Artificial Intelligence and Feedback Control

Verbruggen, Henk B.; Åström, Karl Johan

1989

Document Version:
Publisher's PDF, also known as Version of record

Link to publication

Citation for published version (APA):
Verbruggen, H. B., & Åström, K. J. (1989). *Artificial Intelligence and Feedback Control*. (Technical Reports TFRT-7433). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
2

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-7433)/1-12/(1989)

Artificial Intelligence and Feedback Control

H.B. Verbruggen
K.J. Åström

Department of Automatic Control
Lund Institute of Technology
October 1989

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
Report

Date of issue
October 1989

Document Number
CODEN: LUTFD2/(TFRT-7433)/1-12/(1989)

Author(s)
H.B. Verbruggen and K.J. Åström

Supervisor

Sponsoring organisation

Title and subtitle
Artificial Intelligence and Feedback Control.

Abstract

This paper describes some possible applications of Artificial Intelligence methods in control engineering. Due to the restriction of AI methods to well-defined knowledge domains (expert systems), the growing interest in fuzzy control methods and the possible application of simple neural networks, a large number of applications in all areas of technical science came into consideration. The recent availability of adequate software and lately also of special purpose hardware tools has accelerated this development.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
12

Security classification

Recipient's notes

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

Prepared for IFAC Workshop on Artificial Intelligence in Real Time Control, 19-21 September 1989, Shenyang, PRC.

Artificial Intelligence and Feedback Control

H.B. Verbruggen
Delft University of Technology
Faculty of Electrical Engineering, Control Laboratory
P.O. Box 5031, 2600 GA Delft, The Netherlands

K.J. Åström
Lund Institute of Technology, Department of Automatic Control
Lund, Sweden

Abstract

This paper describes some possible applications of Artificial Intelligence methods in control engineering. Due to the restriction of AI methods to well-defined knowledge domains (expert systems), the growing interest in fuzzy control methods and the possible application of simple neural networks, a large number of applications in all areas of technical science came into consideration. The recent availability of adequate software and lately also of special purpose hardware tools has accelerated this development.

1 Introduction

In process automation there is a tendency to use Plant-wide Control systems which combine in one system tactical, managerial, scheduling, operational and control tasks for production and process control. A multilayered information processing system is defined based on the various levels of decision making. An increasing number of people from different disciplines are involved in the operation of these systems, each with their own background and expertise and their different demands and requirements concerning the overall system behavior. The system should provide the various users with advisory and consultancy possibilities, based on the different kinds of specialized knowledge. It is clear that a lot of quantitative information is processed within and between the various levels of automation, but there is also a growing need for concentrated and qualitative information. In the expert systems used at these levels the main emphasis is on explanation facilities and knowledge translation, highlighting the key functions and parameters on each decision level. There is a need for a distributed expert and intelligent system, passing global information to the various levels of automation and inferencing on the specialized and more detailed information at the given level. This system should be coupled to the distributed computer control system or incorporated in the existing hard- and software. When we classifiy the application areas of expert sytems in automation systems we can distinguish systems which are based mainly on static or on dynamic information. Another division could be based on classification requirements or design and control problems. It is clear that because of the time critical situation the highest demands are put on the use of expert and other intelligent systems on the supervisory and control level. In these systems the expertise of process operators and control systems designers is mixed with the time-varying information obtained directly from the process by the measurements.

This paper is concerned with the lower levels of automation at which the knowledge-based and intelligent system should be implemented in a real-time environment. The most promising applications are found in the areas of:
- alarm monitoring, diagnosis and handling
- supervisory and adaptive control
- modeling of the operator
- intelligent and direct expert control

The application of expert systems in modeling and system identification and in control system design will only be touched on in this paper, because of their mainly off-line character and their emphasis on explanation and user-guidance facilities. However, the development of a combination of a real-time control system which automatically asks for advice and initiates a design and identification session is a most challenging prospect.

The paper is organized as follows. Some general reflections on feedback control and artificial intelligence are given in Section 2, where different ways to use AI methods in feedback control are reviewed. Section 3 deals with the case when an expert system is used to supervise different control strategies, this is called indirect expert control because there is a weak coupling between the control system and the expert system. In Section 4 we discuss direct real-time expert control where the expert system is more tightly coupled to the control system. There is a special class of direct real-time expert control systems that have been used in industry for some time. They are based on modeling the actions of a good operator by rules. Such systems are discussed in Section 5. Section 6 gives a brief treatment of learning systems, which appear to be a very promising area both from the research and the application point of view.

2 Feedback Control and Artificial Intelligence

Historically, feedback control and artificial intelligence have common roots in early cybernetics. After the initial development stage the fields, however, evolved in different directions. A major difference is that AI has almost exclusively focused on static problems while dynamics is the key issue in feedback control. Today, however, there is a growing awareness that AI techniques may be useful in control systems. This is manifested by manufacturers of control systems that are exploring AI and by vendors of AI systems who are looking into process control applications. See Kawakita et al (1988), Moore et al (1987), Oyen et al (1988), Reynolds (1988) and Sachs et al (1986). Among the applications considered we find monitoring, diagnosis, alarm handling, quality control, design, planning and scheduling and control. It is, however, less clear precisely where the advantages may lie. In this section we present an overview of the different application areas. Some of them will be treated in more detail in the following sections.

Algorithms and Heuristics

The development of control theory has for a long time been characterized by algorithm development, both algorithms for on-line control and algorithms to design control systems. In control engineering practice there are, however, many problem areas that are not handled by algorithms alone. It is, for example, necessary to have a good knowledge of a control problem: including the dynamics of the process and the disturbances, specifications and implementation constraints to be able to choose a particular algorithm. This knowledge is difficult to represent in algorithms but it can conveniently be represented in rules or semantic networks. Because of the focus on algorithms the other aspects of control system design have also largely been disregarded in the control research community. The possibilities of representing the design knowledge using AI techniques may refocus the interest.

Supervision Logic

In the actual practical implementation of control systems the algorithms are only a minor part of the code in a control system. Apart from the man-machine interface the major part of the code in a control system is actually the logic that surrounds the control algorithm. This logic takes care of switches between manual and automatic control, bumpless parameter changes and anti-windup in simple controllers. In more complex controllers it also handles the supervision of automatic tuning and adaptation. It is also a common experience that the effort required to implement and debug this code is significant. Since the supervision code is easily expressed in logic it is a natural candidate for use in an expert system.

Merger of PLC and DDC

In process control there are two types of automation systems, the continuous time control executed by the control algorithm (DDC) and the discrete logic and sequencing (PLC). It is a clear trend that these systems are merging. Suppliers of DDC systems are adding logic and sequencing, and suppliers of PLC systems are adding PID algorithms. An expert system can be used as an alternative to logic and sequencing or as an extension.

Merger of MIS and DCCS

In Plant-wide Control there is a merging between Management Information Systems (MIS) and Distributed Digital Control Systems (DCCS). Suppliers of both systems are developing software and network facilities in order to to realize an information processing system for the complete framework of operational and control tasks. At various levels of automation expert systems are introduced. It is clear that these systems should process knowledge related to the demands of their respective automation level, but also should pass, receive and distribute knowledge to their neighboring levels in a concentrated form. A distributed expert system can be used to guide and supervise the shut-down and start-up procedures and the transients between different modes of operations in a multi-product plant under various load conditions.

Modeling and System Identification

Modeling and system identification are important elements in solving a control problem. Conventional static and dynamic models contain a wealth of knowledge which can be exploited in many ways. There are, however, also situations where the models are not known with sufficient accuracy or where they are too complex. Qualitative physics and qualitative modeling developed in the AI community may be an interesting complement. See Bobrow and Hayes (1984), de Kleer and Brown (1984), Forbus (1986) and Kuipers (1986). The field of system identification has developed significantly over the past 30 years. A lot of the knowledge developed has been included in software packages like Idpac and Matlab. Considerable expertise is required to use these packages. Two types of knowledge is required, knowledge about system identification and knowledge about a particular package. It has been shown that both types of knowledge can conveniently be represented using scripts and rules. See Larsson and Persson (1986, 1988a, 1988b).
A challenging feature of these sytems can be experiment planning by the in-line application of system identification and modeling procedures. On the basis of the acquired and required knowledge, a selection of experiments is made, restricted by available or allowed measurement time, allowed signal magnitude, the frequency domain to be excited, etc.

Control System Design

Control system design is an area which requires expertise. Several attempts have been made to capture this expertise in a knowledge based system. See e.g. Taylor, Fredrick, James et al (1987), MacFarlane and Ackermann (1987). In the first experiments the use of conventional expert system shells was attempted. It appears, however, that it would be highly desirable to have tools that can also interact with conventional models of control systems and conventional control design algorithms.

Automatic Tuning Devices

There are several devices on the market that attempt to help a user to tune a controller. Examples are the Supertuner and the Protuner. These devices typically carry out some type of system identification from plant experiments and then give the recommended controller tuning. Related techniques are used in some of the single loop controllers with automatic tuning. See Bristol and Kraus

(1984), Kraus and Myron (1984) and Åström and Hägglund (1988). Although these devices are useful it is clear that the tuning of a controller is not always uniquely determined by the process dynamics. It also depends on the purpose of control. A typical example is level control where the purpose can be tight level control as well as surge tank operation when it is desired that the level swings over the full range. From this viewpoint it appears reasonable to have a more sophisticated system for tuning advice that can also can take the purpose of control into account. It would also be highly desirable to have design data in such a system because sometimes good tuning parameters can be computed from design data. It is also clear that applications in control system design and in automatic tuning are closely related.

In sophisticated controllers such as adaptive and predictive control algorithms a number of parameters have to be preset by a control engineer and to be tuned or supervised by an intelligent system. A few parameters are, however, strongly related to the system requirements and should be tuned by the operator or control engineer. These parameters represent the key parameters in a control loop design, like bandwidth, overshoot and noise reduction. These parameters are the controller knobs of the controller and are translated to parameters inside the controller algorithm which can be completely irrelevant for the user. An example of the tuning of a self-tuning adaptive controller is given in Krijgsman et al. (1988). A knowledge-based system forms the natural link between the idea of the control system designer as to how the system should behave, and the mathematical description of the process. Using the correct calculations and heuristics it sets the right parameters of a controller or chooses the right controller configuration. An example is the choice of the different controller configurations using the Unified Predictive Controller, see Soeterboek et al (1989). Many parameters should be chosen during the design phase, and a number of parameters, related to the given system behaviour, adapted during operation. A rule base is set up to manage this task in an efficient way. Rules can be added when more experience of the controller or the process has been gained. The number of rules is relatively low and to speed up the procedure the knowledge-based system can be translated into conventional software.

Learning Systems

Learning control is perhaps the most interesting point of tangency between control and AI. There are strong similarities between the learning algorithmns of AI systems and adaptive control algorithms. There are also marked differences. Adaptive control systems are often more structured and they use more apriori data. The typical learning algorithms are more general and less structured. It would be an interesting task to attempt a merger of the ideas.

3 Indirect Expert Control

Perhaps the most direct way of using a knowledge-based system for feedback control is to realize that any controller contains algorithms and logic. A natural separation is then to stucture the system in algorithms and logic, possibly in a hierarchical structure as indicated in the introduction. At the lowest level we would then have simple control algorithms of the PID type, possibly with automatic tuning and scheduling and supervision. The heuristic knowledge of how the system should be run in different modes is then implemented in terms of rules. An advantage of such a system is that it gives a clean separation of algorithms and heuristics. Systems of this type have also been used to prototype PID controller with scheduling, tuning and adaptation although the systems have been hardcoded for production.

Supervision and Tuning of PID Controllers

Most industrial control problems are today solved by PID control. Because of the benefits of good tuning, several attempts have been made to tune regulators automatically. One idea based on pattern recognition was developed at Foxboro, see Bristol (1977), Bristol and Kraus (1984) and Kraus and Myron (1984). Another successful approach is based on relay feedback, see Åström and Hägglund (1984, 1988a).

In the pattern recognition approach a PID controller is connected to the process. The response to step changes or disturbances is observed and the controller parameters are adjusted, based on the response pattern. The procedure mimics the procedure used by an experienced process engineer. It requires that reasonable controller settings are known prior to the experiment and it can be implemented as a rule based expert system. This expert system is composed of a transient analyser that determines the damping and the frequency of the closed loop system based on a transient, and a PID designer which is a collection of empirical tuning rules.

The relay auto-tuner is based on the idea that knowledge of the ultimate frequency, i.e., the frequency where the phase lag of the open loop is 180 degrees, the crucial information for tuning a PID controller. The ultimate frequency can be determined from an experiment with relay feedback. When tuning a loop the process is first brought to steady state operation under manual control. When tuning is desired the process is then connected to relay feedback. A limit cycle oscillation is then obtained. The controller setting is calculated from the amplitude and the period of the limit cycle and the controller is automatically switched to PID control. It is practical to introduce hysteresis in the relay to avoid chattering due to noise, and a feedback so that the limit cycle oscillation is kept within specified limits. The ultimate frequency is determined from zero crossings and the ultimate gain from the peak amplitude of the oscillation. The measurements of separate half-periods of the oscillation can be compared to establish that a steady state oscillation is obtained. The only prior information that has to be provided is the initial relay amplitude. The hysteresis of the relay is determined by measuring the noise level.

A system with relay auto-tuning can conveniently be described in terms of algorithms and logic. The algorithms will cover relay feedback, noise analysis, limit cycle analysis, computation of PID parameter, and PID control. The logic will cover the mode switches and the tests for switching between the operating modes. The operation of the system is conveniently described using a script. See Shank (1986).

Gain Scheduling

Gain scheduling is a powerful technique to handle variations in process dynamics when there are measurable signals that correlate well with the changes in dynamics. With automatic tuning it is easy to obtain gain schedules automatically simply by having a table for different operating conditions and to save the parameters obtained when tuning for different operating conditions. The implementation of such a system can also be straightforwardly expressed in terms of scripts, rules and algorithms. These ideas have been incorporated in simple single loop controllers. See Hägglund and Åström (1989).

Assessment of Control Performance

Automatic tuning and adaptation are now well established for simple PID controllers. These controllers can be said to automate those tuning functions normally performed by instrument engineers. To achieve a higher level of automation it is useful to have systems that can also reason about specifications and achievable performance and guides in the selection of controllers. Eexpert systems are well suited to the solving of these problems. Applications are discussed in Åström (1988) and Åström et al (1989).

Supervision of an Adaptive Controller

The key elements of an adaptive controller are a control algorithm, a recursive parameter algorithm and a control parameter calculation. Since an adaptive controller is more complex than a PID controller it also requires more supervision. It is also natural to have a simple PID controller as a backup controller. The supervision of and the adaptive controller itself can be structured as follows:

Main Monitor	Stability and control quality supervisor
Estimation Monitor	Excitation.
	Estimates and covariances.
	Drift and jump detection.
	Perturbation signal generation.
Back-up Control	PID control. Automatic tuning.
Design Supervision	Controllability and observability tests.
	Specification supervisor
Scheduling Supervisor	Scheduling tables. Quality test.

More details are given in Åström et al (1986) and Årzén (1987, 1989).

Example - Expert Control

It is natural to consider a controller which consists of several different algorithms orchestrated by an expert system as was suggested in Åström et. al. (1986) and further elaborated in Årzen (1988, 1989). We call such a system an *expert control system*. The architecture of the system is shown in Figure 1. The system consists of several algorithms for control and estimation, for example a PID controller, a PID tuner, a gain scheduling table, a least squares recursive estimator and a pole-placement self-tuner. The system also has other algorithms for supervision and analysis and signal

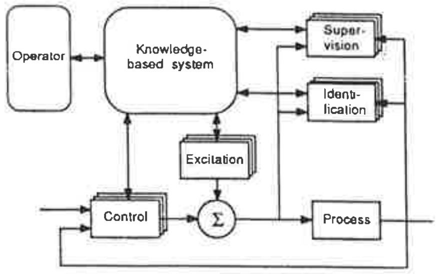

Figure 1: Block diagram of an expert control system.

generation algorithms to improve identifiability. All the algorithms are coordinated by the expert system that decides when to use a particular algorithm. The expert system architecture separates the control algorithms from the logic and it supplies a convenient way to interact with the system. The architecture also forces a disciplined structure. A blackboard structure has been found to be very useful. Such a system which has forward chaining and backward chaining as well as procedural knowledge sources has been implemented by Årzén (1987, 1989).

It is interesting to compare an expert control system with an auto-tuner. The systems are similar in the sense that they both can provide reasonable parameters for a PID regulator. However, the expert control system can also interact with the user in a much more effective way. It can provide answers to queries like, Is the system running normally? List all loops that have been tuned last week. List all regulators where derivative action is used. List all loops where dead-time compensation seems to be required. List the 10 loops with the poorest performance. A functionality like this is certainly useful both for operators and for instrument engineers.

Strategy Switching

Depending on the system requirements and the actual behavior of the system it can be very useful to change the controller configuration. Especially in a MIMO system it can be advantageous to reconsider the connection between controlled variables and controller actions. Due to the mode of operation, the production level and the properties of the raw materials to be processed, an adaptation of the control strategy can be advantageous. In a SISO system the strategy switching can be related to the dynamic selection of a control algorithm (bang-bang controller, P, PI or PID controller). In Baars et al (1987) an example of strategy switching using an expert system in the temperature control system of large buldings is described.

4 Direct Real-Time Expert Control

In a conventional or sophisticated control system the controller design is based on fundamental knowledge, described by mathematical equations (state equations, transfer functions, etc), deduced from physical laws and experimental data. This knowledge can be divided into structural knowledge (order of the process, noise characteristics, etc) and in parametrized and numerical knowledge (i.e. parameter values of the transfer function, value of the delay time, etc). In direct expert control (d.e.c.) a knowledge-based system using qualitatives replaces a controller based on fundamental knowledge, see Figure 2. D.e.c. is based on the experience of the operator and control engineer as well as on the observations of the process and control variables. Relationships among variables may be known or assessed in qualitative terms. Usually the knowledge-based system contains a mixture of qualitative and fundamental knowledge. A d.e.c. approach is less useful for linear systems with well known parameters, but can be applied succesfully in those cases where the process is highly nonlinear or hard to describe while existing theories do not cover the analysis and design of those systems. Because of their very nature d.e.c. systems lack conventional characteristics such as guaranteed stability of the contol loop, consistency and desired prescribed performance. However, by careful supervision also based on an expert system, acceptable control behavior and a certain amount of learning capability can be realized.

It is obvious that in d.e.c. guaranteed response times are crucial and high demands are put on the processing speed of the system. In industrially distributed digital control systems it is crucial that an expert system is embedded in the original automation hierarchy. A good allocation of tasks between the real-time monitoring and control environment and the expert system is necessary. It is very important that the expert system can be embedded in real-time software.
Most of the knowledge-based systems known today have been developed for diagnosis and classification purposes. The main emphasis is on explanation facilities and the handling of a large number of rules and not on time-critical behavior. In a real-time knowledge-based system (the possibility of dividing the knowledge into a multi-layer configuration is very interesting). Each layer has its own kind of knowledge and therefore its own rule base. This kind of reasoning is called progressive reasoning. The inference engine starts with the lowest layer. When a conclusion is obtained in this layer, this conclusion is stored and the "reasoning" continues in the next upper layer. This reasoning/conclusion storage cycle continues upwards to the next layers. As soon as the system is interrupted because a time-critical action should be performed (at the sampling time for example) the 'best' conclusion up to then is taken and the related control action executed.

Example: EXPERT-3

To make real-time experiments possible, a real-time knowledge-based system EXPERT-3 was developed at our laboratory. It is written in FORTH as an extension of EXPERT-2, see Broeders (1988). It contains a backward and forward chaining inference mechanism and a rule compiler to guarantee fast inferencing. The knowledge representation is based on production rules. An agenda-driven scheduling mechanism has been added to the backward chaining mechanism. There are facilities to compile more than one rule base.

EXPERT-3 operates on a 68000 system (ATARI ST) under Real-Time Forth (RTF) which is a small comprehensive development environment for the design, implementation, testing and debugging of microsystems. RTF offers multi-tasking; synchronization and communication between tasks can be performed by the use of semaphores and queues. The real-time facilities include timer and delay statements.

The controller consists of five hierarchically ordered layers. The first three layers classify the the process in an area of the phase plane, spanned by the error signal and its first difference. Experiments were performed with different shapes of the areas dividing the phase plane. The best results were obtained by using elliptically shaped areas. Successively the system is classified by:
- the signs of the error and its first difference (first layer, 4 rules)
- the size of the error (second layer, 6 rules)
- the size of the difference of the error (third layer, 2 rules) resulting in 48 areas in the phase plane.
Each layer classifies the system and calculates a proper control action. Then the next layer is initiated and the conclusion about the control signal is overruled as soon this layer comes to a conclusion and has calculated the related control signal. In the areas far from the origin, this will be the maximum or minimum control signal U_{max} or U_{min} or a fraction of U_{max} or U_{min}; in the areas near to the origin, the control action is calculated by:

$$u[k] = \alpha u[k-1] + (\beta U_{min} + \gamma(U_{max} - U_{min}))$$

The weighing factors α, β and γ are different for the several areas ($0 \leq \alpha \leq 1$, $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$).
The areas we use are elliptical shapes around the origin of the phase plane. They cross the $\Delta e(k) = 0$ axis in fixed points, while the radius on the $e(k) = 0$ axis can be varied by a factor ρ, where $\rho = 1$ means a circle. When this factor is very large the phase plane is almost completely filled by the outer areas, so the process is controlled using maximum and minimum output signals. The variation of ρ is obtained by some heuristic rules which express that the maximum or minimum controller output must be used for large setpoint changes.
Note that the controller actions could be compared to a proportional + reset action in the neighborhood of the origin of the phase plane. When the distance to the origin is large the controller is similar to bang-bang control.
The overall aim of our controller is to bring the system to the origin of the phase plane in a cautious way without any overshoot. To realize this goal a fourth layer has been implemented. This layer is based on what is called a Model Reference Expert Controller (MREC). When the system enters a new area in the phase plane a first-order model refence trajectory is calculated (a straight line to the origin from the point reached by the system at the given sampling instant). A one sampling time ahead prediction is made of the next point in the phase plane along this reference trajectory. The position of this point is influenced by the time constant of the reference model and the sampling period and can be set by the control engineer or a supervisory controller. Also a one sampling time ahead prediction is made of the next point of the actual system, based on a linear or quadratic extrapolation. The prediction error between the predicted reference trajectory and the predicted trajectory of the actual system serves as the basis for the calculation of the control signal. The angle Φ between the two trajectories is used to realize MREC. Some simple rules were implemented, such as: when $\Phi > 0$ the control signal is increased and when $\Phi < 0$ the control signal is decreased, depending on the length of the error vector.

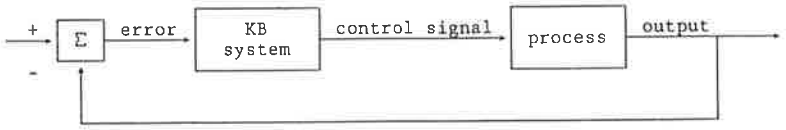

Figure 2: Block diagram of a direct real-time expert control system.

Only the rule sets that belong to areas at a distance not too close to the origin will use this kind of MREC knowledge. Other rule sets contain specialized knowledge such as: how the d.e.c. should act as a regulating controller. In this case it is important to keep the controlled process as close as possible to the origin, instead of steering it fast and secure to the origin. Thus, in the areas surrounding the origin MREC is not useful.

The task of the fifth layer is to supervise the performance of the lower layers. For example, observing the trajectory in the phase plane during the very first samples, allows the determination of whether the controlled system is a "fast" or a "slow" system and provides useful knowledge about the choice of the reference model time constant. These conclusions have a specific effect on the performance of the lower layers; e.g. in the case of a "fast" process, they start calculating the control signals in a more cautious way. Another example of the influence of the supervisor on the lower level knowledge layers is the adaptation of boundaries used to classify the controlled system in the phase plane.
The third function of this layer is to recognize the entry of the steady state area in the phase plane. In this case, the fourth knowledge layer is switched off and all boundaries, α's, β's and γ's that are used in the first three layers are decreased by a scaling factor. In effect, the whole phase plane zooms in on the steady state area positioned around the origin. This zoom function should ensure a zero steady state error.

The forward chaining mechanism is used for the reasoning in the first four layers while the backward chaining mechanism is used in the fifth layer. Observing the knowledge trees in these layers, it can be shown that this is the fastest way to achieve a solution for the respective layers. An agenda control algorithm within the backward chaining inference engine not only assures that the most likely to be concluded hypothesis will be investigated as soon as possible (heuristic search), and that the most important hypothesis will be searched for first, but also that the hypotheses which can never be proved at a specific moment are not investigated. To achieve this last possibility, metaknowledge has been added to the fifth layer.
To summarize, it can be stated that this supervisory layer does not provide a control signal but helps the lower level layers (and itself) to perform their job in a better way. It contains a number of heuristic rules based on general system behavior, provided by the operator and control engineer in order to enable a choice between different strategies and to set parameters in the lower level layers of the d.e.c. system.

To give an idea of the response time of this system we carried out many experiments. The absolute minimal sample time for d.e.c. with EXPERT-3 on the ATARI-ST is 55 ms. The system requires 430 ms to assure a response matching a first-order trajectory and a zero steady state error for processes that have time constants greater than 2 seconds. The fifth knowledge layer adds about 50 ms to the calculation time. We performed the same type of experiments on real-time systems using the commercially available expert shell NEXPERT embedded in our real-time environment MUSIC on a VAX-station. This combination is an excellent tool for the prototyping of dedicated expert systems. A much higher performance in terms of real-time reaction time could be obtained by implementing the final rule-based system in the conventional programming language C.

Currently, we are investigating processes controlled by a d.e.c. system with non-uniform sampling. While linear digital control design is based on uniform sampling there is no reason for uniform sampling in a d.e.c. system. That means adaptive sampling, based on the measured state of the system, can be used by applying some heuristic rules. It is also possible to use predicted error signals to influence the parameters of the given control laws or to introduce additional terms in this control law, introducing a kind of derivative action in the d.e.c.

Figure 3 gives the results of an experiment with a second-order nonlinear system controlled by this direct expert controller. The process

$$H(s) = \frac{2}{(10s+1)(25s+1)}$$

is preceded by a dead zone from -3 to 3. In spite of this nonlinearity the process is controlled in an acceptable way, without having any exact information about the system (unknown order, time constants, gain and no knowledge about the existence and type of the nonlineairity) ; only the sample period is chosen beforehand.

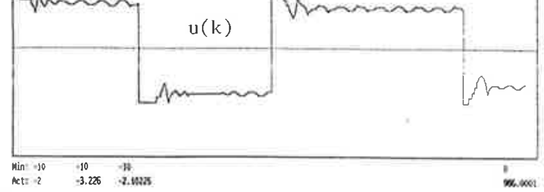

Figure 3: Experiments using EXPERT-3 to control a second order nonlinear plant

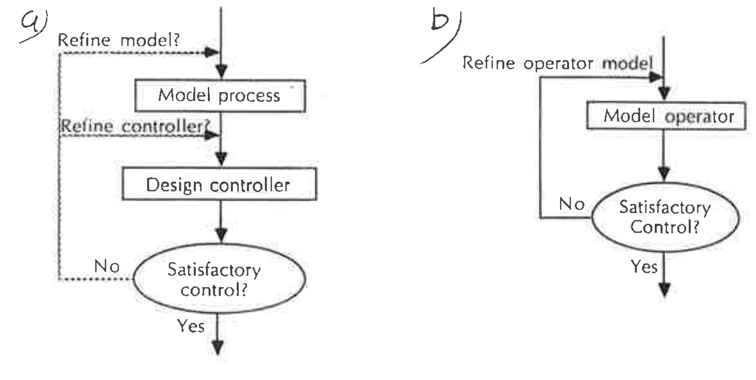

Figure 4: A comparison between a) conventional control and b) control based on modeling of the operator

5 Modeling of the Operator

For some difficult control tasks it has proven very difficult to follow the conventional path of modeling and control design. The main reason for the failure is that the processes are poorly understood. In spite of this it has been demonstrated that human operators are able to control the plants well when they wish to do so. Because the operation is manual there is, however, a significant variation in performance in time and between different shifts of operators. A typical example of this is control of cement kilns, whose dynamics are typically nonlinear and multivariable with long process delays. A different approach to control design has been developed for this class of problem, it is based on the idea of trying to make an automatic system that imitates the actions of the operator. The difference between conventional control and this approach is illustrated in Figure 4. In the conventional approach, the tuning effort consists of modifying both the model of the plant and the controller; in the operator modeling approach the controller is a direct modeling of the operator's decision processes. Successful applications of control based on operator modeling applied to cement kiln control have been reported by F.L. Smith in Denmark (Holmblad and Ostergaard, 1982) and by Blue Cirle Cement in the UK (Haspel et al, 1987). The Danish approach is called fuzzy control and the UK approach linguistic control.

Fuzzy Control

The idea of fuzzy sets is due to Zadeh (1965). The first control applications are described in Mamdami (1974). A system for controlling a cement kiln based on fuzzy control is described in Holmbland and Ostergaard (1982). The key idea is to make a crude quantization of the variables e.g. into five levels *very low*, *low*, *about right*, *high* and *very high*. The fuzzy controller then calculates the output using fuzzy logic. This fuzzy variable is then mapped into a real variable, representing the quantized change in control signal, which is fed to an integrator which drives the control signal. The resulting controller may be viewed as a multivariable, nonlinear integrating controller where the nonlinear function is represented by logic. It has proven very easy to develop the required logic from prior data, operator experiences and minimal experimentation.

Fuzzy control has recently received a significant amount of interest because the calculations required lend themselves very well to parallel computing. Chips for fuzzy control have been developed both in USA and Japan.

Linguistic Control

The lingusitic controller is also a constituent of a nonlinear controller with integral action. The controller is a nonlinear function which gives the changes in control as a function of measurements and command signals. The nonlinear function is described in terms of rules like:

> If the temperature of the furnace is a little high and the combustion gas oxygen level is on the low side, then reduce the kiln fuel rate by a small amount.

This description clearly indicates a crude quantization of the state and the control variables. The approach is therefore also called **linguistic control**. A more concise description of the rule above is:

> **IF** kiln temperature **high** and O_2 **low THEN** +5% feed and 0% fuel

A system called LINKman (Haspel et. al. 1987) has been developed which makes it very easy to implement the system. The system also runs efficiently. In contrast with a normal expert system, the system lacks the power of explanation.

Experiences

Fuzzy control and linguistic control are very similar. Experience from several installations indicates that it is possible to mimic the actions of good operators using rules. In practice, the operator's knowledge is also combined with the knowledge of process engineers, process designers and R&D departments into a coherent strategy. Significant performance increases have been noticed. It is claimed that they derive from using the same strategy consistently. The deviations due to delay and the overreaction of human operators to disturbances and perturbations due to shift changes are avoided. Compared to PID control, the major difference is that the control strategy obtained is nonlinear and multivariable. The rule based systems are also claimed to be easy to commission.

6 Learning Systems

It has been a longstanding goal of control engineers to develop control systems that can learn more and more about the process and its environment as they operate. Devices for automatic tuning and adaptive controllers (Åström 1988) are simple examples of such systems. Much higher levels of learning do, however, occur in biological systems.

The way biological systems are controlled and the learning abilities of rational living beings, based upon experience and observations, has always fascinated control engineers and researchers in Artificial Intelligence. In most cases, the control and learning behavior seems not to be based on fundamental knowledge, but is build up by performing many experiments and applying the gained knowledge to speed up learned actions or to react in a rather similar way in comparable circumstances. This behaviour stimulated the research on neural networks, describing in a certain sense the functioning of the brains of rational beings. Control engineers are, however, not primarily interested in a possible description of the internal structure of the information processing of the brain. They are mainly interested in algorithms that, using a simple black box model, describe reasonably well some of the learning capabilities of the brain.

Neural Networks

Both fuzzy control and linguistic control can be described as nonlinear functions which compute the changes in the control signals. The structure is thus a nonlinear integrating controller. The nonlinear function is derived from the knowledge of operators and process engineers. A neural network is an alternative way of implementing the system. This has the advantage that the nonlinear function can be learned automatically from the actions of an experienced operator. Early attempts in this direction are the experiments with Adaline by Widrow (1963) and BOXES by Chambers and Michie (1968).

A Learning Fuzzy Controller

Another way to introduce learning in a fuzzy controller will now be discussed.

At Delft University, a system has been developed which uses a pattern recognition mechanism and fuzzy decision making, see Van Der Rhee (1988). The fuzzy system uses a data structure built up with cells. The data is stored in cells representing the input and output data at one sampling period; it is represented as a 2-tuple (u_k, y_k) of a single-input single-output process at time $t = kT$, T is the sample time.

The system is used in two phases:

- A learning phase, in which the knowledge about the system is learned and stored in a data structure (a data base of stored input and output data).
- An application phase, in which this knowledge is used by retrieving data from the data structure.

The search algorithm used to find the correct data in the proposed data structure substantiately determines the succesful application of the system. The data structure is organized according to two types of relations:

- Relations in time: with the 2-tuples (u_k, y_k), $k = 0, 1, ..., n$, these relations represent information about the process behavior (time responses).
- Relations in signal magnitudes representing the information about similar (parts of) responses.

An item (2-tuple) stored in the data structure is called a basic cell. Of course, other types of cells can be used to store data in a different way.

The learning phase is used to store input and output data in the data structure, activating the process by a well-suited learning signal. A number of responses of different length can be stored in the data structure. No actions are performed to check the consistency or the redundancy of the responses stored in the data structure. Actually there is no processing of data during the learning phase.

During the application phase a number of algorithms can be applied to the data of a process stored in the data structure during the learning phase. At this moment the system can be used as a predictor and as a controller. Both algorithms evaluate time responses in the data structure: pattern recognition methods are used to compare responses stored in the data structure with the actual behavior of the process based on fuzzy reasoning.

A search algorithm is used to find the relevant data in the data structure in a particular situation. It is obvious that the efficiency of the search algorithm determines the minimum sampling time of the fuzzy system.
When we look at possible industrial application of this system we must think of systems that are very hard to model in a mathematical way. The process may not vary too much in time. At this moment the application as a fuzzy controller takes too much time to be useful in the majority of real-time applications. Research is going on in order to improve the efficiency of the method through findeing a different way to store and retrieve information through efficient hashing schemes and by defining other cell concepts (meta-cells) which can contain more information through compressing parts of the responses.

Albus Cerebellar Model Articulation Controller

The algorithm treated in this section is a very simple representative of a neural network, called CMAC (Cerebellar Model Articulation Controller), and was proposed by Albus (1975). It is based on ideas about the possible functioning of a particular area of the brain called the cerebellum, which area controls the trained movements of the muscles. The application of CMACs are described by Miller (1987), Betz (1988), Handelman (1988) and Vlothuizen (1988).
The method is based on a table lookup technique, different from a classical table lookup. The algorithm produces output values related by this table to multiple input variables. When the CMAC is used as a modeling device it predicts the output of the process based on previous process inputs and outputs, called the input state. When the

CMAC is used as a controller an adequate process input signal should be produced. Depending on the input values fed to the CMAC several table locations are selected and an output value is obtained by a summation of of all values stored in the selected table locations. The main problem is how the table should be filled in and build up.
When we first look at one input the following procedure is performed: the value of the input is quantized using a number of "quantizing functions". Each quantizing function maps the input value in exactly one address of an associated table. By applying a number of quantizing functions, each with a small offset in the active range of the input values, two effects will be produced: several addresses in the table will be selected in parallel (situated in completely different locations) and the resolution of the input variable will be enhanced.

If there are several inputs to the CMAC (i.e. inputs and outputs of a process at different sampling instances) each input is quantized by a number of quantizing functions. Thus, each input produces a number of addresses in the associated table. When we look at all possible combinations of quantized input variables, the size of the necessary memory could be very high and is related to the number of inputs, N, the number of quantizing functions, K, and the number of addresses produced by the quantizing function, Q.
The memory space needed comprises KQ^N addresses. When we call this memory the virtual memory we have to map this virtual memory to a much smaller memory, called physical memory, by a number of transformations. It is clear that the reduction in memory can lead to what is called collisions, i.e. several different inputs are mapped to the same memory location. To minimize the probability of collisions between distant input values, a uniform and pseudo-random mapping should be used. The mapping must be deterministic because the same input value must always select the same memory locations. Attention has to be paid to the problem of finding such a mapping (hashing scheme), see Venema (1989). The effect of a collision depends on the number of memory locations which are common for two different inputs. Note that collisions between nearby input values will not deteriorate the behavior of the CMAC very much. After restricting the number of memory locations to a manageable proportion, we focus our attention on the last step in the algorithm: the generation of the output value of the CMAC. The output due to a given input vector is generated by summing the weights stored in the assigned physical memory locations. The number of weights depends on the number of quantizing functions and determines the generalization effect of the CMAC. When a CMAC is used as a modeling device, the CMAC should first learn about the process by looking at the process in- and outputs. In the learning phase the weights are updated by comparing the output of the CMAC with the real output of the process. Only a limited number of points have to be learned by the CMAC due to the generalization effect. Because nearby inputs influence each other, not all possible points have to be learned. Interpolation between two learned points is used if the distance is equal to the number of quantizing functions. Figure 5 shows the result of an example using CMAC to predict the output of a second-order process. After a certain learning period the output of the process is predicted very well. Successively the input signal to the process, the process output and the output predicted by CMAC are shown.

To use CMAC as a controller of a dynamic process it should be noted that it does not exhibit memory for former input values. Therefore, the dimension of the input vector should be chosen high enough. Initially CMAC can be learned, while the process is controlled in a more or less conventional way. CMAC observes the response of the process and the inputs applied. If similar behavior is desired in the future, CMAC will remember what input signal should be applied to the process. After a certain period of training, CMAC can control the system, although it might keep on learning for further improvements or be able to respond to changes in the process.
It is interesting to note the analogy to the way movements are learned by the brain. Initially, movement is jerky and slow and needs the full attention of the brain. After training, movements are controlled subconsciously by the cerebellum in a kind of play-back mode. After a learning phase which could be short if the right parameters are chosen [Venema], a very fast method for predicting output signals or producing process input signals is obtained due to the table lookup principle. CMAC is capable of controlling nonlinear processes and processes which are difficult to model with a speed independent of the complexity of the process. A problem is the validity of the table lookup. If some of the parameters of the process change, the stored information which is not explicitely related to the process parameters can not be adapted immediately, while the information about the process, due to the mapping procedure, is spread all over the table. Therefore questions arise as to the validity of data, although by a so-called learning factor continuous updating of the table can be performed.

7 Conclusions

In this paper an attempt has been made to review some of the current attempts to introduce ideas from artificial intelligence into feedback control systems. Expert systems seems well suited for the supervison of different control algorithms They offer a good way to structure the systems in such a way that the supervison logic is well separated from the control algorithms. There are also interesting possiblities to introduce knowledge as to when a particular control algorithm can be applied in the feedback system, hence offering control systems with a higher level of automation. A key difficulty which has not been solved properly is the finding of suitable mechanisms for reasoning about time.

Knowledge-based and intelligent systems can be used for direct and indirect control. In the latter case they are able to control unknown, nonlinear or time-varying systems provided that they are supervised. A supervisory layer is necessary to guarantee the stability and consistency of the control system. The most useful features of Learning Systems are the speed of operation and the possibility to control non-linear processes and processes that are difficult to model. A combination with knowledge-based systems can be used to ameliorate the system behavior by checking the consistency of gathered information and tuning some of the learning parameters of the learning system. Intelligent systems will penetrate into all levels of a multi-layered information and control system. Provisions should be considered to pass on adequate information between the different levels of automation and to translate this information to key parameters relevant to the given levels. Finally, it should be stated that intelligent control is a very promising

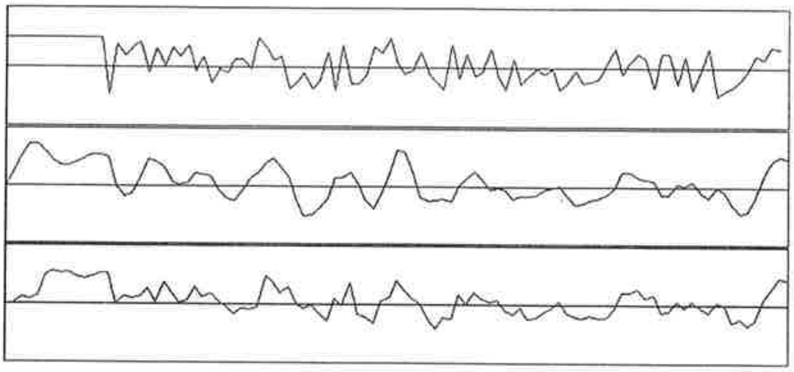

Figure 5: CMAC as a predictor

and challenging field of research, however, in many cases a well-tuned conventional or adaptive controller will perform satisfactorily and should not be replaced by a more sophisticated, intelligent controller as long as there is no evidence that this is necessary because of the possibilities of serious malfunctioning under certain conditions.

Acknowledgement

The research at the Department of Automatic Control of Lund University of Technology has been supported by the Swedish Board of Technical Development (STU) under contract 85/3084T.

The research at the Control Laboratory, Delft University of Technology, was performed in the project "Real-Time Expert Control". Special thanks to Piet Bruijn and Ardjan Krijgsman for their support in writing sections 5 and 6.

References

[1] Albus, J.S., *A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC),* Transactions of the ASME, pp. 220-227, September 1975.

[2] Albus, J.S., *Data Storage in the Cerebellar Model Articulation Controller (CMAC),* Transactions of the ASME, pp. 228-233, September 1975.

[3] Åström, K.J. and T. Hägglund (1988), *A New Auto-Tuning Design,* Proceedings IFAC International Symposium on Adaptive Control of Chemical Processes, Copenhagen, Denmark.

[4] Åström, K.J. and B. Wittenmark (1989), *Adaptive Control,* Addison-Wesley, Reading, Ma.

[5] Åström, K.J. (1988), *Assessment of Achievable Performance of Simple Feedback Loops,* 1988 IEEE Decision and Control Conference, Austin, Texas.

[6] Åström, K.J. (1987), *Adaptive Feedback Control,* Proc. IEEE, 75, 185-486.

[7] Åström, K.J. and T. Hägglund (1984), *Automatic Tuning of Simple Regulators with Specification on Phase and Amplitude Margins,* Automatica, 20, 645-651.

[8] Åström, K.J., J.J. Anton, and K-E. Årzén (1986), *Expert Control,* Automatica, vol. 22, no. 3, pp. 277-286

[9] Åström, K.J. and T. Hägglund (1988), *Automatic Tuning of PID Regulators,* Instrument Society of America, Research, Triangle Park, N.C.

[10] Åström, K.J. (1988), *Dominant pole placement design of PI regulators,* Internal Report CODEN: LUTFD2/TFRT-7381, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[11] Åström, K.J., C.C. Hang and P. Persson (1988), *Heuristics for assessment of PID control with Ziegler-Nichols tuning,* Internal report CODEN: LUTFD2/TFRT-7404, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[12] Baars, W., P.M. Bruijn and H.B. Verbruggen (1987), *Some Experiences with Expert Systems for On-Line Control.* Journal A, vol.28, no.3.

[13] Betz, R.E. and others, *An AI Based Controller for Dynamic Systems,* Proceedings of the IFAC Workshop on Artificial Intelligence in Real-Time Control, Swansea, UK, September 1988.

[14] Bobrow, D.G. and P.J. Hayes (Eds) (1984), *Qualitative reasoning about physical systems,* Artificial Intelligence, Special Volume 24, (1-3).

[15] Bristol, E.H. and T.W. Kraus (1984), *Life with Pattern Adaptation,* Proc. 1984 American Control Conference, San Diego, CA, pp.888-892.

[16] Broeders, H.M.T. (1988), *Real-Time Expert Systems,* Report A88.057 (in Dutch), Control Laboratory, Fac. of El. Eng., Delft University of Technology.

[17] Gupta, M.M. (1988), *Design of Cognitive Feedback Controller,* Proceedings of the 27th Conference on Decision and Control, Austin Texas.

[18] Hägglund T. and K.J. Åström (1989), *An Industrial Adaptive PID Controller,* IFAC Symposium on Adaptive Systems in Control and Signal Processing, Glasgow, UK.

[19] Handelman, D.A., S.H. Lane and J.J. Gelfand (1988), *Integration of knowledge-based system and neural network techniques for robotic control,* Proceedings of the IFAC Workshop on Artificial Intelligence in Real-Time Control, Swansea, UK.

[20] Hang, C.C. and K.J. Åström (1988a), *Practical aspects of PID auto-tuners based on relay feedback,* Preprints IFAC Int Symposium on Adaptive Control of Chemical Processes ADCHEM'88, Copenhagen, Denmark.

[21] Hang, C.C. and K.J. Åström (1988b), *Refinements of the Ziegler-Nichols Tuning Formula for PID Autotuners,* Proc. ISA Annual Conference, Houston, USA.

[22] Haspel, D.W, A.D.J. Lormier, C.J. Southan and R.A. Taylor (1987), *High Level Control,* IEEE San Fransisco, May.

[23] Haspel, D.W., C.J. Southan and R.A. Taylor (1987), *The Benefits of Kiln Optimisation using LINKman and High Level Control Strategies,* World Cement 10, No.6.

[24] Holmblad, L.P. and J. Ostergaard (1982), *Control of a Cement Kiln by Fuzzy Logic.* North Holland Publishing, pp. 389-399.

[25] King, P.J. and E.H. Mamdani (1977), *The Application of Fuzzy Control Systems to Industrial Processes,* Automatica, Vol. 13, pp.235-242.

[26] Kraus T.W. and T.J. Myron (1984), *Self-tuning PID controller uses pattern recognition approach,* Control Engineering Magazine, June.

[27] Krijgsman, A.J., P.M. Bruijn and H.B. Verbruggen (1988), *Knowledge-based Real-Time Control,* 1st IFAC Workshop on Artificial Intelligence in real-time control, Swansea, U.K.

[28] Krijgsman, A.J., P.M. Bruijn and H.B. Verbruggen (1988), *Knowledge-Based Control,* 27th IEEE Conference on Decision and Control, Austin, USA.

[29] Larsson, J.E. and P. Persson (1986), *Knowledge Representation by Scripts in an Expert Interface,* Proc. of the 1986 American Control Conference, Seattle, Washington.

[30] Larsson, J.E. and P. Persson (1988), *An Intelligent Help System for Idpac,* Proc. of the 8th European Conference on Artificial Intelligence, Technischen Universität München, München, pp. 119-123.

[31] Larsson, J.E. and P. Persson (1988), *The Knowledge Database Used in an Expert System Interface for Idpac,* Proc. of the IFAC Workshop on Artificial Intelligence in Real-Time Control, Swansea, UK.

[32] Lattimer Wright, M. and co-workers (1986), *An expert system for real-time control,* IEEE Software, pp. 16-24, March.

[33] MacFarlane, A.G.J., G. Gruebel and J. Ackermann (1987), *Future Design environments for Control Engineering,* Proc IFAC World Congress, Vol. 7, pp.363-369.

[34] Mamdani, E.H. (1974), *Application of Fuzzy Algorithms for the Control of Dynamic Plants,* Proc. IEEE No 12, pp 1583-1588.

[35] Miller, W.T. (1987), *Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm,* IEEE Journal of Robotics and Automation, Vol. RA-3, No. 2, April.

[36] Moore, R.L. and others (1987), *Expert System Methodology for Real-Time Process Control,* Proc. 10th IFAC World Congress, pp. 274-281.

[37] Oven, R.A., M.A. Keyes and M.P. Lukas (1988), *An Expert System Shell embedded in the Control System,* Preprints of the IFAC Workshop on AI in Real-Time Control, Swansea, U.K.

[38] Reynolds, D. (1988), *MUSE: A Toolkit for embedded Real-Time AI,* in R. Engelmore and T. Morgan (Eds.): Blackboard Systems, Addison Wesley.

[39] Rhee, F. van der (1988), *Fuzzy modeling and control based on cell structures,* Ph.D. Thesis, Control Laboratory, Fac. of El. Eng., Delft University of Technology.

[40] Sachs, P.A., A.M. Paterson and M.H.M. Turner (1986), *Escort - an Expert System for Complex Operations in Real Time,* Expert Systems, 3, 1, pp.22-29.

[41] Soeterboek, A.R.M., H.B. Verbruggen and P.P.J. van den Bosch (1989), *Unified Predictive Control - analysis of design parameters,* Proc. of the IASTED International Symposium Modelling, Identification and Control, Grindelwald, Switzerland, pp.164-168.

[42] Taylor, J.H., J.R. James and D.K. Frederick (1987), *Expert-aided Control Engineering Environment for Nonlinear Systems,* Proc IFAC World Congress, Vol. 7, pp.363-369.

[43] Vlothuizen, W.J. (1988), *The CMAC Method,* Report A88.071, Control Laboratory, Fac. of El. Eng., Delft University of Technology.

[44] Widrow, B. and M.E. Hoff (1960), *Adaptive Switching Circuits,* IRE Wescon Connection Record, New York, pp. 96-104.

[45] Zadeh, L.A. (1965), *Fuzzy Sets,* Inform. Contr., Vol.8, pp 330-353.

//