, \ldots, 2n) \tag{3.9}$$

The function $\prod_A(\mathbf{x})$ is positive or null and attains its minimum at $\mathbf{x}_T$, where $\prod_A(\mathbf{x}_T) = 0$. $\prod_A(\mathbf{x})$ defined in this mode has good stabilizing characteristics (Khatib, 1986), since it generates a force $\mathbf{F}_A$ that converges linearly toward 0 when the robot coordinates get closer the goal coordinates:

$$\mathbf{F}_A(\mathbf{x}) = k(\mathbf{x} - \mathbf{x}_T) \tag{3.10}$$

Asymptotic stabilization of the robot can be achieved by adding dissipative forces proportional to the velocity $\dot{\mathbf{x}}$.

3.3 Repulsive potential artificial field

The main idea underlying the definition of the repulsive potential is to create a potential barrier around the obstacle region that cannot be traversed by the robot trajectory. In addition, it is usually desirable that the repulsive potential not affect the motion of the robot when it is sufficiently far away from obstacles. One way to achieve these constraints is to define the repulsive potential function as follows (Latombe, 1991):

$$\Pi_R(\mathbf{x}) = \begin{cases} \frac{1}{2}k\left(\frac{1}{d(\mathbf{x})} - \frac{1}{d_0}\right)^2 & \text{if } d(\mathbf{x}) \le d_0 \\ 0 & \text{if } d(\mathbf{x}) > d_0 \end{cases} \tag{3.11}$$

where k is a positive coefficient, $d(\mathbf{x})$ denotes the distance from $\mathbf{x}$ to obstacle and d_0 is a positive constant called *distance of influence* of the obstacle. In this case $\mathbf{F}_R(\mathbf{x})$ becomes:

$$\mathbf{F}_R(\mathbf{x}) = \begin{cases} k\left(\frac{1}{d(\mathbf{x})} - \frac{1}{d_0}\right)\frac{1}{d^2(\mathbf{x})}\frac{\partial d(\mathbf{x})}{\partial \mathbf{x}} & \text{if } d(\mathbf{x}) \le d_0 \\ 0 & \text{if } d(\mathbf{x}) > d_0 \end{cases} \tag{3.12}$$

For those cases when the obstacle region isn't a convex surface we can decompose this region in a number (N) of convex surfaces (possibly overlapping) with one repulsive potential associated with each component obtaining N repulsive potentials and N repulsive forces. The repulsive force is the sum of the repulsive forces created by each potential associated with a sub-region.

3.4 Dynamic model of the system

The mobile robot is represented as a point in configuration space or as a particle under the influence of an artificial potential field $\prod$ whose local variations are expected to reflect the

"structure" of the space. Usually, the Lagrange method is used to determinate the dynamic model:

$$\frac{d}{dt}\left(\frac{\partial L(\mathbf{q},\ \dot{\mathbf{q}})}{\partial\ \dot{\mathbf{q}}}\right) - \frac{\partial L(\mathbf{q},\ \dot{\mathbf{q}})}{\partial\ \mathbf{q}} = \mathbf{F} \tag{3.13}$$

or

$$\frac{d}{dt}\left(\frac{\partial W_C(\mathbf{q},\ \dot{\mathbf{q}})}{\partial\ \dot{\mathbf{q}}}\right) - \frac{\partial W_C(\mathbf{q},\ \dot{\mathbf{q}})}{\partial\ \mathbf{q}} + \frac{\partial W_P(\mathbf{q})}{\partial\ \mathbf{q}} = \mathbf{F} \tag{3.14}$$

where:

$$\begin{aligned} &\mathbf{L} = W_C - W_P \text{ is Lagrange function} \\ &W_C - \text{total kinetic energy} \\ &W_P - \text{total potential energy} \\ &\mathbf{q} = [\,x\ y]^T - \text{coordinate vector} \\ &\mathbf{F} = [F_X\ F_Y]^T - \text{force vector} \end{aligned} \tag{3.15}$$

The dynamics of the mobile robot becomes:

$$m\ddot{x} + \mu mg - k_f \dot{x} = F_X \tag{3.16}$$

$$m\ddot{y} + \mu mg - k_f \dot{y} = F_Y \tag{3.17}$$

The artificial potential forces which are the control forces are:

$$F_X = -k_1\dot{x} - \frac{\partial \Pi}{\partial x} \tag{3.18}$$

$$F_Y = -k_1\dot{y} - \frac{\partial \Pi}{\partial y} \tag{3.19}$$

The dynamical model of the system is:

$$m\ddot{x} + \mu mg - k_f \dot{x} = -k_1\dot{x} - \frac{\partial \Pi}{\partial x} \tag{3.20}$$

$$m\ddot{y} + \mu mg - k_f \dot{y} = -k_1\dot{y} - \frac{\partial \Pi}{\partial y} \tag{3.21}$$

where:

$$\Pi = \Pi_A + \Pi_R \tag{3.22}$$

The potential function is typically (but not necessarily) defined over free space as the sum of an *attractive* potential pulling the robot toward the goal configuration and a *repulsive* potential pushing the robot away from the obstacles.

3.5 Fuzzy controller

We denote by $\mathbf{x} = [x, y]^T$ the trajectory coordinates of the mobile robot in XOY plane and let be the error between the desired position and mobile robot position.

$$\mathbf{e} = \mathbf{x}_T - \mathbf{x} \tag{3.23}$$

The switching line σ in the real error plan is defined as

$$\sigma(\dot{e}, e) = \dot{e} + me \tag{3.24}$$

A possible trajectory in the $(\dot{e}, e)$ plane is presented in Fig. 3.1.

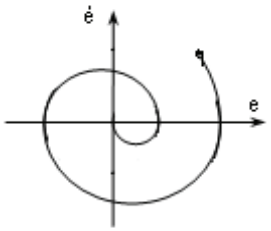


Fig. 3.1. System evolution

We can consider that the final point is attained when the origin O is reached. A great control procedure, DSMC (Ivanescu, 1996) can be obtained if the trajectory toward the moving target has the form as in Fig. 3. 2.

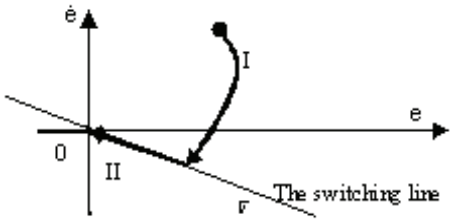


Fig. 3. 2. DSMC procedure

When trajectory in the ($\dot{e}$, e) plane penetrates the switching line, the motion is forced toward the origin, directly on the switching line. The condition which ensure this motion are given in (Ivanescu, 2001). The fuzzy logic controller used here has two inputs and one output. The displacement and speed data are obtained from sensors mounted on the mobile robot. The displacement error and velocity error are taken as the two inputs while the control force is considered to be the output. For all the inputs and the output the range of operation is considered to be from -1 to +1 (normalized values). The fuzzy sets used for the three variables are presented in Fig. 3. 3.
The linguistic control rules are written using the relation (3.24) and Fig. 3.2 and are presented in Table 3.1.

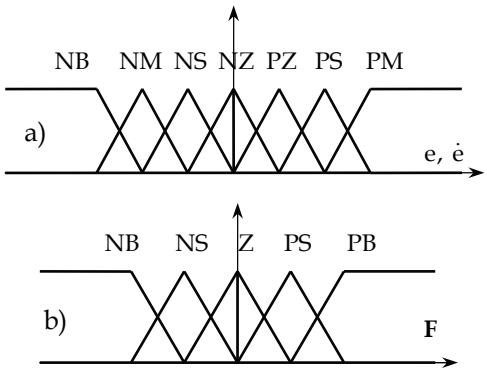


Fig. 3. 3. The fuzzy sets for the inputs and the output variables

ė / e	NB	NM	NS	NZ	PZ	PS	PM	PB
PB	Z	NS	NS	NB	NB	NB	NB	NB
PM	PS	Z	NS	NS	NB	NB	NB	NB
PS	PS	PS	Z	NS	NS	NB	NB	NB
PZ	PB	PS	PS	Z	NS	NS	NB	NB
NZ	PB	PB	PS	PS	Z	NS	NS	NB
NS	PB	PB	PB	PS	PS	Z	NS	NS
NM	PB	PB	PB	PB	PS	PS	Z	S
NB	PB	PB	PB	PB	PB	PS	PS	Z

Table 3.1. The linguistic control rules

3.6 Simulations

We propose the mobile robot to move from initial point (x, y) = (0, 0) to final point (x_T, y_T) = (7, 5). First, we consider that aren't any obstacles in moving area and the mobile robot is driven toward goal point by attractor artificial potential field (Fig.3.4).

$$\Pi(x) = \Pi_A(x) = \frac{1}{2}\left[(x-7)^2 + (y-5)^2\right] \tag{3.25}$$

$$\Pi_R(\mathbf{x}) = \begin{cases} \frac{1}{2}\left(\frac{1}{\sqrt{(x-4)^2+(y-3)^2}} - 1\right)^2 & \text{if } \sqrt{(x-4)^2+(y-3)^2} \le 1 \\ 0 & \text{if } \sqrt{(x-4)^2+(y-3)^2} > 1 \end{cases} \quad (3.26)$$

Second, we consider that there is a dot obstacle, in $(x_R, y_R) = (4, 3)$, with *distance of influence* $d_0 = 0.4$. The expression for repulsive potential is (3.26). The trajectory is shown in Fig. 3.5.

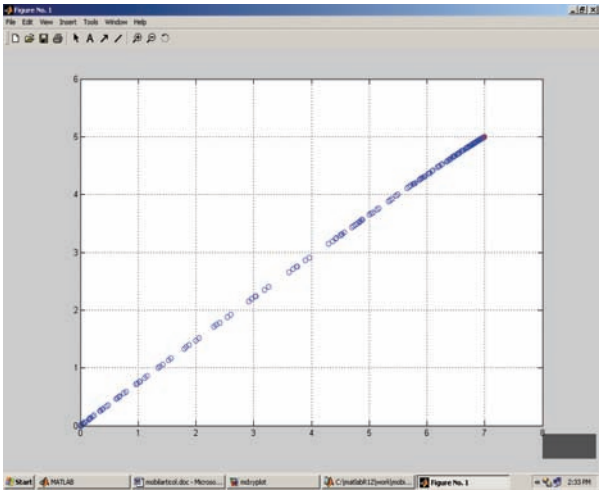


Fig. 3. 4. The robot trajectory without obstacles

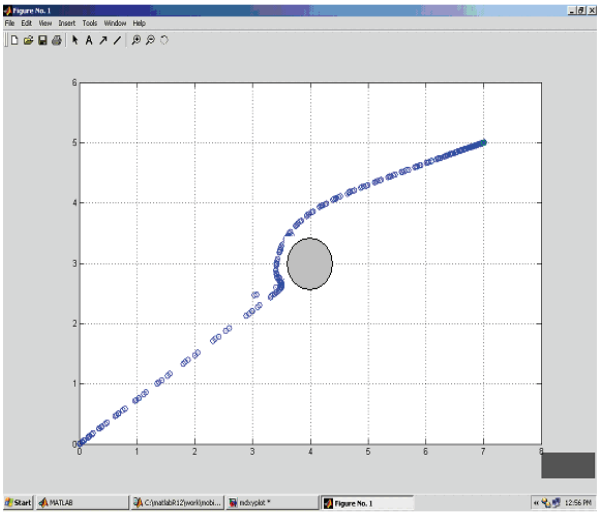


Fig. 3.5. The constrained robot trajectory by one obstacle

4. Fuzzy logic algorithm for mobile robot control next to obstacle boundaries

4.1 Control algorithm

In this section a new fuzzy control algorithm for mobile robots is presented. The robots are moving next to the obstacle boundaries, avoiding the collisions with them.
The mobile robot is equipped with a sensorial system to measure the distance between the robot and object that permits to detect 5 proximity levels (PL): PL1, PL2, PL3, PL4, and PL5. Fig. 4.1a presents the obstacle (object) boundary and the five proximity levels and Fig. 4.1b presents the two degrees of freedom of the locomotion system of the mobile robot. This can move either on the two rectangular directions or on the diagonals (if the two degrees of freedom work instantaneous).

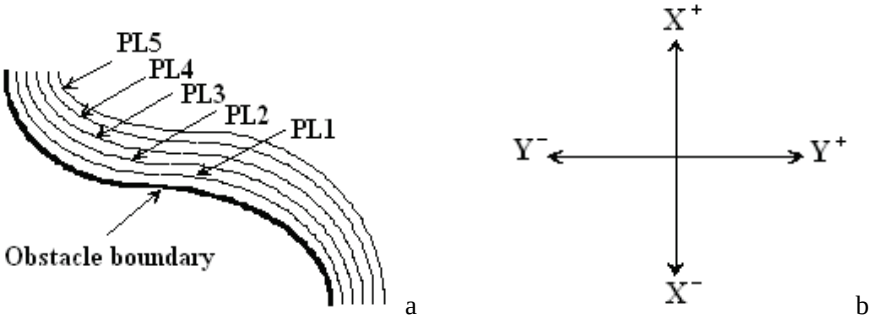


Fig. 4.1. The proximity levels and the degrees of freedom of the robot motion

The goal of the proposed control algorithm is to move the robot near the object boundary with collision avoidance. Fig. 4.2 shows four motion cycles (programs) which are followed by the mobile robot on the trajectory (P1, P2, P3, and P4). Inside every cycle are presented the directions of the movements (with arrows) for every reached proximity level. For example, if the mobile robot is moving inside first motion cycle (cycle 1 or program P1) and is reached PL3, the direction is on Y-axis (sense plus) (see Fig. 4.1b, too).

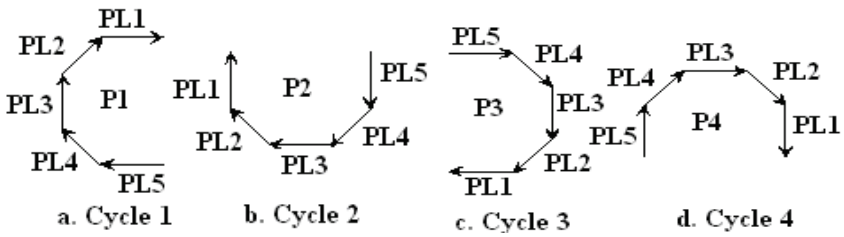


Fig. 4.2. The four motion cycles (programs)

In Fig. 4.3 we can see the sequence of the programs. One program is changed when are reached the proximity levels PL1 or PL5. If PL5 is reached the order of changing is: P1→P2→P3→P4→P1→ …… If PL1 is reached the sequence of changing becomes: P4→P3→P2→P1→P4→ ……

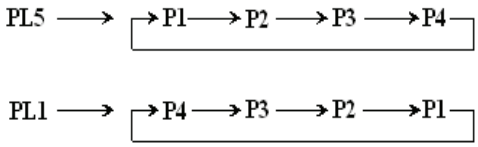


Fig. 4. 3. The sequence of the programs

The motion control algorithm is presented in Fig. 4.4 by a flowchart of the evolution of the functional cycles (programs). We can see that if inside a program the proximity levels PL2, PL3 or PL4 are reached, the program is not changed. If PL1 or PL5 proximity levels are reached, the program is changed. The flowchart is built on the base of the rules presented in Fig. 4.2 and Fig. 4.3.

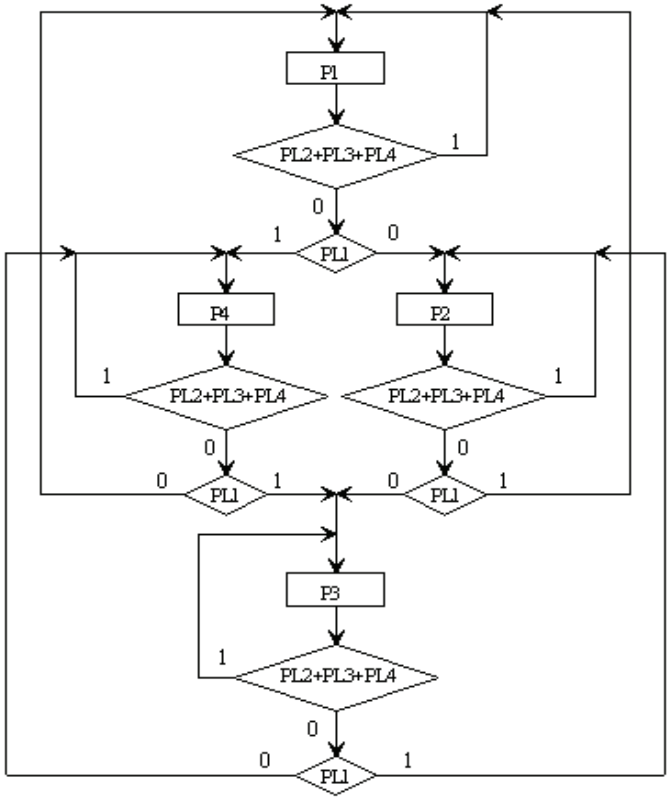


Fig. 4.4 The flowchart of the evolution of the functional cycles (programs)

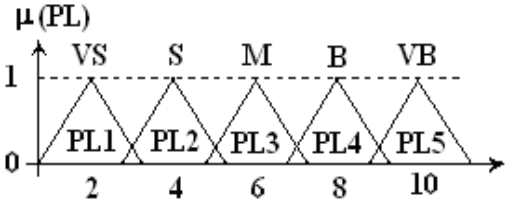
4.2 Fuzzy algorithm

The fuzzy controller for the mobile robots based on the algorithm presented above is simple. Most fuzzy control applications, such as servo controllers, feature only two or three inputs to the rule base. This makes the control surface simple enough for the programmer to define

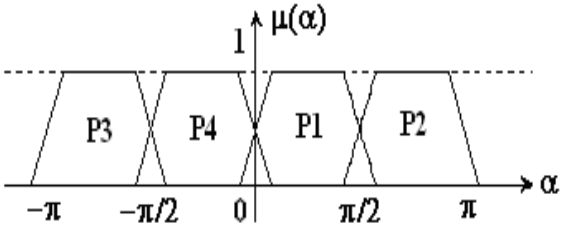
explicitly with the fuzzy rules. The above robot example uses this principle, in order to explore the feasibility of using fuzzy control for its tasks. Fig. 4.5 presents the inputs (distance-proximity levels and the program on k step) and the outputs (movement on X and Y-axes and the program on k+1 step) of the fuzzy algorithm.

Fig. 4.5. The inputs and outputs of the fuzzy algorithm

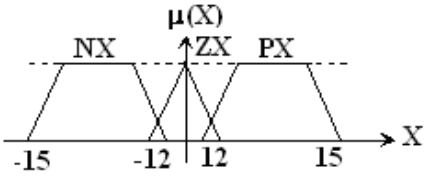
For the linguistic variable "distance proximity level" we establish to follow five linguistic terms: "VS-very small", "S-small", "M-medium", "B-big", and "VB-very big". Fig. 4.6a shows the membership functions of the proximity levels (distance) measured with the sensors and Fig. 4.6b shows the membership functions of the angle (the programs). If the object is like a circle every program is proper for a quarter of the circle.



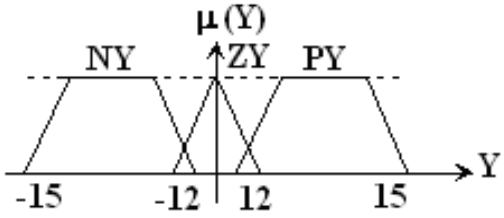
a) Membership functions of the proximity levels (distance) measured with the sensors



b) Membership functions of the angle (the programs)



c) Membership functions of the X commands



d) Membership functions of the Y commands

Fig. 4.6 Membership functions of the I/O variables

Fig. 4.6c and Fig. 4.6d present the membership functions of the X, respectively Y commands (linguistic variables). The linguistic terms are: NX-negative X, ZX-zero X, PX-positive X, and NY, ZY, PY respectively.

	VS	S	M	B	VB
P1	P4	P1	P1	P1	P2
P2	P1	P2	P2	P2	P3
P3	P2	P3	P3	P3	P4
P4	P3	P4	P4	P4	P1

Table 4.1. Fuzzy rules for evolution of the programs

	VS	S	M	B	VB
P1	PX	PX	ZX	NX	NX
P2	ZX	NX	NX	NX	ZX
P3	NX	NX	ZX	PX	PX
P4	ZX	PX	PX	PX	ZX

Table 4.2. Fuzzy rules for the motion on X-axis

	VS	S	M	B	VB
P1	ZY	PY	PY	PY	ZY
P2	PY	PY	ZY	NY	NY
P3	ZY	NY	NY	NY	ZY
P4	NY	NY	ZY	PY	PY

Table 4.3. Fuzzy rules for the motion on Y-axis

Table 4.1 describes the fuzzy rules for evolution (transition) of the programs and Table 4.2 and Table 4.3 describe the fuzzy rules for the motion on X-axis and Y-axis, respectively. Table 1 implements the sequence of the programs (see Fig. 4.2 and Fig. 4.4) and Table 4.2 and Table 4.3 implement the motion cycles (see Fig. 4.2 and Fig. 4.4).

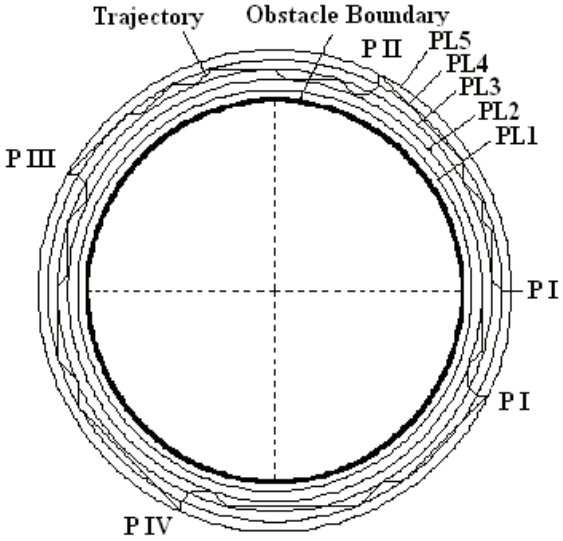


Fig. 4.7. The trajectory of the mobile robot around a circular obstacle

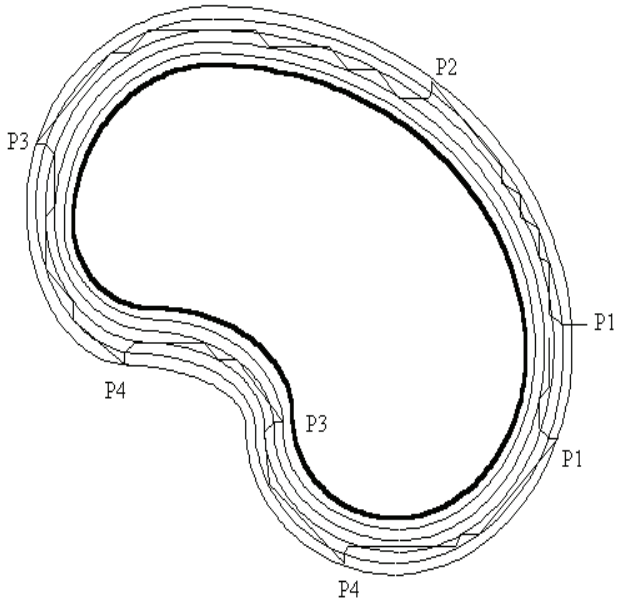


Fig. 4.8. The trajectory of the mobile robot around a irregular obstacle

4.3 Simulations

In the simulations can be seen the mobile robot trajectory around an obstacle (object) with circular boundaries (Fig. 4.7) and around an obstacle (object) with irregular boundaries (Fig. 4.8). One program is changed when are reached the proximity levels PL1 or PL5. If PL5 is reached the order of changing becomes as follows: P1→P2→P3→P4→... If PL1 is reached the order of changing is becomes follows: P4→P3→P2→P1→ P4→ ……

5. Conclusions

The section 3 presents a new control method for mobile robots moving in its work field which is based on fuzzy logic and artificial potential field. First, the artificial potential field method is presented. The section treats unconstrained movement based on attractive artificial potential field and after that discuss the constrained movement based on attractive and repulsive artificial potential field. A fuzzy controller is designed. Finally, some applications are presented.

The section 4 presents a fuzzy control algorithm for mobile robots which are moving next to the obstacle boundaries, avoiding the collisions with them. Four motion cycles (programs) depending on the proximity levels and followed by the mobile robot on the trajectory (P1, P2, P3, and P4) are shown. The directions of the movements corresponding to every cycle, for every reached proximity level are presented. The sequence of the programs depending on the reached proximity levels is indicated. The motion control algorithm is presented by a flowchart showing the evolution of the functional cycles (programs). The fuzzy rules for evolution (transition) of the programs and for the motion on X-axis and Y-axis respectively are described**.** The fuzzy controller for the mobile robots based on the algorithm presented above is simple. Finally, some simulations are presented**.** If the object is like a circle, every program is proper for a quarter of the circle.

6. References

Zadeh, L. D. (1965). *Fuzzy Sets, Information and Control*, No 8, pp. 338-365.

Sugeno, M.; Murofushi, T., Mori, T., Tatematasu, T. & Tanaka, J. (1989). Fuzzy algorithmic control of a model car by oral instructions, *Fuzzy Sets and Systems,* No. 32, pp. 207-219.

Song, K.Y. & Tai, J. C. (1992). Fuzzy navigation of a mobile robot, *Proceedings of the 1992 IEEE/RSJ Intern. Conference on Intelligent Robots and Systems*, Raleigh, North Carolina, USA.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *International Journal of Robotics Research*, Vol. 5, No.1, pp. 90-98.

Mamdani, E.H.; Folger, T.A. & Gaines, R.R. (1981). *Fuzzy reasoning and its spplications*, Academic Press, London.

Yan, I.; Ryan, M., & Power, I. (1994). *Using Logic Towards Intelligent Systems*, Prentice Hall, New York.

Van der Rhee, F. (1990). Knowledge based fuzzy control system, *IEEE Transactions on Automatic Control*, Vol. 35, No. 2.

Gupta, M.M.; Ragade, R. K. & Yager, R.R. (1979). *Advances in Fuzzy Set Theory and Applications*, North Holland, New York.

Dubois, D. & Prade, M. (1979). *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, New York.

Jager, R & Filev, D.P. (1994). *Essentialt of Fuzzy Modeling and Control*, John Wiley-Interscience Publication, New York.

Boreinstein, J. & Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots, *IEEE Trans. on Systems, Man., and Cybernetics*, Vol. 19, No. 5, Sept/Oct. pp. 1179-1187.

Jamshidi, M.; Vadiee, N. & Ross, T. J. (1993). *Fuzzy Logic and Control. Software and Hardware Applications*, PTR, Prentice Hall, New Jersey, USA.

Ivanescu, M. (2007). *From Classical to Modern Mechanical Engineering-Fundamentals*, Ed. Academiei Romane, ISBN 978-973-27-1561-1, Bucharest, Romania.

Schilling, R.J. (1990). *Robot Control*, Prentice Hall Inc. pp. 235-306, New York, USA.

Sprinceana, N.; Dobrescu, R & Borangiu. Th. (1978). *Digital Automations in Industry*, Ed. Tehnica, pp. 115-299, Bucharest, Romania (in Romanian).

Sugeno, M. (1985). An Introductory Survey of Fuzzy Control, *Informational Science*, Vol. 36.

Douskaia, N.V., (1998). Artificial potential method for control of constrained robot motion. *IEEE Trans. On Systems, Man and Cybernetics*, part B, **28**, pp. 447-453.

Hashimoto, H., Y. Kunii, F. Harashima, V.I. Utkin, and S.V. Grakumov (1993). Obstacle Advoidance control of multi degree of freedom manipulator using electrostatic potential field and sliding mode. *J. Robot Soc. Jpn.,* vol. **11**, no. 8, pp. 1220-1228.

Ivanescu, M., Stoian, V. (1995). Variable Structure Controller for a Tentacle Manipulator. *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 21-27, vol. **3**, pp. 3155-3160, ISBN: 0-7803-1967-2.

Ivanescu, M. (2001). Moving target interception for walking robot by fuzzy controller. *Proceedings of the Fourth International Conference on Climbing and Walking Robotics (CLAWAR 2001)*, pp. 363-376.

Khatib., O. (1986). Real-time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res*., vol. **5**, no. 1, pp. 90-98.

Latombe J.C. (1991). *Robot Motion Planning*, Kluwer Academic Publishers, Boston.

Masoud, S.A., Masoud, A.A. (2000). Constrained motion control using vector potential fields, *IEEE Trans. On Systems, Man and Cybernetics*, part A, 30, pp. 251-272.

Mohri, A., X. D Yang & A. Yamamoto, (1995). Collision free trajectory planning for manipulator using potential function. *Proceedings 1995 IEEE International Conference on Robotics and Automation*, pp. 3069-3074.

Morasso, P.G., V. Sanguineti & T. Tsuji, (1993). A Dynamical Model for the Generator of Curved Trajectories, in *Proceedings International Conference on Artificial Neural Networks*, pp. 115-118.

Sundar, S. & Z. Shiller. (1995). Time-optimal Obstacle Avoidance, *Proceedings 1995 IEEE International Conference on Robotics and Automation*, pp. 3075-3080.

Takegaki, M. & S. Arimoto (1981). A new feedback methods for dynamic control of manipulators. *Journal of Dynamic Systems, Measurement and Control*, pp. 119-125.

Tsugi, T., Y. Tanaka, P.G. Morasso, V. Sanguineti & M. Kaneko (2002). Bio-mimetic trajectory generation of robots via artificial potential field with time base generator. *IEEE Trans. On Systems, Man and Cybernetics*, part C, **32**, pp. 426-439.

***, (1994). A Model for the Generator of Target signals in Trajectory Formation, *Advances in Handwriting and Drawing: A Multidisciplinary Approach*, Faure, Kuess, Lorette, and Vinter Eds., Europia, Paris, pp. 333-348.

8

Robust Underdetermined Algorithm Using Heuristic-Based Gaussian Mixture Model for Blind Source Separation

Tsung-Ying Sun, Chan-Cheng Liu, Tsung-Ying Tsai, Yu-Peng Jheng, Jyun-Hong Jheng
National Dong Hwa University
Taiwan, R.O.C.

1. Introduction

Blind source separation (BSS) involves recovering unobserved source signals from several mixed observations, typically obtained at the output of a set of sensors. Each sensor receives a different combination of the source signals. The adjective "blind" emphasizes the fact that: first, the source signals are not observed; and next, no information is available about the mixture. The assumption is often held physically that the source signals are mutually independent.

Recently, BSS in signal processing has received considerable attention from researchers, due to its numerous promising applications in the areas of biomedical signal processing, digital communications and speech signal, sonar, image processing, and monitoring (Cichocki & Unbehauen, 1996), (Tangdiongga et al, 2001), (Yilmaz & Rickard, 2004), (Herault & Juten, 1986). A number of blind separation algorithms have been proposed based on different separation models. These algorithms play increasingly important roles in many applications. Since the pioneering work of Jutten and Herault (Herault & Juten, 1986), a variety of algorithms have been proposed for BSS. In general, the existing algorithms can be divided into five major categories: neural network-based algorithms (Cichocki & Unbehauen, 1996), (Zhang & Kassam, 2004), density model-based algorithms (Amari et al, 1997), (Lee et al, 1999a), algebraic algorithms (Belouchrani et al, 1997), (Li & Wang, 2002), information-theoretic algorithms (Pajunen, 1998), (Pham & Vrins, 2005) and space-based algorithms (Yilmaz & Rickard, 2004), (Lee et al, 1999b).

A source signal with sparse representation means that at most one value of the signal isn't zero at an instant, making the vector of sensor signals (mixtures) equivalent to some mixing vector. Therefore, the sparse-based BSS problem could be solved by searching for mixing vectors; moreover, recovering source signals. Like the mixtures with sparse representation, each base vectors of the unknown mixing matrix will be displayed on a 2-D plane coordinate as a directional line when two sensors are used. The sparse representation is first introduced in underdetermined BSS by Lee et al. (Lee et al, 1999b). After its introduction, several related methods have been proposed continuously for solving underdetermined BSS

cases. Bofill and Zibulevsky proposed a potential function to produce an approximate curve, which is able to describe the histogram of mixtures (Bofill & Zibulevsky, 2001). Shi et al. proposed the generalized exponential mixture model to approximate the distribution of mixtures without any predefined parameters (Shi et al, 2004). Besides, clustering methods such as fuzzy clustering, *k*-means, and other extensions were proposed to search for the mixing matrix (Grady & Pearlmutter, 2004), (Liu et al, 2006), (Vaerenbergh & Santamaria, 2006). These aforementioned algorithms provide good performance in well-conditioned mixing matrices which include identifiable mixing vectors.

Generally speak, the efficiency of signal recovering is dependent upon the precision of mixing matrix identification in a BSS problem; however, some practical and difficult conditions occur in an outside of the lab environment. First, when source signals are not sparse enough, the non-sparseness of the signals has affects the identification with problems like noise. Second, distances between mixing vectors are not far enough to distinguish them; therefore, two or more vectors would be regarded as one. Here, these two problems are termed an ill-conditioned BSS case. So far conventional algorithms produce unsatisfactory performance in such a case since many sub-solutions arise from these interferences. In this study, these difficult will be aimed by a heuristic method and a well-known statistic model.

Heuristic learning has been utilized to tackle similar problems encountered in other BSS categories. For example, the neural network-based BSS algorithms use the Genetic Algorithm (GA) (Yue & Mao, 2002) or Particle Swarm Optimization (PSO) (Song et al, 2007) to deal with linear mixing matrix or nonlinear mixing matrix identification; however, space-based BSS algorithms have never adopted such a heuristic learning process except that is published in (Liu et al, 2007).

Recently Gaussian mixture model (GMM) has been suggested to learn or model a set has multiple mixing data through the maximum-likelihood (ML) estimator or the expectation-maximization (EM) algorithm. And, the validity has been demonstrated in many research fields (Hedelin & Skoglund, 2000), (Todros & Tabrikian, 2007), (Nikseresht & Gelgon, 2008). Since there is not only single mixing vector in a BSS problem, GMM with multiple Gaussian models should be associated. However, the question then arises about the effect of initial parameters and falling into a local optimum from training by ML or EM. At first, the most information is unobservable in BSS problem; thus, the hint is too short to give proper initial parameters. Second, mixture outliers or ill-conditioned mixing vectors would cause a lot of local optimum; therefore, we think that an ability of widely exploring is weighty enough to affect the separate performance.

According to above analyses, this study develops a flexible GMM whose parameters are trained by PSO. The fitness function of PSO is designed to evaluate the inverse of sum of densities of GMM. When centers of all Gaussian models is close to the directions of all mixing vectors, the fitness value would approximate to the low bound. In order to make PSO more efficient, the representation of mixtures are modified from 2-D to 1-D; meantime, the boundary is normalized into the interval $[0,1]$. The search range of PSO therefore becomes more compact. Further, the particle update function of PSO is improved by using a cluster information to replace the global best (*Gb*). This improvement is according to the property of the sparse signal so it is helpful to speed convergence and enhance accuracy. The simulations involving well-conditioned mixing vectors, ill-conditioned mixing vectors and unknown number of source are designed. In order to present advantages of proposed

algorithm, some existing underdetermined BSS algorithms and GMM-based algorithms will be performed in the simulation for performance comparison.

The remainder of this study is organized as follows: Section II presents the fundamental of BSS consisting of mixing model and recovery methods. Section III introduces the standard PSO and GMM. Section IV presents details of the proposed algorithm. Section V presents several BSS simulations and displays the compared results. The validity of appended parameters are analyzed and confirmed in Section VI. Section VII draws a brief conclusion for this work.

2. Underdetermined Blind Source Separation

2.1 Mixtures in Sparse Representation

In a situation where the sparse source signals are unobservable, $\mathbf{s}(t)=[s_1(t),\ldots,s_n(t)]^T$ which is zero-mean and is mutually (spatially) statistically independent (or as independent as possible), where n denotes the number of sources, $[\cdot]^T$ denotes the transpose operation and t=1, …, N is the instant time of sampling. The term "Sparse" means that only a small number of the s_i differs significantly from zero. The degree of sparsity is evaluated by the probability density function (PDF) as follows:

$$P_{saprse_i}(s_i)=\alpha_i\delta(s_i)+(1-\alpha_i)f_i(s_i),\quad i=1,2,\ldots,n \tag{1}$$

where α_i is the probability that a source is inactive, $\delta(\cdot)$ denotes Dirac's delta and $f_i(s_i)$ is the PDF of the ith source when it is active (Luengo et al, 2005). The actual acoustics have a higher degree of sparsity in the frequency domain than in the time domain. Consequently, this study addresses the source signals that fulfill the requirements of sparse in the frequency domain and not in the time domain.

The available sensor vector $\mathbf{x}(t)=[x_1(t),\ldots,x_m(t)]^T$, where m is the number of sensors, is given by

$$\mathbf{x}(t)=\mathbf{A}\mathbf{s}(t) \tag{2}$$

where $\mathbf{A}\in\mathbf{R}^{m\times n}$ is an unknown mixing matrix and is nonsingular. The definition of an underdetermined case is one that satisfies $n\geq m$. Because two is the most applicatory number of sensors to such a BSS problem, $m=2$ is considered in this study. Therefore, eq. (2) could be rewritten as

$$\mathbf{x}(t)=\begin{bmatrix}x_1(t)\\x_2(t)\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\end{bmatrix}\begin{bmatrix}s_1(t) & s_2(t) & \cdots & s_n(t)\end{bmatrix}^T \tag{3}$$

where the components of mixing matrix could be presented as

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \end{bmatrix} \tag{4}$$

The feasibility of applying such an algorithm to identify sparse representation is affected by the sparsity of source signals and the density of mixing vectors. Then, the assumption that the distance between two arbitrary mixing vectors is less than the doubled sum of variances of distribution for the corresponding mixtures is held in this study.

The process of BSS can be divided into two steps: the first is the unknown mixing matrix identification which will be discussed in Section IV. The second is source signals recovery by the estimation of mixing matrix, described in the next subsection.

2.2 Source Signal Recovery

According to the estimated mixing matrix, sparse source signals can be recovered by maximizing the posterior distribution that is formed as (Shi et al, 2004)

$$P(\mathbf{s}|\mathbf{x},\mathbf{A})=\prod_{t=1}^{N} P(\mathbf{s}(t)|\mathbf{x}(t),\mathbf{A}) \tag{5}$$

According to eq. (2) and Bayes' rule, the log-likelihood can be obtained by taking the logarithm of eq. (2):

$$L(\mathbf{s})=\sum_{t=1}^{N}\left\{-0.5(\mathbf{x}(t)-\mathbf{A}\mathbf{s}(t))^{T}\Sigma^{-1}(\mathbf{x}(t)-\mathbf{A}\mathbf{s}(t))+\log\{P(\mathbf{s}(t))\}\right\}+\beta \tag{6}$$

where Σ^{-1} indicates the noise covariance matrix and β is a constant irrelevant to $\mathbf{s}(t)$, and $(\cdot)^{T}$ denotes the transpose operation. In order to maximize eq. (6), the gradient of eq. (6) is derived with respect to $\mathbf{s}(t)$ as

$$\nabla_{\mathbf{s}(t)}L(\mathbf{s}(t))=\mathbf{A}^{T}\Sigma^{-1}(\mathbf{x}(t)-\mathbf{A}\mathbf{s}(t))+\nabla_{\mathbf{s}(t)}\log\{p(\mathbf{s}(t))\} \tag{7}$$

Therefore, the original signals can be recovered gradually by the following iteration:

$$\mathbf{s}^{j}(t)=\mathbf{s}^{j-1}(t)+\eta\nabla_{\mathbf{s}(t)}L(\mathbf{s}^{j-1}(t)) \tag{8}$$

where the superscript of $\mathbf{S}$ indicates the iteration index.

3. Introduced Techniques

3.1 Gaussian Mixture Model

A Gaussian mixture PDF for d-dimensional random vectors $\mathbf{X}$ is a weighted sum of

densities

$$f_{\mathbf{X}|\mathbf{\Theta}}\left(\mathbf{X}|\mathbf{\Theta}\right)=\sum_{i=1}^{M}\rho_i f_{\mathbf{X}|\theta_i}\left(\mathbf{X}|\theta_i\right) \tag{9}$$

where ρ_i are the component weights, M is the number of class and the component densities are Gaussian

$$f_{\mathbf{X}|\theta_i}\left(\mathbf{X}|\theta_i\right)= f_{\mathbf{X}|\mu_i,\mathbf{\Sigma}_i}\left(\mathbf{X}|\mu_i,\mathbf{\Sigma}_i\right)=\frac{1}{\left(2\pi\right)^{d/2}\left|\mathbf{\Sigma}_i\right|^{1/2}}e^{-\frac{1}{2}(\mathbf{X}-\mu_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{X}-\mu_i)} \tag{10}$$

with mean vectors μ_i and covariance matrices $\mathbf{\Sigma}_i$. The weights are constrained by $\rho_i>0$ and $\sum_{i=1}^{M}\rho_i=1$. The parameters of the Gaussian mixture density is represented by the set

$$\mathbf{\Theta}=\left\{\rho_1,\ldots,\rho_M,\mu_1,\ldots,\mu_M,\mathbf{\Sigma}_1,\ldots,\mathbf{\Sigma}_M\right\} \tag{11}$$

Generally, the expectation-maximization (EM) algorithm is a widely used procedure for maximum-likelihood (ML) estimation. It is an iterative algorithm where in each iteration over the same database a monotonic increase in the log-likelihood, $L\left(\mathbf{\Theta}\right)$, is guaranteed, i.e., $L\left(\mathbf{\Theta}^{(k+1)}\right)\geq L\left(\mathbf{\Theta}^{(k)}\right)$, where $\mathbf{\Theta}^{(k)}$ is the value of the parameter set $\mathbf{\Theta}$ at iteration k (Hedelin & Skoglund, 2000), (Nikseresht & Gelgon, 2008). A poor initialization of set $\mathbf{\Theta}$ would have great effect upon final performance; however, some elements are hard to give suitable initial values by experience of user. Consequently, this paper replaces the iterative method by PSO to obtain a more precise solution.

3.2 Heuristic Learning

The PSO is a population based optimization technique proposed by (Eberhart & Kennedy, 1995). The population is referred to as a *swarm*. The particles move and fast converge to local and/or global optimal position(s) over a small number of generations.

A swarm in PSO consists of a number of particles. Each particle represents a potential solution to the optimization task. All of the particles iteratively explore potential solutions through evolution. Each particle moves to a new position according to the new velocity which includes its previous velocity, and the directional vectors according to its own past best solution and global best solution. The best solution is then kept; each particle accelerates in the directions of not only the local best solution but also the global best position. If a particle discovers a new solution better than the global best solution, other particles will move closer to it in order to explore the region with more depth (Gudise & Venayagamoorthy, 2003).

Let *sz* denotes the swarm size. In general, there are three attributes, the particles' current position p_i, current velocity v_i , and local best position Pb_i, for particles in the search space to present their features. Each particle in the swarm is iteratively updated according to the

aforementioned attributes. Assuming that the fitness function $f(\cdot)$ is to be minimized, the new velocity of every particle is updated as

$$v_{ij}(g+1) = v_{ij}(g) + \alpha_1 \cdot r_1(g)\left[Pb_{ij}(g) - p_{ij}(g)\right] + \alpha_2 \cdot r_2(g)\left[Gb_j(g) - p_{ij}(g)\right] \tag{12}$$

where Pb_{ij} denotes the local best position of the ith particle and Gb_j(g) denotes the global best position at the gth generation. For all the index of dimension, $j = 1,2,\ldots,k$, v_{ij} is the velocity of the jth dimension of the ith particle, α_1 and α_2 denote the *acceleration coefficients*, r_1 and r_2 are elements from two uniform random sequences in the range (0, 1), g is the number of generations and has to be bounded in [$Vmin, Vmax$]. The new position of a particle is calculated as follows:

$$p_i(g+1) = p_i(g) + v_i(g+1) \tag{13}$$

The local best position of each particle is updated by

$$Pb_i(g+1) = \begin{cases} Pb_i(g), & \text{if } f(p_i(g+1)) > f(Pb_i(g)) \\ p_i(g+1), & \text{otherwise} \end{cases} \tag{14}$$

And the global best position Gb found from all particles during the previous three steps is defined as

$$Gb(g+1) = \arg\min_{Pb_i}\left\{f(Pb_i(g+1)),\ 1 \le i \le sz\right\} \tag{15}$$

Since Kennedy and Eberhart (Eberhart & Kennedy, 1995) introduced PSO in 1995, many researchers have worked on improving its performance in various ways. One of the variants called the standard PSO (Lin &. Feng, 2007), introduced by Shi and Eberhart (Shi & Eberhart, 1998), incorporates a parameter called inertia weight of velocity α_0 into the original PSO. The new velocity update algorithm is shown as follows:

$$v_{ij}(g+1) = \alpha_0 \cdot v_{ij}(g) + \alpha_1 \cdot r_1\left[Pb_{ij}(g) - p_{ij}(g)\right] + \alpha_2 \cdot r_2\left[Gb_j(g) - p_{ij}(g)\right] \tag{16}$$

This plays the role of balancing the global search and local search. It can be a positive constant or even a positive linear or nonlinear function of time. This value is typically setup to vary linearly from 1 to near 0 during the course of a training run. Note that this is reminiscent of the temperature adjustment schedule found in Simulated Annealing algorithms. The inertia weight is also similar to the momentum term in a gradient descent neural network training algorithm.

Although there are numerous variants for PSO, these methods spend too much time finishing fitness evaluations, and will present similar results in the early parts of convergence. To reach requirements of on-line separation, only a limited amount of computational time is available to produce a reasonable solution. These limitations will require an efficient and simple method. Hence, the variant of PSO with inertia weight is chosen (Shi & Eberhart, 1998).

4. Identify Mixing Vector by PSO-based GMM

4.1 PSO-based GMM

Suppose that only two sensors are available (i.e. $m = 2$) here, and then received mixtures are represented as $[x_1(t), x_2(t)]^T$, $t = 1,2,\ldots,N$,. Section II mentions that source signals are sparse and the mixtures center around the mixing vectors on the $x_1 - x_2$ coordinate plane. Thus, unobservable mixing vectors could emerge from these clusters of mixtures. Because n source signals will form n clusters of mixtures (mixing vectors), GMM with n densities is capable of expressing the distribution of mixtures. Further, the position of mixing vectors can be obtained from the centers of Gaussian model.

In order to find out the optimal parameters of GMM, the particles of PSO are regarded as a parameter set of GMM, $p = [\mu_1, \mu_2, \cdots, \mu_n]$. Because the estimate of each mixing vector should have equal importance in BSS case, the complexity of PSO could be reduced by omitting the estimate of weights, ρ_i. And, since covariance can be evaluated from classed mixtures, the estimate of covariance matrices $\mathbf{C}_i$ also could be omitted.

About the search space of PSO, there are several disadvantages when each element of particles is encoded into 2-D vector representation. First, there is no exact boundary for searching. Second, different elements would be mapped to the same solution. For instance $\mu = [2\times10^5, 10^5]$ is equivalent to $\mu = [1, 0.5]$. It would cause that particles explore in an infinite range; moreover, many equivalent solutions would pin down to each other. In order to enhance efficiency of optimization, the 2-D vector representation is replaced by 1-D angle representation; further, the angles are normalized in the interval from -1 to 1, i.e. $p \in [-1,1]^n$. The mixtures are therefore rewrote as

$$\hat{x}(t) = \frac{2\arctan\left(x_1(t)/x_2(t)\right)}{\pi}, \text{ where } \forall\ \hat{x} \in [-1, 1] \tag{17}$$

t=1,2,...,N; then, the search space of PSO become compact.

4.2 Improved PSO

Some potential improvements can be made to the exploration of PSO. In Fig. 1, which presents the distribution of mixtures, it could be seen that the two directions formed by the gathering mixtures imply two real mixing vectors. According to PSO's evolution, Gb would slowly approach these directions. In Fig. 1, the solid lines placed far from these directions indicate the Gb in the initial generation. The mixtures are further divided into

two clusters according to their distances to Gb; and then, cluster centers denoted by the dotted lines in Fig. 1 could be calculated. The result should reveal that the dotted lines are closer to the real mixing vectors than the solid lines.

Since mixtures gather toward the mixing vectors, cluster centers are more likely to produce a better solution than Gb. Moreover, it not only substantially improves Gb during initial generations, but also fine tunes Gb during final generations. Consequently, cluster information is the more preferable guide for particles compared to Gb. The factor Gb is replaced with cluster centers $\mathbf{C}_{bv}$ in eq.(12), which could be rewritten as:

$$\begin{aligned} v_{ij}(g+1) = \alpha_0 \cdot v_{ij}(g) + \alpha_1 \cdot r_1\left[Pb_{ij}(g) - p_{ij}(g)\right] \\ + \alpha_2 \cdot r_2\left[c_j(g) - p_{ij}(g)\right] \end{aligned} \tag{18}$$

where $\left\{c_j \middle| j = 1,2,\ldots,k\right\} \in \mathbf{C}_{bv}$ is a set of cluster centers according to Gb, and each component is evaluated by:

$$c_j = \frac{\sum_{i=1}^{cn_j} \xi_i^2 \times \hat{x}_i}{\sum_{i=1}^{cn_j} \xi_i^2} \tag{19}$$

where $\xi_t = \sqrt{x_1^2(t) + x_2^2(t)}$, j denotes the index of the cluster, cn_j denotes the number of mixtures which belongs to the jth estimated vector, and i is the index of mixtures. Since the involved signals are sparse, ξ_i could be regarded as a weight to the angle of the ith mixture. In other words, mixtures with a larger ξ have a greater effect upon the cluster center that it belongs to, whereas others are noisy or even voiceless.

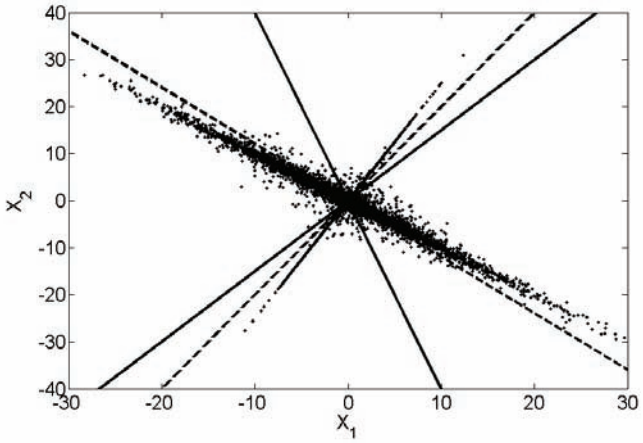


Fig. 1. The cluster vectors (dotted line) are derived from vectors of current *Gb* (solid line). The two directions constructed by gathered mixtures indicate the real mixing vectors.

4.3 Objective Function

As for the fitness function of the PSO, the property of sparse mixture distribution is introduced into our design. Since mixtures respectively gather toward mixing vectors and the vector length of mixtures is in respect to the energy of the signal, the fitness function is defined as

$$fit = \left(\sum_{i=1}^{M} f_{\hat{\mathbf{x}}|\mu_i}\left(\hat{\mathbf{x}}|\mu_i\right) \right)^{-1} \tag{20}$$

where the differential angle between the *i*th mixture vector and the nearest estimated vector is calculated by:

$$\Delta\hat{x}_i = \min\left\{ \left| \hat{x}_i - \mu_j \right|, i = 1,2,\ldots,N \text{ and } j = 1,2,\ldots,n \right\} \tag{21}$$

Consequently, a small fitness value implies a more accurate estimate to mixing vectors.

4.4 Disturbance

Additionally, in order to prevent the search from falling into a local optimum, a disturbance operation is added to PSO. Every current particle is allotted a random value between 0 and 1. A particle will carry out the disturbance sequence if its random value is less than a disturbance rate P_d . A disturbance particle is produced by:

$$p_{dis} = \left[\mu_1, \mu_2, \cdots, \mu_r + \varepsilon, \ldots, \mu_n \right] \tag{22}$$

where μ_r is a randomly selected dimension of the particle, and ε is a tiny disturbance factor. PSO begins with more uniformly scattered particles during initial generations, but incorporates more gathered particles during final generations, in the distribution of particles. Therefore, the value of P_d should be dependent upon the current evolutionary state of PSO. A floating P_d was decided to serve this purpose, its change is in respect to linear monotonic increase:

$$P_d(g) = \frac{P_D \times g}{G} \tag{23}$$

where P_D is the maximal P_d choosing from interval $[0,1]$, g is the current generation of PSO, and G is the maximal generation for evolution of PSO.

4.5 Algorithm Procedure

The procedure which uses the proposed GMM-PSO to deal with the underdetermined BSS problem is explained in Fig. 2, and the detail is described as following steps:

Step 1. Two mixtures mixed by eq. (3) is received.

Step 2. The both mixtures are transformed into frequency domain.

Step 3. (Start PSO) The initial particles are randomly produced from interval $[0,1]$.

Step 4. Fitness values are calculated by eq. (20) for each particle.

Step 5. *Pb* and *Gb* are updated by eq. (14) and eq. (15) respectively.

Step 6. The cluster centers $\mathbf{C}_{bv}$ are calculated by eq. (19) according the *Gb*; and then, replaces ***Gb***.

Step 7. New particles are produced according to eq. (18).

Step 8. Disturbance is carried out according a rate referred from eq. (23).

Step 9. If terminational condition is reached, i.e. $g = G$, then procedure goes to next step (End PSO); otherwise, it goes back to the Step 4.

Step 10. According to the final *Gb* to restructure mixing matrix; and then, recover signals are evaluated by eq. (8).

Further, there are several sub-steps in the Step 4 explaining how to evaluate fitness values:

Sub-step 1. Particle are inputted one by one.

Sub-step 2. The mixtures are classified into n classes by eq. (21) according to a particle.

Sub-step 3. The fitness value is calculated by eq. (20).

Sub-step 4. The fitness values of all particles are outputted.

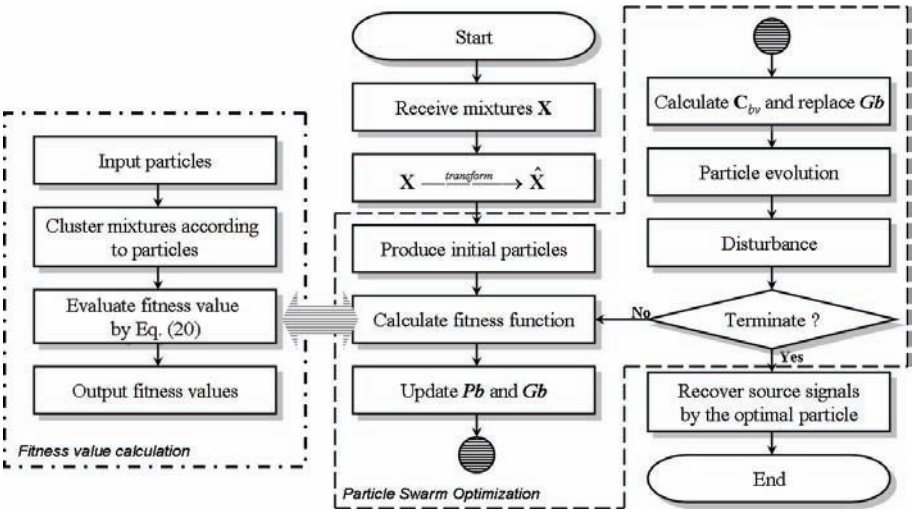


Fig. 2. The flowchart of the proposed algorithm

5. Simulations and Results

5.1 Description and Parameters Setting

Three other underdetermined algorithms with state of the art are tested in the following simulations to compare with the proposed algorithm. Here, the first one is named PF proposed in (Bofill & Zibulevsky, 2001), the second one is named GE proposed in (Shi et al, 2004), and the last one is our previous work which named FC proposed in (Liu et al, 2006).

In order to confirm validation and robustness of these algorithms, four sparse signals recorded from real sounds are taken for the source signals whose waveforms are shown in Fig. 3 and Fig. 4. In the first BSS case, the first three source signals are mixed by a well-conditioned mixing matrix as

$$\mathbf{A}_{well} = \begin{pmatrix} 0.3714 & 0.4472 & 0.7071 \\ 0.9285 & -0.8944 & 0.7071 \end{pmatrix} \tag{24}$$

It involves distinguishable mixing vectors whose normal angles are $\tilde{\mu}_{well}[0.2422, -0.2952, 0.5000]$ respectively. The distribution of mixtures is plotted in Fig. 5. In the second BSS case, the four source signals are mixed by an ill-conditioned mixing matrix as

$$\mathbf{A}_{ill} = \begin{pmatrix} 0.7071 & 0.7071 & 0.6097 & 0.3714 \\ 0.7071 & -0.7071 & 0.7926 & 0.9285 \end{pmatrix} \tag{25}$$

It involves undistinguishable mixing vectors whose normal angles are $\tilde{\mu}_{ill} = [0.5000, -0.5000, 0.4175, 0.2422]$ respectively since the first vector and the third vector are quite close. The distribution of mixtures is plotted in Fig. 6.

The parameters of compared algorithms are referenced from the original setting of their articles. For example, the grid scale is given 720 and λ is entered as 55 in PF. The improved PSO, through the experience of numerous previous experiments, are given the suitable parameters, $sz = 20$, $P_D = 0.5$, $\alpha_0 = 0.12$, $\alpha_1 = 0.3$ and $\alpha_2 = 0.4$. For generation number, $G = 100$ is given in first case and $G = 200$ is given in second case. For all algorithms, the every simulation will be tested by 30 independent runs. And, the performance is evaluated by mean square error (MSE) as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\tilde{\mu}_i - \mu_i)^2 \tag{26}$$

where $\tilde{\mu}_i$ denotes the ith real mixing vector, and μ_i denotes the ith estimation. In final, the average of MSE by 30 independent runs will be presented. An estimated set of mixing vector having a small MSE implies a excellent source separation.

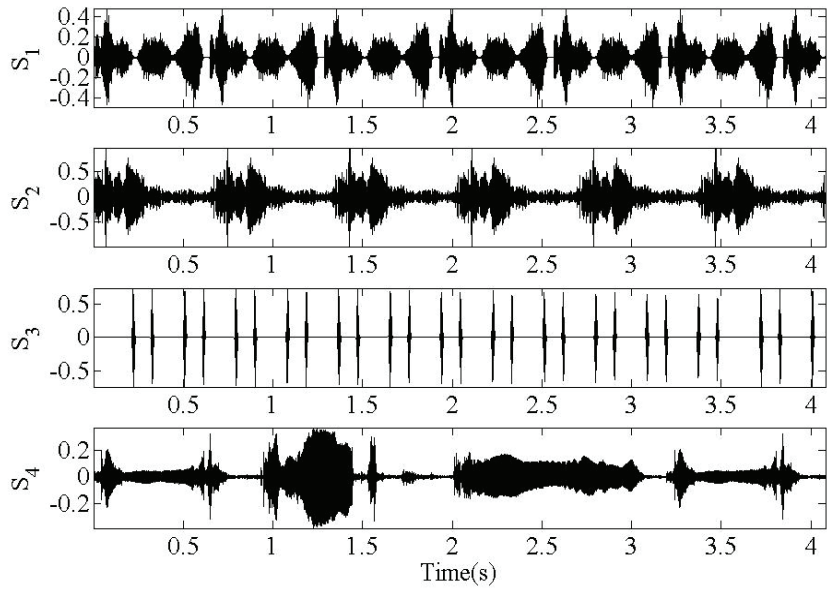


Fig. 3. The waveforms of source signals represented in time domain.

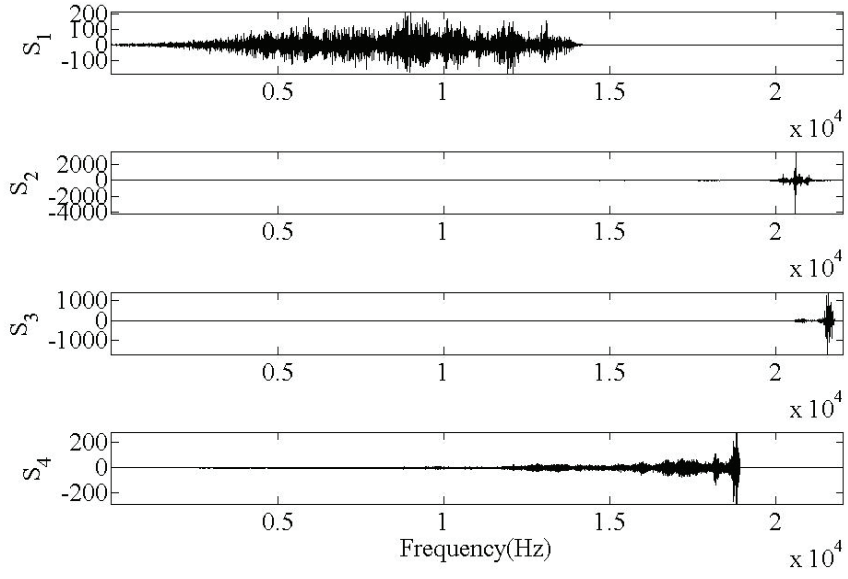


Fig. 4. The waveforms of source signals represented in frequency domain.

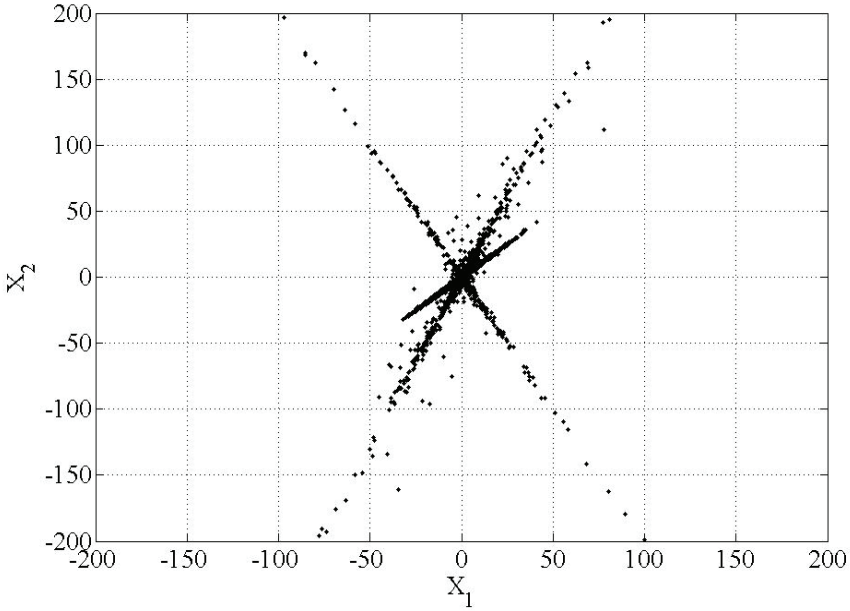


Fig. 5. The distribution of mixtures produced by well-conditioned mixing matrix.

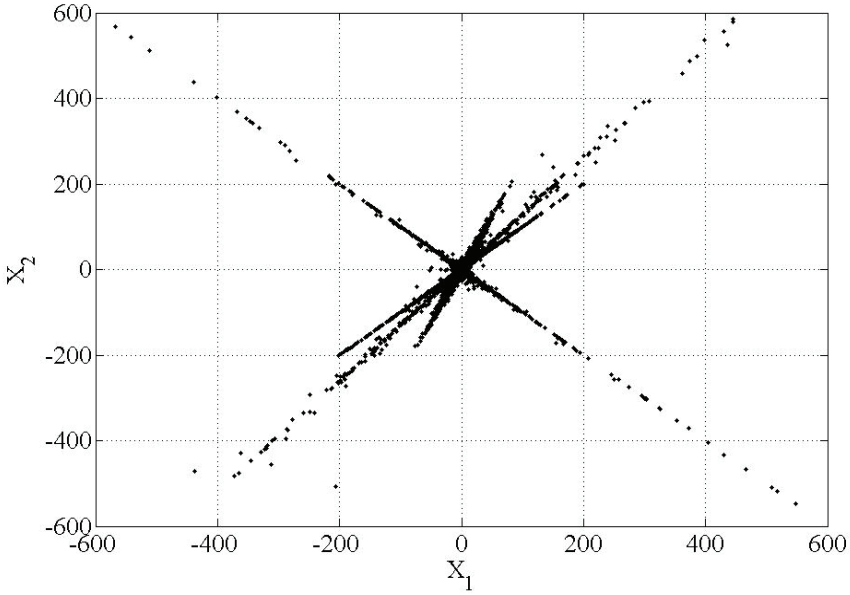


Fig. 6. The distribution of mixtures produced by ill-conditioned mixing matrix.

5.2 Results

After two simulations are implemented by the involved algorithms, the compared data about estimating accuracy are presented in Table 1 and Table 2. The both tables describe the real mixing vectors, the average of estimated mixing vectors and the MSE of the four algorithms for well-conditioned case and ill-conditioned case. From these tables, it could be observed that GE algorithm's performance is always unacceptable in all cases. PF algorithm just work acceptably in well-conditioned case, but it fail in ill-conditioned case. FC algorithm is valid in all cases, but its MSE is not better than that of proposed PSO-GMM algorithm.

In order to compare the improved PSO and standard PSO, their average fitness curves are shown in Fig. 7 (well-conditioned case) and Fig. 8 (ill-conditioned case). Form both figures, it could be observed that improved version has better convergent ability on speed and depth; particularly, that in Fig. 8.

Compared algorithms		PF	GE	FC	PSO-GMM
$\tilde{\mu}_1 = 0.2422$	μ_1	0.2497	0.2292	0.2498	0.2421
$\tilde{\mu}_2 = -0.2952$	μ_2	-0.2190	-0.1958	-0.2903	-0.2896
$\tilde{\mu}_3 = 0.5000$	μ_3	0.1627	0.1402	0.5134	0.4995
MSE		0.0399	0.0465	8.7110e-05	1.0540e-05

Table 1. Comparison of results between the four algorithms in well-conditioned BSS case.

Compared algorithms		PF	GE	FC	PSO-GMM
$\tilde{\mu}_1 = 0.5000$	μ_1	0.5520	0.6856	0.5000	0.4998
$\tilde{\mu}_2 = -0.5000$	μ_2	-0.4895	-0.6469	-0.4982	-0.4929
$\tilde{\mu}_3 = 0.4175$	μ_3	0.5639	0.5687	0.3996	0.4176
$\tilde{\mu}_4 = 0.2422$	μ_4	0.7494	-0.0817	0.2426	0.2426
MSE		0.0704	0.0460	8.0060e-05	1.2662e-05

Table 2. Comparison of results between the four algorithms in ill-conditioned BSS case.

6. Discussion

In comparing the proposed PSO-GMM with related BSS algorithms, the performance of GE algorithm is sensitive to predefined parameters. Tt exhibited a large value in MSE because of the lack of perfect initiations. Unfortunately, there is no rule or criterion that can be referred to for choosing suitable initiations. The PF algorithm is available in well-conditional case, and it does not involves any random initiation. However, the PF algorithm is not robust enough to deal with a complex problem because its settings of parameters is not for general-purpose; moreover, there are no instructions to guide a user on how to adjust them to suit other specific cases. The FC algorithm and PSO-GMM algorithm are efficient and robust enough to handle whether a general toy BSS case or an advanced BSS case. For further comparison between the both algorithms, it can be discovered that PSO method explores variant potential solutions; therefore, its accuracy is more excellent than FC algorithm. For the different PSO versions, the improved PSO exhibits a better convergent curve because it have the additional mechanism which enhances and replaces the globel best solution to rapidly drag particles toward a solution with an exact direction and distance during whole generations.

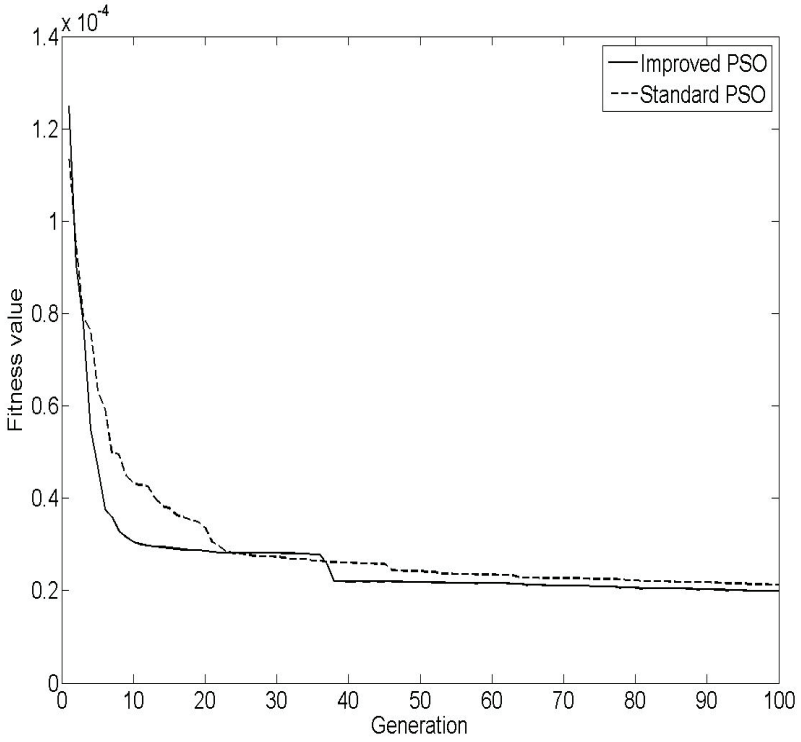


Fig. 7. Fitness convergence comparison between improved PSO and standard PSO for well-conditioned BSS case.

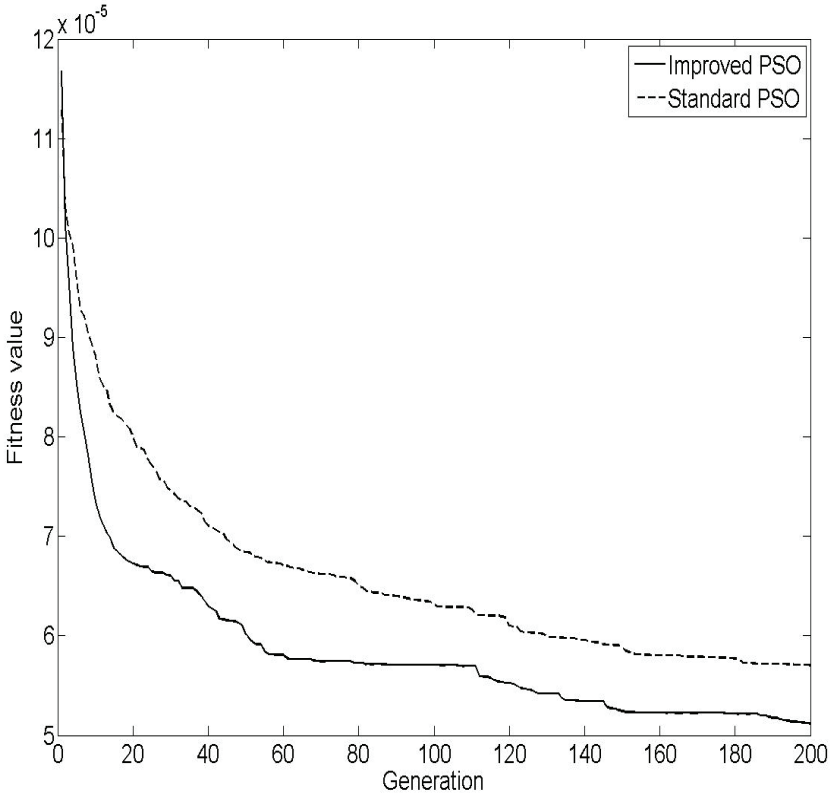


Fig. 8. Fitness convergence comparison between improved PSO and standard PSO for ill-conditioned BSS case.,

7. Conclusion

This study addresses on the BSS problem which involves sparse source signals, underdetermined linear mixing model. Some related algorithms have been proposed, but are only tested on toy cases. For robustness, GMM is introduced to learn the distribution of mixtures and find out the unknown mixing vectors; meantime, PSO is used to tune the parameters of GMM for expanding search range and avoiding local solutions as much as possible. Besides, a mechanism is proposed to enhance the evolution of PSO. For simulations, a simple toy case which includes distinguishable mixing matrix and a difficult case which includes close mixing vectors are designed and tested on several state of the art algorithms. Simulation results demonstrate that the proposed PSO-GMM algorithm has the best accuracy and robustness than others. Additionally, the comparison between standard PSO and improved PSO shows that improved PSO is more efficient than standard PSO.

8. References

Amari, S.; Chen, T. P. & Cichocki, A. (1997). Stability analysis of learning algorithms for blind source separation, *Neural Networks*, vol. 10, issue: 8, pp. 1345-1351

Belouchrani, A.; Abed-Meraim, K.; Cardoso, J. F. & Moulines, E. (1997). A blind source separation technique using second-order statistics, *IEEE Trans. on Acoustics, Speech, and Signal Processing,*vol. 45, issue: 2, pp. 434-444

Bofill, P. & Zibulevsky, M. (2001). Underdetermined blind source separation using sparse representations, *Signal Processing*, vol. 81, pp. 2353-2362

Cichocki, A. & Unbehauen, R. (1996). Robust neural networks with on-line learning for blind identification and blind separation of sources, *IEEE Trans. on circuits and systems-I: fundamental theory and applications*, vol. 43

Eberhart, R. C. & Kennedy, J. (1995). A new optimizer using particle swarm theory," *Proc. of the Sixth International Symposium on Micro Machine and Human*, pp. 39-43

Grady, P. O. & Pearlmutter, B. (2004). Soft-LOST: EM on a mixture of oriented lines, *Proc. of ICA 2004*, ser. *Lecture Notes in Computer Science Granada*, pp. 430-436

Gudise, V. G. & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and Backpropagation as Training Algorithms for Neural Networks, *Proc. of IEEE Swarm Intelligence Symposium*, pp. 110-117

Hedelin, P. & Skoglund, J. (2000). Vector quantization based on gaussian mixture models, *IEEE Trans. on Speech and Audio Processing,* vol. 8, no. 4, pp. 385-401

Herault, J. & Juten, C. (1986). Space or time adaptive signal processing by neural network models, *Proc. of AIP Conf. Snowbird, UT, in Neural Networks for Computing*, J. S. Denker, Ed. New York: Amer. Inst. Phys., pp. 206-211

Lee, T. W.; Girolami, M. & Sejnowski, T. J. (1999a). Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources, *Neural Computation*, vol. 11, issue: 2, pp. 409-433

Lee,T. W.; Lewicki, M. S.; Girolami, M. & Sejnowski, T. J. (1999b). Blind source separation of more sources than mixtures using overcomplete representations, *Signal Processing Letters*, vol. 6, issue: 4, pp. 87-90

Li, Y. & Wang, J. (2002). Sequential blind extraction of instantaneously mixed sources, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. 50, issue: 5, pp. 997-1006

Lin, C. & Feng, Q. (2007). The standard particle swarm optimization algorithm convergence analysis and parameter selection, *Proc. of the 3th International Conference on Natural Computation*, pp. 823-826

Liu, C. C.; Sun, T. Y.; Li, K. Y. & Lin, C. L. (2006). Underdetermined blind signal separation using fuzzy cluster on mixture accumulation, *Proc. of the International Symposium on Intelligent Signal Processing and Communication Systems*, pp. 455-458

Liu, C. C.; Sun, T. Y.; Li, K. Y.; Hsieh, S. T. & Tsai, S. J. (2007). Blind sparse source separation using cluster particle swarm optimization technique, *Proc. of International Conference on Artificial Intelligence and Applications*, pp. 549-217

Luengo, D.; Santamaria, I. & Vielve, L. (2005). A general solution to blind inverse problems for sparse input signals: deconvolution, equalization and source separation, *Neurocomputing*, vol. 69, pp. 198-215

Nikseresht, A. & Gelgon, M. (2008). Gossip-based computation of a gaussian mixture model for distributed multimedia indexing, *IEEE Trans. on Multimedia,* vol. 10, no. 3, pp. 385-392

Pajunen, P. (1998). Blind source separation using algorithmic information theory, *Neurocomputing,* vol. 22, issue: 1-3, pp. 35-48

Pham, D. T. & Vrins, F. (2005). Local minima of information-theoretic criteria in blind source separation," *IEEE Signal Processing Letters*, vol. 12, issue: 11, pp. 788-791

Shi, Z.; Tang, H.; Liu, W. & Tang, Y. (2004). Blind source separation of more sources than mixtures using generalized exponential mixture models, *Neurocomputing*, vol. 61, pp. 461-469

Shi, Y. & Eberhart, R. C. (1998). A modified particle swarm optimizer**,** *Proc. of IEEE World Congress on Computational Intelligence*, pp. 69-73

Song, K.; Ding, M.; Wang, Q. & W. Liu, (2007). Blind source separation in post-nonlinear mixtures using natural gradient descent and particle swarm optimization algorithm, *Proc. of the 4th International Symposium on Neural Networks*, pp. 721-730

Tangdiongga, E.; Calabretta, N.; Sommen, P. C. W. & Dorren, H. J. S. (2001). WDM monitoring technique using adaptive blind signal separation, *IEEE Photonics Technology Letters,* vol. 13, issue: 3, pp. 248 - 250

Todros, K. & Tabrikian, J. (2007). Blind separation of independent sources using gaussian mixture model, *IEEE Trans. on Signal Processing,* vol. 55, no. 7, pp. 3645-3658

Vaerenbergh, S. V. & Santamaria, I. (2006). Spectral clustering approach to underdetermined postnonlinear blind source separation of sparse sources, *IEEE Trans. on Neural Networks,* vol. 17, issue: 3, pp. 811-814

Yilmaz, O. & Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. 52, issue: 7, pp. 1830-1847

Yue, Y. & Mao, J. (2002). Blind separation of sources based on genetic algorithm, *Proc. of the 4th World Congress on Intelligent Control and Automation*, pp. 2099-2103

Zhang, Y. C. & Kassam, S. A. (2004). Robust rank-EASI algorithm for blind source separation, *IEE Proceedings-Communications*, vol. 151, issue: 1, pp. 15-19

9

Pattern-driven Reuse of Behavioral Specifications in Embedded Control System Design

Miroslav Švéda, Ondřej Ryšavý & Radimir Vrba
Brno University of Technology
Czech Republic

1. Introduction

Methods and approaches in systems engineering are often based on the results of empirical observations or on individual success stories. Every real-world embedded system design stems from decisions based on an application domain knowledge that includes facts about some previous design practice. Evidently, such decisions relate to system architecture components, called in this paper as application patterns, which determine not only a required system behavior but also some presupposed implementation principles. Application patterns should respect those particular solutions that were successful in previous relevant design cases. While focused on the system architecture range that covers more than software components, the application patterns look in many features like well-known software object-oriented design concepts such as reusable patterns (Coad and Yourdon, 1990), design patterns (Gamma et al., 1995), and frameworks (Johnson, 1997). By the way, there are also other related concepts such as use cases (Jacobson, 1992), architectural styles (Shaw and Garlan, 1996), or templates (Turner, 1997), which could be utilized for the purpose of this paper instead of introducing a novel notion. Nevertheless, application patterns can structure behavioral specifications and, concurrently, they can support architectural components specification reuse.

Nowadays, industrial scale reusability frequently requires a knowledge-based support. Case-based reasoning (see e.g. Kolodner, 1993) can provide such a support. The method differs from other rather traditional procedures of Artificial Intelligence relying on case history: for a new problem, it strives for a similar old solution saved in a case library. Any case library serves as a knowledge base of a case-based reasoning system. The system acquires knowledge from old cases while learning can be achieved accumulating new cases. Solving a new case, the most similar old case is retrieved from the case library. The suggested solution of a new case is generated in conformity with the retrieved old case. This book chapter proposes not only how to represent a system's formal specification as an application pattern structure of specification fragments, but also how to measure similarity of formal specifications for retrieval. In this chapter, case-based reasoning support to reuse is focused on specifications by finite-state and timed automata, or by state and timed-state

sequences. The same principles can be applied for specifications by temporal and real-time logics.
The following sections of this chapter introduce the principles of design reuse applied by the way of application patterns. Then, employing application patterns fitting a class of real-time embedded systems, the kernel of this contribution presents two design projects: petrol pumping station dispenser controller and multiple lift control system. Via identification of the identical or similar application patterns in both design cases, this contribution proves the possibility to reuse substantial parts of formal specifications in a relevant sub-domain of embedded systems. The last part of the paper deals with knowledge-based support for this reuse process applying case-based reasoning paradigm.
The contribution provides principles of case-based reasoning support to reuse in frame of formal specification-based system design aiming at industrial applications domain. This book chapter stems from the paper (Sveda, Vrba and Rysavy, 2007) modified and extended by deploying temporal logic formulas for specifications.

2. State of the Art

To reuse an application pattern, whose implementation usually consists both of software and hardware components, it means to reuse its formal specification, development of which is very expensive and, consequently, worthwhile for reuse. This paper is aimed at behavioral specifications employing state or timed-state sequences, which correspond to Kripke style semantics of linear, discrete time temporal or real-time logics, and at their closed-form descriptions by finite-state or timed automata (Alur and Henzinger, 1992). Geppert and Roessler (2001) present a reuse-driven SDL design methodology that appears closely related approach to the problem discussed in this contribution.
Software design reuse belongs to highly published topics for almost 20 years, see namely Frakes and Kang (2005), but also Arora and Kulkarni (1998), Sutcliffe and Maiden (1998), Mili et al. (1997), Holzblatt et al. (1997), and Henninger (1997). Namely the state-dependent specification-based approach discussed by Zaremski et. al. (1997) and by van Lamsweerde and Wilmet (1998) inspired the application patterns handling presented in the current paper. To relate application patterns to the previously mentioned software oriented concepts more definitely, the inherited characteristics of the archetypal terminology, omitting namely their exclusive software orientation, can be restated as follows. A pattern describes a problem to be solved, a solution, and the context in which that solution works. Patterns are supposed to describe recurring solutions that have stood the test of time. Design patterns are the micro-architectural elements of frameworks. A framework -- which represents a generic application that allows creating different applications from an application sub-domain -- is an integrated set of patterns that can be reused. While each pattern describes a decision point in the development of an application, a pattern language is the organized collection of patterns for a particular application domain, and becomes an auxiliary method that guides the development process, see the pioneer work by Alexander (1977).
Application patterns correspond not only to design patterns but also to frameworks while respecting multi-layer hierarchical structures. Embodying domain knowledge, application patterns deal both with requirement and implementation specifications (Shaw and Garlan, 1996). In fact, a precise characterization of the way, in which implementation specifications

and requirements differ, depends on the precise location of the interface between an embedded system, which is to be implemented, and its environment, which generates requirements on system's services. However, there are no strict boundaries in between: both implementation specifications and requirements rely on designer's view, i.e. also on application patterns employed.
A design reuse process involves several necessary reuse tasks that can be grouped into two categories: supply-side and demand-side reuse (Sen, 1997). Supply-side reuse tasks include identification, creation, and classification of reusable artefacts. Demand-side reuse tasks include namely retrieval, adaptation, and storage of reusable artefacts. For the purpose of this paper, the reusable artefacts are represented by application patterns.
After introducing principles of the temporal logic deployed in the following specifications, next sections provide two case studies, based on implemented design projects, using application patterns that enable to discuss concrete examples of application patterns reusability.

3. Temporal Logic of Actions

Temporal Logic of Actions (TLA) is a variant of linear-time temporal logic. It was developed by Lamport (1994) primarily for specifying distributed algorithms, but several works shown that the area of application is much broader. The system of TLA+ extends TLA with data structures allowing for easier description of complex specification patterns.
TLA+ specifications are organized into modules. Modules can contain declarations, definitions, and assertions by means of logical formulas. The declarations consist of constants and variables. Constants can be uninterpreted until an automated verification procedure is used to verify the properties of the specification. Variables keep the state of the system, they can change in the system and the specification is expressed in terms of transition formulas that assert the values of the variables as observed in different states of the system that are related by the system transitions.
The overall specification is given by the temporal formula defined as a conjunction of the form

$$I \wedge \Box[N]_v \wedge L,$$

where I is the initial condition, N is the next-state relation (composed from transition formulas), and L is a conjunction of fairness properties, each concerning a disjunct of the next-state relation. Transition formulas, also called actions, are ordinary formulas of untyped first-order logic defined on a denumerable set of variables, partitioned into sets of flexible and rigid variables. Moreover, a set of primed flexible variables, in the form of v′, is defined. Transition formulas then can contain all these kinds of variables to express a relation between two consecutive states. The generation of a transition system for the purpose of model checking verification or for the simulation is governed by the enabled transition formulas. The formula $\Box[N]_v$ admits system transitions that leave a set of variables v unchanged. This is known as stuttering, which is a key concept of TLA that enables the refinement and compositional specifications. The initial condition and next-state relation specify the possible behaviour of the system. Fairness conditions strengthen the specification by asserting that given actions must occur.

The TLA+ does not formally distinguish between a system specification and a property. Both are expressed as formulas of temporal logic and connected by implication $S \Rightarrow F$, where S is a specification and F is a property. Confirming the validity of this implication stands for showing that the specification S has the property F.
The TLA+ is accompanied with a set of tools. One of such tool, the TLA+ model checker, TLC, is state-of-the-art model analyzer that can compute and explore the state space of finite instances of TLA+ models. The input to TLC consists of specification file describing the model and configuration file, which defines the finite-state instance of the model to be analysed. An execution of TLC produces a result that gives answer to the model correctness. In case of finding a problem, this is reported with a state-sequence demonstrating the trace in the model that leads to the problematic state. Inevitably, the TLC suffers the problem of state space explosion that is, nevertheless, partially addressed by a technique known as symmetry reduction allowing for verification of moderate size system specifications.

4. Petrol Dispenser Control System

The first case study pertains to a petrol pumping station dispenser with a distributed, multiple microcomputer counter/controller (for more details see Sveda, 1996). A dispenser controller is interconnected with its environment through an interface with volume meter (input), pump motor (output), main and by-pass valves (outputs) that enable full or throttled flow, release signal (input) generated by cashier, unhooked nozzle detection (input), product's unit price (input), and volume and price displays (outputs).

4.1 Two-level structure for dispenser control

The first employed application pattern stems from the two-level structure proposed by Xinyao et al. (1994): the higher level behaves as an event-driven component, and the lower level behaves as a set of real-time interconnected components. The behavior of the higher level component can be described by the following state sequences of a finite-state automaton with states "blocked-idle," "ready," "full fuel," "throttled" and "closed," and with inputs "release," (nozzle) "hung on/off," "close" (the preset or maximal displayable volume achieved), "throttle" (to slow down the flow to enable exact dosage) and "error":

blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{hung off}}$ full_fuel $\xrightarrow{\text{hung on}}$ blocked-idle
blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{hung off}}$ full_fuel $\xrightarrow{\text{throttle}}$ throttled $\xrightarrow{\text{hung on}}$ blocked-idle
blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{hung off}}$ full_fuel $\xrightarrow{\text{throttle}}$ throttled $\xrightarrow{\text{close}}$ closed $\xrightarrow{\text{hung on}}$ blocked-idle
blocked-idle $\xrightarrow{\text{error}}$ blocked-error
blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{error}}$ blocked-error
blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{hung off}}$ full_fuel $\xrightarrow{\text{error}}$ blocked-error
blocked-idle $\xrightarrow{\text{release}}$ ready $\xrightarrow{\text{hung off}}$ full_fuel $\xrightarrow{\text{throttle}}$ throttled $\xrightarrow{\text{error}}$ blocked-error

The states "full_fuel" and "throttled" appear to be hazardous from the viewpoint of unchecked flow because the motor is on and the liquid is under pressure -- the only nozzle valve controls an issue in this case. Also, the state "ready" tends to be hazardous: when the nozzle is unhooked, the system transfers to the state "full_fuel" with flow enabled. Hence, the accepted fail-stop conception necessitates the detected error management in the form of transition to the state "blocked-error." To initiate such a transition for flow blocking, the error detection in the hazardous states is necessary. On the other hand, the state "blocked-

idle" is safe because the input signal "release" can be masked out by the system that, when some failure is detected, performs the internal transition from "blocked-idle" to "blocked-error."

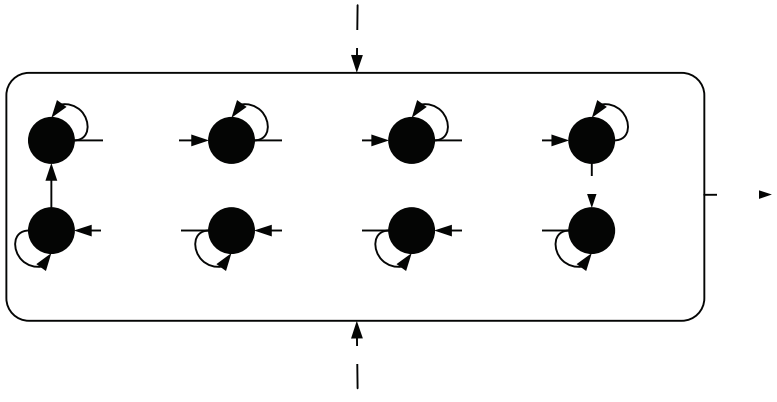


Fig. 1. Noise-tolerant impulse recognition automaton of length 8

4.2 Incremental measurement for flow control

The volume measurement and flow control represent the main functions of the hazardous states. The next applied application pattern, incremental measurement, means the recognition and counting of elementary volumes represented by rectangular impulses, which are generated by a photoelectric pulse generator. The maximal frequency of impulses and a pattern for their recognition depend on electro-magnetic interference characteristics. The lower-level application patterns are in this case a noise-tolerant impulse detector and a checking reversible counter. The first one represents a clock-timed impulse-recognition automaton that implements the periodic sampling of its input with values 0 and 1. This automaton with b states recognizes an impulse after b/2 (b>=4) samples with the value 1 followed by b/2 samples with the value 0, possibly interleaved by induced error values, see an example timed-state sequence:

$$(0, q_1) \overset{inp=0}{\rightarrow} \ldots \overset{inp=0}{\rightarrow} (i, q_1) \overset{inp=1}{\rightarrow} (i+1, q_2) \overset{inp=0}{\rightarrow} \ldots \overset{inp=0}{\rightarrow} (j, q_2) \ldots$$

$$\ldots \overset{inp=1}{\rightarrow} (k, q_{b/2+1}) \overset{inp=1}{\rightarrow} \ldots$$

$$\ldots \overset{inp=1}{\rightarrow} (m, q_{b-1}) \overset{inp=0}{\rightarrow} (m+1, q_b) \overset{inp=1}{\rightarrow} \ldots \overset{inp=1}{\rightarrow} (n, q_b) \overset{inp=0/IMP}{\rightarrow} (n+1, q_1)$$

i, j, k, m, n are integers representing discrete time instances in increasing order.

For the sake of fault-detection requirements, the incremental detector and transfer path are doubled. Consequently, the second, identical noise-tolerant impulse detector appears necessary.

The subsequent lower-level application pattern used provides a checking reversible counter, which starts with the value (h + l)/2 and increments or decrements that value according to

the "impulse detected" outputs from the first or the second recognition automaton. Overflow or underflow of the pre-set values of h or l indicates an error. Another counter that counts the recognized impulses from one of the recognition automata maintains the whole measured volume. The output of the letter automaton refines to two displays with local memories not only for the reason of robustness (they can be compared) but also for functional requirements (double-face stand). To guarantee the overall fault detection capability of the device, it is necessary also to consider checking the counter. This task can be maintained by an I/O watchdog application pattern that can compare input impulses from the photoelectric pulse generator and the changes of the total value; evidently, the appropriate automaton provides again reversible counting.
The noise-tolerant impulse detector was identified as a reusable design-pattern and its abstract specification written using TLA+ can be stored in a case library. This specification is shown in Fig. 2. The actions *Count1* and *Count0* capture the behaviour of the automaton at sampling times. Action *Restart* defines an output of the automaton, which is to pose the signal on *impuls* output as the signalization of successful impulse detection.

MODULE *ImpulseRecognition*

EXTENDS *Naturals*
CONSTANT B
ASSUME $B \in Nat \land B \geq 4$
VARIABLE $hold, input, impuls, b$

$TypeInvariant \triangleq b \in (1 \,..\, B+1) \land input \in \{0, 1\} \land impuls \in \{0, 1\} \land hold \in \text{BOOLEAN}$
$vars \triangleq \langle hold, input, impuls, b \rangle$

$Init \triangleq b = 1 \land input = 1 \land impuls \in \{0, 1\} \land hold = \text{FALSE}$

$Time \triangleq hold = \text{TRUE} \land hold' = \text{FALSE} \land \text{UNCHANGED } \langle input, impuls, b \rangle$

$Input \triangleq input' = 1 - input \land \text{UNCHANGED } \langle impuls, b, hold \rangle$

$Count1 \triangleq hold = \text{FALSE} \land b \in (1 \,..\, B \div 2) \land input = 1$
$\land \; b' = b + 1 \land hold' = \text{TRUE} \land \text{UNCHANGED } \langle input, impuls \rangle$

$Count0 \triangleq hold = \text{FALSE} \land b \in ((B \div 2) + 1 \,..\, B) \land input = 0$
$\land \; b' = b + 1 \land hold' = \text{TRUE} \land \text{UNCHANGED } \langle input, impuls \rangle$

$Restart \triangleq b = B + 1$
$\land \; b' = 1 \land impuls' = 1 - impuls \land \text{UNCHANGED } \langle input, hold \rangle$

$Next \triangleq Time \lor Input \lor Count1 \lor Count0 \lor Restart$

$Spec \triangleq Init \land \Box[Next]_{vars} \land \Box\Diamond(hold = \text{TRUE}) \land \text{WF}_{vars}(Time)$

THEOREM $Spec \Rightarrow \Box\, TypeInvariant$

Fig. 2.Abstract TLA specification of noise-tolerant impulse recognition automaton

4.3 Fault Maintenance Concepts

The methods used a ccomplish the fault management in the form of (a) hazardous state

reachability control and (b) hazardous state maintenance. In safe states, the lift cabins are fixed at any floors. The system is allowed to reach any hazardous state when all relevant processors successfully passed the start-up checks of inputs and monitored outputs and of appropriate communication status. The hazardous state maintenance includes operational checks and, for shaft controller, the fail-stop support by two watchdog processors performing consistency checking for both execution processors. To comply with safety-critical conception, all critical inputs and monitored outputs are doubled and compared; when the relevant signals differ, the respective lift is either forced (in case of need with the help of an substitute drive if the shaft controller is disconnected) to reach the nearest floor and to stay blocked, or (in the case of maintenance or fire brigade support) its services are partially restricted. The basic safety hard core includes mechanical, emergency brakes.
Because permanent blocking or too frequently repeated blocking is inappropriate, the final implementation must employ also fault avoidance techniques. The other reason for the fault avoidance application stems from the fact that only approximated fail-stop implementation is possible. Moreover, the above described configurations create only skeleton carrying common fault-tolerant techniques see e.g. (Maxion et al., 1987). In short, while auxiliary hardware components maintain supply-voltage levels, input signals filtering, and timing, the software techniques, namely time redundancy or skip-frame strategy, deal with non-critical inputs and outputs.

5. Multiple Lift Control System

The second case study deals with the multiple lift control system based on a dedicated multiprocessor architecture (for more details see Sveda, 1997). An incremental measurement device for position evaluation, and position and speed control of a lift cabin in a lift shaft can demonstrate reusability. The applied application pattern, incremental measurement, means in this case the recognition and counting of rectangular impulses that are generated by an electromagnetic or photoelectric sensor/impulse generator, which is fixed on the bottom of the lift cabin and which passes equidistant position marks while moving along the shaft. That device communicates with its environment through interfaces with impulse generator and drive controller. So, the first input, I, provides the values 0 or 1 that are altered with frequency equivalent to the cabin speed. The second input, D, provides the values "up," "down," or "idle." The output, P, provides the actual absolute position of the cabin in the shaft.

5.1 Two-level structure for lift control

The next employed application pattern is the two-level structure: the higher level behaves as an event-driven component, which behavior is roughly described by the state sequence

initialization $\rightarrow$ position_indication $\rightarrow$ fault_indication

and the lower level, which behaves as a set of real-time interconnected components. The specification of the lower level can be developed by refining the higher level state "position_indication" into three communicating lower level automata: two noise-tolerant impulse detectors and one checking reversible counter.

5.2 Incremental measurement for position and speed control

Intuitively, the first automaton models the noise-tolerant impulse detector in the same manner as in previous case, see the following timed-state sequence:

$$(0, q_1) \overset{inp=0}{\rightarrow} \ldots \overset{inp=0}{\rightarrow} (i, q_1) \overset{inp=1}{\rightarrow} (i+1, q_2) \overset{inp=0}{\rightarrow} \ldots \overset{inp=0}{\rightarrow} (j, q_2) \ldots$$

$$\ldots \overset{inp=1}{\rightarrow} (k, q_{b/2+1}) \overset{inp=1}{\rightarrow} \ldots$$

$$\ldots \overset{inp=1}{\rightarrow} (m, q_{b-1}) \overset{inp=0}{\rightarrow} (m+1, q_b) \overset{inp=1}{\rightarrow} \ldots \overset{inp=1}{\rightarrow} (n, q_b) \overset{inp=0/IMP}{\rightarrow} (n+1, q_1)$$

i, j, k, m, n are integers representing discrete time instances in increasing order.

The information about a detected impulse is sent to the counting automaton that can also access the indication of the cabin movement direction through the input D. For the sake of fault-detection requirements, the impulse generator and the impulse transfer path are doubled. Consequently, a second, identical noise-tolerant impulse detector appears necessary. The subsequent application pattern is the checking reversible counter, which starts with the value (h + l)/2 and increments or decrements the value according to the "impulse detected" outputs from the first or second recognition automaton. Overflow or underflow of the preset values of h or l indicates an error. This detection process sends a message about a detected impulse and the current direction to the counting automaton, which maintains the actual position in the shaft. To check the counter, an I/O watchdog application pattern employs again a reversible counter that can compare the impulses from the sensor/impulse generator and the changes of the total value.

The reuse of the noise-tolerant impulse detector is desirable. To do this, suitable patterns stored in a case library need to be identified. The method for identification of candidate patterns is based on the behavioural similarity by means of inclusion of a state sequence in models of stored specifications. The TLA tools can be used for formal checking whether a design pattern stored in a case library contains a state sequence that describes the new design. The new TLA module Query (see Fig. 3) is generated for the purpose of checking whether the design pattern from the previous case study can be reused in the multiple lift control system. Note that the formula is negated in order to get an example of concrete state-sequence in a model of matched specification. The state-sequence is shown in Fig. 4. It has 25 unique states and describes the behaviour that conforms to the required state-sequence defining the intended behaviour of noise-tolerant impulse detector for lift control system. In the reuse scenario, the required size of the new automaton is 4. The stored design pattern in a case library can be parameterized; hence the model-checking procedure instantiates the constant B = 4, which is defined in the accompanying configuration file.

MODULE *Query*

EXTENDS *ImpulseRecognition*

$$Trace \triangleq \neg((b = 1 \wedge input = 1) \rightsquigarrow (b = 2 \wedge input = 1) \rightsquigarrow (b = 3 \wedge input = 0) \rightsquigarrow (b = 4 \wedge input = 0) \rightsquigarrow b = 1)$$

Fig. 3. TLA Module Query containing sought-after timed-state sequence

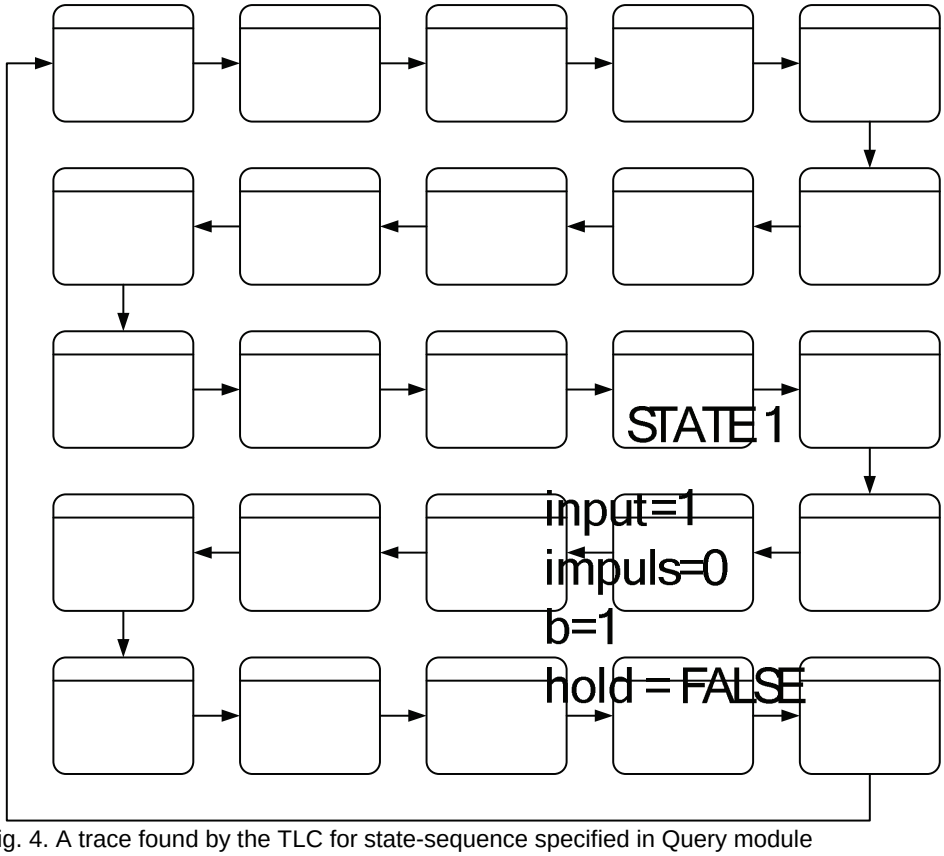


Fig. 4. A trace found by the TLC for state-sequence specified in Query module

5.3 Lift fault management

The approach used accomplishes a consequent application pattern, fault management based on fail-stop behavior approximations, both in the form of (a) hazardous state reachability control and (b) hazardous state maintenance. In safe states, the lift cabins are fixed at any floors. The system is allowed to reach any hazardous state when all relevant processors have successfully passed the start-up checks of inputs and monitored outputs and of appropriate communication status. The hazardous state maintenance includes operational checks and consistency checking for execution processors. To comply with safety-critical conception, all critical inputs and monitored outputs are doubled and compared. When the relevant signals differ, the respective lift is either forced (with the help of a subsystem active if the shaft controller is disconnected) to reach the nearest floor and to stay blocked.

The basic safety hard core includes mechanical, emergency brakes. Again, more detailed specification should reflect not only safety but also functionality with fault-tolerance support: also blocked lift is safe but useless. Hence, the above described configurations create only skeleton carrying common fault-tolerant techniques.

STATE 10
input=0
impuls=0
b=4
hold = FALSE

STATE 11
input=0
impuls=0
b=5

6. Application Patterns Reuse

The two case studies presented above demonstrate the possibility to reuse effectively substantial parts of the design dealing with petrol pumping station technology for a lift control technology project. While both cases belong to embedded control systems, their application domains and their technology principles differ: volume measurement and dosage control seems not too close to position measurement and control. Evidently, the similarity is observable by employment of application patterns hierarchy, see Table 1.

fault management based on fail-stop behavior approximations
two-level (event-driven/real-time) structure
incremental measurement
noise-tolerant impulse detector checking reversible counter /O watchdog

Table 1. Application patterns hierarchy.

The reused upper-layer application patterns presented include the automata-based descriptions of incremental measurement, two-level (event-driven/real-time) structure, and fault management stemming from fail-stop behavior approximations. The reused lower-layer application patterns are exemplified by the automata-based descriptions of noise-tolerant impulse detector, checking reversible counter, and I/O watchdog.
Clearly, while all introduced application patterns correspond to design patterns in the above-explained interpretation, the upper-layer application patterns can be related also to frameworks. Moreover, the presented collection of application patterns creates a base for a pattern language supporting reuse-oriented design process for real-time embedded systems.

7. Knowledge-Based Support

Industrial scale reusability requires a knowledge-based support, e.g. by case-based reasoning (see Kolodner, 1993), which differs from other rather traditional methods of Artificial Intelligence relying on case history. For a new problem, the case-based reasoning strives for a similar old solution. This old solution is chosen according to the correspondence of a new problem to some old problem that was successfully solved by this approach. Hence, previous significant cases are gathered and saved in a case library. Case-based reasoning stems from remembering a similar situation that worked in past. For software reuse, case-based reasoning utilization has been studied from several viewpoints as discussed e.g. by Henninger (1998), and by Soundarajan and Fridella (1998).

7.1 Case-Based Reasoning

The case-based reasoning method contains (1) elicitation, which means collecting those cases, and (2) implementation, which represents identification of important features for the case description consisting of values of those features. A case-based reasoning system can only be as good as its case library: only successful and sensibly selected old cases should be

stored in the case library. The description of a case should comprise the corresponding problem, solution of the problem, and any other information describing the context for which the solution can be reused. A feature-oriented approach is usually used for the case description.
Case library serves as the knowledge base of a case-based reasoning system. The system acquires knowledge from old cases while learning can be achieved accumulating new cases. While solving a new case, the most similar old case is retrieved from the case library. The suggested solution of the new case is generated in conformity with this retrieved old case. Search for the similar old case from the case library represents important operation of case-based reasoning paradigm.

7.2 Backing Techniques

Case-based reasoning relies on the idea that situations are mostly repeating during the life cycle of an applied system. Further, after some period, the most frequent situations can be identified and documented in the case library. So, the case library can usually cover common situations. However, it is impossible to start with case-based reasoning from the very beginning with an empty case library.
When relying on the case-based reasoning exclusively, also the opposite problem can be encountered: after some period the case library can become huge and very semi-redundant. Majority of registered cases represents clusters of very similar situations. Despite careful evaluation of cases before saving them in the case library, it is difficult to avoid this problem.
In an effort to solve these two problems, the case-base reasoning can be combined with some other paradigm to compensate these insufficiencies. Some level of rule-based support can partially cover these gaps with the help of rule-oriented knowledge; see (Sveda, Babka and Freeburn 1997). Rule-based reasoning should augment the case-based reasoning in the following situations:

- No suitable old solution can be found for a current situation in the case library and engineer hesitates about his own solution. So, rule-based module is activated. For a very restricted class of tasks, the rule-based module is capable to suggest its own solution. Once generated by this part of the framework, such a solution is then evaluated and tested more carefully. However, if the evaluation is positive, this case is later saved in the case library covering one of the gaps of the case-based module.
- Situations are similar but rarely identical. To fit closer the real situation, adaptation of the retrieved case is needed. The process of adaptation can be controlled by the rule-based paradigm, using adaptation procedures and heuristics in the form of implication. Sensibly chosen meta-rules can substantially improve the effectiveness of the system.

The problem of adaptation is quite serious when a cluster of similar cases is replaced by one representative only - to avoid a high level of redundancy of the case library. The level of similarity can be low for marginal cases of the cluster. So, adaptation is more important here.
Three main categories of rules can be found in the rule-based module:

- Several general heuristics can contribute to the optimal solution search of a very wide class of tasks.
- However, the dominant part of the knowledge support is based on a domain-specific rule.
- For a higher efficiency, metarules are also attached to the module. This "knowledge about knowledge" can considerably contribute to a smooth reasoning process.

While involvement of an expert is relatively low for case-based reasoning module, the rules are mainly based on expert's knowledge. However, some pieces of knowledge can also be obtained by data mining.

7.3 Similarity measurement of state-based specifications

Retrieval schemes proposed in the literature can be classed based upon the technique used to index cases during the search process (Atkinson, 1998): (a) classification-based schemes, which include keyword or feature-based controlled vocabularies; (b) structural schemes, which include signature or structural characteristics matching; and (c) behavioral schemes; which seek relevant cases by comparing input and output spaces of components.
The problem to be solved arises how to measure the similarity of state-based specifications for retrieval. Incidentally, similarity measurements for relational specifications have been resolved by Jilani, et al. (2001). The primary approach to the current application includes some equivalents of abstract data type signatures, belonging to structural schemes, and keywords, belonging to classification schemes. While the first alternative means for this purpose to quantify the similarity by the topological characteristics of associated finite automata state-transition graphs, such as number and placement of loops, the second one is based on a properly selected set of keywords with subsets identifying individual patterns. The current research task of our group focuses on experiments enabling to compare those alternatives.

8. Conclusions

The book chapter stems from the paper (Sveda, Vrba and Rysavy, 2007) and complements it by TLA specifications. The original contribution consists in proposal how to represent a system's formal specification as an application pattern structure of specification fragments. Next contribution deals with the approach how to measure similarity of formal specifications for retrieval in frame of case-based reasoning support. The above-presented case studies, which demonstrate the possibility to effectively reuse concrete application pattern structures, have been excerpted from two realized design cases.
The application patterns, originally introduced as "configurations" in the design project of petrol pumping station control technology based on multiple microcontrollers (Sveda, 1996), were effectively -- but without any dedicated development support -- reused for the project of lift control technology (Sveda, 1997). The notion of application pattern appeared for the first time in (Sveda, 2000) and was developed in (Sveda, 2006). By the way, the first experience of the authors with case-based reasoning support to knowledge preserving

development of an industrial application was published in (Sveda, Babka and Freeburn, 1997).

8. Acknowledgements

The research has been supported by the Czech Ministry of Education in the frame of Research Intentions MSM 0021630528: Security-Oriented Research in Information Technology and MSM 0021630503 MIKROSYN: New Trends in Microelectronic Systems and Nanotechnologies, and by the Grant Agency of the Czech Republic through the grants GACR 102/08/1429: Safety and Security of Networked Embedded System Applications and GACR 201/07/P544: Framework for the deductive analysis of embedded software.

10. References

Alexander, C. (1977) A Pattern Language: Towns / Buildings / Construction, Oxford University Press.

Alur, R. and T.A. Henzinger (1992) Logics and Models of Real Time: A Survey. In: (de Bakker, J.W., et al.) Real-Time: Theory in Practice. Springer-Verlag, LNCS 600, 74-106.

Arora, A. and S.S. Kulkarni (1998) Component Based Design of Multitolerant Systems. IEEE Transactions on Software Engineering, 24(1), 63-78.

Atkinson, S. (1998) Modeling Formal Integrated Component Retrieval. Proceedings of the Fifth International Conference on Software Reuse, IEEE Computer Society, Los Alamitos, California, 337-346.

Coad, P. and E.E. Yourdon (1990) Object-Oriented Analysis, Yourdon Press, New York.

Frakes, W.B. and K. Kang (2005) Software Reuse Research: Status and Future. IEEE Transactions on Software Engineering, 31(7), 529-536.

Gamma, E., R. Helm, R. Johnson and J. Vlissides (1995) Design Patterns -- Elements of Reusable Object-Oriented Software, Addison-Wesley.

Geppert, B. and F. Roessler (2001) The SDL Pattern Approach – A Reuse-driven SDL Design Methodology. Computer Networks, 35(6), Elsevier, 627-645.

Henninger, S. (1997) An Evolutionary Approach to Constructing Effective Software Reuse Repositories. Transactions on Software Engineering and Methodology, 6(2), 111-140.

Henninger, S. (1998) An Environment for Reusing Software Processes. Proceedings of the Fifth International Conference on Software Reuse, IEEE Computer Society, Los Alamitos, California, 103-112.

Holtzblatt, L.J., R.L. Piazza, H.B. Reubenstein, S.N. Roberts and D.R. Harris (1997) Design Recovery for Distributed Systems. IEEE Transactions on Software Engineering, 23(7), 461-472.

Jacobson, L. (1992) Object-Oriented Software Engineering: A User Case-Driven Approach, ACM Press.

Jilani, L.L., J. Deshamais and A. Mili (2001) Defining and Applying Measures of Distance Between Specifications. IEEE Transactions on Software Engineering, 27(8), 673-703.

Johnson, R.E. (1997) Frameworks = (Components + Patterns), Communications of the ACM, 40(10), 39 42.

Kolodner, J. (1993) Case-based Reasoning, Morgan Kaufmann, San Mateo, CA, USA.

Lamport, L. (1994) Temporal Logic of Actions. ACM Transactions on Programming Languages and Systems, 16(3) :872-923.

Lamport, L. (2002) Specifying Systems. Addison-Wesley, 2002.

Mili, R., A. Mili, and R.T. Mittermeir (1997) Storing and Retrieving Software Components: A Refinement Based System. IEEE Transactions on Software Engineering, 23(7), 445-460.

Sen, A. (1997) The Role of Opportunity in the Software Reuse Process. IEEE Transactions on Software Engineering, 23(7), 418-436.

Shaw, M. and D. Garlan (1996) Software Architecture, Prentice Hall.

Soundarajan, N. and S. Fridella (1998) Inheritance: From Code Reuse to Reasoning Reuse. Proceedings of the Fifth International Conference on Software Reuse, IEEE Computer Society, Los Alamitos, California, 206-215.

Sutcliffe, A. and N. Maiden (1998) The Domain Theory for Requirements Engineering. IEEE Transactions on Software Engineering, 24(3), 174-196.

Sveda, M. (1996) Embedded System Design: A Case Study. IEEE Proc. of International Conference and Workshop ECBS'96, IEEE Computer Society, Los Alamitos, California, 260-267.

Sveda, M., O. Babka and J. Freeburn (1997) Knowledge Preserving Development: A Case Study. IEEE Proc. of International Conference and Workshop ECBS'97, Monterey, California, IEEE Computer Society, Los Alamitos, California, 347-352.

Sveda, M. (1997) An Approach to Safety-Critical Systems Design. In: (Pichler, F., Moreno-Diaz, R.) Computer Aided Systems Theory, Springer-Verlag, LNCS 1333, 34-49.

Sveda, M. (2000) Patterns for Embedded Systems Design. In: (Pichler, F., Moreno-Diaz, R., Kopacek, P.) Computer Aided Systems Theory--EUROCAST'99, Springer-Verlag, LNCS 1798, 80-89.

Sveda, M. and R. Vrba (2006) Fault Maintenance in Embedded Systems Applications. Proceedings of the Engineering of Computer-Based Systems. Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics (ICINCO 2006), INSTICC, Setúbal, Portugal, 183-186.

Sveda, M., R. Vrba and O. Rysavy (2007) Pattern-Driven Reuse of Embedded Control Design -- Behavioral and Architectural Specifications in Embedded Control System Designs. Proceedings of Fourth International Conference on Informatics in Control, Automation and Robotics (ICINCO 2007), INSTICC, Angers, FR, pp. 244-248.

Turner, K.J. (1997) Relating Architecture and Specification. Computer Networks and ISDN Systems, 29(4), 437-456.

van Lamsweerde, A. and L. Willemet (1998) Inferring Declarative Requirements Specifications from Operational Scenarios. IEEE Transactions on Software Engineering, 24(12), 1089-1114.

Xinyao, Y., W. Ji, Z. Chaochen and P.K. Pandya (1994) Formal Design of Hybrid Systems. In: (Langmaack, H., W.P. de Roever and J. Vytopil) Formal Techniques in Real-Time and Fault-Tolerant Systems, Springer-Verlag, LNCS 863, 738-755.

Zaremski, A.M. and J.M. Wing (1997) Specification Matching of Software Components. ACM Trans. on Software Engineering and Methodology, 6(4), 333-369.

10

Optical Speed Measurement and applications

Tibor Takács, Viktor Kálmán, dr. László Vajta
Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
Hungary

1. Introduction

Mobile robot navigation is a well researched discipline, looking back to a relatively long history however it is still a rich, active area for research and development. The ultimate goal for robots and intelligent vehicles seems to be autonomous navigation in complex real life scenarios. In order to achieve higher levels of autonomy sophisticated sensors and a sound understanding of the robot and its interaction with the environment is needed. The tasks involved can be divided to two basic categories; internal tasks involve keeping track of internal dynamic parameters, like speed accelerations, internal states etc. On the other hand the vehicle needs to be aware of external factors like obstacles, points of interest, possible routes from a to b and the respective costs. This is generally called robotic mapping.
Intelligent Vehicle stands for a vehicle that senses the environment and provides information or control to assist the driver in optimal vehicle operation. Intelligent Vehicle systems operate at the tactical level of driving (throttle, brakes, steering) as contrasted with strategic decisions such as route choice, which might be supported by an on-board navigation system. (Bishop, 2005)
Optical sensors supply by far the most information and as greater and greater processing capabilities become readily available their use becomes more widespread. Many researchers and companies have made more or less successful attempts at creating optical sensors for speed measurement, however to the knowledge of the authors no accurate high speed solution exist at the low price range. The aim of this article is to introduce a novel method for optical speed measurement and put it into perspective by summarizing other navigation methods and reviewing recent related work and possible applications. Also an introduction to optical flow calculation is given and practical considerations on texture analysis and sensor parameters are discussed backed up with simulation results.

2. Motion measurement techniques

The development of navigation and dynamic sensors has always had a prominent place in mobile robotics research, as the key to accurate trajectory tracking and precise movements is the exact knowledge of the dynamic parameters of the mobile platform.

2.1 Incremental techniques

The first class of movement measurement methods - called incremental techniques - uses only sensors located on the mobile platform. In this case the actual position is calculated from the previous pose-information and the relative displacement measured by the motion-sensors. This navigation mode is often called "dead reckoning navigation".
On wheeled vehicles the most straightforward method is to measure wheel movements and calculate displacement accordingly. Rotations can be measured with optical encoders, proximity sensors and cog wheels or magnetic stripes with Hall sensors etc. Heading information can be derived from differential odometry, which means calculating the direction based on the distance difference travelled by the left side and right side wheels of the vehicle. In addition to optical encoders, potentiometers, synchros, resolvers and other sensors capable of measuring rotation can be used as odometry sensors. In the last few years in the wake of the invention of the optical mouse optical navigation found its way to mobile robotics, and other similar methods emerged in the transportation industry.
(Borenstein et al. 1996) describes thoroughly the various aspects related to odometry including typical error sources. Many of the systematic errors come from the errors of the kinematical model (wheelbase, wheel radius, misalignment etc.), some depend on the electronics (finite sampling rate, resolution). Non systematic errors occur when the wheel slips due to uneven surface or overacceleration etc. Some of these problems can be eliminated by the use of inertial methods, when accelerations and rotations are measured in three dimensions and integrated over time to derive position, speed and heading information. These methods are very sensitive to sensor quality since the double integration in position determination is prone to drift. (Mäkelä 2001) Due to accumulated errors measurements loose accuracy over time therefore position and speed information is usually periodically updated from an absolute source.

2.2 Absolute methods

In this case the actual position can be calculated without any previous information about the motion of the agent. The global pose is estimated directly (with one measurement) by means of external – artificial or natural – beacons which are totally independent from the platform.
Artificial beacons are objects which are placed at known positions with the main purpose of being used for pose determination. According to this definition, setting up the working environment for a robot using artificial beacons almost always requires a considerable amount of building and maintenance work. In addition, using active beacons requires a power source to be available for each beacon. GPS positioning is one of the major exceptions since the system is almost constantly available for outdoor navigation. Although the ultimate goal of research is to develop navigation systems which do not require beacons to be installed in the working environment, artificial beacons are still preferred in many cases. The reason is that artificial beacons can be designed to be detected very reliably which is not always the case when using natural beacons. For pose estimation in two dimensions, one can either measure the distances or bearings to at least three beacons and calculate the position and the heading by simple geometry. The calculation is called trilateration if it is based on known distances, and triangulation if it is based on bearings. Distance from the beacons can be measured by using several different methods like triangulation, time of flight, phase-shift measurement, frequency modulation, interferometry, swept focus, or

return signal intensity etc. The sensors used can be radio or laser based ultrasonic, or visual. The advantage of artificial beacon based systems is that they can be made very accurate as the environment is controlled, however this same controlled environment is the biggest disadvantage as it decreases flexibility. In certain cases system complexity becomes a problem, as was the case with GPS before mass production of receiver chips started. Artificial beacons are relatively simple to use and pose estimation based on them is straightforward and reliable. However, there are various applications where they can not be used. Natural beacons are objects or features of the environment of the robot that can be used for pose estimation. These beacons can be man made; natural means they were not built for navigation purposes. Navigation using a map is also related to natural beacons, when the map is matched to raw sensor data the whole environment can be considered as a beacon. (Mäkelä 2001)

2.3 Fusion

Through sensor fusion we may combine readings from different sensors, remove inconsistencies and combine the information into one coherent structure. This kind of processing is a fundamental feature of all animal and human navigation, where multiple information sources such as vision, hearing and balance are combined to determine position and plan a path to a goal. While the concept of data fusion is not new, the emergence of new sensors, advanced processing techniques, and improved processing hardware make real-time fusion of data increasingly possible (Bak 2000).

Incremental and absolute navigation techniques have somewhat complementing advantages and disadvantages so developers usually combine them to benefit from the advantages of both. In case of absolute techniques like for example GPS, the navigation system can directly calculate the absolute position of the platform therefore the error of the actual pose comes only from the current measurement and does not accumulate over time. But unfortunately in some cases these methods are unusable for direct positioning or speed measurement, for lack of signal or unacceptable latency. Incremental techniques are usually simpler and have greater data rates, but accumulate error over time.

In addition the ability of one isolated device to provide accurate reliable data of its environment is extremely limited as the environment is usually not very well defined in addition to sensors generally not being a very reliable interface. Sensor fusion seeks to overcome the drawbacks of current sensor technology by combining information from many independent sources of limited accuracy and reliability to give information of better quality. This makes the system less vulnerable to failures of a single component and generally provides more accurate information. In addition several readings from the same sensor are combined, making the system less sensitive to noise and anomalous observations.

Basically motivations for sensor fusion can be categorized into three groups (Bak 2000).

Complementary. Sensors are complementary when they do not depend on each other directly, but can be combined to give a more complete image of the environment.

Competitive. Sensors are competitive when they provide independent measurements of the same information. They provide increased reliability and accuracy. Because competitive sensors are redundant, inconsistencies may arise between sensor readings, and care must be taken to combine the data in a way that removes the uncertainties. When done properly, this kind of data fusion increases the robustness of the system.

Cooperative. Sensors are cooperative when they provide independent measurements, that when combined provide information that would not be available from any one sensor. Cooperative sensor networks take data from simple sensors and construct a new abstract sensor with data that does not resemble the readings from any one sensor.
The tool of choice for sensor fusion and vehicle state estimation has often been the Kalman filter. It is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. The classic linear Kalman filter is very attractive for low-cost applications due to its simplicity and low computational demand. Its main disadvantage is that it can only estimate linear plants. The advantage of the nonlinear Kalman filters is that they can directly estimate the vehicle dynamics (which are non-linear in most cases). Both the vehicle states and the sensor measurement equations can have nonlinear terms. This results in better estimation accuracy, over a wider range of operating conditions. The main disadvantage of the nonlinear Kalman filters is that the algorithms are more complex than the linear implementation, therefore requiring more computational resources. A wide body of literature is available on sensor fusion and Kalman filtering (Gustaffson 2007, Grewal et al. 2007, Bak 2000); however it is out of the scope of this article as in this chapter we only wanted to emphasize the importance of using measurements from a variety of different sensors, to achieve better accuracy and reliability.

2.4 Summary

As we can see automated navigation is a research area with a long history and many achievements, however there is still much to be done, both in the area of sensor technology and processing algorithms. In this section we have shown that for an accurate and reliable measurement it is necessary to use both absolute and incremental techniques. With the development of visual sensor technology and the availability of more computing power, the potential in visual sensors can be exploited. The authors feel that there is a need for new solutions in that field.

3. Introduction to optical motion measurement

As mentioned in the previous section incremental methods hold a prominent place in the area of motion measurement. The main purpose of this chapter is to introduce a cheap, easy to use, but accurate dead reckoning sensor-system based on visual information acquired from the ground.
First we review the techniques to extract motion from image sequences then we review related work in the field of motion measurement, both industrial and academic.

3.1 Optical flow

Visual movements are caused by the relative displacement of the observer (eye, camera) and the objects of the world. The measurement of these motions can be used in several areas of robotics, like object tracking and segmentation, navigation, and optical speed measurement etc.
Most techniques of visual motion measurement are based on the well researched discipline called “optical flow”. The basic idea is to compare consecutive images of a scene produced by camera and calculate a vector field for each image which shows the displacements of the

pixels to get the next image of the scene. This vector field is often called optical flow or optical flow field (Fig. 1.).

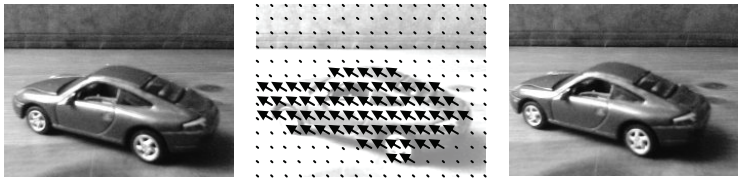


Fig. 1. The principle of optical flow

Since the first algorithm presented by Horn and Schunck (Horn & Schunck 1980) several techniques have been published to determine optical flow field. The common base of these techniques is the optical flow constraint (Horn & Schunck 1980) which presumes that the related points in the consecutive images have the same intensity value. Putting it in another way, a spatial point projected in the image plane has constant (time-invariant or projection-invariant) intensity value:

$$E\big(x(t+\Delta t), y(t+\Delta t), t+\Delta t\big) = E\big(x(t), y(t), t\big) \tag{1a}$$

$$\frac{dE}{dt} = 0 \tag{1b}$$

where $E(x,y,t)$ is the intensity of the (x,y) point in time t.
From a Taylor expansion of (1a) or from the dependencies between the total and partial derivative using (1b) the general form of constrains is easily derived:

$$\frac{\partial E}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial E}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial E}{\partial x} = 0 \tag{2}$$

where $\frac{\partial x}{\partial t}$ and $\frac{\partial y}{\partial t}$ refer to the coordinates of the velocity vector (the two unknowns of the equation), $\frac{\partial E}{\partial t}$ denotes time change of the intensity value , $\frac{\partial E}{\partial x}$ and $\frac{\partial E}{\partial y}$ denote components of the spatial gradient vector of intensity field.
This constraint is not sufficient to determine both components of the velocity vector, only the component in the direction of local gradient can be estimated. As a consequence, to compute the optical flow field it is necessary to introduce additional constraints.
The method of Horn and Schunck (Horn & Schunck 1980) starts from the observation that the points of the image plane do not move independently, if we view opaque objects of finite size undergoing rigid motion or deformation. Therefore the neighbouring points of moving objects have quite similar velocities and the vectors of the optical flow field vary smoothly almost everywhere. This smoothness constrain represent the following equation:

$$\min\left\{\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2\right\} \tag{3}$$

where u and v are the coordinates of the velocity vector.
Therefore the purpose is to determine a velocity vector field which minimizes the optical flow and the smoothness constrain together:

$$\min\left\{\iint_D\left(\frac{\partial E}{\partial x}\frac{\partial x}{\partial t}+\frac{\partial E}{\partial y}\frac{\partial y}{\partial t}+\frac{\partial E}{\partial x}\right)+\alpha^2\left(\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2+\left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2\right)dxdy\right\} \tag{4}$$

It seems that to compute individual velocity vectors it is necessary to take the whole image into consideration, because every vector depends on every other vector. Therefore this method is classified as a global technique (Beauchemin 1995).
Another approach presented by Lucas and Kanade assumes the velocities are the same in a small local area (local techniques) (Barron 1994). Therefore to calculate the velocity vector of a point it is possible to write more than one optical flow constraint because the points in the small region have the same velocity:

$$\mathbf{WAv}=\mathbf{Wb}\text{, where}$$
$$\mathbf{A}=\begin{bmatrix}\nabla E(x_1,y_1)\\ \nabla E(x_2,y_2)\\ \dots\\ \nabla E(x_{m\times m},y_{m\times m})\end{bmatrix};\quad \mathbf{v}=\begin{bmatrix}u\\ v\end{bmatrix};\quad \mathbf{b}=-\begin{bmatrix}E_t(x_1,y_1)\\ E_t(x_2,y_2)\\ \dots\\ E_t(x_{m\times m},y_{m\times m})\end{bmatrix};\quad \mathbf{W}=\begin{bmatrix}w_1 & 0 & 0 & 0\\ 0 & w_2 & 0 & 0\\ \dots & \dots & \dots & \dots\\ 0 & 0 & 0 & w_{m\times m}\end{bmatrix} \tag{5}$$

In this case the local region has m×m points and **W** is a weight matrix.
Because the equation system is over constrained and has no solution (in general) therefore the velocity estimates are computed by minimizing

$$\sum_{\mathbf{x}\in\Omega(m\times m)}\mathbf{W}^2(\mathbf{x})\left[\nabla E(\mathbf{x})\cdot\mathbf{v}+E_t(\mathbf{x})\right]^2 . \tag{6}$$

After using the least mean squares method, the solution is the following:

$$\mathbf{v}=\left(\mathbf{A}^T\mathbf{W}^2\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{b} \tag{7}$$

This method can only measure relatively small displacements therefore it is often called the iterative Lucas-Kanade algorithm.
The previous two algorithms are directly based on the gradients of scenes therefore these techniques are often called differential methods. Unfortunately these techniques suffer from

a serious disadvantage: accurate numerical differentiation is sometimes impractical because of small temporal support (only a few frames) or poor signal-to-noise ratio (Barron 1994). Region-based techniques define velocity as the shift **d** that yields the best fit between image regions at different times. Finding the best match amounts to maximizing (or minimizing) a similarity measure (over **d**), such as the sum of square distances (SSD), normalized cross correlation, etc. The optical flow constraint (namely the related points in consequent images have the same intensity value) can also be found in these techniques indirectly because the best match tries to minimize the difference of the intensity values of the points.
One of the well-known techniques belonging to this group is published by Anandan in 1987 (Barron 1994) which combines the Laplace-pyramid (to decrease the correlation between the pixels of the images) and the "coarse-to-fine" SSD matching method. Another region-based algorithm presented by Singh is also built on the SSD metric but uses three consequent images from the scene to calculate the displacement of the regions in the second image. Therefore the inaccuracy caused by noises and periodical texture is decreased (Beauchemin 1995).
A third class of optical flow techniques is based on the frequency domain of the image-sequence. One of the advantages brought by these methods is that motion-sensitive mechanisms operating on spatiotemporally oriented energy in Fourier space can estimate motion in image signals for which matching approaches would fail. A good example is the motion of random dot patterns, which are difficult to capture with region-based or differential methods, whereas, in frequency domain, the resulting oriented energy may be rapidly extracted to determine optical flow field (Beauchemin 1995).
These methods can be classified in two groups: energy-based approaches are built on the amplitude, phase-based techniques use the phases of the Fourier space to determine the optical flow field. The method developed by Heeger (Heeger 1988), formulated as a least square fit of spatiotemporal energy to a plane in frequency space belongs to the first group. An example for the phase-based methods is the algorithm by Fleet and Jepson (Fleet 1990).

3.2 Basics of optical navigation sensors

As previously discussed, the principle of optical flow can be used in several areas of robotics. For example it is possible to obtain distance information, avoid collision with obstacles, track patterns on the image etc. (Davies 2005). This chapter focuses on one application, motion measurement of a mobile platform based on optical flow field.
The working principle of an optical speed sensor is quite simple: an optical sensor (photo-detector, camera, etc.) is attached to the mobile platform facing the ground. From the periodically captured visual information it is possible to estimate the real velocity of the agent relative to the ground. (Fig. 1.)

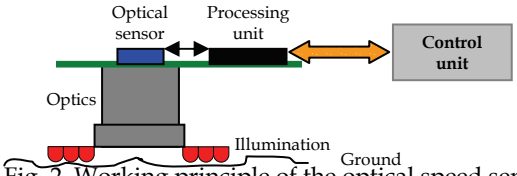


Fig. 2. Working principle of the optical speed sensor

3.3 Related work

Optical speed measurement is an emerging discipline, with existing commercial solutions and research activity in the academic sector, however many problems remain unsolved. As for commercial technologies the most widely known example is the optical mouse, which on the other hand has generated a fair amount of academic research. The optical mouse uses two distinct but essentially similar techniques for displacement calculation. The classical method uses LED illumination and relies on the micro texture of the surface. The more advanced method is laser speckle pattern technology. Laser speckle patterns can be observed when a rough surface (rough, relative to the wavelength) is illuminated with a coherent light and the interference of the reflected light waves creates a surface dependent random intensity map on the detector. When the detector is moved relative to the surface, the speckle pattern changes accordingly and optical flow can be calculated. The advantage over surface texture based methods is its accuracy and ability to function properly on relatively textureless smooth surfaces.

Frequency analysis is a less frequently used method. The light reflected from the surface travels through an optical grating, and is focused on a pair of photo-detectors. The surface elements, passing in front of the grating generate a certain signal frequency in the detectors depending on the sampling frequency, ground speed, grid graduation, ratio of the image, size of the surface elements and the size of the picture on the grating. The difference of the two signals is computed and the frequency of the difference signals corresponds to the true ground speed.

Indoor dead reckoning solutions for small mobile robots using optical mice were suggested by several authors (Palacin et al. 2006, Bonarini et al. 2004, Sorensen 2003) T.W. Ng investigated the usability and accuracy of optical mice for scientific measurements in several articles (Ng 2003, Ng & Ang 2004) with good results. It was found that the readings possessed low levels of error and high degrees of linearity. The mean square error for measurements in the x-axis increased significantly when the distance between the surface and the detector was increased possibly caused by the illumination direction of the mouse. Several researchers proposed the use of optical mice as a dead reckoning sensor for small indoor mobile robots in one and two sensor configurations. By using one sensor and kinematical constraints from the model of the platform, a slip free dead reckoning system can be realized. The kinematic constraint originates from the sensors inability to calculate rotation. By using two sensors the constraint can be removed and the measurements become independent of the platforms kinematics. Systematic errors originate from measurement errors, alignment errors and change of distance from the ground. (Bonarini et al. 2004) achieved results comparable to other dead reckoning systems up to a speed of 0.3 m/s by using the UMB benchmark test (Borenstein & Feng 1994). Sorensen found that the error of the two mice system was smallest when the sensors were as far as possible from the centre of rotation, and when good care were taken of maintaining constant height. He found that when these constraints were met, the system performed significantly better than other dead reckoning systems (Sorensen 2003). In their work Palacin et al. found that if measurements from an array of sensors were averaged the error became independent from the distance traveled. They also found that the sensor needed a different calibration when moving in an arc, possibly due to the sideways illumination used in computer mice. Another problem was the extreme height dependence of the sensor, which made it impossible for them to use it on carpet. They proposed a modified sensor for they found mice to be unfit for mobile robot

navigation (Palacin et al. 2006). The results of the authors of this article were similar to other researchers, they found that one way to make mouse sensors useful for navigation is to equip them with telecentric lens, to avoid magnification changes, to use homogeneous illumination, to avoid directional problems and to use two sensors to get rid of kinematic constraints. (Takács, Kálmán 2007) By using different magnification larger portions of the ground will be projected on the sensor making higher speeds possible, but this is limited by ground texture (section 4.4).
Mouse sensors are cheap and readily available and with certain modifications they can be used for low speed mobile robot dead reckoning. However they are limited by their low resolution and speed and their algorithm can only be changed by the factory.
Horn et al. aimed at developing a sensor system for automobiles. They used a fusion approach with two cameras and a Kalman filter. One of the cameras is a forward looking stereo camera to estimate yaw rate and forward velocity, the other camera is facing the ground and used to estimate two dimensional velocity. It was found that the camera facing the ground gave better results for lateral and longitudinal velocity than the stereo camera. The fusion approach provided good results even when one of the sensors was failing. The system was tested at slow (< 1 m/s) speeds on a towed cart in a lab (Horn et al. 2006).
Chhaniyara et al. followed a somewhat similar approach and used a matrix camera facing the ground to estimate speed over ground. They used a mechanism that moved the camera over sand and compared optical flow speed estimates with measurements from an encoder attached to the mechanism. They used Matlab and the Lukas and Kanade algorithm to compute optical flow. They obtained good results at low speeds (0-50 mm/s), however the suitability of the algorithm they used is questionable (Chhaniyara et al. 2008).
This technology has already found its way to the transportation industry as well. Corrsys - Datron has a one-of-a-kind optical speed sensor (Correvit 2001) used for testing the dynamics of passenger vehicles before mass production. The sensor is claimed to be working on any surface, including water and snow, but it is priced for the big automotive manufacturers. It uses the frequency analysis method. OSMES by Siemens is an optical speed measurement system for automated trains (Osmes 2004). It uses the principle of laser speckle interferometry mentioned above, and "looks" directly on the rails to measure the trains speed.
It is clear that much work has been done in the field of optical navigation however several issues remain open for research. Current industrial solutions are somewhat bulky and definitely not priced for the average mobile robot. Solutions by academic researchers have not matured to the level of really useful applications. Mouse chips are the mostly the sensors of choice. With some modifications their problems of ground distance, lighting and calibration can be helped, but their current speed and resolution is simply not enough for high speed (the order of ten m/s) applications.
More work in the area of texture analysis, optics design and image processing hardware is needed.

4. Optical correlation sensor

In this section we outline the basics of the motion measurement system proposed by the authors. First we introduce basic problems and some assumptions on which we based our investigations: the sensor is facing the ground, which is relatively flat, the field of view is constant due to telecentric optics and our sensor can only measure movements along a

straight line. Then we describe a multisensor setup that is capable of providing two dimensional velocity measurements independent of the platform. Finally we introduce a simulator which we created to verify the feasibility of different sensor embodiments, and the validity of our basic assumptions.

4.1 Basics

The distance between the sensor and the ground is continuously changing because of the macroscopic unevenness of the surface and the movement of the suspension of the platform causing variable field of view which can be a serious source of errors in speed measurement. The use of *telecentric optics* can eliminate this problem in a certain distance range as telecentric optics has constant magnification. In this range the field seen by the camera does not change its size. This approach does not solve the problem of the change in depth of field but blurriness only causes loss of accuracy while change of magnification causes miscalibration.

Two important parameters of the sensor are sampling rate and the size of the image seen by the camera (field of view). Frame rate and field of view determine the maximal measurable velocity of the platform. If the speed of the mobile agent is higher than this limit, there is no correlation between the consequent images as they do not overlap. This can cause false readings thus estimation of the real velocity is impossible. Fortunately a mobile robot or car has a well-determined limit for velocity therefore it is possible to calculate these parameters based on apriori information (Fig. 3.).

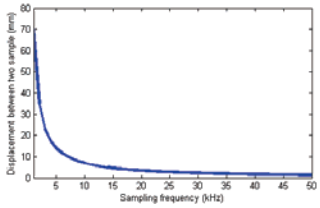


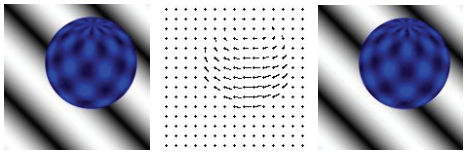
Fig. 3. Displacements between samples (sensor-speed: 70 m/s)

Let's illustrate the effects of limited dynamics with a simple example: the best racing cars in Formula-1 have 4-5 G deceleration at most. If we take a very modest estimation for frame rate like 100 Hz, then the difference between the two measured velocity-values is 0.05 m/s (0.18 km/h) in the worst case. Knowing this a plausibility check can be conducted and erroneous measurements caused by noise or "difficult" texture can be discarded. Also state variables of a vehicle such as speed cannot change abruptly, that is measurements in neighbouring sampling instants have to be close in value.

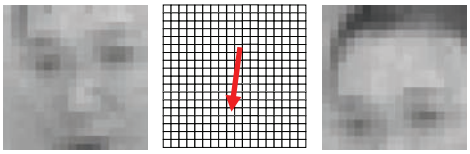
If the visual information about the motion comes from a camera and the displacement-estimations are calculated from the optical flow field of the captured scene, then some additional apriori information facilitates determination of the velocities. First of all it is important to determine what kind of displacement occurs in the image plane. Image movements can be categorized in two groups.

The first class, called *local image movement* belongs to the principle of optical flow presented in the previous section. Several objects of various sizes, velocities are moving in the visual field of the camera in different directions. Therefore the motion in the image plane can be

described with vectors corresponding to individual pixels. With this vector field the motion, shape etc. of the different objects can be estimated.



a.) Local image movements (http://of-eval.sourceforge.net/)



b.) Global image movements (Takács & Kálmán 2007)

Fig. 4. Local and global image movements

But in our case it is necessary to measure the relative movement of the camera to a single object, so the class of *global image movement* is introduced. In this case the motion of all pixels of the image corresponds to the relative movement of the camera and exactly one object with smooth surface covering the whole field of view. The constraint about covering the whole field of view causes a very close relationship between the motion vectors (they have the same length and direction; they can only change smoothly, etc.). This is the reason for the name "global". The condition of smooth surface guarantees that the distance between the camera and every point of the object are quite the same therefore the effect of motion parallax can not cause sharp differences between velocity vectors (Fig. 4.).
These two strict constraints of global image movement can be approximated by a camera facing the ground and taking pictures of it periodically. If a general mobile platform like a car or mobile robot is assumed, and the camera has a sufficiently high frame rate, it is possible to disregard the orientation change between successive images as the arc travelled can be approximated with a straight line, and therefore all vectors in the optical flow field have the same length and direction. The great advantage of this approach is that there is no need to determine the motion of each pixel because they are all the same; therefore the calculation of optical flow is simpler, faster and more accurate.
From the field calculation techniques presented previously region-based methods fit this application best. In this case the window of the region contains the whole image and the comparison is between the two consecutive images. Other solutions which calculate the velocity vectors in pixel level and try to determine the camera movement from the heterogeneous motion vector field, avoid the use of this very important piece of apriori information. Therefore the application of these techniques in optical speed measurement with a camera facing the ground has marginal significance.

4.2 Measurements with multiple sensors

In case of using only one sensor - unless it is placed in the point of interest - the displacement measured needs to be transformed to platform coordinates. Additionally - unless kinematics of the platform is taken into account - rotation information is lost. In the extreme case, if the origin of the rotation is in the centre of the sensor the angle of rotation can not be estimated because the sensor does not measure any displacement.
In consequence it is necessary to apply multiple sensors and calculate the displacement from their geometry.

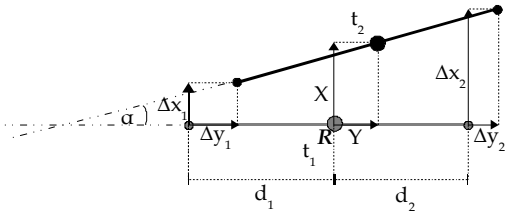


Fig. 5. Multiple sensor displacement model

Figure 5. shows a possible case of sensor placement. As mentioned above the orientation of the coordinate system is constant between two sampling instances because we approximate the movement of the sensors with a straight line. This introduces a small quantization error which can be modelled as noise. d_1 and d_2 are the distances of the sensors from the reference point ***R***, Δx_1, Δy_1, Δx_2 and Δy_2 are the displacement values measured by the sensors 1 and 2 respectively. From this model the displacement and orientation change of the reference point X, Y and α can be easy derived:

$$X = \frac{d_1\Delta x_2 + d_2\Delta x_1}{d_1 + d_2};\quad Y = \frac{d_1\Delta y_2 + d_2\Delta y_1}{d_1 + d_2};\quad \alpha = \arcsin\left(\frac{\Delta x_2 - \Delta x_1}{d_1 + d_2}\right); \tag{8}$$

Displacement of any other point of the platform can be calculated with a simple geometrical transformation.

If the reference point is in the origin of sensor #1 (namely $d_1 = 0$), then the equations in (8) became simpler: $X = \Delta x_1$, $Y = \Delta y1$ and $\alpha = \arcsin\left(\frac{\Delta x_2 - \Delta x_1}{d_2}\right)$. This shows that the system is over determined the y component of the second sensor is not needed.

The equations show another very important property, in particular, that the calculation of the motion information does not depend on the kinematical model of the platform. This is one of the greatest advantages of the method. This property has been noted by others too. (Palacin et al. 2006, Bonarini et al. 2004, Sorensen 2003)

Another very important question is the connection between the distance of the sensors and the accuracy of the measurement. From the equations (8) it is clear that with greater sensor distance higher accuracy can be achieved. The distance required for a given angular resolution can be reduced by increasing the sampling rate and/or resolution as smaller displacements will be detectable.

In real applications parallel mounting of the sensors is not always guaranteed. This alignment error introduces systematic errors in odometry that can be eliminated by calibration as described in the literature (Borenstein 1996).

4.3 Advanced experiments

In the first stage of our experiments a mouse chip was used as image sensor. It quickly became clear that mouse chips are not fit for the purpose of high speed velocity

measurement as they lack both the necessary resolution and speed. This is similar to what other experimenters found.
Our basic assumptions to start with were the following: low speed displacement measurement is most accurate if we look at a relatively small area on the ground with a high resolution image sensor to detect small movements accurately. But for high speed measurements we need to look at a bigger area to ensure that the consecutive images overlap. Also sampling rates need to be higher, but resolution can be lowered to achieve the same relative error rates. This contradiction can be resolved by using a variable image size by changing the magnification rate of the optics. Unfortunately this raises cost, causes calibration and accuracy problems, so we need to assume it to be constant. Therefore it is necessary to find a compromise to be able to measure the whole speed range.
Matrix cameras are very practical for the purpose of movement measurement as two dimensional displacement and even rotation can be calculated from the images (if it is necessary). However they have certain disadvantages. With commercial matrix cameras high (several kHz) sampling rates are currently unachievable and the data rate at high speeds makes processing challenging. We claim that accurate two dimensional measurements can be made with line-scan cameras. The most important advantages of this type of camera in respect of displacement measurement are relatively high – several mega pixels - resolution in one dimension, frame rates at the order of 10 to 100 kHz and relatively low prices.
In this case the field of view is projected to a single line of detectors therefore line-scan cameras with appropriate optics (e.g. cylindrical lens) or with wide pixels can realize an integrating effect (Fig. 6.). This property is very important and useful for our purposes (see details later).

Fig. 6. Projection of matrix and line-scan camera (illustration)

Naturally a line-scan camera can measure the motion only in the direction parallel with its main axis. If two cameras are used perpendicular to each other, two dimensional motion can be detected. Inherently the motion component orthogonal to the main axis causes errors in the calculation of parallel displacement (Fig 7.).

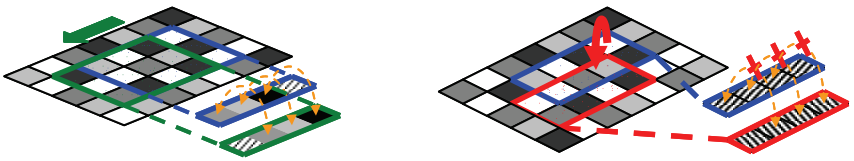
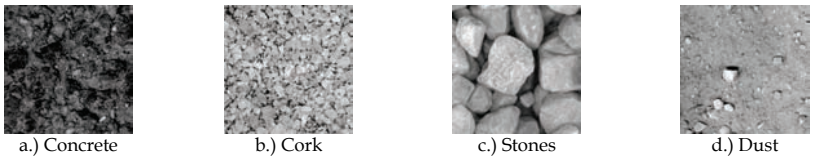


Fig. 7. Illustration of the problem of sideways motion

This error can not be totally eliminated but it is possible to decrease this effect with high frame rate and larger field of view of the camera. If the sampling frequency is high (which is easy to reach with line-scan cameras) then the perpendicular displacement between two consecutive images can be small enough that they will be taken of essentially the same texture element, making correlation in the parallel direction possible. This is of course a texture dependent effect and has to be investigated with texture analysis. Also this effect can be enhanced by widening the field of view of the detector, i.e. by integrating the image in the orthogonal direction. By doing this the images can overlap, giving higher correlation values. (More on this in the experimental results.)
A negative effect of this method is that the integration of the wider field of view can cause contrast in the image to reduce to the level of noise or completely disappear, making estimation of displacement in the parallel direction impossible. For that reason great care should be taken in the choice of pixel shapes and field of view of the line detector.
In order to find the sensor parameters we created an experimental computer program with a simple camera model that simulates a moving line-scan camera over a virtual surface. These surfaces are represented by simple greyscale images taken of real textures (e. g. concrete, soil, stone, PVC etc.) with very high resolution (Fig. 8.). Available, widely used texture databases were not fit for our purposes for they had insufficient resolution and were not calibrated for size. Our pictures were taken with an upside down flatbed scanner to ensure uniform conditions. By using this method we created a controllable environment, light, distance, image size, pixel/mm ratio and viewing angle were equivalent for all pictures taken. These images have different properties in respect to texture-size, contrast and brightness.

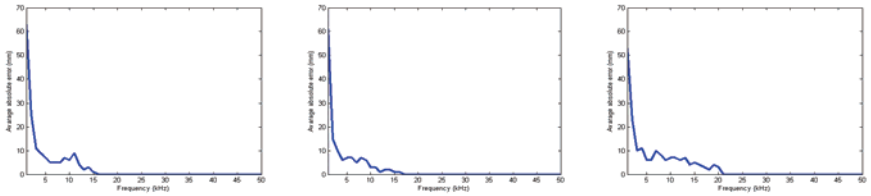


a.) Concrete b.) Cork c.) Stones d.) Dust

Fig. 8. Some of the ground textures used in the experiment

The virtual camera implemented in the simulator has several adjustable parameters: movement speed, frame rate, field of view in two dimensions, signal to noise ratio and resolution. Using the virtual surfaces and line-scan cameras it is possible to simulate different movement scenarios. The maximum virtual speed is over 100 m/s, the limit of frame rate is higher than 100 kHz and the size of field of view is greater than 100 mm in both directions.
The simulator – written in Matlab - works the following way: the ground is represented by a high resolution image, an image detector is chosen by defining an n X m resolution and a pixel size. Then the field of view is determined: a k X l mm rectangle. The image on the detector is created by resampling a k X l mm portion of the high resolution image onto the n X m detector image with additional white noise with an expected value of 0 and a standard deviation of choice. The consecutive image is chosen by translating the k X l mm window on the ground image with a certain amount of pixels according to the pre-defined movement speed, frame rate and direction. Three directions can be chosen, zero, 45, and 90 degrees. The two neighbouring images are then compared according to a distance measure of choice

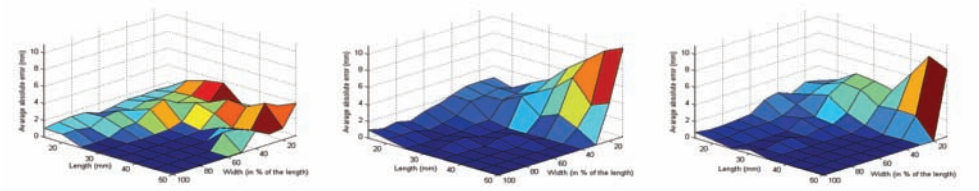
such as correlation, least squares, Manhattan and cosine distances. As the exact distance in pixels is known the error of the measurement can be obtained easily.
The purpose of the simulator was to determine the feasibility of using line-scan cameras for optical velocity measurement. Because of the huge size of the parameter space and various requirements and conditions it is hard to determine the exact properties of the sensor immediately. In this chapter we show the most important results and experiments which are available at this phase of our research. All the following tests were conducted with the simulated velocity of 100 m/s and the direction of movement was 45 degrees.
The first interesting property is the connection between measurement accuracy and the frame rate of the camera. The sampling frequency determines the amount of light needed, the maximal processing time and the quality (and price) of the camera.
Figure 9. shows the measurement error versus the frame rate. The simulated velocity of the platform is 100 m/s and the direction of movement was 45 degrees. This sampling frequency range is usual for common line-scan cameras.



a.) Stone b.) Concrete c.) Cork

Fig. 9. Error versus frequency for different textures

From the figure the tendency can be seen that for "bigger" texture size the errors converge to zero at smaller frequencies, however more experiments are needed with different textures to verify this assumption. The idea is that with bigger texture larger sideways movements (lower frame rates) are tolerated as the texture elements correlate for a greater distance. At this point no quantitative measure was used for texture size, "bigger" or "smaller" was determined by subjective methods.
A very important parameter of the sensor is the field of view and the shape factor of the optics. As we modeled our imaging system with rectangular frames a practical shape factor choice is width/length of the field of view in %. A sensor with a small field of view is more compact and cheaper. If it is possible to avoid the use of cylindrical lens the optics will be simpler and easier to develop. Therefore another purpose of the tests was to obtain the connection between the accuracy and the field of view.



a.) Stone b.) Concrete c.) Cork

Fig. 10. Error surfaces as a function of field of view ratio (width/length) @ 15kfps

Figure 10. shows the error surface as a function of the two dimensions of the field of view. The main axis of the line detector is called length; width of the sensor is scaled in percentage of the length, 100% meaning a square field of vision. It is clear from the images that increasing the length alone does not decrease the error, image ratios of 40% or larger are needed to obtain acceptable measurements. However increasing frame rate allows us to choose ratios around 20% which is demonstrated on figure 11. These results seem logical as an increase in frame rate means smaller displacements between frames making correlation possible for narrower images too.

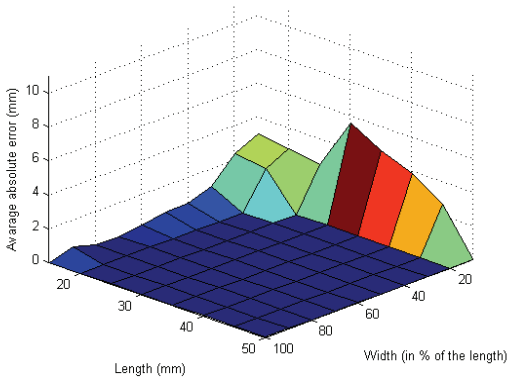
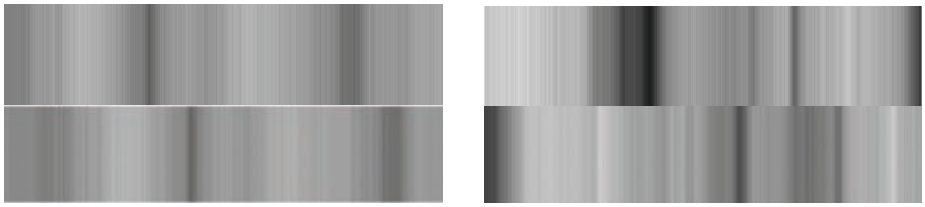


Fig. 11. The effect of increased frame rate Cork @ 30kfps

As mentioned earlier widening the field of view has a negative effect on contrast. This can be seen on figure 11.



a) Consecutive images with wider field of view b) Consecutive images with narrow field of view

Fig. 12. The effect of field of view shape factor

On figure 12. a.) a wider field of view was used than on b.). Both image pairs are one sampling period apart taken on the same surface (Stone) at the same speed, and frame rate. It is clearly visible that a.) has less contrast, due to the integration effect, but the samples correlate, b.) on the other hand has more contrast but a lower cross correlation value. It is important to note here that increasing image width much further leads to total loss of contrast making measurements impossible. However on this particular surface that limit is higher than 100% width/length, which seems impractical anyway.

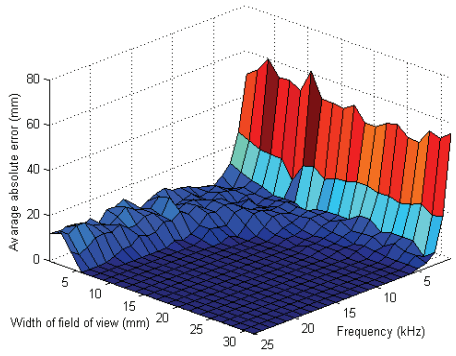


Fig. 13. Error versus frame rate and width of image @ fixed 50mm length (Cork)

Figure 13. shows that we can not reach zero error just by increasing the frame rate, however by increasing the field width we can obtain good results at relatively low frame rates for the given texture.

The experiments conducted with the simulator show that using a line-scan camera for optical speed measurements is a viable idea. Practical parameter choices have lead to exact displacement calculations for most of the investigated textures in the presence of simulated noise. To be fair we have to mention that there were a few textureless surfaces (e.g. plastic tabletop) for which no amount of tuning made correlation work. This shows that experiments with different lighting methods need to be done to be more independent from color based texture. Our initial tests justify further research to find the optimum of the parameters of our sensor. Optimization methods should be used to determine the most cost effective solution in terms of frame rate, resolution and optics. Future work will include hardware implementation of the sensor and the development of texture analysis methods.
(You can read more information about this research and development project on the website http://3dmr .iit.bme.hu/opticalflow)

4.4 Texture analysis

For purely image based systems the importance of texture can not be overlooked as it affects sensor qualities like precision and resolution and determines the necessary criteria the sensor parameters have to meet, like sampling frequency, magnification, resolution, pixel size and shape. Sampling frequency and magnification affect the maximal speed measurable as the consequent images have to overlap. Texture size might be the most important feature of a given texture as it determines the size of the area the sensor needs to look at i.e. the magnification. Texture size can be hard to define as it depends on how closely we look at a given surface. If we look at a gravel road the small stones form the basis of the texture or, if we look closer the rough surfaces on the stones do. The latter might be a better option as micro texture is usually available on otherwise homogeneous surfaces – laser speckle correlation takes advantage of this – but if we use a small image with great magnification, we limit the maximal speed measurable as for a given frame rate we might not get

overlapping images. Several methods exist in the literature to determine texture size. One of the main applications is grain size measurement in chemical or other industrial processes, and some of the methods can be readily adapted for our purposes. For example asphalt and gravel textures can be modelled by a mixture of different sized grains. Lepistö et al. used a histogram based quantifier. They calculated the distances of maximal intensity differences for a given direction on a greyscale image and took the center of gravity of the resulting distance histogram as a good measure to predict average grain size. This method is computationally cheap but suffers from inaccuracies in the presence of noise and areas without grain (Lepistö et al. 2007). Another popular method is to binarize the image and use segmentation on the resulting black and white shapes to determine average particle size (Pi & Zhang 2005), however the result depends greatly on the choice of the binarizing level.
The theoretical limit of geometrical precision of movement calculation also depends on the texture, only the presence of sufficient high frequency components will guarantee precise correlation (Förstner 1982). Sampling frequency and resolution of the instrument has to be chosen to capture these high frequency components. The highest frequency of interest can be determined from the energy spectrum of the image. According to Förstner precision can be estimated by examining the curvature (2^{nd} derivative) of the cross correlation function in the neighborhood of the maximum.
Some of the problems associated with textures can be eliminated by changing the illumination. Optical mice illuminate the surface at a low angle creating long shadows of miniature surface irregularities, making measurement possible on surfaces of homogeneous colour. Laser speckle interferometry – known since the seventies - offers another alternative: In laser speckle correlation the object is illuminated with laser light so that its image is modulated by a fine, high-contrast speckle pattern that moves with the surface. This movement is tracked by cross-correlation of the intensity distribution in successive images (Feiel & Wilksch 2000). This method offers unprecedented resolution and total independence from surface texture. A serious drawback of both the above mentioned illumination methods is that both the shadows created by sideways illumination and the speckle pattern changes with the distance between the light source and the object. This effect makes displacement measurement hard, if not impossible.
In the field of texture analysis many questions remain open such as a quantitative relation between texture and detector parameters and a good measure of texture frequency that determines resolution parameters. The problem of illumination also offers itself to application oriented research.

5. Applications

There are many possible applications of true ground speed measurement. In the following we will outline some of the areas that the authors think are most important.

5.1 Slip measurement

When a wheel contacts the ground usually two kinds of slip can occur, lateral and longitudinal. Longitudinal slip is the difference between the velocity of the centre of the wheel and the velocity of the circumference of the wheel. The difference is usually caused by acceleration or deceleration when there is not enough friction between the wheel and the ground, so slip is heavily dependent on the friction coefficient.

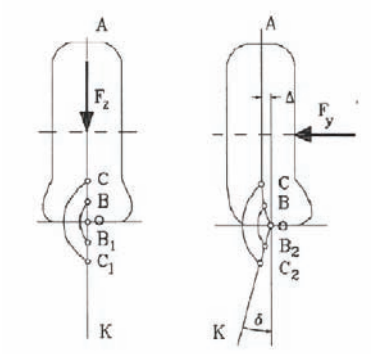


Fig. 14. Lateral slip

Lateral slip occurs if the wheel's angular displacement differs from the path the tire is following. It is caused by wheel deformation. When lateral forces act on the wheel - cornering, driving on a slope or in crosswind – the wheel changes its shape and starts to "crawl" to the side. The angle that corresponds to the rate of sideway movement is called the slip angle. It should be noted that the slip angle is not the same as the steering angle. Knowledge of the sideslip of a vehicle is indispensable for the exact description of its dynamics and kinematics.

The importance of longitudinal slip: in agricultural and off road applications it is considered as a predictor of the tractive efficiency of a given wheel set-up. The other main application is tire road friction estimation which is a discipline with a long history (Gustafsson 1997). Many researchers have worked on the problem of determining tire- road friction on line, (Müller et al. 2003) give a good overview on the literature and propose a method, based on slip curve steepness to estimate maximal available friction. (Miller et al. 2001) also conducted research on slip estimation using GPS and wheel speed sensors, their results show that tire slip and wheel radius can be estimated with good accuracy from these two measurements. By using optical speed sensors the disadvantages of GPS such as latency and limited reception could be eliminated and on-line measurements are possible.

Off road and agricultural applications could greatly benefit from the use of a simple non contact speed sensor, as it could provide speed over ground measurements on rough or slippery surfaces. (Lindgren et al. 2002) describe an odometry model for autonomous agricultural vehicles in which a relation between torque and slip is established. To estimate slip they used laser rangefinders and reflective beacons to obtain ground truth velocity measurements, limiting their application to level surfaces and a calibrated environment. By the use of an optical speed sensor their method can be extended and on line measurements without the use of external beacons can be conducted.

(Hutangkabodee et al. 2008) present a method to identify the set of soil parameters required to predict drawbar pull and wheel drive torque from measurements of slip, sinkage, and drawbar pull for a wheeled vehicle traversing unknown terrain. Knowledge of the terrain characteristics helps the driver to have a better control of the vehicle. From wheel-terrain interaction dynamics, it is seen that soil parameters play a vital role in determining vehicle drawbar pull which can, in turn, be utilized for developing traversability prediction criteria and traction control algorithms.

The importance of lateral slip: Vehicle safety is highly active research topic as car manufacturers keep pushing the boundaries of intelligent vehicle systems. Governments world wide have started programs to promote road safety to lessen the effect of traffic accidents; probably the most ambitious is vision zero from Sweden, trying to achieve zero fatalities on the roads. (Bishop 2005) Vehicle stability systems are among the most researched topics as they provide superior handling in extreme conditions. Modern ESP -s use yaw rate and steering angle as their main input, but in certain cases these are insufficient for correct intervention and knowledge of the slip angle of the vehicle is necessary. A good example would be a cornering vehicle, which is sliding at the same time. Its yaw rate might be considered adequate to its steering angle but it might still leave the road due to its sideways movement. The importance of sideslip is twofold; it allows better description of vehicle dynamics and on the other hand it plays an important role in wheel-road interaction, allowing us to determine friction or cornering forces. (Bevly et al. 2001) proposes a method to integrate inertial sensors with GPS to estimate sideslip angle and cornering stiffness. Sensor fusion is essential to solve this problem since GPS sensors have high noise and low sample rates, inertial sensors are fast and accurate but their measurements need to be integrated leading to unbounded errors. Using an optical speed sensor in the fusion system could provide a fast low noise speed estimate. Errors caused by textureless surfaces or other anomalies could be corrected by inertial sensors.
Big car manufacturers have been working on projects to estimate and use the sideslip angle in their stability systems (Nishio et al. 2001).
By measuring the sideslip angle at individual wheels important parameters of the suspension and wheel alignment can be determined. For example at high slip angles, the rear of the tire footprint actually slides laterally along the surface of the road, which contributes to less capacity for lateral force and reduces the stabilizing self-aligning torque. It may be important to realize that when not completely sliding, the lateral force is not dependent on the coefficient of friction, although this provides the upper limit; instead, it depends on the foundation stiffness. An alternate way to look at this is to say that the lateral force is not dependent on coefficient of friction until the tire has “broken away”, indicating a large slip angle (Smith 2003).

5.2 Mapping

Robotic mapping is an active research area with lots of problems open to research. Thrun conducted a survey of the major mapping methods used by researchers in the last decade (Thrun 2002). Although the survey was six years old when this article was written its statements and general assumptions were still valid.
Intelligent mobile robots navigate around in their environment by gathering information about their surroundings. The most common approach is to use ranging sensors mounted on the robot to form occupancy grids or equivalent. Other approaches avoid this metric division of space and favour topological mapping. By combining these mapping techniques it is possible to form a hierarchical map that has the advantages of both methods while some of the disadvantages can be avoided (Thrun, 1998). The map categories proposed by Dudec and Jenkin overlap with the ones mentioned above, but they are somewhat more differentiated (Dudec & Jenkin 2000):
Sensorial. Raw data signals or signal-domain transformations of these signals.
Geometric. Two- or three-dimensional objects inferred from sensor data.

Local relational. Functional, structural or semantic relations between geometric objects that are near one another
Topological. The large-scale relational links that connect objects and locations across the environment as a whole (for example, a subway map).
Semantic. Functional labels associated with the constituents of the map.
Occupancy grids classify the individual cells based on range data and possibly other features such as colour or surface texture or variation. This becomes very important in outdoor mobile robotics when the robot needs to distinguish between real obstacles and traversable terrain. An extreme case is given by navigation in a field of tall grass. The elevation map will represent the scene as a basically horizontal surface above the ground level; that is, as a big obstacle in front of the vehicle. It is apparent that only by integrating the geometry description with terrain cover characterization will a robot be able to navigate in such critical conditions (Belluta, 2000). This is the case when semantic information would prove useful. Topological maps describe the world in terms of connections between regions. This is usually enough indoors, or in well structured environments, but when travelling through more complex terrain a different representation might be necessary. For example a sloping gravel road or sand dune may only be traversable at a certain speed or only one way, up or downwards. By applying information from the inertial navigational unit, such as slope angle, wheel slippage, actual movement versus desired movement, these characteristics can be learned (or used from apriori information) and the connections of the topological graph can be updated accordingly. Terrain characteristics (and those of our vehicle) determine the maximum safe speed, braking distance curve radius at a given speed, climbing manoeuvres etc. It is obvious that the more information we have about a certain region we are planning to travel through, the more driving efficiency we can achieve, as it is generally unsafe to drive at high speed through bumpy terrain or make fast turns on a slippery surface. By incorporating the data from the navigational unit into the world map, we can associate driving guidelines to a given map segment. Also on the higher, topological or relational level - using apriori information - we can identify the type of the terrain for a given point of our topological graph, as office environment, forest, urban area, desert etc. By doing so, we narrow down our choices when making decisions about terrain coverage. For example it is unlikely to encounter sand, water or foliage in an office environment. If we know the type of terrain ahead we can make a more accurate estimate of the drivability of the area thus increasing driving efficiency. In this section a hierarchical map making method was proposed which uses data from a multi-sensor navigation unit that supplies information about vehicle dynamics. This unit heavily relies on the optical correlation sensor described in the preceding sections. By measuring wheel slip and vehicle slip angle we are able to associate drivability guidelines such as safe speed, friction coefficient, minimal driving speed etc. to a given map segment or type of terrain. A higher level of environment recognition was also proposed: based on apriori information, or sensor data the vehicles control system decides the type of environment (e.g. office, forest, desert) the robot traverses at the time, and changes the probability of terrain types, characteristic of the type of environment, thus simplifying terrain classification. (Takács & Kálmán 2007)

6. Conclusion

An overview on optical speed measurement was presented with a special focus on measurement methods with an image sensor facing the ground. The second section gave a

short overview on motion measurement in general, section 3 described the basics of optical flow and image correlation, and related work in the field of optical motion measurement. The following section described the foundations of a high speed optical correlation sensor based on line-scan cameras. Special attention was given to texture properties, possible problems and their solutions. A simulator was created and several experiments were conducted to verify the assumptions made earlier. The results show that it is possible to use a line-scan camera for one dimensional speed measurement and a range of parameters was defined for independent measurements in orthogonal directions. The last chapter gave a brief overview on possible applications of the sensor. Possible applications include slip free platform independent dead reckoning sensor for mobile robots, slip measurement for vehicles, that can be used for on line friction estimation, wheel geometry alignment and stability systems. A method was proposed to incorporate slip and handling data into maps created by autonomous agents to enhance driving efficiency and facilitate cost effective route planning.

In the future the authors plan to do extensive testing on the simulator as the tests conducted were not exhaustive in the sense of optimizing sensor parameters, and also plan to create a working prototype to address real world problems absent from the simulation and solve them.

Problems to be solved, possible areas of contribution: to create a truly useful sensor for mobile platforms further research and development is needed. One group of problems come from environmental effects. If it was to be mounted on an automobile the sensor has to operate in an environment with constant vibration, high temperature changes and lots of dirt. To counter the effects of dirt several methods might be used: mounting inside a protective tube, blowing air away from the sensor, using protective water repellent coating on the housing, shaking the lens with high frequency to prevent adhesion of dirt and using special image processing techniques to achieve graceful degradation of performance, however viability of these methods is still to be verified.

The effects of highly reflective surfaces such as ice snow and water, and the effects of fog and rain need to be investigated too.

In section 4.4 several aspects of texture processing have been mentioned, however there is still a need for a qualitative measure for texture that shows how good a texture is for movement detection. This measure could incorporate factors such as contrast, texture size and density, number of edges per unit of area and spectral information.

7. References

Bak, T. (2000) Lecture Notes - Estimation and Sensor Information Fusion, *Aalborg University Department of Control Engineering*, Denmark, November 14, 2000

Barron, J. L.; Fleet, D. J. & Beauchemin, S. S. (1994). Performance of optical flow techniques, *International Journal of Computer Vision*, Volume 12, Issue 1, (February 1994), 43-77, ISSN 0920-5691

Beauchemin, S. S. & Barron, J. L. (1995). The computation of optical flow. *ACM Computing Survey (CSUR)*, Volume 27, Issue 3, (September 1995), 433-466, ISSN 0360-0300

Belluta, P., Manduchi, R., Matthies, L., Owens, K., Rankin, A., (2000) Terrain Perception for Demo III. *Proceedings of the Intelligent Vehicles Symposium*. Dearborn, Michigan, USA, 2000

Bevly, D. M.; Sheridan, R. & Gerdes, J. C. (2001) Integrating INS Sensors with GPS Velocity Measurements for Continuous Estimation of Vehicle Sideslip and Tire Comering Stiffness, *Proceedings of the American Control Conference*, Arlington, VA, USA, June 25-27, 2001

Bishop, R. (2005) *Intelligent Vehicle Technology and Trends* Artech house, ISBN 1-58053-911-4, Norwood, MA, USA

Bonarini , A.; Matteucci, M. & Restelli, M. (2004). **A** Kinematic-independent Dead-reckoning Sensor for Indoor Mobile Robotics, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3750-3755, 0-7803-8463-61, Sendai, Japan, September 28 - October 2, 2004

Borenstein, J.; & Feng, L. (1994) UMBmark - a method for measuring comparing and eliminating dead-reckoning errors in mobile robots. *Proceedings of SPIE Conference on Mobile Robots,* Philadelphia, October 1994.

Borenstein, J. & Everett, H. R. & Feng L. (1996) *"Where am I?" Sensors and methods for mobile robot positioning* University of Michigan, USA

Chhaniyara, S.; Bunnun,P.; Seneviratne, L. D. & Althoefer, K. (2008.) Optical Flow Algorithm for Velocity Estimation of Ground Vehicles: A Feasibility Study, *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1, No. 1, (March 2008.) 246-268

Correvit(R)-SL (2001) *Non-Contact Optical Sensor for slip free measurement of longitudinal and transversal dynamics*, Corrsys-Datron Sensorsysteme GmbH, 2001. [Online]. Available: www.corrsys-datron.com

Davies, E. R. (2005). *Machine Vision (Theory, Algorithms, Practicalities), 3rd Edition*, Morgan Elsevier, ISBN 0-12-206093-8, UK-London

Dudec, G., Jenkin, M. (2000) *Computational Principles of Mobile Robotics.* Cambridge University Press, 280 p.

Feiel, R. & Wilksch, P. (2000) High-resolution laser speckle correlation for displacement and strain measurement, *In APPLIED OPTICS* , Vol. 39, No. 1, pp. 54.-60., 1 January 2000

Fleet, D. J., Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, Volume 5, Number 1, (August 1990), 77-104, ISSN: 0920-5691 (Print) 1573-1405 (Online)

Förstner, W. (1982) On the geometric precision of digital correlation, *International Archives of Photogrammetry and Remote Sensing*, vol. 24, no. 3, pp. 176–189, 1982.

Grewal, M. S.; Weill, L. R. & Andrews, A. P. (2007) *Global Positioning Systems, Inertial Navigation, and Integration,* John Wiley & Sons, Inc., ISBN: 978-0-470-04190-1,

Gustafsson, F. (1997) Slip-based estimation of tire – road friction. *In: Automatica*, 33(6), pp. 1087–1099, 1997.

Gustafsson, F. (2007) Sensor fusion, Compendium for the Sensor Fusion course 2007, pp. 323, *Linköping University Sweden* [Online] *http://www.control.isy.liu.se*

Heeger, D. J. (1988). Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, Volume 1, Number 4, (January 1988), 279-302, ISSN 0920-5691 (Print) 1573-1405 (Online)

Horn, B.K.P. & Schunck, B. G. (1980), Determining optical flow. *Artifical Intelligence Memo* 572. Massachusetts Institue of Technology, 1980.

Horn, J.; Bachmann, A. & Dang, T. (2006) A Fusion Approach for Image-Based Measurement of Speed Over Ground, *Proceedings* of *International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 261-266, September 3-6, 2006, Heidelberg, Germany

Hutangkabodee, S.; Zweiri, Y.; Seneviratne, L. D.; & Althoefer, K., (2008) Model-Based Soil Parameter Identification for Wheel-Terrain Interaction Dynamics, *Proceedings of the 7th IARP International WS HUDEM*, pp. 154-160, March 28-30 2008, Cairo, Egypt

Lepistö, L.; Kunttu, I.; Lähdeniemi, M.; Tähti, T.; & Nurmi, J. (2007) Grain Size Measurement of Crystalline Products Using Maximum Difference Method, *In: Image Analysis,* pp. 403-410, Vol. 4522/2007, Springer Berlin, ISBN978-3-540-73039-2

Lindgren, D. R.; Hague, T.; Smith, P. J. P. & Marchant, J. A. (2002) Relating Torque and Slip in an Odometric Model for an Autonomous Agricultural Vehicle, In: *Autonomous Robots 13*, 73–86, Kluwer Academic Publishers, The Netherlands, 2002

Mäkelä, H. (2001) Outdoor Navigation of Mobile Robots, *PhD dissertation Helsinki University of Technology, Finland* (November 30. 2001)

Miller, S. L.; Youngberg, B.; Millie, A.; Schweizer, P. & Gerdes C. (2001) Calculating Longitudinal Wheel Slip and Tire Parameters Using GPS Velocity, *Proceedings of the American Control Conference*, pp. 1800-1805, June 25-27, 2001 Arlington, VA, USA

Müller, S.; Uchanski, M. & Hedrick, K. (2003) Estimation of the Maximum Tire-Road Friction Coefficient, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 125, 2003, pp. 607-617

Nishio, A.; Tozu, K.; Yamaguchi, H.; Asano, K. & Amano, Y. (2001) Development of vehicle stability control system based on vehicle sideslip angle estimation, *Proceedings of 2001 SAE World Congress*, *SAE Paper 2001-01-0137*

Ng, T.W. (2003) The optical mouse as a two-dimensional displacement sensor. *Sensors and Actuators,* A 107, (2003.) 21-25

Ng, T.W. & Ang, K.T. (2004) The optical mouse for vibratory motion sensing. *Sensors and Actuators,* A 116, (2004.) 205-208

OSMES (2004) *OSMES the optical speed measurement system* Siemens Transportation Systems 2004. [Online]. Available: www.transportation.siemens.com

Palacin, J.; Valgañon, I. & Pernia, R. (2006.), The optical mouse for indoor mobile robot odometry measurement, *Sensors and Actuators* A 126 (2006.) 141–147

Pi, M. H. & Zhang, H. (2005) Measurement of fine particle size with wavelet signature, P*roceedings of International Conference on Image Processing*, ICIP 2005. IEEE Volume 3, 11-14 Sept. pp.:III - 165-8

Smith, N. D. (2003) Understanding Parameters Influencing Tire Modeling, *Technical report on Colorado State University, 2004 Formula SAE Platform*

Sorensen, D. K. (2003) *On-line Optical Flow Feedback for Mobile Robot Localization/Navigation* MSc Thesis, 2003, A&M University, Texas, USA

Takács, T.; Kálmán, V. (2007) Optical Navigation Sensor, Incorporating vehicle dynamics information in mapmaking, *Proceedings of ICINCO 2007*, pp. 271-274, Angers France, 9 – 12 May, 2007

Thrun, S. (1998) Learning maps for indoor mobile robot navigation. In: *Artificial Intelligence*, 99(1), pp. 21–71. 1998.

Thrun S. (2002) Robotic mapping: A survey, In: *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002

11

Automatic Construction of a Knowledge System Using Text Data on the Internet

Junichi Takeno, Satoru Ikemasu, Yukihiro Kato
Meiji University, School of Science and Technology, Computer Science
Higashimita 1-1-1, Tama-ku, Kawasaki-shi, 214-8571 Kanagawa, JAPAN
takeno@cs.meiji.ac.jp (juntakeno@gmail.com)

Abstract

In order to be truly useful to humans, robots should be able to communicate smoothly with them. To gain this capability, robots must possess cognitive and speech functions similar to the consciousness function of humans. It is important for robots to be capable of a flow of consciousness and emotions, and for this purpose robots must first of all possess a so-called memory of knowledge.
This paper defines a network of association and *kansei* (a Japanese term relating to emotions and feelings) values that model the memory of knowledge. It also describes the construction of an association-*kansei* database comprising the association-*kansei* networks. This database searches through text data on the Internet, statistically processes the data, automatically calculates the strength of associations between collected words, extracts emotional elements incidental to respective words, and stores the results in the database. This paper further introduces techniques used to handle idioms, separate words of different meanings and classify concepts using superordinate-subordinate relationships. Lastly, the paper describes evaluation experiments that confirm the usefulness of the constructed database.

1. Introduction

Robot technology has been advancing rapidly in recent years. Robots are being manufactured to serve various purposes. In addition to ubiquitous industrial robots, many robots have been developed to work in human living spaces. Robots are advancing into the communities of humans and coexisting with them. For this reason, robots are increasingly being required to possess the capability of communicating with humans.
To be able to communicate with humans, robots must possess a so-called memory of knowledge, which would be the source of the flow of consciousness and an information source for emotion. The authors propose an association-*kansei* network as the model for memory of knowledge. *Kansei*, a Japanese word, is equivalent to emotion and feelings in English. In this model, associations between words are networked and correlated with mental images via certain factors to collectively handle the memory of knowledge and *kansei* of humans. In our present study, the association-*kansei* network has been incorporated into a

database, which we call the association-*kansei* database.
Fairly similar research is being performed at Doshisha University's Graduate School of Engineering in a project called the Commonsense Consciousness Judgment System. The theme of the project is not exactly consciousness per se, but for a given input, a database is searched for a similar conception to generate an image, which makes this a kind of expert system [1]. The primary objective of that research is to have a computer possess a degree of "common sense," and it is not particularly concerned with developing communication capability and constructing consciousness. There is also a website called KwMap. This site is a search site on the Internet and the site's method and results of collecting related words (which are nearly synonymous to what we term association words) are very similar to our approach which will be described in this paper. However, the study in the KwMap website is not relevant to feelings. It can be said that this study is a natural language processing research. However, it is unique in that the grammar is rarely used in our study.
This paper describes the association-*kansei* network and a model of "flow of consciousness," which is represented by said network. Next, the paper describes the automatic computerized construction of the association-*kansei* database that contains the association-*kansei* networks, and reports on the results of evaluation experiments that confirm the usefulness of the database. The paper further discusses how to handle idioms, a technique to separate words of different meanings (polysemic words), and hierarchization of concepts, both of which are indispensable for robots to simulate the knowledge structure of humans more closely.

2. Knowledge and Consciousness Modeling

This section describes the nature of the association-*kansei* network that models the knowledge structure of humans, and introduces the consciousness network that models consciousness.

2.1 Association-*Kansei* Network

In the association-*kansei* network, each word is a node and is connected to its associable words by directed edges. Each edge is given a value that indicates the strength of association. This value is called the association value. Words associated from other words are called association words. This network is capable of calculating the mental images it holds in the face of a subject by using *kansei*-related adjectives that are possessed by the network itself as index. Adjectives used for *kansei* calculation are described shortly. The association-*kansei* network shown in Fig. 1 is an example presented only for illustrative purposes. [2][3][4][5]

2.2 Conscious Network and Flow of Consciousness

Modeling the human flow of consciousness is an effective means to develop a robot capable of thinking like humans. Consciousness in humans is developed by perceiving and recognizing things while integrating information derived from external (via the five senses) and internal stimuli (self-desire and intense association). The authors believe that the consciousness function of humans is a function for recognizing the information derived from external and internal stimuli and integrating it into a unified, single concept. The authors also believe that the flow of consciousness is a process wherein the conscious

concept moves from the self to the other dynamically upon perceiving new internal and external stimuli.

This process is explained using the association-*kansei* network as described below. Upon receiving certain stimuli, a conscious concept and its association word group are extracted as a sub-graph. The association word group is a concept associated from the conscious concept, or a sub-conscious concept. The extracted sub-graph is called a consciousness network. Upon receiving some other internal and external stimuli, the central concept moves to other concepts, generating new consciousness networks. Through the repetition of this process, consciousness networks are generated one after another, thereby generating the flow of consciousness.

3. Association-*Kansei* Database

The authors have constructed a database of association-*kansei* networks and named it the association-*kansei* (AK) database. This section describes the association words and values, *kansei* words and values, and the automatic database construction procedure using text data extracted from the Internet. It also discusses the handling of idioms (a process by which the system can approximate the knowledge structure of humans), a technique for separating polysemic words, and concept hierarchization. [3][4][5]

3.1 Association Words and Association Values

An association word is a word associated from other words. The association value is an index of the ease of association. In Fig. 1, for example, Toyota, Mercedes-Benz and Accident are the association words of the central word Car. The ease of association from the central word Car differs for each of these three association words. The ease of association is quantified by the association value.

3.2 *Kansei* Words and Values

Kansei refers to the mental images that a person has in the face of external stimuli. For example, dog-lovers have a good impression of dogs whereas those who do not like dogs can have a hatred for them. *Kansei* is considered an element that generates emotion in humans. In the present study, *kansei* is represented by adjectives related to five of six basic human emotions: Happiness, Anger, Fear, Disgust and Sadness. These adjectives are called *kansei* words. *Kansei* values refer to the quantified level of mental images humans have for various stimuli. In the above example, "refreshing" and "fearful" are the *kansei* words for Happiness and Fear, respectively (Fig. 1). [6]

The remaining sixth basic emotion of Surprise cannot be expressed by adjectives, and thus is not considered in the present study. A list of some adjectives related to respective emotions is given in the Appendix (Table 1).

3.3 Constructing a Knowledge Database Using the Internet

Construction of the knowledge database is briefly described below. Association words are extracted from text data collected from the Internet, and the association value and *kansei* value are calculated for each association word.

The authors decided to use the Internet as a data source to construct the knowledge database because of its three major merits: (1) a large volume of information may be

collected with ease because the Internet is the world's largest database, (2) information is renewed daily and the latest information is accessible at all times, and (3) text data on the Internet are written by humans and thus reflect human sensibilities very well.

3.3.1 Text Extraction Process

The process of text extraction from the Internet is described below.
At first some websites were selected arbitrarily. The text data and links to other websites found on webpages were picked up and saved. The external links were followed to their linked pages, and then the text data and links to other websites found there were also picked up and saved. These operations were repeated to collect a huge amount of text data.

3.3.2 Extraction of Association Words and Values

To define association words for a given central word, we must check the correlation between the given word and other words. Any sentence written by a human was written with a presumed intention to convey certain information to other humans. Given this fact, we believe that there are certain correlations among the words that appear in any given single sentence. This may be said to be an extensive interpretation of Hebb's rule of simultaneity of stimuli. Based on this idea, we define the words appearing in the same sentence as association words and extract them from the collected text data.
The association value is calculated as the ratio of the simultaneous appearance of the central word and the association words in the same sentence to the total count of the appearance of the central word. Specifically, assume x is the central word, $c(x)$ the total count of appearance of the central word, a_i an arbitrary association word of x, and $c(x:a_i)$ the count of the simultaneous appearance of x and a_i in the same sentence, then the association value $p(x:a_i)$ from the central word x to the association word a_i is given by the following equation:

$$p(x:a_i) = \frac{c(x:a_i)}{c(x)} \quad \left(0 \le p(x:a_i) \le 1 \right) \tag{1}$$

3.3.3 Calculation of *Kansei* Values

The *kansei* value, like the association value, is calculated by how frequently a *kansei* word appears in the sentence containing the central word.
The *kansei* value is calculated in two steps. First, assume x is the central word, $c(x)$ the total count of the appearance of the central word x, and $c(x:k)$ the number of the *kansei* words simultaneously appearing with the central word x, then we perform the following calculation. The derived value is tentatively called pre-*kansei* value $K_p(x)$, because our aim is to obtain the final *kansei* value.

$$K_p(x) = \frac{c(x:k)}{c(x)} \tag{2}$$

The pre-*kansei* value can exceed unity if there are many *kansei* words. Considering that the association value ranges from zero to unity, it is convenient if we express the *kansei* value in the same zero-to-unity range. After obtaining the pre-*kansei* values for all nodes, we therefore identify the maximum pre-*kansei* value $K_{\max}$, and define the *kansei* value $K(x)$ for the central word x as the ratio of the pre-*kansei* value $K_p(x)$ of a given central word x to $K_{\max}$. This is the second step in calculating the *kansei* value.

$$K(x) = \frac{K_p(x)}{K_{\max}} \quad (\ 0 \le K(x) \le 1 \ ) \tag{3}$$

As mentioned before, there are five kinds of *kansei* words, related to the five human emotions of Happiness, Anger, Fear, Disgust and Sadness. This means that each node has five kinds of *kansei* values. All these values are calculated in the same manner.

3.4 Processing to Approximate the Knowledge Structure of Humans

The current database is a simplified model of the human knowledge structure. The information contained in it is insufficient for a robot to communicate with humans. We added the process of hierarchizing concepts to improve the AK network ands make it closer to the human knowledge structure.

The current idiom-generating process uses an existing idiom dictionary. An automatic idiom generation algorithm is under development.

3.4.1 Semantic Classification

Concepts and words do not necessarily have a one-to-one correspondence in languages. Idioms are an example of this. There are also many polysemic words, that is, words that have several meanings.

Semantic classification refers to the handling several meanings of a polysemic word as separate nodes on the knowledge database. For example, the word Book can mean both reading material and to make a reservation. The word Will can mean determination, as well as a written statement specifying the distribution of a deceased person's property. Depending on the context, the word Spring can mean a coil of metal, a season of the year, or water gushing up from underground. Take another example of the word Virus. The root meaning is the same but the image and the usage of the word Virus are quite different when referring to disease-causing microbes or a contagious computer program.

Semantic classification is a process for handling words of multiple different meanings separately according to respective concepts. The final objective is to construct a database of the principle of each node with a single concept.

Clustering is used as a means of semantic classification. As a kind of data-mining technique, clustering is based on the idea that similar data behave similarly. Similar data are picked up from a group of data and collected into respective clusters. Each cluster contains similar objects, and objects of different attributes are collected in different clusters to the extent possible.

Hierarchical clustering analysis is employed in our present study.

3.4.1.1 Hierarchical Clustering

Hierarchical clustering is an approach to grouping together objects that are "close" to one another sequentially. The hierarchical clustering approach is described below.

1) There are initially n objects $O_1, O_2 ... O_n$, each of which belongs to its own cluster, that is, there are n clusters in all.
2) We calculate the value d_{ij} representing the level of similarity between arbitrary objects O_i and O_j according to an arbitrary criterion. The pair of objects with the highest similarity index value is put into a new single cluster. There are n-1 clusters at this stage.

The above process is repeated until all data are finally collected in a single cluster (Fig. 2). In hierarchical cluster analysis, the sequence of agglomeration is shown graphically as a dendrogram (Fig. 3). The objective of hierarchical cluster analysis is to sort the data and generate the dendrogram. The dendrogram clearly shows how clusters are formed and which pairs of elements are closely related to other pairs.

3.4.1.2 Semantic Classification of Association Words

Semantic classification is a process of dividing a node that has different images (a polysemic word node), which is treated as a single node or a word in the database, according to its different meanings. After being divided, association words are newly grouped for each of the divided nodes. Specifically, we calculate the similarity indices of the association words, and group them according to the individual meanings of the node. Each of the association word groups now contains association words of a similar meaning or words that are particularly strongly correlated (Fig. 4).

When clustering, we define the similarity index as the criterion to determine if any given pair of elements should be treated as the same group. The set of elements subject to clustering at this stage consists of the association words of individual polysemic nodes (primary association words). Two procedures may be used to calculate similarity indices as described below.

a. Using Secondary Association Words as Similarity Indices

The association words of a polysemic word (the data used for clustering) are called primary association words. The words associated from the primary association words are called secondary association words. In the first technique, similarity between the secondary association word groups is quantified to obtain the similarity indices.

The similarity between the secondary association word groups is determined by counting the number of common nodes. In Fig. 5, the association word groups for Library and Page have two words in common: Read and Web. The value 2 is directly used as the similarity index, and this is called the absolute similarity index. From another perspective, of the four words in the two secondary association word groups, the two words Read and Web are common, thus the ratio of the number of common nodes to the total number of nodes is 2/4. The similarity index in this representation is called a similarity index by ratio. In the remainder of this paper, when we speak of a similarity index, we are referring to a similarity index by ratio.

In hierarchical clustering, the elements are grouped together in the order of high similarity index. In semantic classification, there remains the problem of how to handle secondary association word groups when the elements are clustered. A sum-set or product-set approach can be used to determine the association words for the new cluster. For example, when the primary association words of Library and Page are clustered using the sum-set approach, the new association word group includes Read, Index, Number and Web. When using the product-set approach, the new association word group includes Read and Web. When the sum-set approach is used, the secondary association word groups increase rapidly as agglomeration proceeds, and the probability of matching other association word groups increases, which will eventually pick up even those elements that should not be classified into the same group. This will end up with space dilation. For this reason, we have selected the product-set approach.

The secondary association words are used to calculate the similarity indices because the association words represent the specific features of the central word. To improve the clustering accuracy, we remove in advance any nondistinctive words that are included in nearly all of the secondary association word groups. In our experiments, we define nondistinctive words to be words that exist in 80% of the secondary association word groups. We performed clustering after removing these nondistinctive words.

b. Using Connectivity for the Similarity Indices of Primary Association Words

The second available procedure involves using connectivity to define similarity indices. Connectivity, the strength of connection between two nodes, is expressed by the sum of the association values of the connected nodes. Association values indicate the ease of association from one given word to another and are shown by directed edges in the AK network. On the other hand, connectivity is shown by bi-directional segments that directly indicate the strength of the connection of the two words involved.

When clustering, the connectivity must be evaluated not only between single nodes but also between clusters. Connectivity between clusters is expressed as the average of the connectivity between individual nodes of one cluster and those in another. Specifically, the connectivity between cluster P (number of elements: n_p) and cluster Q (number of elements: n_q) is defined as follows: Assume p_i is the i-th node of cluster P and q_j the j-th node of cluster Q. Obtain all connectivity $b(p_i, q_j)$, and divide the sum by $p_i \times q_j$ (all combinations of p_i and q_j) to derive the average (equation 4 and Fig. 6).

$$b(P,Q) = \frac{\sum_{i=0}^{n_p} \sum_{j=0}^{n_q} b(p_i, q_j)}{n_p \times n_q} \tag{4}$$

$b(P,Q)$: Connectivity between clusters P and Q

$b(p_i, q_j)$: Connectivity between p_i and q_j

n_p : Number of elements in cluster P

n_q : Number of elements in cluster Q

c. Results of Experiments using the Two Procedures and a Comparison of Results
Experiments were conducted on the procedure of representing the common components of the secondary association word groups as similarity indices and on the procedure of using the similarity indices of the primary association words expressed by connectivity. The results are compared. It was found that, as a clustering technique, using the similarity indices of the primary association words expressed by connectivity yields better results than representing the common components of the secondary association word groups as similarity indices (Fig. 7). This is because of the following reason: in this knowledge database, the association index from one word to another is calculated for all possible combinations of words. As such, evaluating the association indices between words included in the primary association words (connectivity) is more direct than counting the number of common words included in the secondary association word groups. The authors therefore believe that the connectivity-based procedure more readily provides ample information for clustering.

3.4.2 Hierarchical Processing of Concepts

Humans not only share knowledge among themselves but also use hierarchization of language concepts to facilitate mutual understanding when communicating. For example, Hawk and Crow are subordinate concepts of Bird. Bird, on the other hand, is a part of a larger classification of Animal, or put differently, Animal is the superordinate concept of Bird. The concept gets more abstract as we go up in the hierarchy and gets more concrete as we go down the hierarchy. All concepts represented by languages have this type of hierarchical structure. This would indicate that humans are born to accumulate knowledge through hierarchization unconsciously (Fig. 8). [7] This structure helps to extend the span of communication. Humans can explain what they want to say to others by describing things in concrete or in the abstract as required. Assume one of your friends is wondering what kind of pet he/she should have. You want to recommend a dog, and you may simply, and generally, say, "Dogs are obedient to their masters and they're easy to care for," without taking the trouble of mentioning each and every individual breed of dog, say for instance, "A dachshund is obedient to its master and it's easy to care for," "A bulldog is…," and so on. Then you can focus more concretely, saying, for example, "Of all dog breeds, a dachshund is docile and easy to keep." How can we create such a hierarchical structure in our AK network? We explained that superordinate concepts are more abstract than subordinate concepts. Because of this fact, superordinate concepts are easier to find in text data than subordinate concepts. In other words, it is generally said that more superordinate concepts appear in text than subordinate concepts, and that the superordinate concepts have more association words than subordinate concepts. Accordingly, the values of the associations from the subordinate to the superordinate concepts are higher than the values of associations from the superordinate to the subordinate concepts.
We believe that this difference in the association values is an important factor in differentiating between the superordinate and subordinate concepts.
To verify this hypothesis, we conducted the experiments described below.

3.4.2.1 Experiments on Concept Hierarchical Processing Based on Differences in Association Values

Associations from subordinate to superordinate concepts are generally stronger than

associations from superordinate to subordinate concepts as mentioned before. In our experiments, we included one word generally considered a superordinate concept and five words generally considered subordinate words to that superordinate word. These six words were presented to human subjects to determine whether or not the superordinate concept word was correctly extracted. Of the six words, one word with higher association values than the association values of all the other five words was extracted as the superordinate concept.

The superordinate concept was correctly extracted in many cases (Table 2). In one case, however, the hypothesis was not valid as shown by the failure in Table 3. This was because a polysemic word (Apple) was included in the word group. This indicates that proper processing of polysemic words is indispensable for enhancing the accuracy of concept hierarchization.

Our experiments have shown that this hypothesis is valid for a large number of concepts. It is difficult to achieve hierarchization of all concepts using this approach alone, but we consider this approach to be an important element of concept hierarchization.

4. Experiments and Considerations about the AK database

Experiments were conducted on the AK database to confirm the effectiveness of the approach used in our present study. Vocabulary matching was tested with the Standard Vocabulary List SVL12000 and the *Eijiro* electronic English-Japanese dictionary to determine to what extent standard words were included in the AK database. We conducted a questionnaire survey using general public subjects to verify if the association and *kansei* that were artificially calculated by the database approximated human sensibilities. The results of this survey and the information in the database were compared.

4.1 Comparing the AK Database with the Standard Vocabulary List and an E-J Dictionary

We visited about 1.2 million websites, collected text data and created a database containing 500,000 English words.

The purpose of our present study was to construct a knowledge database as mentioned before. As such, it was necessary that words actually used by humans be included in the database. We compared our database with the standard word list and a dictionary to find out to what extent our database contained words used by humans.

Vocabulary matching was performed using two lists. One was SVL12000, a list prepared by ALC who provide various support programs for English learning. The SVL12000 contains 12,000 English words that are generally used in conversation and written sentences, with the exception of proper nouns. The other was an English-Japanese electronic dictionary called *Eijiro* that contains more than 300,000 English words. The result of the comparison of our database with these two reference materials is described below (Table 4).

The coincidence ratio with *Eijiro* was 51.52%, which was an inadequate result. The coincidence ratio with SVL12000 was 100%, meaning that our database would be highly useful, with almost all generally used words included.

The reason why the coincidence ratio with *Eijiro* was low could be explained by the scarcity of text data used as the information source and the poor coverage of various genres in our visits to websites. For our database to be able to pick up more words found in *Eijiro*, we would need to collect much more text data and visit many more websites in different

categories on the Internet.

4.2 Comparing Our Derived Association-*Kansei* with Human Association-*Kansei*

We conducted two questionnaire surveys using 73 Japanese subjects to learn how much the association and *kansei* of our database approximated human values.

4.2.1 Verifying Association Words by Questionnaire Survey

The objective of the first questionnaire survey was to evaluate the reliability of the association words in our AK database. Twelve basic words assumed to be known by everybody were selected from the database and shown to the subjects together with their respective association word groups. The subjects were asked to evaluate the association word groups in three grades: Natural, Not Sure and Unnatural. The result of the survey is shown at the top of Table 5, and the basic words and their association word groups at the bottom of the Table.

In Table 5, on average Natural was selected by about 66% of the respondents, overwhelmingly exceeding the Unnatural response selected by about 9%. Looking at the basic words individually, the number of respondents who thought the relevant association word group was Natural was predominantly larger than the number of respondents who thought it Unnatural, with the exception of the word Mushroom. For the word Mushroom, the respondents were nearly equally divided among Natural, Not Sure and Unnatural responses. This may have been because of dispersion attributable to different associations by people. None of the basic words was rated excessively Unnatural, which would indicate that as far as these 12 well-known words were concerned the association words in our AK database were effective and reliable.

4.2.2 Verifying *Kansei* Values by Questionnaire Survey

In our second questionnaire survey, nine basic words to be appraised were listed in the questionnaire. For each of the words, the emotion words of Happiness, Anger, Fear, Disgust and Sadness were shown opposite the basic word. The subjects were requested to circle each of the emotion words that he/she thought were applicable to the given basic word. The number of circles were tallied for each emotion word and divided by the number of respondents to derive the average *kansei* value of the respondents. The result of the experiment compares the values of human subjects with the *kansei* values of our AK database (Table 6).

To facilitate comparison, each of the five of the nine basic words selected has a significantly high value for a certain *kansei* in the AK database, while others have mixed kinds of *kansei*.

Of the nine basic words tested, four characteristic words are described below.

As seen in our results, for eight of the nine basic words, human subjects and our database shared the same emotion word as the word of the highest *kansei* value. Furthermore, for four of these eight words, human subjects and our database shared another same *kansei* word as the word of the second highest *kansei* value. For the basic word Complaint, which had the largest discrepancy between the human and database *kansei* of all nine basic words, the top two emotion words, put together, were identical (Disgust and Anger). It is true that humans can have emotions which are a mixture of three or more categories of *kansei*, and the agreement of the top two emotion words indicates that our database contains the features of human *kansei*.

5. Problems and Prospects

Our experiments have broadly verified the effectiveness of our AK database in terms of the number of nodes, properties of association words and *kansei*. The problem is that matching with the dictionary was low. Thus, it is necessary to extract more text data.
It is also essential, as mentioned before, to divide polysemic words by their meanings. This division and the accompanying classification would enable us to sort not only nodes but also the frequency of appearance and association words, and then assumed relationships would be applicable to many more concept hierarchies.
Our experiments have demonstrated that differences in association values between the superordinate and subordinate concepts play an important role in the construction of the hierarchical structure. But such relationships alone are not sufficient. To form a hierarchical structure for all concepts, the sorting of concepts into respective categories is necessary, in addition to the above relationships. To hierarchize concepts in practice, we specify central words, and extract superordinate and subordinate concepts based on the association words of respective central words. In reality, association words are a mixture of words of different categories, although the interrelationships of the words are strong. For example, the association words of the central word Dog may include Collar, Dog Food and other words that have no hierarchical relationship among them. These types of words must be eliminated during hierarchization. A method to evaluate association values among the association words is currently being studied. The underlying idea of this method is that the association value of association words belonging to the same category would be higher than the association values of words belonging to different categories.

6. Conclusion

This paper described an association-*kansei* (AK) network that is the information source for the artificial flow of consciousness and the creation of artificial emotions. It discussed the classification of polysemic words and concept hierarchization. The authors' AK database, incorporating these features, is able to associate words similar to human functions and has emotions that approximate those of humans as shown by the experiments. The Internet was used as an information source, so the system can flexibly reflect the trends of the times.
We believe that a robot capable of communicating with humans can be created by making the AK network closer to the knowledge structure of humans and achieving a more human-like flow of consciousness.

7. References

A. Kometani, H. Watabe, T. Kataoka, Constructing a Commonsense Consciousness Judgment System, *The 17th Conference of Japanese Society for Artificial Intelligence*, 2003.

T. Kaneko, J. Takeno : Information Scientific Research of Funniness and Sadness, *IEEE Int. Work. On Robot and Human Interactive Communication*, 0-7803-7222-0/01IEEE, pp.450-455, 2001.9

T. Kanamori, Y. Imai, J. Takeno, Extraction of 430,000 association words and phrases from

internet web sites, *International Conference on Machine Automation*, 2005, 1929-1934.

Y. Imai, M. Tamada, J. Takeno : Method for Extracting Associated Words, Association Value and Kansei Value from the Internet, The 8th World Multiconference on Systemics, Cybernetics, and Systemics(IIIS), *Proceeding Vol.III*, ISBN 980-6560-13-2, pp.160-165, 2004-7.

Y. Kato, Y. Imai, T. Kanamori, K. Furuike, J. Takeno, Extraction of Association and Phrases Information from the Internet and Creation of a Knowledge Database, *3rd International Conference on Autonomous Robots and Agents (ICARA)*, 2006.

A. Ogiso, S. Kurokawa, M. Yamanaka, Y. Imai, J. Takeno, Expression of emotion in robots using flow of artificial consciousness, *IEEE Int. CIRA2005*, 2005, 421-426.

S. Kawakami, *An introduction to cognitive linguistics* (Tokyo, Kenkyusha, 1996).

N. Goldblum, *The brain-shaped Mind* (Cambridge University Press, Cambridge, 2001).

K. Araki, *An introduction to natural language processing* – a computer which can talk and learn a language - (Tokyo, Morikita Pulishing, 2004).

R.F. Simmons, *Computations from the english* (Englewood Cliffs, NJ: Prentice-Hall, 1984).

M. Nagao, *Natural language processing* (Tokyo, Iwanami Shoten, 1996).

J. Takeno, T. Kaneko: "*The Stream of the Consciousness Network of Machine Mind*", International Institute of Informatics and Systematic, 2002.6

Naomi Goldblum: "*The brain-shaped Mind*", Cambridge University Press, 2001

Manfred Spitzer: "*The Mind within the Net*", M.I.T. Press, 1989.

12

Adaptive GPC Structures for Temperature and Relative Humidity Control of a Nonlinear Passive Air Conditioning Unit

Rousseau TAWEGOUM[1], Riad RIADI[1], Ahmed RACHID[2], Gérard CHASSERIAUX[1].

[1]*Unité Propre EPHOR(Environnement Physique de la plante HORticole) Institut National d'Horticulture 2, rue Le Nôtre 49045 Angers*

[2]*Université de Picardie Jules Verne, IUP GEII. 33, rue Saint LEU 80000 Amiens FRANCE*

1. Introduction

The use of climate-controlled greenhouses and growth chambers makes it possible to maintain high crop productivity and quality when outside air conditions are not favourable. Climate control accuracy may help optimize production costs by reducing energy consumption and by better respecting production calendars. Such objectives can be reached by incorporating new technologies for crop growth chambers and reduced greenhouses with an HVAC installation. Various climate control technologies have been used for commercial and experimental greenhouses and growth chambers. Many studies have focused mainly on temperature control, while considering relative humidity to be of secondary importance, a background detail.

Generally speaking, air conditioning units used in crop growth chambers are made up of heating and cooling system components with a compression cycle (Hanan, 1997; Arquello & Velez, 1999; Jones & Jones, 1984). In addition to the energy cost and the high maintenance expenses for this type of system, they present an ecological issue due to the pollutant emissions generated by the use of refrigerating gases.

The main cooling technologies routinely used in greenhouses are ventilation, evaporative cooling, and composite systems. A simple way to reduce the difference between inside and outside air temperature is to improve ventilation. Natural ventilation uses very little external energy, but whether it is natural or forced, ventilation is of limited efficiency and not satisfactory on sunny days.

Evaporative cooling using fan-pads (Kittas et al., 2003) or fog/mist (Montero et al., 1994) inside greenhouse and roof cooling systems (Willits & Peet, 2000) represents an efficient means of greenhouse cooling that can lower the inside air temperature significantly below the ambient air, but the range of relative humidity variation remains limited. Fan-pads work on negative pressure, so that very often outside hot air mixes with the inside cool air through infiltration, which reduces the efficiency of the system quite significantly. Mist or

fog systems can provide more uniform temperature distribution than fan-pad systems, in addition to ensuring uniform high humidity levels. One of the drawbacks of fog mist is that the compressor consumes large amounts of energy, which increases the cost of operating the system. This method also uses expensive foggers or nozzles, which often shocked due to insoluble and soluble salt present in the water, thereby reducing the working efficiency of the system.

Apart from these systems, two primary composite systems, such as earth-to-air heat exchangers (EAHES) and aquifer coupled cavity flow heat exchangers (ACCFHES), can be used for heating as well as cooling greenhouses. The EAHES uses the earth potential for heating and the ground potential of the earth for cooling the greenhouses in summer conditions due to its constant year round temperature. In this case, hot greenhouse air is circulated through the burried pipe (2-4m depth) for dissipation of heat to the undergroung soil. The aquifer coupled cavity flow heat exchanger system (Sharma 2007a) uses deep underground aquifer water from an irrigation tube well at the ground surface at nearly constant temperature. The major disadvantage of using EAHES is the cost of digging and laying the pipes. Deterioration of the pipes under soil pressure also makes this system less reliable for projects of long duration. A substantial review of cooling technologies may be found in (Sethi & Sharma 2007).

We have investigated an ecological approach for climate control based on a passive principle. A similar approach is set out in (Buchholz et al., 2006). The conditioning unit under study is a new design with innovative proprieties and offers various environmental advantages (Tawegoum *et al.*, 2006a). It does not use the more typical compression system or absorption-refrigeration cycle. It was designed to produce a microclimate with variable temperatures and variable relative humidity setpoint values inside the growth chambers. A complete physical model of this plant developed in (Riadi *et al.*, 2006) shows that this complex global system is composed of three nonlinear subsystems.

In this chapter, the adaptive direct and indirect general predictive control is used to control the temperature and the relative humidity produced by the passive air conditioning unit. The chapter is divided into three sections. The first section is devoted to the analytical modelling of a conditioning unit. The second defines the properties of adaptive direct and indirect approaches used for control. The last section presents real time results from the system.

2. Analytical model of the system

2.1 System description

The required micro-climate must be produced by a passive air-conditioning system without a freezing unit and compressor, or refrigeration cycle, and without pollutant emissions (Tawegoum et al., 2006a). The specificity of this system is tied to its capacity to produce a variable microclimate with variable temperatures and relative humidity set points. Since temperature and relative humidity are highly coupled, one way to achieve these objectives is to delink the control of the temperature from the relative humidity control. The air-conditioning cycle is presented in the diagram below.

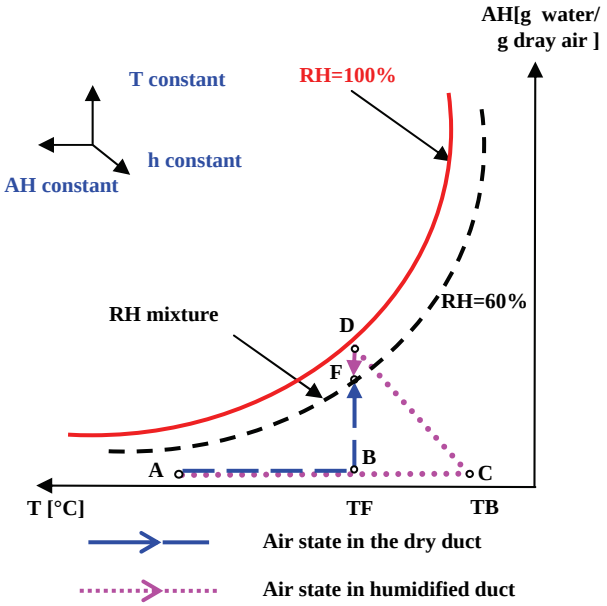


Fig. 1. Thermodynamic cycle of the unit

Figure 1 shows the different thermodynamic phases of the air-conditioning cycle and the region corresponding to our zone of interest. The system depends on mixing two air flows, each with a different humidity level. The air intake can be from inside the greenhouse (point B) or outside the greenhouse (point A). Regardless of the source of the air supply, the characteristics of the air are clearly defined.

The characteristics of the air at point F are also known because the final temperature T_F is the set point temperature, and the moisture RH_F is set by the user. As the air heating operates at a constant absolute humidity, point B can be easily found by knowing the value T_F.

Computing the characteristics of the air at point C is more complex. These characteristics can be deduced from point D, at which the temperature equals T_F. In D, air must be practically saturated. As cooling humidification (from C to D) operates at constant enthalpy, point C can be calculated by determining the characteristics of points D and A.

The energy required for heating can be computed based on the enthalpy values of points A, B and C. The airflow rate required to obtain the relative humidity set point is computed using the relative evolution of the line 'D-B'. Considering the values of q_i, the final expressions of the absolute humidity (absolute moisture content) and the temperature are obtained by the following static thermodynamic equations:

$$AH_F = \frac{q_1 AH_B + q_2 AH_D}{q_1 + q_2} \tag{1}$$

$$T_F = \frac{q_1(\alpha + \beta AH_B)T_B + q_2(\alpha + \beta AH_D)T_D}{q_1(\alpha + \beta AH_B) + q_2(\alpha + \beta AH_D)} \quad (2)$$

with $\alpha = 0.24$, $\beta = 0.46$ and q_i being the air flow mass proportional to the aperture position. Knowing both AH_F and T_F gives a unique value of RH_F (Tawegoum et al., 2006a).

The unit is composed of two flows: a non-saturated flow (or dry duct) and a saturated flow (or humidified duct). As shown in figure 2, in the saturated air flow, fresh air is saturated in humidity after being heated by a coil resistor. Saturation operates at constant enthalpy (Chraibi et al., 1995). The saturation unit consists of a closed system, including a pump, a water tank and cross-corrugated cellulosic pads of the type used in cooling. The suction pump carries water from the tank to the top of the pads. Once a steady state of saturation is reached, the pads contain a constant mass of water with a given water output rate and a given temperature. In the unsaturated air flow, fresh air is only heated by another resistor coil. Dry pads are included to provide pressure drop balance. The low speed of the air and the water through the pads reduces the difference in pressure drop between the two streams.

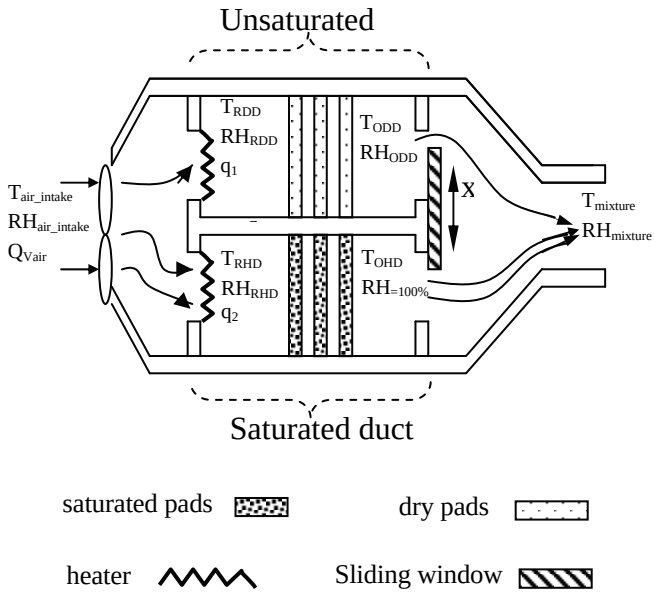


Fig. 2. Air-conditioning system

The proportional mixing of the two air flows is carried out by an aperture operate by a DC motor.Assuming that the two air flows are mixed properly, a local climate can be easily produced in the growth chamber.

2.2 System modelling

2.2.1 Temperature modelling

The differential equations describing the dynamic behaviour of the conditioning unit are derived from the energy conservation law. The temperature behaviour in the mixing zone is given by:

$$\frac{dT_{mixture}}{dt} = -\frac{\alpha(x)\,Q_{Vair}}{V_{mixer}}\left[T_{mixture} - T_{ODD}\right] - \frac{(1-\alpha(x))\,Q_{Vair}}{V_{mixer}}\left[T_{mixture} - T_{OHD}\right] \quad (3)$$

where $T_{mixture}$ is the air temperature (°C) in the mixer, T_{ODD} is the air temperature (°C) after the dry duct, T_{OHD} is the air temperature (°C) after the humidified duct, $a(x) \in [0,1]$ the volumetric air flow percentage in the dry duct (%), x the percentage of aperture opening (%), Q_{Vair} the total volumetric air flow rate (m^3/s), V_{mixer} the volume of the air mixer.

q_1, q_2 are the volumetric air flow rates depending on the aperture position (figure 2). The total volumetric air flow rate Q_{Vair} is given as:

$$Q_{Vair} = q_1 + q_2 = \alpha(x)Q_{Vair} + (1-\alpha(x))Q_{Vair} \quad (4)$$

In the dry duct, the heat balance in the pads is expressed by the following equation:

$$\frac{dT_{ODD}}{dt} = -\frac{\alpha(x)\,Q_{Vair}}{V_{DD}}\left[T_{ODD} - T_{air\_intake}\right] + \frac{k_{RDD}}{\rho_{air}\,C_{air}\,V_{DD}}U_{DD} \quad (5)$$

with T_{air_intake} the intake air temperature (°C), U_{DD} the applied voltage (V), proportional to the resistor heating in the dry duct, k_{RDD} the proportional coefficient between the voltage and the heating-power (J/sV), ρ_{air} the air density (kg/m^3), C_{air} the specific heat of air (J/kg °C), V_{DD} the volume of the dry duct (m^3).

In the humidified duct, the heat balance in the pads and the heater lead to the following equations:

$$\begin{aligned}\frac{dT_{OHD}}{dt} = &-\frac{(1-\alpha(x))\,Q_{Vair}}{V_{pad}}\left[T_{OHD} - T_{RHD}\right] - \frac{h\,A_{pad}}{\rho_{air}\,C_{air}\,V_{pad}}\left[T_{RHD} - T_{wat\_intake}\right] \\ &+ L_v(T_{wat\_intake})\frac{(1-\alpha(x))\,Q_{Vair}}{C_{air}\,V_{pad}} + \left[AH_{sat}(T_{wat\_intake}) - AH_{air\_intake}\right]\end{aligned} \quad (6)$$

where T_{RHD} is the air temperature (°C) after the heater of the humidified duct, T_{wat_intake} the intake water temperature in the pads of the humidified duct, U_{HD} the applied voltage

(V), proportional to the heating in the humidified duct, $AH_{sat}(T_{wat\_intake})$ and AH_{air_intake} are respectively the saturated absolute humidity at the temperature of water intake and the absolute humidity of the air intake (kg of water/kg of dry air), k_{RHD} the proportional coefficient between the voltage and the heating-power (J/sV), ρ_{water} the water density (kg/m³), C_{water} the water specific heat (J/kg °C), V_{RHD} the heater chamber volume of the humidified duct (m³), V_{pad} the pads volume (m³), A_{pad} the pads exchange area (m²), $L_V(T_{wat\_intake})$ latent heat (J/kg of water) at the temperature of the intake water, h the convective heat coefficient (J/m²s°C) .

2.2.2 Relative humidity modelling

The heat and mass conservative law applied to the humid duct, the dry duct and the mixing zone give rise to the following equations for absolute humidity.

$$\frac{dAH_{OHD}}{dt} = -\frac{(1-\alpha(x))Q_{Vair}}{\varepsilon_r V_{pad}}\left[AH_{OHD} - AH_{air\_intake}\right] + \frac{1}{\rho_{air} C_{air} \varepsilon_r V_{pad}}\left[Q_{wat\_intake} - Q_{wat\_intake}\right] \quad (7)$$

$$\frac{dAH_{mixture}}{dt} = -\frac{\alpha(x)Q_{Vair}}{V_{mixture}}\left[AH_{mixture} - AH_{ODD}\right] + \frac{(1-\alpha(x))Q_{Vair}}{V_{mixture}}\left[AH_{mixture} - AH_{OHD}\right] \quad (8)$$

where ε_r is the pad porosity coefficient (%), Q_{wat_intake} is the water intake flow mass (Kg/s). More detailed information may be found in (Riadi, 2007).

The physical models shown above are complex and difficult to use for control objectives, especially with respect to relative humidity. The model structure is MIMO, with internal coupling between the temperature and relative humidity, and an instationarity due to the operating point variation during the control (Riadi et al., 2007). Mention can also be made of the presence of the external disturbance on the controlled outputs (temperature, relative humidity). The air flow measurements for the main aperture positions indicate a nonlinear relationship between the percentage of air flow and the percentage of aperture positions (Tawegoum et al., 2006b).

To take into account these uncertainties and complexities, the process is seen as a time-varying system and the recursive estimation approach must be used to estimate parameters in real time. The predictive control algorithms based on generalized predictive control or even long range predictive control strategies have proven to be efficient, flexible and

successful for industrial applications (Corréa et al., 2000; Nybrant, 1989; Rafilamanana et al., 1992). This strategy is associated with the recursive estimation algorithm in order to obtain better performance for both tracking and regulation problems.

3. Indirect and Direct Generalized Predictive Control (GPC) design

3.1 Indirect Generalized Predictive Control concepts

The synthesis of the generalized predictive controller (GPC) suggested by Clarke (Clarke et al., 1987a; Clarke et al., 1987b) provides one of the methods that may be used as an adaptive control strategy. However, it must be combined with an online identification method (Landau & Dugard, 1986; Msaad & Chebassier, 1992). This method was used successfully in industrial applications of various forms (Dumur et al., 1997; Richalet et al., 1978; Filatov & Unbedhauen, 2004; Dion et al.1991,). Among the declared advantages of the generalized predictive control (Clarke, 1988; Camacho & Bordons, 2000), one may mention that it can be applied to processes with variable pure delay, with a non-minimum phase, and that it does not involve an apparent problem when the process model has too many parameters, contrary to pole placement strategies and linear quadratic control.

The method described in this paragraph is developed by Clarke (Clarke et al., 1987a), (Clarke et al., 1987b), and is given in the SISO case :

1) The basic model is CARIMA (Controlled Auto-Regressive Integrated Moving Average) defined to represent the behaviour of the process around a nominal operating point, given by the following form:

$$A(q^{-1})\Delta y(k) = B(q^{-1})\Delta u(k-d) + C(q^{-1})\varepsilon(k) \tag{9}$$

$y(k)$ is the system output, $u(k)$ the system input, $\varepsilon(k)$ the uncorrelated random sequence, $\Delta(q^{-1}) = 1 - q^{-1}$ the difference operator, $A(q^{-1})$, $B(q^{-1})$ $C(q^{-1})$, are polynomials with n_a, n_b and n_c degree respectively.

2) The optimal j-step ahead prediction of the system output using the available information at instant ' k ', is given by (10):

$$\hat{y}(k+j) = G_j(q^{-1})\Delta u(k+j-1) + l_j \tag{10}$$

where: $l_j = F_j(q^{-1})y(k) + H_j(q^{-1})\Delta u(k-1)$

Where: F_j, E_j, G_j, H_j are polynomial solutions to the Diophantine equations.

the matrix formulation is represented in (11):

$$\hat{Y} = G.\Delta U + L \tag{11}$$

with

$\hat{Y}^T = \left[\hat{y}(k+1)\ldots\hat{y}(k+N_2)\right];\ \Delta U^T = \left[\Delta u(k)\ldots\Delta u(k+N_2-1)\right];\ L^T = \left[l_1(k+1)\ldots l_{N2}(k+N_2)\right]$

3) The performance index is a weighted sum of predicted tracking errors and future control signal increments:

$$J(k) = \sum_{j=N1}^{N2}(w(t+j)-\hat{y}(k+j))^2 + \lambda \sum_{j=N1}^{N2}(\Delta u(k+j-1))^2 \tag{12}$$

where: $\Delta u_j(k+j) = 0$, for $j \geq N_u$.

$w(k+j)$ are the set points values at time $k+j$, $\hat{y}(k+j)$ the output prediction at time $k+j$, N_1 the minimum prediction horizon, N_2 the maximum prediction horizon, N_u the control horizon, λ the control-weighting factor.

4) A closed form solution of the optimal law exists, which takes as inputs $y(k)$ and $u(k-1)$ and as output ΔU_{opt} [13]. The formula is derived through analytical minimization of the previous cost function. The optimal control law is:

$$\Delta U_{opt} = \left[G^T G + \lambda I\right]^{-1} G^T (W - L) \tag{13}$$

With G a $(N_2 - N_1 + 1) \times N_u$ matrix. Only the first control value is finally applied to the system according to the receding horizon strategy:

$$u_{opt}(k) = u_{opt}(k-1) + \overline{G}(W - L) \tag{14}$$

where $\overline{G}$ is the first line of matrix $\left[G^T G + \lambda I\right]^{-1} G^T$.

The equivalent RST controller is computed through a difference equation [17]:

$$S(q^{-1})u(k) = T(q^{-1})w(k+N_2) - R(q^{-1})y(k) \tag{15}$$

In the case of time varying parameters, the previous controller must be included within an adaptive structure. The system parameters $A(\hat{a}, q^{-1})$, $B(\hat{a}, q^{-1})$ are estimated in real time and indirectly. The GPC controller parameters $S(\hat{a}, \hat{b}, q^{-1})$, $R(\hat{a}, \hat{b}, q^{-1})$, $T(\hat{a}, \hat{b}, q^{-1})$ are updated (Ljung, 1999), using the well known least square algorithm with a fixed forgetting factor so as to ensure the closed loop stability and the desired performance (Msaad & Chebassier, 1992), (Bitmeat et al., 1990).

3.2 Direct Generalized Predictive Control concepts

A direct adaptive GPC, based on the work of (Wang & Henrisken, 1993), (Wang & Henrisken, 1994) is used with a direct identification of the controller parameters. In this approach, the GPC algorithm is included in an adaptive framework considering a direct scheme, directly updating the controller parameters. This strategy makes it necessary first to reformulate the polynomial GPC controller in adequate form.

3.2.1 Some basic GPC notations

For the Direct adaptive case, the prediction vector (10) is rewritten in the following form:

$$\hat{\mathbf{y}} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{if}\, y(t) + \mathbf{ih}\,\Delta u(t-1) \tag{16}$$

The minimization of the Eq.(12) written in a matrix form provides from the future control sequence:

$$\tilde{\mathbf{u}} = \mathbf{M}\left[\mathbf{w} - \mathbf{if}\, y(t) - \mathbf{ih}\,\Delta u(t-1)\right] \tag{17}$$

With: $\mathbf{if} = \left[F_{N1}(q^{-1})\ldots F_{N2}(q^{-1})\right]^T$; $\mathbf{ih} = \left[H_{N1}(q^{-1})\ldots H_{N2}(q^{-1})\right]^T$

$\tilde{\mathbf{u}} = \left[\Delta u(t)\ldots\Delta u(t+N_u-1)\right]^T$; $\hat{\mathbf{y}} = \left[\hat{y}(t+N_1)\ldots\hat{y}(t+N_2)\right]^T$

$\mathbf{w} = \left[w(t+N_1)\ldots w(t+N_2)\right]^T$

$$G = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \cdots & \cdots \\ g_{N_1}^{N_1} & g_{N_1}^{N_1+1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_1} & \cdots & g_{N_2-N_u+1}^{N_1} \end{bmatrix}$$

$$\mathbf{M} = \left[\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}_{Nu}\right]^{-1}\mathbf{G}^T \quad \text{of dimension} \quad (N_2 - N_1 + 1)\times N_u$$

The GPC controller is implemented under a RST form through difference equation:

$$S(q^{-1})\Delta u(t) = T(q^{-1})w(t) - R(q^{-1})y(t) \tag{18}$$

With: $S(q^{-1}) = 1 + \mathbf{m}_1\mathbf{ih}\,q^{-1}$, $R(q^{-1}) = \mathbf{m}_1\mathbf{if}\,q^{-1}$, $T(q^{-1}) = \mathbf{m}_1\left[q^{N_1}\ldots q^{N_2}\right]$.

3.2.2 Reformulation as performance index

A- Definition of the performance error.

Consider first the following regressor:

$$\Phi(t) = \left[y(t) \dots y(t - n_a)\ \tilde{\mathbf{u}}\ \Delta u(t-1) \dots \Delta u(t - n_b)\right]^T \tag{19}$$

With $\Phi(t)$ of dimension $(n_a + n_b + N_u + 1)$, and θ the parameter matrix:

$$\theta = \left[\mathbf{M\ if\ \ I_u\ M\ \ ih}\right]^T \tag{20}$$

The control law (17) stems from the new matrix form:

$$\mathbf{Mw} = \theta^T \Phi(t) \tag{21}$$

Let us now introduce the following predictive vector (predicted outputs between the horizons N_1 and N_2 and future control values up to horizon N_u):

$$\mathbf{X}(t + N_2) = \left[\hat{\mathbf{y}}\quad \tilde{\mathbf{u}}\ \right]^T \tag{22}$$

and the vector $\mathbf{Xw}$, of the same dimension $(N_2 - N_1 + N_u + 1)$, called target Vector. Considering the fact that the output vector $\hat{\mathbf{y}}$ has to converge to the reference vector $\mathbf{w}$ while the control signal $\tilde{\mathbf{u}}$ has to tend to zero, Xw is defined by:

$$\mathbf{Xw}(t + N_2) = \left[\mathbf{w}\quad \mathbf{0}\ \right]^T \tag{23}$$

Finally, a weighting matrix **L** is defined to create a cancellation dynamics of performance error so that the filtered error is the following:

$$\mathbf{e_f}(t + N_2) = \mathbf{iP}(t + N_2) - \mathbf{iPw}(t + N_2) = \mathbf{L}^T\left[\mathbf{X}(t + N_2) - \mathbf{Xw}(t + N_2)\right] = \mathbf{L}^T\mathbf{e} \tag{24}$$

With these definitions, $\mathbf{iP}(t + N_2)$ is an indication of the measured performances and $\mathbf{iPw}(t + N_2)$ an evaluation of the expected performances.

B- Performance index.

The performance index to be minimized is quadratic cost function $\Im$ defined by:

$$\Im(t + N_2) = \mathbf{e_f}(t + N_2)^T \mathbf{e_f}(t + N_2) = [\mathbf{X}(t + N) - \mathbf{Xw}(t + N)]^T\ \mathbf{L}\ \mathbf{L}\left[\mathbf{X}(t + N) - \mathbf{Xw}(t + N)\right] \tag{25}$$

From this, the objective in the adaptive case is to minimize this performance index $\Im$ at each step, in order to reach asymptotically and without plant parameter knowledge:

$$\lim_{x \to \infty} \mathbf{e_f}(t+1) = 0$$

Theorem

The fixed GPC control law explicitly cancels the performance index $\Im$ considering the nominal model. The $\mathbf{L}$ matrix is defined by:

$$\mathbf{L} = [\mathbf{M} \quad \lambda]^T = [\mathbf{Q}_2 \quad \mathbf{Q}_1]^T \tag{26}$$

Proof: See (Ramond et al., 1998).

Including the RST structure and the performance error, the DAGPC algorithm is represented in Fig.3

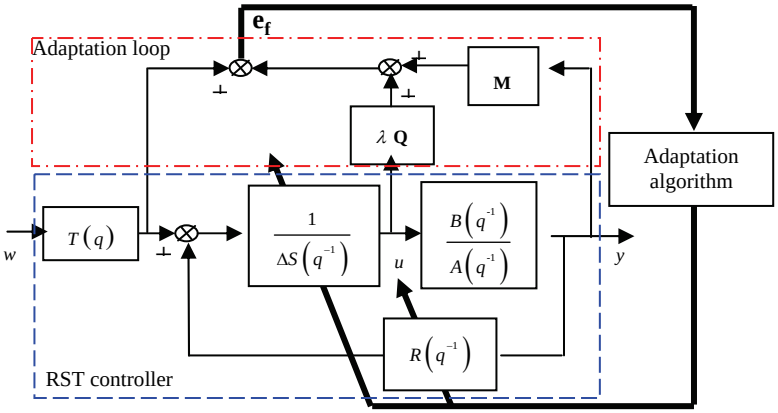


Fig. 3. Equivalent structure of the DAGPC

C- Least-squares identification

The previous section showed that the measured performances index is given by the relation:

$$\mathbf{iP}(t+N_2) = \mathbf{M}\hat{\mathbf{y}} + \lambda\mathbf{Q}\tilde{\mathbf{u}} \tag{27}$$

And the expected index by:

$$\mathbf{iPw}(t+N_2) = \mathbf{Mw} = \theta^T\Phi(t) \tag{28}$$

For the time varying parameters, the fixed controller parameters matrix θ must be moved to an estimated matrix $\hat{\theta}(t)$ (see Astrom and Wittenmark, 1989; Isermann, et al., 1992) to ensure that the same criterion $\Im$ always equals 0. The controller parameter matrix is updated according to a least squares-types method.

4. Real time results

For the Indirect or Direct strategy, the recursive identification and GPC code developed with Matlab® software were connected to the industrial automation via a local area network managed by interface developed with Delphi® software. A set of electronic units was used to apply heating voltage on the resistors or to control the DC motor and thus the Aperture opening rate. Measurements were performed using Pt100 sensors for temperature and encoder sensors for Aperture position. A sampling interval of Te=30 seconds was chosen to satisfy the predominant time constant, and data acquisition time was about twelve hours. The operating point (aperture opening) values interval was $x \in [0\%,100\%]$.

4.1 Indirect strategy

The different discrete models structure of the temperature of dry and humid ducts are given by:

$$\frac{T_{ODD}(k)}{U_{DD}(k)} = \frac{q^{-1}(b_{11}(k) + b_{12}(k)q^{-1})}{1 + a_{11}(k)q^{-1} + a_{12}(k)q^{-2} + a_{13}(k)q^{-3}} \tag{29}$$

$$\frac{T_{OHD}(k)}{U_{HD}(k)} = \frac{q^{-1}(b_{21}(k) + b_{22}(k)q^{-1})}{1 + a_{21}(k)q^{-1} + a_{22}(k)q^{-2} + a_{23}(k)q^{-3}} \tag{30}$$

Concerning the estimator algorithm, the models parameters were initialized by zero vectors and the covariance matrix $F(0) = 10^5$, with a fixed forgetting factor $\eta = 0.95$. In order to facilitate the convergence of the recursive estimation algorithm, a persistent sequence excitation (PRBS) was applied during the first 70 th sample times as can be seeing in figure 4 and figure 6, before running the generalized predictive control algorithm in real time. For the GPC algorithm controller, the control-weighting factor λ =0.97, the minimum prediction horizon was fixed at a value $N_1 = d = 1$, and the maximum prediction horizon $N_2 = 14$, with a control horizon $N_u = 7$. Parameter variation is shown in figures 5 and 7. More detailed information may be found in (Riadi et al., 2007).

Generally speaking, control performance was good, as shown by the IAGPC for different setpoint values. The temperature ducts are closed to the setpoints in figures 4 and 6. The figures generally show an efficient disturbance rejection. These disturbances, caused by the intake air temperature, are eliminated by the integral action existing in the CARIMA basic model. The dry duct controller cancels parametric perturbation due to the abrupt and significant change of aperture position.

The control strategy robustness was also observed through temperature overshoot rejection. This type of disturbance is caused by the aperture commutation (operating point system variations) which in reality affects the air rate flow variation. At 700 th sampling time in figure 6, the overshoots presented by the humid duct air temperature response result from the abrupt aperture opening commutation, which introduces a parametric error estimation and, consequently, instantaneous closed loop instability between the 800 th and the 900 th

sampling time. These can be explained by the non-persistence of the control signal in a steady state, causing the cross-correlation of the covariance matrix vectors, which leads to the estimator divergence.

In figure 8, the air temperature fluctuations do not appear between the 800 th and the 900 th sample time, such as in figure 6, because during this window of time, the humid duct was nearly closed (10%< *x*% <18%). As a result, its contribution to air mixing was reduced.

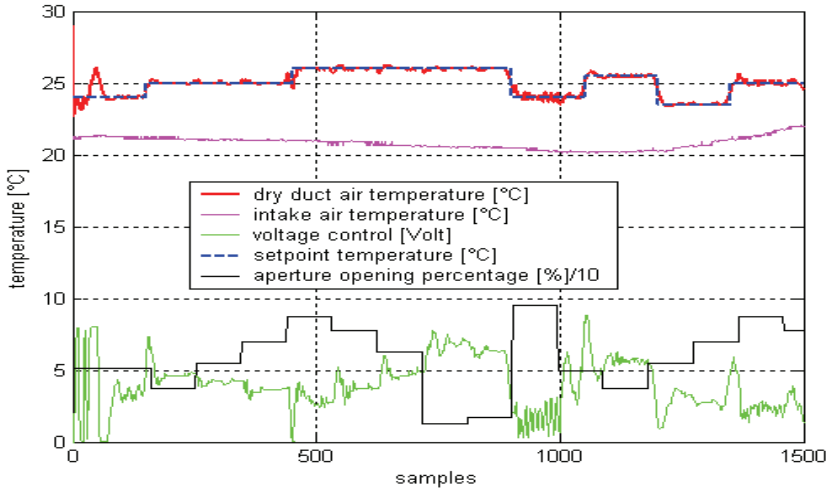


Fig. 4. IAGPC of the air dry duct temperature

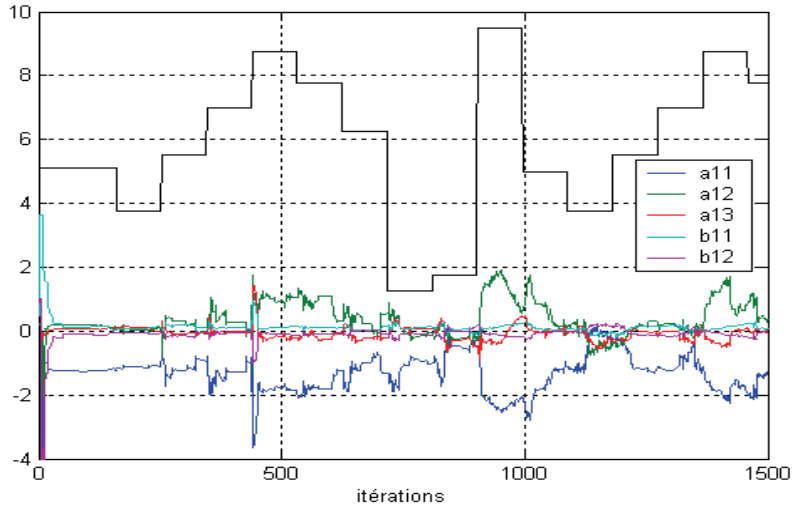


Fig. 5. Dry duct model estimated parameters

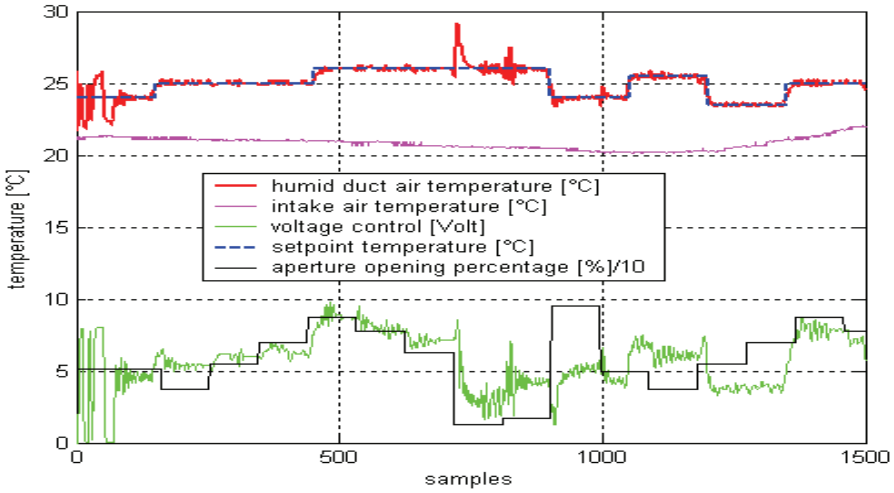


Fig. 6. IAGPC of the air humid duct temperature

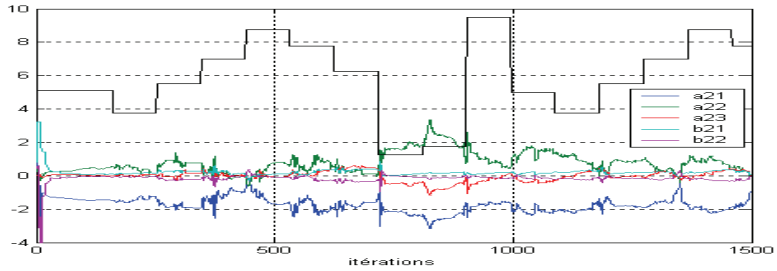


Fig. 7. Humid duct model estimated parameters

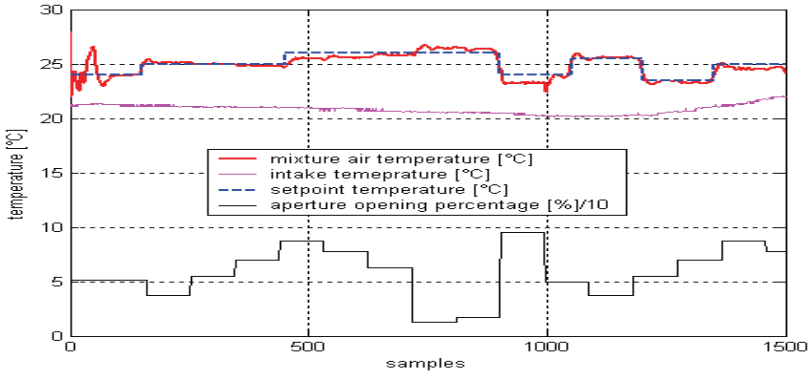


Fig. 8. IAGPC of the air mixture duct temperature

4.2 Direct strategy

Concerning the estimator algorithm, the sub-controllers parameter vectors were initialized by the nominal controller based on the nominal sub-model corresponding to 10% of aperture opening. The covariance matrix $F(0) = 10^5$, with a fixed forgetting factor, was $\eta = 0.96$. For the GPC algorithm controller, the control-weighting factor λ =0.97, the minimum prediction horizon was fixed at a value $N_1 = d = 1$, and the maximum prediction horizon $N_2 = 3$, with a control horizon $N_u = 1$.

Good control performance is also obtained with the DAGPC for different setpoint values. The temperature ducts are closed to the setpoints in figures 9 and 10. The figures show a generally efficient disturbance rejection. Compared to the previous strategy, both ducts are sensitive to abrupt and significant changes in aperture position.

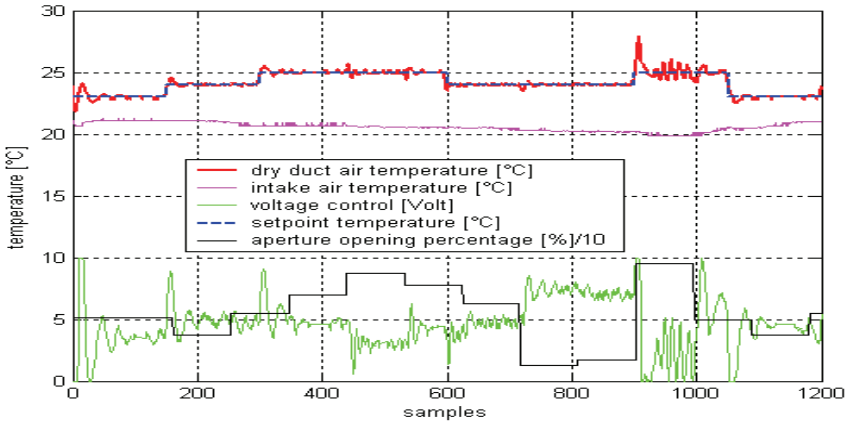


Fig. 9. DAGPC of the air dry duct temperature

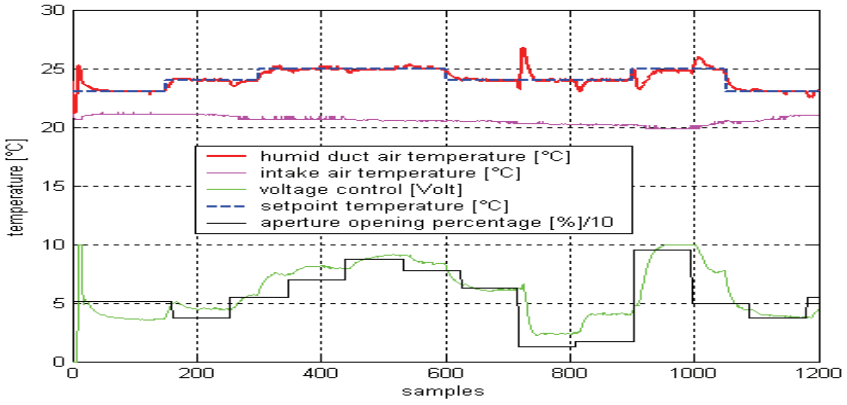


Fig. 10. DAGPC of the air humid duct temperature

The control strategy robustness may be observed, through temperature overshoots rejection. This type of disturbance is caused by the aperture commutation (operating point system variations) which in reality affects the air rate flow variation. At 900 th sampling time in figure 11, the overshoots presented by the humid duct air temperature response result from the abrupt aperture opening commutation, which introduces a performance error and, consequently, instantaneous closed loop instability between the 900 th and the 1000 th sampling time. These can be explained by the necessary time to update the parameter controllers, when the local controllers are trying to maintain the nominal closed loop performances index, in spite of parametric system variations.
In figure 9, air temperature fluctuations appear between the 900 th and the 1000 th sample time, such as in figure 11, because during this window of time the dry duct was nearly open ($x\%=90\%$), and it thus made a significant contribution to air mixing.
To sum up, the air mixture temperature setpoints are guaranteed indirectly as consequence of the accuracy of the two temperatures at the upstream ducts in figure 11, showing the feasibility of the proposed humid air thermodynamic strategy.

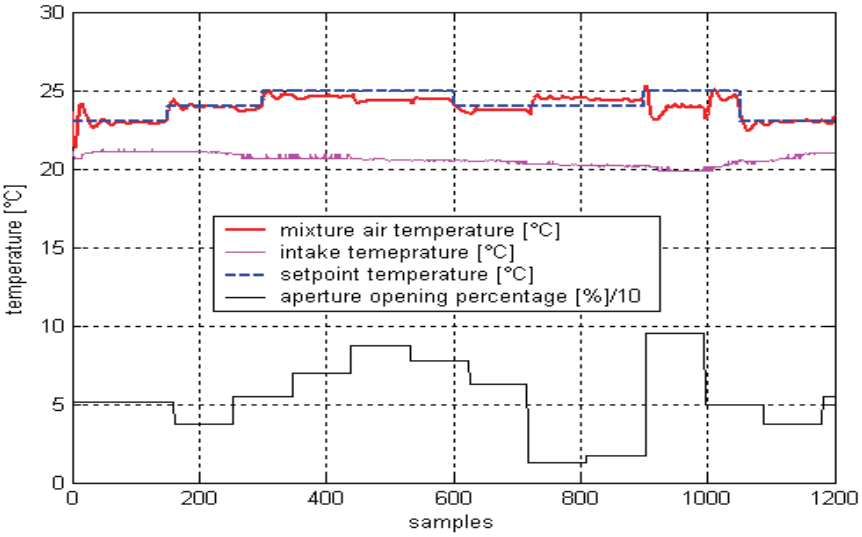


Fig. 11. DAGPC of the air mixture duct temperature

4.3 Humidity control

According to the air humid diagram (figure 1), the cascade strategy applied to the temperature and the relative humidity was intended to decouple these variables. This implies taking care of that relative humidity and temperature set points do not vary simultaneously. In order to validate the air conditioning output relative humidity, the nonlinear function between air and aperture position (Tawegoum et al.,2006b), combined with equation 1 and the relationship between absolute and relative humidity were used to defined the equivalent relative humidity set points.
Figure 12 illustrates the behaviour of relative air humidity in the mixing zone. It may be observed that, with little variation in the aperture position, the measurements are closed to

the set points. As the dynamics of the relative humidity are lower compared to those of the aperture, when the set point values vary permanently or vary abruptly, the tracking performances are too slow. Apart from these cases, one may consider the control to be good in a relative humidity range of between 75% and 95% since the errors remain in the standard deviation of the relative humidity sensors used.

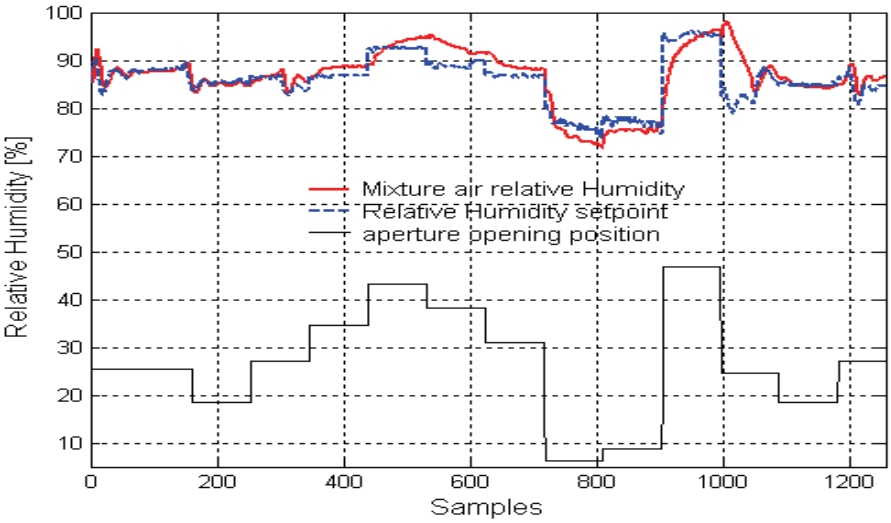


Fig. 12. Relative humidity of the mixture air

5. Conclusion

We have proposed a decentralized control scheme for the temperature and relative humidity control of a complex air conditioning unit. A local controller based on adaptive generalized predictive control theory has been designed for manipulating each sub-system with the objective of ensuring the setpoint temperature at the air mixture output under external and internal disturbances. The adaptive direct and indirect approaches proved to be good in both tracking and regulation. For the dry duct, the IAGPC also cancelled parametric variations, while for the humid duct both IAGPC and DAGPC reacted slowly to significant and abrupt variations. This is probably due to the time constant of the humid duct and to complex phenomena of mass and heat transfer between water and air, which takes place in the humid duct. These transfers also occur in the relative humidity in the mixing zone.
The close loop stability could be improved by taking into account a supervision level. Tracking performance may be enhanced by including the relative humidity dynamics in the control.

6. Acknowledgment

The authors would like to express their gratitude to Alain Travers for his valuable technical assistance.

7. References

Albright, Gates R.S; Aravantis, K.G & Drysdale, A.E. (2001). Environment Control for Plants on Earth and Space, *IEEE Control Systems Magazine*, October 2001,pp. 28-47.

Arguello-Serrano B. & Vélez-Reye M. (1999). Non linear control of heting, ventilating, and air conditioning system with thermal load estimation, *IEEE Tranactions on Control Systems Technology*, vol. 7, No. 1, January, pp. 56–63.

Bitmead, R. R.; Gevers, M. & Wertz, V. (1990). Adaptive Optimal Control, The thinking Man's GPC, Prentice-Hall, New York.

Buchholz, M.; Buchholz, P.; Jochum, P.; Zaragoza, G. & Pérez-Parra, J. (2006). Temperature and humidity control in the watergy greenhouse, *Proceedings International Symposium on Greenhouse Cooling*, Almeria-Spain, April 2006, pp. 401-407.

Camacho, E. F. & Bordons, C. (2002). Model predictive control, *Advanced textbooks in control and signal processing*, Springer 2nd Edition.

Chraibi, A.; Makhlouf, S. & Jaffrin, A. (1995). Refroidissement évaporatif de l'air des serres, *Journal de Physique*, n°III. Juillet 1995.

Clarke, D.W. (1988). Application of generalized predictive control to industrial process, *IEEE Control Magazine*, N° 8, pp. 49-55.

Clarke, D.W. ; Mohtadi, C. & Tuffs, P.S. (1987). Generalized predictive control- part I. The basic algorithm, *Automatica*, Vol 23, N°2, pp. 137-148.

Clarke, D.W. ; Mohtadi, C. & Tuffs, P.S. (1987). Generalized predictive control- part II. Extensions and interpretations, *Automatica*, vol 23 N°2, pp. 149-160.

Corrêa, N.A.; Corrêa, R.G. & Freire, J.T. (2000). Adaptive control of paste drying in spouted bed using the GPC algorithm, *Brazilian Journal of Chemical Engineering*, Sao Paulo dic, vol.17, N°4, pp.1-7.

Cunha, J.B. & Oliveira, J.P.M. (2003). Optimal management of greenhouse environments, *Proceedings of EFITA Conference*, Debrecem, Hungary, 2003, pp. 559-564.

Dion, J.M.; Dugard, L.; Franco, A.; Nguyen Minh Tri & Rey, D. (1991). Mimo adaptive contrained predictive control case study : an environmental test chamber, *Automatica*, vol 27, N°4, pp.611-626.

Dumur, D. ; Boucher, P. ; Murphy, K.M & Déqué F. (1997). On Predictive Controller Design for Comfort Control in Single Residential Housing, *Proceedings of ,ECC'97*, juillet 1997, pp. 109-114.

Filataov, N.M. & Unbedhauen, H. (2004). *Adaptive Dual Control: Theory and Applications* , Springer.

Hanan J.J. (1997). *Greenhouse: advanced technology for protected horticulture*, Chapter 4, pp. 236-260.

Jones, P. ; Jones, J.W. ; Allen, L.H. & Mishve, J.W. (1984). Dynamic computer control of closed environmental plant growth chambers, Design and Verification. *Transaction of ASAE. (American Society of Agricultural Engineers),* pp. 879-888.

Kittas, C.; Bartzanas, T. & Jaffarin, A. (2003). Temperature gradient in a partially shaded large greenhouse equiped with evaporative cooling pads, *Biosystems Engineering*, vol 85, N° 1, pp. 87-94.

Landau, I.D. & Dugard, L. (1986). *Commande adaptative aspects pratiques et théoriques*, J. Masson, Ed. Paris, pp. 1–81.

Ljung, L. (1999). *System identification : Theory for the user*, Prentice Hall.

M'ssad, M. & Chebassier, J. (1992). Commande adaptative des systèmes, *Techniques de l'ingénieur*, vol. 7426, pp. 1–25.

Montero, J.J.; Anton, A.; Beil, C. & Franquet, A. (1994). Cooling of greenhouses with compressed air fogging nozzles, *Acta Horticulturae*, N°281, pp. 199-209.

Nybrant, T.G. (1989). Modelling and adaptive control of concurrent flow dryers, *Computers and Electronics in Agriculture*, vol 3, pp. 243-253.

Rafilamanana, A.; Cabassud, A.; LeLann, M. V. & Casamatta, G. (1992). Adaptive control of a multipurpose and flexible semi-batch pilot plant reactor, *Comp. Chem. Eng*. Vol 16 N°9, pp.837-848.

Ramond, E.; Rumur, D. & Boucher, P. (1998). A direct adaptive generalized predictive control structure : Application to a flexible structures, *Proceedings of IFAC syposium on Adaptive system in control and signal processing*, August, Glasgow, Scootland, pp.351-356.

Riadi, R.; Tawegoum, R.; Rachid, A. & Chassériaux, G. (2006). Modeling and Identification of a Passive Air-Conditioning Unit using the Operating Dependent Parameters-Structure, *Proceedings of CESA-2006: Computational Engineering in Systems Application Conference*, Beijing, Chine-4-6 Octobre 2006, pp. 1485-1491.

Riadi, R.; Tawegoum, R.; Rachid, A. & Chassériaux, G. (2007). Denectralized temperature indirect adaptive predictive control of a passive air conditioning unit. *Proceedinds of the 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, Dec 12-14, 2007, pp 1808-1813

Richalet, J. ; Rault, A. ; Testud J. & Papon J. (1978). Model predictive heuristic control: Applications to industrial processes, *Automatica*, 1978, pp.413-428,.

Sethi, V.P. & Sharma, S.K. (2007a). Experimental and economic study of a greenhouse thermal control system using aquifer water, *Energy Conversion and Management*, vol 48, N° 1, pp. 306-319.

Sethi, V.P. & Sharma, S.K. (2007b). Thermal modeling of a greenhouse integrated to an acquifer coupled cavity flow heat exchanger system, *Solar Energy*, vol 81, N°6 pp. 723-741.

Sethi, V.P. & Sharma, S.K. (2007c). Survey of cooling technologies for wordwide agricultural greenhouse applications, *Solar Energy*, N° 81, pp. 1447-1459.

Tawegoum, R.; Bournet, P.E; Arnould, J.; Riadi, R. & Chassériaux, G. (2006b). Numerical investigation of an air conditioning unit to manage inside greenhouse air temperature and relative humidity, *Proceedings International Symposium on Greenhouse Cooling*, Almeria-Spain, April 2006, pp. 115-122.

Tawegoum, R.; Teixeira, R. & Chassériaux, G. (2006a). Simulation of humidity control and temperature tracking in a growth chamber using a passive air conditioning unit, *Control Engineering Practice Journal*, 2006, vol 14/ N° 8, pp. 853-861.

Wang, W. & Henrisen, R. (1993). Direct adaptive generalized predictive control, *IEEE Transaction on Modelling, Identification and Control*, vol 14, pp.181-191.

Wang, W. & Henrisen, R. (1994). A direct adaptive generalized predictive control, *IEEE Transaction on Modelling, Identification and Control*, vol 15, pp.23-32.

Willis, D.H. & Peet, M.M. (2000). Intermittent application of water to an externally mounted greenhouse shade cloth to modify cooling performance, *Transactions of ASAE*, vol 43, N° 5, pp. 1247-1252.

Young, P.C. & Lees, M.J. (1994). Simplicity out of complexity in glasshouse climate modelling, *Proceedings of 2nd IFAC/ISHS Workshop on Mathematical an Control Application in agriculture and Horticulture*, 12-15 september 1994, Silsoe, United kingdom, Acta Horticuturae N°406, pp.15-28.

13

Development of a Human-Friendly Omni-directional Wheelchair with Safety, Comfort and Operability Using a Smart Interface

[1]Kazuhiko Terashima, [2]Juan Urbano, [3]Hideo Kitagawa, and [1]Takanori Miyoshi
[1]*Toyohashi University of Technology*
[2]*Denso Preas*
[3]*Gifu National College of Technology*
Japan

1. Introduction of the OMW

A variety of wheelchairs with different options and special add-on features have been developed to meet a wide range of needs (Pin & Killough, 1994), (Wada & Asada, 1999), (West & Asada, 1992). In order to satisfy the demand for higher mobility, designers have created new driving concepts such as omni-directional movement which allows any combination of forward, sideways, and rotational movement, thus ensuring users much more freedom and safety in wide or narrow spaces.

Autonomous electric wheelchairs are very useful for people who cannot move their upper bodies freely. However, these wheelchairs need to be fitted with a central control unit and high-level sensors capable of realizing complex navigation and obstacle avoidance tasks, based on a description of the environment and final goals marked out by those sensors. Since autonomous wheelchairs can function well only in special environments, this mode greatly limits the user's freedom.

In order to offer users with a higher degree of independence, the user-controlled movement mode, or semi-autonomous mode, which is operated under absolute user control by an input device such as a joystick, switch, monitor, etc., has been developed. The main difference between autonomous and semi-autonomous systems is that in semi-autonomous systems users interact in real time to perform certain tasks in dynamic environments. Under user control, the wheelchair can go wherever the user wants it to. Therefore, this mode provides a high degree of user independence.

However, it is necessary to keep in mind that some elderly people or handicapped people can not use their arms due to weakness or injury. These people need the help of an attendant. In developed countries in which the number of young people is declining yearly, some healthy elderly people are taking care of other elder or handicapped people. For these attendants, a system that helps them to push the wheelchair and its occupant would be very convenient.

Years ago, the main purpose of research was to develop reliable systems without showing great concern for the comfort of the user when employing them. However, with the advent of ergonomics or "the systematic application of knowledge about the psychological, physical, and social attributes of human beings in the design and use of all things which affect a person's working conditions: equipment and machinery, the work environment and layout, the job itself, training and the organization of work", designers have become more aware of the importance of the user when designing any device. Comfort, or "a state of being relaxed and feeling no pain, when using a piece of equipment" emerged as a design goal. This is especially true in the case of wheelchairs where the occupants are weak people because of age or disease.

Therefore, the development of an omni-directional wheelchair that can provides the occupant with semi-autonomous functions and comfort; and the attend with power assist support is highly desirable.

When considering about wheelchairs, it is necessary to remember that they can be classified in two main groups: manual wheelchairs or wheelchairs that move due to the application of force by the occupant, and electric wheelchairs, or wheelchairs that employ electric energy for generating movement. Just the latter are the object of interest for this research.

In order to offer users with a higher degree of independence, the user-controlled movement mode, or semi-autonomous mode, which is operated under absolute control of users by an input device such as joystick, switch, monitor, etc., has been developed. Under control of users, wheelchair can go wherever users want to go. Therefore, this mode provides a great independence to users. For achieving reliable navigation, obstacle detection and collision avoidance must be considered when designing a wheelchair. In the case of semi-autonomous wheelchairs, most of them rely on reactive obstacle avoidance (Argyros et al., 2002), (Borgolte et al., 1998), (Levine et al., 1999), (Tahboub, 2001), (Yanco et al., 1995) which is in some degree safe but uncomfortable for the user if he is not aware of the obstacle and he is unexpectedly taken away from it. "Not being aware" means that the user is not giving attention to the environment or maybe he is a blind one. It means that environment information must be provided in a way that it can be perceived without using the eyes. Most of semi-autonomous wheelchairs use joysticks as input devices then it appears natural to provide environment information to users through joystick. That is, joystick becomes a haptic device, or a device that provides information through the sensation of touch.

Many power-assisted wheelchairs have been developed for handicapped people who have free use of their arms. Power assist is useful for reducing the burden of manual workers and elderly people. In recent years, it is necessary for elderly people to support other elderly people. Research on power-assist systems has been widely reported (Hayashibara et al., 1999), (Kawai et al., 2004), (Kumar et al., 1997), (Lee et al., 1999), (Naruse et al., 2005) and much study has been devoted to wheelchairs (Sanada et al., 2005), (Seki et al., 2005), (Wu et al., 2004) regarding the chair's straight-line forward and backward movement. However, in spite of its importance, little study has focused on power-assist with respect to rotation, lateral and slanting movements. The application of power-assist for supporting the attendant of an omni-directional wheelchair constitutes a new area of research. Though some research regarding a power-assist system for omni-directional vehicles related to carts is available (Maeda et al., 2000), no report regarding this topic as related to wheelchairs has appeared, to the authors' knowledge. In the case of the Omni-directional Cart with Power-assist System developed by Matsushita Electric Works (Maeda et al., 2000) a cart with the

length being bigger than the width is considered. In this case, they report problems with lateral motion when the length of the cart is very big. Moreover, it looks like they have considered turning, but not rotation over the center of gravity because it could not be possible due to the dimensions of the cart. On the other hand, achieving almost perfect rotation over the center of gravity and high accuracy in lateral and forward-backwards motion is a very important goal in this research.

In recent years special attention has been given to ride comfort. When ride comfort is studied, most of them consider only influences of vibration in up-down direction caused by uneveness of the ground. Various alternatives have been mooted to solve this problem, however, most of them are based on more or less complex mechanical solutions such as soft cushions, vibration absorber, etc. (Sato et al., 2003). With the use of these advanced equipments, the cost has been increased accordingly and, moreover, it also usually increases the weight of the device. According to literature, when automobile drivers were asked about parameters necessary to driving comfort, they mentioned factors such as a well designed seat, adjustable features, correct temperature, ease of reaching controls and pedals, enough space, low noise level as well as vibration-free riding.

For wheelchair users, the factors related to comfort are almost the same as those mentioned above. In fact, two main problems are considered when designing a comfortable wheelchair: comfortable seat design and suppression of vertical vibration caused by rough pavement or the wheelchair's mechanical elements. However, there is another factor that must be considered: vibration due to jerking, or the variation of linear acceleration (Seki et al., 2005). When the natural frequency of this vibration synchronizes with the natural vibration frequency of human beings, the resonance phenomenon causes large oscillations and therefore leads to the discomfort of the wheelchair's occupant. In past studies (Matsuoka, 2000a), (Matsuoka, 2000b), (Nishiyama, 1993), (Okada, 1980), (Smith, 2000), the user's upper body is considered as a series of rigid segments (head, chest, waist) connected by flexible joints. Each joint is given a rotational spring constant (RSC) and a rotational viscous damping constant (RVDC). However, since waist is connected to the seat directly, its swing frequency is much less than that of head and chest. Then, in this research, in order to simplify the model, it is considered that upper human body consists of two rigid segments: head and torso. Moreover, in this research the human model is used for studying vibration of the human body when it moves in a horizontal plane, while previous researchers have used a human model for studying the problem of vibration of the human body when it moves in a vertical direction.

In author's laboratory, a holonomic Omni-directional Wheelchair (OMW) which can act as an autonomous (Kitagawa, Terashima et al., 2002), semi-autonomous (Kitagawa, Terashima et al., 2001) or power assisted (Kitagawa, Terashima et al., 2004) wheelchair has been developed. Because of its omni-directional movement, it is able to navigate smoothly in structured inner environments using range sensors for getting environment information. In order to recognize surrounded environment it can build a local map which provides distance to nearest obstacles. In semi-autonomous mode the input device is a joystick, with velocity of OMW being proportional to the angular displacement of the joystick. In previous research, just the idea of haptic feedback was proposed (Tahboub, 2001), or a combination of a haptic joystick with a virtual simulator for navigation was used (Protho et al., 2000). In this research, a wheelchair provided with a haptic joystick has been built. Moreover impedance of joystick or force feedback changes according not only to velocity of OMW but also to the

distance of OMW to the nearest obstacle in the direction of movement. The nearest the distance becomes, the more difficult it becomes to move the joystick, then the user understands that he is going to collide against an obstacle soon an he can decide to change the direction of movement or to stop OMW. The proposed approach by haptic joystick has been tested with good results (Kitagawa , Terashima et al., 2001) , and furthermore, a novel navigation guidance system to induce evasive movement, while the omni-directional wheelchair performs slide movement without rotated movement, using the haptic feedback joystic is proposed.. The obstacle existing toward the moving direction of vehicle has possibility of collision. Therefore, when the obstacle exists in the direction of the OMW's movement, this approach gives the joystick force to operator's hand such that induces evasive movement to navigate OMW toward the direction without obstacle for operator's safe and smooth driving. The purpose of this study gives a support system to realize operator's safe and smooth driving when the operator passes through the narrow aisle, entrance of room, or enters an elevator,etc.(Kondo, et al. 2008).

In the power assisted mode, in the authors' laboratory (Kitagawa, Terashima et al., 2004) a six-axis force sensor is used for measuring the force applied by the attendant in two orthogonal axes, X and Y, and a rotational direction θ. This force is then changed to reference velocity V_x, V_y, and ω by using a first-order lag controller. Finally, the reference velocity is applied to the servo-motors of the OMW. This system works well as a power-assist system, and provides the attendant effective support. However, a problem related to the operability of the OMW remains. Due to the application of the power-assist system, the operability of the OMW diminishes especially when the attendant tries to rotate the chair in a clockwise (CW), or counter-clockwise (CCW) direction around the OMW's center of gravity CG.

A survey was conducted among various attendants trying to discover some relationships in the way they developed forwards-backwards, lateral, and rotational movements. It was impossible to find general rules that explained all cases, but a relationship was found between lateral and rotational movements. These relationships were used as the basis for constructing a fuzzy reasoning system (Mathworks, 2002), (Harris et al., 1993), (Mamdani & Assilian, 1985), (Shaw, 1998), (Sugeno & Kang, 1998), (Takagi & Sugeno, 1985) that helped to improve the operability of the OMW. An expert operator can move the OMW with ease, but for people not accustomed to its use it is difficult to easily manage in any direction. Thus, the development of a novel power-assist system with easy operability is strongly demanded. In this paper, this system is called a "skill-assist system" (Yamada et al., 2002).

Nevertheless, when the system was tested by different attendants, a completely satisfactory result was not obtained by every attendant, since each operator has his or her own tendencies, and thus the parameters of the fuzzy inference system must be reasonably tuned to respond to the individual. Tuning the fuzzy inference system by trial and error was thus attempted and its results reported in (Kitagawa, Terashima et al., 2004). This was found to be a time-consuming process, leading to fatigue and boredom in the attendants.

Hence, a better tuning method, a method that allows tuning of the fuzzy inference system, is needed. This can be obtained by adding Neural Networks (NN) to the fuzzy inference system, obtaining what is known as a neuro-fuzzy system. Much research has been devoted to this topic (Jang, 1993), (Jang et al., 1997), (Juang & Lin, 1998), (Lian et al., 1999), (Lin & Lee, 1991), (Nguyen et al., 2003). Jang (Jang, 1993) developed the ANFIS (Adaptive-Network-based Fuzzy Inference System), a neuro-fuzzy system in which the fuzzy inference

system is tuned by using the system's input data. Tuning is performed by minimizing the output error of the NN used in combination with the fuzzy inference system. For achieving this goal, the NN is trained by using a hybrid method that combines least squares and the Backpropagation algorithm (BP law). This method is thus thought to be an effective method for tuning the parameters of the OMW's fuzzy inference system.
In the case of comfortable navigation, in the authors' laboratory comfort has been a subject of study in the autonomous mode without the joystick (Terashima et al., 2004) but only when the OMW moves in a single direction, X or Y. In the present paper, comfort is studied when the OMW moves in any direction, such as a slanting direction, when in practical semi-autonomous operation mode using the joystick (Urbano et al., 2005b). For the command input via joystick operation, the velocity control of the OMW is carried out by means of frequency shaping using the Hybrid Shape Approach (HSA) proposed by the authors (Yano et al., 2000) in order to achieve the swing suppression control or comfortable driving by excluding a specific spectrum of elements such as the natural frequency of the OMW and the discomfort frequency of human organs. A human model which considers the human upper body to be composed of two parts, the torso and head, has been developed and used in order to test the effectiveness of the proposed approach.

In this chapter, in section 2, kinematics and dynamics of the OMW are described. In section 3, semi-autonomous operation and guidance system of the OMW for obstacle avoidance by haptic joystick. In section 4, power asist control of OMW for helper and, in section 5, adaptive control by Neuro-Fuzzy system of OMW using a touch Panel as human interface for realizing tailor-made vehicle. In section 6,comfort driving of the OMW and conclusion in section 7.

2. Kinematics and dynamics of the OMW

The OMW, which can act as an autonomous (Kitagawa et al., 2002), semi-autonomous (Kitagawa et al., 2001) or power assisted (Kitagawa et al., 2004) wheelchair is shown in Fig. 1, and its specifications are shown in Table 1.

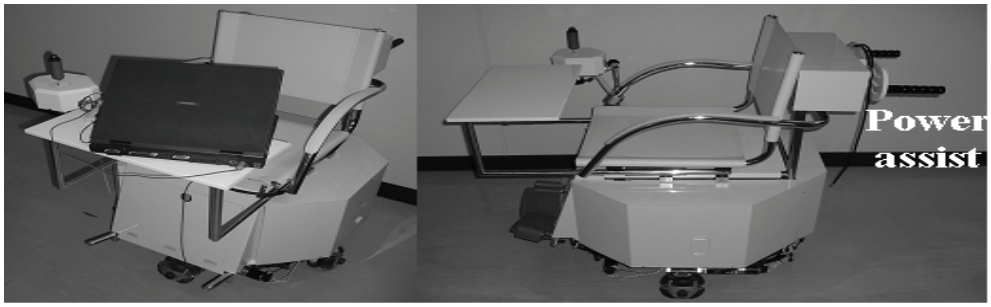


Fig. 1. Omni-directional wheelchair (OMW)

The OMW is able to move in any arbitrary direction without changing the direction of its omni-wheels, which are shown in Fig. 2. In this system, four omni-directional wheels are

individually driven by four motors, as shown in Fig. 3. Each wheel has passively driven free rollers at its circumference. The wheel that rolls perpendicularly to the direction of movement does not stop its movement, because of the passively driven free rollers. These wheels thus allow movement that is holonomic and omni-directional.

Size	Width	0.70 [m]
	Depth	1.50 [m]
	Height	1.10 [m]
Weight		70 [kg]
Maximum velocity		1.0 [m/s]
Maximum accelaration		0.5 [m/s^2]

Table 1. Specifications of the OMW

Fig. 2. Omni-wheel

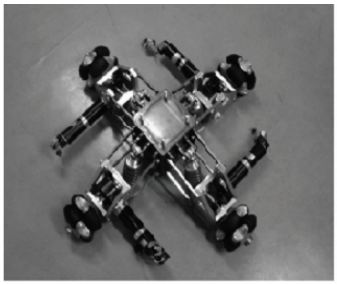


Fig. 3.Omni-wheels and motors

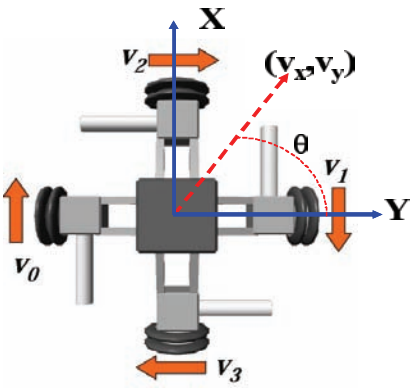


Fig. 4. Velocity vectors of the omni-wheels

In the coordinate system of the OMW, the X-axis is defined when the OMW moves forward or backward. The Y-axis is defined when the OMW moves to the right or left, and rotational

direction is defined according to direction θ perpendicular to the plane determined by X and Y. The joystick's coordinate system is established in the same way as that of the OMW. Furthermore, if V_x is the velocity of the OMW in the X-axis, V_y the velocity of the OMW in the Y-axis, and ω the angular velocity of the OMW when it rotates around the vertical axis, the velocity of the OMW can be expressed as $V_{OMW} = [V_x, V_y, \omega]^T$. The velocity of the OMW is the vectorial sum of the velocities of the four omni-directional wheels. The velocity vector for the omni-wheels is written as $V_{WHEEL} = [V_0, V_1, V_2, V_3]^T$. The velocity vectors corresponding to each omni-wheel are shown in Fig. 1. In Fig. 4, θ is the angle that the velocity vector of the OMW has with the axis Y of the reference system.

From Fig. 4:

$$V_x = \frac{1}{2}(V_0 - V_1) \tag{1}$$

$$V_y = \frac{1}{2}(V_2 - V_3) \tag{2}$$

$$\omega = \frac{1}{4l_{\omega b}}(-V_0 - V_1 - V_2 - V_3) \tag{3}$$

, where $l_{\omega b}$ is the distance from the center of the OMW to the circumference of the omni-wheels. Written in a matrix form, the above equations become:

$$V_{OMW} = B \cdot V_{WHEEL} \tag{4}$$

where

$$B \equiv \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{4l_{\omega b}} & -\frac{1}{4l_{\omega b}} & -\frac{1}{4l_{\omega b}} & -\frac{1}{4l_{\omega b}} \end{bmatrix} \tag{5}$$

Since generally a matrix should be square in order to calculate its inverse, the coefficients' matrix in Eq. (4) should be square in order to calculate V_{WHEEL} from V_{OMW}. Keeping this in mind, the angular velocity ω of OMW is divided into two parts: ω_1 produced by V_0 and V_1, and ω_2 produced by V_2 and V_3.

$$\omega_1 = \frac{1}{2l_{\omega b}}(-V_0 - V_1) \tag{6}$$

$$\omega_2 = \frac{1}{2l_{\omega b}}(-V_2 - V_3) \tag{7}$$

$$\omega = \frac{1}{2}(\omega_1 + \omega_2) \tag{8}$$

By using Eq. (6), Eq. (7) and Eq. (8), it is possible to get:

$$\begin{bmatrix} V_x \\ V_y \\ \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2l_{\omega b}} & -\frac{1}{2l_{\omega b}} & 0 & 0 \\ 0 & 0 & -\frac{1}{2l_{\omega b}} & -\frac{1}{2l_{\omega b}} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{bmatrix} \tag{9}$$

V_{OMW} can be expressed by:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \omega_1 \\ \omega_2 \end{bmatrix} \tag{10}$$

For avoiding the slippage of the wheels, the constraint $\omega_1 = \omega_2$, or $V_0 + V_1 = V_2 + V_3$ is imposed. By considering $\omega_1 = \omega_2$, Eq. (9) can be expressed as follows:

$$V_{WHEEL} = B^{*-1} \cdot V_{OMW} \quad , \tag{11}$$

, where

$$B^{*-1} \equiv \begin{bmatrix} 1 & 0 & -l_{\omega b} \\ -1 & 0 & -l_{\omega b} \\ 0 & 1 & -l_{\omega b} \\ 0 & -1 & -l_{\omega b} \end{bmatrix} \tag{12}$$

Here, $\mathbf{B}^{*-1}$ is a pseudo-inverse matrix that allows the velocity of each wheel to be obtained based on the velocity of the OMW.

3. Semi-autonomous Operation and Guidance System of the OMW for Obstacle Avoidance by Haptic Joystick.

The OMW is able to move in any arbitrary direction without changing the direction of the wheels. In this system, four omni-directional wheels are individually driven by four motors.

Each wheel has passively driven free rollers at its circumference. The wheel that rolls perpendicularly to the direction of movement does not stop its movement because of the passively driven free rollers. Thus, these wheels allow holonomic and omni-directional movement.

The obstacle detection sensors are activated in back and forth of the OMW in order to obtain information regarding its surrouding environment. In this research, the input device is a joystick. The direction of the vehicle movement depends on orientation of the joystick and speed is proportional to declination of the joystick. Moreover, two motors are installed in each x and y axis of the joystick, and the joystick can give virtual spring damper characteristics because of the impedance control. (Kitagawa et al, 2001), (Urbano et al, 2005a).

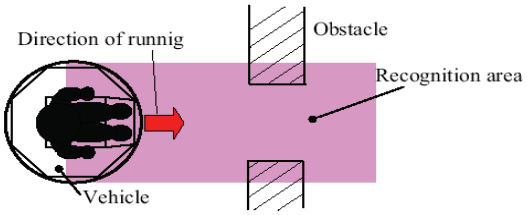


Fig. 5. Schematic diagram of environmental recognition and navigation

It is possible to acquire the surrounding environmental information in real time by two obstacle detection sensors. The obstacle with danger collision exists in the direction of OMW's running with current input. Therefore, the algorithm that choses only environmental information which exists in the direction of the OMW's running from all of the obtained information is constructed. Figure 5 shows the otuline of the environmental recognition system. The recognition area can be changed, corresponding to the direction of OMW's running.

Detection range	20~4000 mm
Accuracy	± 10 mm
Angle range	240°
Angular resolution	0.36°
Response time	100 ms

Fig. 6. Specification of obstacle detection sensor

The specifications of the obstacle detection sensor "URG-04LX" (Hokuyo Automatic Co. Ltd.) are shown in Fig. 6. This sensor gives real-time updates of the surrounding environment. The sensor covers a wide angular range. The maximum angular resolution on

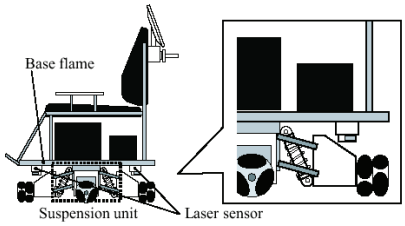


Fig. 7. Configuration of obstacle detection sensor (1)

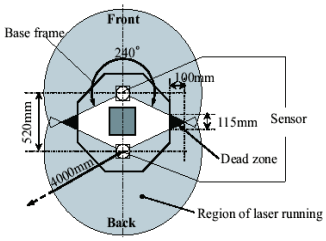


Fig. 8. Configuration of obstacle detection sensor (2)

the sensor is 0.36 [deg] within range of 240 [deg]. And measurement range of the sensor is 20-4000 [mm]. Furthermore, it only takes 100 [ms] for one complete scan. In this research, the angular resolution is applied 3.6 [deg] for the brevity of calculations. Two sensors are attached on frontside and backside of the base frame, where the control unit, the chair and the battery is installed, as shown in Fig. 7. It is possible to get the surrounding environmental information without shelding matters, because the unit above the base frame is braced with the suspension unit at the OMW's center, as shown in Fig. 4. Then, there are small dead areas in laser sensors' scanning in OMW's right and left sides, as shown in triangle black zone in Fig. 8. These dead areas are regarded as a part of OMW as described later. The total output response time to capture environmental map is 200 [ms] in this system, because of using two obstacle detection sensors. Take into account of the velocity of OMW's movement and velocity response of OMW, the response time is of no matter.

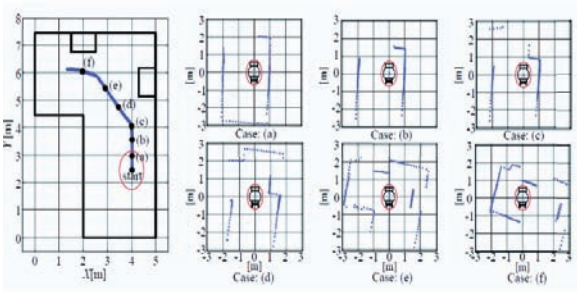


Fig. 9. Experimental result of obtaining environmental information

Figure 9 shows experimental results of obtaining environmental information using the obstacle detection sensor, while human operates the joystick for navigation. In this experiments, the OMW is manually controlled using joystick by operator. Note that, the obstacle and wall is recognized as points, and the map described by points is rotated while OMW is skew movement. It should be considered that this phenomenon is caused by slipping of the omni-wheel ad differences in the diameter of each four omni-wheels.
In this section, the algorithm that choses only environmental information existing toward the moving direction of the OMW from all the surrounding environmental information is presented.

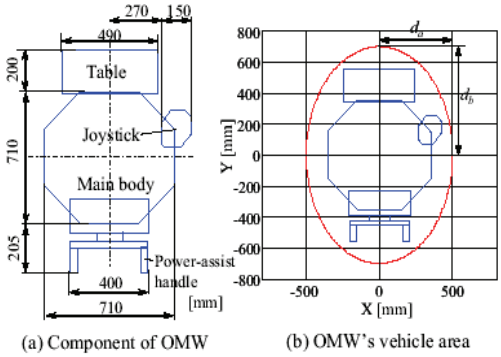


Fig. 10. Definition of OMW's vehicle area

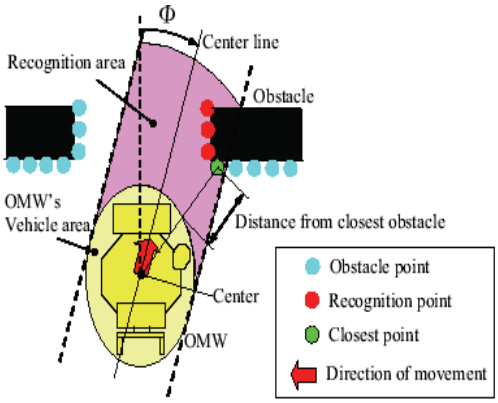


Fig. 11 . The environmental recognition system

First, the OMW's vehicle area which covers the entire OMW is defined. Since the OMW has a table, the joystick, and the power-assist handle, the OMW's vehicle area is defined as the shape of an ellipse covering both parts of OMW's main body and the dead areas of laser scanning as shown in Fig. 7(a). The length of shorter axis of this ellipse is d_a = 0.5 [m] and

longer axis is d_b = 0.7 [m], as shown in Fig. 7. If the obstacle goes inside the OMW's vehicle area, it is considered as a crashing obstacle. Next, an algorithm such that only environmental information existing toward the moving direction of the OMW is chosen, is proposed. Now the OMW is slide moving in direction Φ, as shown in Fig. 11. Then the surrounding environmental information is acquired as a set of points by two obstacle detection sensor. These points set are defined as the obstacle points. The straight line drawn by extending the moving direction from the OMW's center point is defined as the center line. The area between two lines such as being parallel to the center line and tangent to the oval vehicle area, is defined as the recognition area. And the obstacle points existing in the recognition area are defined as the recognition points, The closest recognition point to the OMW's vehicle area is defined as the closest point. Thus, the obstacle existing in the recognition area has danger of collision with OMW. By means of this algorithm, the present system can recognize only the obstacle with possibility of collision of the present moving direction, and teach the distance to the closest recognition point to the OMW. And the environmental recognition system achieves high speed response time to capture environmental map, compared with velocity response of OMW. Therefore, if navigator suddenly changes the joystick's direction, navigator's safe is assured with the presented system.
Derivation of the recognition area is explained in more detail. Now, the vehicle is moving in the direction Φ. A perfect circle shape is defined as the vehicle area. In this algorithm, the vehicle area needs to be perfect circle, but the OMW's vehicle area is really ellipse shape. Therefore, in order to change the shape of OMW's vehicle area into a perfect circle shape, the scale of X-axis or Y-axis is temporarily altered while picking the recognition point. For example, in this study case, all of Y-coordinate value of obstacle points times d_a/d_b, because the OMW's vehicle area is defined as shape of an ellipse, and the length of shorter axis is d_a = 500 [mm] on the X-axis and longer axis is d_b = 700 [mm] on the Y-axis, as shown in Figure 7. And movement direction Φ is converted into Φ′ by following equation (dash ″, ″ indicates the transformed scale of X-Y axis).

$$\Phi' = \tan^{-1}\left(\frac{\sin\Phi}{d_a / d_b \cdot \cos\Phi}\right) \tag{13}$$

First, the circumstance of the vehicle is divided into the layer of l thick ($L1;L2;\ldots;Ln$) and the obstacle points are stored in these layer. Next, the range of angular is derived in several layers such as

$$\Phi' - \theta_n' \le \Theta_n' \le \Phi' + \theta_n', \quad (n = 1,2,3...n) \tag{14}$$

$$\theta_n' = \sin^{-1}\left(\frac{r + nl}{r}\right) \tag{15}$$

The range of angular Θ_n' is indicated the range of the recognition area on n^{th} layer. If the obstacle point exists in the recognition area, this point is defined as the recognition point.

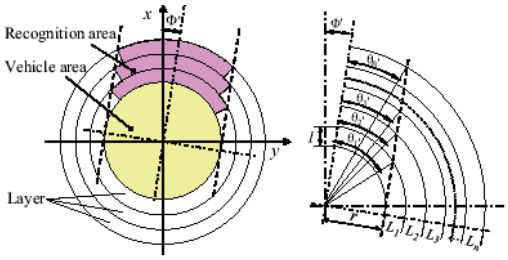


Fig. 12. Definition of recognition area

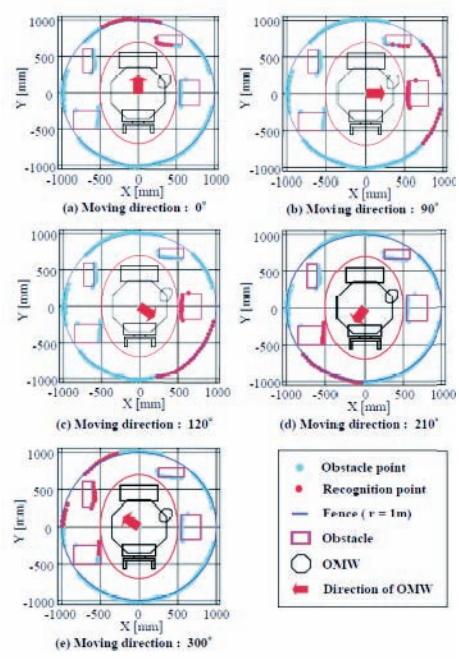


Fig. 13. Experimental results of environment recognition system

After picking the recognition points from all of the obstacle point with the use of above algorithm, all of the Y-coordinate value of obstacle points and recognition points times d_b/d_a in order to reconvert the transformed scale of X-Y axis. The experiment of the environmental recognition system with the recognition area is conducted. The OMW is installed at the center of the fence of radius 1 [m], and four obstacles are installed between the OMW and the fence. The fence is made of white board. The result of this experiment is shown in figure 10. Then, the thickness of the layers is l = 50 [mm]. Note that, the width of the recognition area is changed to fit the OMW's vehicle area defined as shape of an ellipse, and this system can pick out the recognition points from the obstacle points.

Two motors are installed in each x and y axis of the joystick as shown in Fig. 14, and the joystick can give virtual spring-damper characteristics with the impedance control. Based on the distance to the closest obstacle and the wheelchair's input velocity, the impedance of the joystick is provided. By neglecting the effect of joint mechanical compliance and link flexibility, the desired elastic behavior can be described as

$$\tau = d\dot{q} - kq \tag{16}$$

where τ is the joystick's motor torque, d is viscous damping coefficient, k is the stiffness and q is the tilting angle from the neutral position. The viscous damping coefficient is d = 0.015. The desired stiffness k in the input direction of the joystick is described as the following equation, as explained in (Kitagawa et al, 2001).

$$k = k_0 \left\{ \frac{v / v_{\max} + \alpha}{(r / r_{\max})^2} + 1 \right\} \tag{17}$$

where v is the input velocity of the wheelchair, and r is the distance to the nearest obstacle in the input direction. The maximum input velocity is v_{max} = 1.0 [m/s], and the maximum distance in effectiveness range for this impedance control is r_{max}=3.0[m]. The standard stiffness is k_0 = 0.5 [Nm/rad], and the constant value is α= 0.014, determined considering the operator's characteristics of handling etc., as explained in (Kitagawa et al, 2001). The input direction is measured by potentiometers attached to the joystick, as shown in Fig. 15, and the distance to the nearest obstacle r is given by the environmental recognition system described above (Fig. 12).

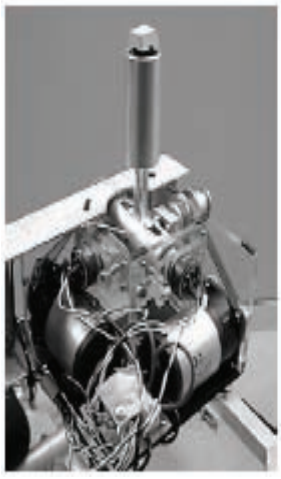


Fig. 14. Joystick

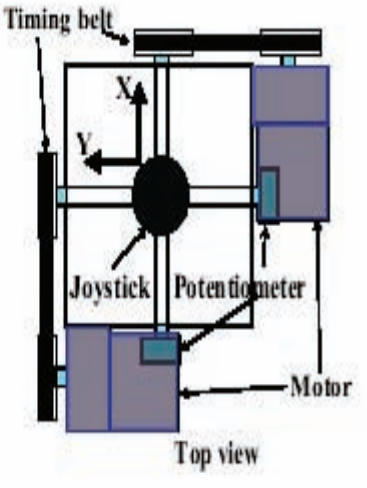


Fig. 15. Joystick system

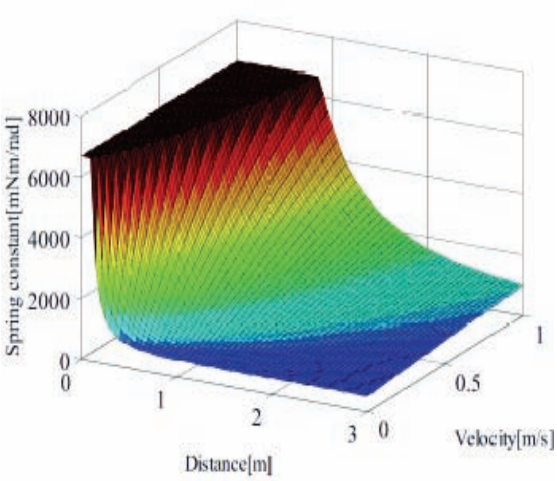


Fig. 16. Plots of stiffness against distance and velocity

Figure 16 shows plots of stiffness k against the distance r and the velocity v in the corresponding direction. Note that, as the distance r becomes smaller, the stiffness k becomes larger, moreover, as the velocity v becomes larger, the stiffness k becomes larger. In the case that the velocity is high, the motion is highly restricted, for safety. In the case that the velocity is low, the motion is scarcely restricted for maneuverability. The operator's input torque is restricted by the impedance of the joystick, hence the motion of the wheelchair is also restricted by means of the operator's joystick commands. Then, given the joystick's motor torque τ by this impedance control system is generated in the opposite direction from the declined direction of joystick by the operator.

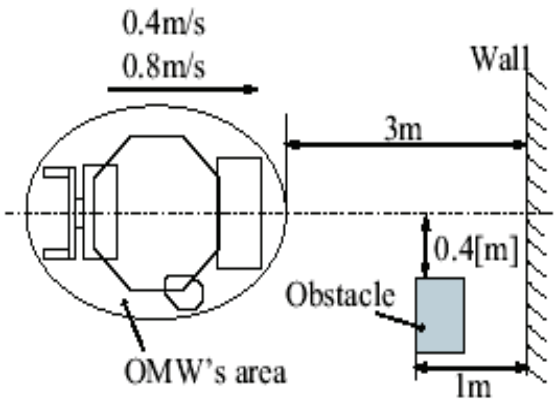


Fig. 17. Experimental condition of impedance control by using joystick

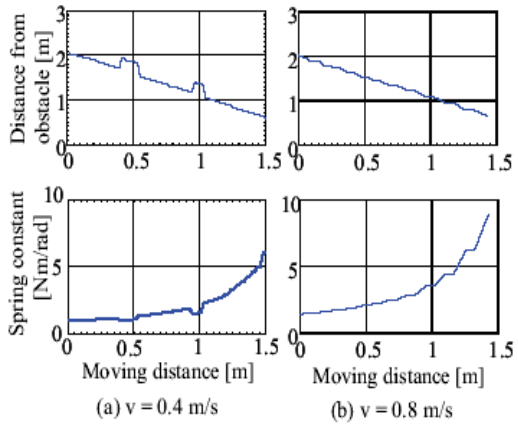


Fig. 18. Experimental result of impedance control by using joystick

Figure 18 shows an experimental result through environmental feedback using obstacle detection sensors while human operates the joystick for navigation. The stiffness k is inspected while the OMW advances toward the wall 3 [m] away from the OMW at a controlled velocity, as shown in Fig. 17. Then, the obstacle is installed between the OMW and wall. Note that, the stiffness k increases as the distance to the obstacle becomes smaller, hence the motor torque grows. Next, the stiffness k is calculated corresponding the direction of joystick's declination, using experimental result shown in Fig. 9, above.

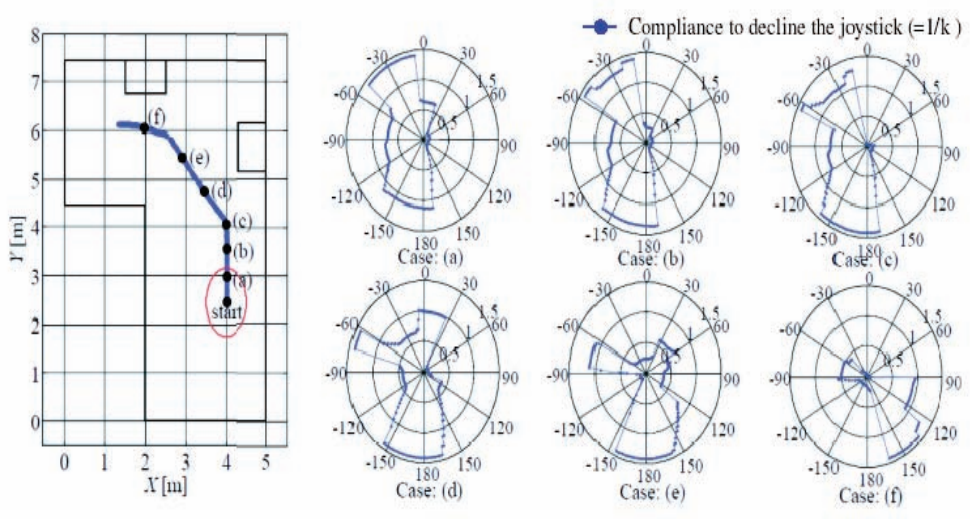


Fig. 19. Experimental result with the change of joystick's compliance during running of OMW.

Figure 19 shows the facilitation (= 1/*k*) to operate joystick for the corresponding orientation of the joystick. Then, OMW runs the speed of 0.4 [m/s]. In the sense of global coordinate as shown in the right side of Fig. 19 (case(a)-(f)), forward direction is 0 [deg], backward direction is 180 [deg], right direction is 90 [deg], and left direction is -90 [deg]. Note that, in the case that the OMW is moving in the direction nearing the obstacle, it is difficult to decline the joystick, because the motor torque grows. In other cases, it is easy to decline the joystick. Therefore the operator can naturally find out about existing obstacle and danger of collision by the present haptic device.

In the previous paragraphs, the haptic feedback joystick informs the danger level of collision to the operator. Additionally, in this section, the navigation guidance system that enable the operator to navigate the moving of OMW in the direction without crashing into obstacle, is built. This system is called the navigation guidance haptic feedback system. To begin with, take a look at the closest layer existing the recognition point.

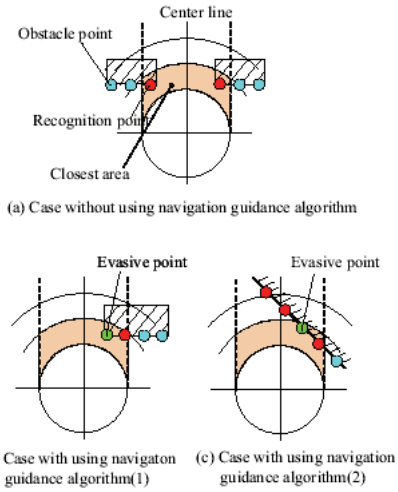


Fig. 20. Classification of cases with and without using navigation assistance algorithm

The recognition area in this closest layer is defined as the closest area, as shown in Fig. 20. Then, if the recognition points exist across the center line in the closest area, as shown in Figure 20 (a), the navigation guidance haptic feedback system is not conducted. On the other hand, if the recognition points exist on one side, as shown in Figure 20 (b) and (c), the navigation guidance haptic feedback system is worked. Moreover, in these cases (for example Figure 20 (b),(c)), the recognition points are described by polar coordinate system, and the recognition point possessing the minimum polar angle is defined as the evasive point.The OMW is controlled to move in the direction Φ by the operator's input force with the joystick, as shown in Fig. 21 (a). Using the recognition area defined previously, the inductive angle Ω_d with joystick is required, as shown in Fig. 21. Then the scale of Y-axis is altered, therefore, dash attached with variable indicates the changing of X-Y axis scale. For example, the evasive point is acquired in the N^{th} layer L_N (layer number is $n = N$), as shown in Figure 21 (b). In this case, the evasive point E' exists in the right side of the recognition

area divided by the center line. The evasive point $E'(d_e'; q_e')$ is described by polar coordinate system which has the origin of coordinate is the center of the vehicle area. This system automatically gives a force to decline joystick in the direction without existing obstacle, regardless of operator's intention. The vector of the automatically given force is defined as the inductive force F_d, and the sum of vector of the inductive force F_d and operator's current input force F_i is defined as the evasive force F_e, as shown in Fig. 21 (a).

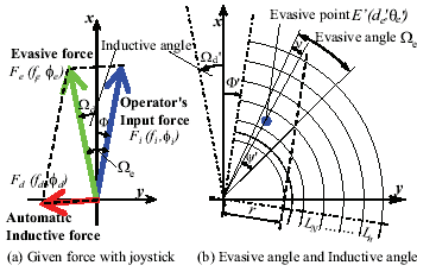


Fig. 21. Proposed method of the automatic inductive force for navigation-assistance.

These vectors are also described by polar coordinate system. The magnitude of vector of operator's current input force f_i is equal to τ witch is derived by the Eq. (16) and (17), therefore, the polar coordinate f_i and q_i are described as

$$f_i = \tau, \qquad \varphi_i = \Phi \tag{18}$$

The difference between the maximum angle range of the closest area defined by the Eq. (15) and the angle of the evasive point is described as

$$\Psi' = \sin^{-1}\left(\frac{r + Nl}{r}\right) - \theta_e' \tag{19}$$

In the case that the evasive point exists in the left side of the recognition area divided by the center line, Ψ′ is

$$\Psi' = -\sin^{-1}\left(\frac{r + Nl}{r}\right) + \theta_e' \tag{20}$$

The evasive angle Ω_e' is described as

$$\Omega_e' = \Psi' + v, \tag{21}$$

where ν is constant number for safe evasive movement and ν = 0.0872. The inductive angle

Ω_d' is described as

$$\Omega_d^{'} = \Phi' + \Omega_e^{'}, \tag{22}$$

The inductive angle Ω_d is reconverted in to the transformed scale of X-Y axis of Ω_d' as following equation.

$$\Omega_d = \tan^{-1}\left(\frac{\sin\Omega_d^{'}}{d_b / d_a \cdot \cos\Omega_d^{'}} \right) \tag{23}$$

Therefore, the evasive force $F_e(f_e, \phi_e)$ is described as

$$f_e = f_i, \quad \phi_e = \Omega_d \tag{24}$$

where the magnitude of the evasive force f_e is as the same as the magnitude of the operator's current input force f_i in order to uphold OMW's velocity of pre-inducting and past-inducting. And the angle of the evasive force ϕ_e is determined in order to get out the evasive point from recognition area. Next, the polar coordinate f_e and ϕ_e are converted into the Cartesian coordinates f_{ex} and f_{ey}, as follows.

$$f_{ex} = f_e \cos\phi_e, \quad f_{ey} = f_e \sin\phi_e \tag{25}$$

Also f_i and ϕ_i are converted to the Cartesian coordinates f_{ix} and f_{iy},as follows.

$$f_{ix} = f_i \cos\phi_i, \quad f_{iy} = f_i \sin\phi_i \tag{26}$$

Therefore the inductive force $F_{d(}\, f_{dx}; f_{dy)}$ is obtained by the following equations, because the evasive force F_e is the sum of vector of the inductive force F_d and operator's current input force F_i.

$$f_{dx} = f_{ex} - f_{ix}, \quad f_{dy} = f_{ey} - f_{iy} \tag{27}$$

The inductive force F_d is given to the motors attached on the joystick, in addition to given the force τ described by Eq. (16).

The OMW is set at the position of 2 [m] away from the obstacle, and the obstacle is closed to the OMW, as shown in Fig. 22. The OMW's wheels are run idle, the joystick is declined toward forward, and OMW's input velocity is 0.4 [m/s]. The inductive force given to the joystick is automatically provided. Figure 19 shows the experimental result of inductive

navigation guidance system, blue point describes the evasive point and green line describes the evasive force.

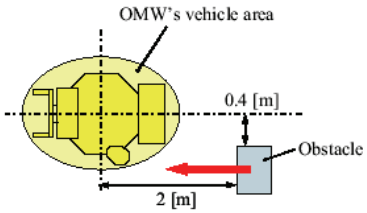


Fig. 22. Experimental condition of the proposed inductive navigation assistance system

As obstacle is approached OMW, the evasive angle is larger, and the force τ given to resist declination of the joystick by Eq. (16) is larger, so the operator's input force also is larger. Therefore, as seen from this figure, the evasive force with is the sum of vector of the inductive force F_d and operator's current input force F_i is appropriately given to the navigator. Then, following the inductive force which given to provide the joystick with the evasive force, the navigator can evade the obstacle. So good navigation-assistance could by achieved. The effectiveness of the proposed assistance system was demonstrated through a lot of experiments.

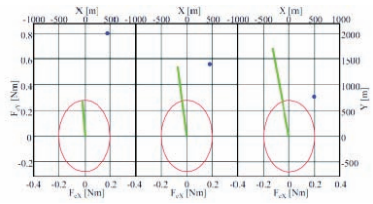
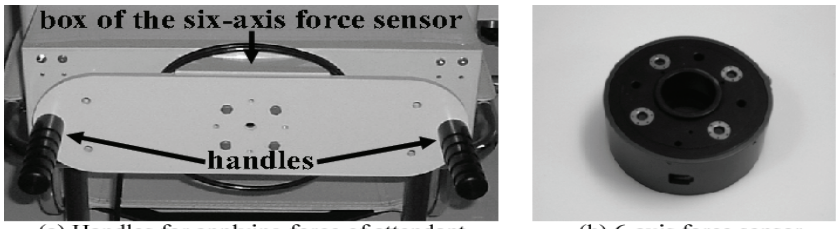


Fig. 23. Experimental results of the proposes inductive navigation assistance system.

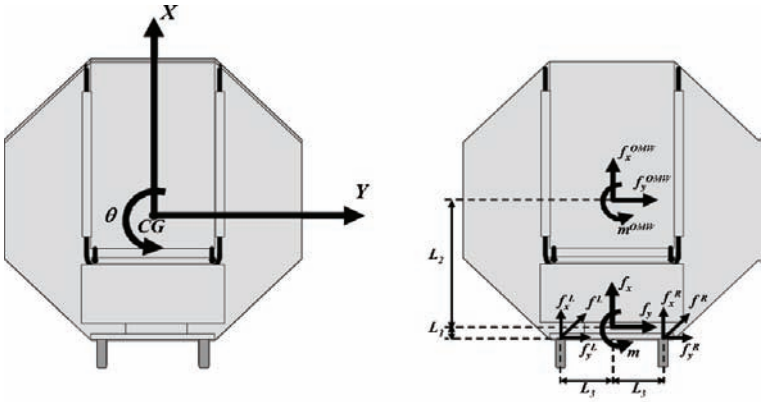
4. Power Assist Control of OMW for Helper

The force of the attendant is applied to the handles of the OMW shown in Fig. 24 (a). This force is then measured by the 6-axis force sensor shown in Fig. 24 (b).



(a) Handles for applying force of attendant (b) 6-axis force sensor

Fig. 24. Position of the handles of the OMW and 6-axis force sensor



(a) Coordinate system of the OMW. (b) Decomposition of the forces applied to the handles of the OMW.

Fig. 25. Decomposition of the force applied to the handles of the OMW

The force applied to the handles of the OMW must be decomposed according to the directions shown in Fig. 7 (a). In Fig. 10 (a), θ represents an axis perpendicular to the plane determined by X and Y. The OMW rotates around this axis in a CW or a CCW direction. The force applied to the handles in the direction X, f_x, and the force applied to the handles in the direction Y, f_y, shown in Fig. 10 (b), are decomposed according to the following equations:

$$f_x = f_x^L + f_x^R \tag{28}$$

$$f_y = f_y^L + f_y^R \tag{29}$$

As there is a perpendicular distance L_3 between the center of the force sensor and the forces and $\mathbf{f}_x^L$, and $\mathbf{f}_y^L$ a perpendicular distance L_1 between the center of the force sensor and the forces $\mathbf{f}_y^L$ and $\mathbf{f}_y^R$, these forces produce a momentum with respect to the center of the force sensor. This momentum is given as:

$$m = \left(f_x^R - f_x^L\right)L_3 + \left(f_y^L + f_y^R\right)L_1 \tag{30}$$

In order to represent the movement of the OMW, the forces and the momentum shown in Eq. (28), Eq. (29) and Eq. (30), must be translated to the center of the OMW, as shown in Fig. 25 (b). This is accomplished by the following equations:

$$f_x^{OMW} = f_x \tag{31}$$

$$f_y^{OMW} = f_y \tag{32}$$

$$m^{OMW} = m + f_y L_2 \tag{33}$$

As shown in Eq. (33), the rotation of the OMW is influenced by the force in the lateral direction Y. This observation will be used later when explaining the navigation direction estimator for force input.
The force input by the attendant is then converted to a reference velocity for the OMW by using a first-order lag controller. The reference velocity is proportional to the input force. This means that the attendant can move the OMW by the same force even when the weight of the occupant or the inclination of the travel surface changes. Moreover, the OMW can stop when the attendant stops pushing. The output signal of the force sensor, considered $F = [f_x, f_y, m_\theta]^T$, is converted to the reference velocity V_{OMW} of the OMW by a first-order lag controller.


Fig. 26. Touch panel of OMW.

A touch panel is a display device that accepts user input by means of a touch sensitive screen. Because of their compact nature and ease-of-use, touch panels are typically deployed for user interfaces in automation systems, such as high-end residential and industrial control. Touch panels are also becoming common on portable computers such as Tablet PCs, Ultra-Mobile PCs and consumer devices such as VOIP phones. In the OMW, a touch panel as shown in Fig. 26 is used as an input device in which the attendant of the OMW draws the desired direction of motion. As shown in Fig. 11, the touch panel is mounted in the rear part of the OMW such as the attendant can reach to it easily. The touch panel used in the OMW is a TFT Touch Monitor HV-141T produced by ULTEC Corporation, Japan.

The first-order lag controller converts the output signal of the force sensor $F = [f_x, f_y, m_\theta]^T$ to the reference velocity $V_{OMW} = [V_x, V_y, \omega]^T$ of the OMW. The input force can be converted to the reference velocity by using a first-order lag controller that contains an integral element.

$$G_i(s) = \frac{V_i(s)}{F_i(s)} = \frac{K_i}{T_i s + 1} \quad (i = x, y, m) \tag{34}$$

The reference velocity V_{OMW} exponentially converges to zero by using this controller when the attendant stop pushing the handle. If the time constant T_i is too small, the effect of vibration of input force or noise becomes large. If the time constant T_i is too large, the manipulability of the OMW degrades because of its slow response. The parameters of the first-order lag controller were determined by trial and error as $K_x = 0.0003$, $K_y = 0.0002$, $K_m = 0.0007$, $T_x = 0.6$, $T_y = 0.75$ and $T_m = 0.75$. A block diagram of the system is shown in Fig. 27.

When a first-order lag controller is used for the transformation from force to velocity (Kitagawa et al., 2004), a large jerk (derivative of acceleration) occurs if the input force changes suddenly. This jerk is considered a factor very significant to riding comfort. For the improvement of riding comfort this jerk must be diminished. A method for diminishing the magnitude of this jerk is proposed as follows:

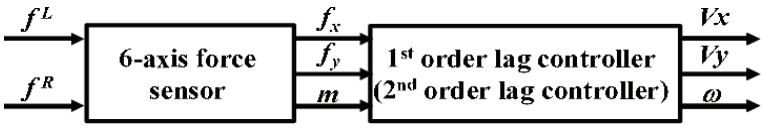


Fig. 27. Block diagram of the power-assist system

1. The gain K_i (i = x, y, ω) is decreased.
2. The value of the time constant T_i (i = x, y, ω) is increased.
3. The largest restriction of the jerk is established.
4. The controller is modified.

In item 1, as the output velocity related to the help force decreases, the jerk becomes small too. However, a large force is necessary for achieving the desired velocity. The effect of power-assist then fades and the OMW becomes, once again, very heavy for the attendant.
In items 2 and 3, the jerk can be made smaller too, but in this case, after the change, the time necessary for reaching the desired velocity increases. This generates a problem of deterioration of operability. In other words, the proposed method provides improvement in riding comfort, but the OMW's operability decays. Thus, a second-order lag controller

$$G_i(s) = \frac{V_i(s)}{F_i(s)} = \frac{K(\omega_n)_i^2}{s^2 + 2\zeta_i(\omega_n)_i s + (\omega_n)_i^2} \quad (i = x, y, m) \tag{35}$$

is chosen as a power-assist controller which can provide compatibility between both operability and riding comfort. Here, ζ is the attenuation factor. Even when the force added by attendant is fixed, if overshoot O_s occurs, a certain amount of time is required for the velocity to achieve convergence and therefore operability declines during this period. Thus, in order to avoid overshoot, ζ(i = x, y, m) is chosen as $\zeta_x = 1$, $\zeta_y = 1$, $\zeta_m = 1$. In addition, $T_x = 0.4$, $T_y = 0.4$, and $T_m = 0.4$, is used.
On the other hand, in the case of the second-order lag controller, ω_n is determined such that the system is not influenced by the noise included in the input and good operability of

OMW is also obtained. Thus, in this case, ω_n is chosen by trial and error as $(\omega_n)_x = 4$, $(\omega_n)_y = 4$, and $(\omega_n)_m = 4$.

An experimental comparison of the jerk produced in the X direction by a first-order lag controller and a second-order lag controller, each at the same reference velocity, was conducted. The experimental parameters were $K_x = 0.02$, $T_x = 0.4$, $\zeta_x = 1.0$, $(\omega_n)_x = 4.0$, and sampling time $t_s = 0.03$ [s]. OMW was moved in automatic mode. Jerk was evaluated by differentiating the output of the encoders of the OMW's motors. Experimental results are shown in Fig. 28. V_x is the reference velocity, and j_x^{OMW} shows the actual jerk that was calculated by using the encoders' output.

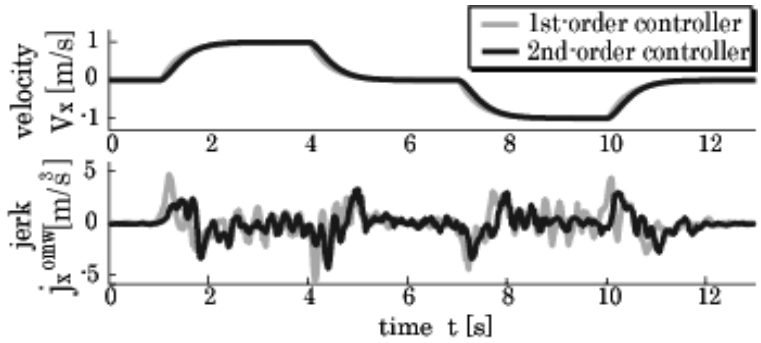
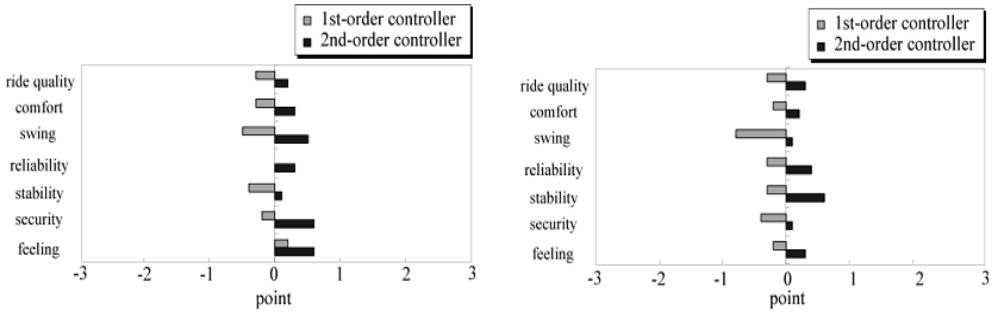


Fig. 28. Experimental results of experiments for measuring the jerk

Since in order to calculate the jerk using the encoders' output this output must be differentiated, the problem occurs that even a small amount of noise present in the encoders' output will case big changes in the jerk's value due to differentiations. Here, instead of concentrating on very precise values of the jerk, attention is given to the big variations in the jerk, so using the jerk's values for (t-1) and (t+1), where "t" is the actual time, the jerk's moving average is calculated. As the velocity is constant between t = 3~4 [s], the jerk observed in this time interval of is due to the erratic reading of encoders and is therefore ignored. Attention will be focused on the interval of time between t = 1~2 [s], in which there occurs acceleration and deceleration. It has been verified that a second-order lag controller can achieve a maximum reduction of 20% of the jerk's value produced during this period. Improvement of riding comfort is then assured by using a second-order lag controller. Moreover, comparing this result with the case in which the reference velocity is input to a first-order lag controller, there is almost no delay of time response and thus operability is not degraded. For these reasons, it is possible to conclude that in this case the second-order lag controller obtains greater performance than does the first-order lag controller. In addition, as riding comfort depends on the subjective judgment of the OMW's occupant, riding comfort was evaluated by using the Semantic Differential (SD) method. The OMW was made to move in automatic mode in X and Y directions, and a questionnaire consisting of 7 items related to driving comfort was presented to 10 different chair users. The mean value of the results obtained in each item are shown in Fig. 29 (a) for the X direction and Fig. 29 (b) for the Y direction. Even when the difference in the results in the X direction and the Y direction is not so significantly large, it is possible to see that the values obtained by the

second-order lag controller are much better than those obtained by the first-order lag controller. Thus, a second-order lag controller will be used as the power-assist controller due to its improvement in riding comfort. Then, in the block diagram of Fig. 26, a second order lag controller is used instead of the first order lag controller.



(a) X direction (b) Y direction

Fig. 29. Results of questionnaire when the OMW moves in the X direction and Y direction

5. Adaptive control by Neuro-Fuzzy System of OMW Using a Touch Panel as Human Interface for Realizing Tailor-made vehicle

When the attendant tries to rotate the OMW around its gravity center, CG, the OMW begins to slide and the radius of rotation becomes very large, as shown in Fig. 30. In this figure, the octagon represents the OMW, and the circle inside the octagon is the trajectory that point "A" in the periphery of the OMW, would describe if the OMW rotates perfectly over the CG. The thick dark line represents the trajectory of the CG, while the thin gray line represents the trajectory of "A". The numbers in arithmetic progression show how the OMW changes its position. All the symbols, lines and colors used in Fig. 30 regarding simulations and experimental results will indicate the same meanings in all the figures that follow throughout the remainder of the text.

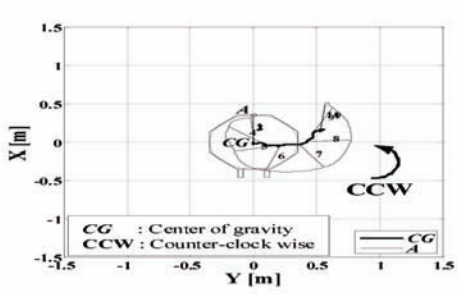


Fig. 30. A case of rotational movement of OMW in counter-clockwise (CCW) direction, when just power-assist is used

Rotation around the CG is very difficult because the OMW's rotation is not pure rotation, but is also influenced by any force that acts in the lateral direction, as shown. A survey was conducted among various attendants trying to discover some relationships in the way they realized forwards-backwards, lateral, and rotational movements. The goal of the survey was to find general rules that drew a relationship between the three described motions. Though it was impossible to find general rules that explained all cases, a relationship was found between lateral and rotational movements. It was found that if the movement in the forward or backward direction is not considered, when most of the attendants want to:

a) rotate in a clockwise (CW) direction, in addition to the rotational momentum they use some force in the lateral left direction.
b) rotate in a counter-clockwise (CCW) direction, in addition to the rotational momentum they use some force in the lateral right direction.
c) move in a lateral right direction, in addition to the lateral force they use some momentum in the CW direction
d) move in a lateral left direction, in addition to the lateral force they use some momentum in the CCW direction

1	If	$\omega < 0$ and $Vy < 0$,	then	$\omega < 0$
2	If	$\omega > 0$ and $Vy > 0$,	then	$\omega > 0$
3	If	$Vy > 0$ and $\omega < 0$,	then	$Vy > 0$
4	If	$Vy < 0$ and $\omega > 0$,	then	$Vy < 0$
5	If	$\omega < 0$ and $Vy \approx 0$,	then	$\omega < 0$
6	If	$\omega > 0$ and $Vy \approx 0$,	then	$\omega > 0$
7	If	$Vy > 0$ and $\omega \approx 0$,	then	$Vy > 0$
8	If	$Vy < 0$ and $\omega \approx 0$,	then	$Vy < 0$
9	If	$Vy \approx 0$ and $\omega \approx 0$,	then	0

Table 2. Fuzzy inference system

According to the traditional convention, CCW rotation is considered to be produced by a positive angular velocity $\omega > 0$, rotation in CW direction is considered to be produced by a negative angular velocity $\omega < 0$, lateral movement to the right is considered to be produced by a positive lateral velocity $V_y > 0$ and lateral movement to the left is considered to be produced by a negative lateral velocity $V_y < 0$. Following what has been established in the previous paragraphs, it is possible to construct Table 2. The system shown in Table 2 can be appropriately represented by a Takagi-Sugeno-Kang fuzzy model (Sugeno & Kang, 1998), (Takagi & Sugeno, 1985), with appropriate membership functions for the input, and the output being a function of the inputs, such as:

$$y_i = A_i \times V_{yj} + B_i \times \omega_j + C_i \tag{36}$$

where y_i represents the output function, "i" is a sub-index that indicates the rule to which the coefficients correspond, and the sub-index "j" can take any value in the set {Negative, Zero, Positive}.Then, by rearranging the rules of Table 2, and using the output function y_i, the system described in Table 2 becomes as shown in Table 3.

R	Antecedent		Consequent
1	If $Vy < 0$ and $\omega < 0$,	then	$y_1 = A_1 \times Vy_N + B_1 \times \omega_N + C_1$
2	If $Vy \approx 0$ and $\omega < 0$,	then	$y_2 = A_2 \times Vy_Z + B_2 \times \omega_N + C_2$
3	If $Vy > 0$ and $\omega < 0$,	then	$y_3 = A_3 \times Vy_P + B_3 \times \omega_N + C_3$
4	If $Vy < 0$ and $\omega \approx 0$,	then	$y_4 = A_4 \times Vy_N + B_4 \times \omega_Z + C_4$
5	If $Vy \approx 0$ and $\omega \approx 0$,	then	$y_5 = A_5 \times Vy_Z + B_5 \times \omega_Z + C_5$
6	If $Vy > 0$ and $\omega \approx 0$,	then	$y_6 = A_6 \times Vy_P + B_6 \times \omega_Z + C_6$
7	If $Vy < 0$ and $\omega > 0$,	then	$y_7 = A_7 \times Vy_N + B_7 \times \omega_P + C_7$
8	If $Vy \approx 0$ and $\omega > 0$,	then	$y_8 = A_8 \times Vy_Z + B_8 \times \omega_P + C_8$
9	If $Vy > 0$ and $\omega > 0$,	then	$y_9 = A_9 \times Vy_P + B_9 \times \omega_P + C_9$

Table 3. Takagi-Sugeno-Kang fuzzy model

Vy	Range	ω	Range
$Vy < 0$	[-1.0 ~ -0.1)	$\omega < 0$	[-3.30 ~ -0.20)
$Vy \approx 0$	[-0.1 ~ 0.1]	$\omega \approx 0$	[-0.20 ~ 0.20]
$Vy > 0$	(0.1 ~ 1.0]	$\omega > 0$	(0.20 ~ 3.30]

Table 4. Range of velocities

The Lateral velocity V_y is in the range [-1 ~ 1], and the angular velocity ω is in the range [-3.3 ~ 3.3]. The units of V_y and ω are [m/s] and [rad/s], respectively. After much trial and error, it was found that the more appropriate values for (V_y < 0), (V_y ~ 0), (V_y > 0), (ω < 0), (ω ~ 0), and(ω > 0), correspond to the ranges shown in Table 4.

The functions used for the partitions of the total range of V_y and ω are called dsigmoidal functions (Mathworks, 2002), and are defined as the difference of two sigmoidal functions. That is, if Eq. 26 is a sigmoidal function, with input data "x", and parameters "a" and "c", where "a" defines the inclination of the curve in the crossover point "c". Crossover points are defined (Jang et al., 1997), as the points in which "μ" = 0.5. Depending on the sign of the parameter "a", the sigmoidal membership function is inherently opened to the right or to the left (if "a" is positive, the sigmoidal function is opened to the right, and if "a" is negative the sigmoidal function is opened to the left).

$$f(x, a, c) = \frac{1}{1 + e^{-a(x-c)}} \tag{37}$$

A dsigmoidal function can be defined as

$$f(x, a_1, c_1) - f(x, a_2, c_2) = f(x, [a_1, c_1, a_2, c_2]) \quad (38)$$

The partitions of the ranges of V_y and ω by using dsigmoidal functions are shown in Fig. 31.

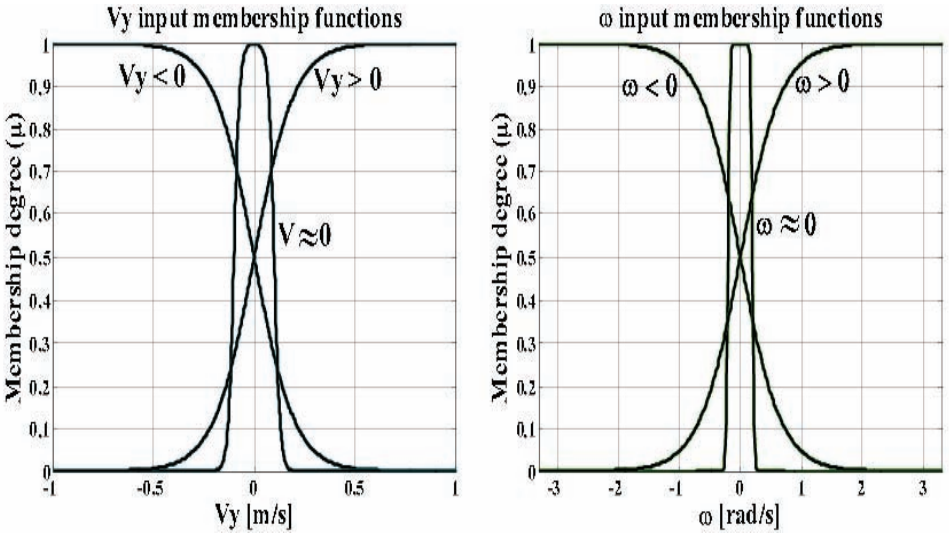


Fig. 31. Partitions of the ranges of V_y and ω by using dsigmoidal functions

R	A	B	C
1	0.05	1	0
2	0	1	0
3	1	0.05	0
4	1	0	0
5	0	0	0
6	1	0	0
7	1	0.05	0
8	0	1	0
9	0.05	1	0

Table 5. Values of the coefficients A, B and C

The coefficients "A" and "B" are decided considering the values shown in Table 2. C is always 0. The values of "A_i", "B_i" and "C_i" are shown in Table 5. The value "0.05" was hosen by trial and error. The reason for choosing "0.05" instead of "0", as could be inferred rom Table 2, is in order not to cancel completely the movement in the undesired direction. igger values, such as "0.1", "0.2", ... , have been tested as well, but it has been found that if alues greater than "0.05" are used, the deviation in direction Y is not reduced as much as desired. In fact, for obtaining the value of the desired lateral velocity V_y and the desired ngular velocity ω, two fuzzy reasoning systems are needed. The inputs are the same, but the outputs are decided as shown in Table 6. These two fuzzy systems will be used in the block labeled "directional reasoning" in the block diagram of the OMW system shown in Fig. 32. The contents of the block "directional reasoning" are shown in Fig. 33.

Rule	Antecedents	Consequents					
		$\hat{V}y$			$\hat{\omega}$		
		A^{Vy}	B^{Vy}	C^{Vy}	A^{ω}	B^{ω}	C^{ω}
1	If $Vy < 0$ and $\omega < 0$, then	0 (Vy)	0 (ω)	0	0.05 (Vy)	1 (ω)	0
2	If $Vy \approx 0$ and $\omega < 0$, then	0 (Vy)	0 (ω)	0	0 (Vy)	1 (ω)	0
3	If $Vy > 0$ and $\omega < 0$, then	1 (Vy)	0.05 (ω)	0	0 (Vy)	0 (ω)	0
4	If $Vy < 0$ and $\omega \approx 0$, then	1 (Vy)	0 (ω)	0	0 (Vy)	0 (ω)	0
5	If $Vy \approx 0$ and $\omega \approx 0$, then	0 (Vy)	0 (ω)	0	0 (Vy)	0 (ω)	0
6	If $Vy > 0$ and $\omega \approx 0$, then	1 (Vy)	0 (ω)	0	0 (Vy)	0 (ω)	0
7	If $Vy < 0$ and $\omega > 0$, then	1 (Vy)	0.05 (ω)	0	0 (Vy)	0 (ω)	0
8	If $Vy \approx 0$ and $\omega > 0$, then	0 (Vy)	0 (ω)	0	0 (Vy)	1 (ω)	0
9	If $Vy > 0$ and $\omega > 0$, then	0 (Vy)	0 (ω)	0	0.05 (Vy)	1 (ω)	0

Table 6. Partitions of the systems shown in Table 3 in two sub-systems

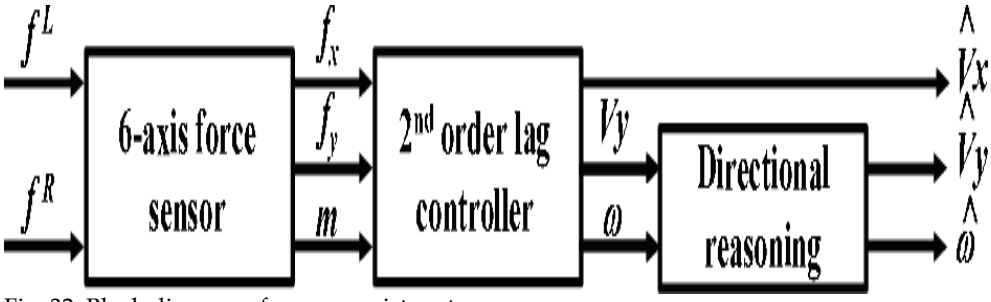


Fig. 32. Block diagram of power-assist system

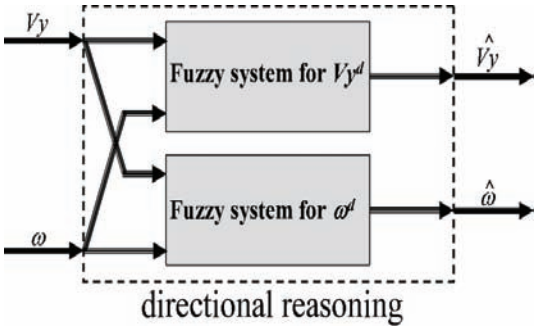


Fig. 33. Contents of the block "directional reasoning" in Fig. 32.

The effectiveness of the fuzzy system described above was tested experimentally. After much trial and error, the values of $K(\omega_n)_i$ in Eq. (35), were determined to be: $K(\omega_n)_x = 0.19$, $K(\omega_n)_y = 0.22$, and $K(\omega_n)_\omega = 1.1$, with $\zeta_i = 1$ and $(\omega_n)_i = 4$. Experiments were conducted for four different attendants by using a fuzzy reasoning system tuned to respond to the characteristics of one of them. The attendants were asked to move the OMW laterally, backwards, and forwards, in CCW rotation, and in the diagonal direction. The results obtained for the 4 attendants in the case of CCW rotation are shown in Fig. 34. As shown in Fig. 20, even when the Fuzzy Reasoning (FR) System helps to improve the OMW's operability, the result is different for each attendant because of the particular characteristics of each of them. According to the results shown in Fig. 34, it is clear that the fuzzy reasoning system was tuned to respond to the characteristics of "Attendant 3" and for that reason the lateral deviation is almost nonexistent in his case. As "Attendant 3" does not use so much force in the not adjusted X direction, then he can rotate almost perfectly over the CG of the OMW. For the other three cases, the not precise tuning for rotation and lateral movement, combined with the effect of the force in direction X, causes that the movement is not as good as expected. As a consequence of the analysis of these results, the idea of tuning the fuzzy system according to the characteristics of each attendant came into view. The tuning by trial and error is time consuming and requires many attempts by the attendants who may grow tired or bored. Therefore, another method of tuning the FR system must be found. Using the input data of the attendants is the most natural alternative. In this case, the FR systems will tune automatically by learning the tendency of the different attendants. As the FR systems do not have an ability to learn, the addition of a complementary system that allows learning is necessary. Learning capability can be obtained by adding a Neural Network (NN) to the FR system. A hybrid system called a neuro-fuzzy system is then obtained. Much research exists regarding this topic (Jang, 1993), (Jang et al., 1997), (Juang & Lin, 1998), (Lian et al., 1999), (Lin & Lee, 1991), (Nguyen et al., 2003). Jang (Jang, 1993), (Jang et al., 1997), developed the ANFIS (Adaptive-Network-based Fuzzy Inference System), a neuro-fuzzy system in which the fuzzy inference system is tuned by using the system's input data. The tuning is performed by minimizing the output error of the NN used in combination with the fuzzy inference system. For achieving this goal, the NN is trained by using a hybrid method that combines least squares and the Backpropagation algorithm (BP law). This method is thus thought to be effective as a parameter tuning method for the OMW's fuzzy inference system.

The ANFIS was proposed as a system that can tune a FR system through the use of input data. First, the FR system is changed to the form of a NN network that can be trained by using the system's input data. Fig. 35 shows the ANFIS system used for obtaining the value of the desired angular velocity and the desired lateral velocity. This system is based on the Takagi-Sugeno-Kang fuzzy models shown in Table 6. The ANFIS system shown in Fig. 36 has 5 layers. The nodes of the layers are represented by circles or squares. Square nodes contain values which can be tuned. Circular nodes contain fixed values or mathematical operators. A brief explanation of how the different layers shown in Fig. 36 are achieved follows. For a more detailed explanation in this topic, and in the training of the ANFIS, refer to references (Jang, 1993), (Jang et al., 1997).

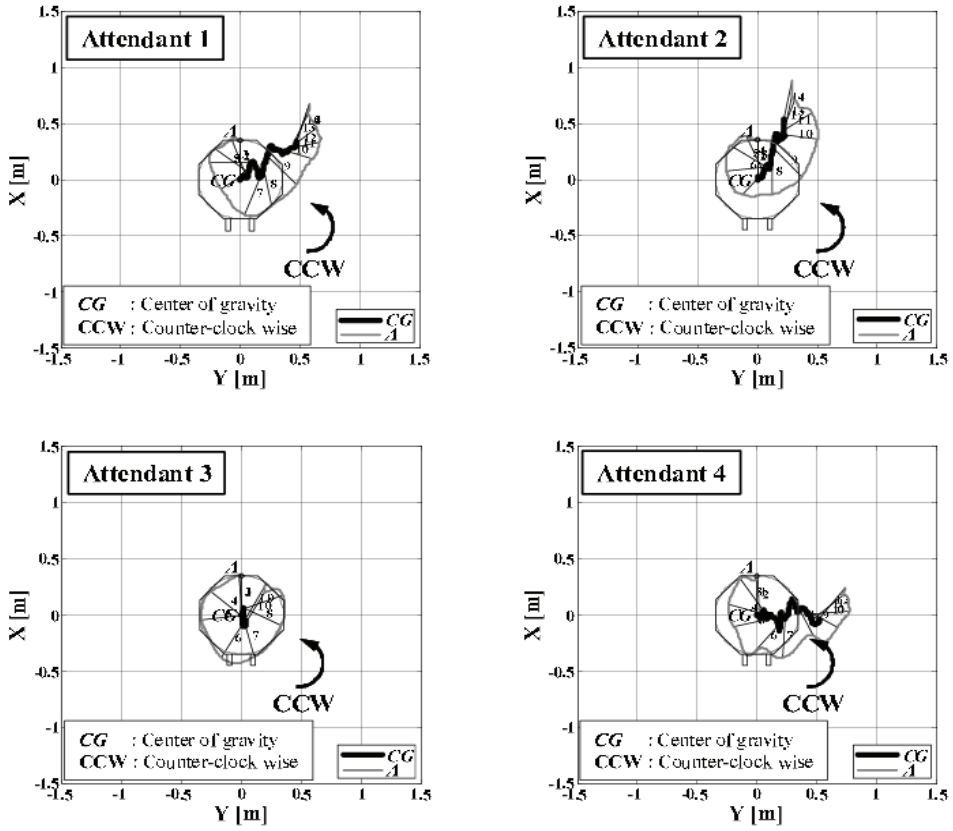


Fig. 34. Experimental results of four attendants when the Fuzzy Reasoning system is used

- 1st Layer: Here the inputs V_y and ω are subjected to the action of the membership functions of Fig. 31, which are represented by its parameters (a_1 … a_{12}) and (c_1 … c_{12}).
- 2nd Layer: In the 2nd Layer the fuzzy rules shown in Table 6 are constructed. As the

antecedents are jointed by a logic "AND", this relationship is mathematically obtained by the product (Π) of the two antecedents. The output of each node represents the firing strength of a rule, which is represented by α_i (i = 1, ... , 9).

- 3rd Layer: This is a normalization layer, where the ratio of the ith rules' firing strength to the sum of all rules firing strength is calculated.
- 4th Layer: Here the normalized firing strength that comes from the 3rd Layer is multiplied by the output functions of the fuzzy reasoning system.
- 5th Layer: The overall output of the system is computed as the sum of all the incoming signals.

In previous research (Terashima et al., 2004), the desired direction of motion of the attendant was input by using the keyboard of the computer of the OMW. However, the attendant could not get a clear idea of the direction in which he wanted to move, neither verify if the real motion of the OMW really corresponded to his desire. In order to provide the attendant with an easy way for inputing the desired direction of motion and for verifying the direction of motion, a human interface consisting of touch panel, as shown in Fig. 26 is used. A GUI (Graphical User Interface) was developed for making easy the interaction with the attendant, as show in Fig. 36. In this GUI the attendant can draw any kind of motion, be it an slanting motion, or a rotational movement. Moreover, the GUI allows the attendant to follow the motion of the OMW and realize the difference between the desired and the real motion of the OMW.

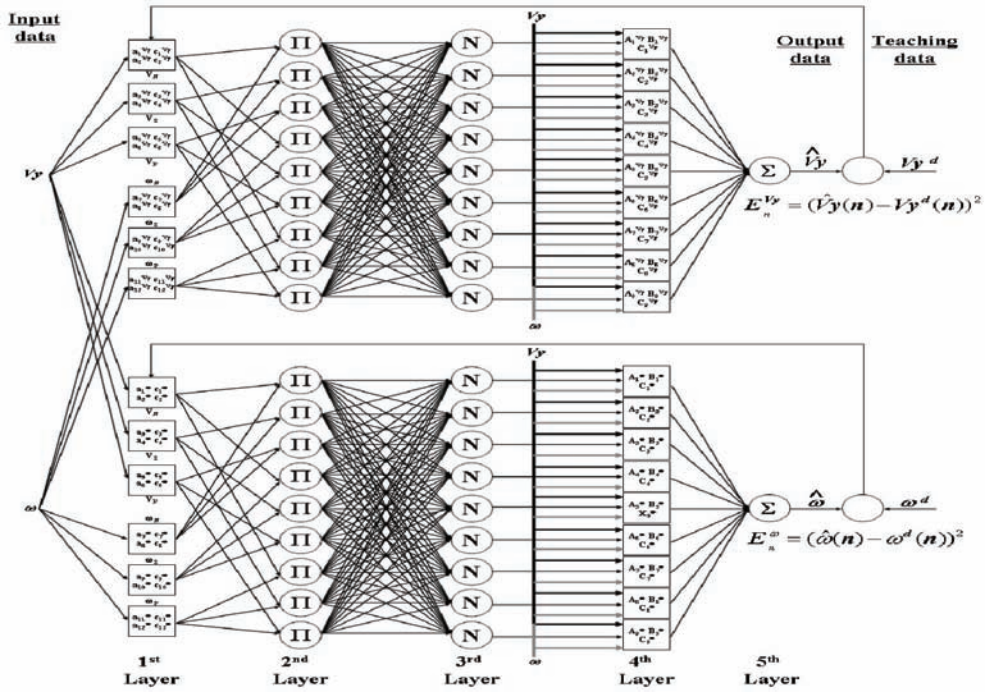


Fig. 35. The ANFIS system

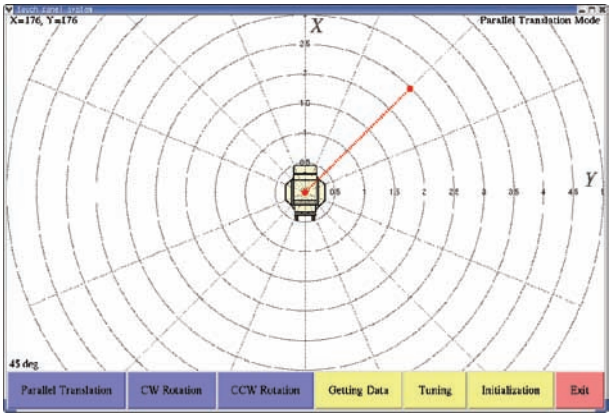


Fig. 36. GUI developed for the touch panel

The procedure for applying the touch panel is as follows:

a) First, the attendant draws in the touch panel the kind of movement that he desires to accomplish.
b) Then, the attendant moves the OMW trying to accomplish the desired motion.
c) However, in the general case, there as a difference between the desired motion and the real motion. This difference is used for the training of the ANFIS system of the OMW, as explained in (Terashima et al., 2004).

A Reduction Multiplicative Factor (RMF) which decreases the value of V_x in the case of rotational motion, and keeps it unchanged in the case of forwards-backwards movement was the solution provided by authors (Urbano et al., 2006a) for improving the forwards-backwards motion, lateral motion and rotational motion over the gravity center of the OMW.
However, as V_y was subjected to fuzzy reasoning and V_x was not, it was not possible to achieve good operability for slanting motions, like diagonal motion. In the case of diagonal motion, for example, the attendant tries to move the OMW in such a way that the inputs of V_x and V_y are almost the same in the beginning. Nevertheless, as V_y is subjected to directional reasoning, its value changes. V_x is not subjected to directional reasoning, then its value remains always the same. As a consequence, it is not possible to achieve good operability in diagonal motion.
For solving this problem, V_x was subjected to directional reasoning too using the fuzzy rules shown in Table 7 These rules make it possible to include V_x in the fuzzy reasoning system without disturbing the values of V_y or ω. The block diagram of the system that considers power assist and fuzzy reasoning is shown in Fig. 37, and the contents of the block labeled as "directional reasoning" are shown in Fig. 38. By including V_x in the ANFIS system it was possible to accomplish a general omni-directional motion. Then, the complete ANFIS of the OMW becomes as shown in Fig. 39, and the complete system, when the touch panel is included, is shown in Fig. 40.

R	Antecedent	Consequent
1	If $Vx < 0$ and $Vy < 0$,	then $Vx < 0$
2	If $Vx \approx 0$ and $Vy < 0$,	then $Vx \approx 0$
3	If $Vx > 0$ and $Vy < 0$,	then $Vx > 0$
4	If $Vx < 0$ and $Vy \approx 0$,	then $Vx < 0$
5	If $Vx \approx 0$ and $Vy \approx 0$,	then 0
6	If $Vx > 0$ and $Vy \approx 0$,	then $Vx > 0$
7	If $Vx < 0$ and $Vy > 0$,	then $Vx < 0$
8	If $Vx \approx 0$ and $Vy > 0$,	then $Vx \approx 0$
9	If $Vx > 0$ and $Vy > 0$,	then $Vx > 0$

Table 7. Fuzzy rules for the change of V_x in order to improve the operability of the OM

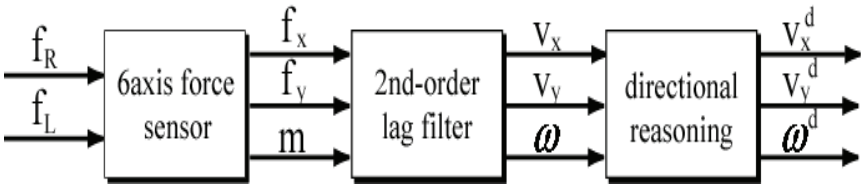


Fig. 37. Block diagram of the power assist system.

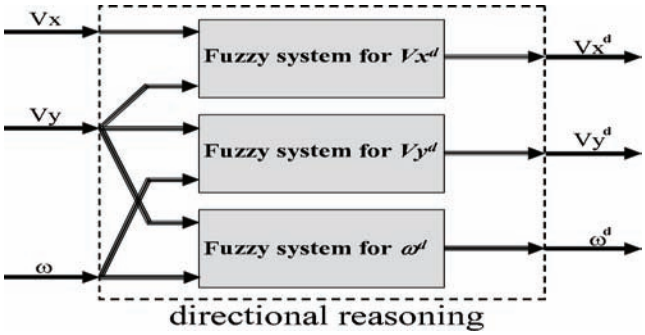


Fig. 38. Contents of the block "directional reasoning".

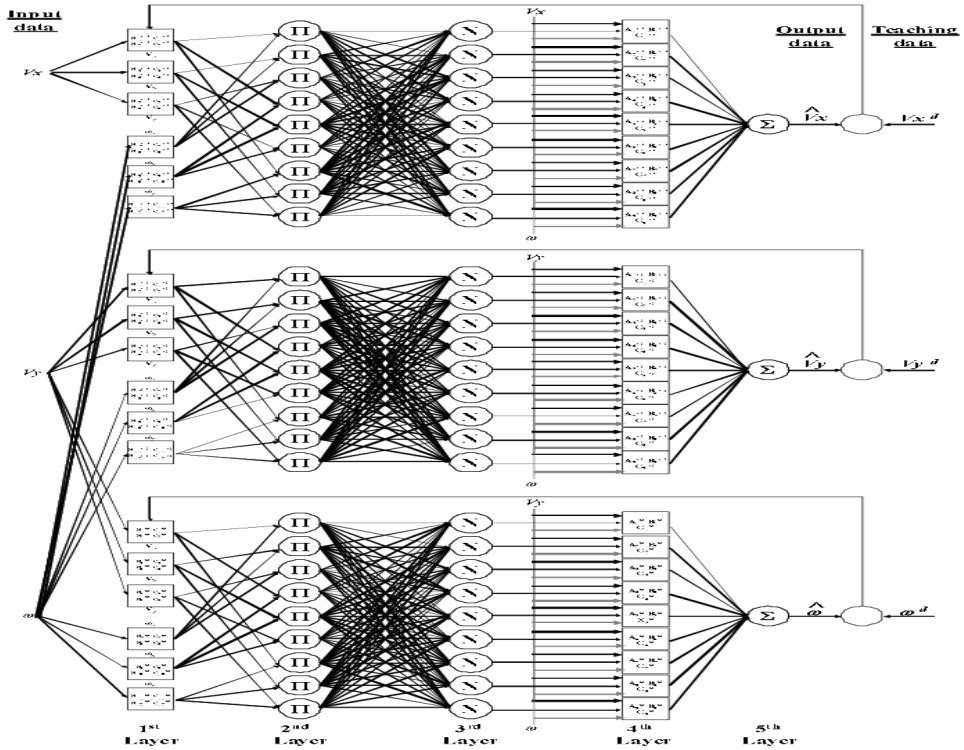


Fig. 39. ANFIS systems of the OMW.

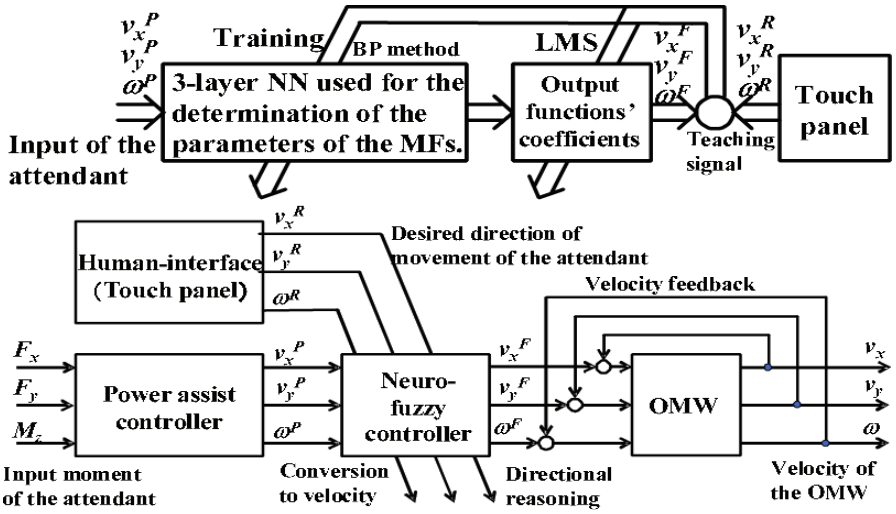


Fig. 40. Complete system when the touch panel is included.

Rule	Antecedents	Consequents
1	**If** Vy_N **and** ω_N,	**then** rotational movement in CW direction
2	**If** Vy_Z **and** ω_N,	**then** rotational movement in CW direction
3	**If** Vy_P **and** ω_N,	**then** lateral movement to the right
4	**If** Vy_N **and** ω_Z,	**then** lateral movement to the left
5	**If** Vy_Z **and** ω_Z,	**then** no movement
6	**If** Vy_P **and** ω_Z,	**then** lateral movement to the right
7	**If** Vy_N **and** ω_P,	**then** lateral movement to the left
8	**If** Vy_Z **and** ω_P,	**then** rotational movement in CCW direction
9	**If** Vy_P **and** ω_P,	**then** rotational movement in CCW direction

Table 8. Fuzzy rules for improving the operability of the OMW

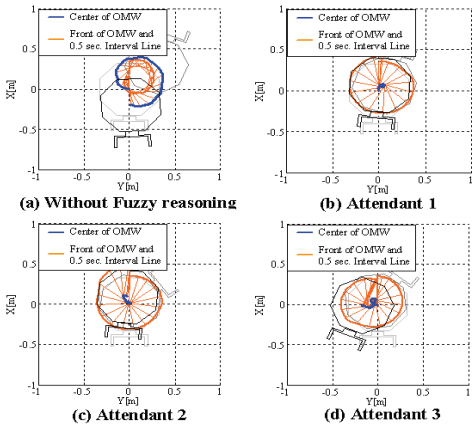


Fig. 41. Results when fuzzy reasoning is not used and when the fuzzy system is tested with different attendants.

As the OMW is so heavy for being easily moved by the attendants, a power assist system has been attached to the OMW in order to help the attendants. However, when the power assist system is attached to the OMW, problems of operability appear, especially in the case of rotation over the center of gravity of the OMW. According to the dynamics of the power assisted system, and the analysis of the data of many attendants, it was possible to determine some rules that the attendants follow when they try to move the OMW in lateral direction, or rotate over the center of gravity of the OMW in counter-clockwise or clockwise direction. These rules are summarized in Table 8, in the form of the fuzzy rules of a fuzzy reasoning system. In Table 8, Vy represents the lateral velocity of the OMW, ω represents the rotational velocity of the OMW and the sub-indices N, Z and P means *negative*, *zero* and *positive*, respectively. Fig. 41 (a) shows the results in the case of a counter-clockwise rotational over the center of gravity of the OMW when no fuzzy reasoning is used. It is possible to see that there is a deviation in the lateral direction as well as in the forwards-backwards direction. For solving this problem, the fuzzy system was used. It was tuned by

trial and error, for an attendant that will be called "Attendant 1", and the results, presented in Fig. 41 (b) show that the rotational movement was improved considerably. However, when the same system was tested with two more different attendants, called "Attendant 2" and "Attendant 3", the results were not as good as in the case of "Attendant 1", as shown in Fig. 41 (c) and Fig. 41 (d). It means that the system must be tuned in order to respond to the individual characteristics of the different attendants. However, the tuning by trial and error is time
consuming and boring for the attendant. For that reason, the automatic tuning of the system by using a neuro-fuzzy system, ANFIS (Adaptive-Neural Fuzzy Inference System) was proposed and developed as described in (Urbano et al, 2006b).

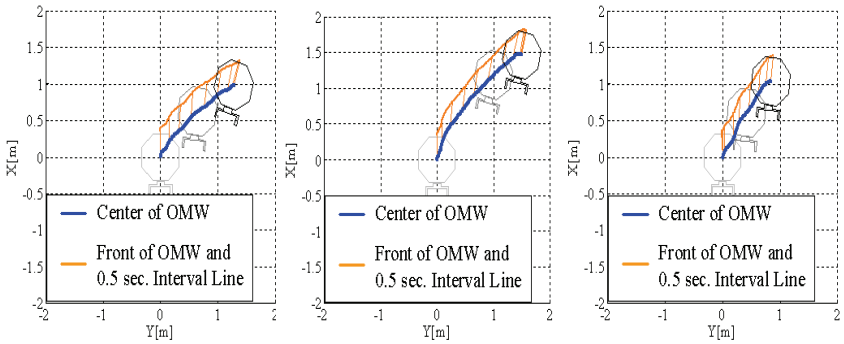


Fig. 42. Experimental results before the tuning of the system.

In previous research (Urbano et al, 2006b), the desired direction of motion of the attendant was inputted by using the keyboard of the computer of the OMW. However, the attendant could not get a clear idea of the direction in which he wanted to move, neither verify if the real motion of the OMW really corresponded to his desire. In order to provide the attendant with an easy way for input of the desired direction of motion and for verifying the direction of motion, a human interface consisting of touch panel, as shown in Fig. 42 is used. A GUI (Graphical User Interface) was developed for making easy the interaction with the attendant. In this GUI the attendant can draw any kind of motion, as for example slanting motion, or a rotational movement. Moreover, it allows the attendant to follow and compare the difference between the desired and the real motion of the OMW. The procedure for applying the touch panel is as follows:

a) First, the attendant draws in the touch panel the kind of movement that he desires to accomplish.

b) Then, the attendant moves the OMW trying to accomplish the desired motion.

c) However, in the general case, there is a difference between the desired motion and the real motion. This difference is used for the training of the ANFIS system of the OMW,

as explained in (Urbano et al, 2005a).

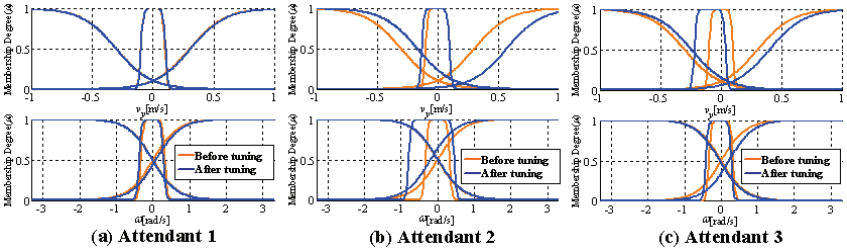


Fig. 43. Variation of the input membership functions of different attendants.

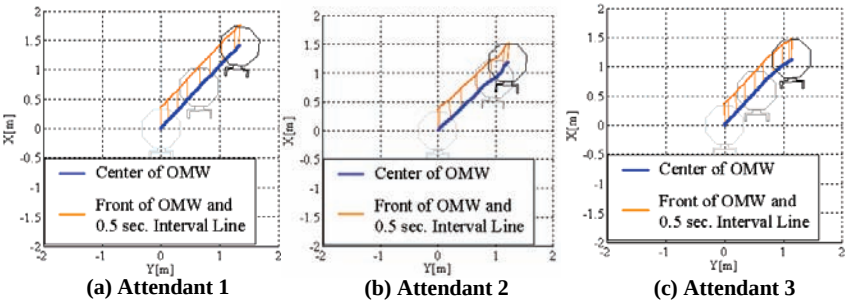


Fig. 44. Experimental results after the tuning of the system

Fig. 42 shows the experimental results for three different attendants before the tuning of the system, and Fig. 44 shows the experimental results for the same three attendants after the system was tuned by using ANFIS. It is possible to see that the movement accomplished by the attendants was greatly improved after the tuning. As shown in the graphs of Fig. 43, the shape of the membership functions of the fuzzy system changes in order to fit to the individual characteristics of each attendant. In the case of "Attendant 1", the membership functions are practically the same before and after the tuning with ANFIS because the system had already being tuned for "Attendant 1". In the case of "Attendant 2", the graphs show that the value of the input of rotational moment and lateral velocity to the right is greater than that of "Attendant 1". Finally, in the case of "Attendant 3" the difference is the width of the null partition of the lateral velocity. These results demonstrate the good performance of the proposed approaches in any case.

6. Comfort Driving of the OMW

The control system is based on a hybrid shape approach recently developed in our laboratory (Yano et al, 2000). Optimization problems formulated in both the time and the frequency domains have scarcely been found in the other researches.
Here, a dynamic model of vibration is not utilized in the control design step, because it

would be difficult to mathematically model the vibrations of the wheelchair and the patient using it. In the design of a controller, the natural frequency of the wheelchair and of the user's organs are utilized as *a priori* information. Controller design is composed of the following elements:

a) Selection of controllers: Controllers are constructed by several fundamental control elements such as a notch filter, low-pass filter, PID elements, and so on.
b) Formulation of design specifications: The design specifications in the frequency and the time domains are expressed by penalty functions.
c) Formulation of an optimization problem: An optimization problem is formulated with constraints expressed by penalty terms.
d) Computation of a controller: The parameters of the controller are computed by solving an optimization problem. In this paper, the Simplex method is applied to solve the problem. Here, the controller is designed in the direction for the first step. In this system, the controller is supposed to be designed in three directions independently, which seems to be achieved easily. In this paper, only X and Y axes will be considered, but not θ axis ($\theta_r = 0$) for brevity. However, consideration of θ is straightforward.

A. Selection of controller

The wheelchair is controlled by a servo system with an integrator. According to the Internal model principle, a proportional control (P control) system is sufficient to avoid the offset. Therefore, a proportional gain is given as an element of this controller.

$$K_1(s) = K_P \tag{39}$$

Two notch filters prevent the controller from exciting vibration in the wheelchair or in the patient's organs, as

$$K_2(s) = \frac{s^2 + 2\zeta\omega_\omega s + \omega_\omega^2}{s^2 + \omega_\omega s + \omega_\omega^2} \tag{40}$$

$$K_3(s) = \frac{s^2 + 2\zeta\omega_0 s + \omega_0^2}{s^2 + \omega_0 s + \omega_0^2} \tag{41}$$

,where the parameters are given as the natural frequency of the wheelchair $\omega_\omega = 15.5$ [rad/s] (X-axis) and $\omega_\omega = 15.8$ [rad/s] (Y-axis), and that of human's organs $\omega_0 = 37.7$ [rad/s]; and $\zeta = 0.0001$. Since the natural frequencies of the user's organs range from 4 [Hz] to 8 [Hz], the intermediate value is adopted. In order to reduce the influence of higher-order vibration and noise, a low-pass filter, which is the low gain in the high frequency domain, is given as

$$K_4(s) = \frac{\omega_l^2}{s^2 + 2\zeta_l\omega_l s + \omega_l^2} \tag{42}$$

where the parameters are given as $\zeta_l = 0.7$. Finally, the transfer function of the controller is given as

$$K(s) = \prod_{i=1}^{4} K_i(s) = \frac{(K_P \omega_l^2)(s^2 + 2\zeta\omega_\omega s + \omega_\omega^2)(s^2 + 2\zeta\omega_0 s + \omega_0^2)}{(s^2 + 2\zeta_l \omega_l s + \omega_l^2)(s^2 + \omega_\omega s + \omega_\omega^2)(s^2 + \omega_0 s + \omega_0^2)} \tag{43}$$

In this equation, K_P and ω_l are unknown parameters. Therefore, both parameters should reasonably be determined by solving an optimization problem.

B. Formulation of design specifications

The specifications of the controller are formulated by making use of penalty function. Penalties are given if any of the following relations do not hold.

- The controller and the closed-loop system are stable.

$$\mathrm{Re}[r_K] < 0, \quad \mathrm{Re}[r_{cl}] < 0 \tag{44}$$

$$K_P > 0, \quad \omega_l > 0 \tag{45}$$

- The controller gain is less than -20[dB] at the natural frequency of the wheelchair ω_ω = 15.5 [rad/s] (X-axis) and ω_ω = 15.8 [rad/s] (Y-axis),or at that of user's organs ω_θ = 37.7[rad/s].
-

$$\left|K(\omega_w)\right| < -20[dB] \tag{46}$$

$$\left|K(\omega_0)\right| < -20[dB] \tag{47}$$

- The controller gain is less than 0[dB] at ω_l = 188 [rad/s] in order to decrease the influence of the higher-order vibration and noise.

$$\left|K(\omega_l)\right| < 0[dB] \tag{48}$$

- The input voltage u does not exceed a magnitude of 24 [V].

$$\max|u| < 24[V] \tag{49}$$

- Maximum overshoot does not exceed a magnitude of 0.001 [m].

$$\max(O_s) < 0.001[m] \tag{50}$$

C. Formulation of an optimization problem

The following optimization problem using penalty terms is formulated with Eqs. (41) to (47).

$$\min_{K(s)} J = T_s + J_P \tag{51}$$

where

$$J_P = \omega_1 + \omega_2 + \cdots + \omega_9 \tag{52}$$

, and T_s is the settling time when the target position is 2.0 [m] and the admissible error is 0.001[m]. If any of the above penalty conditions are not satisfied, the penalty function w_i is given as $w_i = 10^8 (i = 1,\ldots, 9)$.

D. Computation of a controller

For the optimization of a cost function, the Simplex method is used, where the reflection coefficient α =1.0, the expansion coefficient β =0.5 and the contraction coefficient γ = 2.0. The initial simplex is described in Table 9. As the results of computation, optimum values of K_P and ω_l are shown in Table 10, which was calculated by about 20 times repetition.

	1	2	3
K_P	15	20	30
ω_l	30	40	50

Table 9. Initial simplex

	X-axis	Y-axis
K_P	15.6827	14.6374
ω_l	13.4135	31.6110

Table 10. Results of optimization

E. Control Simulation

By means of Eq. (43) using the values of Table 10, control simulations were done. Trapezoidal reference trajectory was given as x_r. Figures 45 and 46 were respectively simulation results of X-axis and Y-axis. In Fig. 45 and 46, the dotted line and solid line show the case of Feedforward (FF) inputs given by trapezoidal form and hybrid shape approach (HSA) respectively. HSA approached realized the good result without residual vibration in both of X and Y-direction. The frequency response of the controller is shown in Fig. 45. It can be seen that the high-speed transfer is achieved because the controller is high-gain in the

low-frequency domain. The simulation result of the trajectory tracking is shown in Fig. 45 and Fig 46. As seen from this figure, it is confirmed that overshoot is not found.

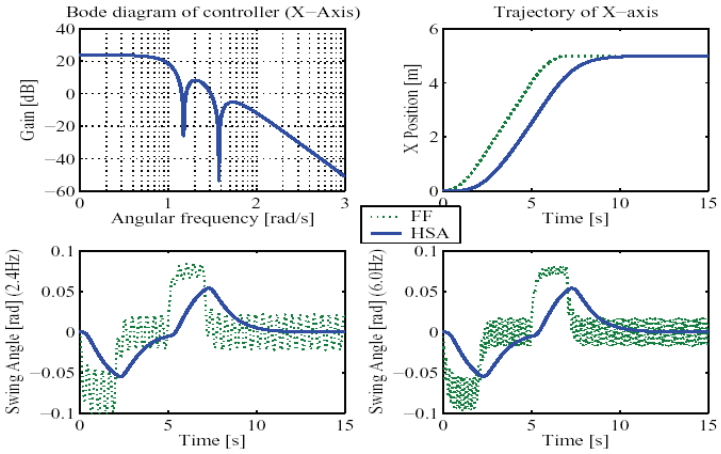


Fig. 45. Simulation results of X-axis

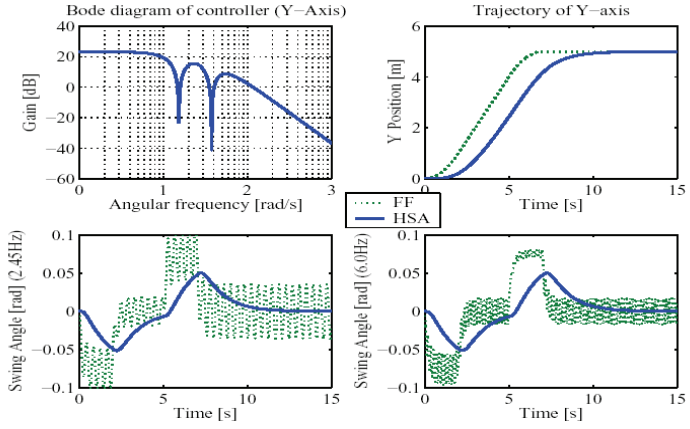


Fig. 46. Simulation results of Y-axis

In order to verify the effectiveness of the control system, laboratory experiments are conducted. In these experiments, the wheelchair moves forward 3.3 [m] (+X direction) and halts three times, for 10 [s] at a time. Two kinds of controllers are examined.

- PD feedback controller without filter
- Proposed controller designed by the hybrid shape approach.

The experimental trajectory is shown in Fig. 47. The experimental results are evaluated by the following two steps. In the first step, the output signal of the acceleration sensor attached to the wheelchair is examined to evaluate the vibration suppression. However, the effectiveness of the consideration of the patient's organs cannot be evaluated in this step. In the second step, the effectiveness of the proposed method on comfort is evaluated by the SD (Semantic Differential), which is a kind of inspection using a scale of verbal. The output of the acceleration sensor attached beneath the seat is shown in Fig. 48. The resultant acceleration and the jerk are suppressed by the hybrid shape approach.

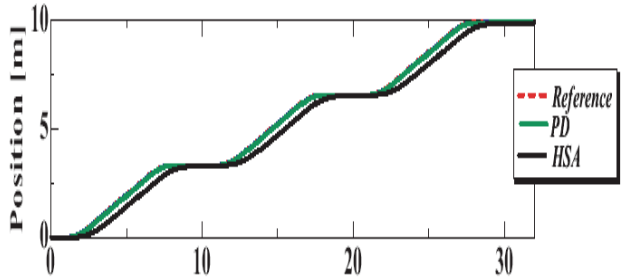


Fig. 47. Trajectory of movement of X-axis

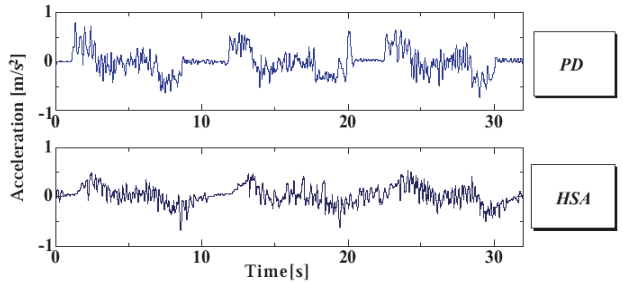


Fig. 48. Experimental results (X-direction)

The SD method is applied to evaluate the effectiveness of the consideration of the patient's organs. In this method, several pairs of adjectives are adopted to evaluate an object or feeling. Within each pair, the adjectives are antonymous each other. To describe the feeling that he or she is experiencing, the examinee selects one of seven grades that form a scale ranging from the one adjective to the other. This method is especially effective for finding the shades of differences among several objects or feelings. The wheelchair was evaluated by 15 examinees. The average value of each item is shown in Fig. 49. The hybrid shape approach seems to enable examinees to provide the greatest sense of patient comfort. Furthermore, Fig. 50 and Fig. 51 are experimental results of Y-direction. The result by HSA

is better than the conventional trapezoidal velocity curve, or, PD controller. Figure 52 shows the experimental results of diagonal direction ($x_r = y_r$, $\theta_r = 0$). In the diagonal movement of OMW, OMW can be transferred comfortably by using the smooth acceleration curve of the proposed HSA. Through this research, it was clarified that vibration suppression and comfort riding in OMW were realized by using the proposed HSA control.

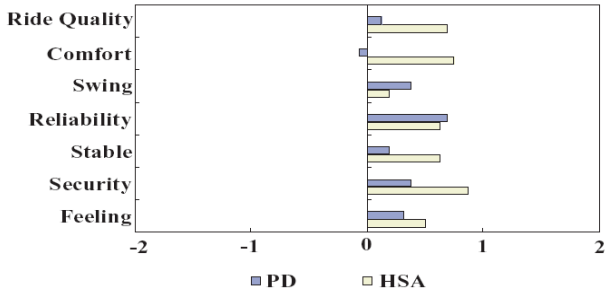


Fig. 49. Result of questionnaire (X-direction)

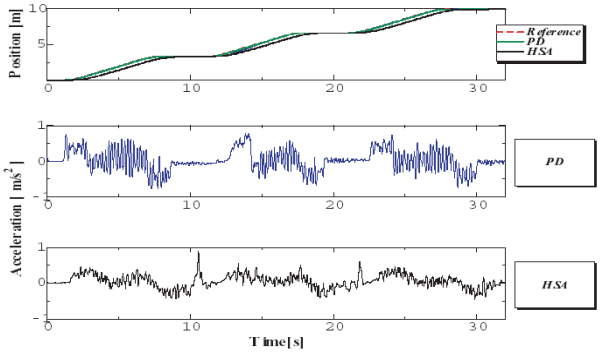


Fig. 50. Experimental Results (Y-direction)

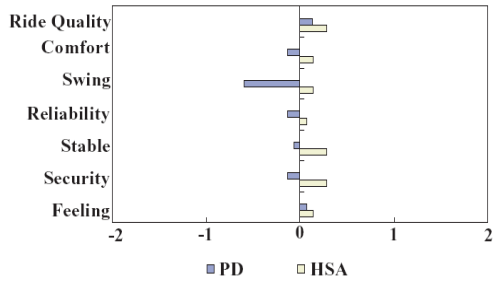


Fig. 51. Results of questionnaire (Y-direction)

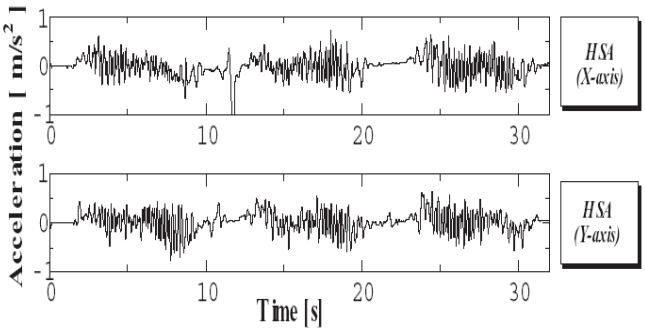


Fig. 52. Experimental result ($x_r = y_r$; $\theta r = 0$)

7. Conclusions

1. A local map was built around the OMW by using range sensors. This local map allows knowing the distance from the OMW to the surrounding obstacles in a circle with a radius of 3 [m].
2. The information provided by the local map, as well as the information of velocity of the OMW were used for varying the stiffness of a haptic joystick that sents information to the hand of the occupant of the OMW. As the distance to the nearer obstacles decreases and the velocity of the OMW increases, the stiffness of the haptic joystick increases, and vice versa. By using the haptic joystick, the occupant of the OMW was able of achieving safety navigation by avoiding collision against obstacles. The sensing system to obtain the surrounding environmental information for any arbitrary direction in real time was built. The algorithm to choose only environmental information existing toward the moving direction of OMW for navigation support system was proposed. Using the constructed environmental recognition system, operation assistance system that informs the danger level of collision to the operator was given. Navigation guidance haptic feedback system that induces an evasive movement to navigate OMW toward the direction without obstacle was proposed.
3. A power assist system was attached to the rear part of the OMW in order to provide support to the attendants of the OMW, specially in the case when the attendant of the OMW is a senior citizen. The operability of the OMW with power assistance was improved by using fuzzy reasoning, but it was found that the membership functions of the fuzzy reasoning system had to be tuned in order to respond to the individual characteristics of each attendant. A neuro-fuzzy system (ANFIS) was used for speeding the tuning of the fuzzy reasoning system of the OMW by using the input data of the attendants. A touch panel with display was attached to the rear part of the OMW for providing a human-friendly interface for the input of the teaching data of the neuro-fuzzy system. Moreover, this touch panel can be used by the attendant for knowing the difference between the desired motion and the real motion of the OMW, and then adjust his behavior according to his observation. The operability of the OMW was improved by using the combined system ANFIS-touch panel.
4. The natural frequencies of the OMW and the natural frequencies of the head and torso

of the occupant of the OMW were suppressed by using the Hybrid Shape Approach (HSA). A human model that considers just the head and the torso of the human being was developed for evaluating the results obtained when the HSA was used. It was found that it was possible to reduce the vibration of the head and torso of the occupant of the OMW by using the HSA.

8. Acknowledgment

We would like to sincerely acknowledge Dr. Y. Noda, Toyohashi University of Technology, and Mr. T. Beppu, T. Kobayashi, T. Nishigaki, Y. Yang and Y.Kondo for author's past graduate students who have collaborated under the supervision of Prof. K. Terashima.
This work was supported in part by COE Program "Intelligent Human Sensing" and furthermore, Global COE Program "Frontiers of Intelligent Sensing" from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

8. References

Ae, M. et al (1992), Estimation of Inertia Properties of the Body Segments in Japanese Athletes, *Journal of Bio-mechanism*, Vol. 11, pp. 23-33

Alsuwaiyan, A. S. & Shaw, S. W. (1999), Localization of Free Vibration Modes in Systems of Nearly-Identical Vibration Absorbers, *Journal of Sound and Vibration*, Vol. 228, No. 3, pp. 703-711

Argyros, A. et al (2002), Semi-autonomous Navigation of a Robotic Wheelchair, *Journal of Intelligent and Robotic Systems,* Vol. 32, pp. 315- 329.

Borgolte, U. et al (1998), Architectural Concepts of a Semi-autonomous Wheelchair, *Journal of Intelligent and Robotic Systems*, Vol. 22, pp. 233-253.

Esen, H. et al (2004), A Virtual Environment Medical Training System For Bone Drilling with 3 DOF Force Feedback, *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, pp. 3631-3636

Funakoshi, M. et al (2004), Measurement of Whole Body Vibration in Taxi Drivers, *Journal of Occupational Health*, Vol. 46, pp. 119-124.

Furusho, H.; Yokoya, K. & Fujiki, S. (1969), Analysis of Occupant's Movements in Head-on Collision (Part1), *Journal of the Society of Automotive Engineering of Japan*, Vol.23, No.10, pp. 1046-1054.

Gen, M. & Cheng, R. (2000), *Genetic Algorithms and Engineering Optimization*, Wiley Series in Engineering Design and Automation.

Goldstein, B. A., et al. (2005), Tinnitus Improvement with Ultra-High-Frequency Vibration Therapy, *International Tinnitus Journal*, Vol. 11, No. 1, pp. 14-22.

Griffin, M. J. (2001), The Validation of Biodynamic Models, *Clinical Biomechanics*, Vol.16 Supplement No. 1, pp. S81-S92.

Hanson, L.; Wienholt, W. & Sperling, L. (2003), A Control Handling Comfort Model Based on Fuzzy Logics, *International Journal of Industrial Ergonomics*, Vol. 31, pp. 87-100.

Harris, C. J. et al (1993), *Intelligent Control*, World Scientific.

Hayashibara, Y. et al (1999), Assist System for Carrying a Long Object with a Human-Analysis of a Human Cooperative Behavior in the Vertical Direction, *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 695-

700,.

Holve, R. et al (1995), Generating Fuzzy Rules for the Acceleration Control of an Adaptive Cruise Control System, *Proceedings of the NAFIPS Conference*

Hogan, N. (1985a), Impedance Control: An approach to Manipulation: Part I - Theory, *Journal of Dynamic Systems, Measurement and Control*, Vol. 107, No. 1, pp. 1-7.

Hogan, N. (1985b), Impedance Control: An approach to Manipulation: Part II - Implementation, *Journal of Dynamic Systems, Measurement and Control*, Vol. 107, No. 1, pp. 8-16.

Hogan, N. (1985c), Impedance Control: An approach to Manipulation: Part III - *Journal of Dynamic Systems, Measurement and Control*, Vol. 107, No. 1, pp. 17-24

Jang, J. (1993), ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685.

Jang, J. et al. (1997), *Neuro-Fuzzy and Soft Computing*, Prentice Hall.

Juang, C. F. and Lin, C. T. (1998), An On-Line Self-Constructing Neural Fuzzy Inference Network and Its Applications, *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 1, pp. 12-32.

Kaneko, C., et al (2005), Evaluation of Whole-Body Vibration by the Category Judgment Method, *Industrial Health 2005*, Vol. 43, pp. 221-232.

Kaneko, C. et al. (2005), Scaling and Evaluation of Whole-Body Vibration by the Category Judgment Method, *Yamaha Motor Technical Review*.

Kawai, S. et al. (2004), A Study For Control of a Power Assist Device, *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2283-2288

Kitagawa, H.; Kobayashi, T.; Beppu, T. & Terashima, K. (2001), Semi-Autonomous Obstacle Avoidance of Omni-directional Wheelchair by Joystick Impedance Control, *Proc. IEEE/RSJ Int. Symposium on Intelligent Robots and Systems*, pp. 2148-2153

Kitagawa, H.; Beppu, T; Kobayashi, T & Terashima, K. (2002), Motion Control of Omni-directional Wheelchair Considering Patient Comfort, *Proceedings of the IFAC World Congress*, T-Tu-E20

Kitagawa, H.; Nishigaki, T.; Miyoshi, T. & Terashima, K. (2004a), Fuzzy Power Assist Control System for Omni-directional Transport Wheelchair, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp.1580-1585

Kitagawa, H.; Beppu, T; Ohno, Y.; Miyoshi, T. & Terashima, K. (2004b), Motion Control of Omni-directional Wheelchair Considering User's Comfort, *Journal of the Robotics Society of Japan*, Vol.22, No.7, pp.933-939

Kitagawa, H.; Nishisaka, S.; Miyoshi, T & Terashima, K. (2005), Development of Power Assist System for Omni-directional Transport Wheelchair, *Journal of the Robotics Society of Japan*, Vol.23, No.3, pp.321-329

Miyoshi, T. & Terashima, K., (2004), Fuzzy Power Assist Control System for Omni-directional Transport Wheelchair, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 1580-1585

Kondo, Y.; Miyoshi, T.; Terashima, K. & Kitagawa, H. (2008), Navigation Guidance Control Using Haptic Feedback for Obstacle Avoidance of Omni-directional Wheelchair, *Proceedings of Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems 2008*, pp. 437-444

Kubo, M. et al (2001), An investigation into a synthetic vibration model for humans: An investigation into a mechanical vibration human model constructed according to

the relations between the physical, psychological and physiological reactions of humans exposed to vibration, *International Journal of Industrial Ergonomics*, Vol. 27, pp. 219-232.

Kumar, V. et al. (1997), *Assistive Devices for People with Motor Disabilities*, Wiley Encyclopedia of Electrical and Electronics Engineering.

Lee, H. et al (1999), Control of Mobile Manipulators for Power Assist Systems, *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 989-994

Levine, S. et al (1999), The NavChair Assistive Wheelchair Navigation System, *IEEE Transactions on Rehabilitation Engineering*, Vol. 7, pp. 443 - 451.

Lewis, C. & Griffin, M. (2002), Evaluating the Vibration Isolation of Soft Seat Cushions Using an Active Anthropodynamic Dummy, *Journal of Sound and Vibration*, Vol. 253, No. 1, pp. 295-311.

Lian, T. et al (1999), Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm, IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, Vol. 29, No. 2, pp. 227-236.

Lin, C. T. & George Lee (1991), C. S., Neural-Network-Based Fuzzy Logic Control and Decision System, *IEEE Transactions on Computers*, Vol. 40, No. 12, pp. 1321-1336.

MathWorks (2002), *Fuzzy Logic Toolbox User's Guide Version 2*, The Mathworks Inc.

Maeda, H. et al (2000), Development of Omni-Directional Cart with Power Assist System (in Japanese), *Proceedings of the 18th Annual Conference of the Robotics Society of Japan*, pp.1155-1156.

Mamdani, E. H. & Assilian, S. (1985), An experiment in linguistic synthesis with a fuzzy logic controller, *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13.

Matsumoto, Y. & Griffin, M. J. (2001), Modeling the Dynamic Mechanisms Associated with the Principal Resonance of the Seated Human Body, *Clinical Biomechanics* , Vol. 16, No. 1, pp. S31-S44.

Matsuoka, Y. (2000a), Vibration Simulation Model for the Transportation of Wheelchair-bound Passengers, *KANSEI Engineering International*, pp. 47-52.

Matsuoka, Y. (2000b), Vibration Evaluation Model on the Wheelchair Transporting Apparatus, *KANSEI Engineering International*, pp. 52-60.

Naruse, K. et al (2005), Three-dimensional Lifting-up Motion Analysis for Wearable Power Assist Device of Lower Back Support, *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3126-3131, (2005).

Nguyen, H. T. et al (2003), *First Course in Fuzzy and Neural Control*, Chapman & Hall CRC.

Nishiyama, S. (1993), Development of Simulation System Vehicle-occupant Dynamic Interaction (in Japanese), *Japanese Society of Mechanical Engineering*, Vol. 59, No. 568) pp. C, 9.

Okada, A. (1980), Sense of Vibration of Man, *Journal of Society of Automotive Engineers of Japan*, Vol. 34, No. 5, pp. 440-450.

Paddan, G. S. & Griffin, M. J. (2000), Transmission of Yaw Seat Vibration to the Head, *Journal of Sound and Vibration*, Vol. 229 No. 5, pp. 1077-1095.

Paddan, G. S. & Griffin, M. J. (2002), Evaluation of Whole-Body Vibration in Vehicles, *Journal of Sound and Vibration*, Vol. 253, No. 1, pp. 195-213.

Parsons, K. C. (2000), Environmental Ergonomics: A Review of Principles, Methods and Models, *Applied Ergonomics*, Vol. 31, pp. 581-594.

Park, S. et al (2001), Single-mode Vibration Suppression for a Beam-Mass-Cart System Using

Input Pre-shaping with a Robust Internal Loop Compensator, *Journal of Sound and Vibration*, Vol. 241, No. 4, pp. 693-716.

Pin, F. & Killough, S. (1994), A new family of omni-directional and holonomic wheeled platforms for mobile robots, *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, pp. 480-489.

Protho, J. et al (2000), An Evaluation of an Obstacle Avoidance Force Feedback Joystick, *Proceedings of the Annual RESNA Conference*, Florida, pp. 447-449

Qiu, Y. & Griffin, M. J. (2004), Transmission of Vibration to the Backrest of a Car Seat Evaluated with Multi-input Models, *Journal of Sound and Vibration*, Vol. 288, pp. 297-321.

Qiu, Y. & Griffin, M. J. (2005), Transmission of Roll, Pitch and Yaw vibration to the Backrest of a Seat Supported on a Non-rigid Car Floor, *Journal of Sound and Vibration*, Vol. 274, pp. 1197-1222.

Sanada, K. et al (2005), A Study on Design and Evaluation of a Power-assisted Chair, *Proceedings of the SICE Annual Conference 2005*, pp. 3074-3078

Sato, R.; Iwahashi, Y. & Matsuoka, Y. (2003), Vibration Absorber of Wheelchair Transporting Apparatus and its Effectiveness on Riding Comfort, *The Science of Design*, Vol.50, No.1, pp. 11-18.

Seki, H. et al (2005), Novel Driving Control of Power Assisted Wheelchair Based on Minimum Jerk Trajectory, *IEEJ Transactions on Electronics, Information and Systems*(in Japanese), Vol. 125-C, No. 7, pp. 1133 - 1139.

Shaw, I. S. (1998), *Fuzzy control of industrial systems : theory and applications*, Kluwer Academic Publishers.

Stearns, S. (2003), *Digital Signal Processing*, CRC Press.

Smith, S. (2000), Modeling Differences in the Vibration Response Characteristics of the Human Body, *Journal of Biomechanics*, Vol. 33, pp. 1513-1516.

Sue, J. et al (2006), An Automatic Travel Control of a Container Crane using Neural Network Predictive PID Control Technique, *International Journal of Precision Engineering and Manufacturing*, Vol. 7, No. 1, pp. 35-41.

Sugeno, M. & Kang, G. T (1998), Structure identification of fuzzy model, *Fuzzy Sets and Systems*, Vol. 28, No. 1, pp. 15-33.

Tahboub, K. (2001), A Semi-Autonomous Reactive Control Architecture, *Journal of Intelligent and Robotic Systems*, Vol. 32, pp. 445- 459.

Takagi, T. & Sugeno, M. (1985), Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, No. 1, pp. 116 - 132.

Terashima, K.; Kitagawa H., Miyoshi, T. & Urbano, J. (2004), Frequency Shape Control of Omni-directional Wheelchair to Increase User's Comfort, *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3119-3124.

Terashima, K.; Urbano, J. & Kitagawa, H. (2006), Enhancement of Maneuverability of a Power Assist Omni-directional Wheelchair by Application of Neuro-Fuzzy Control, *Proceedings of the 3rd International Conference on Informatics in Control Robotics and Automation (ICINCO 2006)*, pp. 67-75.

Trampe, J. (1990), *Principles of Experimental Frequency Analysis*, Elsevier Applied Science.

The Japanese Standards Association, (JSA), (2000), *Evaluation of the whole body vibration: TR Z 0006* (in Japanese), The Japanese Industry Standard Committee Basic Sectional

Meeting.

Ueberle, M. & Buss, M. (2002), Design, Control, and Evaluation of a New 6 DOF Haptic Device, *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, pp. 2949-2954, (2002).

Urbano, J.; Terashima, K.; Miyoshi, T & Kitagawa, H., (2005a), Collision Avoidance in an Omni-directional Wheelchair by using Haptic Feedback, *WSEAS Transactions on Systems*, Vol. 4, No. 1, pp. 79-84.

Urbano, J. ; Yang, Y.; Terashima, K.; Miyoshi, T. & Kitagawa, H., (2005b), Navigation with Comfort of Omni-directional Wheelchair Driven by Joystick, *Proceedings of the IFAC World Congress*, Tu-M04-TP/14, (2005).

Urbano, J.; Terashima, K.; Nishigaki, T.; Miyoshi, T. & Kitagawa, H., (2005c), Development of Power Assist on Omni-directional Mobile Wheelchair Considering Operability and Comfort, *Proceedings of the 2nd International Conference on Informatics in Control Robotics and Automation (ICINCO 2005)*, (2005).

Urbano, J.; Terashima, K. & Kitagawa, H., (2006a), Neuro-Fuzzy Control of a Power Assist Omni-directional Wheelchair to Enhance Maneuverability, *Proceedings of the IEEE International Conference on Control Applications (CCA)*, pp. 939-946

Urbano, J. ; Terashima, K. & Kitagawa, H., (2006b), Skill-Assist Control of an Omni-directional Wheelchair by Neuro-Fuzzy Systems Using Attendants' Force Input, *International Journal of Innovative Computing, Information and Control*, Vol. 2, No. 6, pp. 1219-1248.

Wada, M. & Asada, H. (1999), Design and Control of a Variable Footprint Mechanism for Holonomic Omnidirectional Vehicles and its Application to Wheelchairs, *IEEE Transanctions on Robotics and Automation*, Vol. 15, No. 6, pp. 978-989.

West, M. & Asada, H. (1992), Design of a holonomic omni-directional vehicle, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 97-103

Wu, Y. et al. (2004), Development of a Power Assisting System of a Walking Chair, *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3207-3212

Yamada, Y. et al (2002), Proposal of Skill-Assist for Mounting Operations in Automobile Assembly Processes, *Transactions of the Japan Society of Mechnical Engineers* (in Japanese), Vol. 68, No. 666, pp. 509-516.

Yanco, H. et al (1995), Initial Report on Wheelesley: A Robotic Wheelchair System, *Proceedings of the Workshop on Developing AI Applications for the Disabled, International Joint Conference on Artificial Intelligence*

Yano, K.; S. Higashikawa & K. Terashima (2000), A Vibration Control Design of Liquid Container Transfer System by Hybrid Shaped Approach Considering the Both of Frequency Characteristics and Time Response, *Proceedings of Movic 2000*, pp. 279-284, Sydney

Yoneda, I. et al (1997), Development of a manual attendant-controlled wheelchair with a foldable plate seat and a mechanism absorbing vibration, *Proceeding of 12th Japanese Conference of Advancement of Rehabilitation Technology*, vol.12, pp. 33-36.

Zhao, X. & Chaffin, D. (2000), A three-dimensional dynamic posture prediction model for simulating in vehicle seated reaching movements: development and validation, *Ergonomics*, Vol. 43, pp.1314-1330.

14

Modeling of a Thirteen-link 3D Biped and Planning of a Walking Optimal Cyclic Gait using Newton-Euler Formulation

David Tlalolini, Yannick Aoustin, Christine Chevallereau
Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN), École Centrale de Nantes, Université de Nantes, U.M.R. 6597, 1 rue de la Noë, BP 92101, 44321 Nantes Cedex 3, France.
e-mail: surname.name@irccyn.ec-nantes.fr

1. Introduction

Preliminaries. The design of walking cyclic gaits for legged robots and particularly the bipeds has attracted the interest of many researchers for several decades. Due to the unilateral constraints of the biped with the ground and the great number of degrees of freedom, this problem is not trivial. Intuitive methods can be used to obtain walking gaits as in (Grishin et al. 1994). Using physical considerations, the authors defined polynomial functions in time for an experimental planar biped. This method is efficient. However to build a biped robot and to choose the appropriate actuators or to improve the autonomy of a biped, an optimization algorithm can lead to very interesting results. In (Rostami & Besonnet 1998) the Pontryagin's principle is used to design impactless nominal trajectories for a planar biped with feet. However the calculations are complex and difficult to extend to the 3D case. Furthermore the adjoint equations are not stable and highly sensitive to the initial conditions (Bryson & Ho 1995). As a consequence a parametric optimization is a useful tool to find optimal motion. For example in robotics, basis functions as polynomial functions, splines, truncated fourier series are used to approximate the motion of the joints, (Chen 1991; Luca et al. 1991; Ostrowski et al. 2000; Dürrbaum et al. 2002; Lee et al 2005; Miossec & Aoustin 2006; Bobrow et al 2006). The choice of optimization parameters is not unique. The torques, the Cartesian coordinates or joint coordinates can be used. Discrete values for the torques defined at sampling time are used as optimization parameters in (Roussel et al. 2003). However it is necessary, when the torque is an optimized variable, to use the direct dynamic model to find the joint accelerations and integrations are used to obtain the evolution of the reference trajectory in velocity and in position. Thus this approach requires much calculations: the direct dynamic model is complex and many evaluations of this model is used in the integration process. In (Beletskii & Chudinov 1977; Bessonnet et al. 2002; Channon et al. 1992; Zonfrilli & Nardi 2002; Chevallereau & Aoustin 2001; Miossec & Aoustin 2006) to overcome this difficulty, directly the parametric optimization defines the reference trajectories of Cartesian coordinates or joint coordinates

for 2D bipeds with feet or without feet. An extension of this strategy is given in this paper to obtain a cyclic walking gait for a 3D biped with twelve motorized joints.

Methodology. A half step of the cyclic walking gait is uniquely composed of a single support and an instantaneous double support which is modeled by passive impulsive equations. This walking gait is simpler than the human gait. But with this simple model the coupling effect between the motion in frontal plan and sagittal plane can be studied. A finite time double support phase is not considered in this work currently because for rigid modeling of robot, a double support phase can usually be obtained only when the velocity of the swing leg tip before impact is null. This constraint has two effects. In the control process it will be difficult to touch the ground with a null velocity, as a consequence the real motion of the robot will be far from the ideal cycle. Furthermore, large torques are required to slow down the swing leg before the impact and to accelerate the swing leg at the beginning of the single support. The energy cost of such a motion is higher than a motion with impact in the case of a planar robot without feet (Chevallereau & Aoustin 2001; Miossec & Aoustin 2006). The evolution of joint variables are chosen as spline functions of time instead of usual polynomial functions to prevent oscillatory phenomenon during the optimization process (see Chevallereau & Aoustin 2001; Saidouni & Bessonnet 2003 or Hu & Sun 2006). The coefficients of the spline functions are calculated as functions of initial, intermediate and final configurations, initial and final velocities of the robot. These configuration and velocity variables can be considered as optimization variables. Taking into account the impact and the fact that the desired walking gait is periodic, the number of optimization variables is reduced. In other study the periodicity conditions are treated as equality constraints (Marot 2007). The cost functional considered is the integral of the torque norm, which is a common criterion for the actuators of robotic manipulators, (Chen 1991; Chevallereau & Aoustin 2001; Bobrow et al. 2001; Garg & Kumar 2002). During the optimization process, the constraints on the dynamic balance, on the ground reactions, on the validity of impact, on the limits of the torques, on the joints velocities and on the motion velocity of the biped robot are taken into account. Therefore an inverse dynamic model is calculated during the single phase to obtain the torques for a suitable number of sampling times. An impulsive model for the impact on the ground with complete surface of the foot sole of the swing leg is deduced from the dynamic model for the biped in double support phase. Then it is possible to evaluate cost functional calculation, the constraints during the single support and at the impact.

Contribution. The dynamic model of a 3D biped with twelve degrees of freedom is more complex than for a 2D biped with less degrees of freedom. So its computation cost is important in the optimization process and the use of Newton-Euler method to calculate the torque is more appropriate than the Lagrange method usually used. Then for the 3D biped, in single support, our model is founded on the Newton Euler algorithm, considering that the reference frame is connected to a stance foot. The walking study includes impact phase. The problem solved in (Lee et al. 2005; Huang & Metaxas 2002) is to obtain an optimal motion beginning at a given state and ending at another given state. Furthermore authors used Lie theoretic formulation of the equations of motion. In our case the objective is to define cyclic walking for the 3D Biped. Lie theoretic formulation is avoided because for rigid bodies in serial or closed chains, recursive ordinary differential equations founded on the Newton-Euler algorithm is appropriate see (Angeles 1997).

Structure of the paper. The paper is organized as follows. The 3D biped and its dynamic model are presented in Section 2. The cyclic walking gait and the constraints are defined in Section 3. The optimization parameters, optimization process and the cost functional are discussed in Section 4. A summarize of the global optimization process is given in Section 5. Simulation results are presented in Section 6. Section 7 contains our conclusion and perspectives.

2. Model of the biped robot

2.1 Biped model

We considered an anthropomorphic biped robot with thirteen rigid links connected by twelve motorized joints to form a serial structure. It is composed of a torso, which is not directly actuated, and two identical open chains called legs which are connected at the hips. Each leg is composed of two massive links connected by a joint called knee. The link at the extremity of each leg is called foot which is connected at the leg by a joint called ankle. Each revolute joint is assumed to be independently actuated and ideal (frictionless). The ankles of the biped robot consist of the pitch and the roll axes, the knees consist of the pitch axis and the hips consist of the roll, pitch and yaw axes to constitute a biped walking system of two 2-DoF (two degrees of freedom) ankles, two 1-DoF knees and two 3-DoF hips as shown in figure 1. The action to walk associates single support phases separated by impacts with full contact between the sole of the feet and the ground, so that a model in single support, and an impact model are derived. The dynamic model in single support is used to evaluate the required torque thus only the inverse dynamic model is necessary. The impact model is used to determine the velocity of the robot after the impact, the torques are zero during the impact, thus a direct impact model is required. Since we use the Newton Euler equations to derive the dynamic model, the direct model is much more complicated to obtain than the inverse model.

The periodic walk studied includes a symmetrical behavior when the support is on right leg and left leg, thus only the behavior during an half step is computed, the behavior during the following half step is deduced by symmetry rules. As a consequence only the modeling on leg 1 is considered in the following.

2.2 Geometric description of the biped

For a planar robot any parameterization of the robot can be used, for a 3D model of robot with many degrees of freedom a systematic parameterization of the robot must be developed. Many studies have been conducted for the manipulator robot, thus the parameterization proposed for the manipulator robot is re-used for the walking robot. The first difficulty is to choose a base link for a walking robot. Since the leg one is in support during all the studied half step. The supporting foot is considered as base link.

To define the geometric structure of the biped walking system we assume that the link 0 (stance foot) is the base of the biped robot while the link 12 (swing foot) is the terminal link. Therefore we have a simple open loop robot which the geometric structure can be described using the notation of (Khalil & Kleinfinger 1985). The definition of the link frames is presented in figure 1 and the corresponding geometric parameters are given in table 1. Frame R_0, which is fixed to the tip of the right foot (determined by the width l_p and the

length L_p), is defined such that the axis z_0 is along the axis of frontal joint ankle. The frame R_{13} is fixed to the tip of the left foot in the same way as R_0.

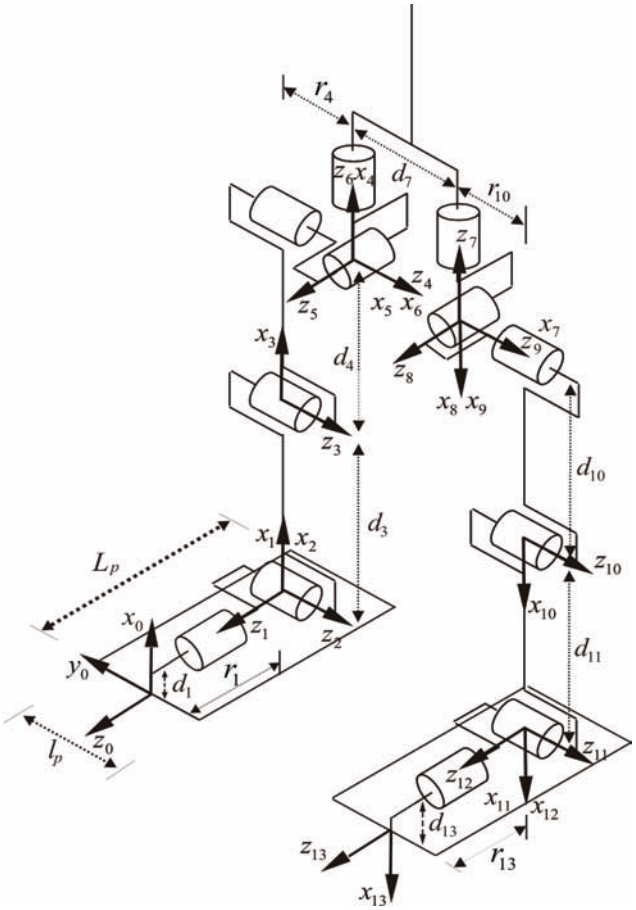


Fig. 1. Coordinate frame assignment for the biped robot.

j	a_j	α_j	q_j	r_j	d_j
1	0	0	θ_1	r_1	d_1
2	1	$\frac{\pi}{2}$	θ_2	0	0
3	2	0	θ_3	0	d_3
4	3	0	θ_4	r_4	d_4
5	4	$-\frac{\pi}{2}$	$\theta_5 - \frac{\Pi}{2}$	0	0
6	5	$-\frac{\pi}{2}$	θ_6	0	0

j	a_j	α_j	q_j	r_j	d_j
7	6	0	θ_7	0	d_7
8	7	$\frac{\pi}{2}$	$\theta_8 - \frac{\Pi}{2}$	0	0
9	8	$-\frac{\pi}{2}$	θ_9	0	0
10	9	0	θ_{10}	$r_{10} = r_4$	$d_{10} = d_4$
11	10	0	θ_{11}	0	$d_{11} = d_3$
12	11	$\frac{\pi}{2}$	θ_{12}	0	0
13	12	0	θ_{13}	$r_{13} = -r_1$	$d_{13} = d_1$

Table 1. Geometric parameters of the biped.

2.3 Dynamic model in single support phase

During the single support phase, our objective is only to determine the inverse dynamic model. The joint position, velocity and acceleration are known. The actuator torques must be calculated. Since the contact between the stance foot and the ground is unilateral, the ground reaction (forces and torques) must also be deduced. The Newton-Euler algorithm (see Khalil & Dombre 2002) must be adapted to determine the ground wrench. During the single support phase the stance foot is assumed to remain in flat contact on the ground, i.e. , no sliding motion, no take-off, no rotation. Therefore the biped is equivalent to a 12-DoF manipulator robot. Let $q \in R^{12}$ be the generalized coordinates, where $q_1,...,q_{12}$ denote the relative angles of the joints, $\dot{q} \in R^{12}$ and $\ddot{q} \in R^{12}$ are the velocity vector and the acceleration vector respectively. The dynamic model is represented by the following relation:

$$\begin{bmatrix} R_{F_R} \\ \Gamma \end{bmatrix} = f(q, \dot{q}, \ddot{q}, F_t) \tag{1}$$

where $\Gamma \in R^{12}$ is the joint torques vector, R_{F_R} is the ground wrench on the stance foot and F_t represents the external wrenches (forces and torques), exerted on links 1 to 12 . In single support phase we assume that $F_t = 0$.

The Newton-Euler method is used to calculate the dynamic model as defined in equation (1). This method proposed by Luh, Walker et Paul (Luh et al. 1980) is based on two recursive calculations. Associated with our choice of parameterization the following algorithm is obtained (Khalil & Dombre 2002). The forward calculation, from the base (stance foot) to the terminal link (swing foot) determines the velocity, the accelerations and the total forces and moments on each link. Then the backward calculations, from swing foot to stance foot, gives the joint torques and reaction forces using equation of equilibrium of each link successively.

Forward recursive equations

Taking into account that the biped robot remains flat on the ground, the initial conditions are:

$$\omega_0 = \mathbf{0},\ \dot{\omega}_0 = \mathbf{0} \text{ and } \dot{V}_0 = -\mathbf{g} \tag{2}$$

the real acceleration is $\dot{V}_0 = 0$ but the choice to write $\dot{V}_0 = -g$ allows to take into account the gravity effect.

For the link j with its associated frame R_j, and considering the link $j-1$ as its antecedent, its angular velocity ${}^j\omega_j$, and the linear velocity ${}^jV_j$ of the origin O_j of R_j are :

$${}^j\omega_j = {}^j\omega_{j-1} + \bar{\sigma}_j \dot{q}_j {}^j a_j \tag{2}$$

$${}^jV_j = {}^jA_{j-1}\left({}^{j-1}V_{j-1} + {}^{j-1}\omega_{j-1} \times {}^{j-1}P_j\right) + \sigma_j \dot{q}_j {}^j a_j \tag{3}$$

with ${}^jA_{j-1}$, the orientation matrix of the frame R_{j-1} in the frame R_j, $\sigma_j = 0$ when the j joint is a revolute joint, $\sigma_j = 1$ when the j joint is prismatic joint and $\bar{\sigma}_j = 1 - \sigma_j$, ${}^j a_j$ is an unit vector along the z_j axis, ${}^{j-1}P_j$ is the vector expressing the origin of frame R_j in frame R_{j-1}. The angular acceleration of link j and the linear acceleration of the origin O_j of R_j are:

$${}^j\dot{\omega}_j = {}^jA_{j-1}\,{}^{j-1}\dot{\omega}_{j-1} + \bar{\sigma}_j\left(\ddot{q}_j {}^j a_j + {}^j\omega_{j-1} \times \dot{q}_j {}^j a_j\right) \tag{4}$$

$${}^j\dot{V}_j = {}^jA_{j-1}\left({}^{j-1}\dot{V}_{j-1} + {}^{j-1}U_{j-1} \times {}^{j-1}P_j\right) + \sigma_j\left(\ddot{q}_j {}^j a_j + 2\,{}^j\omega_{j-1} \times \dot{q}_j {}^j a_j\right) \tag{5}$$

where ${}^jU_j = {}^j\hat{\dot{\omega}}_j + {}^j\hat{\omega}_j\,{}^j\hat{\omega}_j$. Matrices ${}^j\hat{\dot{\omega}}_j \in R^{3\times3}$ and ${}^j\hat{\omega}_j \in R^{3\times3}$ designate the skew matrices associated with the vectors ${}^j\dot{\omega}_j \in R^{3\times3}$ and ${}^j\omega_j \in R^{3\times3}$ respectively.

$$\hat{\omega} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{7}$$

The total inertial forces and moments for link j are:

$${}^jF_j = M_j\,{}^j\dot{V}_j + {}^jU_j\,{}^jMS_j \tag{6}$$

$${}^jM_j = {}^jJ_j\,{}^j\dot{\omega}_j + {}^j\omega_j \times \left({}^jJ_j\,{}^j\omega_j\right) + {}^jMS_j \times {}^j\dot{V}_j \tag{8}$$

with ${}^jJ_j$ inertia tensor of link j with respect to R_j frame, ${}^jMS_j$ is the first moments vector

of link j around the origin of R_j frame and M_j the mass of the link j. The antecedent link to the link 0 (stance foot) is not defined. For the iteration of the stance foot, only the equations (6) and (**Pogreška! Izvor reference nije pronađen.**) are used.

Backward recursive equations

The backward recursive equations are given as, for $j = 12,...,0$:

$$^{j}f_j = {}^{j}F_j + {}^{j}f_{j+1} \tag{7}$$

$$^{j-1}f_j = {}^{j-1}A_j \, {}^{j}f_j \tag{8}$$

$$^{j}m_j = {}^{j}M_j + {}^{j}A_{j+1} \, {}^{j+1}m_{j+1} + {}^{j}P_{j+1} \times {}^{j}f_{j+1} \tag{9}$$

These recursive equations will be initialized by the forces and moments exerted on the terminal link by the environment $^{j}f_{j+1}$ and $^{j}m_{j+1}$. In single support $^{j}f_{j+1} = 0$, $^{j}m_{j+1} = 0$. When $j = 0$, $^{0}f_0$ and $^{0}m_0$ are the forces exerted on the link 0 or the ground reaction force and moment rewritten as $^{0}F_R$ and $^{0}M_R$ expressed in the frame R_0.

If we neglect the friction and the motor inertia effects, the torque (or the force) Γ_j, is obtained by projecting m_j (or f_j) along the joint axis (z_j):

$$\Gamma_j = {}^{j}M_j + \left(\sigma_j \, {}^{j}f_j + \bar{\sigma}_j \, {}^{j}m_j\right)^{T} {}^{j}a_j \tag{10}$$

Γ_0 is not defined, since there is no actuator.

The ground reaction wrench is known in the frame R_0. This frame is associated with the stance foot, and the axis y_0, z_0 defined the sole of the stance foot. The position of the zero moment point (***ZMP***) position which is the point of the sole such that the moment exerted by the ground is zero along the axis y_0 and z_0 is such that:

$$y_{ZMP} = \frac{-{}^{0}m_{0z}}{{}^{0}f_{0x}} \tag{11}$$

$$z_{ZMP} = \frac{{}^{0}m_{0y}}{{}^{0}f_{0x}} \tag{12}$$

If the position of ***ZMP*** is within the support polygon, the biped robot is in dynamic equilibrium, the stance foot remains flat on the ground.

2.4. Impact model for the instantaneous double support

At the impact, the previous supporting foot becomes the swing foot, and its velocity after the impact can be different from zero. As a consequence the modeling of the robot must be able to described a non fixed stance foot. Since the dynamic model is calculated with the Newton-Euler algorithm, it is very convenient to define the velocity of the link 0 with the Newton variables: V_0 the linear velocity of the origin of frame R_0 and ω_0 the angular velocity.

For the impact model, or the double support model the robot position is expressed by $X = [X_0, \alpha_0, q]^T \in R^{18}$, X_0 and α_0 are the position and the orientation variables of frame R_0; the robot velocity is $V = [{}^0V_0, {}^0\omega_0, \dot{q}]^T \in R^{18}$ and the robot acceleration is $\dot{V} = [{}^0\dot{V}_0, {}^0\dot{\omega}_0, \ddot{q}]^T \in R^{18}$.

The impact model is deduced from the dynamic model in double support, when we assume that the acceleration of the robot and the reaction force are Dirac delta-functions.

The dynamical model in double support can be written:

$$D(X)\dot{V} + C(V,q) + G(X) + D_f R_f = D_\Gamma + D_R R_{F_R} \tag{13}$$

where $D \in R^{18\times 18}$ is the symmetric definite positive inertia matrix, $C \in R^{18}$ represents the Coriolis and centrifugal forces, $G \in R^{18}$ is the vector of gravity. $R_{F_R} = [F_R, M_R]^T \in R^6$ is the vector of the ground reaction forces on the stance foot, $R_f \in R^6$ represents the vector of forces F_{12} and moments M_{12} exerted by the swing foot on the ground, D_f, D_Γ and D_R are matrices that allows to take into account the forces and torques in the dynamic model.

The model of impact can be deduced from (13) and is:

$$D(X)\Delta V + D_f I_{R_f} = D_R I_{RF_R} \tag{14}$$

where I_{R_f} and $I_{R_{F_R}}$ are the intensity of Dirac delta-function for the forces R_f and R_{F_R}. ΔV is the variation of velocity at the impact, $\Delta V = V^+ - V^-$, where V^- is the velocity of the robot before impact and V^+ its velocity after impact.

The impact is assumed to be inelastic with complete surface of the foot sole touching the ground. This means that the velocity of the swing foot impacting the ground is zero after impact. Two cases are possible after an impact: the rear foot takes off the ground or both feet remain on the ground. In the first case, the vertical component of the velocity of the taking-off foot just after an impact must be directed upwards and the impulsive ground reaction in this foot equals zeros $I_{R_{F_R}} = 0$. In the second case, the rear foot velocity has to be zero just after an impact. The ground produces impulsive forces in both feet. This implies that the vertical component of the impulsive ground reaction in the rear foot (as in the fore foot) is directed upwards. An impacting foot with zero velocity at impact, is a solution of the two cases, there is no impact, the reaction forces on the two leg are zero and the velocity of the two feet after impact is zero.

For our numerical tests, for the robot studied, only the first case gives a valid solution. The swing foot is zero velocity before the impact (and there is no impact) or the previous stance foot does not remain on the ground after the impact. Thus, the impact dynamic model is (see (Formal'skii 1982) and (Sakaguchi 1995)):

$$D(X)\Delta V = -D_f I_{R_f} \tag{15}$$

$$D_f^T V^+ = 0 \tag{16}$$

$$\begin{bmatrix} {}^0V_0^- \\ {}^0\omega_0^- \end{bmatrix} = \begin{bmatrix} 0_{3\times1} \\ 0_{3\times1} \end{bmatrix} \tag{17}$$

These equations form a system of linear equations which determines the impulse forces I_{R_f} and the velocity vector of the biped after impact V^+.

$$I_{R_f} = \left(D_f^T D^{-1} D_f\right)^{-1} D_f^T V^- \tag{18}$$

$$V^+ = -D^{-1}D_f\left(D_f^T D^{-1} D_f\right)^{-1} D_f^T V^- + V^- \tag{19}$$

As the wrench R_f is naturally expressed in the frame R_{12} : ${}^{12}F_{12}$, ${}^{12}M_{12}$. The matrix D_f is the transpose of the Jacobian of velocity of the link R_{12} with respect to the robot velocity V. The velocities of link 12 can be expressed as :

$$\begin{bmatrix} V_{12} \\ \omega_{12} \end{bmatrix} = \begin{bmatrix} V_0 + \omega_0 \times {}^0P_{12} \\ \omega_{12} \end{bmatrix} + J_{12}\dot{q} \tag{20}$$

where ${}^0R_{12}$ is the vector linking the origin of frame R_0 and the origin of frame R_{12} expressed in frame R_0, $J_{12} \in R^{6\times12}$ is the Jacobian matrix of the robot, $J_{12}\dot{q}$ represents the effect of the joint velocities on the Cartesian velocity of link 12. The velocities V_{12} and ω_{12} must be expressed in frame R_{12}, thus we write:

$$\begin{bmatrix} {}^{12}V_{12} \\ {}^{12}\omega_{12} \end{bmatrix} = \begin{bmatrix} {}^{12}A_0 & -{}^{12}A_0\,{}^0\widehat{P}_{12} \\ 0_{3\times3} & {}^{12}A_0 \end{bmatrix} \begin{bmatrix} {}^0V_0 \\ {}^0\omega_0 \end{bmatrix} + {}^{12}J_{12}\dot{q} \tag{21}$$

where ${}^{12}A_0 \in R^{3\times3}$ is the rotation matrix, which defines the orientation of frame R_0 with respect to frame R_{12}. Term ${}^0\widehat{P}_{12}$ is the skew-symmetric matrix of the vector product associated with vector ${}^0P_{12}$.

For the calculation of the inertia matrix D, following the same way, as the force R_{F_R} is applied on the stance leg, in equation (13), $D_R = \left[I_{6\times6} \mid 0_{12\times6}\right]^T \in R^{18\times6}$. The matrix D_Γ defines the actuated joint thus we have : $D_\Gamma = \left[0_{6\times12} \mid I_{12\times12}\right]^T \in R^{18\times12}$.
When no force is applied on the swing leg, the dynamic model (13) becomes:

$$D(X)\dot{V} + C(V,q) + G(X) = \begin{bmatrix} R_{F_R} \\ \Gamma \end{bmatrix} \tag{22}$$

Since the stance foot is assumed to remain in flat contact, the resultant ground reaction force/moment R_{F_R} and the torques Γ can be computed using the Newton-Euler algorithm (see section 2.3). According to the method of (Walker & Orin 1982), the matrix D is calculated by the algorithm of Newton-Euler, by noting from (13), that the i^{th} column of D is equal to $\begin{bmatrix} R_{F_R} \\ \Gamma \end{bmatrix}$ if

$$V = \mathbf{0},\ g = 0,\ \dot{V} = e_i,\ R_f = 0 \tag{25}$$

$e_i \in R^{18\times1}$ is the unit vector, whose elements are zero except the i^{th} element which is equal to 1. The vectors $C(V,q)$ and $G(X)$ can be obtained in the same way that D, however for the impact model the knowledge of these vectors are not necessary.

3. Definition of the walking cycle

Because a walking biped gait is a periodical phenomenon our objective is to design a cyclic biped gait. A complete walking cycle is composed of two phases: a single support phase and a double support phase which is modeled through passive impact equations. The single support phase begins with one foot which stays on the ground while the other foot swings from the rear to the front. We shall assume that the double support phase is instantaneous. This means that when the swing leg touches the ground the stance leg takes off. There are two facets to be considered for this problem. The definition of reference trajectories and the method to determine a particular solution of it. This section is devoted to the definition of reference trajectories. The optimal process to choose the best solution of parameters, allowing a symmetric half step, from the point of view of a given cost functional will be described in the next section.

3.1. Cyclic walking trajectory

Since the initial configuration is a double support configuration, both feet are on the ground, the twelve joint coordinates are not independent. Because the absolute frame is attached to the right foot we define the situation of the left foot by $(y_{lf}, z_{lf}, \phi_{lf})$ and the situation of the middle of the hips $(x_h, y_h, z_h, \theta_h)$, both expressed in R_0 frame. (y_{lf}, z_{lf}) are the Cartesian coordinates, in the horizontal plane, of the left foot position, ϕ_{lf} denotes the left foot yawing

motion, (x_h, y_h, z_h) is the hip position and θ_h defines the hip pitching motion. The two others parameters, orientation for the middle of the hips in frontal and transverse planes, are considered to be equal to zero. The values of the joint variables are solution of the inverse kinematics problem for a leg, which may also be considered as a 6-link manipulator. The problem is solved with a symbolic software, (SYMORO+, see (Khalil & Kleinfinger 1985).
Let us consider, for the cyclic walking gait, the current half step in the time interval $[0, T_S]$. In order to deduce the final configuration of the biped robot at time $t = T_S$, we impose a symmetric role of the two legs, therefore from the initial configuration $q_0 = q(t = 0)$ in double support, the final configuration $q_{T_S} = q(t = T_S)$ in double support is deduced as:

$$q_{T_S} = Eq_0 \tag{23}$$

where $E \in R^{12\times 12}$ is an inverted diagonal matrix which describes the exchange of legs.
Taking into account the impulsive impact (15)-(17), we can compute the velocity vector of the biped after the impact. Therefore, the joint velocities after impact, $\dot{q}^+$ can be calculated when the joint velocities before the impact, $\dot{q}^-$ is known. The use of the defined matrix E allows us to calculate the initial joint velocities $\dot{q}_0 = \dot{q}(t = 0)$
for the current half step as:

$$\dot{q}_0 = E\dot{q}^+ \tag{24}$$

By this way the conditions of cyclic motion are satisfied.

3.2. Constraints

In order to insure that the trajectory is possible, many constraints have to be considered.

Magnitude constraints on position, velocities and torque:

- Each actuator has physical limits such that:

$$|\Gamma_i| - \Gamma_{i,max} \le 0, \quad \text{for } i = 1,...,12 \tag{25}$$

where $\Gamma_{i,max}$ denotes the maximum value for each actuator.

$$|\dot{q}_i| - \dot{q}_{i,max} \le 0, \quad \text{for } i = 1,...,12 \tag{26}$$

where $\dot{q}_{i,max}$ denotes the maximum velocity for each actuator.

- The upper and lower bounds of joints for the configurations during the motion are:

$$q_{i,min} \le q_i \le q_{i,max}, \quad \text{for } i = 1,...,12 \tag{27}$$

$q_{i,min}$ and $q_{i,max}$ respectively stands for the minimum and maximum joint limits.

Geometric constraints in double support phase:

- The distance $d(hip, foot)$ between the foot in contact with the ground and the hip must remain within a maximal value, i.e., :

$$d(hip, foot) \le l_{hip} \tag{28}$$

This condition must hold for initial and final configurations of the double support phase.

- In order to avoid the internal collision of both feet through the lateral axis the heel and the toe of the left foot must satisfy:

$$y_{heel} \le -a \quad \text{and} \quad y_{toe} \le -a \tag{31}$$

with $a > \frac{l_p}{2}$ and l_p is the width of right foot.

Walking constraints:

- During the single support phase to avoid collisions of the swing leg with the stance leg or with the ground, constraints on the positions of the four corners of the swing foot are defined.
- We must take into account the constraints on the ground reaction $R_{F_R} = \left[R_{F_{Rx}}, R_{F_{Ry}}, R_{F_{Rz}} \right]^T$ for the stance foot in single support phase as well as impulsive forces $I_{R_f} = \left[I_{R_{fx}}, I_{R_{fy}}, I_{R_{fz}} \right]^T$ on the foot touching the ground in instantaneous double support phase. The ground reaction in single support and the impulsive forces at the impact must be inside a friction cone defined by the friction coefficient μ. This is equivalent to write:

$$\sqrt{R_{F_{Ry}}^2 + R_{F_{Rz}}^2} \le \mu R_{F_{Rx}} \tag{32}$$

$$\sqrt{I_{R_{fy}}^2 + I_{R_{fz}}^2} \le \mu I_{R_{fx}} \tag{33}$$

- The ground reaction forces in single support and the impulsive forces at the impact only can push from the ground but cannot pull from ground, then the conditions of no take off are deduced:

$$R_{f_x} \geq 0 \tag{34}$$

$$I_{R_{f_x}} \geq 0 \tag{35}$$

- In order to maintain the balance in dynamic walking, the Zero Moment Point which is equivalent to the Center of Pressure (CoP), (Vukobratovic & Borovac 2004; Vukobratovic & Stepanenko 1972; Vukobratovic & Borovac 1990), of the biped's stance foot must be within the interior of the support polygon. Then for a rectangular foot the (CoP) must satisfy:

$$\frac{-l_p}{2} \leq CoP_y \leq \frac{l_p}{2} \tag{36}$$

$$-L_p \leq CoP_z \leq 0 \tag{37}$$

where l_p is the width and L_p is the length of the feet.

4. Parametric optimization

4.1. The cubic spline

To describe the joint motion by a finite set of parameters we choose to use for joint i, $(i = 1,...,12)$ a piecewise function of the form:

$$q_i = \varphi_i(t) = \begin{cases} \varphi_{i1}(t) & \text{if} \quad t_0 \leq t \leq t_1 \\ \varphi_{i2}(t) & \text{if} \quad t_1 \leq t \leq t_2 \\ \quad . & \\ \quad . & \\ \quad . & \\ \varphi_{in}(t) & \text{if} \quad t_{n-1} \leq t \leq t_n \end{cases} \tag{38}$$

where $\varphi_k(t)$ are polynomials of third-order such that:

$$\varphi_{ik}(a_{ik}, t) = \sum_{j=0}^{3} a_{ikj}(t - t_{k-1})^j, \quad k = 1,...,n \ \forall t \in [t_0, t_n] \tag{39}$$

where a_{ikj} are calculated such that the position, velocity and acceleration are always continuous in $t_1,...,t_{n-1}$. The motion is defined by specifying an initial configuration q_0, an initial velocity $\dot{q}_0$, a final configuration q_{T_s} and a final velocity $\dot{q}_{T_s}$ in double support, with

$n-1$ intermediate configurations in single support and T_S the duration of this single support.

4.2. Optimization parameters

A parametric optimization problem has to be solved to design a cyclic bipedal gait with successive single supports and passive impacts (no impulsive torques are applied at impact). For a half step defined on the time interval $[0, T_S]$ this problem depends on parameters to prescribe the $n-1$ intermediate configurations, the final velocity $\dot{q}_{T_S}$ in the single support phase and, using the geometric model, the limit configuration of the biped at impact. Taking into account the conditions (23) and (24) the minimal number of parameters necessary to define the joint are:

1. $(n-1)\times 12$ parameters are needed to define the $n-1$ intermediate configurations in single support phase.
2. The joint velocities of the biped before the impact are also prescribed by twelve parameters, $\dot{q}_i^-$ $(i=1,...,12)$.
3. The left foot yawing motion denoted by ϕ_{lf} and its position (y_{lf}, z_{lf}) in the horizontal plane as well as the situation of the middle of the hips defined by $(x_h, y_h, z_h, \theta_h)$ in double support phase are chosen as parameters.

Then the total number of parameters is: $19+(n-1)\times 12$. Let us remark that to define the initial and final configurations for the half step, when both feet touch the ground, nine parameters are required. However we define these configurations with six parameters only. These six parameters are defined by the vector $p=[p_1 \quad p_2 \quad p_3 \quad p_4 \quad p_5 \quad p_6]^T$ with the following geometric configuration data:

- p_1 : height of pelvis.
- p_2 : distance between the feet in the frontal plane each foot.
- p_3 : distance of the trunk with respect to the hip of the stance leg in the frontal plane.
- p_4 : orientation of the trunk in the sagittal plane.
- p_5 : position of the stance foot following y in frame R_0.
- p_6 : position of the stance foot following z in frame R_0.

The two others parameters, orientation of the middle of the hips in frontal and transverse planes, are fixed to zero. The duration of a half step, T_S, is arbitrarily fixed.

Four our numerical tests $n=3$ and then two intermediate configurations q_{int1} and q_{int2} of the 3D biped in single support are considered. To summarize, considering q, $\dot{q}$ and $\ddot{q}$ of which the components equal the basis functions q_i (**Pogreška! Izvor reference nije pronađen.**) and their associated time derivatives $\dot{q}_i$ and $\ddot{q}_i$, $i=1,...,12$, we can write:

$$q=\varphi(q_0, \dot{q}_0, q_{int1}, q_{int2}, q_{T_S}, \dot{q}_{T_S}) \tag{40}$$

$$\dot{q} = \dot{\varphi}(q_0, \dot{q}_0, q_{int1}, q_{int2}, q_{T_S}, \dot{q}_{T_S}) \tag{41}$$

$$\ddot{q} = \ddot{\varphi}(q_0, \dot{q}_0, q_{int1}, q_{int2}, q_{T_S}, \dot{q}_{T_S}) \tag{42}$$

where φ is the vector of components $\varphi_i(t)$ (**Pogreška! Izvor reference nije pronađen.**) defining the cubic splines for joint i, $i = 1,...,12$. The chosen vector of optimization parameters P_O can be written:

$$P_O = \begin{bmatrix} P_O(1) \\ P_O(2) \\ P_O(3) \\ P_O(4) \end{bmatrix} = \begin{bmatrix} q_{int1} \\ q_{int2} \\ \dot{q}_{T_S} \\ p \end{bmatrix} \tag{43}$$

4.3. Cost functional

In the optimization process we consider, as cost functional J_Γ, the integral of the norm of the torque divided by the half step length. In other words we are minimizing a quantity proportional to the lost energy in the actuators for a motion on a half step of duration T_S. This general form of minimal energy performance represents the losses by Joule effects for the electrical motors to cover distance d.

$$J_\Gamma = \frac{1}{d}\int_0^{T_S} \Gamma^T \Gamma dt \tag{44}$$

4.4. Statement of the optimization problem to design a cyclic walking gait for the 3D biped

Generally, many values of parameters can give a periodic bipedal gait satisfying constraints (25)-(**Pogreška! Izvor reference nije pronađen.**). A parametric optimization process, that objective is to minimize J_Γ under nonlinear constraints, is used to find a particular nominal motion with the splines (**Pogreška! Izvor reference nije pronađen.**) as basis functions. This optimization problem can be formally stated as:

$$\left.\begin{array}{l} \text{Minimize } J_\Gamma(P_O) \\ \text{subject to } g_j(P_O) \le 0 \quad j = 1,2,...,l \end{array}\right\} \tag{45}$$

where $J_\Gamma(P_O)$ is the cost functional to minimize with l constraints $g_j(P_O) \le 0$ to satisfy. These constraints are given in section 3.2. The optimization problem (**Pogreška! Izvor reference nije pronađen.**) is numerically solved by using the Matlab function ***fmincon***. This optimization function provides an optimization algorithm based on the Sequential Quadratic Programming (SQP). There are forty-three parameters for this nonlinear optimization problem: twenty-four for the two intermediate configurations in single

support, twelve for the joint velocities before the impact and seven to solve the inverse kinematics problem, subject to the constraints given by (25)-(**Pogreška! Izvor reference nije pronađen.**).

5. Algorithm for generating an optimal cyclic walking gait

In this section the algorithm to obtain an optimal cyclic walking gait for the biped is given.

- **Step1:** Given initial values for each components of parameter vector P_O (**Pogreška! Izvor reference nije pronađen.**).
- **Step 2:** With the parameters $P_O(4) = p$ compute the initial configuration and from the equation (23) the final configuration.
- **Step 3:** With the initial and final configurations, the parameters $P_O(3) = \dot{p}$ and the equations (15), (16) and (24) compute the initial velocity $\dot{q}_0$.
- **Step 4:** For time $t = 0$ to $t = T_S$, compute the spline functions (**Pogreška! Izvor reference nije pronađen.**) for the initial and final configurations and the parameters $P_O(1) = q_{int1}$ and $P_O(2) = q_{int2}$. Compute their first and second derives with respect to time.
- **Step 5:** For sampling time $\{0,...,t_k,...,T_S\}$, solve recursively the inverse dynamics (**Pogreška! Izvor reference nije pronađen.**)-(10) to compute the torques, the position of the Center of Pressure *CoP*, the constraints.
- **Step 6:** For sampling time $\{0,...,t_k,...,T_S\}$, approximate the integral of the square vector of torques to compute the cost functional.
- **Step 7:** Check convergence. If yes, terminate. If no, go to step1 for a new parameter vector P_O and begin a new optimization process.

6. Simulation results

To validate our proposed method, we present the results of an optimal motion for the biped, SPEJBL, shown in figure 2. SPEJBL has been designed in the Department of Control Engineering of the Technical University in Praha. Its physical parameters are given in table 2. The inertia of each link are also taken into account in the dynamic model. The results shown have been obtained with $T_S = 0.58\,s$. The optimal motion is such that the step length is $0.18m$ and the optimal velocity is $0.315\,m/s$. These values are results of the optimization process presented in Section IV, with the minimization of the cost functional (**Pogreška! Izvor reference nije pronađen.**) satisfying the constraints given by (25)-(**Pogreška! Izvor reference nije pronađen.**). The simulation of the optimal motion for one half step is illustrated in figure 3 and for 3 walking steps in figure 4. The normal components of the ground reactions, in function of time, of the stance foot during one half step in single support are presented in figure 5. The average vertical reaction force is $20\,N$, which is coherent with the weight of the robot which the mass equals $2.14\,Kg$. The chosen friction coefficient is 0.7. The figure 6 shows the *CoP* trajectory which is always inside the support polygon determined by $l_p = 0.11\,m$ and $L_p = 0.18\,m$, that is, the robot maintains the balance during the motion. Because the minimal distance between of *CoP* and the boundary of the

foot is large, smaller foot is acceptable for this cyclic motion. The figure 7 shows the evolutions of joint variables $q_i(t)$ $i = 1,...,12$, versus time, defined by the third-order spline function presented in Section III, in the single support phase during one half step. Let us remark that the evolution of each joint variable depends on the boundary conditions $(\dot{q}_i(0), \dot{q}_i(T_S)$ for $i = 1,...,12)$ and also on the intermediate configurations $\dot{q}_{i,int1}, \dot{q}_{i,int2}$ for $i = 1,...,12$ whose values are computed in the optimal process. For a set of motion velocities, the evolution of criterion J_Γ is presented in figure 8. With respect to the evolution of J_Γ we can conclude that the biped robot consumes more energy for low velocities to generate one half step. Due to the limitations of the joint velocities we could not obtain superior values to $0.36\ m/s$. The energy consumption increases probably for higher velocity (Chevallereau & Aoustin 2001). The robot has been designed to be able to walk slowly, this walk require large torque and small joint velocities. Its design is also based on large feet in order to be able to use static walking, as a consequence the feet are heavy and bulky, thus the resulting optimal motion is close to the motion of a human with snowshoes.

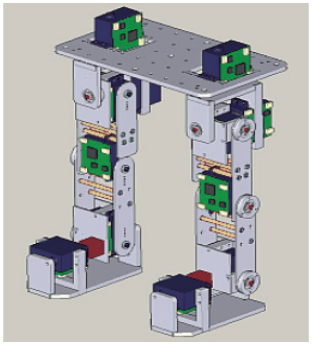


Fig. 2. Dimensional drawing of SPEJBL.

Physical Parameters	Mass (kg)	Length (m)
Torso	0.39	0.14
Hip joints	0.26	linked to torso
Thigh	0.12	0.12
Shin	0.05	0.12
Ankle joints	0.13	0.042
Foot	0.30	0.18x0.11

Table 2. Parameters of SPEJBL.

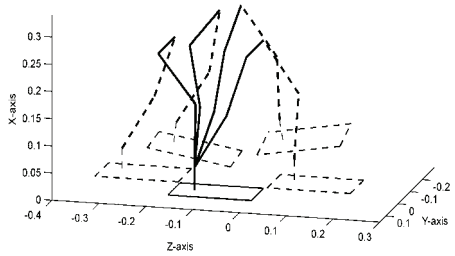


Fig. 3. Walking simulation for a half step.

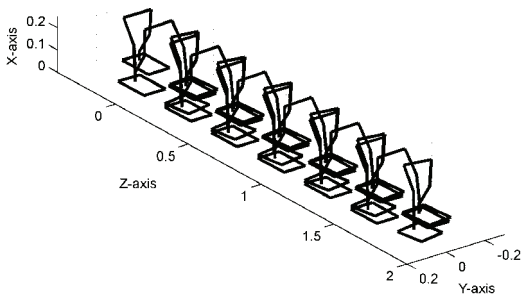


Fig. 4. Cyclic motion of biped SPEJBL.

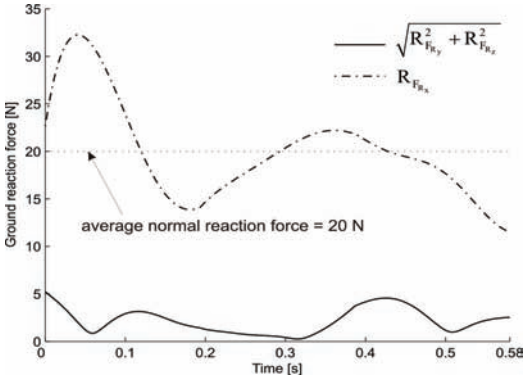


Fig. 5. The ground reaction force during the single support phase.

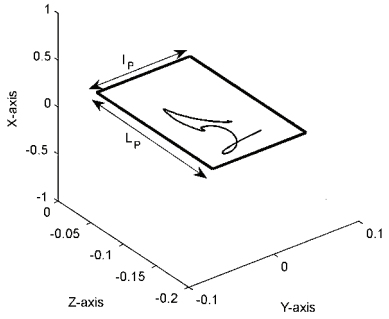
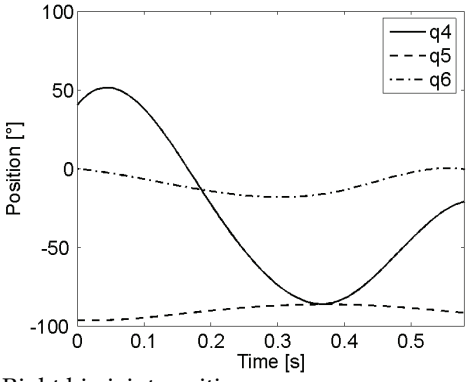
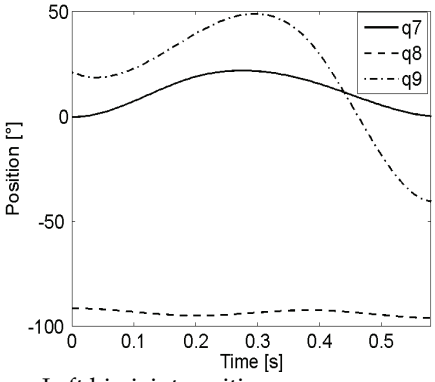


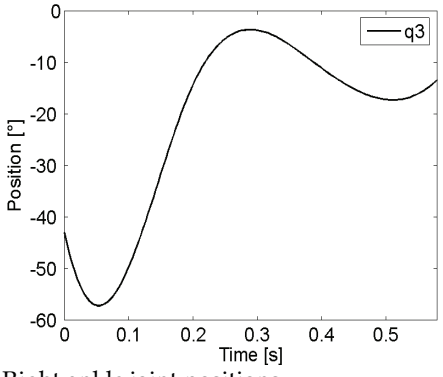
Fig. 6. The evolution of the *CoP* trajectory during a half step.



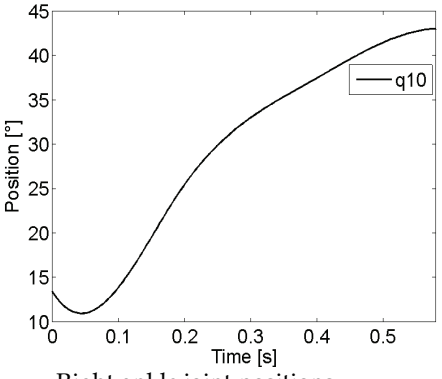
Right hip joint positions



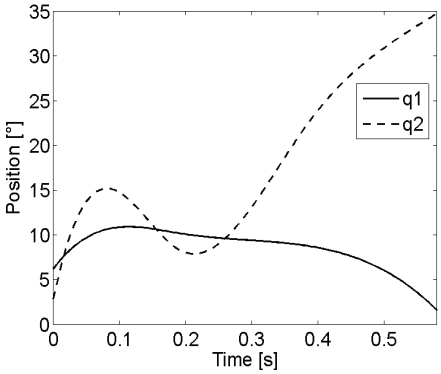
Left hip joint positions



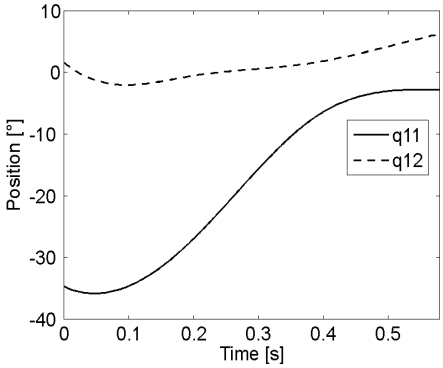
Right ankle joint positions



Right ankle joint positions



Right ankle joint position



Left ankle joint positions

Fig. 7. Evolution of joint positions.

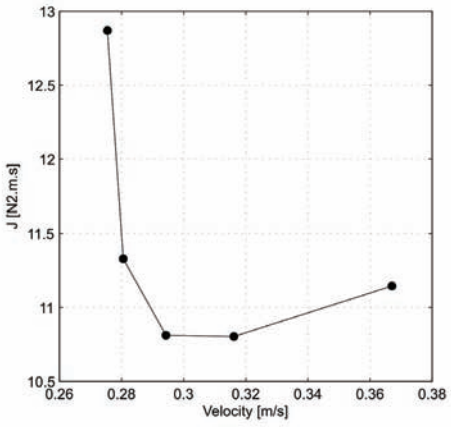


Fig. 8. J_Γ in function of several motion velocities for SPEJBL.

7. Conclusion

Optimal joint reference trajectories for cyclic walking gaits of a 3D experimental biped, SPEJBL are found. A methodology to design such optimal trajectories is developed. The definition of optimal trajectories is useful to test a robot design. In order to use classical optimization technique, the optimal trajectory is described by a set of parameters: we choose to define the evolution of the actuated relative angle as spline functions. A cyclic solution is desired. The number of the optimization variables is reduced by taking into account of the cyclicity condition explicitly.

Some inequality constraints such as the limits on the torques and the velocities, the condition of no sliding during motion and impact, some limits on the motion of the free leg are taken into account. The cost functional is calculated from the integral of the torques

norm. The torques are computed for sampling times using the inverse dynamic model. This model is obtained with the recursive Newton-Euler algorithm. The reference frame is connected to the stance foot. Optimal motions for a given duration of the half step have been obtained. The half step length and the advance velocity are the result of the optimization process. The numerical results obtained are realistic with respect to the size of the robot under study. Optimal motion for a given motion velocity can also be studied, in this case the motion velocity is considered as a constraint.

The proposed method to define optimal motion will be tested, considering a sub-phase of rotation of the supporting phase about the toe, closer to human. Another perspective is to evaluate the gradient of the cost functional and of the constraints with respect to the optimization parameters.

8. References

Grishin, A. A.; Formal'skii, A. M.; Lensky, A. V. & Zhitomirsky, S. V. (1994). Dynamic walking of a vehicle with two telescopic legs controlled by two drives, *International Journal of Robotics Research,* Vol. 13, No. 2, 137-147.

Rostami, M. & Besonnet, G. (1998). Rabbit: Impactless sagital gait of a biped robot during the single support phase, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 1385-1391.

Bryson, A. M. & Ho, Y. C. (1995). Applied optimal control., *Wiley*, New-York.

Chen, Y. C. (1991). Solving robot trajectory planning problems with uniform cubic b-splines, *Optimal Control Applications and Methods*, Vol. 12, No. 4, 247-262.

Luca, A. D; Lanari, L & Oriolo G. (1991). A sensitive approach to optimal spline robot trajectories, *Automatica*, Vol. 27, No. 3, 535-539.

Ostrowski, J. P.; Dessai, J. P. & Kumar V. (2000). Optimal gait selection for nonholonomic locomotion systems, *International Journal of Robotics Research*, Vol. 19, No. 3, 225-237.

Dürrbaum, A.; Klier, W. & Hahn H. (2002). Comparaison of automatic and symbolic differentiation in mathematical modeling and computer simulation of rigid-body, *Multibody System Dynamics*, Vol. 7, No. 4, 331-355.

Lee, S. H.; Kim, J.; Park, F.C.; Kim, M. & Bobrow, J.E. (2005). Newton-type algorithm for dynamics-based robot movement optimization, *IEEE Transactions on Robotics and Automation*, Vol. 21, No. 4, 657-667.

Miossec, S. & Aoustin, Y. (2006). Dynamical synthesis of a walking cyclic gait for a biped with point feet, *Special issue of lecture Notes in Control and information Sciences, Fast Motions in Biomechanics and Robotics, Ed. M. Dhiel and K. Mombaur, Springer-Verlag*.

Bobrow, J. E.; Park, F.C. & Sideris, A. (2006). Recent advance on the algorithme optimization of robot motion, *Special issue of lecture Notes in Control and information Sciences, Fast Motions in Biomechanics and Robotics, Ed. M. Dhiel and K. Mombaur, Springer-Verlag*.

Roussel, L. ; Canudas de Witt, C. & Goswami, A. (2003). Generation of energy optimal complete gait cycles for biped, *Proceedings of the 30th IEEE Conference on Robotics and Automation*, pp. 2036-2042.

Beletskii, V. V. & Chudinov, P. S. (1977). Parametric optimization in the problem of bipedal locomotion, *Izv. An SSSR. Mekhanika Tverdogo Tela [Mechanics of Solids]*, No. 1, 25-35.

Bessonnet, G.; Chesse, S. & Sardin, P. (2002). Generating optimal gait of a human-sized biped robot, *Proceedings of the CLAWAR'03 Conferencel*, pp. 717-724.

Channon, P. H.; Hopkins, S. H. & Pham D. T. (1992). Derivation of optimal walking motions for a bipedal walking robot, *Robotica*, Vol. 10, No. 2, 165-172.

Zonfrilli, F.; Oriolo, M. & Nardi, T. (2002). A biped locomotion strategy for the quadruped robot Sony ers-210, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 2768-2774.

Chevallereau, C. & Aoustin, Y. (2001). Optimal reference trajectories for walking and running of a biped, *Robotica*, Vol. 19, No. 5, 557-569.

Saidouni, T. & Bessonnet, G. (2003). Generating globally optimised saggital gait cycles of a biped robot, *Robotica*, Vol. 21, No. 2, 199-210.

Hu, C. Z. L. & Sun, Z. (2006). Biped gait optimization using spline function based probability model, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 830-835.

Luh, J.Y.S; Walker, M.W. & Paul, R.C.P. (1980). Resolved-acceleration control of mechanical manipulators. *IEEE Transactions on Automatic Control,* Vol. 25, No. 3, 468-474.

Marot, J. (2007). Contribution `a synthèse dynamique optimale de la marche, *PhD dissertation, Université de Poitiers, France*.

Bobrow, J. E.; Martin, B.; Sohl, G.; Wang, F. C. & Kim, K. (2001). Optimal robot motions fo physical criteria, *Journal of Robotic Systems*, Vol. 18, No. 12, 785-792.

Garg, D. P. & Kumar, M. (2002). Optimization techniques applied to multiple manipulators for path panning
and torque minimization, *Engineering Applications of Intelligence*, Vol. 15, 241-252.

Angeles, J. (1997). Fundamentals of Robotic Mechanical Systems, *Springer-Verlag, New-York*.

Khalil, W. & Kleinfinger, J. (1985). A new geometric notation for open and closed loop robots, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 1174-1180.

Khalil, W. & Dombre, E. (2002). Modeling, identification and control of robots, *Hermes Sciences Europe*.

Formal'skii, A. M. (1982). Locomotion of Anthropomorphic Mechanisms, *Nauka, Moscow* [In Russian], 368 pages.

Sakaguchi, F.; Furushu, J. & Koizumi, E. (1995). A realization of bounce gait in a quadruped robot with articular-joint-type legs, *Proceedings of IEEE Conference on Robotics and Automation*, pp. 697-702.

Walker, M.W. & Orin, D. E. (1982). Efficient dynamic computer simulation of robotics mechanism. *ASME Journal of Automatic Dynamic Systems Measurement and Control,* Vol. 104, No. 3, 205-211.

Vukobratovic, M. & Borovac, B. (2004). Zero moment point-thirty five years of its life, *International Journal of Humanoid Robotics,* Vol. 1, No. 1, 157-173.

Vukobratovic, M. & Stepanenko, Y. (1972). On the stability of anthropomorphic systems, *Mathematical Biosciences,* Vol. 15, 1-37.

Vukobratovic, M. & Borovac, B. (1990). Biped Locomotion-Dynamics, Stability, Control and Application, *Springer-Verlag*.

15

Robust Position Estimation of an Autonomous Mobile Robot

Touati Youcef, Amirat Yacine*, Djamaa Zaheer* & Ali-Chérif Arab
University Paris 8 Saint-Denis
France
**University Paris 12 Val de Marne*
France

1. Introduction

In the past few years, the topic of localization has received considerable attention in the research community and especially in mobile robotics area (Borenstein, 1996). It consists of estimating the robot's pose (position, orientation) with respect to its environment from sensor data. Therefore, better sensory data exploitation is required to increase robot's autonomy. The simplest way to estimate the pose parameters is integration of odometric data which, however, is associated with unbounded errors, resulting from uneven floors, wheel slippage, limited resolution of encoders, etc. However, such a technique is not reliable due to cumulative errors occurring over the long run. Therefore, a mobile robot must also be able to localize or estimate its parameters with respect to the internal world model by using the information obtained with its external sensors. In system localization, the use of sensory data from a range of disparate multiple sensors, is to automatically extract the maximum amount of information possible about the sensed environment under all operating conditions.

Usually, for many problems like obstacle detection, localization or Simultaneous Localization and Map Building (SLAM) (Montemerlo et al., 2002), the perception system of a mobile robot relies on the fusion of several kinds of sensors like video cameras, radars, dead-reckoning sensors, etc. The multi-sensor fusion problem is popularly described by state space equations defining the interesting state, the evolution and observation models. Based on this state space description, the state estimation problem can be formulated as a state tracking problem. To deal with this state observation problem, when uncertainty occurs, the probabilistic Bayesian approaches are the most used in robotics, even if new approaches like the set-membership one (Gning & Bonnifait, 2005) or Belief theory (Ristic and Smets, 2004) have proved themselves in some applications.

SLAM is technique used by mobile robots to build up a map within an unknown environment while at the same time keeping track of their current position. Several works implementing SLAM algorithms have been studied extensively over the last years in this direction, leading to approaches that can be classified into three well differentiated paradigms depending on the underlying map structure: metric (Sim et al., 2006) (Tardos et

al., 2002), topological (Ranganathan et al., 2006) (Savelli & Kuipers, 2004), or hybrid representations (Estrada et al., 2005) (Kuipers & Byun, 2001) (Dissanayake et al., 2001) (Thrun et al., 2004). These techniques deal mainly with the localization problem using mainly visual features and exteroceptive sensors, such as camera, GPS unit or laser scanner.
Localization algorithms have also been developed in sensors networks and applied in a myriad of applications such as intrusion detection, road traffic monitoring, health monitoring, reconnaissance and surveillance. Their main objectives is to estimate the location of sensors with initially unknown location information by using knowledge for absolute positions of a few sensors and their inter-sensor measurements such as distance and bearing measurements (Chong & Kumar, 2003) (Mao et al., 2007).
Ubiquitous computing technology is gradually being used to analyze people's activities. In this case, several research efforts on localization function have been conducted into recognizing human position and trajectories (Letchner et al., 2005) (Madhavapeddy & Tse, 2005) (Kanda et al., 2007). For example, Liao et al. used locations obtained via GPS with relational Markov model to discriminate location-based activities (Liao et al., 2005). Wen et al. developed an approach for inhabitant location and tracking system in a cluttered home environment via floor load sensors (Liau et al., 2008). In this approach, a probabilistic data association technique is applied to analyze the cluttered pressure readings collected by the load sensors so as to track their movements.
The main idea of data fusion methods is to provide a reliable estimation of robot's pose, taking into account the advantages of the different sensors (Harris, 1998). The main data fusion applied methods are very often based on probabilistic methods, and indeed probabilistic methods are now considered the standard approach to data fusion in all robotics applications. Probabilistic data fusion methods are generally based on Bayes' rule for combining prior and observation information. Practically, this may be implemented in a number of ways: through the use of the Kalman and extended Kalman filters, through sequential Monte Carlo methods, or through the use of functional density estimates.
There are a number of alternatives to probabilistic methods. These include the theory of evidence and interval methods. Such alternative techniques are not as widely used as they once were, however they have some special features that can be advantageous in specific problems.
The rest of the presented work is organized as follows. Section 2 discusses the problem statement and related works in the field of multi-sensor data fusion for the localization of a mobile robot. Section 3 describes the global localization system which is considered. We develop the proposed robust pose estimation algorithm in section 4 and its application is demonstrated in section 5. Simulation results and a comparative analysis with standard existing approaches are also presented in this section.

2. Background & related works

The Kalman Filter (KF) is the best known and most widely applied parameter and state estimation algorithm in data fusion methods (Gao, 2002). Such a technique can be implemented from the kinematic model of the robot and the observation (or measurement) model, associated to external sensors (gyroscope, camera, telemeter, etc.). The Kalman filter has a number of features which make it ideally suited to dealing with complex multi-sensor estimation and data fusion problems. In particular, the explicit description of process and

observations allows a wide variety of different sensor models to be incorporated within the basic algorithm. In addition, the consistent use of statistical measures of uncertainty makes it possible to quantitatively evaluate the role each sensor plays in overall system performance. Further, the linear recursive nature of the algorithm ensures that its application is simple and efficient. For these reasons, the Kalman filter has found wide-spread application in many different data fusion problems (Bar-Shalom, 1990) (Bar-Shalom & Fortmann, 1988) (Maybeck, 1979). In robotics, the KF is most suited to problems in tracking, localisation and navigation; and less so to problems in mapping. This is because the algorithm works best with well defined state descriptions (positions, velocities, for example), and for states where observation and time-propagation models are also well understood.

The Kalman Filtering process can be considered as a prediction-update formulation. The algorithm uses a predefined linear model of the system to predict the state at the next time step. The prediction and updates are combined using the Kalman gain which is computed to minimize the Mean Square Error (MSE) of the state estimate. Figure 1 illustrates the block diagram of KF cycle (Bar-Shalom & Fortmann, 1988), and for further details, refer to (Siciliano & Khatib, 2008).

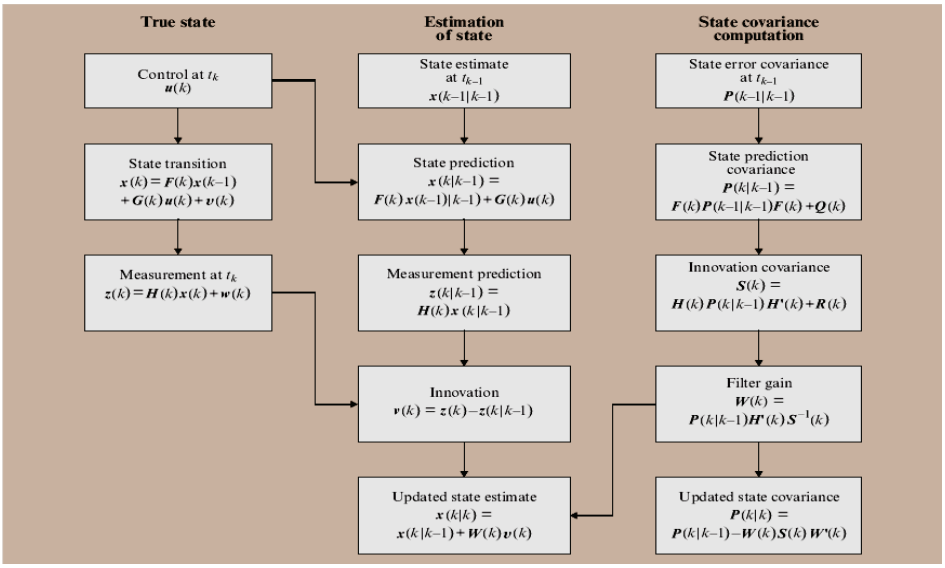


Fig. 1. Block diagram of the Kalman filter cycle (Bar-Shalom & Fortmann, 1988; Siciliano & Khatib, 2008)

The Extended Kalman Filter (EKF) is a version of the Kalman filter that can handle non-linear dynamics or non-linear measurement equations. Like the KF, it is assumed that the noises are all Gaussian, temporally uncorrelated and zero-mean with known variance. The EKF aims to minimise mean-squared error and therefore compute an approximation to the conditional mean. It is assumed therefore that an estimate of the state at time $k-1$ is available which is approximately equal to the conditional mean. The main stages in the derivation of

the EKF follow directly from those of the linear Kalman filter with the additional step that the process and observation models are linearised as a Taylor series about the estimate and prediction, respectively. The algorithm iterates in two update stages, measurement and time, see figure 2. Each positioning operation is generated once a new observation is assumed. Localization can be done from odometry or visual input changes. The complete algorithm is implemented for each landmark perception. In this sense, the processing time is saved by reducing covariance matrix function size per landmark. Detailed computations may be found in any number of books on the subject (Samperio & Hu, 2006).

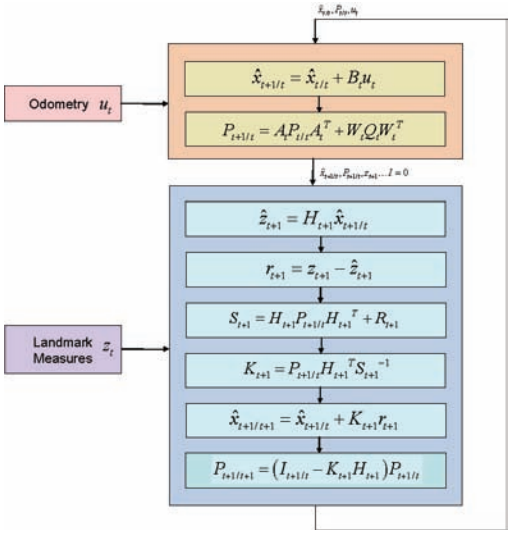


Fig. 2. Flowchart of Extended Kalman filter Algorithm (after Samperio & Hu, 2006)

Various approaches based on EKF have been developed. These approaches work well as long as the used information can be described by simple statistics well enough. The lack of relevant information is compensated by using models of various processes. However, such model-based approaches require assumptions about parameters which might be very difficult to determine (white Gaussian noise and initial uncertainty over Gaussian distribution). Assumptions that guarantee optimum convergence are often violated and, therefore, the process is not optimal or it can even converge. In fact, many approaches are based on fixed values of the measurement and state noise covariance matrices. However, such an information is not a priori available, especially if the trajectory of the robot is not elementary and if changes occur in the environment. Moreover, it has been demonstrated in the literature that how poor knowledge of noise statistics (noise covariance on state and measurement vectors) may seriously degrade the Kalman filter performance (Jetto, 1999). In the same manner, the filter initialization, the signal-to-noise ratio, the state and observation processes constitute critical parameters, which may affect the filtering quality. The stochastic Kalman filtering techniques were widely used in localization (Gao, 2002) (Chui, 1987) (Arras, 2001)(Borthwick, 1993) (Jensfelt, 2001) (Neira, 1999) (Perez, 1999) (Borges, 2003). Such approaches rely on approximative filtering, which requires ad doc tuning of stochastic

modelling parameters, such as covariance matrices, in order to deal with the model approximation errors and bias on the predicted pose. In order to compensate such error sources, local iterations (Kleeman, 1992), adaptive models (Jetto, 1999) and covariance intersection filtering (Julier, 1997) (Xu, 2001) have been proposed. An interesting approach solution was proposed in (Jetto, 1999), where observation of the pose corrections is used for updating of the covariance matrices. However, this approach seems to be vulnerable to significant geometric inconsistencies of the world models, since inconsistent information can influence the estimated covariance matrices.
In the literature, the localization problem is often formulated by using a single model, from both state and observation processes point of view. Such an approach, introduces inevitably modelling errors which degrade filtering performances, particularly, when signal-to-noise ratio is low and noise variances have been estimated poorly. Moreover, to optimize the observation process, it is important to characterize each external sensor not only from statistic parameters estimation perspective but also from robustness of observation process perspective. It is then interesting to introduce an adequate model for each observation area in order to reject unreliable readings. In the same manner, a wrong observation leads to a wrong estimation of the state vector and consequently degrades the performance of localization algorithm. Multiple-Model estimation has received a great deal of attention in recent years due to its distinctive power and great recent success in handling problems with both structural and parametric uncertainties and/or changes, and in decomposing a complex problem into simpler sub-problems, ranging from target tracking to process control (Blom, 1988) (Li, 2000) (Li, 1993) (Mazor, 1996).
This paper focuses on robust pose estimation for mobile robot localization. The main idea of the approach proposed here is to consider the localization process as a hybrid process which evolves according to a model among a set of models with jumps between these models according to a Markov chain (Djamaa & Amirat, 1999) (Djamaa, 2001). A close approach for multiple model filtering is proposed in (Oussalah, 2001). In our approach, models refer here to both state and observation processes. The data fusion algorithm which is proposed is inspired by the approach proposed in (Dufour, 1994). We generalized the latter for multi mode processes by introducing multi mode observations. We also introduced iterative and adaptive EKFs for estimating noise statistics. Compared to a single model-based approach, such an approach allows the reduction of modelling errors and variables, an optimal management of sensors and a better control of observations in adequacy with the probabilistic hypotheses associated to these observations. For this purpose and in order to improve the robustness of the localization process, an on line adaptive estimation approach of noise statistics (state and observation) proposed in (Jetto, 1999), is applied for each mode. The data fusion is performed by using Adaptive Linear Kalman Filters for linear processes and Adaptive EKF for nonlinear processes.

3. Localization system description

This paper deals with the problem of multi sensor filtering and data fusion for the robust localization of a mobile robot. In our present study, we consider an autonomous robot equipped with two telemeters placed perpendicularly, for absolute position measurements of the robot with respect to its environment, a gyroscope for measuring robot's orientation, two drive wheels and two separate encoder wheels attached with optical shaft encoders for

odometry measurements. The environment where the mobile robot moves is a rectangular room without obstacles, see figure 3.

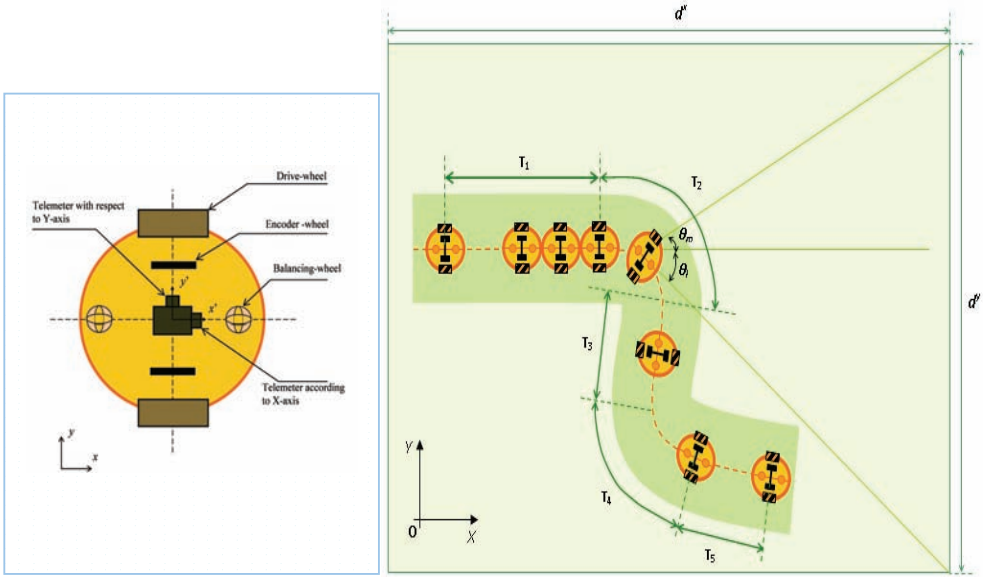


Fig. 3. Mobile robot description and its evolution in the environment with Nominal trajectory

The aim is not to develop a new method for environment reconstruction or modelling from data sensors; rather, the goal is to propose a new approach to improve existing data fusion and filtering techniques for robust localization of a mobile robot.

For an environment with a more complex shape, the observation model which has to be employed at a given time, will depend on the robot's situation (robot's trajectory, robot's pose with respect to its environment) and on the geometric or symbolic model of environment.

Initially, all significant information for localization is contained in a state space vector. The usefulness of an observer in a localization system evokes the modelling of variables that affect the entire behaviour system. The observer design problem relies on the estimation of all possible internal states in a linear system.

3.1 Odometric model

Let $X_e(k) = [x(k)\ \ y(k)\ \ \theta(k)]^T$ be the state vector at time k, describing the robot's pose with respect to the fixed coordinate system.

The kinematic model of the robot is described by the following equations:

$$x_{k+1} = x_k + l_k \cdot \cos(\theta_k + \Delta\theta_k / 2) \tag{1}$$

$$y_{k+1} = y_k + l_k \sin(\theta_k + \Delta\theta_k / 2) \tag{2}$$

$$\theta_{k+1} = \theta_k + \Delta\theta_k \tag{3}$$

with: $l_k = (l_k^r + l_k^l)/2$ and $\Delta\theta_k = (l_k^r - l_k^l)/d$. l_k^r and l_k^l are the elementary displacements of the right and the left wheels; d the distance between the two encoder wheels.

3.2 observation model of telemeters

As the environment is a rectangular room, the telemeters measurements correspond to the distances from the robot location to walls (Fig. 3.).

Then, the observation model of telemeters is described as follows:
for $0 \le \theta(k) < \theta^l$, according to *X*-axis:

$$d(k) = \left(d^x - x(k)\right) / \cos(\theta(k)) \tag{4}$$

for $\theta^l \le \theta(k) \le \theta^m$, according to *Y*-axis:

$$d(k) = \left(d^y - y(k)\right) / \sin(\theta(k)) \tag{5}$$

With d^x and d^y , respectively the length and the width of the experimental site; θ^l and θ^m , respectively the angular bounds of observation domain with respect to *X* and *Y* axes; $d(k)$ is the distance between the robot and the observed wall with respect to *X* or *Y* axes at time k .

3.3 observation model of gyroscope

By integrating the rotational velocity, the gyroscope model can be expressed by the following equation:

$$\theta_l(k) = \theta(k) \tag{6}$$

Each sensor described above is subject to random noise. For instance, the encoders introduce incremental errors (slippage), which particularly affect the estimation of the orientation. For a telemeter, let's note various sources of errors: geometric shape and surface roughness of the target, beam width. For a gyroscope, the sources of errors are: the bias drift, the nonlinearity in the scale factor and the gyro's susceptibility to changes in ambient temperature.

So, both odometric and observation models must integrate additional terms representing these noises. Models inaccuracies induce also noises which must be taken into account. It is well known that odometric model is subject to inaccuracies caused by factors such as: measured wheel diameters, unequal wheel-diameters, trajectory approximation of robot between two consecutive samples. These noises are usually assumed to be Zero-mean white Gaussian with known covariance. This hypothesis is discussed and reconsidered in the proposed approach.
Besides, an estimation error of orientation introduces an ambiguity in the telemeters measurements (one telemeter is assumed to measure along *X* axis while it is measuring along *Y* axis and vice-versa). This situation is particularly true when the orientation is near angular bounds θ^l and θ^m. This justifies the use of multiple models to reduce measuring errors and efficiently manage robot's sensors. For this purpose, we have introduced the concept of observation domain (boundary angles) as defined in equations (4) and (5).

4. Proposed approach for mobile robot localisation

As mentioned in (Touati et al., 2007), we present our data fusion and filtering approach for the localization of a mobile robot. In order to increase the robustness of the localization and as discussed in section 2, the localization process is decomposed into multiple models. Each model is associated with a mode and an interval of validity corresponding to the observation domain; the aim is to consider only reliable information by filtering erroneous information. The localization is then considered as a hybrid process. A Markov chain is employed for the prediction of each model according to the robot mode. The multiple model approach is best understandable in terms of stochastic hybrid systems. The state of a hybrid system consists of two parts: a continuously varying base-state component and a modal state component, also known as system mode, which may only jump among points, rather than vary continuously, in a (usually discrete) set. The base state components are the usual state variables in a conventional system. The system mode is a mathematical description of a certain behavior pattern or structure of the system. In our study, the mode corresponds to robot's orientation. In fact, the latter parameter governs the observation model of telemeters along with observation domain. Other parameters, like velocity or acceleration, could also be taken into account for mode's definition. Updating of mode's probability is carried out either from a given criterion or from given laws (probability or process). In this study, we assume that each Markovian jump (mode) is observable (Djamaa 2001) (Dufour, 1994). The mode is observable and measurable from the gyroscope.

4.1 Proposed filtering models

Let us consider a stochastic hybrid system. For a linear process, the state and observation processes are given by:

$$X_e(k/k-1,\alpha_k) = A(\alpha_k)\cdot X_e(k-1/k-1,\alpha_k) + B(k,\alpha_k)\cdot U(k-1,\alpha_k) + W(k,\alpha_k) \quad (7)$$

$$Y_e(k,\alpha_k) = C(\alpha_k)\cdot X_e(k/k-1,\alpha_k) + V(k,\alpha_k) \quad (8)$$

For a nonlinear process, the state and observation processes are described by:

$$X_e(k/k-1,\alpha_k) = F(X_e(k-1/k-1),\alpha_k,U(k-1)) + W(k,\alpha_k) \tag{9}$$

$$Y_e(k,\alpha_k) = G_e(X_e(k/k-1),\alpha_k) + V(k,\alpha_k) \tag{10}$$

where X_e, Y_e and U are the base state vector, noisy observation vector and input vector; α_k is the modal state or system mode at time k, which denotes the mode during the k^{th} sampling period; W and V are the mode-dependent state and measurement noise sequences, respectively.

The system mode sequence $\langle \alpha_k \rangle$ is assumed for simplicity to be a first-order homogeneous Markov chain with the transition probabilities, so for $\forall \alpha_i, \alpha_j \in S$:

$$P\{\alpha_{k+1}^j \mid \alpha_k^i\} = \pi_{ij} \tag{11}$$

Where α_k^j denotes that mode α_j is in effect at time k and S is the set of all possible system modes, called mode space.

The state and measurement noises are of Gaussian white type. In our approach, the state and measurement processes are assumed to be governed by the same Markov chain. However, it's possible to define differently a Markov chain for each process. The Markov chain transition matrix is stationary and well defined.

4.2 Statistics parameters estimation

It is well known that how poor estimates of noise statistics may lead to the divergence of Kalman filter and degrade its performance. To prevent this divergence, we propose an adaptive algorithm for the adjustment of the state and measurement noise covariance matrices.

a. Measurement noise variance

Let $R = \left(\sigma_{v,i}^2(k)\right)(i = 1:n_0)$, be the measurement noise variance at time k for each component of the observation vector. Parameter n_0 denotes the number of observers (sensors number).

Let $\hat{\beta}(k)$ the squared mean error for stable measurement noise variance, and $\gamma(k)$ the innovation, thus:

$$\hat{\beta}(k) = \frac{1}{n}\sum_{j=0}^{n} \gamma_i^2(k-1) \tag{12}$$

For $n+1$ samples, the variance of $\hat{\beta}(k)$ can be written as:

$$E\left(\hat{\beta}(k)\right) = \frac{1}{n+1}\sum_{j=0}^{n}\begin{pmatrix} C_i(k-j)\cdot P(k-j,k-j-1)\cdot \\ C_i(k-j)^T + \sigma_{v,i}^2 \end{pmatrix} \tag{13}$$

Then, we obtain the estimation of the measurement noise variance:

$$\hat{\sigma}_{v,i}^2 = \max\left\{\frac{1}{n}\sum_{j=0}^{n}\begin{pmatrix} \gamma_i^2(k-j) - \frac{n}{n+1}\cdot C_i(k-j)\cdot \\ P(k-j,k-j-1)\cdot C_i(k-j)^T \end{pmatrix}, 0\right\} \tag{14}$$

The restriction with respect to zero is related to the notion of variance. A recursive formulation of the previous estimation can be written:

$$\hat{\sigma}_{v,i}^2(k) = \max\left\{\hat{\sigma}_{v,i}^2(k-1) + \frac{1}{n}\cdot\begin{pmatrix} \gamma_i^2(k) \\ -\gamma_i^2(k-(n+1)) \\ -\frac{n}{n+1}\cdot\Psi \end{pmatrix}, 0\right\} \tag{15}$$

where:

$$\begin{aligned}\Psi = {} & C_i(k)\cdot P(k,k-1)\cdot C_i(k)^T - C_i(k-(n+1))\cdot \\ & P(k-(n+1),k-(n+1)-1)\cdot C_i(k-(n+1))^T\end{aligned} \tag{16}$$

b. State noise variance

To estimate the state noise variance, we employ the same principle as in subsection a. One can write:

$$\hat{Q}_e(k) = \hat{\sigma}_{n,i}^2(k)\cdot Q_d \tag{17}$$

By assuming that noises on the two encoder wheels measurements obey to the same law and have the same variance, the estimation of state noise variance can be written:

$$\hat{\sigma}_{n,i}^2(k) = \max\left\{\begin{array}{l} \gamma_i^2(k-1) - C_i(k+1)\cdot P(k+1,k)\cdot \\ \dfrac{C_i(k+1)^T - \hat{\sigma}_{v,i}^2(k+1)}{C_i(k+1)\cdot Q_d \cdot C_i(k+1)^T}, \\ 0 \end{array}\right\} \tag{18}$$

with:

$$\hat{Q}_d(k) = B(k)\cdot B(k)^T \tag{19}$$

By replacing the measurement noise variance by its estimate, we obtain a mean value given by the following equation:

$$\hat{\sigma}_n^2(k) = \max\left\{\frac{1}{(m+1)\cdot n_0}\sum_{j=1}^{m}\sum_{i=1}^{n_0}\hat{\sigma}_{n,i}^2(k-j), 0\right\} \tag{20}$$

Where, the parameter m represents the sample number.

The algorithm proposed above carries out an on line estimation of state and measurement noise variances. Parameters n and m are chosen according to the number of samples used at time k. The noises variances are initialized from an "*a priori*" information and then updated on line. In our approach, variances are updated according the robot's mode and the measurement models.

For an efficient estimation of noise variances, an ad hoc technique consisting in a measure selection is employed. This technique consists of filtering unreliable readings by excluding readings with weak probability like the appearance of fast fluctuations. For instance, in the case of Gaussian distribution, we know that about 95% of the data are concentrated in the interval of confidence $[m-2\sigma, m+2\sigma]$ where m represents the mean value and σ the variance.

The sequence in which the filtering of the state vector components is carried out is important. Once the step of filtering completed, the probabilities of each mode are updated from the observers (sensors). One can note that the proposed approach is close, on one hand, to the Bayesian filter by the extrapolation of the state probabilities, and on the other hand to the filter with specific observation of the mode.

5. Implementation and results

The proposed approach for robust localization was applied for the mobile robot described in section 2. The nominal trajectory of the mobile robot includes three sub trajectories T1, T2 and T3, defining respectively a displacement along X axis, a curve and a displacement along Y axis, see figure 4.

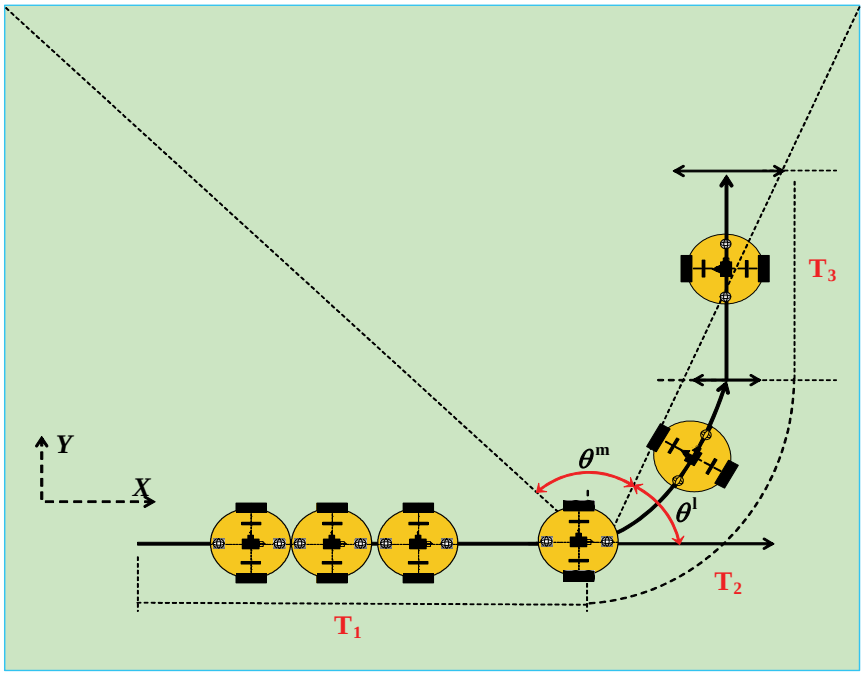


Fig. 4. Mobile robot in moving in the environment with Nominal trajectory T1, T2 and T3.

Note that the proposed approach remains valid for any type of trajectory (any trajectory can be approximated by a set of linear and circular sub trajectories). For our study, we have considered three models. This number can be modified according to the environment's structure, the type of trajectory (robot rotating around itself, forward or backward displacement, etc.) and to the number of observers (sensors). Notice that the number of models (observation and state) has no impact on the validity of the proposed approach.

To demonstrate the validity of our proposed Adaptive Multiple-Model approach and to show its effectiveness, we've compared it to the following standard approaches: Single-Model based EKF without estimation variance, single-model based IEKF. For sub trajectories T1 and T3, filtering and data fusion are carried out by iterative linear Kalman filters due to linearity of the models, and for sub trajectory T2, by iterative and extended Kalman filters.

The observation selection technique is applied for each observer before the filtering step in order to control, on one side, the estimation errors of variances, and on the other side, after each iteration, to update the state noise variance. If an unreliable reading is rejected at a given filtering iteration, this has for origin either a bad estimation of the next component of the state vector and of the prediction of the corresponding observation, or a bad updating of the corresponding state noise variance. The iterative filtering is optimal when it is carried out for each observer and no reading is rejected. In the implementation of the proposed approach, the state noise variance is updated, for a given mode i , is carried out according to the following filtering sequence: x, y and then θ .

Firstly, let's consider the set of the following notations, table 1:

$(\varepsilon x, \varepsilon y, \varepsilon\theta)$	Estimation errors corresponding to (x, y, θ)
$(Ndx, Ndy, Nd\theta)$	Percentage of selected data for filtering corresponding to (x, y, θ)
$(Ndxe, Ndye, Nd\theta e)$	Percentage of selected data for estimation of the variances of state and measurement noises, corresponding to (x, y, θ)
SM (-+)	Single-Model based EKF
SMI (°)	single-model based IEKF
AMM (--)	Adaptive Multiple-Model

Table 1. Set of notations

Several scenarios have been studied according to the variation of statistics parameters, i.e., sensors signal-to-noise ratio, initial state variance, noise statistics (measurement and state variances). Simulations were carried out to analyze the performances of each approach in various scenarios. Thus, in scenarios 1 and 2, we show the influence of measurement and state noises variances estimation on the quality of localization. In scenario 3, it will concern the sensors signal-to-noise ratio.

Scenario 1:
-Noise-to-signal Ratio of odometric sensors: right encoder: 4%, left encoder: 4%
-Noise-to-signal Ratio of Gyroscope: 1%
-Noise-to-signal Ratio of telemeters: 2% of the odometric elementary step
-"A priori" knowledge on the variance in initial state: Good
-"A priori" knowledge on measurement noise variances: Good
-State and measurement noise variances estimation: 10 times real average variances of encoders

This scenario is characterized by weak state and measurement noises and by high initial value of state noise variance. One can note that although a bad initialization (10 times the average variance), the AMM approach presents better performances for estimation of the 3 components of state vector (Tables 2-4, figures 5-11). On section T1, (figure 12), the estimated variance remains constant compared to the *a priori* average variance (10 times the average variance) corresponding to the initial state. Indeed, the algorithm of estimation of variances does not show any evolution because of the high value of variance in the initial state. However, for section T2 and T3, the variance decreases by half compared to the initial variance, and approaches the actual average variance.

	T1			T2			T3		
	SM	**SMI**	**AMM**	**SM**	**SMI**	**AMM**	**SM**	**SMI**	**AMM**
εx (cm)	3.46	6.12	0.64	8.3	6.15	9.6	4.76	3.38	0.72
εy (cm)	4.58	3.69	0.5	12.3	7.3	9.7	4.64	3.58	1.82
$\varepsilon\theta$ (10^{-3} rad)	22.7	30.5	2.7	8.2	11.9	9.7	21.6	29.3	7.9

Table 2. Average estimation errors

Ndx	*Ndy*	*Ndθ*	*Ndxe*	*Ndye*	*Ndθe*
98.75%	90%	97.5%	98.75%	98.75%	97.5%

Table 3. Selected data percentage

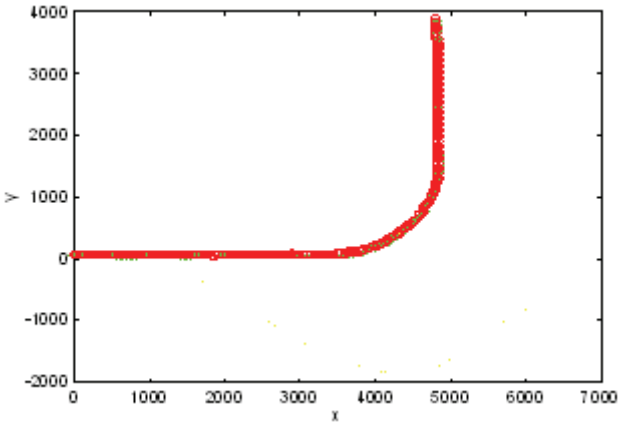


Fig. 5. Estimated trajectories by Encoders and, SM, SMI and AMM Filters

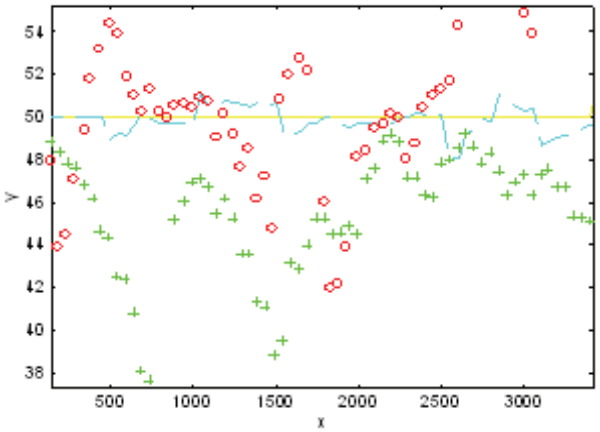


Fig. 6. Estimated trajectories (sub trajectory T1)

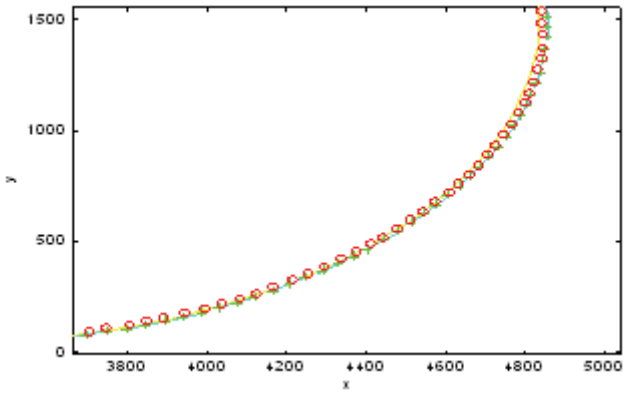


Fig. 7. Estimated trajectories (sub trajectory T2)

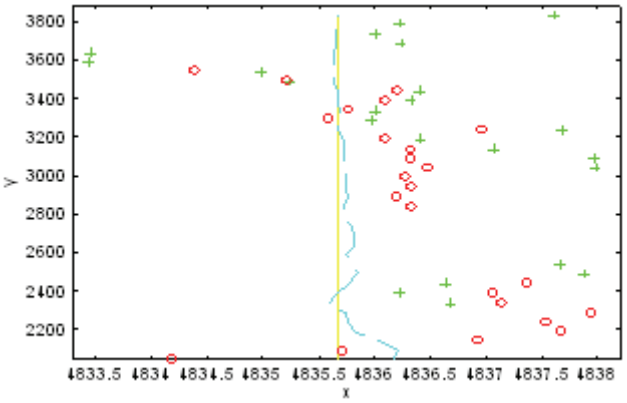


Fig. 8. Estimated trajectories (sub trajectory T3)

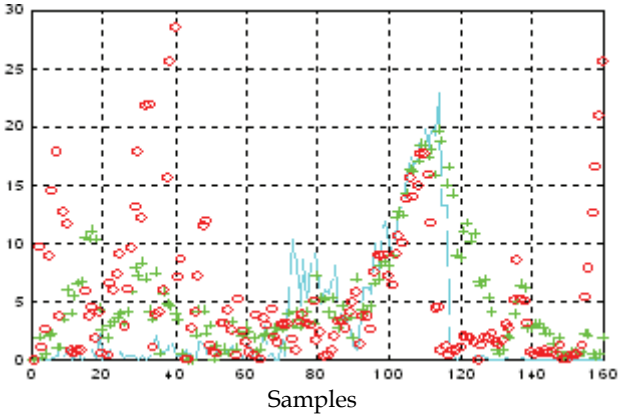


Fig. 9. Position error with respect to X axis

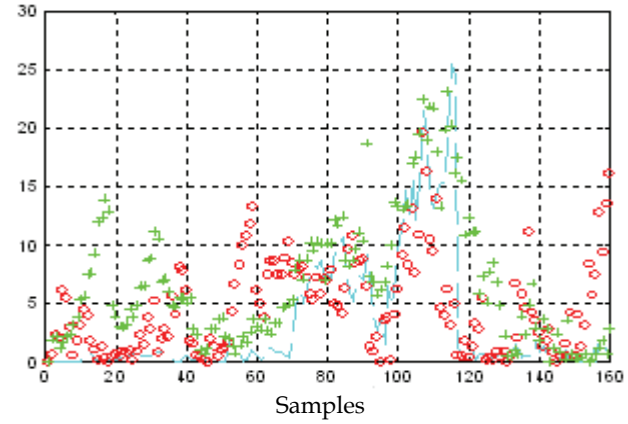


Fig. 10. Position error with respect to Y axis

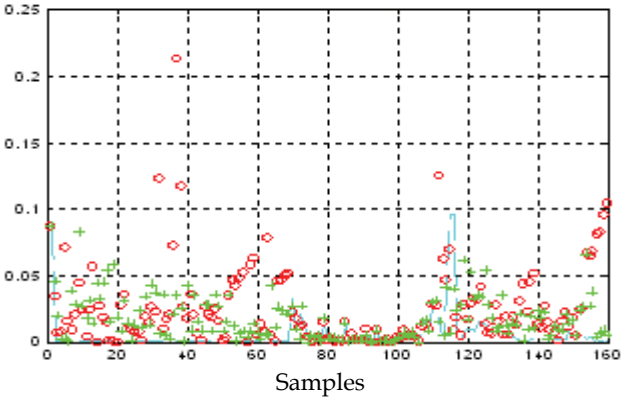


Fig. 11. Absolute error on orientation angle

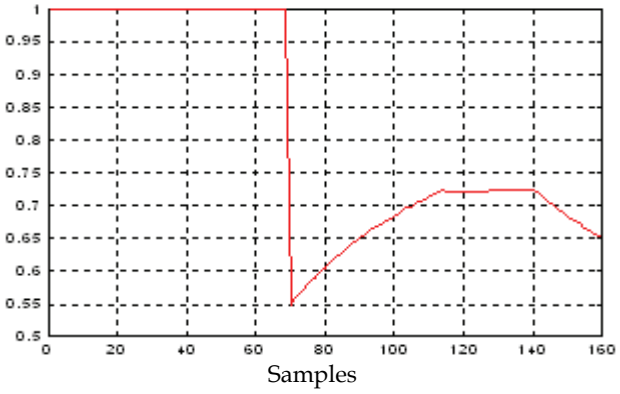


Fig. 12. Ratio between the estimate of state noise variance and the average variance

Scenario 2:
-Noise-to-signal Ratio of odometric sensors: right encoder: 10%, left encoder: 10%
-Noise-to-signal Ratio of Gyroscope: 3%
-Noise-to-signal Ratio of telemeters: 4% of the odometric elementary step (40% of the state noise)
-"A priori" knowledge on the variance in initial state: Good
-"A priori" knowledge on noise variances (i) telemeters and state: Good; (ii) gyroscope: Bad

The results presented here (Tables 4-5 and figures 13-20) show the influence of signal-to-noise ratio and estimation of noise variances on performances of SM and SMI filters. In this scenario, the initial variance of measurement noise of the gyroscope is incorrectly estimated. Unlike AMM approach, filters SM and SMI do not carry out any adaptation of this variance, leading to unsatisfactory performance.

	T1			**T2**			**T3**		
	SM	**SMI**	**AMM**	**SM**	**SMI**	**AMM**	**SM**	**SMI**	**AMM**
εx (cm)	11.7	11	1.8	19	75	13.6	17.3	40	1.3
εy (cm)	16.7	21	1	39	179	17.4	15.7	117	1.93
$\varepsilon\theta$ (10^{-3} rad)	99.3	129	1.5	42.9	175	35.4	97.5	167	37.8

Table 4. Average estimation errors

Ndx	*Ndy*	*Ndθ*	*Ndxe*	*Ndye*	*Ndθe*
87.5%	66%	99.37%	87.5%	82.5%	99.37%

Table 5. Selected data percentage

Figure 20 illustrates the evolution of state noise variance estimate compared to the average variance. Note that the ratio between variances reaches 1.7 on sub trajectory T1, 3.0 on sub trajectory T2, and 3.3 on sub trajectory T3. It is important to mention that the algorithm proposed for estimation of variances estimates the actual value of state noise variance and not its average value. These results are related to the fact that the signal-to-noise ratio is weak both for the odometer and the telemeters.

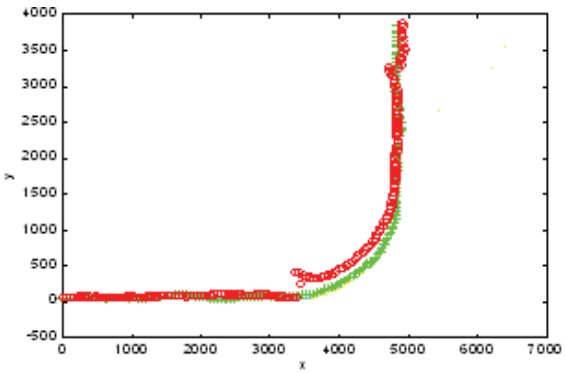


Fig. 13. Estimated trajectories by Encoders and, SM, SMI and AMM Filters

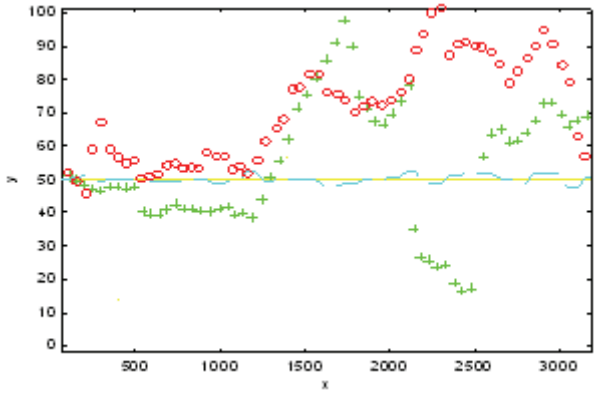


Fig. 14. Estimated trajectories (sub trajectory T1)

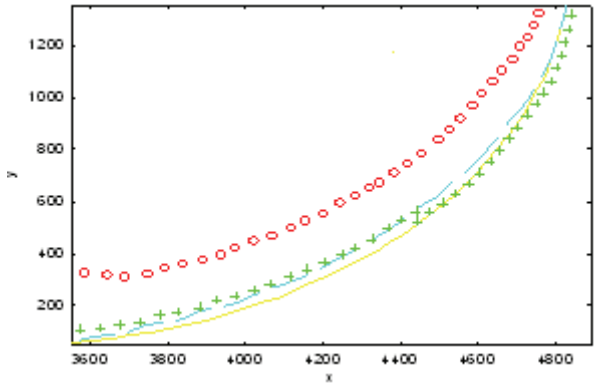


Fig. 15. Estimated trajectories (sub trajectory T2)

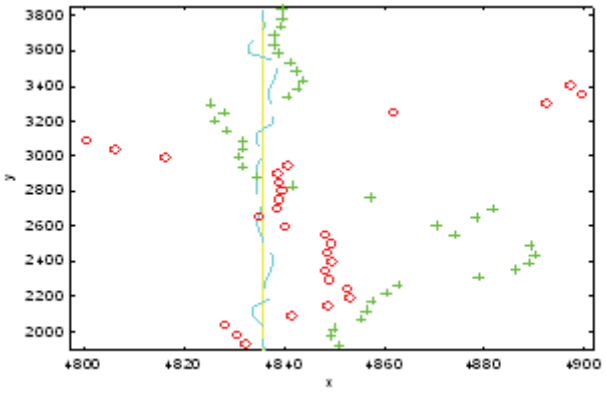


Fig. 16. Estimated trajectories (sub trajectory T3)

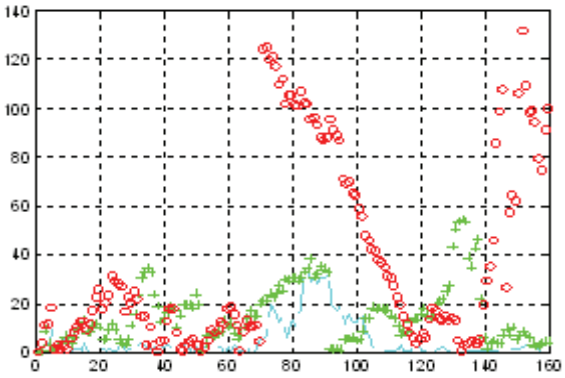


Fig. 17. Position error with respect to X axis

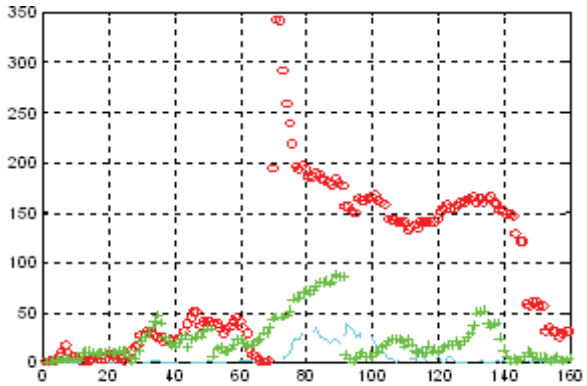


Fig. 18. Position error with respect to Y axis

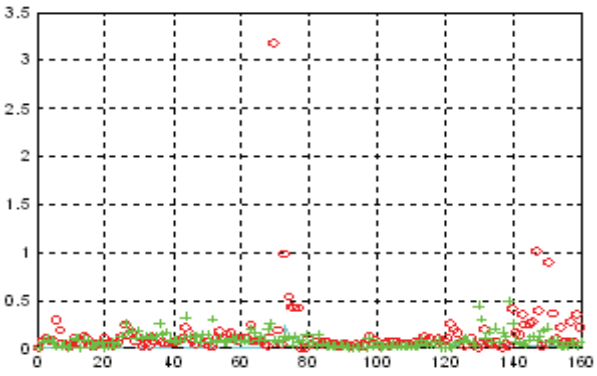


Fig. 19. Absolute error on orientation angle

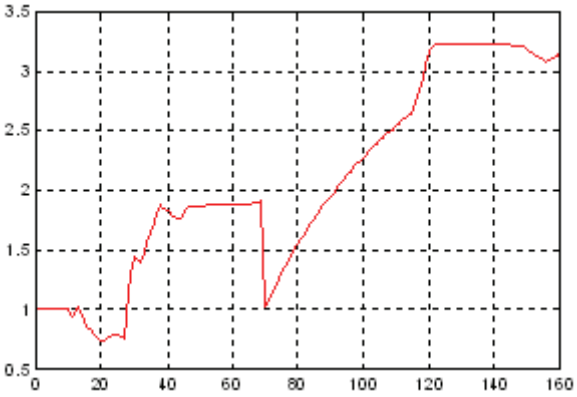


Fig. 20. Ratio between the estimate of state noise variance and the average variance

Scenario 3:
-Noise-to-signal Ratio of odometric sensors: right encoder: 8%, left encoder: 8%
-Noise-to-signal Ratio of Gyroscope: 3%
-Noise-to-signal Ratio of telemeter 1: 10% of the odometric elementary step
-Noise-to-signal Ratio of telemeter 2: 10% the odometric elementary step
-"A priori" knowledge on the variance in initial state: Good
-"A priori" knowledge on noise statistics (measurement and state variances): Good

In this scenario, the telemeters measurement noise is higher than state noise. We remark that performances of AMM filter are better that those of SM and SMI filters concerning x and y components (tables 6-7; figures 21-28). In sub trajectory T3, the orientation's estimation error relating to AMM filter (Table 6) has no influence on filtering quality of the remaining components of state vector. Besides, one can note that this error decreases in this sub trajectory, see figure 27. In this case, only one gyroscope is used for the prediction and updating the Markov chain probabilities. In sub trajectory T2, we remark that the estimation error along x-Axis for AMM filter is lightly higher than those relating to other filters. This error is concentrated on first half of T2 sub trajectory (figure 25) and decreases then on second half of the trajectory. This can be explained by the fact that on one hand, the estimation variances algorithm rejected 0.7% of data, and on the other hand, the filtering step has rejected the same percentage of data. This justifies that neither the variances updating, nor the x-coordinate correction, were carried out (figure 28).
Note that unlike filters SM and SMI, filter AMM has a robust behavior concerning pose estimation even when the signal-to-noise ratio is weak. By introducing the concept of observation domain for observation models, we obtain a better modeling of observation and a better management of robot's sensors. The last remark is related to the bad performances of filters SM and SMI when the signal-to-noise ratio is weak. This ratio degrades the estimation of the orientation angle, observation matrices, Kalman filter gain along with the prediction of the observations.

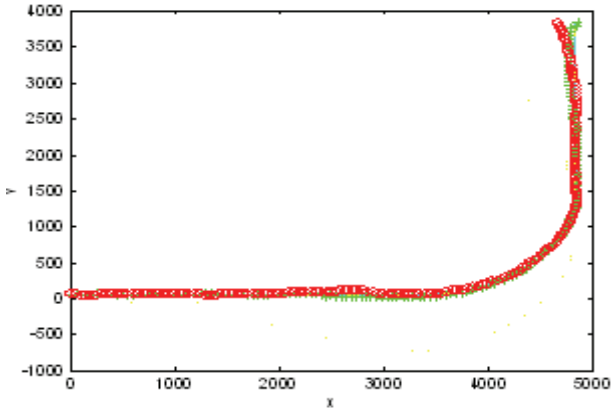


Fig. 21. Estimated trajectories by Encoders and, SM, SMI and AMM Filters

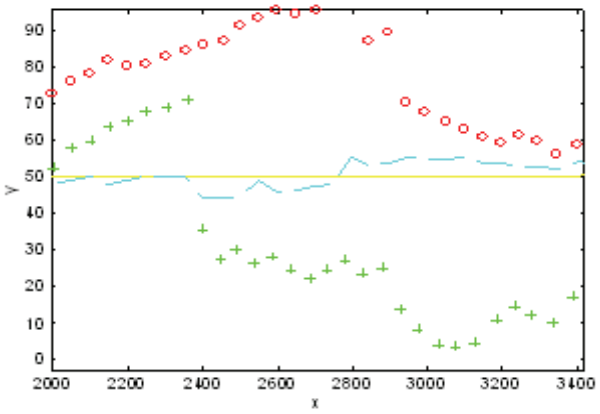


Fig. 22. Estimated trajectories (sub trajectory T1)

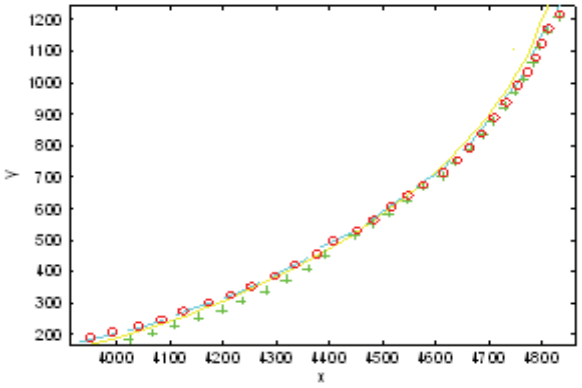


Fig. 23. Estimated trajectories (sub trajectory T2)

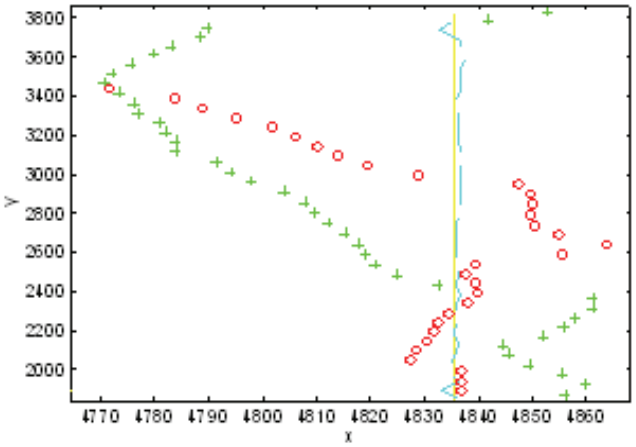


Fig. 24. Estimated trajectories (sub trajectory T3)

	T1			T2			T3		
	SM	**SMI**	**AMM**	**SM**	**SMI**	**AMM**	**SM**	**SMI**	**AMM**
εx (cm)	6.25	3.23	2.5	13.2	10.8	15.3	31.9	31.2	1.2
εy (cm)	13.6	16.7	2.3	23.9	11.9	8.25	19.2	5.75	3.23
$\varepsilon\theta$ (10^{-3} rad)	81.1	66.9	3.8	32.2	39.9	35.6	136	125	267.9

Table 6. Average estimation errors (Scenario 1)

Ndx	*Ndy*	*Ndθ*	*Ndxe*	*Ndye*	*Ndθe*
99.37%	84.37%	99.37%	99.37%	97.5%	99.37%

Table 7. Selected data percentage

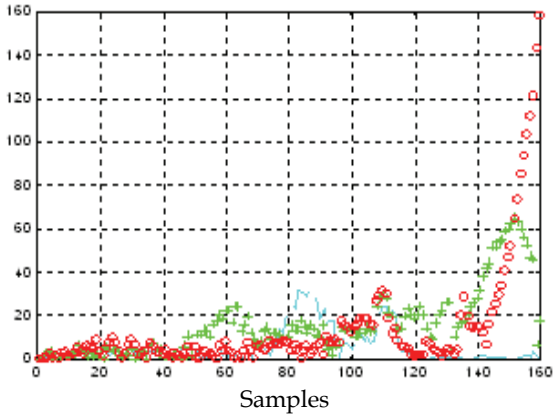


Fig. 25. Position error with respect to X axis

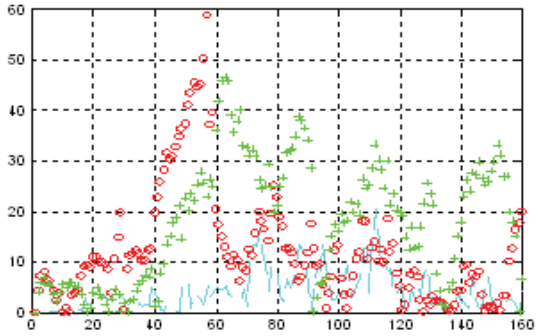


Fig. 26. Position error with respect to Y axis

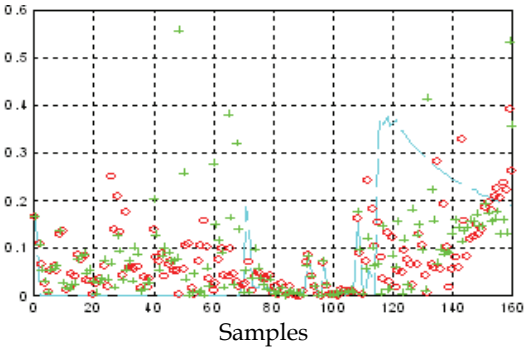


Fig. 27. Absolute error on orientation angle

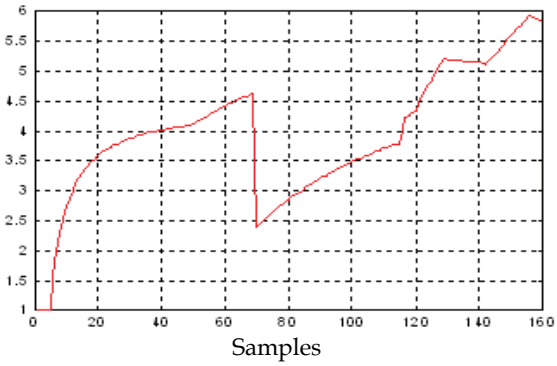


Fig. 28. Ratio between the estimate of state noise variance and the average variance

6. Conclusion

This research work introduces a multiple model approach for the robust localization of a mobile robot. The localization method is considered as a hybrid process, which is

decomposed into multiple models. Each model is associated with a mode and an interval of validity corresponding to the observation domain. A Markov chain is employed for the prediction of each model according to the robot mode. To prevent divergence of standard Kalman Filtering and to increase its robustness, we proposed an adaptive algorithm for the adjustment of the state and measurement noise covariance matrices. For an efficient estimation of noise variances, we introduced an ad hoc technique consisting in a measure selection for filtering unreliable readings. The simulation results we obtain in different scenarios show better performances of the proposed approach compared to standard existing filters. Some future research need to be conducted to complete the proposed approach and particularly in probabilistic data fusion through sequential Monte Carlo methods, or through the use of functional density estimates. These investigations into utilizing multiple model technique for robust localization show promise and demand continuing research. Fuzzy logic theory can also be considered to increase robustness of the proposed localization algorithm.

7. References

Arras, K.O.; Tomatis, N.; Jensen, B.T.; Siegwart, R. (2001). Multisensor on-the-fly localization: precision and reliability for applications, *Robotics and Autonomous Systems*, 34, 131–143

Bar-Shalom, Y. (1990). Multi-target multi-sensor tracking (Artec House, Norwood 1990)

Bar-Shalom, Y. & Fortmann, T.E. (1988). Tracking and data association, (Academic, New York 1988)

Blom, H. A.P. & Bar-Shalom, Y. (1998). The interacting multiple model algorithm for systems with Markovian switching coefficients, *IEEE Transactions Automation and Control*, Vol. 33, pp. 780–783

Borenstein, J.; Everett, B. & Feng, L. (1996). Navigating mobile robots: systems and techniques, A.K. Peters, Ltd., Wellesley, MA

Borges, G.A. & Aldon, M.J. (2003). Robustified estimation algorithms for mobile robot localization based geometrical environment maps, *Robotics and Autonomous Systems,* 45 (2003) 131-159

Borthwick, S.; Stevens, M. & Durrant-Whyte, H. (1993). Position estimation and tracking using optical range data, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2172–2177

Chong, C.Y. & Kumar, S. (2003). Sensor networks: evolution, opportunities and challenges, *Proceeding of the IEEE*, Vol.91, No. 8, pp. 1247-1256

Chui, C. & Chen, G. (1987). Kalman filtering with real time applications, *Springer Series in Information Sciences, Springer-Verlag*, New-York 17 23-24

Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H. & Csorba, M. (2001). A Solution to the simultaneous localization and map building (SLAM) problem, *IEEE Transactions on Robotics and Automation*, Vol.17, No.3, pp. 229-241

Djamaa, Z. & Amirat, Y. (1999). Multi-model approach for the localization of mobile robots by multisensor fusion, *Proceedings of the 32th International Symposium On Automotive Technology and Automation*, pp. 247-260, Vienna, Austria

Djamaa, Z. (2001)., Approche multi modèle à sauts Markoviens et fusion multi capteurs pour la localisation d'un robot mobile. *PhD Thesis*, Paris XII University, France

Dufour, F. (1994). Contribution à l'étude des systèmes linéaire à saut markoviens, *PhD Thesis*, University of Paris Sud University, France

Estrada, C.; Neira, J. & Tardós, J.D. (2005). Hierarchical SLAM: real-Time accurate mapping of large environments, *IEEE Transactions on Robotics*, Vol. 21, No. 4, pp. 588-596

Gao, J.B. & Harris, C.J. (2002). Some remarks on Kalman filters for the multi-sensor fusion, *Journal of Information Fusion*, 3 191-201

Gning, A. & Bonnifait, P. (2005). Dynamic vehicle localization using constraints Propagation techniques on intervals. A comparison with Kalman Filtering. Proceedings of the International Conference on Robotics and Automation, pp. 4144- 4149, ISBN: 0-7803-8914-X, Barcelona, Spain

Harris, C.; Bailley, A. & Dodd, T. (1998). Multi-sensor data fusion in defense and aerospace, *Journal of Royal Aerospace Society*, 162 (1015) 229-244

Jensfelt, P. & Christensen, H.I.(2001). Pose tracking using laser scanning and minimalistic environment models, *IEEE Transactions on Robotics and Automation*, 17 (2) 138–147

Jetto, L.; Longhi, S. & Venturini, G. (1999) Development and experimental validation of an adaptive Kalman filter for the localization of mobile robots, *IEEE Transactions on Robotics and Automation*, 15 (2) 219–229

Julier, S.J. & Uhlmann, J.K. (1997) A non-divergent estimation algorithm in the presence of unknown correlations, *Proceedings of the American Control Conference*

Kanda, T.; Shiomi, M.; Perrin, L.; Nomura, T.; Ishiguro, H. & Hagita, N. (2007). Analysis of people trajectories with ubiquitous sensors in a science museum, *IEEE International Conference on Robotics and Automation*, pp.4846-4853, Roma, Italy

Kleeman, L. (1992). Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2582–2587

Kuipers, B. & Byun, Y.T. (2001). A robot exploration and mapping strategy based on a semantic hierarchy of Spatial Representations, *Robotics and Autonomous Systems*, Vol. 8, pp. 47-63

Li, X. R. (2000). Engineer's guide to variable-structure multiple-model estimation for tracking, in Multitarget-Multisensor Tracking: Applications and Advances, *Y. Bar-Shalom and D.W. Blair,* Eds. Boston, MA: Artech House, Vol. 3, Chap. 10, pp. 499–567

Li, X. R. (1996). Hybrid estimation techniques, control and dynamic systems: *Advances in Theory and Applications, C. T. Leondes*, Ed. New York: Academic, Vol. 76, pp. 213–287

Letchner, J.; Fox, D. & LaMarce, A. (2005). Large-scale localization from wireless signal strength, *Proceedings of the National Conference on Artificial Intelligence* (AAAI-05)

Liau, W. H.; Wu, C. L. & Fu, L. C. (2008). Inhabitants tracking system in a cluttered home environment via floor load sensors, *IEEE Transactions on Automation Science and Engineering*, Vol.5, No. 1, pp. 10-20

Liao, L.; Fox, D. & Kautz, H. (2005). Location-based activity recognition using relational Markov networks, *International Joint Conference on Artificial Intelligence* (IJCAI-05)

Madhavapeddy, A. & Tse, A. (2005). A study of Bluetooth propagation using accurate indoor location mapping, *International Conference Ubiquitous Computing* (Ubicomp2005), pp. 105-122

Mao, G.; Fidan, B. & Anderson, B.D.O. (2007). Wireless sensor network localization techniques, *International Journal of Computer and Telecommunications Networking*, Vol. 51, No. 10, pp. 2529-2553

Mazor, E.; Averbuch, A.; Bar-Shalom, Y. & Dayan, J. (1996). Interacting multiple model methods in target tracking: A survey, *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 34, No. 1, pp. 103–123

Maybeck, P. S. (1979). Stochastic models, Estimaton and Control, Vol. 1 (Academic, New York 1979)

Montemerlo, M.; Thrun, S.; Koller, D. & Wegbreit., B. (2002). Fast-SLAM. A factored solution to the simultaneous localization and mapping problem. Proceedings of the National Conference on Artificial Intelligence AAAI, pp. 593-598, ISBN:0-262-51129-0, Canada

Neira, J.; Tardós, J. D.; Horn, J. & Schmidt, G. (1999). Fusing range and intensity images for mobile robot localization, *IEEE Transactions on Robotics and Automation*, 15 (1) 76–84

Oussalah, M. (2001). Suboptimal multiple model filter for mobile robot localization, *International Journal of Robotics Research*, Vol. 20, No. 12, pp. 977-989

Pérez, J.A.; Castellanos, J.A.; Montiel, J.M.M.; Neira, J. & Tardós, J.D. (1999). Continuous mobile robot localization: vision vs. laser, *Proceedings of the IEEE International Conference on Robotics and Automation,* pp. 2917–2923

Ranganathan, A.; E. Menegatti, E. & Dellaert, F. (2006). Bayesian inference in the space of topological Maps", *IEEE Transactions on Robotics*, pp. 92-107

Ristic, B. & Smets, P. (2004). Kalman filters for tracking and classification and the transferable belief model. IF04-0046, FUSION04. Stockholm, Sweden

Samperio, R. & Hu, H. (2006). Kalman Filter localization for AIBO walking robots, *5th International Symosium on Robotics and Automation*, Mexico, August 25-28

Savelli, F. & Kuipers, B. (2004). Loop-closing and planarity in topological map building, *International Robotics and Systems*, Vol. 2, pp. 1511-1517

Siciliano, B. & Khatib, O. (2008). Multisensor data fusion, In: *Springer Handbook of Robotics,* Springer Berlin Heidelber (Ed.), 1-26, ISBN 978-3-540-23957-4, Berlin, Germany

Sim, R.; Elinas, P.; Griffin, M.; Shyr, A. & Little, J. J. (2006). Design and analysis of a framework for real-time vision-based SLAM using Rao-Blackwellised particle filters, *Canadian Conference on Computer and Robot Vision*

Tardos, J.D.; Neira, J.; Newman, P.M.& Leonard, J.J. (2002). Robust mapping and localization in indoor environments Using Sonar Data", *International Journal of Robotics Research*, Vol. 21, pp. 311-330

Thrun, S.; Liu, Y.; Koller, D.; Ng, A. Y.; Ghahramani, Z. & Durrant-Whyte, H. (2004). Simultaneous Localization and Mapping with Sparse Extended Information Filters. *International Journal of Robotics Research*, Vol.23, No. 7-8, pp.693-716

Touati, Y.; Amirat, Y. & Ali Chérif, A. (2007). Multiple model adaptive extended Kalman filter for the robust localization of a mobile robot, *4th International Conference on Informatics in Control, Automation and Robotics*, Vol. 2, pp.446-454, Angers, France

Xu, X. & Negahdaripour, S. (2001). Application of extended covariance intersection principle for mosaic-based optical positioning and navigation of underwater vehicles, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2759–2766

16

A Semantic Inference Method of Unknown Words using Thesaurus based on an Association Mechanism

Seiji Tsuchiya*, Hirokazu Watabe**, Tsukasa Kawaoka**, Fuji Ren*
*The University of Tokushima, **Doshisha University
*, **Japan

1. Introduction

In recent years, a computer technology has advanced and a robot designed for home use is actively being developed. Home-entertainment robot like "AIBO"(Sony) and humanoid robot like "ASIMO"(Honda) which can walk on two feet are examples of such robot. It is not an exaggeration to say that hardware technology has advanced enough to fit for practical use at home.

Currently the field of artificial intelligence is expected to be developed as a software technology. For a robot to coexist with human, it should be equipped the ability to feel, think, talk or behave just like a person.

Therefore, we are engaged in research aiming to develop a robot which can smoothly make conversation with human beings. Such robot needs to have abilities to understand and interpret words. Currently, a technique based on a large-scale language dictionary or a corpus is predominantly used in the field of the language processing. Only one wrong response from a robot gives human beings very unfavourable impression during the conversation. This might become one of the fatal causes for human not to accept a robot.

Therefore, to achieve our purpose, a very large-scale language dictionary and a corpus are needed. As quite a lot of costs, resources and time are necessary to create such linguistic capital, automatic construction technique is also being researched. However, the knowledge in a category of common sense is inherent to human and difficult to construct automatically although it is indispensable knowledge for robot to realize conversation with human beings without sense of unease.

In this paper, we propose a technique which contributes to semiautomatic construction of a large-scale language dictionary and a corpus. Concretely, a system using the proposed technique indicates a position of an unknown word to be registered in an existing thesaurus dictionary. We show the effectiveness by comparing traditional techniques with the proposal technique. In addition, we evaluate how performance of the proposal technique approaches performance of human.

This system presents answer candidates so that a time and effort required for making a large-scale language dictionary and a corpus can be reduced.

2. Traditional Techniques

In this chapter, we explain techniques based on the vector space model and the statistical model. Then, these techniques and the proposed technique in this paper are compared, and the accuracy of an unknown word registration processing is evaluated.

2.1 Technique based on the Vector Space Model

In a technique based on vector space model, a similarity is calculated by using the cosine between a feature vector of each node in a thesaurus and a feature vector of an unknown word(Uramoto 1996). Then, the unknown word is registered in a node with a high similarity computation. In a simplest vector space model, a feature vector consists of a co-occurrence frequency of a noun and a verb. Each element of a feature vector at a node is calculated by adding co-occurrence frequencies of a verb and a noun at a node. Moreover, each element of a feature vector of an unknown word is a co-occurrence frequency of the unknown word and a verb.

$node_i$ shows a node of a thesaurus and $NODE$, $NODE = \{ node_1, node_2, \cdots, node_{|NODE|} \}$ is a node set with limited number of elements and $|\cdot|$ shows the number of the elements in a set. Moreover, (w, z), $w \in NODE, z \in VERB$ is a binomial class indicating one training data and means that node w and verb z are co-occurring. $(w, z)^N$ indicates series consisting of a training data of N pieces. $unknown$ meaning an unknown word and $(unknown, y^M)$ shows a binomial class of unknown word $unknown$ and series y^M of verb y co-occurring with unknown word $unknown$. When we define as above mentioned, in a technique based on vector space model, a node to register an unknown word is decided as follows:

$$d_{\cos}\left((w, z)^N, (unknown, y^M)\right) = \arg\max_{node_i} \cos(\, vec(node_i), vec(unknown)\,)$$

$$= \arg\max_{node_i} \frac{vec(node_i) \cdot vec(unknown)}{\| vec(node_i) \| \| vec(unknown) \|}$$

However,

$$vec(node_i) = \left(co\left((node_i, verb_1) \mid (w, z)^N\right),\ co\left((node_i, verb_2) \mid (w, z)^N\right), \cdots, co\left((node_i, verb_{|VERB|}) \mid (w, z)^N\right)\right)$$

$$vec(unknown) = \left(co\left(verb_1 \mid y^M\right), co\left(verb_2 \mid y^M\right), \cdots, co\left(verb_{|VERB|} \mid y^M\right)\right)$$

Here, $d_{\cos}\left((w, z)^N, (unknown, y^M)\right)$ is a function to decide a node where an unknown word should be registered. $vec(A)$ shows a feature vector of A and $\cos(B \mid C)$ shows the number of B in C. Moreover, $\cos$ is a function which calculates a value of the cosine between vectors, $vec_A \cdot vec_B$ is inner product between vector vec_A and vec_B, and $\|vec\|$ is a norm of vector vec.

Besides a simple vector space model using co-occurrence frequency as mentioned above, vector space model using TF-IDF as a weight for each co-occurrence frequency has been proposed.

2.2 Technique based on the Statistical Model

Based on the statistical decision theory(Maeda, 2000), this technique minimizes an error rate that is a probability of registering an unknown word in a wrong node. This technique can be defined as follows:

$$d_{Bayes}\left(y^M\right) = \arg\max_{x \in NODE} \int_\Theta p\left(\theta \mid w^N z^N\right) p(x \mid \theta)\, d\theta \prod_{i=1}^{M} \int_\Theta p\left(\theta \mid w^N z^N, x, y^{i-1}\right) p\left(y_i \mid x, \theta\right) d\theta \quad (1)$$

Here, $p(\theta)$ shows a prior probability density function of parameter θ.

Moreover, the integration part can be transformed as below by assuming a beta distribution as a prior probability density function $p(\theta)$ of the parameter θ. $\beta(x)$ shows a parameter of a beta distribution corresponding to $p(x \mid \theta)$.

$$\int_\Theta p\left(\theta \mid w^N z^N\right) p(x \mid \theta)\, d\theta = \frac{co\left(x \mid w^N\right) + \beta(x)}{\sum_x \left(co\left(x \mid w^N\right) + \beta(x) \right)} \quad (2)$$

$$\int_\Theta p\left(\theta \mid w^N z^N, x, y^{i-1}\right) p\left(y_i \mid x, \theta\right) d\theta = \frac{co\left(xy_i \mid w^N z^N\right) + co\left(y_i \mid y^{i-1}\right) + \beta\left(y_i \mid x\right)}{\sum_{y_i} \left(co\left(xy_i \mid w^N z^N\right) + co\left(y_i \mid y^{i-1}\right) + \beta\left(y_i \mid x\right) \right)} \quad (3)$$

3. Proposed Technique

A proposed technique process an unknown word by evaluating the relevance between words using an Association Mechanism which has already been proposed. Concretely, semantic relation between word at a node of a thesaurus and an unknown word is evaluated with the Degree of Association then the unknown word is registered to a node with the closest relation.

An Association Mechanism consists of a Concept Base(Hirose et al., 2001)(Kojima et al., 2002) and the Degree of Association Algorithm(Watabe & Kawaoka, 2001). A Concept Base generates semantics from a certain word, and the Degree of Association Algorithm uses results of a expanded semantics to express the relation between one word and the other with a numeric value.

3.1 Concept Base

A Concept Base is a large-scale database constructed both manually and automatically from multiple electronic dictionaries. It has concept words, which are entry words taken from electronic dictionaries, and concept attributes, which are content words in the explanations of each entry word. In our research, a Concept Base containing approximately 90,000 concepts was used. The Concept Base went through auto refining process after the base had

been manually constructed. In this processing, inappropriate attributes from the standpoint of human sensibility were deleted and necessary attributes were added.
In the Concept Base, Concept A is expressed by Attributes a_i indicating the features and meaning of the concept in relation to a Weight w_i denoting how important an Attribute a_i is in expressing the meaning of Concept A. Assuming that the number of attributes of Concept A is N, Concept A is expressed as indicated below. Here, the Attributes a_i are called Primary Attributes of Concept A.

$$A = \{(a_1, w_1), (a_2, w_2), ..., (a_N, w_N)\} \tag{4}$$

Because the primary Attributes a_i of Concept A are taken as the concepts defined in the Concept Base, attributes can be similarly elucidated from a_i. The Attributes a_{ij} of a_i are called the Secondary Attributes of Concept A. Figure 1 shows the elements of the Concept "train" expanded as far as the Secondary Attributes.

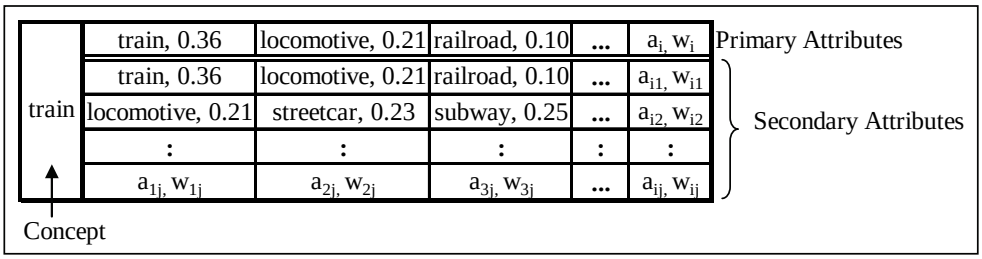


Fig. 1. Example demonstrating the Concept "train" expanded as far as Secondary

3.2 Calculation of the Degree of Association

For Concepts A and B with Primary Attributes a_i and b_i and Weights u_i and v_j, if the numbers of attributes are L and M, respectively ($L \leq M$), the concepts can be expressed as follows:

$$\begin{aligned} A &= \{(a_1, u_1), (a_2, u_2), ..., (a_L, u_L)\} \\ B &= \{(b_1, v_1), (b_2, v_2), ..., (b_M, b_M)\} \end{aligned} \tag{5}$$

The Degree of Identity $I(A, B)$ between Concepts A and B is defined as follows (the sum of the weights of the various concepts is normalized to 1):

$$I(A,B) = \sum_{a_i = b_i} \min(u_i, v_j) \tag{6}$$

The Degree of Association is obtained by calculating the Degree of Identity for all of the targeted Primary Attribute combinations and then by determining the correspondence between Primary Attributes. Specifically, priority is given to determine the correspondence between matching Primary Attributes. The correspondence between Primary Attributes that do not match is determined so as to maximize the total degree of matching. Using the

degree of matching, it is possible to consider the Degree of Association even for Primary Attributes that do not match perfectly.
When the correspondences are thus determined, the Degree of Association R (A, B) between Concepts A and B is as follows:

$$R(A,B)=\sum_{i=1}^{L} I(a_i,b_{xi})(u_i+v_{xi})\times\{\min(u_i,v_{xi})/\max(u_i,v_{xi})\}/2 \tag{7}$$

In other words, the Degree of Association is proportional to: the Degree of Identity of the corresponding Primary Attributes, the average of the weights of those attributes and the weight ratios.

4. Unknown Word Registration Experiment

4.1 The Thesaurus used in This Experiment

A thesaurus is a dictionary where words are semantically classified and generally indicated with a tree structure. The thesaurus has two types: 1) a classification thesaurus with words only on leaf nodes and 2) a hierarchical thesaurus with words on root nodes and intermediate nodes besides leaf nodes.
In this paper, a hierarchical NTT thesaurus(NTT, 1997) was used for the experiment of the unknown word registration. Figure 2 shows a part of NTT thesaurus.

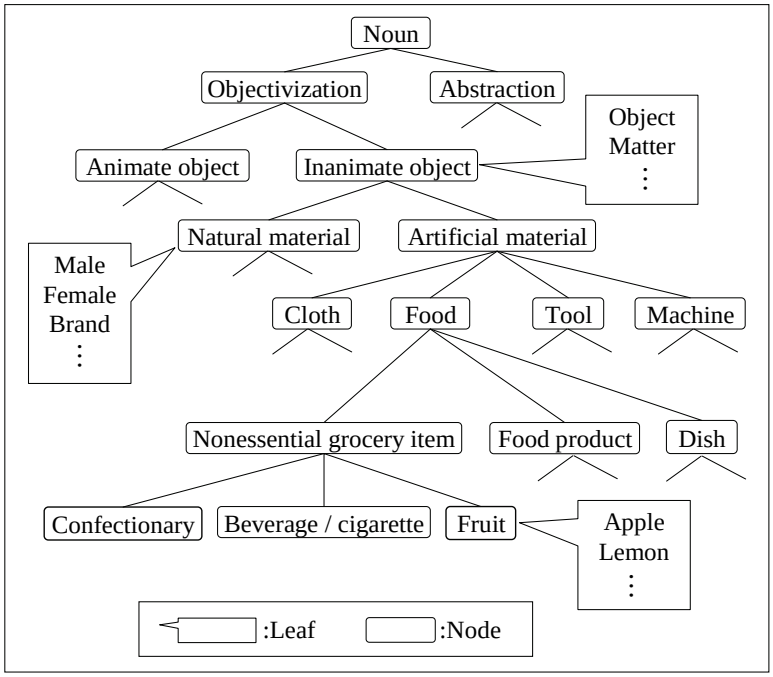


Fig. 2. An example of NTT thesaurus used in this paper

4.2 Method of Experiment

1000 words were extracted as unknown words from the words registered in NTT thesaurus explained in section 4.1. Two-stage sampling method was used as the extraction method. On the first stage, an equal probability sampling was carried out on the nodes with ten or more registered words. Then on the second stage, an equal probability sampling of non-restoration was carried out on noun words at the nodes. Examples of the words extracted as unknown words are shown in table 1.

Registered node	Unknown word
Remove	Diversion
Inversion	Opposition
Rejection	Declination
Union	Bridal
Price	Advance
Woman	Girl
Dropping	Drip
Superior	Seniority
Detainment	Captivity
City	Suburb

Table 1. Examples of unknown words

4.3 Evaluation

We evaluated each technique according to the correct answer rate of the top 10 candidate nodes to register an unknown word. Here, the node where the unknown word is registered in NTT thesaurus is considered as correct answer. When the unknown word is registered in two nodes or more, we judge the answer correct if the outputted node matches one of the registered nodes.

A result of an unknown word registration experiment is shown in figure 3. The axis of abscissas in figure 3 is the number of the considered accumulative candidates and the spindle is the correct answer rate (rate of accuracy). "Cos" is a vector space model using only co-occurrence frequency, "TF-IDF" is the vector space method using TF-IDF for a weight of co-occurrence frequency, "Bayes" is a technique using statistical model (Bayes theory) introduced in section 2.2 and "DA" is proposed technique in this paper.

In addition, the accuracy of this unknown word registration by human is approximately 89.4%.

4.4 Discussion

Figure 3 shows that the proposed technique is generally better than the traditional techniques. When the first answer was outputted, the accuracy improved only approximately 4%. But the accuracy improved approximately 20% if the top five answers or more were output. This result suggests that the proposed technique which semantically extends a word using the Concept Base and evaluates a semantic relation between words using the Degree of Association should be able to understand a vocabulary efficiently than the traditional techniques based on the probability and statistics.

As described in section 4.3, the accuracy when human solves the test used in this paper is approximately 89.4%. So, when human's accuracy is considered 100%, the accuracy of the first answer outputted by proposed technique is approximately 45.9% and the accuracy of the top 10 answers outputted by proposed technique is approximately 82.4%. Thus, when the top four answers or more are outputted, performance of proposed technique approaches performance of human by approximately 70% or more.

The main purpose of this paper is to construct a large-scale language dictionary and a corpus not automatically but semi-automatically. Therefore we think that it is more important to have the correct answer efficiently included in two or more answer candidates than in the first answer. So, it is considered that the proposed technique in this paper is a very effective technique.

However, we do not think that the accuracy of the first answer is enough. The proposed technique calculates the Degree of Association between an unknown word and a node of a thesaurus using only words at the node. In the future, we would like to improve the accuracy by a new registration method which uses registered nodes and leaf nodes extending from the node for the calculation of Degree of Association between an unknown word and a node in a thesaurus. Moreover, in this paper an unknown word was pseudo made from the word registered in an existing thesaurus. However, in the future, we would like to conduct a similar experiment which uses a true unknown word not registered in an existing thesaurus and the Concept Base.

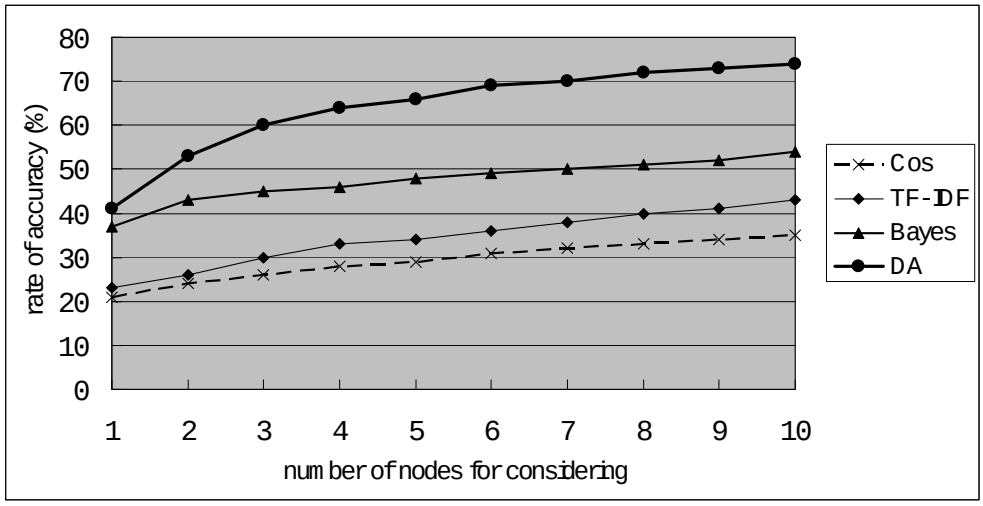


Fig. 3. A result of an unknown word registration experiment

5. Conclusion

In this paper, to reduce costs, resources and time, we proposed a technique which semi-automatically constructs a large-scale dictionary and corpus by using an Association Mechanism based on the Concept Base and the Degree of Association.

The proposed technique was able to improve the accuracy approximately 20% as a result compared with the traditional techniques.
In addition, when the top four answers or more were outputted, performance of proposed technique approached performance of human by approximately 70% or more.

References

Sony. http://www.sony.jp/products/Consumer/aibo/

Honda. http://www.honda.co.jp/ASIMO/

Uramoto, N. (1996). Corpus-based Thesaurus – Positioning Word in Existing Thesaurus Using Statistical Information from a Corpus, Journal of Information Processing Society of Japan, vol.37, No.12, pp.2182-2189.

Maeda, Y. (2000). A Note on Positioning Unknown Words in an Existing Thesaurus Based upon Statistical Decision Theory, The IEICE transactions on information and systems (Japanese edition), Vol.J83-A, No.6, pp.72-710.

Hirose, T.; Watabe, H. & Kawaoka, T. (2002). Automatic Refinement Method of Concept-base Considering the Rule between Concepts and Frequency of Appearance as an Attribute, *Technical Report of the Institute of Electronics, Information and Communication Engineers*, *NLC2001*(93), pp.109-116.

Kojima, K.; Watabe, H. & Kawaoka, T. (2002). A Method of a Concept-base Construction for an Association System: Deciding Attribute Weights Based on the Degree of Attribute Reliability, *Journal of Natural Language Processing*, *9*(5), pp.93-110.

Watabe, H. & Kawaoka, T. (2001). Measuring Degree of Association between Concepts for Commonsense Judgements, *Journal of Natural Language Processing*, *8*(2), pp.39-54.

NTT Communication Science Laboratory. (1997). *NIHONGOGOITAIKEI*, Iwanami Shoten, ISBN4-00-009884-5 C3581.

17

Homography-Based Control of Nonholonomic Mobile Robots: a Digital Approach

Andrea Usai and Paolo Di Giamberardino
University of Rome "La Sapienza"
Italy

1. Introduction

Why does one need to waste time deriving a discrete-time model from a continuous-time one? The answer is: it depends on how fast the system dynamics is.
The classical approach, especially for nonlinear systems, is to develop a continuous-time control law and then realize the controller by means of a computer implementation.
Such an approach can lead to poor controller performance due to the intrinsic system approximation: the sampling period, at which the controller commands are updated, fixes the sampling period at which the system dynamics is observed.
All the works done in the field of visual servoing, deal with continuous time systems (see, for example, in the case of mobile robots: Chen et al., 2006; Lopez-Nicolas et al., 2006; Mariottini et al., 2006). Image acquisition, elaboration and the nonlinear control law computation are very time consuming tasks. So it is not uncommon that the system is controlled with a sampling rate of 0.5Hz. Is it slow? It obviously depends.
To face the drawbacks of poor system approximation, the trivial solution is slowing down the system dynamics slowing down the controls variation rate. This is not always possible, but when the kinematic model is considered, the system dynamics depends only on the velocity imposed by the controller (there is no drift in the model) and so it is quite easy to limit the effect of a poor system approximation.
In general, taking into account the discrete-time nature of the controlled system can lead to better closed-loop system performances. This means that the control law evaluated directly in the discrete-time domain can better address the discrete-time evolution of the controlled system. This is well understood in the linear case and it is also true for (at least) a particular case of nonlinear systems, the ones that admits a finite or an exact sampled representation (Monaco and Normand-Cyrot, 2001). In fact, the possibility of an exact discretization of the continuous-time model is necessary to lead to the performances improvement previously discussed, since in this case no approximations are performed in the conversion continuous time - discrete time. The commonly used approximations can lead to a discrete-time model with a behavior that diverges from the continuous-time one.
In Section 2, the system model is presented. From this model a dicrete-time derivation is shown in Section 3. In Section 4, the design of a multirate digital control is then described

and discussed, showing that exact solutions are obtained. An improved control (and planning) strategy for the discrete-time system is introduced and applied. Simulation results are reported to put in evidence the effectiveness of the proposed approach.

2. The Camera-Cycle Model

In this section the kinematic model of the system composed of a mobile robot (an unicycle) and a camera, used as a feedback sensor to close the control loop, is presented.
To derive the mobile robot state, the relationship involving the image space projections of points that lie on the floor plane, taken from two different camera poses, are used. Such a relationship is called homography. A complete presentation of such projective relations and their properties is shown in (R. Hartley, 2003).

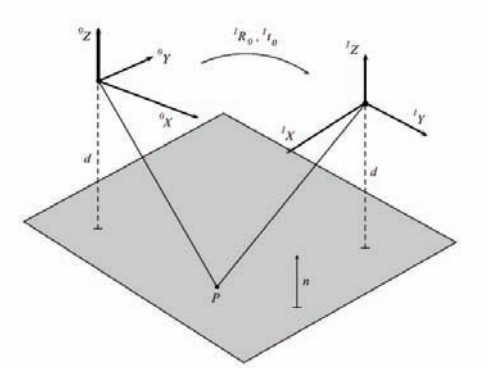


Fig. 1. 2-view geometry induced by the mobile robot.

2.1 The Geometric Model

With reference to Figure 1, the relationship between the coordinates of the point P in the two frames is

$$\begin{bmatrix} {}^1P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^1R_0 & {}^1t_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} \tag{1}$$

It is an affine relation that becomes a linear one in the homogeneous coordinate system. If the point P belongs to a plane in the 3D space with normal versor n and distance from the origin d, it holds that

$$\begin{bmatrix} n^T & d \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} = 0 \quad \Rightarrow \quad -\frac{n^T {}^0P}{d} = 1 \tag{2}$$

note that $d > 0$, since the interest is in the planes observed by a camera and so they don't pass trough the optical center (that is the camera coordinate system origin).

Combining the Equation 1 and the right term of 2, the following relation holds

$$ {}^1P = \left({}^1R_0 - \frac{1}{d} {}^1t_0 n^T \right) {}^0P = H \, {}^0P \tag{3} $$

The two frame systems in Figure 1 represent the robot frame after a certain movement on a planar floor.
Choosing the unicycle-like model for the mobile robot, the matrix H become

$$ H = \begin{bmatrix} \cos\theta & \sin\theta & \frac{1}{d}\left(X\cos\theta + Y\sin\theta \right) \\ -\sin\theta & \cos\theta & \frac{1}{d}\left(-X\sin\theta + Y\cos\theta \right) \\ 0 & 0 & 1 \end{bmatrix} \tag{4} $$

is the homography induced by the floor plane during a robot movement between two allowable poses. Note that $[X,Y,\theta]^T$ is the state vector of the mobile robot with reference to the first coordinate system.

2.2 The Kinematic Model

Taking four entries of matrix H such that

$$ \begin{aligned} h_1 &= \cos\theta \\ h_2 &= \sin\theta \\ h_3 &= x\cos\theta + y\sin\theta \\ h_4 &= -x\sin\theta + y\cos\theta \end{aligned} \tag{5} $$

and noting that, for the sake of simplicity, the distance d has been chosen equal to one, since it is just a scale factor that can be taken into account in the sequel, the kinematic unicycle model is

$$ \begin{aligned} \dot{X} &= v\cos\theta \\ \dot{Y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{6} $$

where v and ω are, respectively, the linear and angular velocity control of the unicycle.
Differentiating the system in Equation 5 with respect to the time and combining it with the system in Equation 6, one obtains

$$\begin{aligned}\dot{h}_1 &= -h_2\omega \\ \dot{h}_2 &= h_1\omega \\ \dot{h}_3 &= h_4\omega + v/d = h_4\omega + {}_d v \\ \dot{h}_4 &= -h_3\omega\end{aligned} \tag{7}$$

that is the kinematic model of the homography induced by the mobile robot movement.

3. From Continuous to Discrete Time Model

In the first part of this section, it will be presented how to derive a the discrete-time system model from the continuous-time one. Afterwards, a control (and planning) strategy for the discrete-time system is introduced and applied. Simulations will be presented to prove the presented strategy effectiveness.

3.1 The General Case

With reference to Figure 1, the relationship between the coordinates of the point P in the two frames is

$$\dot{x} = f(x) + \sum_{i=1}^{m} u_i g_i(x) \tag{8}$$

with f , g_1 , ..., $g_m : M \rightarrow R^n$, analytical vector fields.
To derive a discrete-time system from the previous one, suppose to keep constant the controls u_1 , ..., u_m , by means of a zero order holder, for $t \in [kT, (k + 1)T)$ and $k \in N$. Suppose that the system output is sampled (and acquired) every T seconds, too. The whole system composed by a z.o.h , the system and the sampler is equivalent to a discrete-time system.
Following (Monaco and Normand-Cyrot, 1985; Monaco and Normand-Cyrot, 2001), it is possible to characterize the discrete-time system derived by a continuous-time nonlinear system. Sampling the system in Equation 8 with a sampling time T , the discrete-time dynamics becomes

$$\begin{aligned}x(k+1) = x(k) + T\left(f + \sum_{i=1}^{m} u_i(k) g_i \right)\Bigg|_{x(k)} + \\ + \frac{T^2}{2}\left(L_{f+\sum_{i=1}^{m} u_i(k) g_i} \right)^2 (Id) + \ldots = \\ = F^T(x(k), u(k))\end{aligned} \tag{9}$$

where $L(.)$ denotes the Lie derivative. It is possible to see that, this series is locally convergent choosing an appropriate T . See (Monaco and Normand-Cyrot, 1985) for details.
The problem here is the analytical expression of $F^T(x(k),u(k))$. In general, it doesn't exist. Otherwise, if from the series of Equation 9 it is possible to derive an analytical expression for

its limit function, the system of Equation 8 is said to be exactly discretizable and its limit function is called an exact sampled representation of 8.
If, better, the series results to be finite, in the sense that all the terms from a certain index on goes to zero, a finite sampled representation is obtained.
Finite sampled representations are transformed, under coordinates changes, into exact sampled ones ((Monaco and Normand-Cyrot, 2001)). As obvious, finite discretizability is not coordinate free, while exact is. Note that the existence of an exact sampled representation corresponds to analytical integrability.
A nonholonomic system as the one of Equation 8, can be transformed into a chained form system by means of a coordinate change.
This leads to a useful property for discretization pointed out in (Monaco and Normand-Cyrot, 1992). In fact it can be seen that a quite large class of nonholonomic systems admit exact sampled models (polynomial state equations). Among them, one finds the chained form systems which can be associated to many mechanical systems by means of state feedbacks and coordinates changes.
So, in general, to be able to get a finite discretization of the nonlinear system of Equation 8, one needs to find a coordinates change (if it exists) that allows to derive an exact sampled model from the nonlinear one.

3.2 The Camera-Cycle Case

Suppose the controlled inputs are piecewise constant, such that

$$\begin{cases} v(t) = v_k \\ \omega(t) = \omega_k \end{cases} \quad t \in \left[kT, (k+1)T \right) \tag{10}$$

where $k = 0, 1, ..$ and T is the sampling period.
Since the controls are constant, it is possible to integrate the system in Equation 7 in a linear fashion. It yields to

$$\begin{aligned} \begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix} &= A(\omega_k) \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} \\ \begin{bmatrix} h_3(k+1) \\ h_4(k+1) \end{bmatrix} &= A(\omega_k)^T \begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} + B(\omega_k)_d v_k \end{aligned} \tag{11}$$

where

$$\begin{aligned} A(\omega_k) &= \begin{bmatrix} \cos \omega_k T & -\sin \omega_k T \\ \sin \omega_k T & \cos \omega_k T \end{bmatrix} \\ B(\omega_k) &= \begin{bmatrix} \sin\left(\omega_k T\right)/\omega_k \\ \left(\cos\left(\omega_k T\right) - 1\right)/\omega_k \end{bmatrix} \end{aligned} \tag{12}$$

If one considers the angular velocity input as a time-varying parameter, the system in Equation 11 become a linear time-varying system. Such a property allows an easy way to compute the evolution of the system. Precisely, its evolution becomes

$$\begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1}) \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix}$$
$$\begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1})^T \begin{bmatrix} h_3(0) \\ h_4(0) \end{bmatrix} + \sum_{j=0}^{k-2} \left(\prod_{i=j+1}^{k-1} A(\omega_{k-i})^T \right) B(\omega_j)\, {}_d v_j \tag{13}$$

where the sequences $\{v_k\}$ and $\{\omega_k\}$ are the control inputs. This structure will be useful in the sequel for the control law computation.

4. Controlling a Discrete-Time Nonholonomic System

Interestingly, difficult continuous control problems may benefit of a preliminary sampling procedure of the dynamics, so approaching the problem in the discrete time domain instead of in the continuous one.
Starting from a discrete-time system representation, it is possible to compute a control strategy that solves steering problems of nonholonomic systems. In (Monaco and Normand-Cyrot, 1992) it has been proposed to use a multirate digital control for solving nonholonomic control problems, and in several works its effectiveness has been shown (for example (Chelouah et al., 1993; Di Giamberardino et al., 1996a; Di Giamberardino et al., 1996b; Di Giamberardino, 2001)).

4.1 Camera-cycle Multirate Control

The system under study is the one in Equation 11 and the form of its state evolution in Equation 13.
The problem to face is to steer the system from the initial state $h_0 = [1, 0, 0, 0]^T$ (obviously corresponds to the origin of the configuration space of the unicycle) to a desired state ${}^d h$, using a multirate controller. If r is the number of sampling periods chosen, setting the angular velocity constant over all the motion, one gets for the state evolution

$$\begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} = A^r({}^c\omega) \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} h_3(r) \\ h_4(r) \end{bmatrix} = A^{r-1}({}^c\omega) B({}^c\omega)\, {}_d v_0 + A^{r-2}({}^c\omega) B({}^c\omega)\, {}_d v_1 + \ldots + B({}^c\omega)\, {}_d v_{r-1} \tag{14}$$

At this point, given a desired state, one just need to compute the controls. The angular velocity ${}^c\omega$ is firstly calculated such that:

$$\begin{bmatrix} {}^d h_1 \\ {}^d h_2 \end{bmatrix} = A^r({}^c\omega) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{15}$$

Once ${}^c\omega$ is chosen, the linear velocity values ${}_d v_0$, ..., ${}_d v_{r-1}$ can be calculated solving the linear system

$$\begin{bmatrix} {}^d h_3 \\ {}^d h_4 \end{bmatrix} = \begin{bmatrix} A^{r-1}({}^c\omega)B({}^c\omega) & A^{r-2}({}^c\omega)B({}^c\omega) & \dots & B({}^c\omega) \end{bmatrix} \begin{bmatrix} {}_d v_0 \\ {}_d v_1 \\ \dots \\ {}_d v_{r-1} \end{bmatrix} = RV \tag{16}$$

which is easily derived from the second two equations of 14.
Note that, for steering from the initial state to any other state configuration, at least $r = 2$ steps are needed, except for the configuration that present the same orientation of the initial one. More precisely, it can be seen that if this occurs, the angular velocity ${}^c\omega$ is equal to zero or Π/T and the matrix R in Equation 16 become singular.
Exactly that matrix shows the reachability space of the discrete-time system: if ${}^c\omega \neq \{0, \Pi/T\}$ then the vector B is rotated r-times and the whole configuration space is spanned.
Furthermore, if 2 is the minimum multirate order to guarantee the reachability of any point of the configuration, one can choose a multirate order such that $r > 2$ and the further degrees of freedom in the controls can be used to accomplish the task obtaining a smoother trajectory or avoiding some obstacles, for instance. Note that it can be achieved solving a quadratic programming problem as

$$\min_{v \in V} \frac{1}{2} V^T \Sigma V + \Gamma V \tag{17}$$

where Σ and Γ are two weighting matrixes, such that the robot reaches the desired pose, granting some optimal objectives. Other constraints can be easily added to take account of further mobile robot movements requirements.

4.2 Closing the Loop with the Planning Strategy

Let $[{}^d\theta, {}^d h_3, {}^d h_4]^T$ be the desired system state and mark the actual one with the subscript k. Note that, in the control law development, the orientation of the mobile robot is used, instead of the first two components of the system of Equation 11.
Summarize the algorithm steps as

0. Set $r_k = r$.
1. Choose
$$ {}^c\omega = \left({}^d\theta - \theta_k \right) \Big/ r_k T$$
2. Compute the control sequence V as
$$\min_V V^T V$$
such that
$$RV = \begin{bmatrix} {}^d h_3 \\ {}^d h_4 \end{bmatrix}$$
with the same notation of Equation 16.

3. If r_k>2 then r_k= r_k+1 and go to step *1*. Otherwise, the algorithm ends.

The choice of the cost function shown leads to a planned path length minimization. If the orientation error of the point 1 is equal to zero, it needs to be perturbed in order to guarantee some solution admissibility to the programming problem of point 2.
Furthermore, since the kinematic controlled model derives directly from an homography, it is possible use the homographies compositional property to easily update the desired pose, from the actual one, at every control computation step. Exactly, since

$$ {}^{d}H_0 = {}^{d}H_{r-1}\,{}^{r-1}H_{r-2}\ldots{}^{k}H_{k-1}\ldots{}^{1}H_0 \tag{18} $$

it is possible to easily update the desired pose as needed for the close-loop control strategy.
A Simulated path is presented in Figures 3: the ideal simulated steer execution (Figure 2) is perturbed by the presence of some additive noise in the controls. This simulate the effect of some non ideal controller behavior (wheel slipping, actuators dynamics, ...). The constrained quadratic problems involved in the controls computation are solved using an implementation of the algorithm presented in (Coleman and Li, 1996).

4.3 Setting Up the Trajectory Planning

The angular velocity defines the span of the system configuration space by means of the vectors in the matrix *R* of the Equaion 16: varying it during the planning can lead to a better functional minimization. Moreover, it is needed to settle the ω choice in case of no orientation error.
First of all, we need ω is equal to zero at the beginning of the planned path and at the end of it. Consider the function

$$ \omega_0(k) = {}^{\max}\omega_0 \sin^2\left(k\frac{\pi}{r}\right) \quad k \in \{0,1,\ldots,r\} \tag{19} $$

it starts from zero and softly goes to 1 up to return, as softly as before, to 0. For using this function as angular velocity control we have to find its maximum value to take to zero the orientation error. It can be done setting

$$ {}^{d}\theta = {}^{\max}\omega_0 \sum_{k=0}^{r-1} \sin^2\left(k\frac{\pi}{r}\right) \tag{20} $$

In order to face the null orientation error case, intuitively, the mobile robot should first point the desired pose and then compensate for the desired orientation. Take a look at the function in Equation 19 with the doubled frequency

$$ \omega_p(k) = \begin{cases} +{}^{\max}\omega_0 \sin^2\left(k\dfrac{2\pi}{r}\right) & k \in \{0,1,\ldots,r \setminus 2\} \\ -{}^{\max}\omega_0 \sin^2\left(k\dfrac{2\pi}{r}\right) & k \in \{r \setminus 2+1,\ldots,r\} \end{cases} \tag{21} $$

where with the symbol \ it is denoted the integer division operation. Note that integral of the function in the previous Equation is zero, in the interval of interest, as the final orientation displacement, due to its contribution. After half interval can be found the maximum velocity needed to cancel the pointing angle displacement as in Equation 20. The angular velocity is chosen as

$$\omega(k) = \omega_o(k) + \omega_p(k) \tag{22}$$

Since we want to get the shortest path, from the planning strategy, we minimize $V^T V$. Furthermore, we would like to minimize the variations between the planned controls ${}_d v_0,\ldots,$ ${}_d v_{r-1}$, in order to have a smoother behavior of the controlled robot movement. It is possible to accomplish to these specifications, solving the quadratic programming problem

$$\min_{v \in V} \left[V^T \Sigma V + \gamma \sum_{k=1}^{r-1} \left({}_d v_k - {}_d v_{k-1} \right)^2 \right] \tag{23}$$

such that

$$\begin{aligned} & RV = b \\ & \left| {}_d v_k \right| < 1 \quad k = 0,\ldots,r-1 \\ & {}_d v_0 = {}_d v_{r-1} = 0 \end{aligned} \tag{24}$$

where the γ is a minimization parameter and the matrix R and the vector b are the same of Equation 16. Note that it is a quadratic programming problem since the second addend of the cost function of Equation 23 can be expressed as

$$V^T \begin{bmatrix} 1 & -1 & & & & & \\ -1 & 2 & -1 & & & & \\ & -1 & 2 & -1 & & & \\ & & \ldots & \ldots & \ldots & & \\ & & & -1 & 2 & -1 & \\ & & & & -1 & 1 & \end{bmatrix} V \tag{25}$$

The last two constraints of Equation 24 are added to make explicit that a pose to pose trajectory is planned. Note that the planned trajectory shown in Figure 4 it is shorter than the corresponding one of Figure 2. This happens because the programming problem referred to the trajectory of Figure 4 is more constrained than the other one: varying ω leads to a better suboptimal solution. As expected.

4.4 Trajectory Tracking: a Multirate Digital Approach

Once the trajectory is planned, it is necessary to introduce a tracking technique to execute it. Classical continuous approaches can be obviously applied: for instance, in Usai & Di

Giamberardino (2006), we use a linear controller to track a multirate planned trajectory, again for a visual servoing problem.
In the preceding sections, it has been introduced a close-loop control strategy, iterating the previously discussed planning technique. It will be now discussed an extension of this approach to derive a digital trajectory tracker.
Trivially, if we want to follow a chosen trajectory we can re-apply the multirate planning strategy to take the system from the actual point to a next intermediary one. The incessant desired pose update constraints the system to follow the previously planned path. More precisely, every m steps it is possible to re-solve a quadratic programming problem similar to the one solved for the planning problem but, this time, the desired pose is the one planned τ steps ahead the actual one.
Summarizing the described tracking algorithm, we have

0. Set r and plan the trajectory. Set $K=0$, $m_k=0$.
1. Execute the planned controls at step k and set $k=k+1$ and $m_k= m_k+1$.
2. If $k=r$, the algorithm stops.
3. If $m_k<m$ come back to point *1.*
4. *If* $k\leq(r-\tau)$, update the ω computing

$$\omega(i+1)=\omega(i)+\frac{{}^{d}\theta_{k-1}-\theta_{k}}{mT} \quad i=k,\ldots,k+m$$

And, subsequently, the sequence ${}_{d}v$ minimizing

$$\min_{{}^{m}V}({}^{m}V)^{T}({}^{m}V)+\gamma_1({}^{m}R\,{}^{m}V-{}^{m}b)^{T}({}^{m}R\,{}^{m}V-{}^{m}b)+\gamma_2({}^{d}v_{k+\tau}-{}_{d}v_{k+\tau})^{2}$$

5. Set $m_k=0$ and go to point *1.*

Note that the programming problem has been relaxed in order to have a smoother tracking behaviour. For the same reason it is added a term regarding the desired pose velocity.
Furthermore, the two parameters m and τ influence how the trajectory is tracked. For instance, if $\tau = 2$ the error in the trajectory execution is less than the one obtained with larger values of τ. On the other hand, large values of τ allow many degrees of freedom for the programming problem solution. This determines the deformation of the previously planned trajectory, putting in the programming problem new constraints. For example, these constraints would be useful to let the robot avoid other unexpected obstacles that come out during robot movement. An interesting implementation could be the one in which the parameter τ varies when some event occurs (a new obstacle detected by sensors), allowing the path deformation to a new feasible one. With regard to the parameter m (remark that $m < \tau$), since it influences how many times the control is computed during the tracking, it depends on the computer on which the controller will be implemented.

5. Conclusion

In this chapter, a kinematic model for a system composed by a mobile robot and a camera, has been presented.
Since such a model is exactly discretizable, it has been possible to propose a multirate digital control strategy able to steer the system to a desired pose in an exact way.

As can be seen from the trajectory obtained in the simulations of Figures 2 and 3, there are large differences between the ideal path execution and the perturbed one.
It happens because the iterated planning strategy has is no memory of the previously planned path. The controls are just constrained to take the system to the desired state. There is no control on how get there.
In a real implementation it is advisable to have a certain degree of predictability of the robot behavior during its movement and the respect to some optimal criteria (short paths, smooth movements, obstacles avoidance,...), too. This is why it has been chosen to present a separated planning phase and a subsequent the tracking phase. The effectiveness of the control scheme adopted has been verified by simulations and presented in Figures 4.

6. References

Chelouah, A., Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1993). Digital control of nonholonomic systems two case studies. *In Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 2664–2669vol.3.

Chen, J., Dixon, W., Dawson, M., and McIntyre, M. (2006). Homography-based visual servo tracking control of a wheeled mobile robot. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 22(2):406–415.

Coleman, T. and Li, Y. (1996). A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058.

Di Giamberardino, P. (2001). Control of nonlinear driftless dynamics: continuous solutions from discrete time design. *In Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1731–1736vol.2.

Di Giamberardino, P., Grassini, F., Monaco, S., and Normand-Cyrot, D. (1996a). Piecewise continuous control for a car-like robot: implementation and experimental results. *In Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 3, pages 3564-3569vol.3.

Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1996b). Digital control through Finite feedback discretizability. *In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3141–3146vol.4.

Lopez-Nicolas, G., Sagues, C., Guerrero, J., Kragic, D., and Jensfelt, P. (2006). Nonholonomic epipolar visual servoing. *In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2378–2384.

Mariottini, G., Prattichizzo, D., and Oriolo, G. (2006). Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera. *In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 538–544.

Monaco, S. and Normand-Cyrot, D. (1985). On the sampling of a linear analytic control system. *In Decision and Control, 1985., Proceedings of the 24th IEEE Conference on*, pages pp. 1457–1461.

Monaco, S. and Normand-Cyrot, D. (1992). An introduction to motion planning under multirate digital control. *In Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 1780–1785vol.2.

Monaco, S. and Normand-Cyrot, D. (2001). Issues on nonlinear digital control. *European Journal of Control*, 7(2-3).

R. Hartley, A. Z. (2003). *Multiple View Geometry in Computer Vision*. Number ISBN: 0-521-54051-8. Cambridge University Press.

Usai , A., Di Giamberardino, P. (2006). A multirate digital controller for non holonomic mobile robot pose regulation via visual feedback. *WSEAS Transactions on Systems*, 5, 1129–1136.

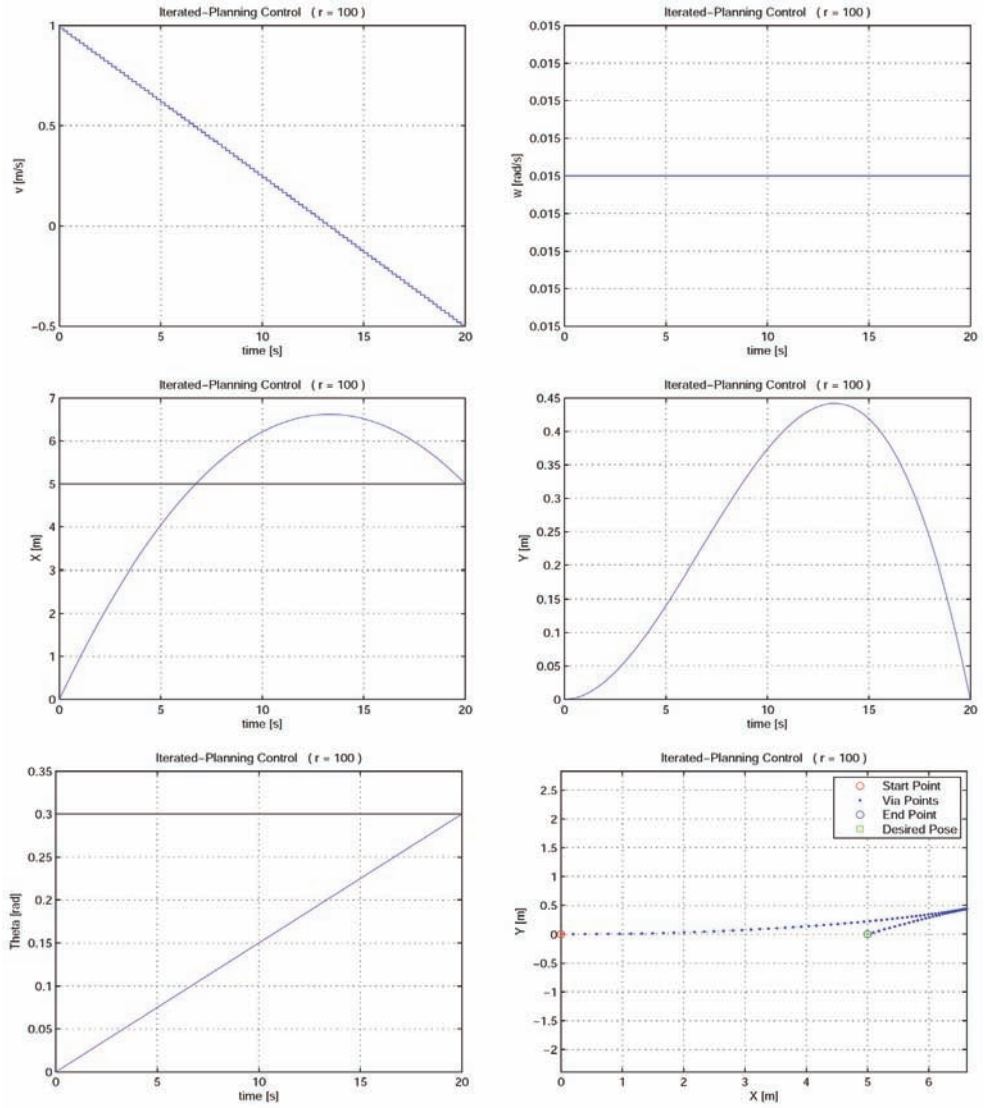


Fig. 2. Multirate control simulation ($d = 1m$). Ideal (no noise) path execution.

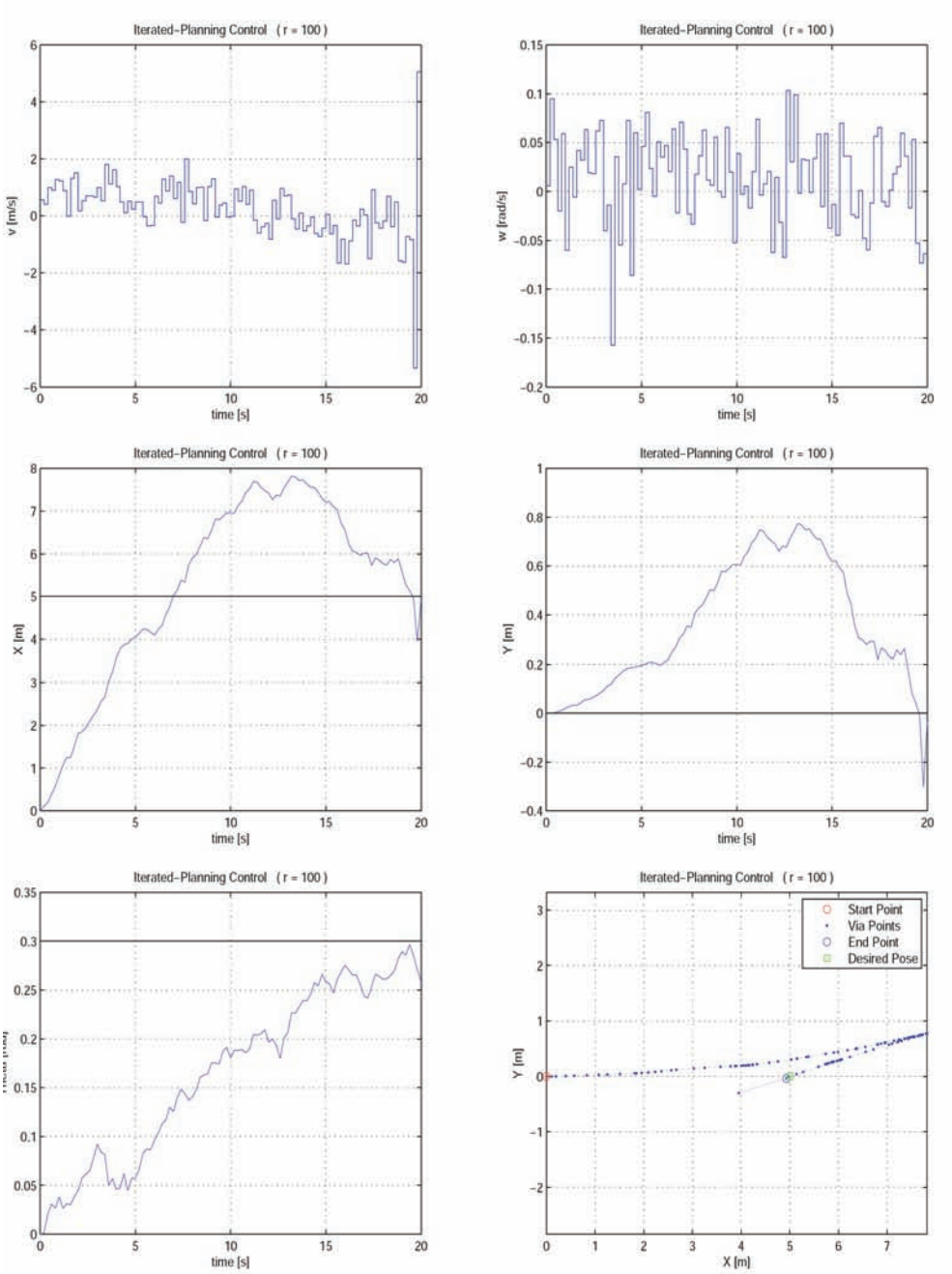


Fig. 3. Multirate control simulation (re-iterated planning, $d = 1m$). Additive random noise on controls (gaussian with std.dev. 0.5 and 0.05, for v and ω respectively).

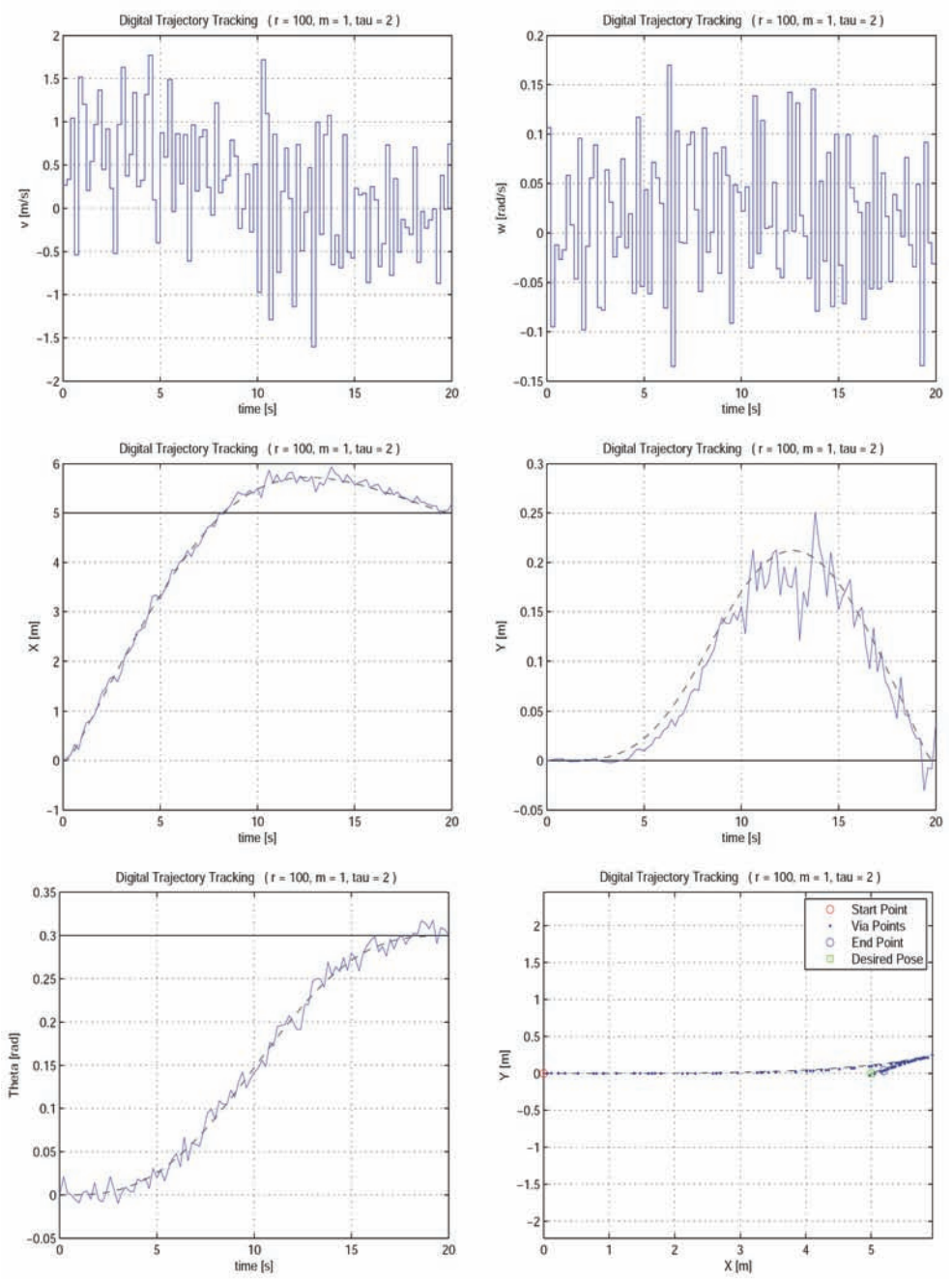


Fig. 4. Multirate control simulation (planning + tracking, $d = 1m$). Additive random noise on controls (gaussian with std.dev. 0.5 and 0.05, for v and ω respectively).

18

Fault Detection with Bayesian Network

Verron Sylvain, Tiplica Teodor, Kobi Abdessamad
LASQUO – ISTIA
France

1. Introduction

Nowadays, process control (or process monitoring) is becoming an essential task especially when dealing with complex manufacturing processes (like automatized processes containing a lot of sensors and actuators). In (Chiang et al., 2001), authors give two principal approaches to perform the process control, namely, data-driven techniques and analytical techniques. The analytical techniques are based on analytical (physical) models of the system and enable to simulate the system. Though, at each instant, the theoretical value of each sensor can be known for the normal operating state of the system. As a consequence, it is relatively easy to see if the real process values are similar to the theoretical values. But, the major drawback of this family of techniques is the fact that a detailed model of the process is required in order to control it efficiently. An effective detailed model can be very difficult, time consuming and expensive to obtain, particularly for large-scale systems with many variables. The data-driven approaches are a family of different techniques based on the analysis of the real data extracted from the process. These methods are based on rigorous statistical development of the process data (i.e. control charts, methods based on Principal Component Analysis, Projection to Latent Structure or Discriminant Analysis) (Chiang et al., 2001).
The process control can be viewed as a three-step procedure: the fault detection, the fault diagnosis and the process recovery.
Many data-driven techniques for the fault detection can be found in the literature: univariate control charts (Shewhart charts) (Shewhart, 1931), multivariate control charts (T^2, Q, MEWMA, MCUSUM charts) (Hotelling, 1947; Lowry et al., 1992; Pignatiello & Runger, 1990), and some PCA (Principal Component Analysis) based techniques (Jackson, 1985), like Moving PCA (Bakshi, 1998). In (Kano et al., 2002), authors make comparisons between these different techniques. Other important approaches are PLS (Projection to Latent Structures) based approaches (MacGregor & Kourti, 1995).
In order to accomplish the fault diagnosis, many approaches have been proposed (Kourti & MacGregor, 1995). The fault diagnosis procedure can also be considered as a classification task. Many classifiers have been developed (i.e. Fisher Discriminant Analysis (Duda et al., 2001), Support Vector Machine (Vapnik, 1995), k-nearest neighborhood (Cover & Hart, 1967), Artificial Neural Networks (Bishop, 2001) and bayesian classifiers (Friedman et al., 1997)).

When dealing with a great number of variables and faults, the performances of these classifiers are diminished. To overcome the inconvenience, a feature selection is often performed before the classification task. By feature selection one means the identification of the key process variables allowing capturing the signature (identity) of the various faults affecting the process.
In this article, we are presenting a new method for the detection and the diagnosis of faults in multivariate processes with bayesian networks.
The chapter is structured as follows: the second section present the utilization of the multivariate control charts for fault detection; the third section highlights some aspects on bayesian networks and particularly on bayesian network classifiers; in the fourth section we are showing how to model some multivariate control charts (T^2 and MEWMA), with a bayesian network. In the last section, we conclude on the proposed approach and give some outlooks.

2. Detection and classification

As the new method for the fault detection that we proposed is based on multivariate control charts and on discriminant analysis, we present this two techniques in this section.

2.1 Multivariate control charts

Hotelling (Hotelling, 1947) was the first to propose a control chart for the monitoring of multivariate processes: the T^2 control chart. (Hawkins, 1991) demonstrated that this chart represents the best statistical test for the detection of a change in the mean of a multivariate process. For a process with p variables, we can write the T^2 statistics as the following:

$$T^2=(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) \tag{1}$$

where: $\mathbf{x}$ is the observation vector of size $1\times p$, $\boldsymbol{\mu}$ is the mean vector of size $1\times p$, $\boldsymbol{\Sigma}$ is the variance-covariance matrix of size $p\times p$ and the symbol ^{T} represents the transpose of a vector or a matrix.
For each instant sampling, the T^2 will be compared to an upper control limit CL (lower control limit is fixed to 0) in order to conclude about the state of the process: if the value of T^2 is lower than the CL, then the process is "in control", either ($T^2 > CL$) the process is declared out of control signifying a fault has occurred in the process. The computation of the limit will depend of the estimation of the process parameters. Montgomery (Montgomery, 1997) gives the different computations for this limit.
The major drawback of the T^2 control chart is its moderate performances to detect small mean shifts. In order to solve this problem, other multivariate control charts have been proposed: MEWMA (Multivariate Exponentially Weighted Moving Average) (Lowry et al., 1992) and MCUSUM (Multivariate CUmulative SUM) (Pignatiello & Runger, 1990). These charts are respectively the multivariate analogous of the EWMA and CUSUM control charts.
The principle of the MEWMA control chart is to take into account the process evolution in weighting past observations extracted from the process. So, the MEWMA variable $\mathbf{y}_t$ is

computed recursively, for each sample, by the equation (2) where the initialization is given by $\mathbf{y}_0=\boldsymbol{\mu}$.

$$\mathbf{y}_t=\boldsymbol{\Lambda}\mathbf{x}_t+(\mathbf{I}-\boldsymbol{\Lambda})\mathbf{y}_{t-1} \tag{2}$$

In equation (2), $\mathbf{x}_t$ is the observation vector at instant t, $\mathbf{I}$ is the identity matrix and $\boldsymbol{\Lambda}$ the weighting diagonal matrix which elements $\lambda_1,\lambda_2,\cdots,\lambda_p$ are comprised between 0 and 1. ($0<\lambda_i\leq 1$).
Lowry proposed to use $\lambda_i=\lambda$ for $i=1,2,\cdots,p$ if there is no particular reason to differently weight the variables. In this case, the MEWMA control chart is considered "directionally invariant" because no direction in the multivariate space is advantaged. By choosing different λ_i in order to increase the detection performance in certain directions, one obtains a "directional" control chart which is quite more difficult to compute and to interpret. So, in the rest of this chapter, we will consider the non-directional MEWMA control chart ($\lambda_i=\lambda$ for all i). Thus, we can rewrite equation (2) as:

$$\mathbf{y}_t=\lambda\mathbf{x}_t+(1-\lambda)\mathbf{y}_{t-1} \tag{3}$$

On the MEWMA control chart, we plot the following statistic:

$$T_t^2=(\mathbf{y}_t)^T\boldsymbol{\Sigma}_{\mathbf{y}_t}^{-1}(\mathbf{y}_t) \tag{4}$$

Where $\boldsymbol{\Sigma}_{\mathbf{y}_t}$ is the variance-covariance matrix of the variable $\mathbf{y}$ at instant t. This matrix is defined as:

$$\boldsymbol{\Sigma}_{\mathbf{y}_t}=\left\{\frac{\lambda\left[1-(1-\lambda)^{2t}\right]}{2-\lambda}\right\}\boldsymbol{\Sigma} \tag{5}$$

But, if λ is not too small ($\lambda>0.1$), this matrix approaches rapidly (in 4 or 5 samples) his asymptotic value defined by:

$$\boldsymbol{\Sigma}_{\mathbf{y}}=\left\{\frac{\lambda}{2-\lambda}\right\}\boldsymbol{\Sigma} \tag{6}$$

The process is declared out-of-control if T_t^2 is greater than a control limit h_M. This limit is function of p and λ, in order to respect a given false alarm frequency (Lowry et al., 1992). We can precise that performances of the MEWMA control chart are function of λ. Indeed, a small λ allows a performing detection of small magnitude shifts, but a higher λ will be more adapted for large magnitude shifts. So, the choice of λ will be function of the

magnitude shift that one wants to detect. A particular case of the MEWMA control chart is the case where λ=1 . In this case, the MEWMA chart is equivalent to the T^2 control chart.

2.2 Discriminant analysis

The discriminant analysis can be viewed as a statistical technique for supervised classification, based on the Bayes rule. Indeed, for a problem with k classes, this technique allocates to a new observation $\mathbf{x}$ the class C_i having the maximal a posteriori probability ($P(C_i|\mathbf{x})$):

$$\mathbf{x} \in C_i \text{, if } \quad i=\underset{i=1,\cdots,k}{\arg\max}\left\{P(C_i|\mathbf{x})\right\} \tag{7}$$

This rule is called "Bayes rule" because it allows obtaining the value of $P(C_i|\mathbf{x})$ with the use of equation (8), where $P(C_i)$ represents the a priori probability of the class C_i .

$$P(C_i|\mathbf{x}) = \frac{P(C_i)P(\mathbf{x}|C_i)}{P(\mathbf{x})} \tag{8}$$

We can see that for each class, the denominator of equation (8) is the same and it will not interfere in the discriminant function. So, the equation (7) can be rewritten as:

$$\mathbf{x} \in C_i \text{, if } \quad i=\underset{i=1,\cdots,k}{\arg\max}\left\{P(C_i)P(\mathbf{x}|C_i)\right\} \tag{9}$$

But, for more simplicity, we will write this decision rule under the form of a cost function K as given by:

$$K_i(\mathbf{x}) = -2\log(P(C_i)P(\mathbf{x}|C_i)) \tag{10}$$

Thus, the attribution rule of a new observation $\mathbf{x}$ to a class C_i can be written by the following equation:

$$\mathbf{x} \in C_i \text{, if } \quad i=\underset{i=1,\cdots,k}{\arg\min}\left\{K_i(\mathbf{x})\right\} \tag{11}$$

Generally, this rule is applied with parametric distribution laws; the most used is the multivariate normal (Gaussian) distribution.

The multivariate Gaussian distribution is the generalization of the one-dimensional normal distribution to higher dimensions. The probability density function ϕ, conditionally to a

class C_i is given in equation (12), where $\boldsymbol{\mu}_i$ is the mean vector of the class C_i, and $\boldsymbol{\Sigma}_i$ is the variance-covariance matrix of the class C_i.

$$\phi(\mathbf{x}|C_i)=\frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_i|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right) \tag{12}$$

As the exact values of $\boldsymbol{\mu}_1,\boldsymbol{\Sigma}_1,\boldsymbol{\mu}_2,\boldsymbol{\Sigma}_2,\cdots,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k$ are generally unknown, it is necessary to estimate them with data. In order to estimate these parameters, one can use MLE (Maximum Likelihood Estimation). An advantage of MLE is the fact that this estimator has good convergence propriety when the sample size is large enough. More, this is one of the simplest techniques for the estimation of law parameters. In the case of the multivariate Gaussian distribution, the MLE of the mean vector is:

$$\hat{\boldsymbol{\mu}}=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \tag{13}$$

In the same way, an unbiased estimation of $\boldsymbol{\Sigma}$ is:

$$\hat{\boldsymbol{\Sigma}}=\frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i-\hat{\boldsymbol{\mu}})(\mathbf{x}_i-\hat{\boldsymbol{\mu}})^T \tag{14}$$

For a more detailed justification of these estimators, see (Duda et al., 2001).
The discriminant analysis rule (equation 10) applied in the case of the multivariate Gaussian distribution can be developed as:

$$K_i(\mathbf{x})=(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)-2\log(P(C_i))+\log(|\boldsymbol{\Sigma}_i|)+p\log(2\pi) \tag{15}$$

In the equation, we can see that the last term ($p\log(2\pi)$) is constant for each K_i and we have not to take it into account for the discrimination. This rule is named "Quadratic Discriminant Analysis". We can also remark that $(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)$ is the T^2 of $\mathbf{x}$ for the class C_i. If the assumption of independent variables is made, thus $\boldsymbol{\Sigma}_i$ is a diagonal matrix (all the covariances are null). This decision rule is also known as the Bayes classifier or naïve bayesian network. The major problem of the quadratic discriminant analysis is that this technique requires an important amount of data for the correct estimation of all the parameters. In the case of non-sufficient data, we can use the linear discriminant analysis.
The important assumption of the linear discriminant analysis, compared to the quadratic one, is the supposed equality of all variance-covariance matrices. So, for each class C_i, $\boldsymbol{\Sigma}_i=\boldsymbol{\Sigma}$, with $\boldsymbol{\Sigma}$ the pooled sample covariance matrix. This matrix is easily obtained with

equation (16) where n_i is the sample number of the class C_i and where n is the total number sample (so $n = n_1 + n_2 + \ldots + n_k$).

$$\boldsymbol{\Sigma} = \frac{(n_1 - 1)\boldsymbol{\Sigma}_1 + (n_2 - 1)\boldsymbol{\Sigma}_2 + \ldots + (n_k - 1)\boldsymbol{\Sigma}_k}{n - k} \tag{16}$$

So, in this case, equation (15) becomes:

$$K_i(\mathbf{x}) = \left(\mathbf{x} - \boldsymbol{\mu}_i\right)^T \boldsymbol{\Sigma}_i^{-1} \left(\mathbf{x} - \boldsymbol{\mu}_i\right) - 2\log\left(P(C_i)\right) + cst \tag{17}$$

where $\log(|\boldsymbol{\Sigma}|) + p\log(2\pi) = cst$. One can see in equation (17) that if the a priori probabilities ($P(C_i)$) of each class are equal, thus the decision rule comes to the computation of the T^2 for each class and the attribution of a new observation to the class with the lower T^2. This cost function makes linear bound between the different classes. But, as for the case of quadratic discriminant analysis, we can do the assumption that $\boldsymbol{\Sigma}$ is diagonal. The linear discriminant analysis is quite robust to the assumption of normality of each class and to assumption of equality of the different variance-covariance matrices. For these reasons, this technique is widely used and is considered as a reference method of supervised classification.

3. Discriminant analysis and bayesian networks

An interesting tool using statistics is bayesian network, an oriented probabilistic graphic model. Bayesian networks can be efficient supervised classifiers. So, after the general presentation of bayesian networks and especially bayesian classifiers, we will present how to make discriminant analysis with them.

3.1 Bayesian networks

A bayesian network (Pearl, 1988; Jensen, 1996) is a triplet {**G, E, D**} where:
{**G**} is a directed acyclic graph, **G=(V,A)**, where **V** is the set of nodes of **G**, and **A** is the set of edges of **G**,
{**E**} is a finite probabilistic space (Ω, Z, P), where Ω is a non-empty space, Z is a collection of subspace of Ω, and P is a probability measure on Z with $P(\Omega)=1$,
{**D**} is a set of random variables associated to the nodes of **G** and defined on **E** such as:

$$P(V_1, V_2, \ldots, V_n) = \prod_{i=1}^{n} P\left(V_i \middle| C(V_i)\right) \tag{18}$$

where $C(Vi)$ is the set of parents of Vi in the graph **G**.
Bayesian network classifiers are particular bayesian networks (Friedman et al., 1997). They always have a discrete node C coding the k different classes of the system. The remaining variables X_i represent the descriptors (variables) of the system.

A Naïve Bayesian Network (NBN) is a particular type of bayesian network classifiers (Langley et al., 1992). It is also known as the Bayes classifier. In a NBN, the class node is linked with all other variables of the system (descriptors) as indicated on the figure 1.

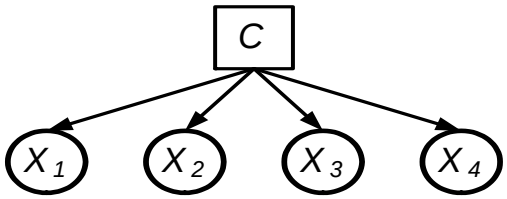


Fig. 1. Example of a Naïve Bayesian Network (NBN)

The NBN is called naïve because it makes the naïve (but strong) assumption that all descriptors (variables of the system) are class conditional statistically independent (no correlation between each descriptor in each class). In (Inza et al., 1999), authors made a comparative study of classification algorithms used in artificial intelligence. NBN is compared with other methods like k-nearest neighborhood, C4.5, decision tree and so on. If the independence assumption of the descriptors is verified and that probabilities are estimated with enough precision the NBN is an optimal classifier in term of misclassification rate (Domingos & Pazzani, 1996). This optimality is obtained when continuous variables are discretized in such a way that every distribution can be well approximated by discretization. Of course, the discretization of variables is a loss of information because it is a reduction of the variables space. But, assuming that continuous variables follow normal probability density function, we can deal with them directly in the network. And, if this assumption is verified, we keep this optimality. But, in many systems, it is very frequent to have high correlations between variables, and a NBN will not take into account these correlations. Extensions of NBN have been developed in order to solve this problem.
A first interesting extension is the TAN (Tree-Augmented bayesian Network) (Friedman et al., 1997). In a TAN, a maximum weighted spanning tree is constructed with the descriptors following the algorithm of (Chow & Liu, 1968). So, each descriptor will have at most one other descriptor as parent. After that, edges from the class node to each descriptor are added (like a NBN). An example of a TAN is given on the figure 2.

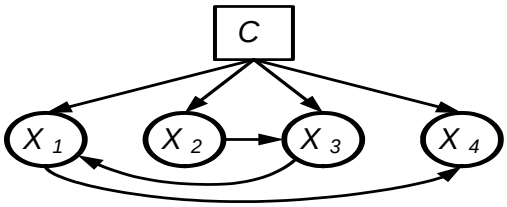


Fig. 2. Example of a Tree-Augmented bayesian Network (TAN)

Another extension is the k-dependence bayesian classifier structure (kDB structure) (Sahami, 1996) that extends the TAN structure allowing a maximum of k predictor parents plus the class for each predictor variable (TAN structures are equivalent to kDB structures

with k = 1). Finally, the bayesian multinets (Geiger & Heckerman, 1996) introduce a different structure for each value of the class variable (a particular case is to take a different TAN for each value of the class). In (Friedman et al., 1997), authors show that these classifiers give a lower misclassification rate than the NBN. But, these classifiers do not take into account all the correlations between variables.
An other extension to the NBN is the Condensed Semi Naïve Bayesian Network (CSNBN) (Kononenko, 1991). The principle of this classifier is to represent some variables in a joint node (i.e. some normally distributed variables can be modeled with a node representing a multivariate normal distribution). In this way, all correlations of the system will be taken into account. A CSNBN will be composed of two nodes: the class node and a multivariate node. An example of a CSNBN is given on the figure 3.

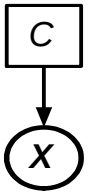


Fig. 3. Example of a Condensed Semi Naïve Bayesian Network (CSNBN)

3.2 Discriminant analysis as a bayesian network

Discriminant analysis techniques can be easily transposed to a bayesian network. Indeed, inference in a bayesian network is based on the Bayes rule, like the discriminant analysis (see section 2.2). The structure of the network in order to make classical discriminant analysis (quadratic or linear) on a system with p variables and k classes can be modelized with two linked nodes. The first one (node C) is a discrete node (with k modalities) representing the classes of the system. The second one (node $\mathbf{X}$) is a multivariate Gaussian node. We have previously presented this type of bayesian network: it is a CSNBN (figure 4). This network represents a multivariate normal law conditionally to the class, as the discriminant analysis. The choice between the different types of discriminant analysis (quadratic, linear, spherical, diagonal, etc) is made on the choice of the k variance-covariance matrices that will be attributed to $\mathbf{X}$. It is also possible to make a quadratic discriminant analysis in viewing all the univariate variables X_i composing the multivariate variable $\mathbf{X}$, and then, linking all them in order to take into account all possible relations between these univariate variables X_i. A four variable example of such structure is given on figure 4.

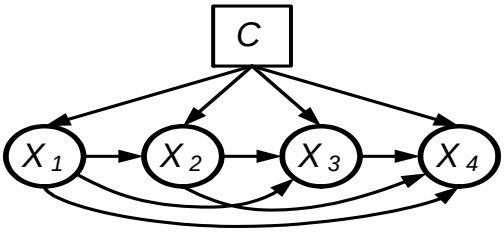


Fig. 4. Bayesian Network for Discriminant Analysis with incomplete observations

This figure 4 represents a succession of *p-1* linear regressions. The interest of this type of structure is the fact that it can give response even if the values of some X_i are unknown. However, the estimation of all the regressions parameters and the inference time will be more consequent than for the CSNBN. So, in the basic case (complete data), we will prefer CSNBN.

4. Multivariate control charts with a bayesian network

The detection, as we previously said, consists in identifying the presence of faults in the process. The detection can be considered as a classification in two classes: no fault in the process, and fault in the process. For the detection, we suppose to have samples for the normal working state of the system (we will call this class "In Control" (IC) class), but having fault samples should not be necessary. This classification type is called one-class classification (Tax & Duin, 2001). In contrast with normal classification problems where one tries to distinguish between two (or more) classes of objects, one-class classification tries to describe one class of objects, and distinguish it from all other possible objects. A useful solution for this is to create at least a second class, a virtual class named "Out-of-Control" (OC) class. This virtual class represents the set of observation that cannot be attribute to the IC class.

An example of one-class classification is the T^2 control chart (new observation is attributed to the IC class if his T^2 is lower than CL, and to the OC class if his T^2 is greater than CL). In this case, it is evident that the decision boundary can be represented as the CL. For this control chart, it is possible to represent, for a bidimensional example, the decision boundary induced by CL. On the figure 5, we can see that this boundary is an ellipse rounding the normal working class of the process (IC class), and that each observation out of this ellipse is attributed to the fault class (OC class).

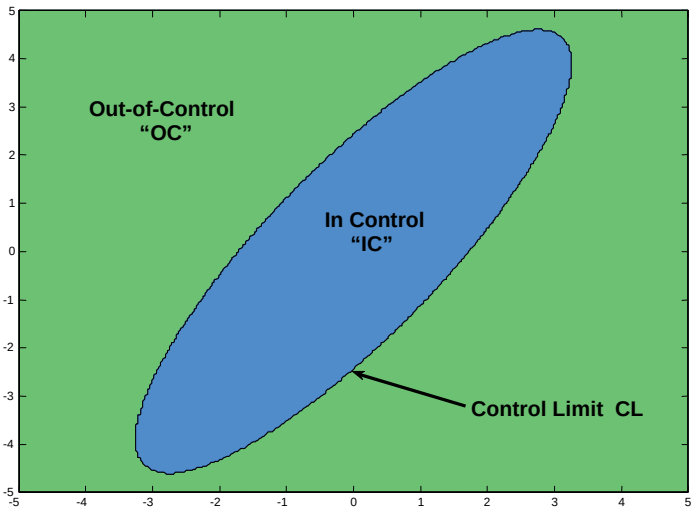


Fig. 5. Decision boundary of the control chart for a bivariate system

On the figure 5, we can see that the classification technique producing this boundary cannot be linear. So, it will be impossible to make such boundary with linear discriminant analysis. But, this boundary shape is typical of quadratic discriminant analysis.
In order to define the In Control class, we make the assumption that some samples of in control data are available. In the same way that for the control chart, these samples allow to estimate the mean vector $\boldsymbol{\mu}$ and the variance-covariance matrix $\boldsymbol{\Sigma}$ of the process in normal working conditions. The major problem is the definition of the virtual class OC because no sample is available. In studying the figure 5, we can see that if we make the assumption that the mean of the OC class is the same that the mean of the IC class, the only difference between the two classes is the fact that the OC class has more variability that the IC class. So, we will define the virtual class as following: the mean vector of the OC class is equal to the mean vector of the IC class; the variance-covariance matrix of the OC class is equal to $c\boldsymbol{\Sigma}$ where $\boldsymbol{\Sigma}$ is the variance-covariance matrix of the IC class, and where c is a coefficient strictly greater than 1. This coefficient c allows increasing the variability of the OC class in function of the variability of the IC class. The table 1 presents the parameters of each class.

Class	Distribution
In Control (IC)	$\mathbf{X}\sim N(\boldsymbol{\mu},\boldsymbol{\Sigma})$
Out-of-Control (OC)	$\mathbf{X}\sim N(\boldsymbol{\mu},c\times\boldsymbol{\Sigma})$

Table 1. Classes parameters

However, fixing the different distributions laws is not sufficient for taking decisions about the process state (in or out-of-control). Indeed, the definition of these parameters makes possible the application of a discriminant analysis in a bayesian network. So, when a new observation $\mathbf{x}$ is injected in the bayesian network, we will obtain an a posteriori probability $P(IC|\mathbf{x})$ that this observation belongs to the IC class and an a posteriori probability $P(OC|\mathbf{x})$ that this observation belongs to the OC class (with $P(IC|\mathbf{x}) + P(OC|\mathbf{x}) = 1$). In order to decide about the state of the process, one has to fix a false alarm rate α in addition to these two previous calculated probabilities. As in the case of multivariate control charts, we have to fix a false alarm rate. Moreover, the coefficient c will play an important role in the probabilities computations. In the next section, we will see how to set the different parameters to obtain the equivalency between the bayesian network and the multivariate control charts.

4.1 Equivalency proof

As in the case of the multivariate control charts (T^2 or MEWMA), we will fix a threshold (limit) on the a posteriori probabilities allowing to take decisions on the process: if, for a given observation, the a posteriori probability to be out-of-control ($P(OC|\mathbf{x})$) is greater than the a priori probability to be out-of-control ($P(OC)$), then this observation is out-of-control. This rule can be rewritten as: "process out-of-control if $P(OC|\mathbf{x})>P(OC)$", or equivalently "process in control if $P(IC|\mathbf{x})>P(IC)$". The objective of the following developments is to define c in order to obtain the equivalency between the bayesian network and the multivariate control charts.

We want to keep the following decision rule:

$$\mathbf{x} \in \text{IC, if} \quad T^2 < CL \tag{19}$$

with this decision rule:

$$\mathbf{x} \in \text{IC, if} \quad P(IC|\mathbf{x}) > P(IC) \tag{20}$$

We develop the second decision rule:

$$\begin{aligned} & P(IC|\mathbf{x}) > P(IC) \\ & P(IC|\mathbf{x}) > (P(IC))(P(IC|\mathbf{x}) + P(OC|\mathbf{x})) \\ & P(IC|\mathbf{x}) > (P(IC))P(IC|\mathbf{x}) + (P(IC))P(OC|\mathbf{x}) \\ & P(IC|\mathbf{x}) > \left(\frac{P(IC)}{1-P(IC)}\right)P(OC|\mathbf{x}) \\ & P(IC|\mathbf{x}) > \left(\frac{P(IC)}{P(OC)}\right)P(OC|\mathbf{x}) \end{aligned} \tag{21}$$

But, the Bayes law gives:

$$P(IC|\mathbf{x}) = \frac{P(IC)P(\mathbf{x}|IC)}{P(\mathbf{x})} \tag{22}$$

And

$$P(OC|\mathbf{x}) = \frac{P(OC)P(\mathbf{x}|OC)}{P(\mathbf{x})} \tag{23}$$

So, we obtain:

$$\begin{aligned} & \frac{P(IC)P(\mathbf{x}|IC)}{P(\mathbf{x})} > \left(\frac{P(IC)}{P(OC)}\right)\frac{P(OC)P(\mathbf{x}|OC)}{P(\mathbf{x})} \\ & \left(\frac{P(IC)}{P(OC)}\right)P(\mathbf{x}|IC) > \left(\frac{P(IC)}{P(OC)}\right)P(\mathbf{x}|OC) \\ & P(\mathbf{x}|IC) > P(\mathbf{x}|OC) \end{aligned} \tag{24}$$

In the case of a discriminant analysis with k classes C_i, the conditional probabilities are computed with equation (25), where ϕ represents the probability density function of the multivariate Gaussian distribution of the class.

$$P(\mathbf{x}|C_i)=\frac{\phi(\mathbf{x}|C_i)}{\sum_{j=1}^{k}P(C_j)\phi(\mathbf{x}|C_j)} \tag{25}$$

So, equation (24) can be written as:

$$\phi(\mathbf{x}|IC)>\phi(\mathbf{x}|OC) \tag{26}$$

We recall that the probability density function of a multivariate Gaussian distribution of dimension p, of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, of an observation $\mathbf{x}$ is given by:

$$\phi(\mathbf{x})=\frac{\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \tag{27}$$

If the law parameters are $\boldsymbol{\mu}$ and $c\times\boldsymbol{\Sigma}$, then the density function becomes:

$$\phi(\mathbf{x})=\frac{\exp\left(-\frac{1}{2c}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}c^{p/2}} \tag{28}$$

In identifying the expression $(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$ as the T^2 of the observation $\mathbf{x}$, we can write:

$$\begin{gathered}\phi(\mathbf{x}|IC)>\phi(\mathbf{x}|OC)\\ \frac{\exp\left(-\frac{T^2}{2}\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}}>\frac{\exp\left(-\frac{T^2}{2c}\right)}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}c^{p/2}}\\ \exp\left(-\frac{T^2}{2}\right)>\frac{\exp\left(-\frac{T^2}{2c}\right)}{c^{p/2}}\\ -\frac{T^2}{2}>-\frac{T^2}{2c}-\frac{p\ln(c)}{2}\\ T^2<\frac{p\ln(c)}{1-\frac{1}{c}}\end{gathered} \tag{29}$$

However, we search the value(s) of c allowing the equivalency with the control chart decision rule: $\mathbf{x} \in$ IC, if $T^2<CL$. So, we obtain the following equation for c:

$$\frac{p\ln(c)}{1-\frac{1}{c}}=CL \tag{30}$$

Or, equivalently:

$$1-c+\frac{pc}{CL}\ln(c)=0 \tag{31}$$

Equation (31) admits two solutions: c=1 (not acceptable) and a second solution (numerically computable) which depends of p and α. With the coefficient c correctly computed, we obtain the equivalence between the bayesian network and the multivariate control charts. We precise that, as univariate charts are simply a particular case of multivariate control chart, the proof given is also available for univariate control charts. In order to demonstrate the proposed approach, we illustrate it on a simple system with two variables.

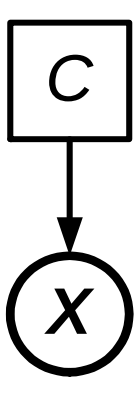
4.2 Detection with bayesian network

We will study a T^2 control chart and a MEWMA control chart (with λ=0.1) modelized by bayesian networks. We choose a false alarm rate α=1%. When the system is in-control, it follows a multivariate Gaussian distribution with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ such as:

$$\boldsymbol{\mu}=\begin{pmatrix}5 & 10\end{pmatrix} \tag{32}$$

$$\boldsymbol{\Sigma}=\begin{pmatrix}1 & 1.2\\ 1.2 & 2\end{pmatrix} \tag{33}$$

In order to monitor this process, we apply the proposed method of detection with bayesian network. So, for a T^2 control chart, we obtain the bayesian network of the figure 6. We have also given the conditional probability table of each node, and where c is equal to 95.28 (solution of equation (31) for α=1% and p=2).

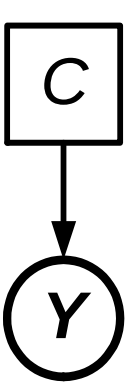


Class C	
IC	OC
1–α	α

C	**X**
IC	**X**~N (**μ**,**Σ**)
OC	**X**~N (**μ**,c×**Σ**)

Fig. 6. Bayesian Network similar to T^2 control chart

In the same way, we can also monitor the process with a MEWMA control chart modelized by the bayesian network of the figure 7, where c is equal to 90.29 (solution of equation (31) for α=1% and p=2 in the MEWMA case).



Class C	
IC	OC
$1-\alpha$	α

C	**Y**
IC	$\mathbf{Y} \sim N\left(\boldsymbol{\mu}, \left(\frac{\lambda}{2-\lambda}\right)\boldsymbol{\Sigma}\right)$
OC	$\mathbf{Y} \sim N\left(\boldsymbol{\mu}, c\times\left(\frac{\lambda}{2-\lambda}\right)\boldsymbol{\Sigma}\right)$

Fig. 7. Bayesian Network similar to MEWMA control chart

We have simulated this system on 30 observations. But, a fault has been introduced from observation 6 to 30. This fault is a mean step of magnitude 0.5 on the first variable. The figure 8 represents the decision taken at each instant respectively for the T^2 chart (left graphs) and for the MEWMA chart (right graphs). On this figure, upper graphs represent the computation of the statistical distance associated with the control chart (T^2 or T_t^2). The lower graphs give the a posteriori probability to be in control. The control limit is given on each graph, so we can view the limit on the bayesian network fixed to $1-\alpha$=99% .

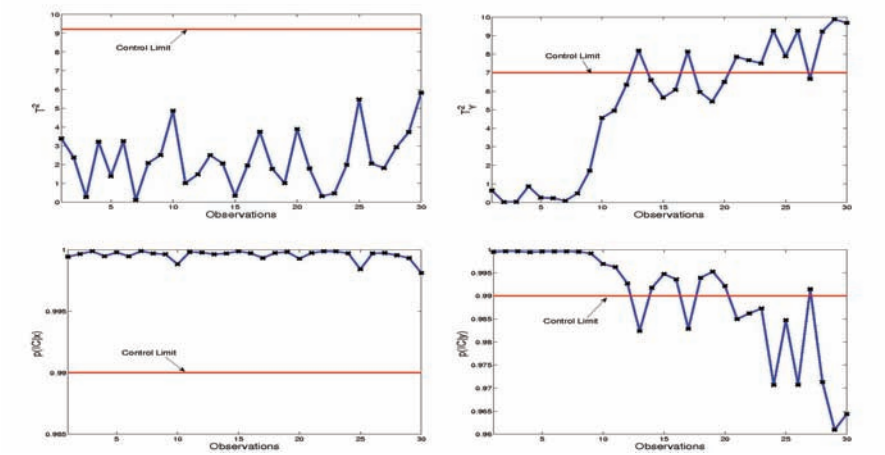


Fig. 8. Results of the T^2 and MEWMA chars, and their equivalency in Bayesian Network

On the figure 8, we can see that for each instant, the decision between a control chart and its modelization by bayesian network is equivalent.
We demonstrated that it is possible to have detection of faults in multivariate processes with bayesian networks and we proved that we can easily modelize multivariate control charts with them.

5. Conclusions and outlooks

In this chapter, we show that a bayesian network can be an efficient way to diagnose a fault in multivariate processes. We have selected two statistical fault detection techniques (the T^2 chart and the MEWMA chart) and we have demonstrated that these charts can be viewed as a discriminant analysis and so can be implemented in a simple bayesian network.
As the efficiency of bayesian network for the diagnosis of systems has already been demonstrated (Verron et al., 2006; Verron et al., 2007), the evident outlook of this work is the full study of the use of bayesian network in order to monitor and control a multivariate process (detection and diagnosis in the same network).

6. References

Bakshi, B. R. (1998). Multiscale PCA with application to multivariate statistical process monitoring. *AIChE Journal*, Vol. 44, No. 7, pp. 1596-1610.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*, Oxford University Press.

Cover, T. & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, Vol. 13, pp. 21-27.

Chiang, L.H.; Russell, E.L. & Braatz, R.D. (2001). *Fault detection and diagnosis in industrial systems*, Springer-Verlag, New York.

Chow, C. & Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, Vol. 14, pp. 462-467.

Domingos, P. & Pazzani, M.J. (1996). Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier, *In Proceedings of the Thirteen International Conference on Machine Learning*, pp. 105-112.

Duda, R.O.; Hart, P.E. & Stork, D.G. (2001). *Pattern Classification 2nd edition*, Wiley.

Friedman, N.; Geiger, D.& Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, Vol. 29, No. 2, pp 131-163.

Geiger, D. & Heckerman D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, Vol. 82, pp. 45-74.

Hawkins, D. M. (1991). Multivariate quality control based on regression-adjusted variables. *Technometrics*, Vol. 33, pp. 61-75.

Hotelling, H. (1947). Multivariate Quality Control. In *Techniques of Statistical Analysis*, C. Eisenhart; M.W. Hastay & W.A. Wallis, pp. 111-184, McGraw-Hill, New York.

Inza, I.; Larranaga, P.; Sierra, B.; Etxeberria, R.; Lozano, J. & Pena, J. (1999). Representing the behaviour of supervised classification learning algorithms by Bayesian networks. *Pattern Recognition Letters*, Vol. 20, pp. 1201-1209.

Jackson, E.J. (1985). Multivariate quality control. *Communication Statistics - Theory and Methods*, Vol. 14, No. 2, pp. 657-688.

Jensen, F.V. (1996). *An introduction to Bayesian Networks*, Taylor and Francis, London.

Kano, M.; Nagao, K.; Hasebe, S.; Hashimoto, I.; Ohno, H.; Strauss, R.; & Bakshi, B. (2002). Comparison of multivariate statistical process monitoring methods with applications to the Eastman challenge problem. *Computers and Chemical Engineering,* Vol. 26, No. 2, pp. 161-174.

Kononenko, I. (1991). Semi-naive bayesian classifier, *In Proceedings of the Sixth European Working Session on Learning*, pp. 206-219, Porto, Portugal, Springer-Verlag.

Kourti, T. & MacGregor, J.F. (1996). Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, Vol. 28, No. 4, pp. 409-428.

Langley, P; Iba, W. & Thompson, K. (1992). An analysis of bayesian classifiers, *In Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 223-228, San Jose, CA: AAAI Press.

Lowry, C.A.; Woodall, W.H.; Champ, C.W. & Rigdon, S.E. (1992). A multivariate exponentially weighted moving average control chart. *Technometrics*, Vol. 34, No. 1, pp46-53.

MacGregor, J. and Kourti, T. (1995). Statistical process control of multivariate processes. *Control Engineering Practice*, Vol. 3, No. 3, pp. 403-414.

Montgomery, D.C. (1997). *Introduction to Statistical Quality Control, Third Edition*, John Wiley and Sons.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers.

Pignatiello, J. & Runger, G. (1990). Comparisons of multivariate CUSUM charts. *Journal of Quality Technology*, Vol. 22, No. 3, pp. 173-186.

Sahami, M. (1996). Learning Limited Dependence Bayesian Classifiers, *In Proceedings of the Second International Conference on Knowledge Discovery in Databases*, pp. 335-338.

Shewhart, W.A. (1931). *Economic control of quality of manufactured product*, D. Van Nostrand Co., New York.

Tax, D. M. J. & Duin, R. P. W. (2001). Combining One-Class Classifiers. *Lecture Notes in Computer Science*, Vol. 2096, pp. 299-308.

Vapnik, V.N. (1995). The Nature of Statistical Learning Theory, Springer.

Verron S., Tiplica T., Kobi A. (2006). A new procedure based on mutual information for fault diagnosis of industrial systems, *in Workshop on Advanced Control and Diagnosis*.

Verron S., Tiplica T., Kobi A. (2007). Fault diagnosis of industrial systems with bayesian networks and mutual information, *In Proceedings of the Ninth International European Control Conference.*

19

A Hierarchical Bayesian Hidden Markov Model for Multi-Dimensional Discrete Data

Shigeru Motoi [1], Yohei Nakada [1], Toshie Misu [2], Tomohiro Yazaki [1]
Takashi Matsumoto [1] and Nobuyuki Yagi [2]
[1] *Faculty of Science and Engineering, Waseda University, Tokyo, Japan*
[2] *Science and Technical Research Laboratories, NHK (Japan Broadcasting Corporation), Tokyo, Japan*

1. Introduction

1.1 Motivation

A fundamental problem encountered in many fields is to model data o_t given a discrete time-series data sequence $y := (o_1, \cdots, o_T)$. This problem is found in diverse fields, such as control systems, robotics, event detection (Motoi et al., 2007), handwriting recognition (Yasuda et al., 2000 ; Funada et al., 2005), and protein structure prediction (Krogh et al., 2001 ; Tusnady & Simon, 1998 ; Kaburagi et al., 2007). The data o_t can often be a multi-dimensional variable exhibiting stochastic activity. A powerful tool for solving such problems is multi-dimensional discrete Hidden Markov Models (HMMs), and the effectiveness of this approach has been demonstrated in numerous studies (Motoi et al., 2007 ; Yasuda et al., 2000 ; Funada et al., 2005 ; Kaburagi et al., 2007). The hidden states of the HMMs are treated as hidden factors for emission of the observed data o_t. However, if redundant components having low dependencies on the hidden states are contained in the data o_t, these components often have a negative impact on the HMM performance. Overcoming this problem requires a method of quantifying the redundancy (state independence) of these components and/or reducing their influence.

In this chapter, we describe an extension of the HMM for these kinds of data sequences within the framework of a hierarchical Bayesian scheme. In this extended model, we introduce *commonality hyperparameters* to describe the degree of commonality of the emission probabilities among different hidden states (that is, hidden factors of the data o_t). Additionally, there is a one-to-one relationship between each hyperparameter and a component of the data o_t. This allows us to identify low-dependency components and to minimize their negative impact.

Like other Bayesian HMMs, the extended model requires complicated integrations in the learning and prediction processes, usually involving a posterior distribution. Analytic solutions of these integrations are often intractable or non-trivial due to their inherent

complexity. In this chapter, therefore, we also describe an implementation based on a Markov Chain Monte Carlo (MCMC) method (Scott, 2002).

1.2 Related work

In one detailed study, several feature selection methods were considered, such as discriminant feature analysis, principal component analysis, and the sequential search method (Nouza, 1996). In addition, that study also described a fast feature selection algorithm. Our approach described in this chapter may be regarded as a Bayesian feature selection scheme based on the dependencies of the hidden states.

There have been a number of studies examining Bayesian HMMs and their implementations, such as (Funada et al., 2005 ; Motoi et al., 2007 ; Huo et al., 1995 ; MacKay, 1997 ; Scott, 2002). Reference (Huo et al., 1995) describes a Maximum A Posteriori (MAP) estimation for Bayesian HMMs, and reference (MacKay, 1997) describes a Variational Bayesian method (so-called ensemble learning). In addition, references (Funada et al., 2005 ; Motoi et al., 2007 ; Scott, 2002) discuss Bayesian HMMs using MCMC. The model that we describe here is an extension of such Bayesian HMMs for discrete multi-dimensional data containing redundant components.

There is a well-known successful method to determine redundant components of multi-dimensional (input) data in the field of Bayesian Neural Networks (BNNs), called Automatic Relevance Determination (ARD) (MacKay, 1992 ; Neal, 1996 ; Qi et al., 2004 ; Tipping, 2000 ; Matsumoto et al., 2001 ; Nakada et al., 2005). ARD was first described in (MacKay, 1992); that method used a Laplace approximation. Reference (Neal, 1996) described another ARD using MCMC, and reference (Qi et al., 2004) discusses a variant based on Expectation Propagation. Several studies have also described extensions of the BNN using the ARD method, including, for example, the Relevance Vector Machine (Tipping, 2000) and BNNs for nonlinear time-series data (Matsumoto et al., 2001 ; Nakada et al., 2005). The structure of the extended HMM is completely different from that of such BNNs;nevertheless, the fundamental hierarchical Bayesian concepts show a number of underlying similarities.

2. Model specification

In this section, we describe the extended Bayesian HMM. The setting of hyperparameters is the principal difference between our extended model and the conventional Bayesian HMMs (see Sec. 2.5).

2.1 HMM Topology

The HMM structure depends on the particular topology employed and the number of states *N*. Topologies commonly employed include “ergodic” and “left-to-right”. Here we describe only the ergodic topology, since we employed that topology in our experiments, described later.

2.2 Data and hidden variables

In the HMM framework, we must consider the time-series data sequence (observation data sequence) $y := (o_1, \cdots, o_T)$ and the hidden variable sequence $z := (q_1, \cdots, q_T)$. The terms o_t and

q_t represent the time-series data and the hidden variable at time t, and T is the sequence length. The hidden variable q_t is a one-dimensional variable that takes finite values among the available N states (that is, $q_t \in \{1,\cdots,N\}$), whereas the data o_t is a multi-dimensional discrete variable defined by $o_t := (o_{1,t},\cdots,o_{D,t})$. Here, D represents the dimension of the data o_t, the variable $o_{k,t}$ the k-th component of o_t, and M_k the number of symbols for $o_{k,t}$ (in other words, $o_{k,t} \in \{1,\cdots,M_k\}$).

2.3 Observation model

Consider the complete parameter set θ of an HMM. The probability of the data y_t is

$$P(y \mid \theta) := \sum_z P(y \mid z,b)P(z \mid a,c), \quad \theta := (a,b,c), \tag{1}$$

Here,

$$P(y \mid z,b) := \prod_{t=1}^{T} P(o_t \mid q_t,b), \tag{2}$$

$$P(z \mid a,c) := P(q_1 \mid c)\prod_{t=2}^{T} P(q_t \mid q_{t-1},a). \tag{3}$$

The emission probability of the data o_t in (2) is

$$P(o_t \mid q_t,b) := \prod_{k=1}^{D} P(o_{k,t} \mid q_t,b_k), \tag{4}$$

where $b := (b_1,\cdots,b_D)$. The probability $P(o_{k,t} \mid q_t,b_k)$ in Eqn. (4) represents the emission probability of the k-th component $o_{k,t}$. It is defined as

$$P(o_{k,t} = j \mid q_t = i,b_k) := b_{k,ij}, \tag{5}$$

where $b_k := (b_{k,1},\cdots,b_{k,N})$, $b_{k,i} := (b_{k,i1},\cdots,b_{k,iM_k})$, $\sum_{j=1}^{M_k} b_{k,ij} = 1$, and $0 \le b_{k,ij} \le 1$.

The hidden variable transition probability and the initial hidden variable probability in Eqn. (3) are

$$P(q_t = j \mid q_{t-1} = i,a) := a_{ij}, \quad \theta > 1, \tag{6}$$

$$P(q_1 = i \mid c) := c_i, \tag{7}$$

Here, $a := (a_1, \cdots, a_N)$, $a_i := (a_{i1}, \cdots, a_{iN})$, $\sum_{j=1}^{N} a_{ij} = 1$, $0 \le a_{ij} \le 1$, $c := (c_1, \cdots, c_N)$, $\sum_{i=1}^{N} c_i = 1$, and $0 \le c_i \le 1$.

2.4 Prior distribution for parameters

Within a Bayesian framework, both the observation model (the likelihood function) and the prior distribution of the parameter set are defined. For the sake of simplicity, many Bayesian HMMs assume parameter independency in the prior distribution. That is to say:

$$P(\theta \mid \phi) = P(a \mid \alpha) P(b \mid \beta) P(c \mid \gamma), \tag{8}$$

$$P(a \mid \alpha) := \prod_{i=1}^{N} P(a_i \mid \alpha_i), \tag{9}$$

$$P(b \mid \beta) = \prod_{k=1}^{D} \prod_{i=1}^{N} P(b_{k,i} \mid \beta_{k,i}), \tag{10}$$

where

$$\phi := (\alpha, \beta, \gamma), \quad \alpha := (\alpha_1, \cdots, \alpha_N)$$
$$\beta := (\beta_1, \cdots, \beta_D), \quad \beta_k := (\beta_{k,1}, \cdots, \beta_{k,N}).$$

The prior distributions of a_i, $b_{k,i}$ and c in Eqns. (8)-(10) are also defined using the "natural conjugate" Dirichlet prior distribution:

$$P(a_i \mid \alpha_i) := \mathcal{D}(a_i; \alpha_i), \tag{11}$$

$$P(b_{k,i} \mid \beta_{k,i}) := \mathcal{D}(b_{k,i}; \beta_{k,i}), \tag{12}$$

$$P(c \mid \gamma) := \mathcal{D}(c; \gamma), \tag{13}$$

where $\mathcal{D}(\cdot; \chi)$ is the Dirichlet distribution with the parameter vector χ, and $\alpha_i := (\alpha_{i1}, \cdots, \alpha_{iN})$, $\alpha_{ij} > 0$, $\beta_{k,i} := (\beta_{k,i1}, \cdots, \beta_{k,iN})$, $\beta_{k,ij} > 0$, $\gamma := (\gamma_1, \cdots, \gamma_N)$, $\gamma_i > 0$.

2.5 Settings for hyperparameter set

As in a number of conventional Bayesian HMMs, for example, (Funada et al., 2005 ; Huo et

al., 1995), all components of the hyperparameter vectors are fixed at 1.0, except for $\beta_{k,i}$.[1] With our approach on the other hand, we consider a reparameterization of the hyperparameter vectors $\{\beta_{k,i}\}_{i=1}^{N}$, and the prior distribution of the reparameterized hyperparameters in order to identify components having low dependency on the states (redundant components).

A. Reparameterization of $\beta_{k,i}$

We define the hyperparameter vector $\beta_{k,i}$ as:

$$\beta_{k,i} := \lambda_k \eta_k, \quad @=1,\cdots,N, \tag{14}$$

where $\lambda_k (\in \mathbf{R}) > 0$, $\eta_k := (\eta_{k,1},\cdots,\eta_{k,M_k})$, $0 < \eta_{k,i} < 1$, and $\sum_{i=1}^{M_k} \eta_{k,i} = 1$. Here, λ_k is the *commonality hyperparameter* describing the degree of commonality for the emission probabilities of $\{o_{k,t}\}_{t=1}^{T}$: $P(o_{k,t} \mid q_t, b_k)$ among different hidden states.[2] The hyperprameter η_k is a *common shape hyperparameter* that described the average shape of the emission probabilities $P(o_{k,t} \mid q_t, b_k)$ for different hidden states.

Here, we examine the effect of the commonality hyperparameter λ_k on the emission probability $b_{k,i}$. The shapes of the prior distribution (10) for various values of λ_k are shown in Figure 1. Fig. 1 (c) shows a case where λ_k is large. Here, the parameter vectors $\{b_{k,i}\}_{i=1}^{N}$, exhibit only small differences, i.e., $b_{k,1} \approx b_{k,2} \approx \cdots \approx b_{k,M_k} \approx \eta_k$, meaning that there is low dependency of $\{o_{k,t}\}_{t=1}^{T}$ on the states. For smaller λ_k on the other hand (Fig. 1 (a) or (b)), the diversity of $\{b_{k,i}\}_{i=1}^{N}$ among each state is greater; in other words, the dependency of $\{o_{k,t}\}_{t=1}^{T}$ on the states is not low.

B. Prior distribution for λ_k and η_k

Here we describe the prior distribution of the hyperparameters λ_k and η_k used for learning these hyperparameters in a Bayesian learning method described later.

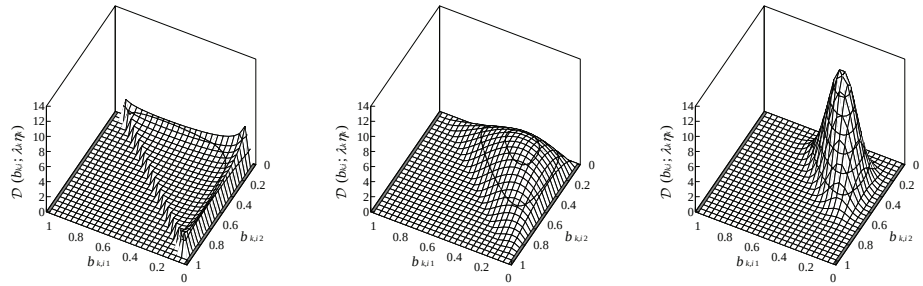
The commonality hyperparameter λ_k has no well-known "natural conjugate" prior distribution. Therefore, the prior distribution for λ_k is defined using only information in the

---

[1] This basic setting of the Dirichlet prior distribution makes it equivalent to a non-informative uniform prior distribution.

[2] The diversity of $P(o_{k,t} \mid q_t, b_k)$ among the states corresponds to that of $\{b_{k,i}\}_{i=1}^{N}$ because the emission probability of $o_{k,t}$: $P(o_{k,t} \mid q_t, b_k)$ is defined by using $\{b_{k,i}\}_{i=1}^{N}$, as shown in equation (5).

range $\lambda_k \in (0,\infty)$. Although there are a number of alternative prior distributions for a positive continuous variable (for example, the log-normal prior distribution), the prior distribution of λ_k is given by the following gamma prior distribution:



(a) λ_k = 2. (b) λ_k = 6. (c) λ_k = 18.

Fig. 1. Dirichlet prior distribution for $b_{k,i}$, for various values of the commonality hyperparameter λ_k. The parameters $\{b_{k,i}\}_{i=1}^N$ are 3D variables $b_{k,i} = (b_{k,i1}, b_{k,i2}, b_{k,i3})$, and the common shape hyperparameter η_k is constant, $\eta_k = (0.3, 0.3, 0.4)$. The component $b_{k,i3}$ is omitted because it can be determined from $b_{k,i3} = 1 - b_{k,i1} - b_{k,i2}$. This figure clearly shows that, for larger λ_k, the parameters $\{b_{k,i}\}_{i=1}^N$ concentrate more around the average η_k.

$$P(\lambda_k) := \mathcal{G}(\lambda_k; \kappa, \omega), \tag{15}$$

where $\mathcal{G}(\cdot; \kappa, \omega)$, is the gamma distribution having shape parameter κ and scale parameter ω.[3] These hyperhyperparameters are set to $\kappa = 1.0$ and $\omega = 100$ in the experiments described in Sec. 4, which allows λ_k to be widely distributed within in its available range.

There is also no known "natural conjugate" prior distribution for η_k. However, there are a limited number of options for the prior distribution because of the constraints of η_k, namely, $\sum_{i=1}^{M_k} \eta_{k,i} = 1$ and $0 < \eta_{k,i} < 1$. Therefore, we use the Dirichlet distribution as the prior distribution for η_k:

$$P(\eta_k) := \mathcal{D}(\eta_k; \eta_0), \tag{16}$$

where η_0 denotes the hyperhyperparameter vector. By considering a non-informative

[3] The gamma distribution is defined as $\mathcal{G}(x; \kappa, \omega) := \dfrac{x^{\kappa-1} \exp\left(-\omega^{-1} x\right)}{\omega^{\kappa} \Gamma(\phi)}$, where $\Gamma(\cdot)$ is the gamma function.

setting for η_k, the vector η_0 is set to $\eta_0 = (1.0, \cdots, 1.0)$ in the experiments described later.

Fig. 2 graphically summarizes the model specifications described in this section.

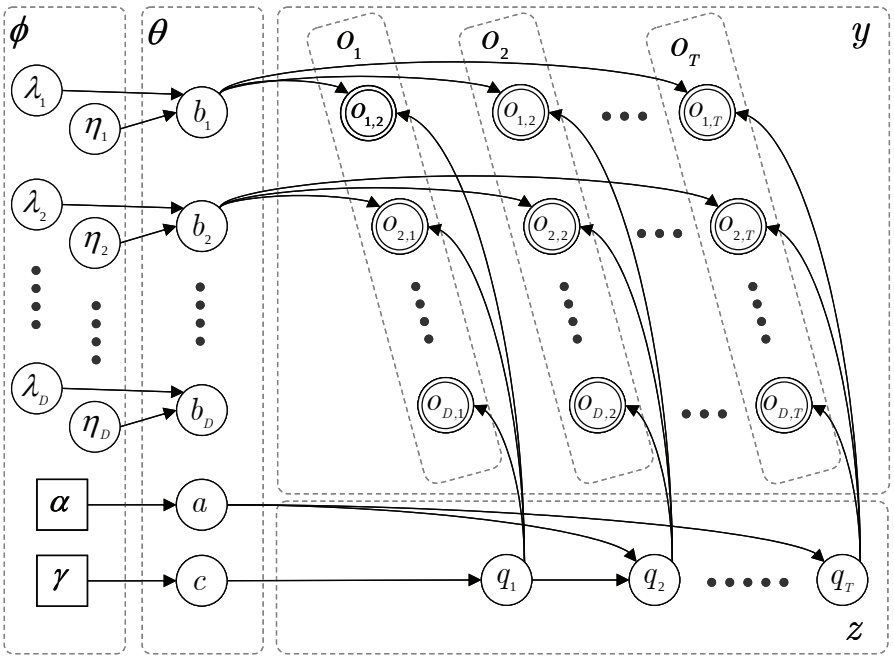


Fig. 2. Graphical representation of the model. The double circles are observable probabilistic variables, and the single circles are unobservable probabilistic variables. The squares are the fixed variables, the arrows probabilistic dependencies between variables, and the dashed lines groups of variables. Hyperhyperparameters and their dependencies are omitted for clarity.

3. Bayesian learning for the model

We define a training dataset Y as the set of time-series data sequences $\{y_l\}_{l=1}^{L}$, where L is the number of sequences and l is the index of the sequence. The goal of Bayesian learning is, given the training dataset Y and the above model, to evaluate the (joint) posterior distribution for θ and ϕ:

$$P(\theta, \phi \mid Y) = \sum_{Z} P(\theta, \phi, Z \mid Y), \tag{17}$$

where

$$P(\theta,\phi,Z\mid Y)= \\ @\ @\ @\ @\ \frac{P(Y\mid Z,\theta)P(Z\mid\theta)P(\theta\mid\phi)P(\phi)}{\sum_Z\int\int P(Y\mid Z,\theta)P(Z\mid\theta)P(\theta\mid\phi)P(\phi)d\theta d\phi}, \tag{18}$$

and Z is the set of hidden variable sequences $\{z_l\}_{l=1}^{L}$, corresponding to the dataset Y.

3.1 Implementation with MCMC

The integrations in equation (18) have no closed-form analytical solution, because of their complexity. Monte Carlo methods can generate samples from the posterior distribution (17), and we therefore adopt this approach. [4]Once the samples $\{(\theta^{(r)},\phi^{(r)})\}_{r=1}^{R}$ are generated, it is an easy matter to approximate the posterior distribution (17) with

$$P(\theta,\phi\mid Y)\approx\frac{1}{R}\sum_{r=1}^{R}\delta\left((\theta,\phi)-(\theta^{(r)},\phi^{(r)})\right), \tag{19}$$

where $\delta(\cdot)$ is the Dirac delta function, R is the number of samples, and r is the index of the sample. Fig. 3 summarizes the procedure used in our implementation.

3.2 Model evaluation

We introduce a fitness score as a metric to evaluate the degree of fitness between a set of test data sequences Y_{NEW} and the trained model:

$$\text{Score}(Y_{NEW}):=\log P(Y_{NEW}\mid Y), \tag{20}$$

Here, $P(Y_{NEW}\mid Y)$ is the (conditional) marginal likelihood, that is, the likelihood function $P(Y_{NEW}\mid\theta)$ averaged over the posterior distribution $P(\theta,\phi\mid Y)$:

$$P(Y_{NEW}\mid Y)=\int\int P(Y_{NEW}\mid\theta)P(\theta,\phi\mid Y)d\theta d\phi. \tag{21}$$

Using the Monte Carlo approximation (19), we can approximate this marginal likelihood as

$$P(Y_{NEW}\mid Y)\approx\frac{1}{R}\sum_{r=1}^{R}P(Y_{NEW}\mid\theta^{(r)}). \tag{22}$$

[4] Specifically, we consider the joint posterior distribution (18) for generating the samples using an MCMC technique based on that in (Scott, 2002). By discarding samples of Z after taking the samples of (θ,ϕ,Z) from the joint posterior distribution (18), it becomes relatively straightforward to obtain samples of θ and ϕ from the posterior distribution (17).

Implementation using MCMC methods

(a) Initialization step:
Initialize $\theta^{(0)}$ and $\phi^{(0)}$ by sampling. $\psi^{(0)}$ is generated from the prior distribution, whereas $\theta^{(0)}$ is generated uniformly within the range of θ.

(b) MCMC step:
For g = 1 to G, repeat the following:
(i) Generate the g-th sample of Z by with the forward-backward sampling method (Scott, 2002).
(ii) Generate the g-th sample of θ using the Gibbs sampling method (Scott, 2002 ; Geman & Geman, 1984).
(iii) Generate the g-th sample of ϕ using the Metropolis-Hastings method (Hastings, 1970). [5]

(c) Selection step:
For the Monte Carlo approximation (19), select the sample set $\{\theta^{(r)}\}_{r=1}^{R}$ from $\{\theta^{(g)}\}_{g=1}^{G}$. [6]

Fig. 3. MCMC implementation.

4. Experiments

4.1 Artificial dataset experiment

We conducted an experiment using artificial datasets to evaluate our extended model. These datasets contain state-independent variables serving as redundant components.

A. Target HMM

In this experiment, we used multi-dimensional data sequences, each data component having 5 symbols. We generated these sequences from a 5-state ergodic HMM in which the hidden variable transition parameter a^* and the initial hidden variable parameter π^* were retained:

[5] In actual implementation, a well-known strategy to improve the acceptance rates is to apply the Metropolis-Hastings method separately to each hyperparameter λ_k and hyperparameter vector η_k. We use proposal distributions designed on the basis of information from the model, because this approach also improves the efficiency of the Metropolis-Hastings method in many cases.
We show details of the designed proposal distributions in the appendix.

[6] In the MCMC method, it is usually necessary to discard the initial samples. In the experiments described in Sec. 4, we generated 1000 samples in the MCMC step (b) (G = 1000), and we used the last 500 samples for the Monte Carlo approximation (R = 500).

$$a^* = \begin{bmatrix} 0.70 & 0.10 & 0.05 & 0.05 & 0.10 \\ 0.10 & 0.70 & 0.10 & 0.05 & 0.05 \\ 0.05 & 0.10 & 0.70 & 0.10 & 0.05 \\ 0.05 & 0.05 & 0.10 & 0.70 & 0.10 \\ 0.10 & 0.05 & 0.05 & 0.10 & 0.70 \end{bmatrix}, \quad \pi^* = \begin{bmatrix} 0.20 \\ 0.20 \\ 0.20 \\ 0.20 \\ 0.20 \end{bmatrix}^T.$$

The column and row numbers of the matrix representing parameter a^* are the next and current values of the hidden variable. The column number in the matrix representing parameter π^* is equivalent to the index of the initial hidden variable. We explain the emission probabilities of the target HMM in detail in the following.

B. State-dependent and state-independent components

In this experiment, we considered the following two probability matrices, b^*_{DEP} and b^*_{IND}, for the emission probability parameter of the k-th data component, b^*_k:

$$b^*_{DEP} = \begin{bmatrix} 0.50 & 0.20 & 0.05 & 0.05 & 0.20 \\ 0.20 & 0.50 & 0.20 & 0.05 & 0.05 \\ 0.05 & 0.20 & 0.50 & 0.20 & 0.05 \\ 0.05 & 0.05 & 0.20 & 0.50 & 0.20 \\ 0.20 & 0.05 & 0.05 & 0.20 & 0.50 \end{bmatrix}, \tag{23}$$

$$b^*_{IND} = \begin{bmatrix} 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \\ 0.20 & 0.20 & 0.20 & 0.20 & 0.20 \end{bmatrix}. \tag{24}$$

Here, the column number represents the index of the hidden state, and the row number represents the index of the observable symbols. It should be noted that the matrix b^*_{IND} contains identical probability vectors in each column (state); in other words, the components with b^*_{IND} state-independent, whereas the components with b^*_{DEP} are state-dependent. Using these matrices, we considered the following 5 cases with different numbers of components:

(i) $b^*_1 = b^*_2 = b^*_{DEP}$,

(ii) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_{IND}$,

(iii) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_4 = b^*_{IND}$,

(iv) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = b^*_4 = b^*_5 = b^*_{IND}$,

(v) $b^*_1 = b^*_2 = b^*_{DEP}$ and $b^*_3 = \cdots = b^*_6 = b^*_{IND}$.

The first 2 components in each case are state-dependent.

C. Model settings

In each of the cases described above, we trained and tested the extended model using various datasets containing 10 independent sequences (T = 100) generated from the target HMM. We also trained and tested a conventional Bayesian HMM with fixed hyperparameters with the same datasets for comparison.[7] We trained the conventional model using an MCMC implementation based on (Scott, 2002). In our extended model and in the conventional model, we set the number of hidden states to N = 5, i.e. the same number of hidden states as that of the target HMM.

D. Results

Figure 4 shows the averaged differences between the fitness score (20) of the extended model and that of the conventional model. When all components were state-dependent (case (i)), the extended model performed slightly worse than the conventional model. When the training dataset contained state-independent components (cases (ii) to (v)), however, the extended model performed better than the conventional one, as indicated by the higher averaged score differences as the number of state-independent components increased. This result demonstrates that the extended model is robust against state-independent components.

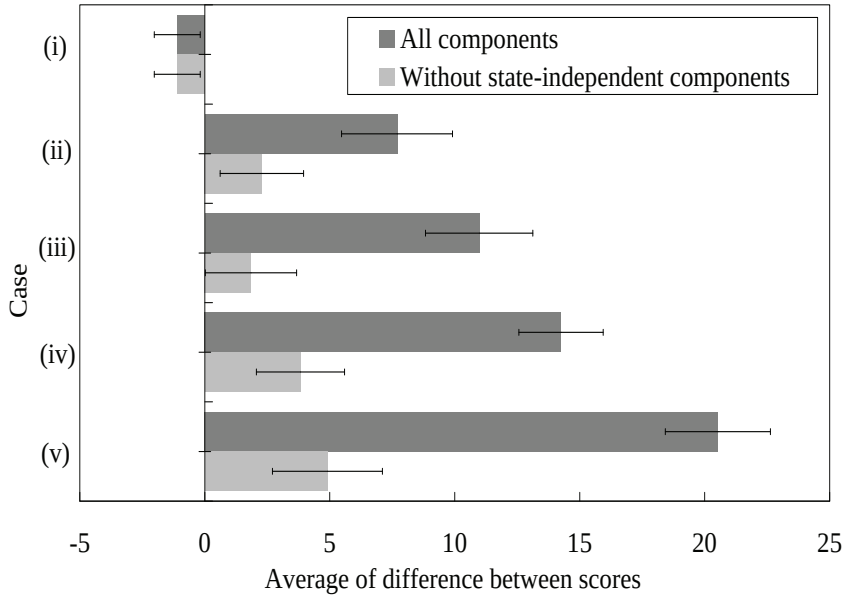


Fig. 4. The differences between scores (extended model score minus conventional model score) for all components of the test datasets. Differences based on the scores for the first two components (state-dependent components) of the test datasets are also shown. Differences of 10 independent trials were averaged. The error bars indicate the standard error.

[7] Each component of the hyperparameters is fixed at 1.0.

4.2 Soccer dataset experiment

We used real-world datasets with an additional irrelevant component to verify the performance of our extended model. This experiment was designed to demonstrate the ability of our model to discriminate the irrelevant component by Bayesian modeling with the commonality hyperparameter λ_k . This hyperparameter is closely related to the redundancy (state-independency) of a particular data component $o_{k,t}$.

This is a preliminary experiment a project involving event detection of Bayesian modeling for soccer games (Motoi et al., 2007).

A. Target data sequence

In our previous work (Motoi et al., 2007), the original dataset consisted of data sequences for 5 half-games of soccer. Each sequence was composed of 27-dimensional time-series data obtained from the position sequences of players. These positions were automatically extracted from video images by tracking the players using a method based on that in (Misu et al., 2005 ; Misu et al., 2002).

We used the sequence for only 1 half-game (length T = 2390) for the sake of simplicity. This sequence contained only 6 selected components and 1 additional component.

B. Selected and additional components

We used the following 6 selected variables for modeling: (a) the center of all players in the x direction; (b) the center of all players in the y direction; (c) the center of the left team players in the x direction; (d) the center of the left team players in the y direction; (e) the center of the right team players in the x direction; and (f) the center of the right team players in the y direction. We also added another variable to the target data sequence as the irrelevant component: (g) the x center of all the players in another half-game. The x and y directions correspond to the long axis and short axis of the playing field, respectively.

C. Model settings

In modeling the target data, we discretized all components in the extended model into 10 symbols (in other words, M_k = 10 for all components). We also set the number of hidden states to N = 10. Two examples of the discretized data components are shown in Figure 5.

D. Results

Boxplots of the commonality hyperparameter samples generated from the posterior distribution are shown in Figure 6 (18). The irrelevant component (g) has the largest hyperparameter λ_k , suggesting the possibility of discriminating irrelevant components by using the hyperparameters $\{\lambda_k\}$.

5. Application to real event detection

The results described in the previous section demonstrated the capability of our extended model in an event detection problem in soccer games (Motoi et al., 2007). In this section, we apply the extended model to event detection in sports videos. Our goal here is to detect

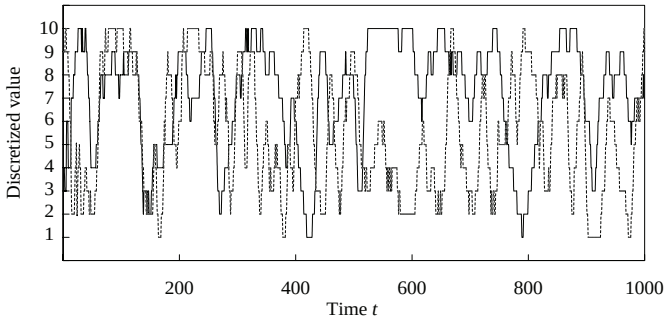


Fig. 5. Trajectories of the (discretized) variables (a) and (b), plotted in the range t = 1 to 1000 for clarity. The solid line is (a), and the dotted line is (b).

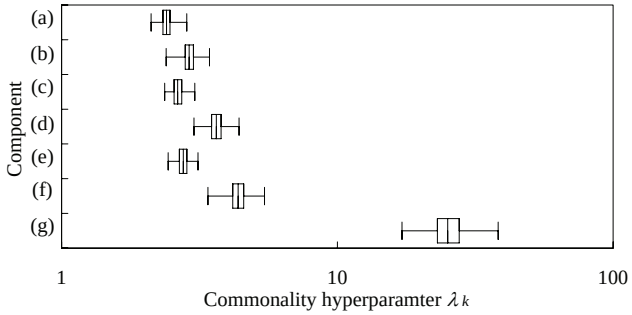


Fig. 6. Boxplots of the commonality hyperparameters $\{\lambda_k\}$ (500 samples) in Experiment 4.2. The smallest sample, lower quartile, median, upper quartile, and largest sample are shown for each λ_k .

target events from data sequences. Such events include kick offs, corner kicks, free kicks, throw ins, and goal kicks. Details of the data sequences are described in the following.

5.1 Modeling with a given data sequence

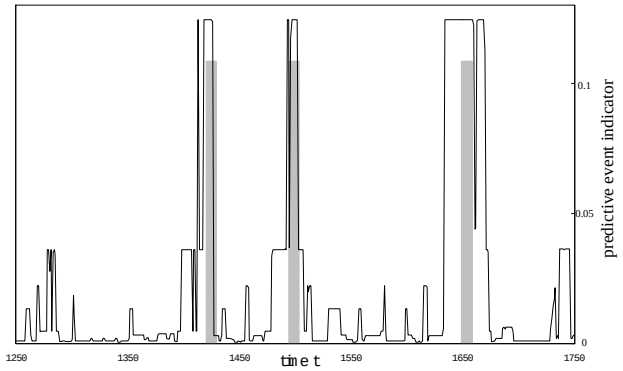
In this modeling, the raw dataset consisted of the positions of all players, which were automatically extracted from videos of 7 half games. Forty components associated with each target event were contained in the given data sequence.[8] We trained both the conventional and extended HMMs using the sequences for the 40 associated component in all 7 half games.

5.1.1 Demonstration

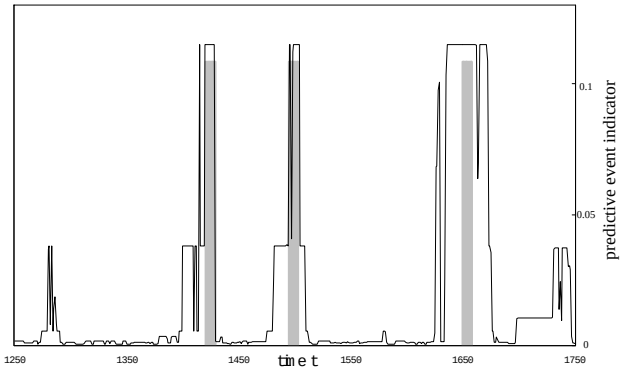
In this section, we show the predicted results for a corner kick event in another half game. This half game was independent from the 7 half games used to train the HMMs. Examples

[8]First, 1065 candidate components were generated from players' positions. We then selected the 40 associated components using standard information-based criteria showing the degree of the association with each event.

of the predicted results with our extended model and the conventional in (Motoi et al., 2007) are shown in Fig. 7. Actual events are indicated in gray. These results show that the conventional model gives more false alerts compared with the extended model, indicating the capability of the extended model to reduce the negative influence of redundant components in the 40 given components.



(a) Extended model.



(b) Conventional model.

Fig. 7. Predicted results for corner kick event. The range t = 1250 to 1750 contains 3 of the 4 target events in this half game. The regions of actual events are shown in gray.

6. Conclusions

In this chapter, we have described an extended Bayesian HMM for multidimensional discrete data sequences including redundant components. For the extended model, we also described an implementation of Bayesian learning based on a Markov chain Monte Carlo scheme. We evaluated the performance of the extended model with this implementation using two example datasets. We also demonstrated its application to an event detection problem with 40-dimensional data sequences extracted from videos of actual soccer games.

Our results showed that the extended Bayesian HMM has reasonable performance in the presence of redundant components in the data.

7. Acknowledgements

The authors greatly appreciate insightful comments from Prof. A. Doucet. The authors also thank A. Matsui, S. Clippingdale, I. Yamada, and M. Takahashi of NHK for their help and advice. T. Kaburagi, H. Sasaki, and J. Maruyama at Waseda University also greatly contributed to this study.

Appendix: Proposal distributions

The proposal distribution can be any probability distribution so long as certain conditions are satisfied. The design of the proposal distribution, however, strongly affects the efficiency. When applying the Metropolis-Hastings method to each variable separately, a promising approach is to employ the full conditional (posterior) distribution as the proposal distribution $\mathrm{Q}(\cdot)$. [9] However, it is difficult to use the full conditional distributions of λ_k and η_k as their proposal distributions in the model, because these distributions do not belong to any standard families of probability density functions having known direct sampling methods. Therefore, we use proposal distributions designed based on information from the full conditional distributions.

A. Proposal distribution of λ_k

The full conditional distribution of λ_k is

$$P(\lambda_k \mid Y, Z, \theta, \{\eta_{k'}\}, \{\lambda_{k'}\}_{k' \neq k}) = P(\lambda_k \mid b_k, \eta_k). \tag{25}$$

Applying the log-normal distribution $\mathcal{LN}(\cdot)$, the full conditional distribution (25) can be approximated by:

$$P(\lambda_k \mid b_k, \eta_k) \approx \mathcal{LN}\left(\lambda_k ; \mu\left(\lambda_k^{(g-1)}\right), v\left(\lambda_k^{(g-1)}\right)\right). \tag{26}$$

Here,

$$\mu(\lambda_k) := \log \lambda_k + l_k''(\lambda_k)^{-1} l_k'(\lambda_k), \quad v(\lambda_k) := l_k''(\lambda)^{-1}$$
$$l_k(\lambda_k) := \log P(\log \lambda_k \mid b_k, \eta_k) = \log P(\lambda_k \mid b_k, \eta_k) + \log \lambda_k,$$

$l_k'(\cdot)$ is the first-order derivative of $l_k(\cdot)$, and $l_k''(\cdot)$ is its second-order derivative. However, the approximation (26) is not valid when the previous sample $\log \lambda_k^{(g-1)}$ is far from the peak

[9] In this scenario, the Metropolis-Hastings algorithm is completely "rejection-less"; in other words, it is identical to Gibbs sampling.

of $P(\log\lambda_k \mid b_k, \eta_k) = \lambda_k P(\lambda_k \mid b_k, \eta_k)$. In a number of preliminary numerical experiments, this proposal distribution (26) showed low acceptance rate. Slightly expanding the logarithm variance of the proposal distribution is a simple way to improve the low acceptance rate in such cases. Thus

$$Q(\lambda_k; \cdot) := \mathcal{LN}\left(\lambda_k; \mu\left(\lambda_k^{(g-1)}\right), (1+\varepsilon)v\left(\lambda_k^{(g-1)}\right)\right), \tag{27}$$

where $\varepsilon(\geq 0)$ is a user-settable variable. In the experiments described in this chapter, we used the proposal distribution (27) with $\varepsilon = 0.2$. This gave reasonable and stable performance in our preliminary experiments.

B. Proposal distribution of η_k

The full conditional distribution of η_k is

$$P(\eta_k \mid Y, Z, \theta, \{\eta_{k'}\}_{k' \neq k}, \{\lambda_{k'}\}) = P(\eta_k \mid b_k, \lambda_k). \tag{28}$$

It is difficult to approximate the distribution (28) itself with basic methods. Therefore, we consider only a rough approximation of the center of the distribution (28) in this study.
When the parameter b_k is given, one of the simplest estimators for the common (average) shape of $\{b_{k,i}\}$ is $\overline{\eta}_k(b_k) = \frac{1}{N}\sum_{i=1}^{N} b_{k,i}$. We assume that the center of the distribution (28) can be roughly approximated by this estimator $\overline{\eta}_k(b_k)$. In view of this assumption and the simplicity of the implementation, we use the Dirichlet proposal distribution centered on $\overline{\eta}_k(b_k)$:

$$Q(\eta_k; \cdot) := \mathcal{D}(\eta_k; v\overline{\eta}_k(b_k)). \tag{29}$$

Here, $v(>0)$ is a user-settable variable. In this study, we set $v = 100$, which resulted in reasonable performance in a number of preliminary numerical experiments.

8. References

Funada, A.; Sasaki, H.; Nakada, Y.; Matsumoto, T. (2005). Monte Carlo HMM for on-line handwriting recognition. *Proc. 15th Jpn. Neural Netw. Soc. Ann. Conv.* (*JNNS2005*), pp. 137–138, Sep. 2005 [in Japanese].

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721-741, Nov. 1984.

Günter, S. & Bunke, H. (2003). Fast feature selection in an HMM-based multiple classifier system for handwriting recognition. *Proc. 25th DAGM Symp. Pattern Recognit.* (*DAGM2003*), pp. 289–296, Sep. 2003.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, vol. 57, no. 1, pp. 97-109, Apr. 1970.

Huo, Q.; Chan, C.; Lee, C-H. (1995). Bayesian adaptive learning of the parameters of hidden Markov model for speech recognition. *IEEE Trans. Speech Audio Process.*, vol. 3, no. 5, pp. 334–345, Sep. 1995.

Kaburagi, T.; Muramatsu, D.; Matsumoto, T. (2007). Transmembrane protein structure predictions with hydropathy index/charge two-dimensional trajectories of stochastic dynamical systems. *J. Bioinform. Comput. Biol.*, 2007 (in press).

Krogh, A.; Larsson, B.; Heijne, G.; Sonnhammer, E. (2001). Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. *J. Mol. Biol.*, vol. 305, no. 3, pp. 567–580, Jan. 2001.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Comput.*, vol. 4, no. 3, pp. 448–472, May 1992.

MacKay, D. J. C. (1992). *Bayesian modeling and neural networks*, PhD thesis, Dept. of Computation and Neural Systems, CalTech, 1992.

MacKay, D. J. C. (1997). Ensemble learning for hidden Markov models. Technical report, Cavendish Laboratory, University of Cambridge, 1997.(Available from http://wol.ra.phy.cam.ac.uk/mackay/)

Matsumoto, T.; Nakajima, Y.; Saito, M.; Sugi, J.; Hamagishi, H. (2001). Reconstructions and predictions of nonlinear dynamical systems: a hierarchical Bayesian approach. *IEEE Trans. Signal Process.*, vol. 49, no. 9, pp. 2138–2155, Sep. 2001.

Misu, T.; Naemura, M.; Zheng, W.; Izumi, Y. and Fukui, K. (2002). Robust Tracking of Soccer Players Based on Data Fusion. *Proc. 16th Int. Conf. Pattern Recognit.* (*ICPR2002*), vol. 1, pp. 556-561, Aug. 2002

Misu, T.; Takahashi, M.; Gohshi, S.; Tadenuma, M.; Fujita, Y. and Yagi, N. (2005). Visualization of offside lines based on realtime video processing. *IEICE Trans.*, vol. J88-D-II, no. 8, pp. 1681-1692, Aug. 2005 [in Japanese].

Motoi, S.; Misu, T.; Nakada, Y.; Matsumoto, T.; Yagi, N. (2007). Bayesian hidden Markov model approach for events detection in sports movie. *Spec. Interest Group Notes of Inf. Process. Soc. Jpn.* (*IPSJ SIG Notes*), vol. 2007, no. 1, pp. 133-139, Jan. 2007 [in Japanese].

Nakada, Y.; Matsumoto, T.; Kurihara, T.; Yosui, K. (2005). Bayesian reconstructions and predictions of nonlinear dynamical systems via the hybrid Monte Carlo scheme. *Signal Process.*, vol. 85, no. 1, pp. 129–145, Jan. 2005.

Neal, R. M. (1996). *Bayesian learning for neural networks*, Lecture Notes in Statistics 118, Springer-Verlag, New York, USA, Aug. 1996.

Nouza, J. (1996). Feature selection methods for hidden Markov model-based speech recognition. *Proc. 13th Int. Conf. Pattern Recognit.* (*ICPR96*), vol. 2, pp. 186– 190, Aug. 1996.

Qi, Y.; Minka, T.; Picard, R.; Ghahramani, Z. (2004). Predictive automatic relevance determination by expectation propagation. *Proc. 21st Int. Conf. Mach. Learn.* (*ICML2004*), pp. 85–92, Jul. 2004.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: recursive computing in the 21st century. *J. Am. Stat. Assoc.*, vol. 97, no. 457 pp. 337–351, Mar. 2002.

Tipping, M. E. (2000). The relevance vector machine. *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 652–658, Jun. 2000.

Tusnady, G.; Simon, I. (1998). Principles governing amino acid composition of integral membrane proteins: application to topology prediction. *J. Mol. Biol.*, vol. 283, no. 2, pp. 489–506, Oct. 1998.

Yasuda, H.; Takahashi, K.; Matsumoto, T. (2000). A discrete HMM for online handwriting recognition. *Int. J. Pattern Recognit. Artif. Intell.*, vol. 14, no. 5, pp. 675–688, Aug. 2000.

20

Development of Rough Terrain Mobile Robot using Connected Crawler -Derivation of sub-optimal number of crawler stages-

Sho Yokota[1], Yasuhiro Ohyama[1], Hiroshi Hashimoto[2],
Jin-Hua She[1], Hisato Kobayashi[3], Pierre Blazevic[4]
[1] *Tokyo University of Technology, Japan*
[2] *Advanced Institute of Industrial Technology, Japan*
[3] *Hosei University, Japan*
[4] *Université de Versailles, France*

1. Introduction

The application fields of autonomous mobile robots recently extend from indoor uses to outdoor uses. Rescue systems and planetary explorations are typical examples for such outdoor mobile robots. In such field, it is required to have both of rapid movement and adaptive function to rough terrain, while general wheel mechanisms are not suitable for such rough environments. To move in these environments, the robots need to be flexible to various environments.

There are many researches concerning rough terrain mobile robots for rescue and planetary exploration. In such field, the robots require high mobile ability on rough terrain. When we design such kinds of robot, it become very important to choose the mechanism as a mobile platform. Several types of mechanisms have been proposed as a mobile platform: Crawler type, wheel type, leg type, and their combinations.

Wheel type mechanism is the simplest mechanism and can be controlled easily, but in terms of moving on rough terrains, its performance is obviously inferior to the other two mechanisms. If we adopt wheel type and try to get enough mobility on slight obstacles, we have to utilize pretty large wheels.

The leg mechanism is able to adapt various kinds of environment, but, its weak points are low energy efficiency and complicated mechanism and control, that imply high cost and product liability problems. Those might be high barrier to develop them as a consumer product.

The crawler mechanism shows the high mobile ability on various terrains; moreover it is simple mechanism and easy to control. Therefore a lot of rough terrain mobile robots adopt a crawler mechanism.

However conventional single track mechanism has also mobility limitations; the limitation is determined by attacking angle, radius of sprockets, and length of crawler. In order to

improve its mobility, it is required to adjust the attack angle against the obstacles, enlarge the radius of its sprockets, and lengthen its crawler tracks. And the mobility on the area like the stairs is inferior to that of the leg (S. Hirose, 2000). Therefore, a lot of researches have been done to supplement these weak points. The main theme common to those researches is to improve the mobility performance on rough terrain. Generally, the method which changes the form of crawler is adopted as an approach for this main theme. In order to realize these transformations, many researches proposed the connected crawler mechanism.
The purpose of this chapter is also to develop a connected crawler robot for rough terrain. The connected mechanism is that; some stages with motor-driven crawler at its left and right side are serially connected by active joints. When this mechanism is adopted, it becomes problem that how many crawler stages should be connected.
Lee et al (C.H. Lee et al, 2003) designs the mechanism of two stages one joint type that uses two triangular crawlers, and shows the high mobility performance by the comparison of climb-able step height between proposed mechanism and a conventional one track type. "Souryu-III" (T. Takayama et al, 2004) is the connected crawler robots of 3 stages 2 joints type, and it shows high mobility by using some basic experiments such as climbing a step and stepping over a gap. "MOIRA" (K. Osuka & H. Kitajima, 2003) is 4 stages 3 joints type connected crawler, and it reports the maximum climb-able step height which was measured by some experiments.
As mentioned above, the mobility performance was improved by the number of stages. However this number was different in each research. The mobility performance was also evaluated by using different experiment and criterion.
Although we can observe such researches, there are no researches which show the standardized relationship between the number of stages and mobility performance. When a connected crawler mechanism is designed, there is no design guideline which indicates how many stages would be optimal. That is a big problem, because the number of stages is influenced to mobility performance strongly.
Therefore this chapter derives the each actuator's motion which conforms to the environments, and tries to derive the relationship between the number of stages and mobility performance. In particular, we set the environment as one step, and derive its relationship (Fig.1). Because the climbing step ability is important factor as one of the most fundamental mobility index (T. Inoh et al, 2005), moreover a climbing step experiment is adopted by many researches as an evaluation experiment for mobility performance on rough terrain. Thus this chapter shows sub-optimal number of crawler stages for connected crawler robot which isn't cleared, through deriving the relationship between the number of stages and maximum climb-able step height and expected value to climb a step. After that, it proposes the actual connected crawler robot, and show basic experimental result.

2. Deriving the sub-optimal number of crawler stages

In order to find sub-optimal number of crawler stages, we derive the maximum climb-able step height of n-stages crawler (n=2~10). In this derivation, there is an optimization problem for the joint motions. Because, if the joint can't realize suitable motion for the step, it might be impossible to exercise climbing ability which the mechanism has. Therefore, the optimized joint motions for the step are required. We set the environment to one step (Fig.

1), and then we try to solve the motion planning of each joint and derive the maximum climb-able step height.

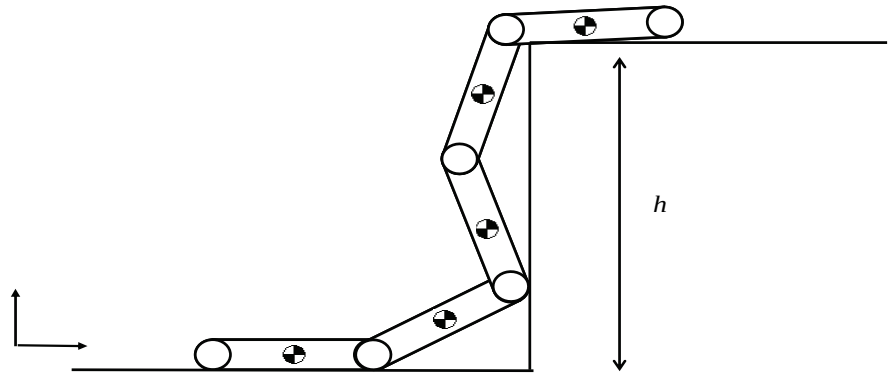


Fig. 1. The assumed case

The motions of climbing a step are divided into 2 phases which shown in Fig.2.

1. Lifting up crawlers phase
   This motion is strongly influenced by friction forces, contact forces and impact forces between environments and crawlers.
2. Passing over phase
   In order to generate a crock wise torque at the point of edge of the step and crawlers, the robot has to change its posture. This motion is strongly influenced by friction, balance of centre of gravity of the robot and inertia.

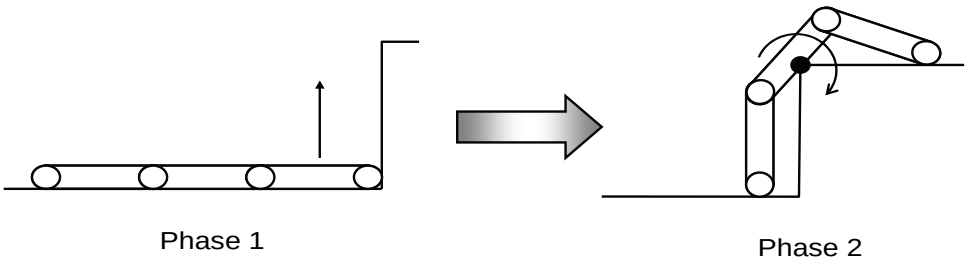


Fig. 2. The phases of climbing up a step

In each phase, changing the robot's posture is important. If the robot can not lift up the body as high as possible in Phase 1, the maximum climb-able step height can not be derived. Even if the robot can lift up the body as high as possible in Phase 1, if the robot can not generate the clock wise moment at the point of step edge, the climbing up a step can not be realized. Therefore, it is need to consider not only the moment in phase 2 but also both of Phase 1 and Phase 2. The maximum climb-able step height is distinguished by changes of postures. That is to say, the problem of driving the maximum climb-able step height is the

optimization problem of each joint motion. If the each joint can not realize suitable motion to the environments, it is impossible to exercise the ability of step climbing of the robot as maximal as possible. Thus the each joint motion is required to realize the suitable motion to the environments. But it is almost impossible to solve this problem by using analytical methods, because the amount of patterns of changing postures (from Phase 1 to Phase 2) becomes fatness by increasing the number of crawler stages. Thus the motion of each joint is required to be derived by using a certain searching method. However, the round robin-like searching method isn't so realistic, because the amount of searching becomes fat and calculation time becomes enormous.
Therefore, we propose the following idea as one of the approach to solve this problem. If certain approximate function can express an optimal joint motion in a few parameters, the required joint motion can be derived in shorter time than a round robin-like search. Therefore we try to express each joint angle function by using approximate function and search the parameters in this function. Thus the problem of the parameter searching can be substituted for the problem of the trajectory searching.
Moreover the robot has to change its posture with taking into interactions with environment, in order to climb a maximum step height.

2.1 Proposed method

In previous section, we described each joint motion are determined by certain approximate function, and to search parameters in this approximate function. In the following parts, we will mention the approximate function and how to search parameters, and show the method to derive maximum climb-able step height.

2.1.1 The approximate function

There are n-order approximation, a tailor progression, a Fourier series, a spline function, and so on, as an available approximate function. The approximate function must be possible to differentiate twice, so as to find an angular velocity and angular acceleration. It is also required that the function is periodic, and has a few parameters, and contains boundary conditions. Therefore, Fourier series is useful function to satisfy these conditions (Y. Yokose et al, 2004) . Thus, Fourier series approximates a joint angle functions. And the equation (1) is Fourier series for this approximation,

$$\theta_n(t) = \sum_{i=0}^{j} \alpha_i \cos\frac{i}{T} 2\pi t + \sum_{i=0}^{j} \beta_i \sin\frac{i}{T} 2\pi t \tag{1}$$

Here, n means the number of joints, j refers to the number of order of Fourier series, T means the period. a_i, β_i, T are parameters which are searched.

2.1.2 Searching for parameters in the Fourier series

Searching for each coefficient and period in the Fourier series corresponds to the problem which is to derive the optimized answer in a wide area. There are many approaches to solve such optimization problems. Many researches proposed to use GA for such a problem (Mohammed, 1997) ~ (S. Kawaji et al, 2001). Because, GA is able to find comparatively an excellent answer in the utility time, and fit various problems. Therefore this chapter also

adopts GA to search unknown coefficients in the Fourier series. We use simple GA (S. Kobayashi et al, 1995), and set following parameters (Table. 1).

Number of chromosomes	10
Gene Length for one coefficient[bit]	10
Crossover rate [%]	25
Mutation rate[%]	1

Table 1. Parameters of GA

We also set the equation (2) to evaluate the chromosomes.

$$E = h + \frac{1}{t} + \frac{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} z_i}{1000} \tag{2}$$

Here, h is the step height which the robot could climb up, t is the time for climbing up a step. Then, it is understood that the evaluation is high when the robot could higher step in shorter time. On the other hand, when the robot couldn't climb a step, we set h=0, t=100 as a penalty. However, in these conditions, the evaluation of gene which couldn't climb up a step becomes equal, and it makes difficult to execute crossover. Therefore the third clause of the equation (1.2) exists as the valuation item. Here, x_n, z_n are the centre of gravity coordinates of each stage. Thousand in the denominator is numerical value to scale it 1000 down .

2.1.3 The method to derive the maximum climb-able step height

In order to evaluate gene, we have to acquire appropriate position of centre of gravity in each stage and distinguish whether the robot could climb or not. Because mobility performance of the mobile mechanism concerns with topography characteristic closely, the consideration of the interaction with the environment is very important. Therefore we must consider dynamics and an interaction between robot and environment, for appropriate acquisition of centre of gravity position and distinction of climbing. Thus we adopt ODE(Open Dynamics Engine)(R. Smith) to calculate these values. ODE is open source software, and is adopted by many robotic simulators to calculate dynamics. We derived maximum climb-able step height by integrating ODE and GA. The calculation System is shown in Fig.3.

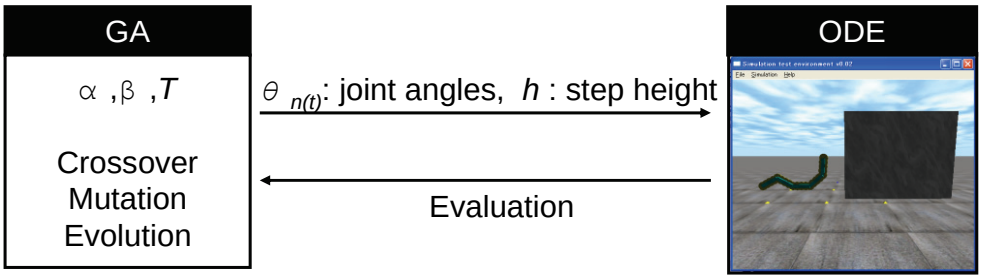


Fig. 3. Proposed simulation system

GA gives joint angles and step height, and ODE calculate dynamics. After that, ODE distinguishes whether the robot could climb or not, and returns the evaluation to GA. GA makes a gene evolve, and optimizes joint angle function. Then the robot can climb higher step in shorter time. A robot is considered to climb a step, when all centres of gravity in each stage are higher than the height of the step h and it is on the right of A in Fig1.

2.2 Deriving the maximum climb-able step height of *n*-stages

In this part, we derived maximum step height of n-stages based on the above mentioned method. We set the conditions and assumption as follows.

Each initial joint angle is set 0.0[rad], and the range is -2.0 ~ 2.0[rad]. The range of Fourier coefficients is -2.0 ~ 2.0. The range of Fourier series period T is 10 ~ 60[sec], and the order of Fourier Series is 5. The initial genes are determined randomly. The specifications of the connected crawler robots are shown in Table. 2 and Fig. 4. Other conditions are as follows.

Total length L [m]	2
Total mass M [kg]	2
Radius of the sprocket [m]	0.1

Table 2. Parameters of the connected crawler robot

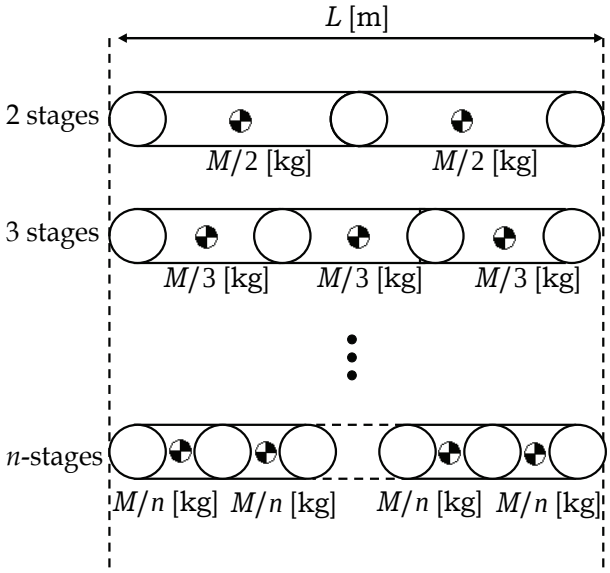


Fig. 4. The dividing definition of the robot

- Each stage is divided in constant total length L by corresponding to the number of links.
- The crawler velocity is constant 0.1[m/s].
- The actuators have enough torque for driving joints.

The range of step height h is 0.5 ~ 2.0[m], because the total length of connected crawler is L=2.0[m]. By using above conditions, the simulation is done which is 4 stages and the number of generations is 500. Then the maximum climb-able step height is derived.

2.3 Relationship between mobility and the number of stages

The results are shown in Fig.5 ~ Fig. 7.
In the Fig. 5, we can confirm that the robot could climb higher step when the number of generations is increased, and time for climbing was shorten.

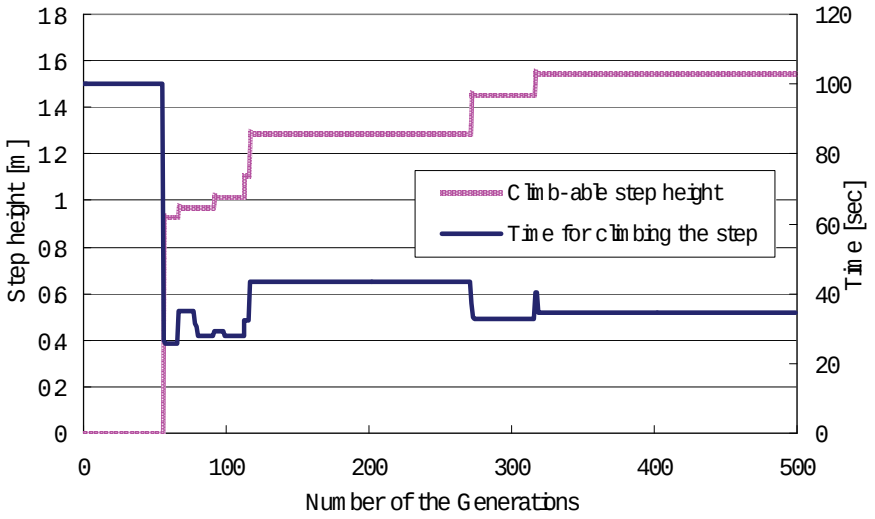


Fig. 5. Transition of the climb-able step height derived by GA (4 stages)

Fig.6 and Fig. 7 are snapshot when the robot is climbing up a step. In Fig.6, the height of step is 0.9[m] and the number of generations is 56. In Fig.7, the height of step is 1.544[m] (this is the maximum climb-able step height of 4 stages). From these figures, the climb-able step height becomes higher and the motions of the joins are changed when the number of generations is increased.
We also derive the maximum climb-able step height of 2 ~ 10 stages by using same method. The results are shown in Fig.7. It is confirmed that the robot can climb higher step when the number of generations is increased as well as the case of 4 stages, and maximum climb-able step height of each link is derived.
Since the maximum climb-able step height of each stage has been shown in Fig.8, the relationship between the number of stages and mobility performance of connected crawler is shown in Fig.9.
For this figure, the mobility performance improves by increasing the number of stages. This is natural result which can be expected. That does not clarify the sub-optimal number of stages. We, therefore introduce new criteria to derive sub-optimal number.

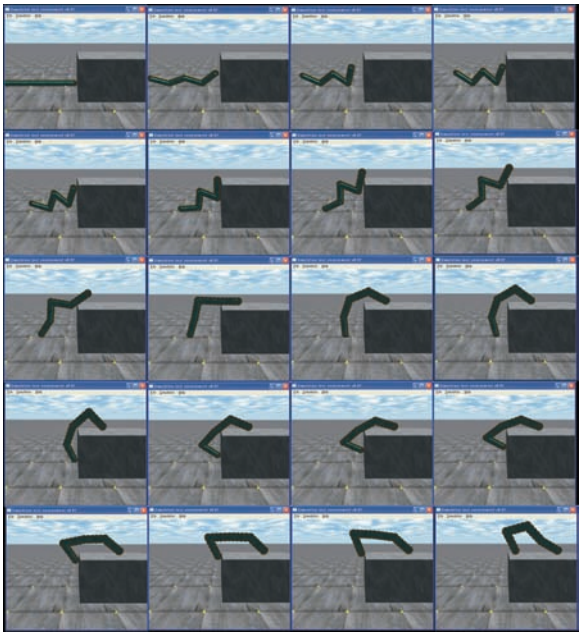


Fig. 6. Connected Crawler robot climb the step by using sub-optimized joint motion by GA (4 stages, h=0.9 m, 56 generations)

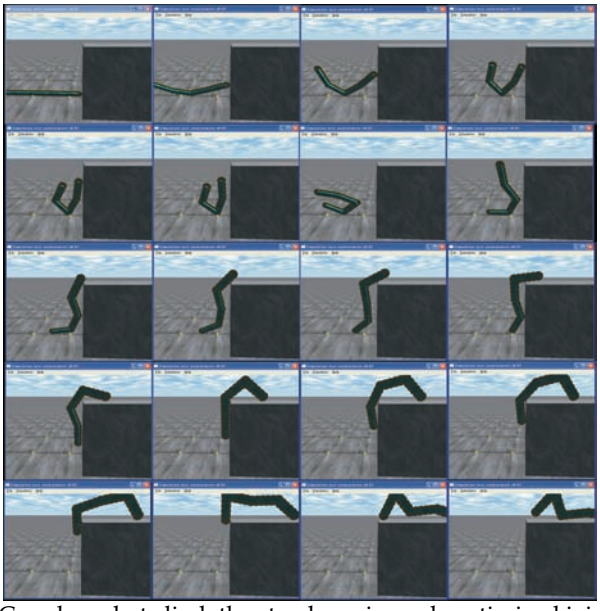


Fig. 7. Connected Crawler robot climb the step by using sub-optimized joint motion by GA (4 stages, h=1.544 m, 500 generations)

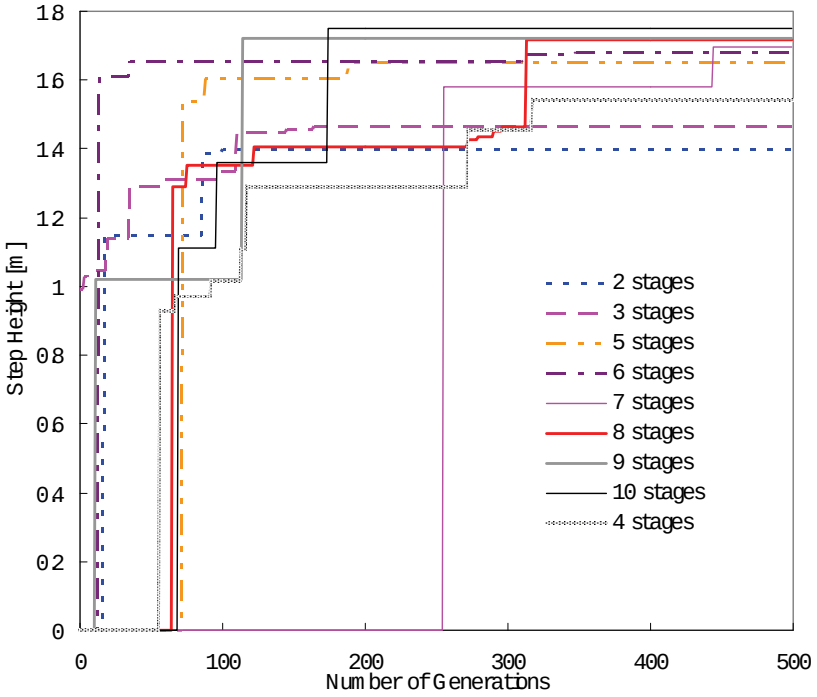


Fig. 8. Transition of the climb-able step height derived by GA (2 ~ 10 stages)

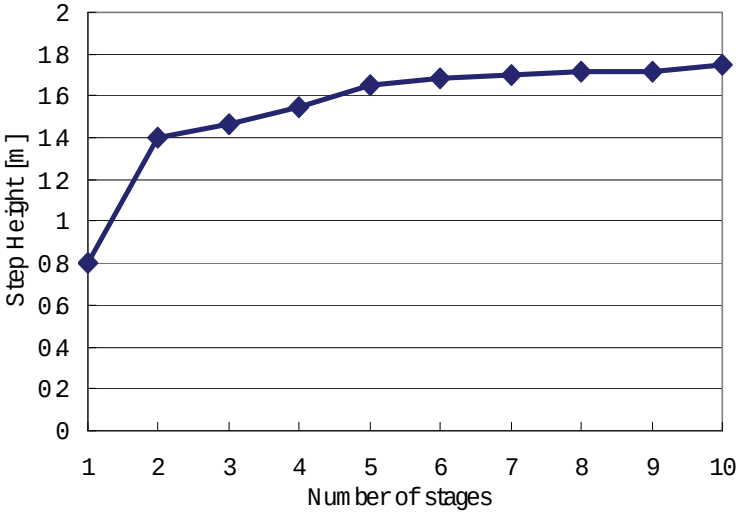


Fig. 9. Relationship between the number of stages and climb-able step height

2.4 Introducing expected value to climb a step

We introduce an expected value to climb a step to clarify the sub-optimal number of stages. Although the mobility improves by increasing the number of stages, failure probability of system also increases, because a connected crawler mechanism is one of the complex mechanical systems. It is considered that the relation between mobility and the number of stages is trade-off relation. Therefore, by introducing expected value which contains failure probability, sub-optimal number of stages is derived.
To derive the expected value, a certain probabilistic values P and the maximum step height h_{max} of each number of links are needed.

$$E = h_{\max} \times P \tag{3}$$

This certain probabilistic value P shows the probability to climb a step. Then we adopt the robot availability (rate of operation) as this certain probabilistic values.

How can we derive the availability of the system? This problem is categorized into the field of reliability engineering. And it is almost impossible to derive availability of a complex system like a robot precisely. Therefore there is a Fault Tree Analysis for deriving availability of such complex system. Fault Tree Analysis is a method to analyze faults and troubles, and is called FTA. For analyzing frequency of troubles, this method traces the risk of the cause theoretically and adds each probability of trouble. This method is one of the top down analysis method. The failure probability is derived by following steps.

1. First the undesirable event is defined.
2. The cause of the undesirable event is extracted.
3. The FAULT TREE is generated by using logic symbol.
4. The each failure probability is assigned.

The value which derived by FTA is the failure probability of the system. Then the availability P_a is derived following relationship between failure probability P_f and availability P_a.

$$P_a = 1 - P_f \tag{4}$$

In order to derive availability, we set following assumption for robot conditions.

- The joint has an optical encoder and a DC motor.
- The motor for driving a crawler is in each link.
- The failure probability of the optical encoder is 0.0155.
- The failure probability of the DC motor is 0.00924.

Mentioned failure probabilities above are determined by the reference (C. Carreras et al,2001). From the view point of availability engineering the Fault Tree of the connected crawler is shown in Fig. 10.

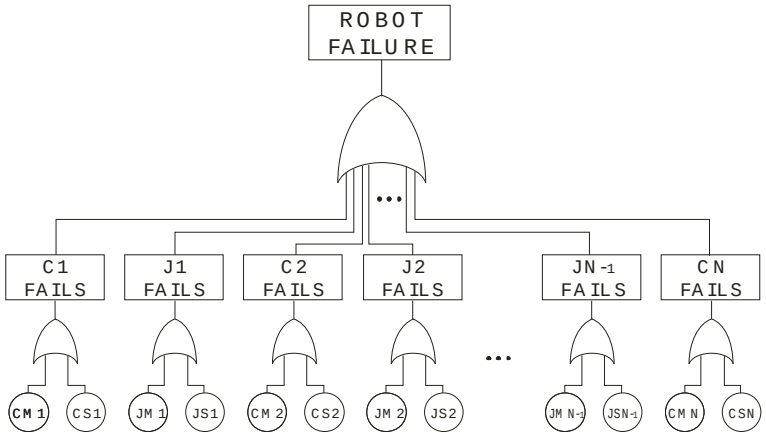


Fig. 10. Fault Tree for n-stages connected crawler robot

Here, J means joint, C is a crawler, M means DC motor, S means optical encoder. Therefore MJ1 refers to the DC motor on joint J1. SJ1 refers to the optical encoder on joint 1. CM1 represents the DC motor for driving crawlers on link one, C1. AlsoCS1 means the optical encoder on link one. By combining these value using OR logic, the failure probabilities of link 1 system or joint 1 system are derived. And each failure probability of joint and crawler is combined by OR logic, then the total failure probability of the robot is derived. By using Fig. 10, the availabilities of connected crawler robot are derived which are shown in Fig. 11.

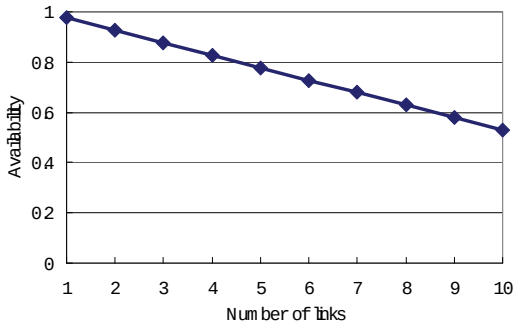


Fig. 11. The availability of each number of stages on connected crawler robot.

2.5 Sub-optimal number of stages

Previous section showed the availability of each number of stages in Fig. 11. Therefore the expected value of climbing a step can be now derived by using equation (3). Fig. 9. is used for h_{max}. The results are show in Fig. 12.

From Fig. 12., it turned out that the peak of expected value of connected crawler is from 2 to 5 links. In case of more than 6 links, the expected value is decreased. Therefore the sub-optimal number of stages is 2 to 5.

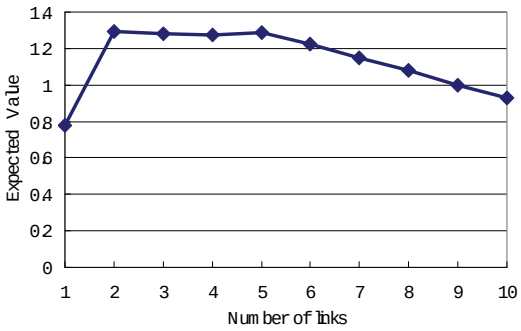


Fig. 12. Expected value of connected craweler

3. Constructing the prototype

In the previous section, we have been able to obtain the sub-optimal number of crawler stages, that is 2 to 5. Based on this conclusion, we have designed and developed the prototype of connected crawler robot. It is shown in Fig. 13. The length is 0.59 m, width is 0.130 m, mass is 1.28 kg.

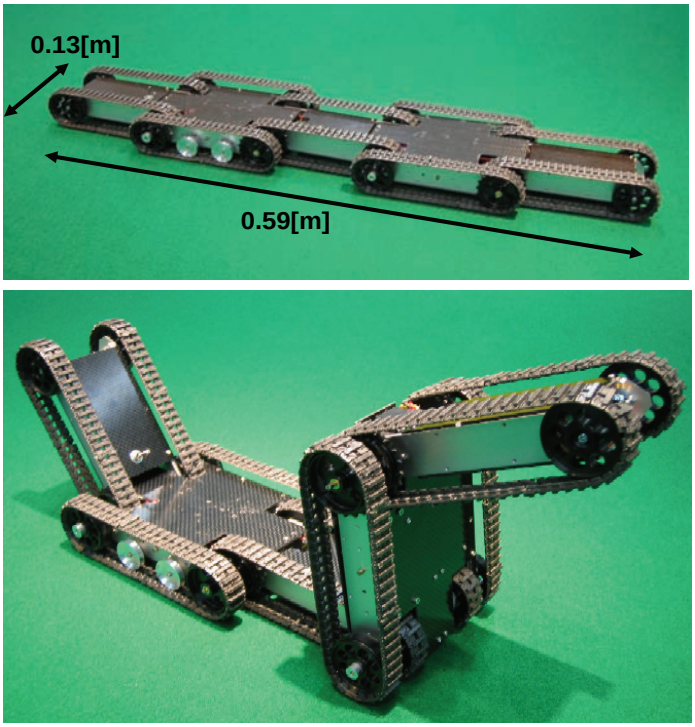


Fig. 13. Prototype of connected crawler robot

3.1 Mechanical structure

Our mechanism has 5 connected stages with the motor-driven crawler tracks on each side (Fig. 14). RC-servo motors are used for driving joints between the stages. The left and right crawlers are driven by 4 DC motors independently, while the 5 crawlers on each side are driven by a motor simultaneously. The output of each motor is transmitted to the sprockets of the three or two crawlers through several gears (Fig.15).

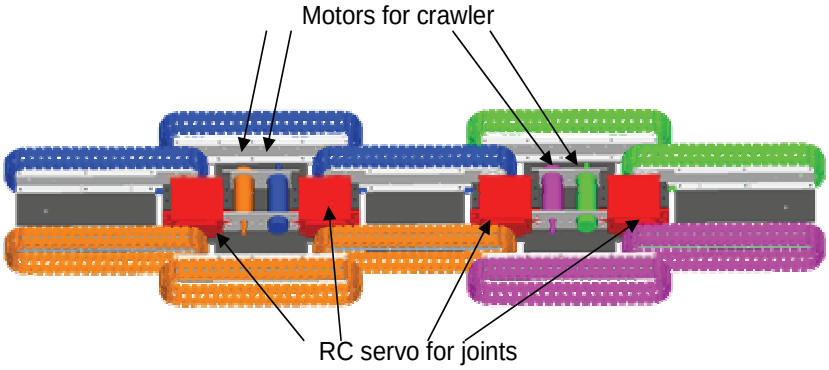


Fig. 14. The driving structure (Color indicates driving relationship between motors and crawlers)

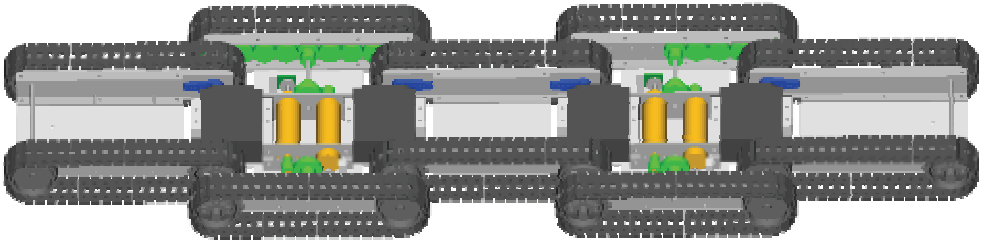


Fig. 15. Transmission of motor outputs to the crawlers

3.2 Control structure

The control architecture is hierarchical structure by connecting master controller and servo unit (Fig .16, and Fig. 17).

The servo units control low level task: crawler velocity and joint angle by PID control law.

Each servo unit consists of one microcontroller (PIC16F873) and 2 DC motor drivers (TA8440H). One microcontroller is installed to control two RC-servo units for the joint control, where RC-servo is controlled only by PWM signal. Master controller controls high level task: such as calculating robot trajectory. Table.3 shows the communication data format. The command sent by master controller consists of 3 bytes. First byte indicates mode ID and motor ID. The mode ID distinguishes 2 kinds of control modes: position control and velocity control. The motor ID is used for selecting motor to control.

Second byte shows the data depends on control modes. The third byte is checksum.

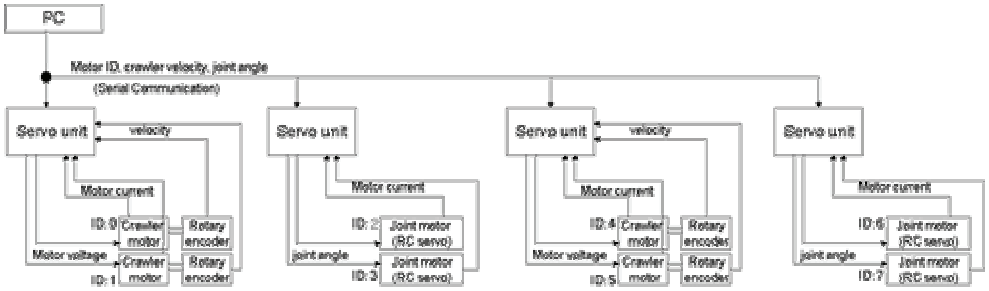


Fig. 16. The control system

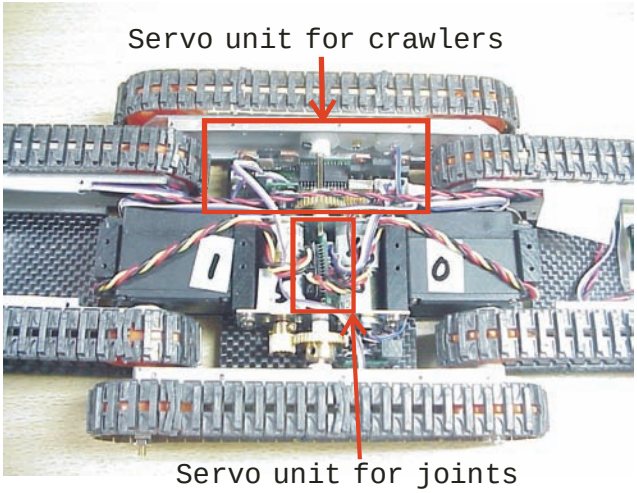


Fig. 17. The servo units

1 byte								2 byte	3 byte
Data 1								Data 2	Check Sum
7	6	5	4	3	2	1	0	0~254	Data1 | Data2
Mode=0~2			ID=0~7						

Table 3. Communication data format

4. Experiments

The climbing step experiment is conducted to verify the performance of our prototype. The height of step is 0.23 m. The master controller sends instructions to each actuator through servo units. Li-Polimer battery (1320mAh, 11.1V) is embedded to the robot for supplying electric power. In this experiment, PC is used as master controller. The USB cable is used for connecting robot to PC. The result is shown in Fig. 18. As we can observe, the robot can climb up a step. Therefore the mobility of this robot is confirmed.

5. Conclusion

This chapter showed sub-optimal number of crawler stages for connected crawler robot, through deriving the relationship between the number of stages and maximum climb-able step height and expected value to climb a step. After that, it proposed the actual connected crawler robot, and indicated basic experimental result. The conclusions of this chapter are as follows.

- A joint angle function was approximated by Fourier series and parameters were searched by GA.
- Due to fusion of GA and ODE, it has been possible to consider the interactions between robot and environment.
- The relationship between the number of crawler stages and mobility performance was cleared.
- Though mobility performance was raised by increasing the number of stages. However its increasing rate was small in comparison between before 5 stages and after 6 stages.
- To clarify sub-optimal number of stages, the expected value to clime a step was introduced.
- The peak of expected value is from 2 to 5 links.
- Therefore the sub-optimal number of stages is 2 to 5.
- By basic experimental results, the mobility of the prototype was confirmed.

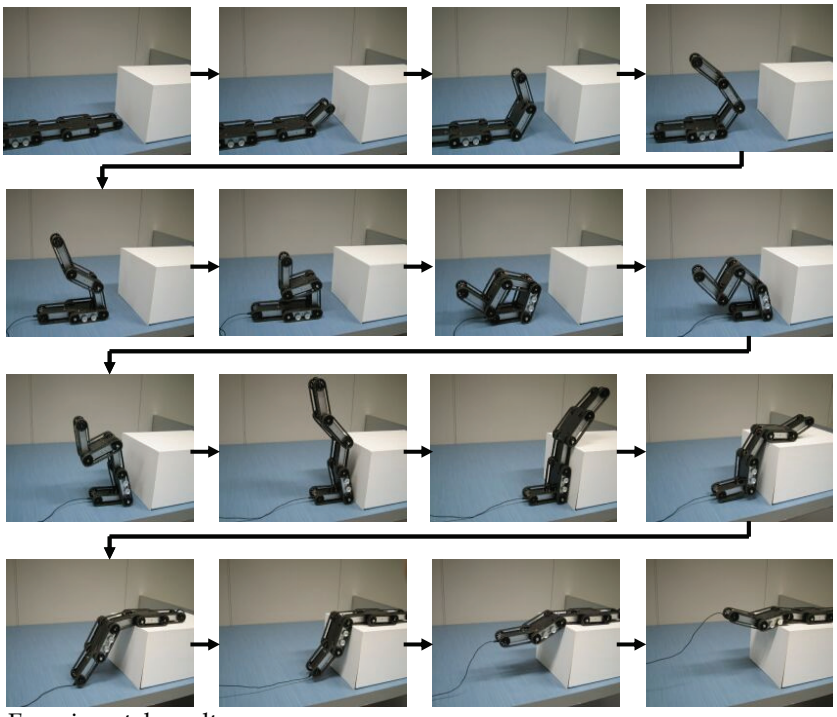


Fig. 18. Experimental results

References

C.H. Lee, S. H. Kim, S. C Kang, M.S.Kim, Y.K. Kwak (2003). "Double –track mobile robot for hazardous environment applications", *Advanced Robotics*, Vol. 17, No. 5, pp 447-495, 2003

K. Osuka, H. Kitajima (2003). "Development of Mobile Inspection Robot for Rescue Activities:MOIRA", *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems,* pp3373-3377, 2003

Mohammed G.F.Uler (1997). "A Hybrid Technique for the Optimal Design of Electromagnetic Devices Usign Direct Search and Genetic Algorithms"*IEEE Trans. on Magnetics*, 33-2, pp1931-1937, 1997

R. Smith, "Open Dynamics Engine", http://ode.org/

S. Hirose (2000). "Mechanical Designe of Mobile Robot for External Environments", *Journal of Robotics Society of Japan,* Robotics Society of Japan, vol.18, No.7, pp904-908, 2000 (*in Japanese*)

S. Kawaji et al, (2001). "Optimal Trajectory Planning for Biped Robots"*The Transactions of the Institute of Electrical Engineers of Japan. C*, vol.121, No.1, pp282-289, 2001 *(in Japanese)*

S. Kobayashi et al, (1995). "Serarch and Learning by Genetic Algorithms"*Journal of Robotics Society of Japan*, vol.13, No.1, pp57-62, 1995 (*in Japanese)*

T. Inoh et al (2005). "Mobility of the irregular terrain for resucue robots"10th Robotics symposia pp 39-44, 2005 (*in Japanese)*

T. Takayama, et al (2004). Name of paper. "Development of Connected Crawler Vehicle "Souryu-III" for Rescue Application "*Proc. of 22nd conference of Robotics Society of Japan CD-ROM,* 3A16, 2004 (*in Japanese*)

Y. Yokose et al (2004). "Minimization of Dissipated Energy of a Manipulator with Coulomb Friction using GA Increasing the Calculated Genetic Information Dynamically" *Transaction of JSCES*, Paper No.20040024, 2004 (*in Japanese)*

Y.Yokose V.Cingosaki, H.Yamashita (2000). "Genetic Algorithms with Assistant Chromosomes for Inverse Shape Optimization of Electromagnetic devices" *IEEE Trans. on Magnetics*, 36-4, pp1052-1056, 2000

C. Carreras, I. D. Walker (2001). "Interval Methods for Fault-Tree Analysis in Robotics", *Transaction on Reliability*, Vol. 1, pp. 1-11, 2001

21

Automatic Generation of Appropriate Greeting Sentences using Association System

Eriko Yoshimura[1], Seiji Tsuchiya[2], Hirokazu Watabe[1], Tsukasa Kawaoka[1]
[1] Doshisha University, [2]The University of Tokushima
[1,2]Japan

1. Introduction

When we humans start a conversation, we are greeting at first. If computer and robot are greeting like us, they can communicate smoothly with us because the next subject comes easily after greeting. That is to say, greeting conversation plays an important part to smooth communications in speaking. In this report, we describe a method of increase the number of appropriate greeting sentences for conversation and selecting sentences based on the situation by machine.

Many of conversation system tend to use templates. Lots of chatter bots (Eliza, A.L.I.C.E., Ractor, Verbot, Julia etc.) have been developed. For example, Eliza(Weizenbaum, J 1965) which is one of the well-known system acts for counselling by a personification therapist agent. Eliza does not evaluate an answer of a partner for the reply.

It memorizes only a part of the content that the partner spoke in the past and replies by using the word. It is prepared for several kinds patterns about the topic.

Like these, as for the natural language processing, task processing type conversation (e.g. automatic systems for tourist information and reservations) becomes the mainstream. However, even under the limited situation, it is known that it is difficult to make a knowledge base of all response case. Moreover, a method using only the prepared template makes monotonous reply and a reply except sentences made by a designer don't appear. So, to make various sentences automatically by machine is important, more than the method to select sentences designer prepared.

We herein propose an intelligent greeting processing by which a machine generates various reply sentences automatically by obtaining information about the surrounding state and then generating the best conversation response based on the situation.

All sentences are extended automatically from a small quantity of model sentences by using the concept base which is kind of natural-language ontology/concept networks. Simply mechanical extension of conversation sentences makes many improper sentences. So, the proposed method uses language statistics information to delete the improper sentences. In addition, for greetings conversation, we suggest "status space" expressing a certain situation. This is a model to give a weight to sentences automatically by taking the consequence at two

points of view that the appropriate selection that a human being performs unconsciously is classified in.

2. Requirements for Conversation Sentence

The greeting conversation is a "no insistence conversation" that does not cause argument or discussion. In the present paper, such greeting conversation sentences are synthesized automatically by machine.

The requirements for automatic conversation sentence synthesis by machine as follows:

1) Grammatical consistency
2) No contradiction in meaning
3) The use of usual words
4) Situation adaptability

"Grammatical consistency" refers to sentences in which no grammatical mistakes are found. "No contradiction in meaning" refers to sentences that have a reasonable meaning, e.g., the sentence "The sun is so bright tonight." is not reasonable from the point of view of time. "The use of usual words" indicates words used in daily life, including colloquialisms. "Situation adaptability" refers to sentences that do not contradict reality. For example, "It's a rainy today" contradicts the reality if the weather is fair.
Therefore, it is necessary to meet these requirements after a mechanical synthesis. The proposed system is constructed using the Japanese language and so is adapted to the characteristics of the Japanese language and Japanese culture.

3. Greeting Conversation System

Figure 1 shows the structure of the greeting conversation system proposed in the present paper. The greeting conversation system obtains inputs of the surrounding information and input sentence and then outputs greeting sentences. There are fixed pattern greetings, e.g., "Good morning" and "Hello" and greetings for starting a conversation, e.g., "It's been raining all day." and "It's very hot, isn't it?":

Human: "Good morning."
System: "Good morning.
It's very hot, isn't it?"

The former output is a fixed pattern greeting, the system can output sentence matching of the situation or input-sentences by a fixed pattern knowledge base. This problem can easily be solved. Thus, the latter greeting for starting a conversation is considered in the present paper. Greeting sentences indicate the latter.
The proposed method extends the small-scale template database of greeting sentences. Suitable sentences are then selected automatically from the extension template considering the situation. This is achieved by using an association knowledge system that will be described later herein.

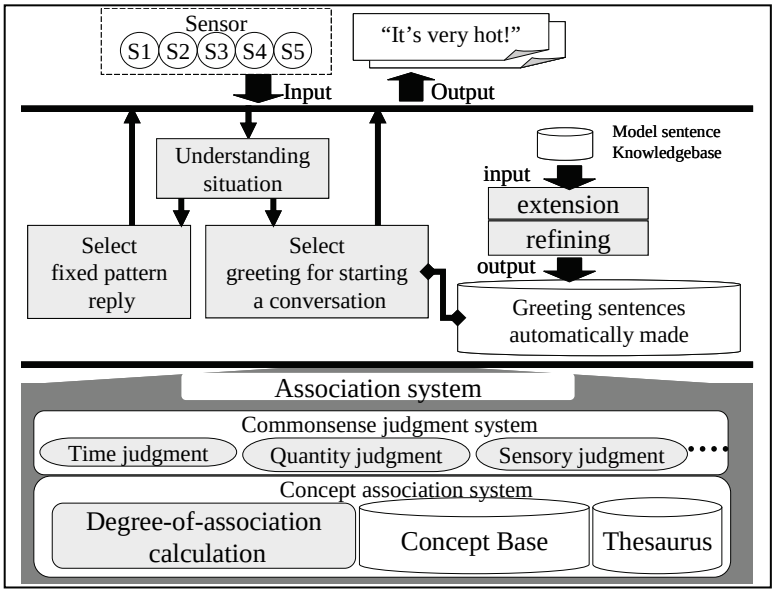


Fig. 1. Structure of greeting conversation system

4. Association System

Humans can interpret received information appropriately because we accumulate basic knowledge about the language and we have empirical common sense knowledge related to words. In other words, the ability to relate images to words is thought to be important. In order to have such common sense judgment ability, a machine must understand the basic knowledge related to words.

Therefore, the machine must generate modelling knowledge about conversations and words. This model is useful for achieving a human-like conversation mechanism. The association system is composed of a concept association system and a common-sense judgment system. The concept association system defines the common meanings related to words, and the common-sense judgment system defines common sense related to words.

4.1 Concept Association System

The Concept Association Mechanism incorporates word-to-word relationships as common knowledge. The Concept Association Mechanism is a structure that includes a mechanism for capturing various word relationships. In this section, we describe the concept base (Kojima et al., 2002) and a method of calculating the degree of association (Watabe & Kawaoka, 2001) using this base.

The concept base is a knowledge base consisting of words (concepts) and word clusters (attributes) that express the meaning of these words, which are mechanically and automatically constructed from multiple sources, such as Japanese dictionaries and newspaper texts. The concept base used in the present study contains approximately 90,000 registered words organized in sets of concepts and attributes. These concept and attribute sets are assigned weights to denote their degree of importance. For example, an arbitrary

concept, A, is defined as a cluster of paired values, consisting of attribute, a_i, which expresses the meaning and features of the concept, and weight, w_i , which expresses the importance of attribute a_i , in expressing concept A:

$$A = \{(a_1 , w_1), (a_2 , w_2), ..., (a_N , w_N)\} \quad (1)$$

Attribute a_i is called the first-order attribute of concept A. In turn, an attribute of a_i (taking a_i as a concept) is called a second-order attribute of concept A.

The degree of association is a parameter that quantitatively evaluates the strength of the association between one concept and another. The method for calculating the degree of association involves developing each concept up to the second-order attributes, determining the optimum combination of first-order attributes by a process of calculation using weights, and evaluating the number of these matching attributes. The value of the degree of association is a real number between 0 and 1. The higher the number, the greater the association of the word. Table 1 lists examples of the degree of association.

Concept A	Concept B	Degree of association between A and B
Flower	Cherry blossom	0.208
Flower	Plant	0.027
Flower	Car	0.0008
Car	Bicycle	0.23

Table 1. Examples of the degree of association

4.2 Common-Sense Judgement System

The common-sense judgment system derives common-sense associations from words in terms of various factors (e.g., quantity, time, and physical sense). These associations are constructed using the Concept Association Mechanism. In this section, we describe a time judgment system, a part of the common-sense judgment system.

The time judgment system (Tsuchiya et al., 2005) assesses elements of time, such as season and time of day, from nouns, using a knowledge base (Time Judgment KB) of words indicating the time (time word). The system sorts the relationships between a noun and time through the construction of the Time Judgment KB and extracts the necessary time words.

We identified a set of basic representative time words—"spring, summer, autumn, winter, rainy season," and "morning, daytime, evening, night"—and applied these words to all of the time words registered in the system.

Input	Output		
	Time word	Start time	End time
Sunset	Evening	4 p.m.	5 p.m.
Snow	Winter	December	February
Sea bathing	Summer	June	September

Table 2. Examples of the time judgment system

The system can also handle time words not contained in the time judgment knowledge base (unknown words) through the use of the concept base of common knowledge. Table 2 lists examples of this system.

5. Extension of Conversation Sentences

Upon greeting to start a conversation, not all conversation sentences can be gathered in a database manually. Therefore, all conversation sentences are extended automatically from a small quantity of model sentences by using the association system. That is to say, if a machine obtains several model sentences, then the machine can produce several new sentences by association (Yoshimura et al., 2006).
First, parts (noun) that change are selected from the model sentences. These words are called element words. Second, all words that have the opposite meaning to these element words are extracted for getting words of the same affiliate as this element. Third, the number of these words is increased by synonyms and the attributes of the concept base.
Using this method, the machine associates several words that are related to the element word. This association word is returned to the beginning model sentence. Thus, several sentences can be produced.

6. Removal of Erroneous Sentences

Simple mechanical extension of conversation sentences produces several improper sentences. Therefore, the proposed method uses the association system to delete the improper sentences.
First, parts of speech are used to refine the sentence. For example, the original model sentence "It's a season of cherry-blossoms" produces the extension sentence "It's a season of open blossom." This seems strange. Therefore, association words other than the part of speech of the original element word.
Second, a thesaurus (NTT Communication Science Laboratory,1997) is used to refine the sentence. For example, the original model sentence "It's a beautiful mountain" produces the extension sentence "It's a beautiful climax." In Japanese, the word *mountain* has many meanings, including mountain, climax, and important event. In this case, the original element word, *mountain* expresses the meaning 'mountain'. *Climax* is an extension of *mountain*, but the extension is to a meaning that differs from the original element word. The thesaurus is used to remove such ambiguity.
Next, the degree of association is used to refine the sentence. This is especially important for words strongly related to the original element word. Therefore, high degree words are extracted using the degree of association.
Finally, the common sense-judgement system is used to refine the sentence. In the present paper, strange sentences with respect to time are excluded. For example, simply mechanical extension of conversation sentences includes improper sentences such as "Tonight is cool daytime." This sentence is improper from the point of view of time. The time judgement system in the common sense judgment system judges the adaptability of *tonight* and *daytime* from the point of view of time.
This technique makes it possible to increase the appropriateness of sentences as greeting. These sentences include some sentences that can be used under in suitable situations. For

this reason, we must consider the situation related to greeting when we use these extension sentences.

7. A State about Greeting Conversation

Situation is important for some topics in greeting when the speaker recognizes some information. Information sources are divided into verbalized sources and non-verbalized sources. Verbalized sources are expressed using words such as news and weather. Non-verbalized sources are measurement sources that can measure: temperature, humidity, brightness, and volume of a sound, etc. When a computer uses a non-verbalized source, measured information obtained using the computer's sense organ (sensor) should change into some word. A situation related to greeting is expressed as mostly non-verbalized source. This paper pays attention to this non-verbalized source and verbalized source is used for a part.

Computers do not have sensory organs like humans. Computers recognize surroundings using mechanical sensors. In the present paper, the proposed method uses sensors such as thermometers, hygrometers, sound meters (microphones), luminance meters, and clocks with built-in computers. These are non-verbalized sources. After these non-verbalized sources are obtained, they are verbalized according to decided rules. Moreover, weather as a verbalized source is important information for greetings. Humans judge weather by looking at the sky. However, it is difficult for a machine to judge the weather based on photographs. Therefore, sources of weather obtain information by verbalized sources from the Web.

Using this situational information, appropriate sentences are selected from among the extension sentences. When discussing a topic related your situation, you unconsciously select the most suitable word. In the present paper, we separate this unconscious selection into two types, based on which the sentences obtain a weight indicating the importance of the greeting.

The first type is based on degree of peculiarity. When a certain state is considered to be special, the state is mentioned. For example, if you feel that "It is very hot.", you will mention the heat. This means that a signal that is different from daily life indicates a noteworthy topic. The weight given based on this idea is called the "degree of peculiarity".

Simple and comprehensible words are often used for greetings in spoken language. For example, "It is cloudy." is used more often than "It is cloudy weather." as a greeting. This means that words generally used in daily conversation, i.e., words of high utilization, are important. The weight given based on this conceit is called the "degree of importance" of a word.

Using the degree of peculiarity and the degree of importance of word, a certain situation is expressed. A space storing relation between a status word and its weight is referred to as a *status space*. Figure 2 shows a status space image, including the sensor, the status word, and its weight. The status space includes all words used in greeting conversation, and the weight of a word changes according to the state. All words are related to a status word. In a status space, all words are categorized based on words obtained from status words. A word group related to a certain status word is called a status word group of the word.

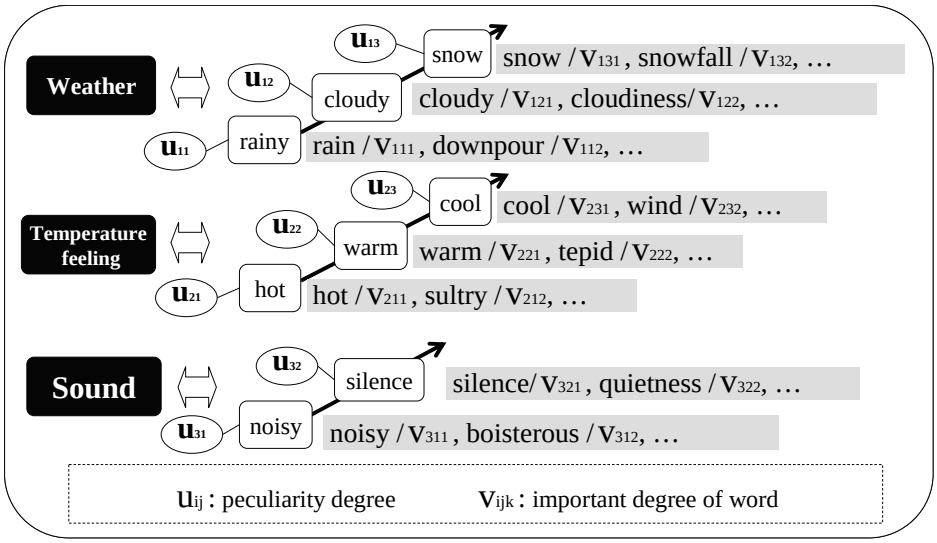


Fig. 2. Status space image

8. Weight for situation in Greeting

Simple and comprehensible words are often used for greetings in spoken language. The weight based on this idea is referred to as the *degree of importance* of the word. Each word has a weight in status space. To show the degree of usefulness of a word, we use the concept base Inverted Document Frequency (IDF). The concept base IDF is the weight related to the frequency of use of a word in the concept base. This technique means that low-frequency words in the concept base are not used frequently in daily conversation. Such words are excluded by using the concept base IDF. The concept IDF can be expressed as follows:

$$idf(t) = \log N_{ALL} / df(t)$$

t : object word

N_{ALL} : number of all concept in the concept-base

$df(t)$: number of t in an attribute in the concept-base (2)

Another weight for the situation in greeting is expressed as the *degree of peculiarity*. When we feel that a certain state is special, the matter is mentioned. The degree of peculiarity expresses a noteworthy level as topic, and each status word is assigned a degree of peculiarity, which takes a discontinuous value of {-1, 0, 1, 2}. A minus value is assigned to a status word that should not be mentioned as topic. If the state is "hot", the status word "hot" is assigned the value of 1, but the status words "cold", "cool", and "warm" are assigned the value of -1. If the state is "very hot", then status word "hot" is assigned the value of 2 in order to increase the noteworthy level as the topic. Moreover, if a state is not hot, cool, warm, or cold, these status words are assigned the value of 0, which expresses a

state of no feeling and is also used when the machine has no related sensor. All words of a status word group have the same value as that status word.
These weights depend on the state and are expressed as status-space. The weight of a word is decided depending on the multiplication of these weights. Extension sentences can have such weights. If there is some word having a negative value in a sentence, the sentence is given a weight of negative value. However, when a sentence is given a mean value of all word weights in the sentence, the sentences corresponding to the corresponding situation can be selected.

9. Evaluation

This section mentions evaluation of proposal system in this present paper. First, we evaluate the accuracy and number of extended and refined sentences, and then we evaluate the appropriateness of sentences based on the situation.

9.1 Increasing the Number of Greeting Sentences for Conversation

This section evaluates the accuracy and the number of sentences that are extended and refined.
A total of 300 evaluation sentences are chosen at random from among the extension sentences using the technique described herein. Three individuals evaluated these sentences based on common sense correctness of the greeting.
Accuracy is the mean value of common sense correctness of the word as judged by these three individuals. Using the method described herein, we showed that the proposed method was able to increase the accuracy and number of the greeting sentences, as shown in Fig. 3.

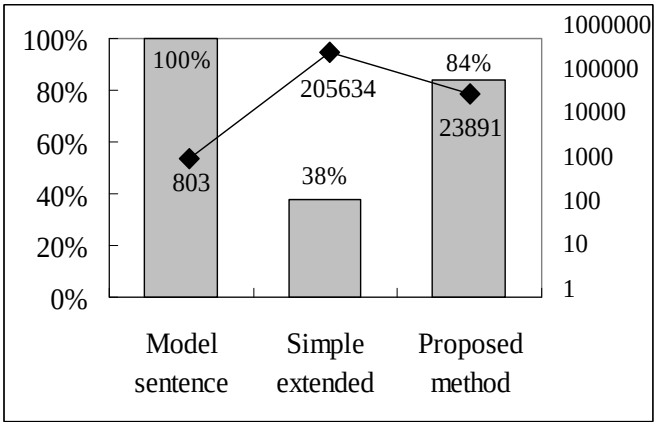


Fig. 3. Evaluation of extended greeting sentences

9.2 Greeting Sentence Selection Based on the Situation

This section evaluates the accuracy of the appropriateness of a sentence based on the situation. An evaluation set is prepared for 20 situations (weather, temperature and humidity, brightness, time, and volume of sound). In each situation, the status space is

produced and then extension sentences are assigned a value using the proposed technique. Six sentences having higher values are used as evaluation sentences. The three individuals evaluated the sentences and the situation patterns. They evaluated whether the sentences were appropriate to the situation. In addition, the accuracy is compared with the accuracy of randomly outputting sentences in order to verify the effectiveness of the proposed method. This result is shown in Fig. 4.

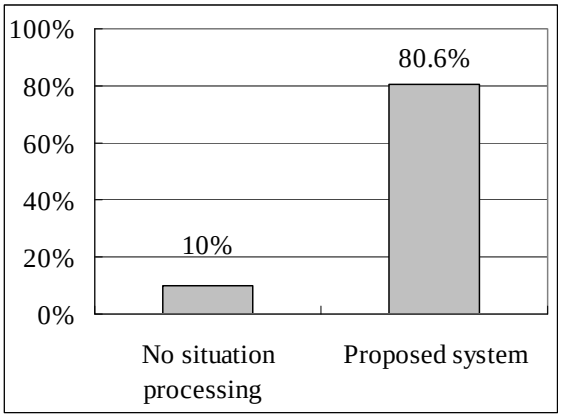


Fig. 4. Evaluation of Greeting Sentence Selection Based on Situation

10. Discussion

With respect to the technique of increasing the accuracy of greeting sentences, the proposed technique could improve the accuracy to 84%, compared to a simple extension technique, for which the accuracy is 38%. Moreover, the number of sentences increased from 803 sentences to 23,891 sentences. Table 3 shows examples of extension sentences.

Template		It's a beautiful river.
Extension Sentences	Refined sentence	It's a beautiful flow.
		It's a beautiful spectacle.
		It's beautiful scenery.
	Deleted sentence	It's a beautiful water imp
		It's a beautiful muddy stream.
		It's a beautiful crime case.

Fig. 5. Example of Extension Sentences

"It's a beautiful flow" and "It's a beautiful spectacle" are used as greeting sentences, but "It's a beautiful water imp" and "It's a beautiful muddy stream" are not used. These sentences were deleted automatically using the proposed method. Therefore, the proposed method of generating extended greeting sentence is shown to be effective.

The proposed technique of weighting corresponding to the situation was shown to have an accuracy of 80.6%, whereas randomly output sentences have an accuracy of 10%. The proposed technique of weight selection corresponding to the situation was shown to be effective.

11. Conclusion

Using the method described herein, we show that the proposed method increased the accuracy of greeting sentences to 84% and increased the number of greeting sentences to 23,891 sentences from 803 model sentences. Moreover, the technique used to select an appropriate sentence based on the situation was able to achieve an accuracy of 80.6%. When the situation was not considered, the accuracy was 10%. Thus, the proposed method is shown to be effective.

When the proposed technique is applied, even for the same situation, the machine does not reply using the same sentences in different conversations because a different reply can be chosen for high-weight sentences. As a result, the machine will be able to have an intelligent conversation with a high level of satisfaction.

References

Kojima, K.; Watabe, H. & Kawaoka, T.(2002). A Method of a Concept-base Construction for an Association System: Deciding Attribute Weights Based on the Degree of Attribute Reliability. *Journal of Natural Language Processing*, Vol.9, No.5, 2002.10, 93-110, ISSN1340-7619.

NTT Communication Science Laboratory.(1997). *NIHONGOGOITAIKEI*, Iwanami Shoten, ISBN4-00-009884-5 C3581.

Tsuchiya, S.; Watabe, H. & Kawaoka, T.(2005). A Time Judgement System Based on an Association Mechanism. *Proc. of KES2005*, Part3, 2005.9, 742-748

Watabe, H. & Kawaoka, T. (2001). Measuring Degree of Association between Concepts for Commonsense Judgements. *Journal of Natural Language Processing*, Vol.8, No.2, 2001.4, 39-54, ISSN1340-7619.

Weizenbaum, J. (1965). ELIZA – A Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the Association For Computing Machinery*, Vol.9, No.1, 36-45

Yoshimura, E.; Watabe, H. & Kawaoka, T. (2006). An Automatic Enhancing Method of Greeting Sentences Using Association Knowledge Mechanism,.*Journal of Natural Language Processing*, Vol.13, No.1, 2006.1, 117-141, ISSN1340-7619.

22

Extending AI Planning to Solve more Realistic Problems

Joseph Zalaket
Holy Spirit University of Kaslik
Lebanon

1. Introduction

Applying AI planning to solve real-world problems is still difficult despite many attempts done in this area. Classical planning systems are able to handle a limited number of symbolic data elements, without taking into account the numerical aspect of many real-world problems. Some recent planners have moved to solve more realistic problems involving resource consumptions and time management (Bresina et al., 2002; Bacchus & Ady, 2001; Edelkamp, 2002), but the most of these planners deal with the numerical side of the planning problem as an assisting feature to a main symbolic problem, without being able to tightly mix the two sides of the problem to be solved as one homogeneous problem. However, there are still many real-world problems that involve dominant even absolute numerical processing (Zalaket & Camilleri, 2004b; Hoffmann et al., 2007) and for which planning problem representation and data type handling are to be extended. In order to cover this latter type of problems, we propose many extensions that allow the application of the planning process evenly over symbolic and numerical data that can constitute any realistic planning problem.

Despite the multiple extensions that have been made to the Planning Domain Definition Language (PDDL) (Ghallab et al., 1998; Fox & Long, 2002; Gerevini & Long, 2005), this language is still insufficient for representing real-world problems that need complex action representation and complex state transformation expression. This lack motivates the organizers of the sixth international planning competition (IPC-6) to request a new extension to PDDL (the PDDL3.1 version). Inspiring from the continuous extensions to PDDL, we propose our first extension that concerns the data representation, in which we introduce the concept of using non-invertible functions to update the numerical and non-numerical data throughout a planning process. This type of functions allows the integration and the handling in an easy way of uncertainty as well as of temporal and numerical knowledge into planning.

As non-invertible functions can be only applied in forward traversal in a search space, hence we focus on the adaptation of forward planning systems to support the application of this kind of functions. We show that our technique can be used by any forward planner with a minor expansion, and we detail the extension of the Graphplan (Blum & Furst, 1995) structure and algorithm to support the execution of functions. The advantage of the

Graphplan compared to other forward planning algorithms is that: Graphplan is constructed in progression, thus it can support the application of our proposed functions like any other forward planning system. Furthermore, the compact structure of Graphplan and its capability of delaying the search of the plan (the sequence of operators) to the end of the graph construction process give it the ability of dealing with the effects of our proposed non-invertible functions during the regression search phase.
As we illustrate in this chapter, it is enough to simply save the effects of functions during the construction phase of the Graphplan to be able to extract the plan later in backward search. Saving these effects allows also the application of black-box functions with no side effects over state variables irrespective of their content. This can lead to the integration of control structures like conditional and iteration structures into functions in order to perform more complex computation.
As additional extensions: we present the relaxation of the numeric tasks of planning by ignoring the effects of actions that move away the values of numeric variables from their goal values.
We present the calculation of a heuristic function, which can be uniformly derived for numerical and symbolic facts from a relaxed planning graph-like structure.
We introduce the representation of numerical facts as multi-valued attributes in the relaxed graph to ease the search of a relaxed plan. In this way, the effects of the applied non-invertible functions will be transparent during the search for a relaxed plan.
Finally, we present some empirical results that show the effectiveness of our extensions to solve more realistic planning problems.

2. Apply functions in planning

Using functions in planning has been studied in Functional Strips (Geffner, 1999) and FSTRIPS (Schmid et al., 2002). Functional Strips has argued that the generated literals can be reduced by replacing relations by functions. We are still supporting this idea in our extension, but our main interest in functions is their ability to handle complex numerical and conditional effects, as well as to express complex preconditions and goals. For example, to plan the motion of a robot that is expected to travel from an initial position to a target position in the presence of physical obstacles in its environment. To avoid collisions between the robot and the obstacles, we have to use trigonometric functions that allow the robot to follow a circular path in its environment (Samson & Micaelli, 1993; BAK et al., 2000). Trigonometric expressions are simple to be solved by classical programming languages, but they can not be expressed in classical AI planning languages like PDDL3.0. For this reason we introduce the extension of AI planning to allow the application of external functions that can be written in any programming language. Inspiring from the representation of functions in object-oriented database, we add to the problem definition written in PDDL3.0 (Gerevini & Long, 2005) the declaration of external functions by specifying their execution path. In addition to their capabilities of solving numerical problems like the circular movement of a robot external function are able to handle all kind of conditional and probabilistic effects.
Non-invertible function can be only applied during a forward traversal in the search space, as at each state we can apply functions that generate the next state in that space. This process can continue by testing at each new state the satisfaction of the goal conditions, until

reaching a state that satisfies the goal or until no more memory space is available to continue the search. In this way, a function can act as a black-box irrespective of its content, but domain independent forward planning systems are not able to solve large planning problems because of the high space and time complexities. So, heuristic functions should be used to guide the search in forward chaining in order to allow the application of functions and the resolution of large planning problems. As Graphplan is constructed in forward chaining traversal, thus it can support the application of this kind of functions. It will be also possible to extract the plan in backward search from the Graphplan structure by navigating through the effects of the functions irrespective of their content.
To avoid all kind of side effects during the construction of the planning graph we restrict all the formal parameters of the functions to be constant, which means, a function can only return a value without updating any state variable. This returned value can then be affected to one state variable at a time.
Many real-world problems have complex characteristics, such that durative actions, temporal or uncertain conditions, resource consumption, numeric functions, etc. Most of these characteristics can be modeled with the PDDL3.0 language which is the mostly used by recent planners. PDDL3.0 gives the possibility of expressing optimization criterion such that maximization or minimization of resource consumption. Many planners have been extended from propositional versions to deal with optimization problems like Metric-FF (Hoffmann, 2002) and SAPA (Do & Kambhampati, 2001). The most of these extensions are paying attention to one or the other of the real world complex characteristics, each time with a special computation procedure which is added to a classical propositional planner.
We propose an extension to the Graphplan (Blum & Furst, 1995) that allows its structure to handle all kind of non-invertible functions application. In the next section, we start by presenting the extension of the planning language to allow the integration of external functions that we will integrate later to the Graphplan.

3. Language extension

The language that we propose to define a planning problem is an extension to PDDL3.0 language in which we introduce the integration of external functions to update numeric values in addition to the arithmetic expressions allowed in PDDL3.0. This extension allows the use of mathematical functions like COS, SIN, SQRT, EXP, ROUND,. . . and user defined functions instead of simple arithmetic expressions supported by PDDL3.0 in which only classical arithmetical operators (+, -, /, *) are allowed. The control flow (conditional statements and loops) can be used within an external function to hold up complex numerical computation and thus complex conditional effects of actions can be expressed and handled within the core of external functions. Note that, external functions can also be used to update non numeric variables such as propositional facts. But, we restrict in this chapter the use of external functions for updating numeric variables which is more beneficial for the planning process.
We also introduce the separation of the constraints from the precondition in the definition of actions, in a way that the constraints will be added to a new separate list that should be tested before the instantiation of the action. Hence, an action will be instantiated if and only if its constraint list is satisfied by the current state which can reduce the number of ground

actions, and consequently useless tests that should be done for precondition satisfaction will be avoided and the memory space used to store the ground actions will be reduced.

3.1 State space nature and transition

Each distinct instance of the world is called a state, denoted by s. The set of all possible states is called a state space S.
S is formed by two disjoint sets: a set of logical propositions P and a set of numeric variables N.
We denote by P(s) the subset of logical propositions of a state s and by N(s) the subset of its numeric variables.
The transformation of the world from a state $s \in S$ into another state $s' \in S$ is done through the application of a set of actions A, such that $s' = t(s, A)$ where t is a state transition function. A state transition function t transforms s by 3 ways:

- Adds logical propositions to P(s)
- Removes logical propositions from P(s) (when propositions become false)
- Assigns new values to existing numeric variables in N(s)

3.2 Action definition

An action *a* is defined as a tuple (args, con, pre, eff), where:

- args is the list of arguments made by variables which represent constant symbols in S and/or numeric variables in N.
- con is the list of constraints. The constraints are tested before the instantiation of the actions to avoid instantiating actions with incoherent arguments. Constraints follow the same definition format as preconditions.
- $pre = pre_P \cup pre_N$ is the list of preconditions.
    - pre_P defined over P are propositional preconditions
    - pre_N are numerical preconditions, such that: $\forall c \in pre_N$, $c = (n\ \theta\ g)$, where $n \in N$, $\theta \in \{<, \leq, =, >, \geq\}$ and g is an external function or an expression.

Definition-1: *An expression is an arithmetic expression over N and the rational numbers, using the operators +, -, * and /.*

Definition-2: *An external function f is a constant function written in a high level programming language on the form of: type $f(n_1, n_2, , n_m)$, where arguments $n_1, n_2, , n_m \in N$.*

The function and all its arguments (if exist) are declared as constants in a way that the function calculates and returns a value without affecting any numeric state variable. By constant function we mean that the function is not allowed to internally modify the state variables.

- $eff = eff_P \cup eff_N$ is the list of effects.
    - $eff_P = eff_P^+ \cup eff_P^-$: defined over P are positive and negative propositional effects that add or remove literals.
    - eff_N are numeric effects, such that: $\forall e \in eff_N$, $e = (n := g)$, where $n \in N$ and g is an external function or an expression.

Figure-1 illustrates the definition of the water jugs domain using the above extensions to the planning domain definition language.

Example-1: *The water jugs domain definition using condition list and external functions.*

```
(define (domain Jugs)
(:requirements :typing :fluents :external)
(:types jug int)
(:functions ((capacity ?j - jug) – int)
((fill ?j - jug) – int))
(:external java (and ( (path (int max (int, int, int)) (c:/javaplan/WaterJug.max))
(path (int min (int, int, int)) (c:/javaplan/WaterJug.min))))

(:action poor
        :parameters
                (? j1 ? j2 – jug)
        :constraints
        (not(= ?j1 ?j2))
        :precondition
                (and (<(fill ?j) (capacity ?j)) (> (fill ?j) 0))
        :effect
                (and (assign (fill ?j1) (max ((fill ?j1), (fill ?j2), (capacity ?j2)))
                (assign (fill ?j2) (min ((fill ?j1), (fill ?j2), (capacity ?j2))))


External functions written in Java at c:\javaplan\.;
 public class WaterJug{
        public static final int max (final int v1, final int v2, final int c2){
                if (v1+v2 > c2) return v1+v2-c2;
        return 0;
        }

        public static final int min (final int v1, final int v2, final int c2){
        if (v1+v2 < c2) return v1+v2;
        return c2;
        }
}
```

Fig. 1. The water jugs domain definition

Remark: An external function can be written in any host programming language in this example we show functions written in java language, but these same functions can be written in a different programming language like the C++ for example. A special parameter should be set to allow the planner to know which interpreter should call to execute the functions as it is the case for functions written in Java or if the functions are directly executable as for functions written in C or in C++.

3.3 Planning problem definition

A planning problem is defined as P=<S, A, s_I, G>:

1. A nonempty state space S, which is a finite or countable infinite set of states.
2. For each state s ∈ S, a finite set of applicable actions A(s). A state transition function *t* produces a state s′ = *t*(s, A) ∈ S, for every s ∈ S and applicable actions A(s).
3. An initial state s_I ∈ S, where s_I is made by initially true logical propositions and initial values of numeric variables.
4. A set of goal conditions G that should be satisfied at a state s_G ∈ S. The set of goal conditions G = $G_P \cup G_N$, where G_P defined over P are propositional goal conditions and G_N are numeric goal conditions that should be satisfied at a goal state s_G.

   Note that: ∀ $l \in G_N \Rightarrow l = (n\ \theta\ c)$, where $n \in N$, $\theta \in \{<, \leq, =, >, \geq\}$ and *c* is a constant numeric value, an external function or an arithmetic expression.

4. Graphplan extension

Graphplan (Blum & Furst, 1995) was the subject of many extensions (Smith and Weld, 1999; Do & Kambhampati, 2000; Cayrol et al., 2000) in which researchers have tried to adapt the Graphplan to solve more expressive planning problems than those that can be expressed in pure propositional STRIPS language (Fikes & Nilsson, 1971). The adaptation that we propose to the Graphplan allows its structure to handle the execution of external functions. This type of functions allows Graphplan to handle uncertainty and to integrate temporal and numerical knowledge. Before introducing our extension the the Graphplan and to its planning graph structure we briefly overview the Graphplan algorithm.

4.1 Graphplan overview

Graphplan solves problems represented in STRIPS language. An action in STRIPS has a name and a parameter list and it is specified in terms of preconditions and effects. Preconditions are conjunctions of positive literals and effects are conjunctions of positive and negative literals that are respectively stored into two separated lists the ADD list and the DEL list.

Graphplan alternates between two phases: graph construction and solution extraction. In the graph construction phase Graphplan expands a planning graph in forward chaining until either a solution is found or it is sure that no solution exists. Then, in solution extraction phase Graphplan searches for a solution into the planning graph in backward chaining, if no solution is found then Graphplan will continue by expanding one more level until a solution is found or until a sufficient condition indicates that there is no solution to the problem.

A planning graph consists of a sequence of levels, where $level_0$ corresponds to the initial state. Each level contains a set of literals and a set of actions. The actions of a level are those that have their preconditions reachable (satisfied and mutually consistent) by the literals of the same level. The literals of a next $level_{i+1}$ are the effects of the applicable actions of previous $level_i$. The literals that satisfy the preconditions of action instances at the same level are connected with these actions via direct edges. Also, the action instances are connected via direct edges with their literals effects at the next level. ADD edges connect the action to

its next level added literals and DEL edges relate the action to its next level deleted literals. To maintain the existence of literals from one level to its successor level Graphplan uses the "no-op" actions. A "no-op" action is an action that "does nothing", its only role is to allow every literal that appears at $level_i$ to also reappear at $level_{i+1}$.
To speed up its solution extraction Graphplan uses binary mutex relations for both actions and literals during graph expansion.
A mutex relation holds between two actions at a given level if there is:

- *Inconsistent effects:* one action negates an effect of the other.
- *Interference:* one literal effect of one action is the negation of a precondition of the other.
- *Competing needs:* a literal precondition of one action is mutually exclusive with a literal precondition of the other.

Two literals are mutually exclusive if one is the negation of the other or if each possible pair of actions that could achieve the two literals has a mutex relation.
Graphplan expands the planning graph up to a level in which all the literals of the goal are present without mutex relation between any pair of them. Then, it starts the solution extraction phase in a way that for each sub-goal at the last $level_n$, it tries to find an action instance at $level_{n-1}$ that has this sub-goal as an add effect and that is not mutually exclusive with another action instance that is already selected. The preconditions of the selected action instances become the new set of sub-goals at $level_{n-1}$ and so on until reaching the initial state ($level_0$). If Graphplan fails to find an action that can achieve a sub-goal, it backtracks and tries to find another path within the planning graph by trying other action instances. Graphplan uses the 'memoization' concept which allows it memorizing its trace to ease the backtracking process. Therefore, if Graphplan fails to find a set of consistent action instances at a certain level and backtracking becomes useless, then Graphplan expands an additional planning graph level and restarts its search until finding a solution or until reaching a saturated graph that can not be expanded any more.

4.2 Graphplan adaptation for integrating external functions

In algorithm-1 we present the adaptation of the Graphplan algorithm to support the handling of numeric variables and the execution of external functions. In our implementation the adapted Graphplan consists of 2 types of levels fact levels and action levels. Compared to the original Graphplan implementation a fact level is a combination of propositional and numeric facts, instead of being only propositional. An action level supports the execution of external functions and the evaluation of arithmetic expressions in addition to its capability of adding negative and positive propositional effects. Each numeric variable that belong to a fact level is represented as a multi-valued attribute in the planning graph structure. When an action updates a numeric variable using an external function or an arithmetic expression, we add this updated value as a new value to the multi-valued concerned attributed. In this way, we consider that the old value is deleted and a new value is added to the multi-valued attribute. This approach allows us to maintain the addition and deletion edges of actions for numeric values. An action that updates a numeric variable is related by an ADD edge to its new returned value and by a DEL edge to its previous value in the multi-valued attribute.
Numeric domains can be infinite and thus, instantiating actions for all possible values become impossible. For this reason, we propose an incremental instantiation of actions for numeric values. To allow incremental instantiation, we introduce the concepts of implicit

parameters and implicit conditions in order to maintain the precondition edges of the actions that relate them to numeric facts.

Definition-3: *A numeric variable that belongs to arguments of an external function is an implicit precondition action parameter.*

Definition-4: *A numeric variable affected by an update function application or assigned to an arithmetic expression is an implicit precondition action parameter.*

Definition-5: *An implicit precondition action parameter is related by a precondition edge to its action.*

This incremental instantiation is done by replacing the numeric variables by their corresponding values from the current fact level, which avoids the flood of numeric values and can be used for discrete and continuous numeric values. Only the variables that are parameters of external functions and the right hand side numeric variables of expressions are instantiated from the current fact level. All these incrementally instantiated variables are added implicitly to the action preconditions by direct edges. These edges to implicit preconditions will allow us to follow the trace of the functions application within the planning graph during the extraction process. Therefore, we run through the planning graph during the extraction of a plan without making difference between implicit and explicit preconditions. This technique allows us to use black-box functions, because we will be only concerned by the parameters and the returned value of a function to be able to follow its trace through edges during the extraction process.

Algorithm-1 (see Fig. 3) starts by calling an initialization subroutine (see. Fig.2) in which the construction of the planning graph begins by setting the first fact level to the initial state of the planning problem, then by testing if the goal conditions are satisfied at the initial state to return an empty plan, otherwise the algorithm instantiates the propositional variables of all the actions of the planning domain.

```
initialization()/* subroutine to initialize the adapted Graphplan algorithm (Fig-2)*/
begin/*initialisation*/
        Stop:= false;        /*condition to stop the graph expansion */
        i:=0;                /*the planning graph level number*/
        Fact_i := S_0;       /*the initial planning graph fact level*/
        Facts := {Fact_i};   /*the collection of all the planning graph fact levels*/
        Actions := {};       /*the collection of all the planning graph action levels*/
        MutF := {};          /*the set of mutual exclusions between facts*/
        MutA := {};          /*the set of mutual exclusions between actions */
        Plan := ∅;
        A :={};   /* the set of ground actions */
        /*test if the goal is satisfied in the initial state*/
        Stop=testForSolution(Facts, Actions, MutF, MutA, G, Plan);
        /*instantiate propositional action variables*/
        for all a ∈ Act do
                Instantiate a over P;
                if con_P(a) = true then A:= A ∪ {a};
        end for
end
```

Fig. 2. Adapted Graphplan initialization subroutine

Algorithm-1: ***Adapted Graphplan with external function application***
Input: S_0: initial state, G: Goal conditions, Act: Set of actions
Output: Plan: sequence of ground actions
begin
 call initialization(); */* a subroutine that initializes variables*/*
 */*planning graph construction iterations*/*
 while (*not* Stop) **do**
 $Action_i$:={};
 for all a ∈ A **do**
 InstNumeric(pre_N(a), $Fact_i$);
 if con_N(a) = *true* in $Fact_i$ **then**
 if pre_P (a) ⊆ $Fact_i$ *and* pre_N (a) *are all true in* $Fact_i$ **then**
 InstNumeric(eff_N (a), $Fact_i$);
 $Action_i$:=$Action_i$ ∪ a ;
 PointPreconditions(a, $Fact_i$);
 end if
 end if
 end for
 Actions := Actions ∪ $Action_i$;
 */*Add the facts of previous level with their "no-op" actions"*/*
 i := i+1;
 $Fact_i$:= $Fact_{i-1}$;
 for *each* f ∈ $Fact_{i-1}$ **do**
 $Action_{i-1}$:=$Action_{i-1}$ ∪ "no-op";
 end for
 */*Apply the applicable instantiated actions*/*
 for *all* a ∈ $Action_{i-1}$ and a ≠ "no-op" **do**
 $Fact_i$:=$Fact_i$ ∪ eff_p^+ - eff_p^- ;
 for *each* e ∈ eff_N *such that* e = (n := g) **do**
 if g is an external function **then**
 call the function g;
 else
 evaluate the expression g;
 end if
 n=n ∪ g; */*add a new value to the muli-valued attribute n*/*
 end for
 Connect a to its added or deleted effects;
 / an updated numeric value is considered as being deleted than added*/*
 end for
 Facts := Facts ∪ $Fact_i$;
 calculate the mutex relations for $Action_{i-1}$ and add them to MutA;
 calculate the mutex relations for $Fact_i$ and add them to MutF;
 nonStop=testForSolution(Facts, Actions, MutF, MutA, G, Plan);
 end while
end

Fig. 3. The adapted Graphplan algorithm to support the application of external functions

Propositional instantiation is done once before the graph expansion by substituting the propositional variables by all the possible combinations of the propositional problem objects and by keeping only the instantiated actions that respect the validity of the list of propositional constraints (con_P list). We call these actions initially instantiated for their propositional objects the partially instantiated actions. We note that, none of the numeric variables are substituted before the planning graph expansion process.
Returning back to algorithm-1 (in Fig. 3) after initialization, now at each iteration *i* of the planning graph expansion, the algorithm instantiates the numerical variables of the partially instantiated actions for the actions that respect the validity of the list of numeric constraints (con_N list) and keeps only the actions that have their preconditions (propositional and numeric) satisfied at the fact level *i-1*. The obtained actions form the set of applicable actions at level *i-1*. The algorithm copies all the facts of level *i-1* to the fact level *i* and adds their corresponding "no-op" actions to the action level *i-1* (one "no-op" for each fact). Each non-"no-op" action of the applicable actions of level *i-1* is applied at the iteration *i*, negative and positive propositional effects are simply added to fact level *i*, and the numeric effects are evaluated by interpreting their arithmetic expressions or by executing their external functions, and then by assigning their returned values to the corresponding numeric variables. After this step, the mutual exclusions between actions of level *i-1* and between actions of level *i* are calculated in the same manner as it is done in the original Graphplan algorithm. At the end of the iteration *i*, the algorithm tests if the goal conditions are satisfied at facts level *i* and there are no mutual exclusions between the facts satisfying the goal, otherwise the algorithm expands a new graph level until finding a valid plan or a sufficient condition (i.e. when the algorithms starts generating duplicated levels in the graph) that indicates that there is no solution to the problem.

In the following we detail the processing of the main functions used by algorithms-1 (Fig. 3):

- **boolean testForSolution**(Facts: the set of all fact levels, Actions: the set of action levels,
  MutF: the set of fact mutexes, MutA: the set of action mutexes,
  G: the set of goal conditions, Plan: ordered set of actions to be returned){

```
        /*this function tests if  G is satisfied in Facts and if a valid plan can be found*/
        if G is satisfied in Facts then
        if Actions = {} then
                Plan:={};
                return true;
        elseif the graph is saturated  then
                Plan:={'failure'};
                return true;
        else // search for a valid plan
                return ExtractPLAN (Facts, Actions, MutF, MutA, G, Plan)
        end if
        end if
        return false;
}
```

- **boolean ExtractPLAN**(Facts: the set of all fact levels, Actions: the set of action levels,
  MutF: the set of fact mutexes, MutA: the set of action mutexes,

G: the set of goal conditions, , Plan: ordered set of actions to be returned){

In backward-chaining start by the set of goals at the last level *n*, find a set of actions ("no-op"s included) at level n-1 that add these goal facts. The preconditions to these actions form a set of subgoals at level *n− 1*. Find the actions at level *n-2* that add these subgoals and so forth until reaching the first level *0*. If at a certain level the subgoals can not be reached then try to find a different set of actions that add these subgoals and continue.
If *level 0 is reached* **then**
Plan:= $\cup_{0,\,n}$ Set of actions that add subgoals;
return true;
end if
//more levels have to be expanded
return false;
}

- **InstNumeric**(N: numeric variables set, Fact: fact level){
  Instantiate the parameters of the external functions and the numeric variables of the expressions in N (e.g. numerical effects of actions) by values from Fact
  (e.g. Numerical fact level items)
}

- **PointPreconditions**(*a*: instantiated action, Fact: fact level){
  Add as preconditions pointers to *a* all the facts from Fact that appear in $pre_P(a)$ as well as in $pre_N(a)$ and in $eff_N(a)$;
  */*A numeric state variable is added implicitly as a precondition to a ground action if it is assigned to an external function or if it appears in an expression or if it is a parameter of an external function of this ground action*/*
}

In figure 4 (see Fig. 4) we show the structure of the planning graph that handles multi-valued numeric variables. The *fact level 0* represents the initial state. It contains two propositional facts "$proposition_1$" and "$proposition_2$" and three numeric variables "n_1", "n_2", "n_3" that have respectively "v_{10}", "v_{20}", "v_{30}" as single values. We note that, at initial state each numeric variable has a single value. At *action level 0*, "*Action1*" and "*Action2*" are considered as having their preconditions satisfied at *fact level 0*. "*Action1*" has 2 satisfied numeric preconditions "n_1= v_{10}" and"n_2=v_{20}". "*Action2*" has one satisfied numeric precondition "n_2=v_{20}" and one satisfied propositional precondition "$proposition_2$" which is true at the current fact level.
Applying the two actions at *level 0*, "*Action1*" is considered to update the values of numeric variables "n_1" and "n_2" respectively to "v_{11}" and "v_{21}" each of which is the result of a function application or an arithmetic expression evaluation. As "*Action1*" modifies the value of "n_1= v_{10}" to "n_1= v_{11}" (respectively the value of "n_2= v_{20}" to "n_2= v_{21}") then it is considered as adding "n_1= v_{11}" (respectively "n_2= v_{21}") and deleting "n_1= v_{10}" (respectively "n_2= v_{20}"). "*Action2*" is considered as adding "$proposition_3$" and deleting "$proposition_2$".

All facts that are pre-existed in the previous fact level (*level 0*) are connected as being added from this level by "no-op" actions to the next level (*level 1*).

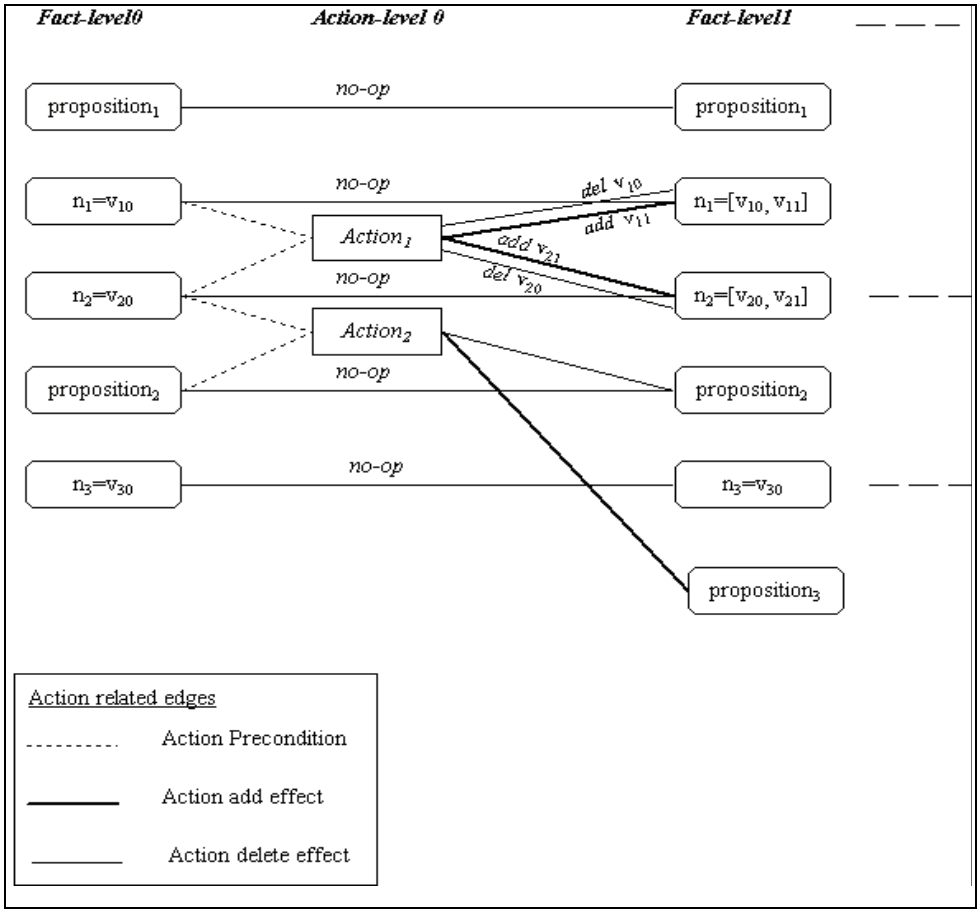


Fig. 4. Adapted Graphplan construction with multi-valued numeric variables

We remark in this figure that, propositional facts are growing horizontally (by line) from level to level and numeric facts are growing vertically (by column). Therefore, the number of numeric variables will not vary during the graph expansion and will stay the same at the last fact level. However, the number of values of each numeric variable can increase with each new expanded fact level.

5. Heuristic search

Solving domain independent planning problems is PSPACE-complete (Bylander, 1994). To reduce this complexity a heuristic function can be used to guide the search for a plan in the search space instead of using a blind search strategy.

The idea of deriving a heuristic function consists of formulating a simplified version of the planning problem by relaxing some constraints of the problem. The relaxed problem can be solved easily and quickly compared to the original problem. The solution of the relaxed problem can then be used as a heuristic function that estimates the distance to the goal in the original problem.
The most common relaxation method used for propositional planning is to ignore the negative effects of actions. This method was originally proposed in (McDermott, 1996) and (Bonet et al., 1997) and then used by the most of propositional heuristic planners (Bonet & Geffner, 2001; Hoffman, 2001; Refanidis & Vlahavas, 2001).
With the arising of planners that solve problems with numerical knowledge such as metric resources and time, a new relaxation method has been proposed to simplify the numerical part of the problem. As proposed in Metric-FF (Hoffmann, 2002) and SAPA (Do & Kambhampati, 2001), relaxing numerical state variables can be achieved by ignoring the decreasing effects of actions. This numerical relaxation has been presented as an extension to the propositional relaxation to solve planning problems that contain propositional and numeric knowledge.
Knowing that, some planning problems contains actions that strictly increase or decrease numeric variables without alternation, other problems uses numeric variables to represent real world objects that have to be handled according to their quantity (Zalaket & Camilleri, 2004a) and thus applying the above proposed numerical relaxation method can be inadmissible to solve this kind of problems. In this section, we start by explaining the relaxed propositional task as it was proposed for STRIPS problems (McDermott, 1996), we introduce a new relaxation method for numerical tasks in which we relax the numeric action effects by ignoring the effects that move away numeric variable values from their goal values, then we present the calculation of a heuristic function using a relaxed planning graph over which we apply the above relaxation methods, and finally we present the use of the obtained heuristic to guide the search for a plan in a variation of hill-climbing algorithm.

5.1 Propositional task relaxation

Relaxing a propositional planning task can be obtained by ignoring the negative effects of actions.
Definition-6: *Given a propositional planning task $P=<S, A, s_I, G>$, the relaxed task P' of P is defined as $P'=<S, A', s_I, G>$, such that: $\forall a \in A$ and $eff_P(a) = eff_P^+(a) \cup eff_P^-(a) \Rightarrow \exists a' \in A' / \; eff_P(a') = eff_P^+(a)$(which means $eff_P(a') = eff_P(a) - eff_P^-(a)$).*
And thus, $A' = \{ con_P(a), Pre_P(a), eff_P^+(a) , \forall a \in A \}$.
The relaxed plan can be solved in polynomial time as it is proven by bylander (Bylander, 1994).

5.2 Numerical task relaxation

Relaxing a numerical planning task can be obtained by ignoring the negative effects of actions that move away numeric values from the goal values.
Definition-7: *Given a numerical planning task $V=<S, A, s_I, G>$, the relaxed task V' of V is defined as $V'=<S, A', s_I, G>$, such that: $\forall a \in A$ and $eff_N(a) = eff_N^+(a) \cup eff_N^-(a)$, such that:*
$\forall (n:=v) \in eff_N(a)$, where n is a numeric variable and v is a constant numeric value that can be the result of an arithmetic expression or an executed external function.
Positive numeric effects $eff_N^+(a)$ and negative numeric effects $eff_N^-(a)$ are defined as follows:

$\forall$ ***(n=v_I) $\in s_I$,*** *where v_I is a constant numeric value that represents the initial value of n.*

if

(n θ v_G) $\in$ G, *where $\theta \in \{<, \leq, =, >, \geq\}$ and v_G is a constant numeric or the result of an arithmetic expression or an executed external function.*

if

distance (v, v_G,) $\leq$ distance (v_I, v_G) and distance (v_I, v) $\leq$ distance (v_I, v_G) *(the current value v of the numeric variable n is closer to the goal value v_G of n than the initial value v_I from the initial value side.)*

then

(n:=v) $\in$ eff_N^+ (a)

else

(n:=v) $\in$ eff_N^- (a)

endif

else

(n:=v) $\in$ eff_N^+ (a) *//(n does not appear in the goal state)*

end if

Example of the distance calculation:

Assume that:

- We have a numeric variable n which is equal to *0* at the initial state (v_I=*0*) and is equal to *5* in the goal state (v_G=*5*).
- We have an action *a*, which assigns to n respectively the values v_1=*-3*, v_2=*-1*, v_3=*1*, v_4=*5*, v_5=*7*, v_6=*11*.

In this case the distance can be calculated as: $distance\ (v_j, v_i)=|v_j - v_i|$

By testing for relaxed action effects:

$distance(v_I, v_G)= |v_G - v_I|=5$

- ***v_1=-3****: $distance(v_1, v_G)=|v_G - v_1|=8 > distance(v_I, v_G) \Rightarrow (v_1=-3) \in eff_N^-$ (a) $\Rightarrow v_1$=-3 is ignored in the relaxed task.*
- ***v_2=-1****: $distance(v_2, v_G)=|v_G - v_2|=6 > distance(v_I, v_G) \Rightarrow (v_2=-1) \in eff_N^-$ (a) $\Rightarrow v_2$=-1 is ignored in the relaxed task.*
- ***v_3=1****: $distance(v_3, v_G)=|v_G - v_3|=4 \leq distance(v_I, v_G)$ and $distance(v_I, v_3)=|v_3 - v_I|=1 \leq distance(v_I, v_G) \Rightarrow (v_3= 1) \in eff_N^+$ (a) $\Rightarrow v_3$=1 is held in the relaxed task.*
- ***v_4=5****: $distance(v_4, v_G)=|v_G - v_4|=0 \leq distance(v_I, v_G)$ and $distance(v_I, v_4)=|v_4 - v_I|=5 \leq distance(v_I, v_G) \Rightarrow (v_4= 4) \in eff_N^+$ (a) $\Rightarrow v_4$=4 is held in the relaxed task.*
- ***v_5=7****: $distance(v_5, v_G)=|v_5 - v_G|=2 \leq distance(v_I, v_G)$, but $distance(v_I, v_5)=|v_5 - v_I|=7 > distance(v_I, v_G) \Rightarrow (v_5= 7) \in eff_N^-$ (a) $\Rightarrow v_5$=7 is ignored in the relaxed task.*
- ***v_6=11****: $distance(v_6, v_G)=|v_6 - v_G|=6 > distance(v_I, v_G) \Rightarrow (v_6= 11) \in eff_N^-$ (a) $\Rightarrow v_6$=11 is ignored in the relaxed task.*

Remarks:

The distance formula can vary according to the comparison operator used in the goal state, but it is the same for all numeric values used in the initial and the goal state.

Each numeric variable that appears in the initial state and doesn't appear in the goal conditions is automatically added to the positive numeric effects, because the values of these variables are often used as preconditions for actions and thus, they can not be ignored.

Figure 5 (see Fig. 5) shows (in red) how negative numeric effects of an action that updates a numeric variable *n* are considered. It also shows (in blue) the positive numeric effects of the action which are considered according to the initial and the goal values of the variable *n*. Note that, exchanging the values of n between initial and goal states will not affect the ranges of selected positive and negative numeric effects.

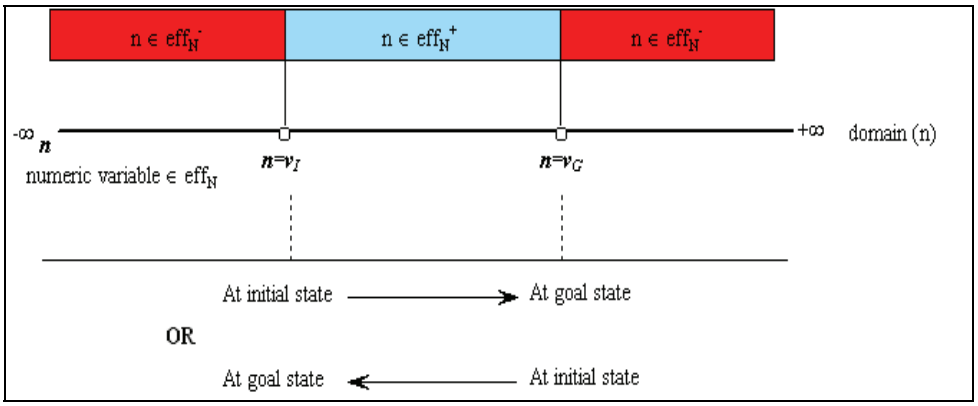


Fig. 5. Choosing negative and positive numeric action effects

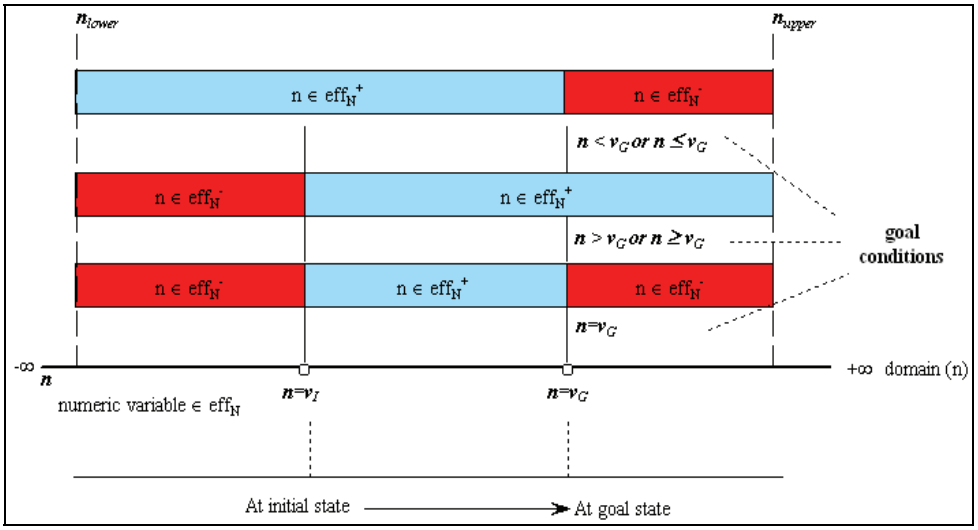


Fig. 6. Numeric relaxed action effects variation according to goal comparison operators.

Figure 6 (see Fig. 6) shows how the selection of negative (in red) and positive (in blue) numeric effects depends on the comparison operator used for comparing the numeric variable *n* in the goal conditions. Therefore, the distance formula is calculated according to the operator used irrespective of the values of n in initial and goal states. As can be observed in this figure, a tighter range of positive numeric effects can be obtained when the equal

operator is used to compare the value of *n* in the goal conditions, and consequently a smaller search space will be generated for the relaxed problem, which accelerates the process of search for a plan for that problem.

5.3 Mixed planning problem relaxation

Definition-8: *Given a mixed propositional and numerical planning problem* $P=<S, A, s_I, G>$*, the relaxed problem* P' *of* P *is defined as* $P'=<S, A', s_I, G>$*, such that:* $\forall a \in A$ *and* $eff(a)= eff_P(a) \cup eff_N(a)$ *and* $eff_P(a) = eff_P^+(a) \cup eff_P^-(a)$ *and* $eff_N(a) = eff_N^+(a) \cup eff_N^-(a) \Rightarrow \exists a' \in A' / \quad eff(a')= eff_P^+(a) \cup eff_N^+(a)$.

And thus, $A' = \{ con_P(a), Pre(a), eff^+(a) = eff_P^+(a) \cup eff_N^+(a), \forall a \in A \}$.

Definition-9: *A sequence of applicable actions* $\{a_1, a_2, \ldots, a_n\}$ *is a relaxed plan for the planning problem* $P=<S, A, s_I, G>$ *if* $\{a'_1, a'_2, \ldots, a'_n\}$ *is a plan of its relaxed problem* $P'=<S, A', s_I, G>$.

6. Relaxed planning graph with functions application

Like the planning graph structure used in the adapted Graphplan algorithm, the relaxed planning graph consists of 2 types of levels fact levels and action levels. Algorithm-2 (see Fig. 7) shows how the relaxed planning graph is expanded until reaching a fact level that satisfied the goal conditions or until obtaining consecutive duplicated fact levels. This test is done by using the function *testForSolution(Facts, Actions, G, Plan)*, which will be modified compared to the one used in the adapted Graphplan implementation (Fig. 3).

Compared to algorithm-1 (Fig. 3), algorithm-2 (Fig. 7) applies only the positive propositional and numeric effects of actions for generating the next fact level, as discussed in sextion-5. An additional relaxation is added to the planning graph construction in algorithm-2, which consists of ignoring the mutual exclusion between facts and between actions. Therefore, the initialization subroutine for algorithm-2 will be the same as in Fig. 2 but without the mutual exclusion lists. This latter relaxation allows the relaxed planning graph to apply conflicting actions in parallel, and thus to reach the goal state faster in polynomial time.

The test for solution

- **boolean testForSolution**(Facts: the set of all fact levels, Actions: the set of action levels, G: set of goal conditions, Plan: ordered set of the actions to be returned){

```
/*this function tests if G is satisfied in Facts and if a relaxed plan can be found*/
if G is satisfied in Facts then
if Actions = {} then
        Plan:={};
                return true;
elseif the graph is saturated then
        Plan:={'failure'};
                return true;
elseif G is satisfied at Facts[final_level] then
                // extract a relaxed plan see algorithm-3
        ExtractRelaxedPLAN (Facts, Actions, G, Plan)
end if
end if
return false;
}
```

Algorithm-2: ***Relaxed planning graph with external function application***
Input: S_0: initial state, G: Goal conditions, Act: Set of actions

Output: Plan: sequence of ground actions

```
begin
        call initialization(); /* a subroutine that initializes variables*/
        /*relaxed planning graph construction iterations*/
        while (not Stop) do
                Action_i:={};
                for all a ∈ A do
                    InstNumeric(pre_N(a), Fact_i);
                    if con_N(a) = true in Fact_i then
                        if pre_P (a) ⊆ Fact_i and pre_N (a) are all true in Fact_i then
                                InstNumeric(eff_N (a), Fact_i);
                                Action_i:=Action_i ∪ a ;
                                PointPreconditions(a, Fact_i);
                        end if
                    end if
                end for
                Actions := Actions ∪ Action_i;
                /*Add the facts of previous level with their "no-op" actions"*/
                i := i+1;
                Fact_i := Fact_i-1;
                for each f ∈ Fact_i-1 do
                        Action_i-1:=Action_i-1 ∪ "no-op";
                end for
                /*Apply the applicable positive instantiated actions*/
                for all a ∈ Action_i-1 and a ≠ "no-op" do
                        Fact_i:=Fact_i ∪ eff_p+
                        for each e ∈ eff_N+ do
                                if g is an external function then
                                        call the function g;
                                else
                                        evaluate the expression g;
                                end if
                                /*add a new value to the muli-valued attribute n*/
                                n=n ∪ g;
                        end for
                        Connect a to its added effects;
                end for
                Facts := Facts ∪ Fact_i;
                nonStop=testForSolution(Facts, Actions, G, Plan);
        end while
end
```

Fig. 7. The relaxed planning graph construction algorithm

6.1 Relaxed plan extraction

Once the relaxed planning graph is constructed using the algorithm-2 (Fig. 7) up to a level that satisfies the goals, the extraction process can be applied in backward chaining as shown in algorithm-3 (Fig. 8) which details the *ExtractRelaxedPLAN* function called by the *testForSolution* function of in algorithm-2 as detailed in section-6:

```
Algorithm-3: Extract plan in backward chaining from the relaxed planning graph
Name: ExtractRelaxedPlan

Input: Facts: Set of fact levels, G: Goal conditions, Actins: Set of action levels

Output: Plan: sequence of ground actions
begin
    Plan:={};
    G_final_level:= { g ∈ Facts[final_level] / g satisfies G};
    for i = final_level to 1 do
            G_i-1:={};
            for each g ∈ G_i do
                acts:= {actions at level final_level-1 that add g};
                selAct:= get_first_element_of(acts);
       if act ≠ "no-op" then
            for act ∈ acts do
                        if act= 'no-op' then
                    selAct:=act;
                    break;
                end if
                // Select the action that has the minimum number of preconditions
                if nb_preconditions_of(act)< nb_preconditions_of (selAct) then
                    selAct:=act;
                end if
            end for
       end if
       plan:=plan ∪ selAct;
       G_i-1:= G_i-1 ∪ { f ∈ Facts[i-1] s.t. f is a precondition of selAct};
            end for
    end for
end
```

Fig. 8. Plan extraction from a relaxed planning graph

Each sub-goal in the final fact level (the level that satisfies the goal conditions), is replaced by the preconditions and by the implicit preconditions (definitions 3 and 4) of the action that adds it and the action is added to the list of relaxed plan. Normally, a "no-op" action will be preferred if it adds a sub-goal. If there is not a "no-op" action that adds the sub-goal and there is more than one action that add it, then we choose the action that has the minimum number of preconditions and implicit preconditions from these latter. We replace the sub-goal fact by the facts that serve as preconditions and implicit preconditions to the chosen

action. We can backtrack in the graph to choose another action adding the sub-goal if a selected action doesn't lead to a solution. Once all goals of the final level are replaced by the sub-goals of previous level, this previous level becomes the final level and the sub-goals become the new goals. This process is repeated until reaching the first fact level. The resulting heuristic is considered as the distance to the goal and it is calculated by counting the number of actions of the relaxed plan.

$h = \sum_{i=0}^{final\_level-1} | a_i |$, where $[a_0, a_1, ..., a_{final\_level-1}]$ is the relaxed plan.

Note that, during the backward plan extraction, we don't make any difference between numeric and propositional facts as all facts even that are results of applied functions are accessed via action edges that are stored in the planning graph structure.

6.2 Heuristic planner running over the effects of applied functions

The main search algorithm that we use to find a plan in the original problem is a variation of hill-climbing search guided by the heuristic *h* detailed in section-6.1. The heuristic is calculated for each state *s* in the search space. At each step we select the child having the lowest heuristic value compared to other children of the same parent to be the next state step, and so on until we reach a state with a heuristic equal to zero. If at some step, algorithm-2 doesn't find a relaxed plan that leads a state *s* to the goal state then the heuristic *h* will be considered as infinite at this step.

Each time a state is selected (except of the initial state) the action which leads to this selected state is added to the plan list. The variation of hill-climbing is when a child having the lowest heuristic is selected, if its heuristic value is greater than the parent state heuristic then the child can be accepted to be the next state step as long as the total number of children exceeding the parent heuristic value is less than a given threshold number. Another variation of hill climbing is: The number of consecutive plateaus (where the calculated heuristic value stays invariable) is accepted up to a prefixed constant. After that a worst-case scenario is launched. This scenario consists of selecting the child who has the lowest heuristic greater than the current state heuristic (invariable), and then to continue the search from this children state by trying to escape the plateau. This scenario can also be repeated up to a prefixed threshold.

In all the above cases, if hill-climbing exceeds one of the quoted thresholds or when the search fails to find a plan the hill-climbing is considered as unable to find a solution and an A* search begins. As HSP and FF, we have added to hill climbing search and to A* search a list of visited states to avoid calculating a heuristic more than once for the same state. At each step a generated state is checked to see if it exists in the list of visited states in order cut it off to avoid cycles. According to our tests, we have noticed that most of the problems can be solved with hill-climbing algorithm. Only some tested domain problems (like ferry with capacity domain) have failed with hill-climbing search so early. But, the solution has been found later with the A* search.

7. Empirical results

We have implemented as prototypes all the above algorithms in Java language. We have run these algorithms over multiple foremost numeric domains that necessitate non classical

handling such as the water jugs domain, the manufacturing domain, the army deployment domain and the numeric ferry domain as introduced in (Zalaket & Camilleri 2004a). We note that, some of these domains such as manufacturing and army deployment are usually expressed and solved with scheduling or with mathematical approach.
Our tests can be summarized in three phases: In the first phase, we have started by running a blind forward planning algorithm that supports the execution of external functions. Our objective at this phase was only to study the feasibility and the effectiveness of integrating such functions written in a host programming language to planning in order to accomplish some complex computation. In the second phase, we have run the adapted Graphplan algorithm with which we have obtained optimal plans for all the problems, but it was not able to solve large problems. In the third phase, we have run the heuristic planner over all the above cited domains. Larger problems are solved with this planner, but the generated plans were not always optimal as it was the case in the second phase.
We have made minor efforts for optimizing our implementation in the one or the other of the above phases. Even though, we can conclude that the heuristic algorithm is the most promising one despite its non-optimal plans. We think that some additional constraints can be added to this algorithm to allow it generating better plans quality. We also remark that some planning domains can be modelled numerically instead of symbolically to obtain extremely better results. For example, in the numeric ferry domain the heuristic algorithm was able to solve problems that move hundreds of cars instead of tenth with classical propositional planners.

8. Conclusion

In this chapter, we have presented multiple extensions for classical planning algorithms in order to allow them to solve more realistic problems. This kind of problems can contain any type of knowledge and can require complex handling which is not yet supported by the existing planning algorithms. Some complicated problems can be expressed with the recent extensions to PDDL language, but the main lack remains especially because of the incapacity of the current planners. We have suggested and tested the integration to planning of external functions written in host programming languages. These functions are useful to handle complicated tasks that require complex numeric computation and conditional behaviour. We have extended the Graphplan algorithm to support the execution of these functions. In this extension to GraphPlan, we have suggested the instantiation of numeric variables of actions incrementally during the expansion of the planning graph. This can restrict the number of ground actions by using for numeric instantiation only the problem instances of the numeric variables instead of using all the instances of the numeric variable domain which can be huge or even infinite. We have also proposed a new approach to relax the numeric effects of actions by ignoring the effects that move away the values of numeric variables from their goal values. We have then used this relaxation method to extract a heuristic which we have used it later in a heuristic planner.
According to our tests on domains like the manufacturing one, we conclude that scheduling problems can be totally integrated into AI planning and solved using our extensions. As future work, we will attempt to test and maybe customize our algorithms to run over some domains adapted from the motion planning, in order to extend the AI planning to also cover

the motion planning and other robotic problems currently solved using mathematical approaches.

9. References

Bacchus, F. & Ady, M. (2001). Planning with resources and concurrency a forward chaining approach. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, August, 2001, Seattle, Washington, USA

Bak, M.; Poulsen, N. & Ravn, O. (2000). Path following mobile robot in the presence of velocity constraints. *Technical report*, Technical University of Denmark, 2000, Kongens Lyngby, Denmark

Blum, L. & Furst, L. (1995). Fast planning through planning graph analysis. *Proceedingsof the 14th International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1636–1642, August, 1995, Montreal, Quebec, Canada

Bonet, B. & Geffner, H. (2001). Planning as heuristic search. *Journal of Artificial Intelligence*, 129:5–33,2001

Bonet, B.; Loerincs, G. & Geffner, H.(1997). A robust and fast action selection mechanism for planning. *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, pages 714–719, July, 1997, convention center in Providence, Rhode Island

Bresina, L. J.; Dearden, R.; Meuleau, N; Smith, E. D. & Washington, R. (2002) Planning Under Continuous Time and Resource Uncertainty: A Challenge for AI. *Proceedings of the AIPS Workshop on Planning for Temporal Domains*, pages 91–97, April, 2002, Toulouse, France

Bylander, T. (1994). The computational complexity of propositional strips planning. *Journal of Artificial Intelligence*, 69:165–204, 1994

Cayrol, M. ; Régnier, P. & Vidal, V. (2000). New results about LCGP, a least committed graphplan. *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000),*pages 273–282, 2000, Breckenridge, CO, USA

Do, B. & Kambhampati, S. (2000). Solving planning graph by compiling it into a CSP. *Proceedings of the 5th International Conference on Artificial Intelligence Planning and Scheduling (AIPS-2000)*, 2000, Breckenridge, CO, USA

Do, B. & Kambhampati, S. (2001). Sapa: A domain-independent heuristic metric temporal planner. *Proceedings of the 6th European Conference on Planning (ECP 2001)*, September, 2001, Toledo, Spain

Edelkamp (2002). Mixed propositional and numerical planning in the model checking integrated planning system. *Proceedings of the AIPS Workshop on Planning for Temporal Domains*, April, 2002, Toulouse, France

Fikes, R.E. & Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Journal of Artificial Intelligence*, 2:189–208, 1971.

Fox, M. & Long, D. (2002). PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Proceedings of the 7th International Conference on Artificial Intelligence Planning and Scheduling (AIPS- 2002)*, April, 2002, Toulouse, France

Geffner, H. (1999). Functional strips: A more flexible language for planning and problem solving. *Logicbased AI Workshop,* June, 1999, Washington D.C.

Gerevini, A. & Long, D. (2005). Plan constraints and preferences for PDDL3. *Technical Report Technical report*, R.T. 2005-08-07, Dept. of Electronics for Automation, 2005, University of Brescia, Brescia, Italy

Ghallab, M.; Howe, A.; Knoblock, G.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D. & Wilkins, D. (1998). PDDL : The planning domain definition language, version 1.2. *Technical Report CVC TR-98 003/DCS TR-1165. Yale Center for Computational Vision and Control*, October, 1998, Yale, USA

Hoffman, J. (2001) FF: The fast-forward planning system. *AI Magazine*, 22:57 – 62, 2001.

Hoffmann, J. (2002). Extending FF to numerical state variables. *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI2002),* pages : 571-575, July, 2002, Lyon, France

Hoffmann, J.; Kautz, H.; Gomes, C. & Selman B. (2007). SAT encodings of state-space reachability problems in numeric domains. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07),* pages 1918– 1923, January, 2007, Hyderabad, India

McDermott, D. (1996). A heuristic estimator for means ends analysis in planning. *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems.* May, 1996, Edinburgh, UK.

Refanidis, I. & Vlahavas, I. (2001). The GRT planning system: Backward heuristic construction in forward state-space planning. *Journal of Artificial Intelligence Research*, 15:115–161, 2001.

Samson, C. & Micaelli, A. (1993). Trajectory tracking for unicycle-type and Two steering-wheels mobile robots. *Technical report*, Institut National de Recherche en Automatique, 1993, Sophia-Anitpolis, France

Schmid, U.; Müller, M. & Wysotzki, F. (2002). Integrating function application in state based planning. *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, pages 144 – 162, September 2002, Aachen, Germany.

Smith, D. & Weld, D. (1999). Temporal planning with mutual exclusion reasoning. *Proceedings of 16th InternationalJoint Conference on Artificial Intelligence (IJCAI-99)*, August, 1999, Stockholm, Sweden

Zalaket, J. & Camilleri, G. (2004a). FHP : Functional heuristic planning. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES 2004)*, pages 9–16, September, 2004, Wellington, New Zealand

Zalaket, J. & Camilleri, G. (2004b). NGP: Numerical graph planning. *proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004),* pages 1115–1116, August, 2004, Valencia, Spain.

23

Network Optimization as a Controllable Dynamic Process

Alexander Zemliak
Puebla Autonomous University
Mexico
National Technical University of Ukraine "KPI"
Ukraine

1. Introduction

One of the basic challenges in the design of large systems is how to reduce time spent to attain the optimal point of the objective function of the design process. The design process itself includes optimization of the structure of the future system, but since this stage is related to an artificial intelligence problem still unresolved, in the general case it is performed "by hand", and thus is absent in the CAD systems. In other words, the traditional approach to computer-aided design consists of two main parts: a model of the system set up in the form of a network described by some algebraic or integro-differential equations, and the parametric optimization procedure – to seek the optimum of the objective function corresponding to the sought characteristics of the system under design.
There are some powerful methods that reduce the necessary time for the circuit analysis. Because a matrix of the large-scale circuit is a very sparse, the special sparse matrix techniques are used successfully for this purpose (Osterby & Zlatev, 1983). Other approach to reduce the amount of computational required for the linear and nonlinear equations is based on the decomposition techniques. The partitioning of a circuit matrix into bordered-block diagonal form can be done by branches tearing as in (Wu, 1976), or by nodes tearing as in (Sangiovanni-Vincentelli et al., 1977) and jointly with direct solution algorithms gives the solution of the problem. The extension of the direct solution methods can be obtained by hierarchical decomposition and macromodel representation (Rabat et al., 1985). Other approach for achieving decomposition at the nonlinear level consists on a special iteration techniques and has been realized for example in (Ruehli et al., 1982; George, 1984) for the iterated timing analysis and circuit simulation. Optimization technique that is used for the circuit optimization and design exert a very strong influence on the total necessary computer time too. The numerical methods are developed both for the unconstrained and for the constrained optimization (Fletcher, 1980; Gill et al., 1981). The practical aspects of use of these methods are developed for VLSI circuit design, yield, timing and area optimization (Brayton et al., 1981; Ruehli, 1987; Massara, 1991). It is possible to suppose that the circuit analysis methods and the optimization procedures will be improved later on. Meanwhile, it is possible to reformulate the total design problem and generalize it to obtain a set of different design strategies. It is clear that a finite but a large number of different strategies

include more possibilities for the selection of one or several design strategies that are time-optimal or quasi-time-optimal ones. This is especially right if we have infinite number of the different design strategies.
The time required for optimization grows rapidly as the system complexity increases. The known measures of reduction of the time for system analysis (in the traditional approach) turned out to be insufficiently advanced.
By convention, the generally accepted ideas of network design will be called the traditional strategy of design, meaning that the method of analysis is based on Kirchhoff's laws. A new formulation of the network optimization problem without strict adherence to Kirchhoff's laws was suggested in (Kashirskiy, 1976; Kashirskiy & Trokhimenko, 1979). This process was called the generalized optimization and used the idea of ignoring Kirchhoff's laws for the whole network or some part of it. In this case, apart from minimization of the previously defined objective function, we also had to minimize the residual of the equation system describing the network model. In the extreme case, when the residual function included all equations of the network mathematical model, this idea was practically implemented in two CAD systems (Rizzoli et al., 1990; Ochotta et al., 1996). The authors of these works asserted that overall time of design was reduced considerably. This latter idea may be termed the modified traditional design strategy. As distinct from the traditional approach proper, including network model analysis at every step of the optimization procedure, the modified traditional strategy of deign may be defined as a strategy which does not include at all the model analysis in the process of optimization.
Another formulation of the network design problem based on the idea proposed in (Zemliak, 2001) can be introduced by generalization and formulation of this idea to obtain a set of different design strategies. Here we may pass to the problem of selecting, among this set, a strategy optimal in some sense – for instance, from the running-time viewpoint. Then the optimal strategy of design may be defined as a strategy permitting us to reach the optimal point of the objective function in minimal time. The main issue in this definition is what conditions have to be fulfilled to construct the algorithm providing for the optimal time. The answer to this question will make it possible to reduce substantially the computer time necessary for the design.

2. Problem formulation

By the traditional design strategy we mean the problem of design of an analogue network with a given topology based on the process of unconditional minimization of an objective function $C(X)$ in a space R^K, where K is the number of independent variables. Simultaneously, we are seeking the solution to a system of M dependent on some components of the vector X. It is assumed that the physical model can be described by a system of nonlinear algebraic equation:

$$g_j(X) = 0, \quad j = 1,2,\ldots, M \tag{1}$$

The vector $X \in R^N$ is broken into two parts: $X = (X', X'')$, where the vector $X' \in R^K$ is the vector of independent variables, the vector $X'' \in R^M$, is the vector of dependent

variables and $N = K + M$. This partition into independent and dependent variables is a matter of convention, because any parameter may be considered independent or dependent. Due to such definition, some parameters of the design process, for example, frequency, temperature, etc., are beyond our consideration. We can easily include them in the general design procedure, but here we presume them to be constant and include them the coefficients of system (1).

In the general case, the process of minimization of the objective function $C(X)$ in the space R^K of independent variables for the two-step procedure can be described by the following vector equation:

$$X'^{s+1} = X'^{s} + t_s \cdot H^s \tag{2}$$

where *s* is the iterations number, t_s is an iteration parameter, $t_s \in R^1$, and *H* is a function establishing the direction of lowering the objective function $C(X)$. The constraints for independent variables can be bypassed easily, which is shown in the examples given in the second part of this work.

A particular feature of the design process, at least for electronic network applications, is that we are not obliged to fulfill conditions (1) at every step of the optimization procedure. It is sufficient to satisfy these conditions at the final point of the design process. In this event the vector function H depends on the objective function $C(X)$ and on some additional penalty function $\varphi(X)$, whose structure includes all the equations of system (1) and can be defined, for instance, as:

$$\varphi\left(X^s\right) = \frac{1}{\varepsilon}\sum_{j=1}^{M} g_j^2\left(X^s\right) \tag{3}$$

In this case we define the design process as an unconditional optimization problem:

$$X^{s+1} = X^{s} + t_s \cdot H^s \tag{4}$$

in the space R^N without any additional system of constraints, but for a new objective function $F(X)$, which can be defined, for instance, as an additive function:

$$F(X) = C(X) + \varphi(X) \tag{5}$$

Then at the point of minimum of the objective function $F(X)$ we also have the minimum of the objective function $C(X)$, and system (1) is satisfied at the final point of the optimization process. This method can be called the modified traditional method of design: it reproduces a different strategy of design and a different trajectory in the space R^N.

On the other hand, we can generalize the idea of using of an additional penalty function, if the penalty function is formed only from a part of system (1) while the remaining part is regarded as a system of constraints. In this event the penalty function includes, for example, only Z first terms of $\varphi\left(X^{s}\right)=\frac{1}{\varepsilon}\sum_{i=1}^{Z} g_i^2\left(X^{s}\right)$, where $Z\in[0,M]$ and other M - Z equations form, instead of (1), a modified system make up one modification of the system (1):

$$g_j(X)=0, \quad j=Z+1,Z+2,\ldots,M \tag{6}$$

Obviously, every new value of the parameter Z generates a new design strategy and a new trajectory. This notion can be easily extended to a situation when the penalty function $\varphi(X)$ includes Z arbitrary equations of system (1). The overall number of different design strategies in this case equals 2^M. All these strategies exist within the same optimization procedure. The optimization procedure is realized in the space R^{K+Z}. The number of dependent parameters M grows together with complexity of the system while the number of different design strategies grows by exponential law. These strategies are characterized by different numbers of operations and different overall running time. Accordingly, we may formulate the problem of searching for the design strategy optimal in time, i.e., having a minimum running time of the processor.

Let us estimate the number of operations for several design strategies. The traditional design strategy includes two systems of equations. To be specific, assume that the optimization procedure is based on a gradient method and can be defined by a system of ordinary differential equations for independent variables in the form

$$\frac{dx_i}{dt}=-b\cdot\frac{\delta}{\delta x_i}C(X), \quad i=1,2,\ldots,K \tag{7}$$

where b is the iterative parameter. The operator $\frac{\delta}{\delta x_i}$ means that

$$\frac{\delta}{\delta x_i}\varphi(X)=\frac{\partial\varphi(X)}{\partial x_i}+\sum_{p=K+1}^{K+M}\frac{\partial\varphi(X)}{\partial x_p}\frac{\partial x_p}{\partial x_i}.$$

Nevertheless, the use of the gradient method does not mean loss of generality of the results obtained. It is necessary only that we represent the optimization process as a system of ordinary differential equations for independent variables. The mathematical model of an electronic system in this case is a system of constraints and is described by equation (1). The number of operations for solution of system (1) by Newton's method will be $S\cdot\left[M^3+M^2(1+P)+MP\right]$, where P is the average number of operations for

calculation of $g_j(X)$, and S is the number of iterations in Newton's method for resolving system (1). The number of operations in a single step of integration of system (7) by Newton's method is $K + C \cdot (1+K) + (1+K) \cdot S \cdot \left[M^3 + M^2(1+P) + MP \right]$, where C is the number of operations for calculation of the objective function. The overall number of operations for resolving the problem (1) and (7) by Newton's method, i.e.,

$$N_1 = L_1 \left\{ K + (1+K) \left\{ C + S \cdot \left[M^3 + M^2(1+P) + MP \right] \right\} \right\} \tag{8}$$

where L_1 is the overall number of steps in the optimization algorithm.
The modified traditional strategy of design is fully defined by the equation system of the optimization procedure without any additional limitations. In this case the number of independent variables equals $K+M$. The fundamental system has the form

$$\frac{dx_i}{dt} = -b \cdot \frac{\delta}{\delta x_i} F(X), \; i = 1,2,\ldots, K+M \tag{9}$$

where $F(X)$ is the generalized objective function: $F(X) = C(X) + \frac{1}{\varepsilon} \sum_{j=1}^{M} g_j^2(X)$.

The overall number of operations for resolving (9) is

$$N_2 = L_2 \left\{ K + M + (1+K+M) \cdot \left[C + (P+1) M \right] \right\} \tag{10}$$

A more general strategy of design can be defined as a strategy having a variable number of independent parameters equal to $K+Z$. Here we use two systems of equations, (6) and (11):

$$\frac{dx_i}{dt} = -b \cdot \frac{\delta}{\delta x_i} F(X), \; i = 1,2,\ldots, K+Z \tag{11}$$

where $F(X) = C(X) + \frac{1}{\varepsilon} \sum_{j=1}^{Z} g_j^2(X)$.

Then the overall number of operations N_3 for resolution of systems (6) and (11) can be evaluated as:

$$N_3 = L_3 \{ K + Z + (1+K+Z) \{ C + (P+1)Z + S \cdot [(M-Z)^3 + (M-Z)^2(1+P) + (M-Z)P] \} \} \tag{12}$$

This formula turns into (8) if *Z=0*, and into (10) when *Z=M*. Analysis of the number of operations N_3 as a function of the parameter *Z* permits us to find the conditions for defining the strategy characterized by a minimum running time. For linear system (1), in the Newton's method the number of iterations *S*=1, and the traditional approach is optimal, but for a nonlinear system this is not the case. Also, we assume that the iterations number L_3 and the number of operations *C* for objective function calculation depend on the number of independent parameters as $L_3 = L_0 \cdot (K+Z)^n$, $C = C_0 \cdot (K+Z)^m$. This assumption may be considered trivial, but the main difficulty consists in indeterminacy of the powers *n* and *m*. On the other hand, the number of iterations *S* in Newton's method does not depend, to a first approximation, on the order of system (12), and represents some constant S_0. In practice, to obtain the accuracy $\delta = 10^{-10} - 10^{-12}$, this constant value is within 4 - 5. The average number of operations *P* for calculation of the function $g_j(X)$ is invariant to Z in the case of analysis of an electronic system. This is true, because the conductance matrix of an electronic network is sparse. We assume that this number of operations is constant and equal to P_0. Then expression (12) for calculation of the function $N_3(Z)$ can be reduced to the form

$$\begin{aligned} N_3(Z) = L_0 \cdot (K+Z)^n \cdot \{K+Z+(1+K+Z)[C_0 \cdot (K+Z)^m + Z(1+P_0)] \\ + S_0 \cdot ((M-Z)^3 + (M-Z)^2(1+P_0) + (M-Z)P_0)]\} \end{aligned} \tag{13}$$

In conformity with the fundamental definition of optimal design strategy, we can find this strategy from analysis of this formula. We have to find the optimum point Z_{opt}, in which the function $N_3(Z)$ has a minimum value. If the case of Z_{opt} =*0*, the traditional strategy is optimal. If Z_{opt} =*M*, the modified traditional strategy is optimum. If Z_{opt} is confined within (*0*, *M*) interval, then some intermediate strategy is optimal. The derivative of the function $N_3(Z)$ defined by equation (13) is defined by the formula:

$$\begin{aligned} N_3'(Z) = L_0 n(K+Z)^{n-1}\{K+Z+(1+K+Z)[C_0(K+M)^m + Z(1+P_0) + S_0((M-Z)^3 + (M-Z)^2(1+P_0) \\ + (M-Z)P_0)]\} + L_0(K+Z)^n\{1+C_0(K+M)^m + (1+K+2Z)(1+P_0) + S_0[(M-Z)^3 \\ + (M-Z)^2(1+P_0) + (M-Z)P_0 - (1+K+Z)(3(M-Z)^2 + 2(M-Z)(1+P_0) + P_0)]\} \end{aligned}$$

To ensure that the optimal point lies within the [*0*, *M*] interval, it is necessary and sufficient to fulfill the following two conditions for the derivative at the interval boundaries: $N_3'(0) < 0$ and $N_3'(M) > 0$. It is expedient to introduce an additional parameter $q = \frac{M}{K}$. Then the value of derivative $N_3'(0)$, under the conditions *m*=1 and $M, K \to \infty$, can be calculated by

the formula $N_3'(0) = L_0 K^{n+1} M^2 S_0 \left[(1+n)q - 3 \right]$. We have to impose a special condition for the parameter n to meet the inequality $N_3'(0) < 0$. This condition is set by the formula $n < \frac{3}{q} - 1$. In the majority of systems, $q \le 1$. In this case, for the parameter n the condition is set in the form $n < 2 + \varepsilon$. On the other hand, the derivative $N_3'(Z)$ in the point $Z=M$, under the condition $M, K \to \infty$ has the form:

$$N_3'(M) = L_0 (K+M)^{n+1} \left[C_0 (1+n) + \frac{(1+K+2M+nM)(1+P_0)}{K+M} - S_0 P_0 \right].$$

Provided that $n=2$, the inequality $N_3'(M) > 0$ provides the condition $3C_0 + \frac{1+4q}{1+q}(1+P_0) - S_0 P_0 > 0$. When $q \to 1$ and $C_0 \approx P_0$, this formula turns into $P_0(5.5 - S_0) + 2.5 > 0$. If $n=1$, then the condition $P_0(4 - S_0) + 2 > 0$ is valid. The condition $N_3'(M) > 0$ can be fulfilled if the number of iterations S_0 equals 4 or 5. Consequently, in this case the optimal point Z_{opt} lies within [0, M] interval.

3. Problem formulation by control theory approach

The most general approach to the problem of construction of the optimal design algorithm can be worked out based on the optimal control theory. We can define the design strategy with the aid of equations (4) and (6) with the variable parameter Z during the whole process of optimization. It means that we may change the number of independent variables and the number of terms in the penalty function formula at every point of optimization procedure. Also, it is worth introducing into our consideration a vector of control functions $U = (u_1, u_2, ..., u_m)$, where $u_j \in \Omega$ and $\Omega = \{0;1\}$. In other words, every control function u_j can take the value 0 or 1. These functions have the meaning of control functions of the design process and generalize this process. Particularly, the meaning of the control function u_j is as follows: the equation with the ordinal number j belongs to system (6), while the term $g_j^2(X)$ is eliminated from the right-hand part of formula (3) if $u_j = 0$, and vice versa - the j-th equation is excluded from system (6) and the respective term appears in the right-hand part of formula (3) if $u_j = 1$. Then the system model equations and the type of the penalty function can be rewritten in the form:

$$(1 - u_j) g_j(X) = 0, \quad j = 1,2,\ldots, M \tag{14}$$

$$\varphi(X) = \frac{1}{\varepsilon} \sum_{j=1}^{M} u_j \cdot g_j^2(X) \tag{15}$$

All the control functions u_j are functions of a current point of the design process. In this case the directed motion vector $H = f(X,U)$ is the function of the vectors *X* and *U*. The number of various design strategies generated within a single optimization procedure is virtually unlimited. Among all these strategies, there are one or several strategies, which are optimal and accomplish all the goals of design in a minimum possible time. Hence, the problem of search for the optimal strategy is now formulated as a typical problem of minimization of some functional in the optimal control theory. The functional value represents the actual running time of the processor. The main difficulty of such definition consists in unknown optimal dependencies of the control functions u_j. However, if we have an optimal vector of control functions, the optimal design strategy will be realized with the aid of this vector.

The idea of formulation of optimal design of a system from the viewpoint of time as a problem of minimization of a functional invoked from the optimal control theory, does not depend on some specific realization of the optimization algorithm, and can be embedded into an arbitrary optimization procedure. All this has been shown in (Zemliak, 2001), with approbation of three different algorithms, which are typical representatives of three major groups of optimization methods: the gradient method, the Newton's method and the Davidon-Fletcher-Powell method (DFP).

Now the process of network optimization is formulated as a controllable dynamic process. We have possibility to control the design process by means of the control vector *U* variation. In the above formulation, every possible design strategy, defined by the vector *U*, has its own trajectory in the space of variables. Obviously, the comparison of different trajectories by the time of moving over them, or by some other parameters, is consistent only in situation when these trajectories have identical initial and final points. On the other hand, the objective function $C(X)$ has a number of local minima, since design problems are nonlinear in principle, even if the design concerns a physically linear system. In this case, for consistent comparison of different strategies and their trajectories, it is desirable to impose additional conditions of single-valuedness for attainment of one and the same point in the space parameters. At the same time, the problem of ambiguity is not a peculiar feature of the new formulation of the design methodology. We face this problem every time when starting the design process from different initial points. In future, both in theoretical reasoning and in practical examples, we shall assume that the problem of single-valuedness of the final point is overcome by imposing some additional conditions on the variables. It should be specially stressed that this problem is essential only in comparison of different strategies and their trajectories, while in actual design we do not need any additional conditions except for feasibility requirements.

The process of system design, formulated in the terms of the control theory, can be described either in discrete or in continuous form. The continuous form is traditional for the optimal control theory. To represent the problem in the continuous form, we assume that numerical equation (2) corresponding to the optimization process can be replaced by a differential equation:

$$\frac{dX}{dt} = f(X,U) \tag{16}$$

where the right-hand part $f(X,U)$ is the vector of directed motion of H and depends on the generalized objective function $F(X,U)$.

What this means is that the design process is formulated as the problem of integration of system (16) with additional conditions (14). The structure of the function $f(X,U)$ can be defined as follows:

$$f(F(X,U)) = -F'(X,U) \tag{17}$$

- for the gradient method,

$$f(F(X,U)) = -\{F''(X,U)\}^{-1} \cdot F'(X,U) \tag{17′}$$

- for the Newton method, where $F''(X,U)$ is a matrix of second derivatives,

$$f(F(X,U)) = -B(X,U) \cdot F'(X,U) \tag{17″}$$

- for the DFP method, where $B(X,U)$ is a symmetric, positive definite matrix of the DFP algorithm.

In this case the problem of construction of the optimal, in terms of running time, design algorithm is formulated as a typical problem of functional minimization of the control theory for differential equation system (16). The right-hand part of (16) depends on the particular method of optimization, for instance, (17), (17′), or (17″), and has an objective function defined by formulas (5), (15) and constraints (14). An additional difficulty is that the right-hand parts of system (16) are piecewise continuous functions rather than strictly continuous. Such a problem for system (16) with piecewise continuous control functions can be resolved most effectively based on the known maximum principle (Pontryagin et al., 1962), but straightforward application of this principle for nonlinear problems of large dimensionality is highly problematical. This problem can be resolved based on provisions developed in the process of approximate solution of control theory problems (Fedorenko, 1969; Pytlak, 1999).
The fundamental system of equations of design process for the three algorithms of optimization is given below. System (16), which can be rewritten in component-wise form as

$$\frac{dx_i}{dt} = f_i(X,U), \quad i = 1,2,\ldots,N \tag{18}$$

in combination with (14) defines the design process.

In the case of gradient method, the right-hand part of (18), i.e.,

$$f_i(X,U) = -\frac{\delta}{\delta x_i} F(X,U), \qquad i = 1,2,\ldots,K \tag{19}$$

$$f_i(X,U) = -u_{i-K}\frac{\delta}{\delta x_i}F(X,U) + \frac{(1-u_{i-K})}{t_s}\left\{-x_i^s + \eta_i(X)\right\}, \quad i = K+1,K+2,\ldots,N \tag{19´}$$

where $F(X,U) = C(X) + \frac{1}{\varepsilon}\sum_{j=1}^{M} u_j g_j^2(X)$. The function $\eta_i(X)$, written in an implicit form, defines the current value of the variable x_i^{s+1} $\left(x_i^{s+1} = \eta_i(X)\right)$ obtained after resolving the system (14), and the control variables u_j are the functions of "current time".

In the case of Newton's method or DFP, equations (19) and (19´) undergo some modifications:

$$f_i(X,U) = -\sum_{k=1}^{N} b_{ik}\frac{\delta}{\delta x_k}F(X,U), \quad i = 1,2,\ldots,K$$

$$f_i(X,U) = -u_{i-K}\sum_{k=1}^{N} b_{ik}\frac{\delta}{\delta x_k}F(X,U) + \frac{(1-u_{i-K})}{t_s}\left\{-x_i^s + \eta_i(X)\right\}, \quad i = K+1,K+2,\ldots,N$$

where b_{ik} is an element of the inverse Hessian $\left\{F''(X,U)\right\}^{-1}$ for Newton's method, or an element of the matrix *B(X,U)* in DFP method.

In the latter case the matrix *B(X,U)* is defined by expressions $B_{s+1} = B_s + \frac{R^s (R^s)^T}{(R^s)^T Q^s} - \frac{(B_s Q^s)(B_s Q^s)^T}{(Q^s)^T B_s Q^s}$, where B_0 is the unitary matrix, $s = 0,1,\ldots$, while $R^s = X^{s+1} - X^s$ and $Q^s = F'(X^{s+1},U^s) - F'(X^s,U^s)$.

4. Numerical results

Numerical results in conformity with the new approach to formulation of the design process are presented below. They point to the prospects arising in construction of the optimal (in terms of minimum running time) algorithm. The primary emphasis is placed on demonstration of new opportunities appearing due to application of the new methodology. The number of nodes in the networks taken for illustrations in this part varies from 3 to 5. We deal with the problem of dc analysis, where the objective function $C(X)$ is defined as the sum of squared differences between the preset and current values of nodal voltages for some

nodes, supplemented with additional inequalities for some elements of the network. The calculations presented below correspond to the different optimization methods: the gradient method, the Newton's method, and the Davidon-Fletcher-Powell method (DFP).
The basic system of equations (16) was integrated by the fourth-order Runge-Kutta method. The integration step was variable and optimal for every new strategy to minimize the processor running time. The processor operation time indicated in the calculations corresponds to a computer with the processor Pentium 4, 2.2 GHz.
Fig. 1 shows the equivalent circuit of the network to be designed. The circuit has four independent variables (*K*=4), conductances y_1, y_2, y_3, y_4, three dependent variables (*M*=3), nodal voltages V_1, V_2, V_3, and two nonlinear elements.

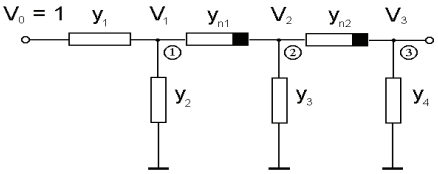


Fig. 1. Topology of three-node network

The nonlinear elements are defined as follows: $y_{n1} = a_{n1} + b_{n1}(V_1 - V_2)^2$, $y_{n2} = a_{n2} + b_{n2}(V_2 - V_3)^2$. The nonlinearity parameters are $b_{n1} = b_{n2} = 1$. The components of the vector *X* are defined by formulas $x_1^2 = y_1$, $x_2^2 = y_2$, $x_3^2 = y_3$, $x_4^2 = y_4$, $x_5 = V_1$, $x_6 = V_2$, $x_7 = V_3$. Defining the components x_1, x_2, x_3, x_4 by the above formulas automatically results in positive magnitudes of the conductances, which eliminates the issue of positive definiteness for each resistance and conductance and make it possible to carry out the optimization in the whole space of magnitudes of these variables without any limitations.

In this case we have a system of seven equations playing the role of the optimization algorithm, while the network model can be expressed by three nonlinear equations:

$$\frac{dx_i}{dt} = -\frac{\delta}{\delta x_i} F(X,U), \qquad i = 1,2,3,4 \tag{20}$$

$$\frac{dx_i}{dt} = -u_{i-4} \cdot \frac{\delta}{\delta x_i} F(X,U) + \frac{(1-u_{i-4})}{dt}\{-x_i(t-dt) + \eta_i(X)\}, \; i = 5,6,7$$

where $F(X,U) = C(X) + \sum_{j=1}^{3} u_j g_j^2(X)$,

$$
\begin{aligned}
&g_1(X) \equiv \left(x_1^2 + x_2^2 + a_{n1} + b_{n1}x_6^2\right)x_5 - \left(a_{n1} + b_{n1}x_6^2\right)x_6 - x_1^2 = 0 \\
&g_2(X) \equiv -\left(a_{n1} + b_{n1}x_6^2\right)x_5 + \left(x_3^2 + a_{n1} + b_{n1}x_6^2 + a_{n2} + b_{n2}x_7^2\right)x_6 - \left(a_{n2} + b_{n2}x_7^2\right)x_7 = 0 \\
&g_3(X) \equiv -\left(a_{n2} + b_{n2}x_7^2\right)x_6 + \left(x_4^2 + a_{n2} + b_{n2}x_7^2\right)x_7 = 0
\end{aligned}
\qquad (21)
$$

The system (21) can be transformed into the following one:

$$\left(1 - u_j\right) g_j\left(x_1, x_2, x_3, x_4, x_5, x_6, x_7\right) = 0 \qquad j = 1,2,3.$$

The results of analysis of the full structural basis of design strategies, arising in this case at a fixed value of the controlling vector *U*, are given in Table 1.

N	Control functions vector U (u1, u2, u3)	Gradient method		Newton method		DFP method	
		Iterations number	Total design time (sec)	Iterations number	Total design time (sec)	Iterations number	Total design time (sec)
1	(0 0 0)	59	0.229	8	0.1331	12	0.0853
2	(0 0 1)	167	0.273	6	0.0637	17	0.0552
3	(0 1 0)	174	0.291	11	0.1193	15	0.0503
4	(0 1 1)	185	0.154	11	0.0758	22	0.0421
5	(1 0 0)	63	0.122	8	0.0986	22	0.0839
6	(1 0 1)	198	0.245	9	0.0905	19	0.0499
7	(1 1 0)	228	0.258	15	0.1382	18	0.0442
8	(1 1 1)	293	0.176	17	0.0853	33	0.0504

Table 1. Complete structural basis of design strategies for three- nodes passive network

Here we have 8 different strategies. The strategy corresponding to the control vector (0, 0, 0) is a traditional design strategy (TDS); the strategy with the control vector (1, 1, 1) is a modified traditional design strategy (MTDS) while the rest are some new intermediate strategies. Strategy 5 is optimal for the gradient method, strategy 2 is optimal for Newton's method and strategy 4 is optimal for the DFP method. Nevertheless, these strategies are not optimal in the whole. The optimal strategies for all methods have been found due to application of a special procedure – by variation of the control vector. The respective results are presented in Table 2.

N	Method	Optimal control functions vector U (u1, u2, u3)	Iterations number	Switching points	Total design time (sec)
1	Gradient method	(101); (000); (111)	81	3; 7	0.0636
2	Newton method	(111); (000); (011)	6	1; 2	0.0492
3	DFP method	(101); (011)	15	2	0.0301

Table 2. Characteristics of optimal design strategy for three optimization methods

The optimal strategy has two points of "switching" for gradient and Newton's methods and one "switching" point for DFP optimization methods. The gain in time for the optimal

strategy, as compared with the traditional one, is 3.6 times for the gradient method, 2.7 for Newton's method and 2.83 times for the DFP method.

In the second case we analyze a network (Fig. 2) with five independent variables (K=5), conductances y_1, y_2, y_3, y_4, y_5, four dependent variables (M=4), nodal voltages V_1, V_2, V_3, V_4, and two nonlinear elements.

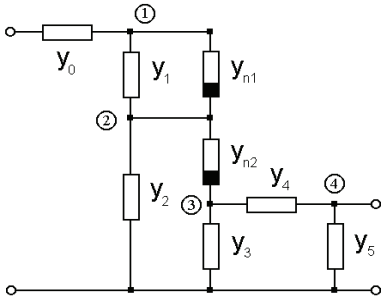


Fig. 2. Topology of four-node network

The nonlinear elements are defined by formulas $y_{n1} = a_{n1} + b_{n1} \cdot (V_1 - V_2)^2$, $y_{n2} = a_{n2} + b_{n2} \cdot (V_2 - V_3)^2$. System (16) includes nine equations, and system's model includes four equations. The results of analysis of the full set of strategies are presented in Table 3.

N	Control functions vector U (u1,u2,u3,u4)	Gradient method		Newton method		DFP method	
		Iterations number	Total design time (sec)	Iterations number	Total design time (sec)	Iterations number	Total design time (sec)
1	(0000)	114	0.819	10	0.366	15	0.187
2	(0001)	87	0.388	8	0.251	26	0.207
3	(0010)	51	0.269	10	0.361	11	0.103
4	(0011)	77	0.176	8	0.171	29	0.129
5	(0100)	100	0.526	7	0.251	12	0.112
6	(0101)	217	0.486	9	0.189	14	0.061
7	(0110)	166	0.338	14	0.273	19	0.076
8	(0111)	402	0.596	11	0.182	34	0.104
9	(1000)	111	0.586	5	0.181	11	0.103
10	(1001)	73	0.166	7	0.149	14	0.062
11	(1010)	115	0.256	8	0.167	16	0.104
12	(1011)	135	0.225	10	0.184	20	0.068
13	(1100)	519	1.055	13	0.253	14	0.056
14	(1101)	595	0.882	10	0.166	18	0.055
15	(1110)	159	0.169	15	0.185	20	0.046
16	(1111)	330	0.276	24	0.234	51	0.101

Table 3. Complete set of design strategies for four- nodes passive network

There are 16 different design strategies for this case. Of interest is the fact that among all the diversity of the design strategies, for the gradient method we have thirteen strategies, for Newton's method – 15, and for DFP – 14 strategies, whose processor time is less than that of traditional strategy 1 with the vector of controlling functions such as (0, 0, 0, 0). Among the whole stock of the strategies, the optimal one is strategy 10 for the gradient method, which provides a gain in time by a factor of 4.93 – compared to the traditional strategy, strategy 10 for Newton's method with a gain by 2.46, and strategy 14 – for the DFP method, with a gain by 4.06 times.
The optimal or, to be more exact, quasi-optimal strategies have been found by using a special procedure, whose application results are given in Table 4.

N	Method	Optimal control functions vector U (u1, u2, u3, u4)	Iterations number	Switching points	Total design time (sec)
1	Gradient method	(1011); (0000); (1110)	53	3; 4	0.0644
2	Newton method	(1010); (1001)	6	3	0.1265
3	DFP method	(0111); (1110)	17	2	0.0405

Table 4. Characteristics of quasi-optimal design strategy for three optimization methods

Here we have two switching point for the gradient method and one – for Newton's method and DFP method. These quasi-optimal strategies provide a gain in time by a factor of 12.7 for the gradient method, 2.89 for Newton's method, and 4.6 for DFP method.
In the next case we analyzed an electronic network of an amplifier in two transistors (Fig. 3). The design process was performed for dc conditions. The transistors are represented by the steady-state Ebers-Moll model embedded in the SPICE system (Massobrio & Antognetti, 1993). The objective function was defined as the sum of squared differences between present and current values of voltages across transistor's junctions.

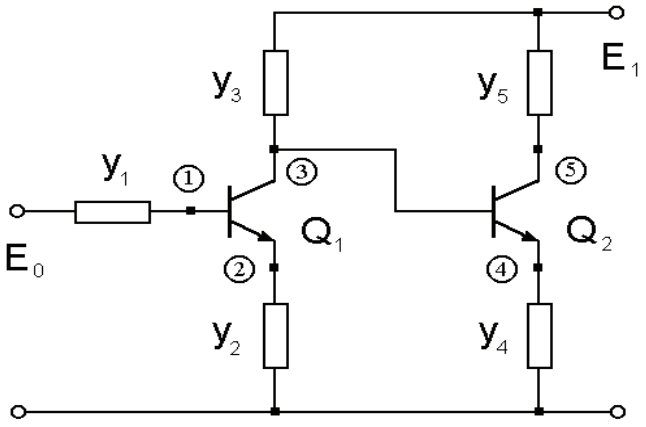


Fig. 3. Circuit topology for two-stage transistor amplifier

For this example we establish five independent variables y_1, y_2, y_3, y_4, y_5 (K=5), and five dependent variables V_1, V_2, V_3, V_4, V_5 (M=5). In this case the optimization algorithm is based on a system of ten equations, and the network model is defined by five nonlinear equations. The full set of design strategies includes 32 different strategies. Table 5 contains the results of analysis of the TDS, and some other strategies among those requiring less time for implementation than the traditional one. Strategy 17 is optimal for gradient method and strategy 13 is optimal for DFP method. It makes possible to obtain a gain in time, as compared to the traditional one, by 19.7 times for the gradient method, and by 26.2 times for the DFP method. However, as in the previous examples, these strategies are not "optimal-in-whole". The results of quasi-optimal strategies are listed in Table 6.

N	Control functions vector U (u1,u2,u3,u4,u5)	Gradient method		DFP method	
		Iterations number	Total design time (sec)	Iterations number	Total design time (sec)
1	(0 0 0 0 0)	979	20.466	274	7.361
2	(0 0 1 0 0)	889	15.396	196	6.381
3	(0 0 1 0 1)	704	9.571	57	1.474
4	(0 0 1 1 0)	989	13.455	251	5.818
5	(0 0 1 1 1)	512	5.405	154	2.986
6	(0 1 1 0 0)	859	11.861	218	5.631
7	(0 1 1 0 1)	420	4.503	125	2.522
8	(0 1 1 1 0)	751	8.011	129	2.591
9	(0 1 1 1 1)	528	4.228	90	1.371
10	(1 0 1 0 0)	780	10.745	199	2.956
11	(1 0 1 0 1)	249	1.734	62	0.462
12	(1 0 1 1 0)	1253	13.297	135	1.545
13	(1 0 1 1 1)	386	2.161	30	0.281
14	(1 1 1 0 0)	1683	17.702	105	1.187
15	(1 1 1 0 1)	263	1.471	56	1.769
16	(1 1 1 1 0)	1191	9.459	77	0.656
17	(1 1 1 1 1)	637	1.039	65	0.296

Table 5. Some design strategies for two-stage transistor amplifier

N	Method	Optimal control functions vector U (u1, u2, u3, u4, u5)	Iterations number	Switching points	Total design time (sec)	Computer time gain
1	Gradient method	(11101); (11111)	220	11	0.403	50.8
2	DFP method	(10111); (11111)	37	11	0.157	46.9

Table 6. Characteristics of quasi-optimal design strategy for two-stage transistor amplifier

The optimal strategy in both cases has a single switching point. The gain in time, as compared to the traditional strategy, in this case is 50.8 times for the gradient method and 46.9 times for DFP method.
The results of analyzed passive circuits for M from 1 to 5 are presented in Fig. 4 for three different optimization procedures.

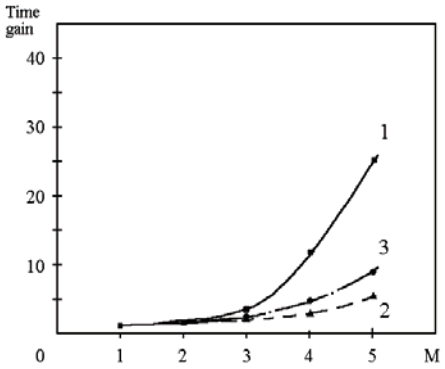


Fig. 4. Computer time gain for passive networks

This is the computer time gain of the optimum design strategy with respect to the traditional design strategy as the function of the dependent parameters' number M. The traditional design approach is not time-optimal and the time gain increases very fast with the M increasing.
The results of the active networks optimization for different number of transistor cells N_{TR} are shown in Fig. 5.

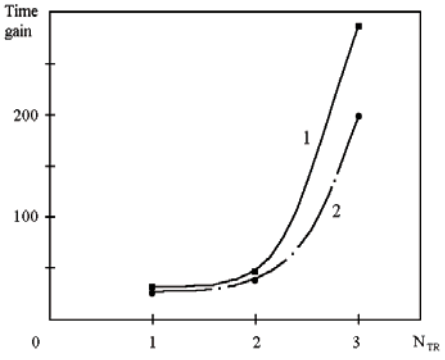


Fig. 5. Computer time gain for active networks for different number of transistor cells

This result confirms the rule that the total computer time gain of the time-optimal design strategy increases when the complexity of the network increases. The obtained results permit us to make the following current conclusions:

1) In practice, the traditional design strategy is never optimal in terms of time.

2) The new design strategies form a basis, which can be used for generation of a design strategy optimal by the running time criterion.

3) The potential gain in time, attained with the aid of optimal strategy, grows as the size and complexity of the electronic network increase.

5. Acceleration effect

We can obtain the possible gain in the computation time for quasi-optimal strategy. However, this expected gain can become reality only if we manage to generate the algorithm making it possible to determine the optimal trajectory of the design process. Thus the problem of investigation of major intrinsic properties and restrictions of the optimal trajectory of design is the principal task to be resolved in creation of the optimal algorithm.
Based on the concepts suggested, it is possible to analyze new effects (Zemliak, 2002) arising in the process of network design by the control theory methods. The inquiries started from a simplest nonlinear circuit with a single node and two parameters (*N*=2), which has no practical applications but serves a good illustration for comprehension of the process in the network design based on the new methodology. After that an *N*-dimensional problem will be considered. All the examples included demonstrate some phenomenon, which may be called the effect of acceleration of design process. The latter arises because of different behavior of design trajectories having different control functions. The circuit diagram of the nonlinear electronic network with a single node is shown in Fig. 1.

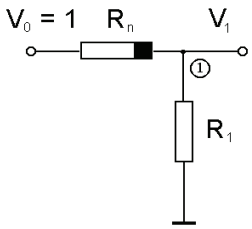
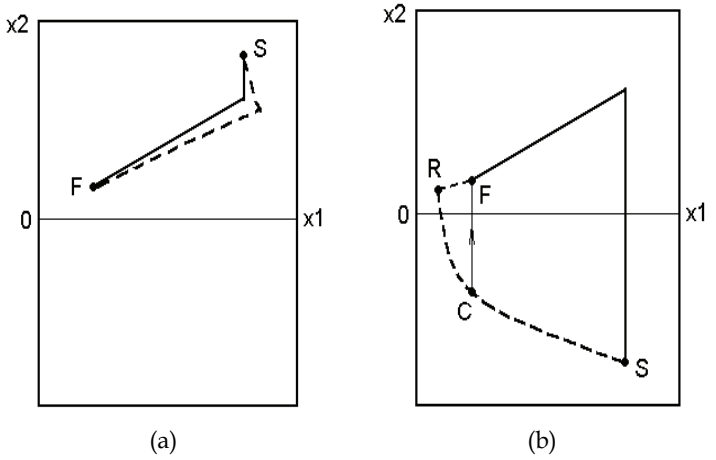


Fig. 6. Simplest nonlinear electronic network

Assume that the element R_n is described by a nonlinear function having the form $R_n = r_{10} + b_n \cdot V_1^2$. For this example we defined only two parameters: resistance R_1 as independent parameter (*K*=1), and nodal voltage V_1 as dependent parameter (*M*=1). In this example, as well as in the subsequent ones, we also assume that all the resistances are positive-valued. For automatically meeting the latter requirement the following definition of the vector *X* can be used: $X = (x_1, x_2)$ where $x_1^2 \equiv R_1$ and $x_2 \equiv V_1$.
The structural basis of various design strategies, defined for the control vector *U* in our case consists of two strategies – at u_1=0 and u_1=1. The first one is the TDS while the second is the

MTDS. The design trajectories for the initial point $X^0 = (1,1)$, are depicted in Fig. 7a: the solid line – for TDS, and the dashed one – for MTDS. The number of iterations and the processor time in the first case are equal to 44 and 0.092_{10}-3 s and in the second case – 78 and 0.149_{10}-3 s. As can be seen, the traditional strategy is preferable. An insignificant reduction of the designing time (5%) can be obtained if in the process of motion the control function u_1 changes form 0 to 1 at step 18. A different result is observed in the event of selecting a negative value of initial point for the variable x_2, for instance, -1, i.e., $X^0 = (1,-1)$. The trajectories corresponding to this situation are depicted in Fig. 7b.



(a) (b)

Fig. 7. Trajectories for TDS (solid) and for MTDS (dash) for: (a) X_{in} =(1,1), (b) X_{in} =(1,-1)

The trajectory corresponding to TDS remains almost unchanged. In the first case we have a jump downward from the initial point to the line corresponding to the matched solution. In the second case we have the jump to the same line, but upward. Since the jump occurs instantaneously, the time in both cases is the same. A somewhat different situation is observed for MTDS trajectory. At the negative initial value of the variable x_2 (x_2 = -1) the first part of the trajectory lies in the unfeasible (negative in terms of the variable x_2) half-space, while the second part – in the positive one. It is pertinent to note that the motion of the current point over the first part of the trajectory from the point *S* to point *R* occurs rapidly enough, and then slows down. The total time in this event is larger than at the positive initial approximation. We emphasize that the trajectories of both different strategies approach the final point *F* of the design process from opposite sides. There opens a possibility for accelerating the process by changing the control function u_1 from 1 to 0 in the *C*, which represents the projection of the final point *F* on the trajectory corresponding to MTDS. In this case the optimal strategy has two parts. The first one, described by the curve *SC*, corresponds to u_1=1, and MTDS lies in the physically unreal space. In the point *C* the

value of the control function u_1 takes the zero value, and we make a jump to the final point *F* or close to it, which depends on the step of integration and on prescribed accuracy. The second part of the trajectory, starting in the point *C* and corresponding to u_1=0, and TDS either degenerates into a jump, so that we have a single step in addition to the first part of the trajectory, or into several (few in number) additional steps corresponding to TDS. The number of iterations for our example, corresponding to u_1=1, is equal to 9, plus a single step (a jump) corresponding to u_1=0. The time for this optimal trajectory is 0.0194_{10}-3 s, which means acceleration of the process by a factor of 4.7. The effect of acceleration can be also observed for more complex examples. However, in this case the trajectories lie in the *N*-dimensional space, and we must analyze various projections of *N*-dimensional curves. Some different passive and active nonlinear networks were investigated to observe an additional acceleration effect. The potential computer time gain of the optimum design strategy with an additional acceleration as the function of the transistor cell number N_{TR} is presented in Fig. 8 for two different optimization procedures (gradient and DFP methods).

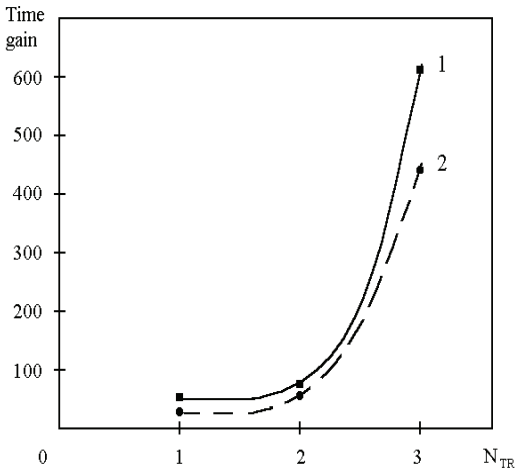


Fig. 8. Computer time gain for active networks with acceleration effect for different number of transistor cells. 1-Gradient method, 2-DFP method

Fig. 8 shows that the time gain for three-stage transistor amplifier increase till 500-600 due to the acceleration effect.

The analysis of acceleration effect shows that this effect appears with special initial conditions. For instance for the network in Fig. 6 this effect exists when the initial point has a negative value of the coordinate x_2, and the value of this coordinate itself, corresponding to the position under the special line - separatrix, are the sufficient conditions for obtaining the acceleration effect. A more detailed analysis shows that in reality these conditions are not necessary. Fig. 9 shows the phase portrait of design process, corresponding to MTDS for the network in Fig. 6, but for all possible values of the coordinate x_2.

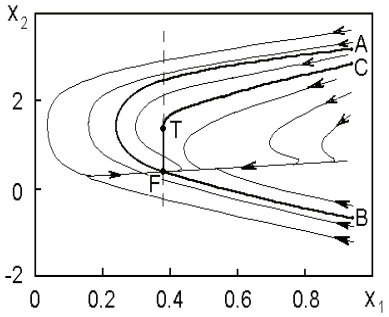
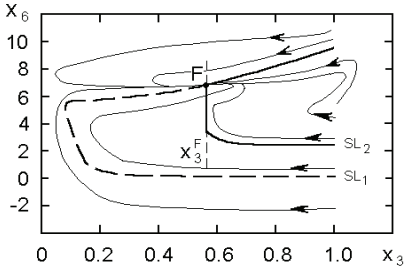


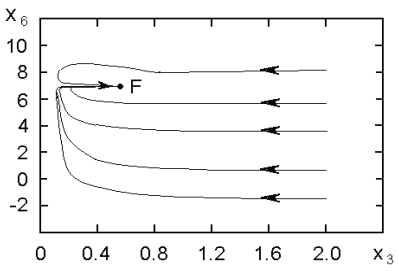
Fig. 9. Design trajectories for MTDS and separatrix for one-node network

The portrait includes two types of separatrix. The first one is separatrix *AFB* separating the trajectories approaching the final point from the left and from the right. The second type separatrix *CTFB* divides the phase space in two other subspaces. The points of the subspace surrounded by this separatrix do not give trajectories permitting to realize the acceleration effect. Conversely, the points lying outside the separatrix correspond to trajectories able to produce the acceleration effect. The jump to the final point can be performed from below as well as from above the separatrix *CTFB* along the extension of the line *TF*. These geometric conditions are necessary and sufficient for existence of the acceleration effect.

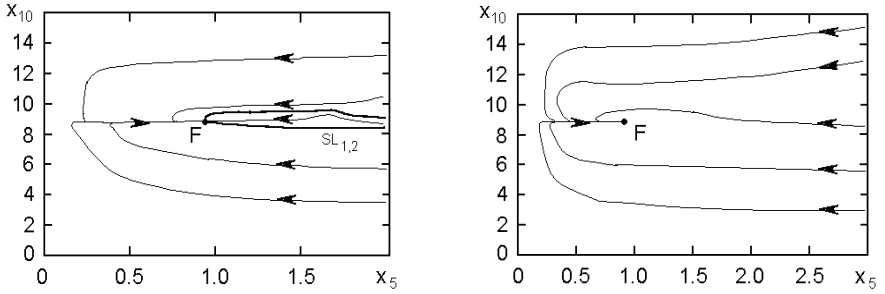
The examples given below refer to design of a transistorized amplifier in one and two transistors presented in Fig. 3. The design of the single- and two-stage amplifiers has been performed separately. In the case of the single-stage amplifier, there are three independent variables y_1, y_2, y_3 (*K*=3), and three dependent ones V_1, V_2, V_3 (*M*=3). The vector *X* includes six components: $x_1^2 = y_1,\ x_2^2 = y_2,\ x_3^2 = y_3,\ x_4 = V_1,\ x_5 = V_2,\ x_6 = V_3$. For the two-stage amplifier, the vector *X* contains 10 components. Fig. 10 shows two-dimensional projections of phase portraits of the design process for MTDS in both cases: *a, b* – for the single stage, and *c, d* – for the two-stage amplifier. Fig. 10a,b illustrate behavior of the trajectories on the plane $x_3 - x_6$ at initial values x_i^0 =1,0, i =1,2,3 (case *a*) and at initial values x_i^0 =2,0 (case *b*). We can see a pronounced difference in behavior of phase trajectories for passive and active networks.



(a)



(b)



(c) (d)

Fig. 10. Family of curves that correspond to MTDS and separatrix for: (a), (b) one-stage; (c), (d) two-stage amplifier

Projections of the separatrix SL_1 and separatrix SL_2 are clearly expressed in the case of x_i^0 =1,0, which is indicative of the presence or absence of the trajectories offer the possibility of the "jump" into the final point of the design process. Of interest is the fact that an increase in network complexity results in expansion of the domain of existence of acceleration effect, which can be seen in Fig 10*c* for the two-stage amplifier. Here we analyze the behavior of projections of trajectories on the plane $x_5 - x_{10}$ at x_i^0 =2,0, i =1,2,…,5. The zone confined by the separatrix, where the acceleration effect is absent, becomes narrower for the two-stage amplifier. An increase in initial values of originally independent variables x_i^0 up to 3,0, i =1,2,…,5 for the two-stage amplifier (Fig. 10*d*) results in disappearance of separatrix projections – as in the case of the single-stage network. Based on analysis of the above examples we come to the conclusion that complication of electronic network structure and an increase in initial values of originally independent variables expands the domain of existence of the acceleration effect of design process.

The optimal choice of the initial point of the design process permits to realize the acceleration effect with a larger probability. Analysis of trajectories for different design strategies shows that the separatrix concept is useful for comprehension and determination of necessary and sufficient conditions of existence of the design acceleration effect. The separatrix divides the whole phase space of design into a domain where we can achieve the acceleration effect, and a domain in which this effect does not exist. The first domain may be used for constructing the optimal design trajectory. Selection of the initial point of design process outside the domain encircled by separatrix constitutes the necessary and sufficient conditions for existence of the acceleration effect. In the general case, a separatrix is a hyper surface having an intricate structure. However, the real situation is simplified in the most important case, corresponding to active nonlinear networks, because of narrowing the area inside the separatrix, or its complete disappearance – at the initial values of the originally independent variables large enough. It means that the acceleration effect can be realized almost in any case for the networks of large complexity.

6. Stability analysis

Basic concepts of a new methodology in analogue networks optimization in terms of the control theory were stated in previous sections. It was shown that the new approach potentially allows to significantly decreasing the processor time used to design the circuit. This quality appears due to a new possibility of controlling the design process by redistributing computational burden between the circuit's analysis and the procedure of parametric optimization. It may be considered to be a proven fact that traditional design strategy (TDS) including the circuit's analysis at every step of its design is not optimal with respect to time. More over the benefit in time used to design the circuit for some optimal or more precisely quasi-optimal strategy compared to TDS increases with increasing size and complexity of the designed circuit. This optimal strategy and corresponding design's trajectory were obtained using special search procedure and serve only as a proof existing strategies which are much more optimal than TDS. However, it is clear that the problem lies in the ability to move along an optimal trajectory of the circuit's design process from the very beginning of designing the circuit. Only in this case it is possible to obtain the mentioned potentially tremendous advantage in time, which corresponds to the optimal design strategy. During the building the optimal strategy and its corresponding trajectory at the present moment it is necessary to analyze their most significant characteristics. The study of the optimal trajectory's qualitative characteristics and their differences from those of the other trajectories appears to be the only possible way to solve the problem.

The discovery of an effect expecting additional acceleration of the design process and exploration of conditions determining this effect's existence lead to increased time advantage and serve as an initial point of quasi-optimal design strategy building. The analysis of this effect allowed to state three most significant moments: 1) to obtain the acceleration effect the initial point of the design process should be chosen outside the domain limited with a special hypersurface (separatrix), 2) the acceleration effect appears during a transition from a trajectory corresponding to a modified traditional design strategy (MTDS) to the trajectory which corresponds to TDS and from any trajectory similar to MTDS to any trajectory similar to the trajectory of TDS, 3) the most significant element of the acceleration effect is an exact position of the switch point corresponding to a transition from one strategy to another.

To obtain an optimal sequence of switching points during the design process it is necessary to select a special criterion, which depends on the internal properties of the design strategy. The problem of searching for the optimal with respect to time design strategy deals with a more general problem of convergence and stability of each trajectory. On the basis of experiment, the design time for each strategy determines by properties of convergence and stability of corresponding trajectory. One of the common approaches of analysis of dynamic systems stability is based on the direct Lyapunov method (Barbashin, 1967; Rouche et al., 1977). We consider that the time design algorithm is a dynamic controlled process. In this case, the main control aim is determined as minimization problem of transient time of this process. As result, the analysis of stability and characteristics of transient process (process of designing is one of these) for each trajectory are possible on the basis of the direct Lyapunov method. Let's introduce Lyapunov function of process of designing. It will be used for analysis of properties and structure of optimal algorithm and for searching of optimal switch point positions of control vector particularly.

There is a certain freedom of Lyapunov function choice as the latter has more than one form. Let's denote the Lyapunov function of process of designing (1)–(5) in form:

$$V\left(X\right) = \sum_i \left(x_i - a_i\right)^2 \tag{22}$$

where a_i is a stationary value of coordinate x_i. The set of all coefficients a_i is the main result of process of designing as the minimum of target function $C(X)$ is achieved at these values of coefficients, i.e. the aim of designing is succeeded. It is clear, that these coefficients are accurately known only at the end of designing. The other variables $y_i = x_i - a_i$ could be determined instead of x_i variables. In this case equation (5) takes the form:

$$V(Y) = \sum_i y_i^2 \tag{23}$$

Taking into account the new variables y_i, the process of designing (1)–(5) remains the same form. However, equation (23) satisfies all conditions of Lyapunov function definition. Indeed, this function is piecewise continuous function having piecewise continuous first partial derivatives. In addition, three main properties of function (23) V(Y)>0, 2) V(0)=0, and 3) $V(Y) \to \infty$ for $\|Y\| \to \infty$) are presented. In this case we obtain the possibility to analyze the stability of equilibrium position (point *Y*=0) by Lyapunov theorem. On other hand, the stability of point $a = (a_1, a_2, ..., a_N)$ analyzes on basis of (22). It is clear, both of these problems are identical. The point $a = (a_1, a_2, ..., a_N)$ can be defined only at the end of the process of designing that is inconvenience of equation (22). As result, we could analyze the stability of various designing strategies by the equation (22) if the problem's solution (i.e. point *a*) was determined already in another way. Moreover, the possibility to control the stability of process during optimization procedure is of interest. In this case we have to determine another form of Lyapunov function which would be irrespective of final stationary point *a*. Let's define Lyapunov function in the form:

$$V(X,U) = \left[F(X,U)\right]^r \tag{24}$$

$$V\left(X,U\right) = \sum_i \left(\frac{\partial F\left(X,U\right)}{\partial x_i}\right)^2 \tag{25}$$

where $F(X,U)$ is a generalize target function of process of designing and $r > 0$. Under

additional conditions both of these equations determine Lyapunov function having properties similar to (23) in the sufficiently great neighborhood of stationary point. Meanwhile the dependence on control vector U appears too. Indeed, we can see that the value of (24) is equal zero in the stationary point if the target function of this process $C(X)$ in the same point is equal zero as well. The equation (24) is positive defined function in all points distinct from a stationary point as the function $C(X)$ is nonnegative. The function $V(X,U)$ increases without bound when we are going away from the stationary point.

The equation (25) also determines Lyapunov function if $\partial F / \partial x_i = 0$ in the stationary point $a = (a_1, a_2, ..., a_N)$ and $V(a,U)=0$. On the other hand, $V(X,U)>0$ for all X. In conclusion, Lyapunov function determined by (25) is a function of vector U i.e. all coordinates x_i depend on U. The third property of Lyapunov function is proven wrong as the behavior of function $V(X,U)$ when $\|X\| \to \infty$ is unknown. However, as known a posteriori, the function $V(X,U)$ is the increasing function in the sufficiently great neighborhood of a stationary point. According to Lyapunov method, the information about trajectory stability is connected with time derivative of Lyapunov function. Direct calculation of time derivative of Lyapunov function $\dot{V}$ lets estimate the dynamic system stability. The process of designing and a corresponding trajectory is stable if this derivative is negative. On the other part, the direct Lyapunov method gives sufficient but not necessary stability conditions. This implies that the process can lose stability or can remain stable in the case of positive derivative. The appearance of positive values of derivative $\dot{V}$ on set of positive measure only states the instability displaying in a few growth of Lyapunov function instead of decreasing latter. If such process exists far from the stationary point, then the process of designing is divergent function and we cannot obtain the solution of this trajectory. In this case the initial point of process of designing or strategy should be changed. If the positive derivative $\dot{V}$ appears at the end of process of designing (i.e. not far from the stationary point), then we could say that the process of designing significantly decelerates. This designing strategy is going round in a circle and cannot provide required accuracy. As result, the engineering time substantially grows. This effect is well known in practical designing. If we obtain the unacceptable accuracy, the strategy of designing or initial point have to be changed. The detailed behavioral analysis of Lyapunov function and its derivative for different strategies of designing makes it possible to choose the perspective strategies. This analysis also allows determining on qualitative level the relationship between design time and Lyapunov function and its derivative being the main factors of the process of designing.

Two-stage transistor amplifier, depicted in Fig.3, is used for stability analysis of different strategies of design. The direct calculation of time derivative $\dot{V}$ of Lyapunov function, determined by (29) for r=0.5, shown that the derivative is negative in the initial point of design for all trajectories, i.e., all possible design's strategies and its trajectories are stable at the beginning if integration step of system (1) is enough small. In the same time, when the current point of trajectory reaches some ε-neighbourhood of the stationary point

$(a_1, a_2, ..., a_N)$, the derivative of Lyapunov function comes positive and the current design's strategy loses stability. This implies that this strategy dose not ensure the convergence of trajectories to the stationary point $(a_1, a_2, ..., a_N)$ starting from some value of ε-neighbourhood, i.e. achievement of minimum of target function *F(X,U)* and so function *C(X)* with accuracy to ε dose not guarantee. In fact, each trajectory has eigen ε-neighbourhood determining maximum available accuracy for this one and the convergence problem arises inside this area. The process of designing significantly decelerates for current strategy before approaching of the critical value of ε-neighbourhood. Alias, the derivative $\dot{V}$ remains negative but has enough small absolute value. The results for two-stage amplifier are presented in Table 7. The design realized on the basis of strategies coming into structural basis 2^M and determined by control vector *U*. The appearance of positive values of derivative $\dot{V}$ on set of positive measure determines the termination of process of designing. Process optimization realized on the basis of equation (18) and gradient method with a variable optimal step t_s hereupon this step t_s could be both small and large. As result, we have non-smooth behavior of derivative from one step to other.

N	Control vector U(u1,u2,u3,u4,u5)	Iterations number	Computer time (sec)	Critical value of ε-neighborhood
1	(0 0 0 0 0)	3177	7.25	2.78E-08
2	(0 0 0 0 1)	3074	8.02	3.36E-07
3	(0 0 0 1 1)	11438	26.36	8.18E-07
4	(0 0 1 0 1)	799	1.16	9.38E-09
5	(0 0 1 1 0)	1798	2.61	1.61E-08
6	(0 1 0 1 1)	43431	76.89	3.16E-05
7	(0 1 1 0 0)	1378	2.25	1.67E-08
8	(0 1 1 0 1)	571	0.72	6.83E-09
9	(0 1 1 1 0)	1542	2.03	2.05E-08
10	(1 0 0 1 1)	11839	21.37	1.68E-05
11	(1 0 1 0 0)	2097	3.57	5.47E-07
12	(1 0 1 1 0)	6026	8.31	4.94E-07
13	(1 1 1 0 0)	6602	8.84	7.41E-07
14	(1 1 1 0 1)	935	0.71	1.33E-08
15	(1 1 1 1 0)	2340	2.31	1.62E-07
16	(1 1 1 1 1)	1502	0.38	1.09E-08

Table 7. Critical value of the ε-neighborhood for design strategies for two-stage amplifier

For smoothing of derivative $\dot{V}$ the value averaging on the interval 30 steps was used. The

analysis of results of Table 7 has shown some important laws. First of all, the strong correlation between processor time and critical value of ε -neighbourhood, after which the value of derivative $\dot{V}$ stays positive, is presented. As a rule, the fewer available value of ε -neighbourhood corresponds to the fewer processor time. We could order all strategies in Table 7. from the smallest processor time (0.38 sec, No. 16) to the longest one (76.89 sec, No. 6).

On the other hand, the strategies in ascending order of critical value of ε -neighbourhood are presented in Table 8.

Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of strategies regulated by the computer tim	16	14	8	4	9	7	15	5	11	1	2	12	13	10	3	6
Number of strategies regulated by the ε -neighborhood	8	4	16	14	5	15	7	9	1	2	12	11	13	3	10	6

Table 8. Strategies' ordering by processor time and by critical value of ε -neighborhood

The No. of each strategy in Table 8 determined by two different ways of order is slightly different. Two strategies (13 and 6) have the same number. Seven strategies have the difference in one place, four ones – in two places, and three strategies – in three places. The average difference is equal to 1.5. Taking into account that the critical values of ε -neighbourhood are obtained approximately by the averaging during integration of system (1) we can see that the correspondence of processor time with critical value of ε -neighbourhood is enough acceptable. Contrariwise, the parameters of ε -neighbourhood are obtained on the basis of Lyapunov function and its derivative analysis. Therefore, we could say that the strong correlation between processor time and properties of Lyapunov function is presented.

From the analysis above the assumption is induced: Lyapunov function of process of designing and its derivative are enough informative source to select more perspective design strategies.

7. Conclusion

The traditional approach for the analogue network optimization is not time-optimal. The problem of the minimal-time design algorithm construction can be solved adequately on the basis of the control theory. The network optimization process in this case is formulated as a controllable dynamic system. Analysis of the different examples gives the possibility to conclude that the potential computer time gain of the time-optimal design strategy increases when the size and complexity of the system increase. The Lyapunov function of the optimization process and its time derivative include the sufficient information to select more perspective strategies. The above-described approach gives the possibility to search the

time-optimal algorithm as the approximate solution of the typical problem of the optimal control theory.

8. References

Barbashin, E.A. (1967). *Introduction in Stability Theory*, Nauka, Moscow

Brayton, R.K.; Hachtel, G.D. & Sangiovanni-Vincentelli, A.L. (1981). A survey of optimization techniques for integrated-circuit design. *IEEE Proceedings*, Vol. 69, pp. 1334-1362

Fedorenko, R.P. (1969). *Approximate Solution of Optimal Control Problems*, Nauka, Moscow

Fletcher, R. (1980). *Practical Methods of Optimization*, John Wiley and Sons, New York, N.Y.

George, A. (1984). On block elimination for sparse linear systems. *SIAM J. Numer. Anal*., Vol. 11, No.3, pp. 585-603

Gill, P.E.; Murray, W. & Wright, M.H. (1981). *Practical Optimization*, Academic Press, London

Kashirskiy, I.S. (1976). General Optimization Methods. *Izvest. VUZ USSR -Radioelectronica*, Vol. 19, No. 6, pp. 21-25

Kashirskiy, I.S. & Trokhimenko, Y.K. (1979). *General Optimization for Electronic Circuits*, Tekhnika, Kiev

Massara, R.E. (1991). *Optimization Methods in Electronic Circuit Design*, Longman Scientific & Technical, Harlow

Massobrio, G. & Antognetti, P. (1993). *Semiconductor Device Modeling with SPICE*, Mc. Graw Hill, Inc., New York, N.Y.

Ochotta, E.S.; Rutenbar, R.A. & Carley, L.R. (1996). Synthesis of high-performance analog circuits in ASTRX/OBLX", *IEEE Trans. on CAD*, Vol. 15, No. 3, pp. 273-294

Osterby, O. & Zlatev, Z. (1983). *Direct Methods for Sparse Matrices*, Springer-Verlag, New York, N.Y.

Pontryagin, L.S.; Boltyanskii, V.G.; Gamkrelidze, R.V. & Mishchenko, E.F. (1962). *The Mathematical Theory of Optimal Processes*. Interscience Publishers, Inc., New York

Pytlak, R. (1999). *Numerical Methods for Optimal Control Problems with State Constraints*, Springer-Verlag, Berlin

Rabat, N; Ruehli, A.E.; Mahoney, G.W. & Coleman, J.J. (1985). A survey of macromodeling. *Proc. IEEE Int. Symp. Circuits Systems*, pp. 139-143, April 1985.

Rizzoli, V. Costanzo, A. & Cecchetti, C. (1990). Numerical optimization of broadband nonlinear microwave circuits. *IEEE MTT-S Int. Symp*., Vol. 1, pp. 335-338

Rouche, N.; Habets, P. & Laloy, M. (1977). *Stability Theory by Liapunov's Direct Method*, Springer-Verlag, New York, N.Y.

Ruehli(a), A.E.; Sangiovanni-Vincentelli, A. & Rabbat, G. (1982). Time analysis of large-scale circuits containing one-way macromodels. *IEEE Trans. Circuits Syst.*, Vol. CAS-29, No. 3, pp. 185-191

Ruehli(b), A.E. (1987). *Circuit Analysis, Simulation and Design*, Vol. 3, Elsevier Science Publishers, Amsterdam

Sangiovanni-Vincentelli, A.; Chen, L.K. & Chua, L.O. (1977). An efficient cluster algorithm for tearing large-scale networks. *IEEE Trans. Circuits Syst.*, Vol. CAS-24, No. 12, pp. 709-717

Wu, F.F. (1976). Solution of large-scale networks by tearing. *IEEE Trans. Circuits Syst*., Vol. CAS-23, No. 12, pp. 706-713

Zemliak(a), A.M. (2001). Analog system design problem formulation by optimum control theory. *IEICE Trans. on Fundam.*, Vol. E84-A, No. 8, pp. 2029-2041

Zemliak(b), A.M. (2002). Acceleration Effect of System Design Process. *IEICE Trans. on Fundam.*, Vol. E85-A, No. 7, pp. 1751-1759