Expert Control

Åström, Karl Johan; Anton, John J

CODEN:LUTFD2/(TFRT-7271)/1-07/(1984)

EXPERT CONTROL

KARL JOHAN ÅSTRÖM
JOHN J. ANTON

DEPARTMENT OF AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY
MARCH 1984

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 725
S 220 07 Lund 7 Sweden

Document name
Report
Date of issue
March 1984
Document number
CODEN:LUTFD2/(TFRT-7271)/1-07/(1984)
Supervisor
Sponsoring organization

Author(s)
Karl Johan Åström
John J. Anton

Title and subtitle
EXPERT CONTROL

Abstract
There has been substantial progress is theory and practice of automatic control through application of mathematical analysis and numerics. Nonnumerical data processing has, however, so far only had marginal influence on control systems. Actual implementations of control systems also contain a substantial amount of heuristic logic. The paper shows that this logic may be replaced by an expert system. This leads to simplifications in implementation as well as new capabilities in the control system.

Key words
PID control. Relay logic. Auto-tuning. Adaptive control. Heuristics. Expert system.

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
7

Recipient's notes

Security classification

EXPERT CONTROL

K. J. Aström
Lund Institute of Technology
Lund, Sweden

J. J. Anton
Systems Control Technology
Palo Alto, CA, USA

Abstract. There has been substantial progress in theory and practice of automatic control through application of mathematical analysis and numerics. Nonnumerical data processing has, however, so far only had marginal influence on control systems. Actual implementations of control systems also contain a substantial amount of heuristic logic. The paper shows that this logic may be replaced by an expert system. This leads to simplifications in implementation as well as new capabilities in the control system.

Keywords. PID control. Relay logic. Auto-tuning. Adaptive control. Heuristics. Expert system.

1. INTRODUCTION

The purpose of this paper is to identify possible uses of expert system techniques in implementation of control systems. It is first observed that actual implementation of control laws often involves a substantial amount of heuristic logic. This is true for simple regulators as well as for more sophisticated multivariable control loops. Expert system methodologies provide a systematic approach for dealing with heuristic logic. Selected basic elements of an expert system are presented. Stochastic dynamic programming offers a framework in which the heuristics can be embedded. This points to requirements for a new artificial intelligence approach for heuristic planning under uncertainty. An example is sketched to illustrate the ideas. Once the expert system approach is taken it is also possible to obtain control systems with new functions. This is also illustrated.

2. THE PRACTICE OF AUTOMATIC CONTROL

There has been a very significant development of control theory over the past thirty years. This has led to many new ideas and concepts, as well as increased insight and new design procedures. The inspiration has largely come from two sources: mathematics and digital computing. However, this vigorous theoretical development has so far only had a modest impact on the practice of automatic control.

The making of a control system can be composed of the following activities: modeling, identification, analysis, simulation, control law design and implementation. It is fair to say that the development that has taken place during the past 30 years has had a drastic influence on identification, analysis and design. Implementation has changed in the sense that digital systems are now replacing analog systems. There are, however, many aspects of the implementation that have not changed much.

Although major progress has been made in linear and nonlinear systems theory, there are several instances where theory lags application needs. Typical examples are systems with selectors and anti-windup. Such systems are frequently approached purely empirically.

Consider for example an ordinary PID-regulator. Its linear behaviour can be described by the model:

$$u(t) = \left[e(t) + \frac{1}{T_i}\int^t e(s)ds + T_d\frac{de(t)}{dt}\right]. \quad (1)$$

PID-control can be understood very well from this linear equation, suitable values of the parameters can be determined, etc. However, in practice there are many important aspects which are not captured by the simple formula. To obtain a good PID regulator it is also necessary to consider operator interfaces, operational issues like switching between manual and automatic operation, transients due to parameter changes, nonlinear actuators, windup of the integral term, selectors etc. An operational industrial PID regulator thus consists of an implementation of the equation (1) and some heuristic logic that takes care of these issues. Although these heuristic factors are of extreme importance for good control they have not attracted much interest from theoreticians. They are instead hidden in practical designs and rarely discussed in the control literature. One reason for this is that the theoretical analysis is quite difficult.

We can thus conclude that practical solutions even to such mundane problems as PID control are not done by theory alone, but that heuristics play an important role. These heuristics show up in terms of logic that surrounds the implementation of the linear control law given by equation (1). The standard way of building industrial process control systems is to combine PID regulators, selectors, logic and sequencing circuits. It is a very noticeable trend that DDC (Direct Digital Control) systems and PLC (Programmable Logic Controllers) systems are merging. The design of such systems also involves a lot of heuristics.

Heuristics is even more important in multivariable and self-tuning regulators. In these case the fundamental control law is much more complicated than the control law given by equation (1). To obtain a well functioning adaptive control system it is also necessary to provide it with a considerable amount of heuristic logic. This goes under names like safety nets or safety jackets. Experience has shown that it is quite time consuming to design and test this heuristic logic.

The background considerations to this point have been directed at a very elementary configuration for a process control problem, namely, a collection of structurally similar control loops. There is one performance measure (small error) and the loops are running under manual or self-tuning control. A more sophisticated configuration, within the theoretical framework of dynamic programming, has been available for some time. In practice dynamic programming formulations have not been widely used due to computer requirements. See Bellman (1957). In fact some attempts have been made to use this framework, while making heuristic approximations to avoid some of the computational and storage demands of the pure algorithm. See Åström and Helmersson (1982). Still the focus has been on low level system components, such as isolated loops.

Although this paper refers to control activities for low level components, the main consideration for expert control is the higher level problem of control of a plant with all its interacting lower level components. At the higher level it is also possible to formulate the control problem in a dynamic programming framework. Stages are periods between changes in the control laws applied to the lower level components. States are "plant-wide". Decisions correspond to selections of the automatic activities (control law type for a given loop, estimation, etc.) carried out on the components. There are also multiple "plant-wide" performance measures. Heuristics are important at both high and low levels in this control problem.

3. HEURISTICS

In the previous section it was mentioned that heuristics plays a role in ordinary PID regulators and an even greater role in systems which combine PID regulators with logic, and in adaptive regulators. In implementations of the regulators heuristics shows up in the form of selectors, if-then-else or case statements. In many cases this part of the code may be much larger than the pure code for the control algorithm. The debugging, modification, and testing of the control logic can also be very time consuming. From a purely pragmatic point of view of efficient engineering it thus make sense to look at better ways of implementing the heuristic part of the code.

Once we accept as a fact that control algorithms will contain heuristics we may also ask what a more extensive use of heuristics may contribute to control systems. As an illustration let us consider typical adaptive algorithms like model reference adaptive controllers or self-tuning regulators that are currently being used. See Åström (1983). These algorithms may be viewed as local gradient algorithms in the following sense. Starting from reasonably good a priori guesses of system order, sampling period, and parameters, the algorithms can adjust the regulator parameters to give a closed loop system with good performance provided that the initial guesses are not too far off. The algorithms are also capable of tracking a system provided that the parameters do not change too quickly. The algorithms will however not work if the prior guesses are too far off. The sampling period is a typical case. Most digital adaptive algorithms will fail if the sampling period is too short. The present adaptive control algorithms thus have a limited range in which they will operate satisfactorily. Outside this range they may result in unstable closed loop system. This is in fact the development which has led to the safety jackets mentioned in Section 2.

There have recently been proposals for other types of tuning algorithms that have a wide range of operability, although they do not have the good local properties of the self-tuners. See Åström and Hägglund (1983). It seems appealing to explore the possibility of designing systems that combine a wide range of algorithms with different properties. To do this efficient ways of orchestrating different algorithms to achieve varying control objectives are needed. This problem has important analogues in existing expert system applications, so it is useful to look there for tools that may aid in solving the control problem.

4. EXPERT SYSTEMS

Expert systems is a rapidly expanding area within the field of Artificial Intelligence (AI). See Winston (1977), Waterman and Hayes-Roth (1978), Nilsson (1980), Barr and Feigenbaum (1982), Davis (1982) and Davis and Lenat (1982). One objective of AI is to develop computer-based models for problem solving. It is distinguished from physical modeling because it attempts to model those aspects of a problem, which are not naturally amenable to numerical representation or more efficiently represented by heuristics. One objective for expert systems is to model the knowledge and procedures used by a human expert in solving problems within a well-defined domain. Important examples of expert systems are documented in Barr and Feigenbaum (1982) and Davies (1982).

A typical expert system has three principal components:

1. System data base
2. Knowledge sources
3. Supervisory strategy

The System Data Base

The system data base is the repository of facts, evidence, hypotheses, and goals. For a process control example, the facts would include static data such as sensor measurement tolerances, operating thresholds, alarm level thresholds, constraints on operational sequencing, plant component configurations etc. Evidence includes dynamic data from sensors, instrument engineering reports, and laboratory and test reports.

A practical observation for plant operations is that evidence as described above is typically diverse in type, often noisy, somewhat delayed, possibly incomplete, and sometimes contradictory. An experienced process control engineer has techniques for dealing with these complications. He can develop hypotheses on-line to supplement the current collection of facts. In an expert system, hypotheses are also generated and stored in the data base to cope with the limitations of known facts or measured evidence. One important class of hypotheses in an expert control system will be the various state estimates made by parameter estimation algorithms. Including them under hypotheses acknowledges both that they are derived from evidence (data), and that their derivation is conditioned on other model assumptions.

It is worth commenting that the hypotheses are available in a way that permits external audit of system logic. The hypotheses are thus supported with the rationale and evidence for their creation.

Goals are other important entries in the data base. In an expert system they are usually both static and dynamic in nature. Static goals include the wide array of performance objectives like maintain stable operation, find optimal stationary operating points or determine if the current control law can be improved. Dynamic goals include those established on-line, either by external command or from the program itself.

Knowledge Sources and Knowledge Representation

Representation of knowledge in an AI system is often the most challenging aspect of the design. The knowledge sources in an expert system are models for the essential problem elements. In a process control application these elements include the portion of the operator's skill to be automated, the control and estimation algorithms that may be applied, the appropriate characterization of these algorithms, and judgemental knowledge on when to apply them.

A wide variety of approaches have been developed and used for the task of modeling knowledge representation. Among the more prominent are first order predicate calculus (logic), procedural representations, semantic networks, production systems (the AI versions of the if-then-else structure mentioned above), frames, see e.g. IEEE (1983). The kinds of knowledge necessary include:

- a characterization of the available control and estimation algorithms,
- indicators for invoking supervision, planning, and fault diagnosis, and
- instructions for supervision, diagnosis and planning.

Production rules and frame structures have some very desirable features for a process control application. Production rules operate on the entities in the data base, resulting in new entries and modifications of earlier ones. They may be viewed as functions operating on the state. Since the data base is broader in concept than the usual notion of state, production rules are also richer in content than common transition functions. For expert systems, production rules are typically described as: "if <situation> then <action>". The 'situation' is some collection of facts, evidence, hypotheses, and goals. The 'action' can include physical operations, parameter estimations, plausible inferences, activation of a new controller, or specification of a new goal to be pursued.

Frames are data structures for representing common situations. See Winston (1977). A frame can contain several kinds of information, including instructions for its use, what can happen next, what to do next, preconditions for an operation to be applicable, and so forth. A frame may be viewed as a network of nodes and relations. The top level nodes define the expanse of knowledge, that is, they present the collected topics. The lower nodes (called slots) are filled with specific information details on the higher level topic. Slots can be defined to limit the conditions to be met in their entries. Since these entries may be "frames" (subframes at lower levels), it is evident that this knowledge structure can become complex.

Frames provide an efficient approach to organizing expectations and presumptions. They provide a convenient mechanism for default assignment of information. They can be coded to produce in operation the expected entry, unless something causes this to be overwritten. They can also be used for making generalizations. Frames also incorporate a useful inheritance mechanism. For example, when a characteristic at a lower level slot is not available, it is possible to infer its value from slot entries in higher level frames.

Supervisory Strategy

The purpose of the supervisory strategy is to decide from the context (current data base of facts, evidence, hypotheses, and goals) which production rules to select next. Picture here the experience of a human operator who, given time and enough information, knows how to bring a process in line with required operational conditions. In an expert system this knowledge of planning what to do may also be represented in the same "situation-action" format used for the production rules or in terms of frames. Separating the supervision knowledge (what to do) from the production knowledge (how to do it in detail) offers a significant flexibility for developing and modifying a process control system.

Current process control systems have considered selection of alternatives in a very limited scope by hardwired logic. If the scope were widened (e.g., to include use of some adaptive control algorithms), the number of considerations to be taken into account may grow unwieldy. Rather than to preprogram a logic to treat each case in an explicit branching logic, the objective of expert control is to encode the knowledge sufficient to make intelligent decisions and recommendations automatically.

An on-line fault, a command to change production goals, etc., calls for a sequence of steps to bring the process in line with requirements. Each step in this plan will involve some action to adjust the process. The actions taken must not interfere with the preconditions of actions to follow in accordance with the control plan. With many actions available, and with many possible sequences, the development of a control plan can be recast as a search through a large network for that path that reaches the currently established goals. This searching and planning in a complex environment is a fundamental activity in AI systems. Consider the following to see why this might be needed.

In classical algorithmic control there are well-defined and highly constrained notions of state and state transition. These are associated with physical parameters and operations. In expert control we want to deal with more ambiguous, less constrained, more qualitative notions, in addition to the available quantitative knowledge. Control strategy development in classical control can use tools like dynamic programming with an exhaustive but efficient search in control space for a plan. Expert control is meant to deal with those process control problems where there are still too many alternatives with complicated interrelationships for search to be practical. In an expert system the problem of finding a plan for 8 steps with 4 options at each step, would not deal with all $4^8 = 64000$ possibilities. Heuristics might be found, for example, to fix the steps at plan stages 2, 4, and 6, whereupon the search would be reduced to $4 \cdot 4^2 = 64$ options.

Important work on expert systems for planning has been carried out in the context of robot motion, Nilsson (1980), Sacerdoti (1977), and genetics experiment design, Stefik (1981a,b). But the planning problem for expert control of a complex plant has elements not addressed in these works: uncertainty in the state, in the models, and in the outcome of an applied action. (Nilsson does treat the last consideration in his program STRIPS, op.cit.) The stochastic dynamic programming problem formulation provides a framework for the requisite planning. But this algorithmic approach which amounts to global search, may not be feasible. The use of an expert system has been turned upon this search task before with noteworthy success on a complex problem, Buchanen et al (1969). Before turning to this problem in its fullness, however, initial efforts will be limited to simpler cases. In the next section, although no specific planning formulation is put forth, some of the elements of the planning problem are described for the case of a single objective for a single loop.

5. AN ADAPTIVE REGULATOR

To illustrate the concepts a simple example in steady state control of an industrial process will be investigated. It is shown that expert control can give added performance to such a mundane operation. The different actions that we would like the system to perform are first given. The process state and different ways to organize the control logic are then discussed.

Operations

For simplicity we consider only a single regulation loop. Let the goal be to keep the fluctuations of a process variable close to a given set point with reasonable control actions. If the dynamics of the process and the disturbances were known a minimum variance regulator could be designed. Examples of such control strategies and the underlying theory are given in Åström (1970). If the process is described by the sampled model

$$Ay(t) = Bu(t-d) + Ce(t), \tag{1}$$

where u is the control variable, y the output, e white noise, and A, B and C are polynomials in the forward shift operator, then the optimal control law is

$$Ru = -Sy \tag{2}$$

where the polynomials R and S are given by

$$z^{d-1}\ CB = AR + BS. \tag{3}$$

The operations necessary to obtain and maintain the minimum variance control law given by equation (2) in a safe way will now be discussed.

In the terminology of Section 4 the "action" minimum variance control is thus the main function of the system. To apply such an action it is necessary to have a model (1) for the process, the disturbances and a sampling interval. With the simple minimum variance control law it is also necessary to insure that the preconditions for minimum variance control are satisfied. The most important condition is that the dynamics is minimum phase, i.e., that the process zeros are inside the unit disc. A particular feature of the minimum variance control law is that the process zeros are cancelled. This may lead to ringing if the zeros are not sufficiently well damped. The trade-off between input and output variance is governed by the prediction horizon d·h, where d is the delay in (1), h is the sampling period. Notice that if the process is stable then the model will always have zeros with arbitrarily good damping if the sampling period is long enough. See Åström, et al., (1983). To be able to detect ringing and to take appropriate actions it is useful to include a ringing detector.

There is a convenient way to find out if a process is under minimum variance control because the output would then be a moving average

$$y(t) = \lambda\ \left[e(t) + \ldots + f_{d-1}e(t-dh+h)\right]. \tag{4}$$

where

$$F = \frac{R}{B} \tag{5}$$

and h is the sampling period. A minimum variance supervisor can thus be based on the calculation of the correlation function of the output.

If the process model is not available a self-tuning regulator (STR) may be used. See Åström and Wittenmark (1973). Such a regulator may under certain conditions converge to the minimum variance regulator which could be designed if the process model were known. The simple self-tuner is in essence a parameter estimator. In the STR the model is reparameterized in terms of the regulator parameters. A parameter estimator will thus be included as an operator. The preconditions for parameter estimation is experimental data obtained when the process is properly excited. To ensure a proper operation of the estimator we will therefore also introduce an excitation supervisor. This would in essence determine the energy of the input signal in the useful frequency range.

If there is not enough excitation we have two options. Either to stop the updating of the parameters or else to introduce perturbation signals. In those cases when perturbation signals are allowed the system will be provided with a perturbation signal generator. The generation of the perturbation signal requires some information about the frequency range of interest and about the allowable perturbation levels. This may be derived from the knowledge of d·h.

The self-tuning regulator also requires prior knowledge. In particular the following data is needed for the basic STR. See Wittenmark (1973).

h	sampling period
d	delay in number of sampling periods
nr	degree of the polynomial R
ns	degree of the polynomial S
λ	forgetting factor
θ_o	initial estimate
p_o	initial covariance
uh, ul	high and low control limits

The parameters d and h are crucial because the closed loop system may become unstable if they are underestimated. To detect this the system should therefore be provided with a stability supervisor. There are possibilities to find out if the integers nr and ns are large enough simply by calculating the covariance functions $r_{yy}(\tau)$ and $r_{yu}(\tau)$. See Åström and Wittenmark (1973). We can thus construct a degree supervisor and include it in the system.

The importance of knowledge of the product d·h has been emphasized. A robust estimate of this quantity may be derived from the Ziegler-Nichols auto-tuner discussed in Åström (1982). This procedure gives the critical gain k_c and the critical period t_c. An kc-tc-estimator with some supervision, as is discussed in Åström and Hägglund (1983), is thus also provided. A safe estimate of d·h is actually $t_c/2$. When using the kc-tc-estimator we will also get data which is useful to estimate other parameters.

Other functions may also be provided. Assume that it is known that the process dynamics changes with a few parameters like production. Gainscheduling may then be considered. For this purpose it is useful to have functions like smooth and store regulator parameters, get regulator parameters and test scheduling hypothesis. The last operator scans the parameter values stored in a table for a given process state and determines if the values are reasonably close. These tables themselves are produced during operations where conditions, parameters, and outcomes are stored. The operator that compiles these tables is a learning supervisor.

The operators discussed may be grouped as follows:

Main monitor:
- stability_supervisor
- compute_means_and_variances

Main control:

Back_up control:
- pid_control
- kc_tc_estimator

Fixed gain minimum variance control:
- minimum_variance_control
- minimum_variance_supervisor
  - ringing_detector
  - degree_supervisor

Estimation:
- parameter_estimation
- estimation_supervisor
  - excitation_supervisor
  - perturbation_signal_generator
  - jump_detector

Self-tuning:
- self_tuning_regulation

Learning:
- get_regulator_parameters
- smooth_and_store_regulator_parameters
- test_scheduling_hypothesis

In the terminology of expert systems we thus have six knowledge sources.

The Data Base

The system data base will be required to store current process data, to support technical audit and learning. The entries in the data base must accommodate these demands.

Data base planes. The data base hosts static and dynamic data consisting of facts, evidence, hypotheses, and goals. These data types need not be maintained separately in an implementation. Rather it is usually more appropriate to divide the data base along planes that are the focus of the knowledge sources. One of the most important planes is the event list.

Event lists. One convenient and often used approach for dealing with a time-varying environment in an expert system is to make the processing 'event-driven'. Suitable actions are proposed under the direction of the supervisory control according to the nature of events entered onto a so-called event list. These events may be entered into the processing from an external source, or they may result from internal processing by the knowledge sources on earlier events. Event types include: threshold crossings for process levels and rates, human operator command entries, entry of new hypothesis on process conditions, modification of an earlier hypothesis, request for control mode change, announcement of control mode change, and requests by the human operator for information,

A few examples of event lists will now be given. Some lists are organized to provide data for the different knowledge sources. Data used for the main monitor is shown in Table 1. The major control modes are manual backup, minimum variance and self-tuning. An entry is made in Table 1 when the mode changes or when the set point is changed.

Table 1 - Main monitoring table. An entry is made whenever there is a mode switch or a set-point change.

#	Time	u	σ_u	y	σ_y	Stable	Regulator type

It may be useful to add a few entries in the table such as max and min values or percentile values. The mean values entered in the table for even n are flat means between the events n and n+1. From the data shown in Table 1 it is possible to make deductions like: What are the relations between the mean values of u and y? Do these relations change with time? Are there any relations between the standard deviations and the mean value of the control signal? What are the patterns of the mode switches? Does the system go to tuning mode after large set point changes? What control modes are used for most of the time? Are these drastic variations in performance with time and modes? The answers to these questions will allow us to make inference about the characteristics of the process.

The essential data used in the backup mode is shown in Table 2. An entry in this table is made whenever the system goes into backup mode or when a kc_tc_tuning is made when the system is in back_up mode

Table 2 - Backup Control table. An entry is made whenever the back-up control is activated.

#	Time	k_c	t_c	P	I	D

The reason for entering both the critical point and the PID parameters are that we have introduced options for the system to modify the PID design rules.

The main data generated during periods with fixed gain minimum variance control are given in Table 3. An entry into this table is made whenever the fixed gain minimum variance control is involved.

Table 3 - Minimum variance control table.

#	Time	n_R	n	d	h	Parameters

From Tables 1 and 3 we can ask questions like: What structures of minimum variance regulators are used? Do the structures have any relations to the operating conditions? Are there any patterns in the performances obtained?

In practice the number of event types is large - a system of modest complexity may have in excess of fifty. The event type together with the rules in the supervisory control element for the ordering of their processing combine to produce the flow for an event-driven system. There are also other tables in the system data base like a parameter estimation table and a learning control table.

The hypothesis list. The hypothesis list is another important part of the data base. This is an organized collection of the derived working understandings of the process conditions. The organization often takes the form of levels of abstraction. The lower levels are typically concerned with immediate inferences from sensor data. For example, the hypothesis that deviations are small is easily deduced from a list of the current means and variances. See Table 2. More complicated hypotheses may include making an estimate of the current level of stability of the process. Both numerical evaluations and rule-of-thumb heuristics can be used for this purpose.

As the level of abstraction increases it may be necessary to interact with the process engineer to blend his capability with that of the machine. In order to provide a mixed initiative capability, it is important for the engineer to have access to the rationale upon which system inferences are based. In an expert system this is provided by the simple mechanism of attaching rule numbers to internal events. In this way the data base supports the capability for a technical audit of the processing.

Another important function of the data base is to store processing history in a manner suitable for automatic learning. The idea is similar to learning from experience in chess or checkers. See Samuel (1967) and Michalski et al (1983). In the process control problem there are analogous questions. For example, it is currently difficult to determine analytical approaches for selecting thresholds for control mode switching.

The process supervisor. It is good design practice in building an expert system to separate the knowledge appropriate for control of the decision flow from the detailed knowledge that concerns the application at hand. See e.g. Winston (1977). In many practical cases, however, the organization of the expert system's data base and production rules leaves little of substance for the system supervisor to do. See Nii et al. (1982). In the simple structure outlined here, the functions left over for the process supervisor include logic for establishing event priorities from the control context and logic for deciding processing order among equal priority events. If learning modes are installed in the system, the logic for enabling/disabling their use is another function of the process supervisor.

6. CONCLUSIONS

It is straightforward to extract a more general pattern from the example in Section 5 and from similar problems. To solve a control problem using this approach we first determine a number of design approaches that may be appropriate to the problem. The design methods are analysed carefully to determine the conditions when they will work and when they do not work. Next we attempt to find criteria for those conditions. Finally an expert system is used to decide when and how to take appropriate actions. The approach which clearly can be applied to a wide variety of problems seems to offer interesting possibilities to combine analytical and heuristic approaches.

For simplicity the use of AI techniques in process control has been introduced for the case of single loop control and without regard to the specific details of the potential industrial application. This approach permitted the uncluttered presentation of some of the elemental concepts that arise with the merging of the AI and the control technologies. A heuristic component has been added to the familiar estimation and control algorithms. A key point is that the incorporation of heuristics through AI structures results in systems that are far more flexible and transparent than and selector and safety-jacket hardware logic.

Experience in building expert systems for real applications has shown that the power of the AI approach only comes to the forefront when the problem at hand is sufficiently complex. Plant operators run systems with multiple loops, unpredictable material variations, etc. Over time and with experience the operators generate rules-of-thumb that help them deal with this complexibility. This set of rules is intimately confined to their plant and process. An AI structure that permits planning under uncertainty has been shown to be useful in dealing with this situation. In important ways this planning is an heuristic extension to stochastic dynamic programming (SDP). This observation is sobering for the difficulties attendant even in simple systems with SDP are well known. This paper has pointed out that the use of an expert system can provide a framework for the blending of the numerical algorithms and the detailed expertise of the plant operator. Experimental investigations of systems of this type are currently being performed.

7. REFERENCES

Aström, K.J. (1970): Introduction to Stochastic Control Theory. Academic Press, New York.

Aström, K.J. (1982): A Ziegler-Nichols Auto-tuners Report TFRT-3167. Department of Automatic Control, Lund Institute of Technology.

Aström, K.J. (1983): Theory and applications of adaptive control. Automatica 19, 471-486.

Aström, K.J., P. Hagander and J. Sternby (1983): On the zeros of sampled systems. Automatica, to appear.

Aström, K.J. and A. Helmersson (1982): Dual control of a low order system. CNRS Collque National, Belle-Ile, France.

Aström, K.J. and T. Hägglund (1983): Automatic tuning of simple regulators. Preprints IFAC workshop on Adaptive systems in control and signal Processing, San Francisco, CA.

Aström, K.J. and B. Wittenmark (1973): On self-tuning regulators. Automatica 9, 185-199.

Barr, A. and E.A. Feigenmaum (Eds.) (1982): The Handbook of Artificial Intelligence, Volume 2. William Kaufmann, Inc., Los Altos, California.

Bellman, R. (1957): Dynamic Programming. Princeton, University Press, New Jersey.

Buchanen, B. et al. (1969): Heuristic DENDRAL: A program for generating exploratory hypotheses in organic chemistry. Machine Intelligence, 1, 209-254, Mettzer and Michie, eds., American Elsevier, New York.

Davis, R. (1982): Expert systems: Where are we? And where do we go from here?. The AI Magazine, Spring 1982, 3-22.

Davis, R. and D. Lenat (1982): Knowledge-Based Systems in Artificial Intelligence. McGraw-Hill, New York.

Feigenbaum, E.A. and J. Feldman (Eds) (1963): Computers and Thought. McGraw Hill, New York.

IEEE (1983): Knowledge Representation. Computer 16:10.

Michalski, R., J. Carbonell, and T. Mitchell (1983): Machine Learning, An Artificial Intelligence Approach. Tioga Publishing Co., Palo Alto, California.

Nii, H.P., E.A. Feigenbaum, J.J. Anton, and A.J. Rockmore (1982): Signal-to symbol transformation: HASP/SIAP case study. The AI Magazine, Spring 1982, 23-35.

Nilsson, N. (1980): Principles of Artificial Intelligence. Toga Press Palo Alto, CA.

Sacerdoti, E. (1977): The structure of Plans and Behavior. Elsevier North Holland, New York.

Samuel, A.L. (1967): Some studies in machine learning using the game of checkers II - Recent progress. IBM J. Research and Development 11, 601 - 617.

Stefik, M. (1981a): Planning with constraints (MOLGEN: Part 1). Artificial Intelligence 16, No. 2, 111-139.

Stefik, M. (1981b): Planning and meta-planning (MOLGEN: Part 2), Artificial Intelligence, 16, No. 2, 141-169.

Waterman, D.A., and F. Hayes-Roth (Eds.) (1978): Pattern-Directed Inference Systems. Academic Press, New York.

Winston, P.H. (1977): Artificial Intelligence. Addison-Wesley, Reading, Mass.

Wittenmark, B. (1973): A self-tuning regulator. Report TFRT-3054, Dept. of Automatic Control, Lund Institute of Technology.