7

Expert Control

K. J. Åström and K.-E. Årzén
Department of Automatic Control
Lund Institute of Technology
Box 118, S–221 00 Lund, Sweden

Abstract. *Expert control is a paradigm for controllers with a higher degree of automation than ordinary controllers. Such controllers perform several tasks that are normally done by operators, process engineers, and control engineers. The system is composed of ordinary algorithms which are combined with a knowledge-based system that captures some of the heuristics in design and operational practice. The chapter gives an overview of expert control systems, the ideas they are based on and how they are implemented. Expert control may be viewed as a natural extension of conventional automation systems with controllers and relays for logic and sequencing. An interesting fact is that many less-talked-about features of conventional control systems, as well as some of the unconventional control systems like fuzzy and neural control, fit into the paradigm. It is thus possible to present these systems in a unified framework.*

1. INTRODUCTION

Practically all automation systems are intended to be used by humans. There is a trend to increase the degree of automation of control systems by including more and more of the functions performed by operators, process engineers, and control engineers into the the control systems. Typical examples are systems for autonomous vehicles, systems for industrial automation and process control systems with automatic tuning and adaptation. There are several reasons for the increase in automation degree. It is highly desirable to make systems easier to use. The knowledge about how to design, commission and operate systems is increasing significantly and the computing power required for implementing more automated systems is becoming cost effective. One consequence is that words like intelligent sensors, intelligent actuators and intelligent systems are being used to describe sensors, actuators and controllers with automatic calibration, diagnosis and automatic tuning. The purpose of this chapter is to describe one paradigm that is used to obtain controllers with increased functionality.

The field of automatic control has for a long time focused on algorithms. To obtain flexible systems it is useful to add other elements like logic, sequencing, reasoning and heuristics. Such features are found in many conventional control systems, and to a much higher extent in adaptive control systems. In adaptive control it is attempted to automate modeling and control system design. Modeling includes several features that are difficult to describe by algorithms, like selection of model structure, assessment of experimental conditions and model validation. Control system design also includes steps that are difficult to describe by algorithms, e.g., assessment of achievable performance, selection of appropriate design methods, trade offs between different specifications, etc. Typical examples are systems for autonomous vehicles, systems for industrial automation and process control systems with automatic tuning and adaptation. In implementation of systems it has been the experience that control algorithms are often straightforward to implement but that heuristics is time consuming to implement and validate. Expert control is one possibility to obtain controllers with increased functionality.

Section 2 provides background by giving examples of algorithms and heuristics in typical control system configurations. A knowledge-based system is one way to describe heuristics. Such a description naturally leads to the notion of an expert control system of the type proposed in [1], which is a flexible architecture for combining real time algorithms and logic. Such a system is described in Section 3. This leads to simplification of conventional systems and makes it possible to obtain control systems with new capabilities.

In Section 4 we go a little deeper into some issues that must be considered when attempting to automate design and operation of simple controllers. This provides the background for Section 5, which describes some applications. Section 6 gives examples of implementation of expert control systems.

2. ALGORITHMS AND HEURISTICS

Heuristics plays an important role in conventional control systems. It shows up as logic around linear control algorithms that help them to work over wider operating ranges. Typical examples are anti-windup protection and logic mode switches. Heuristics is also an important part of tuning and commissioning procedures. A recent investigation of industrial control systems has revealed that the development and maintenance of the heuristic part of a system require a large engineering effort. Some examples of heuristics will be given in this section. It will also be indicated that the expert control paradigm is an excellent way to deal with heuristics.

2.1 Simple Controllers

Consider an ordinary PID controller. The small signal behaviour of a system with such a controller can be understood very well from linear analysis. To obtain a good PID regulator it is also necessary to consider operator interfaces, operational issues like switching smoothly between manual and automatic operation, transients due to parameter changes, the effects of non-linear actuators, wind-up of the integral term, maximum and minimum selectors, etc. An operational industrial PID regulator contains heuristic logic that takes care of these issues. Although these heuristic factors are of extreme importance for good controller performance they have not attracted much interest from theoreticians. They are instead hidden in practical designs and rarely discussed in the control theory literature. One reason for this is that the theoretical analysis is quite difficult, another is that many researchers are unaware of these issues. Practically the heuristics shows up as **if-then-else** statements that are intermingled with the ordinary control code.

There are systematic methods to design the linear control algorithms. Similar methods for dealing with the heuristics are presently lacking. A disadvantage with this is that it is often poorly understood and poorly documented.

2.2 PLC and DDC

Industrial automation systems were traditionally composed of two categories of equipment, analog controllers for regulation, and relay systems for interlocks, sequencing and logic. With these systems there was a separation between the control algorithms and the logic. The separation was very strong, since the systems were also handled by different organizations. With the introduction of microprocessors analog controllers were replaced with digital controllers (DDC) and the relays have been replaced with programmable logic controllers (PLC). Since both systems are implemented in the same technology using microprocessors, a natural merging of the techniques of logic, sequencing and algorithms is occurring. DDC systems thus commonly contain some PLC functions and vice versa. This has created many interesting possibilities to make systems with increased capabilities. For example, it is possible to process alarm signals to give

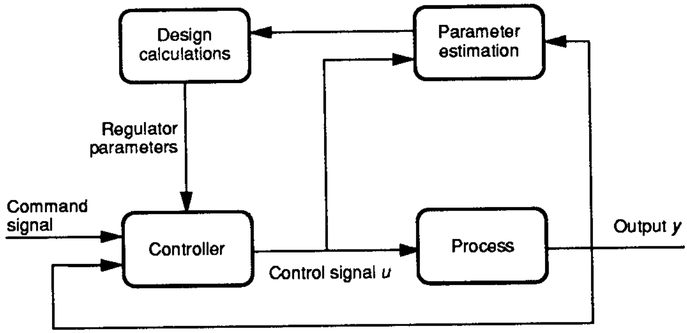

Figure 2.1 Block diagram of an adaptive controller.

more meaningful information. It is also possible to provide the alarm system with capabilities for inquiries.

Logic can be expressed very conveniently in terms of rules. The use of an AI programming style admits system descriptions that are much more compact than those normally used for PLCs. For example, it is possible to have generic rules that apply to all processes of a certain type, allowing significant simplification in programming, modifications, and troubleshooting.

The merger of algorithms and logic is also noticeable for simple controllers. A recent standard proposal for a PID controller has 256 different modes. The reason for the large number of modes is that it is attempted to cover all possible situations. A much smaller number of the modes will be used in each specific application. An alternative implementation would be to incorporate a small knowledge-based system in the controller that admits easy customization.

2.3 Multivariable Controllers

Many multivariable control problems are solved by interconnecting simple single-loop controllers. The problems with windup and mode switching are much more difficult in this case. A systematic approach to anti-windup, mode switches, and reconfiguration in case of faults for true multivariable controllers is still only partially solved.

2.4 Adaptive Controllers

Adaptive systems is another example of a system which contains a mixture of algorithms and logic. An adaptive controller has conventional algorithms for digital control and algorithms for parameter estimation and control design, see Figure 2.1. Since parameter estimation and control design are performed autonomously, it is essential to provide several safeguards. First, it is necessary to make sure that occasional outliers do not give rise to poor estimates. Forgetting of old data is another key issue.

For adaptation it is necessary to discard old data, on the other hand it is important not to discard old data if relevant new information is not received. For example, very little information can be deduced from normal steady state operations when outputs and control signals are constant. It is also necessary to perform various validation procedures to ensure that the models obtained are reasonable before passing them on to control design calculations. Many different ways have been suggested to cope with the problems. Practically all schemes rely on heuristics, which are implemented as supervisors or safety nets for the adaptive systems. These are typically implemented as a collection of **if-then-else** statements that are mixed with the algorithms.

3. THE EXPERT CONTROL PARADIGM

Development of a control system consists of the following activities: modeling, identification, analysis, simulation, control law design, and implementation. It is fair to say that developments over the past 30 years have had a drastic influence on identification, analysis and design. Implementation has also changed mainly, because digital systems are now replacing analogue systems. The vigorous development of concepts and theory are now having an impact on the practice of automatic control. This is accelerated by ideas like expert systems, fuzzy logic and neural networks. In this section we will describe the idea of expert control.

3.1 Basic Ideas

The visionary goal of expert control is a controller

- that can satisfactorily control a large class of processes, which may be time-varying, nonlinear, and exposed to a variety of disturbances;
- which requires minimal prior process knowledge;
- which can make intelligent use of available prior knowledge;
- where the user can enter specifications on the closed-loop performance in qualitative terms, e.g. “as fast as possible”, “small overshoot”, etc.;
- that successively increases its knowledge about the process and improves control performance accordingly;
- that performs diagnosis of the control performance and loop components including detection of actuator and sensor problems;
- with an effective communication scheme where a user can get information about things like process dynamics, statistics on control performance, factors that limit the control performance, explanations for the controllers current actions;
- where the underlying control knowledge and heuristics is stored transparently in such a way that it can easily be examined, modified and extended.

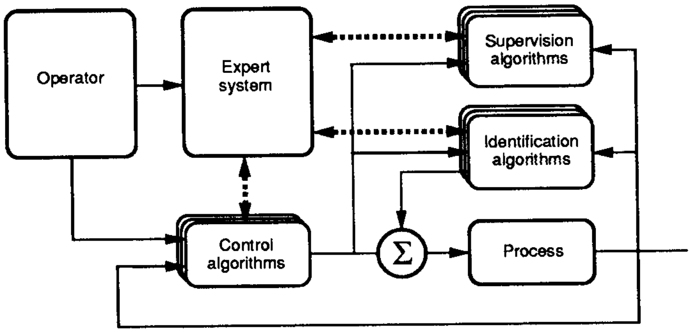

Figure 3.1 Block diagram of an expert controller.

This definition of an expert controller is vague and unprecise. Elements of expert control are, in fact, found in many conventional control systems. There is, however, no existing system that has all features listed above. A key element that is absent from most systems are questions that are related to explicit knowledge representation.

It may also be questioned if it is possible to build a system with all the features listed above. If the class of systems is restricted to single-input single-output processes, which are open-loop stable, the goal is probably not too far away. This will be discussed further in Section 4.

The way to reach the visionary goal can be metaphorically described as an attempt to include an experienced control engineer in the control loop and to provide him with a toolbox consisting of algorithms for control, identification, measurement, monitoring, and control design.

A strong motivation for expert control is to reduce the engineering effort in using feedback control. An expert controller thus supports several of the functions that are traditionally performed by operators, process engineers and control system specialists. The functions are either fully automated or computer supported. An expert controller thus represents a system with a higher degree of automation than an ordinary control system.

A block diagram of an expert controller is shown in Figure 3.1. The system consists of an ordinary feedback loop with a process and a controller. There are, however, many other algorithms in the system apart from the control algorithm. These algorithms perform parameter estimation, control design, supervision, fault detection and diagnosis. There may also be several alternative algorithms for the same task, e.g. several different controllers. This is indicated by the different layers in the figure. For example, the controller may be a simple PI controller or a more complicated algorithm based on an observer and state feedback. There are also algorithms for generating perturbation signals to excite the process. The fault detection and diagnosis tasks are aimed at finding faults that are local to the control loop that the expert controller is part of. This differs from the plant-wide approach to diagnosis taken by the majority of the work in diagnosis, e.g., [2].

The algorithms are coordinated by an expert system, or a knowledge based system, which decides what algorithm to use when. The knowledge-based system also interacts with the operator. The system in Figure 3.1 is very general, it contains the conventional adaptive control system shown in Figure 1 as a special case. An advantage of the system is that it admits a nice separation of algorithms and logic.

3.2 Expert Systems

An expert system consists of an explicit representation of the domain knowledge for a certain application, in this case expert control. The knowledge is usually a combination of pure rules-of-thumb heuristics and knowledge based on a deeper, theoretical understanding of the problem.

The block labeled expert system in Figure 3.1 is composed of a database, a rule base and an inference engine. It communicates with the operator by a user interface. The database could simply be a list of strings with statements like:

s1: Gain margin is 1.2
s2: Variation in pressure of vessel V576 unusually high.

Alternativerly the database could be represented as objects with attributes defined in class definitions. Natural objects in the expert control domain are control loops, numerical algorithms, models derived by the expert system, etc.

The rule base consists of a collection of rules of the type

R1: **If** {premises} **then** {conclusions or actions}

The premises are conditional expressions that operate on the contents of the database. The conclusions add new information to the database. The actions could be commands to the different algorithms, e.g.,

A1: Measure the amplitude margin of loop 52.

A2: Introduce perturbations to obtain better estimates of the transfer function in the range 0.5 to 2 rad/s.

A3: Change control law in loop 15 to PI control.

It is natural to group the rules into classes that are associated with different algorithms and different tasks to be performed. It is very convenient to have generic rules, i.e. rules that apply to classes of objects.

The inference engine is an algorithm that draws conclusions based on the data and the rules. Several strategies can be used for this purpose. The forward chaining strategy is data driven. Starting with premises in the database, it generates conclusions by applying the rules until all possibilities are exhausted. Simultaneously it executes the corresponding actions. This can also generate new conclusions. Backward chaining is another strategy which is hypothesis driven. Starting with a statement like, Reduce variations in the process output of loop 5, the strategy finds rules that has this conclusion. It then chains all rules backwards from conclusions until it find premises that support the desired conclusion or finds a contradiction.

Expert systems usually have an explanation facility that explains how a conclusion was obtained, or the reasoning that supported a hypothesis. The user interface often has nice features like a syntax sensitive editor or a natural language interface.

Expert systems are described in [3], [4] and [5]. They have been applied to a wide variety of problems with varying success. Some commonly given criteria for success are that the problem is nontrivial and sufficiently complex, that the problem can be solved by human experts and that experts are available. The control problems we are considering satisfy all of these criteria.

Expert systems were originally developed to solve static problems, i.e. situations where the premises do not change with time. The control problems we are considering are not static. A statement may, e.g., suddenly switch from true to false because of a change in the physical system being controlled. Reasoning with time is a very complicated problem where many theoretical problems are unresolved [6]. Some pragmatic approaches are taken to deal with these issues. One method is to replace the dynamic problem by a static problem by assuming that all premises hold over a small sliding time-window. Another method is to keep track of the chain of reasoning so that all conclusions drawn from a statement can be withdrawn when the statement ceases to be true. It is also important that conclusions are reached in a reasonable time. Since the time increases rapidly with the number of rules, it is useful to structure the rules into groups. It is also useful to focus the reasoning to a given set of rules.

3.3 Planning

Rules is the standard knowledge representation formalism in expert systems. Rules are also a natural way to to describe much of the logic that is built around conventional control algorithms. However, rules are not very well suited for problems that have a strong sequential element. Although expert control is not dominated by sequential elements, some parts, e.g. control design, are clearly sequential.

The sequential parts of the problem can be represented in different ways. One approach is to combine the rules with a conventional procedural programming language. This solution is adopted in the G2 expert system shell. Another approach is to use sequential function chart formalisms, e.g. Grafcet, to structure the activation and deactivation of groups of rules. Here a rule group can be seen as a knowledge source specialized on one specific subproblem.

Both methods for representation sequences mentioned above have the drawback that the sequential parts are fixed and must be supplied by the developer of the expert controller. Planning is the automatic generation of a sequence of actions that lead to a desired goal. One example is to find a method to bring an oscillating system to a stable operation, another is to move a system from one operating condition to another in a smooth way. Planning has received a lot of attention from AI researchers. See, e.g., [7], and [8]. One possibility is to characterize each action by preconditions and postconditions. The preconditions tell what is required to perform an action and the postconditions describe possible situations after the action. Many of the tasks required in expert control can be described as planning problems.

4. KNOWLEDGE STRUCTURING

Domain knowledge is a key issue in expert control. In this section we will illustrate acquisition of knowledge and reasoning by discussing a single loop controller. Many issues can be illustrated in this way. Notice, however, that there are also important issues, e.g. in diagnosis that require a global view of the system, where the interaction of many loops is considered.

Automation of control system design and operation should consider the tasks of design, commissioning, normal operation and emergencies. Control system design involves issues like control performance, modeling and choice of control laws. Commissioning involves initialization, tuning, trouble shooting and loop auditing. Normal operation involves supervision, diagnosis and fault detection. To perform these tasks we have to represent knowledge about

a) process dynamics
b) actuator saturation
c) disturbances
e) specifications.

There is an interplay between several of these factors. Dynamics is, in principle, no limitation for linear systems that are strictly positive real (SPR) or with first- and second-order dynamics. For such systems the speed of response is limited by measurement noise and actuator saturations. Large pole excess and non-minimum phase dynamics, like time delays and inverse response, impose severe limitations on the achievable performance. It is thus essential to find methods to determine whether the performance is limited by the dynamics or other factors. It is also essential to characterize the complexity of the dynamics, e.g. the presence of oscillatory modes, the order of the dynamics, etc. For systems with difficult dynamics an attempt can be made to change the system so that the dynamics becomes simpler. Time delays can be reduced by repositioning sensors and actuators. Dynamics can be improved by replacing sensors and actuators with devices having faster responses. An attempt to use local feedback to make the dynamics simpler and more reproducible can be made.

The disturbances include set point changes, load disturbances and measurement noise. It is essential to find the ranges and the character of these disturbances. The range of set point changes the required precision in the controlled variable and the maximum loop gain indicate whether proportional control is sufficient or integral action is needed. The magnitude of the error due to load disturbances depends on the amplitude and frequency characteristics of the disturbance and of the loop gain.

Several actions could be contemplated with respect to the disturbances. They can be reduced at the source. Feedforward control can be considered if there is a measurable signal, which is correlated with the disturbance and appropriately located. Filtering can also be used to reduce disturbances and possibly to reconstruct signals that can be modeled.

Measurement noise results in variations in the control signal. Together with actuator saturation this limits the achievable regulator gain and thus also the achievable bandwidth. If an actuator saturates because of measurement noise and high gain, an attempt can be made to reduce the gain, to reduce the disturbance level by filtering or to replace the actuator with a more powerful device.

Model uncertainty is another limiting factor. It can be minimized to some extent by having a high loop gain at those frequencies where the uncertainty is large. To maintain a high loop gain, however, it is necessary to know the phase reasonably well around the cross-over frequency. Uncertainties in the time delay, which give very large phase uncertainties at high frequencies, is a severe limitation on the achievable bandwidth.

Several of the issues discussed above pertain to selection and positioning of sensor and actuators, particularly their sizing and resolution. An important task of an expert control system is also to assess if good design choices have been made. Capabilities to help in auditing control systems can therefore be very valuable. Useful knowledge for this purpose can be derived by observing the operation of a control system. Investigation of static process characteristics gives important information for this purpose. It is also useful to have diagnosis systems that indicate if some component of the control loop is degrading.

4.1 Static Properties

Static input-output characteristics are an important system property, which can be described simply as a function. This function gives the ranges of the input and output signals and indicates the degree of non-linearity. By observing the inputs and outputs of a system during stationary conditions we can also derive useful information about the system.

Preconditions. To determine stationary characteristics it is necessary to first have some criterion to decide that a system is in stationary operation. In typical process control problems this means that we would like to determine cases when there are set-point changes and large process upsets. Since the set point is available, it is easy to find out when it changes. It is also useful to have information about the *time scale* of the process to know how long a set-point upset lasts. Load disturbances are more difficult to determine, but criteria can be based on the magnitude and frequency content of the signals. To obtain good data it is useful to lowpass filter the signals. To do this properly it is necessary to know the time scales of the closed-loop system.

Signal ranges. Observation of the signal ranges and calculation of simple statistics, e.g. mean value, variance, maximum and minimum deviations, will tell if the actuators are properly sized and if sensors and actuators have the proper resolution. If the variations are only a small part of the signal span, it is an indication that a poor selection has been made. It could, for example, be indicated that a system with parallel actuators, one for large deviations and one for fine control, should be used.

The static input-output relation. If a detector for stationarity is available, it is simple to keep a statistic for the fraction of time that the system is stationary. A simple case is, for example, to say that the conditions are stationary if the set-point changes are sufficiently small. The static input-output relation can then be obtained simply by logging the process input and output. To obtain good data the signals should be filtered with respect to the time scale of the closed loop. Curves like the ones shown in Figure 4.1 are then obtained. From these curves it can be determined whether the major variations in the output are due to set-point changes or load disturbances, i.e., whether we are dealing with a servo problem or a regulation problem. We have a servo problem if the experimental data give a well-defined curve and a regulation problem if there is no definite relation between inputs and outputs. A simple statistic of the fraction of the total time when there are set-point changes or transients due to set-point changes is also a useful indicator. Of course, there are also systems which are mixtures of servo and regulation problems. It may be useful to let the operators participate in the assessment. For a regulation problem it may

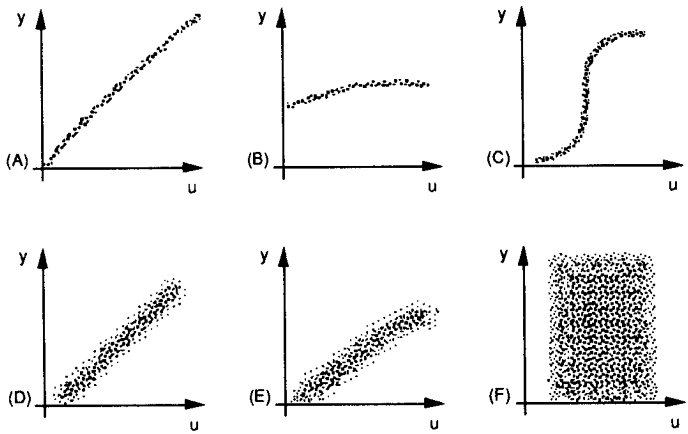

Figure 4.1 Examples of static input-output data logged during normal operation. The results shown in A, B and C indicate a pure servo problem. The results in F indicate a pure regulation problem. Case D and E are mixed cases. Case B indicates poor resolution of the sensor and case F indicates poor actuator sizing.

be useful to request the operator to look for candidates for feedforward signals by looking for signals that are related to the control signal.

For a servo problem the variations in the static gain of a system can also be determined. This gives a valuable indication of whether gain scheduling is required. The static gain curve can also be used for diagnostic purposes. Changes in the curve indicate changes in the process. By comparing the slope of the static gain curve with the incremental process gain measured during tuning or adaptation, we can also get indications whether there is some hysteresis in the loop or not.

To perform the operations it is useful to represent signals in such a way that statistical data over different time ranges are available. This can be done as follows.

Basic signal processing. Let us assume that each signal is associated with four numbers: mean, variance, maximum and minimum. These are called the signal characteristics. Each signal is also associated with a time scale T_s. This can, for example, be the ultimate period of the control loop associated with the signal. The characteristics of each signal are first averaged over T_s. The average is then stored in a ring buffer. Each time the signal has circled the buffer, the mean buffer value is transferred to another ring buffer, etc. The buffers are chosen so that they correspond to intervals such as minute, hour, day, etc. The primary buffer can respond in the primary loop. The others may conveniently be located at higher levels in the system hierarchy. Wavelets are also convenient ways to represent signals.

4.2 Process Dynamics

This section attempts to characterize the process dynamics. We start with crude characteristics and proceed to descriptions that require more details.

Qualitative features. The following are essential system features:

a) stable/unstable
b) monotone/oscillatory
c) essential monotone, minimum phase

These features can be determined from simple experiments on the process. The assessment can be made by a properly trained operator or by a neural network. Some of the features may also be known from design data. Experiments are required to make the assessment or to verify estimates obtained from design data. Two methods, step response and frequency response, are simple to apply and commonly used.

Step response. The step test is a simple experiment that yields useful information about a dynamic system. The test is performed by having the system in equilibrium with a constant input signal. The input signal is then suddenly changed to a new value and the response is recorded. A visual inspection of the step response gives the crude classification discussed above.

A characterization of the step response can be made in terms of a time delay L and the maximum slope, α, of the step response. These parameters are the ones used in Ziegler–Nichols tuning rules.

For processes that are stable with monotone or essentially monotone step responses it is possible to determine three parameters: *process gain* k_p, *apparent dead time* L and *apparent time constant* T. For processes with oscillatory step responses it is possible to determine the period T_p and damping d of the oscillation.

Frequency response. Frequency response is another simple way to characterize the dynamics. It is of particular interest to note that the intersection of the Nyquist curve with the coordinate axes can be determined from simple experiments with relay feedback. A crude classification of the dynamics can also be made from features of the Nyquist curve.

Ultimate gain and ultimate period. The intersection of the frequency response with the negative real axis is of particular interest. It can be described with the parameters k_{180} and ω_{180}. The equivalent parameters $k_u = 1/k_{180}$ and $T_u = 2\pi/\omega_{180}$, called *ultimate gain* and *ultimate period*, are sometimes used for historical reasons. The parameters can be determined approximately by applying relay feedback to the process. The period of the limit cycle obtained is the ultimate period (T_u) and the process gain is approximately given by

$$k_{180} = \pi a_m / 4d$$

where a_m is the amplitude of the limit cycle and d is the relay amplitude.

Knowledge of T_u and k_{180} is sufficient for crude design of a PID regulator. If an additional parameter, e.g. k_p, is known, it is also possible to improve the tuning and to assess the suitable regulator type, see [9].

The characterization of the Nyquist curve can be gradually refined by including more points such as k_{90}, ω_{90}, k_{270} and ω_{270}. The parameters k_{90} and ω_{90} can be determined by relay feedback, where the process is cascaded with an integrator. Since an integrator has a phase lag of 90°, the closed-loop system with ideal relay feedback will oscillate at a frequency close to ω_{90}. The process gain at that frequency is approximately given by

$$k_{90} = \pi a_m \omega_{90} / 4d$$

where d is the relay amplitude and a_m is the amplitude of the relay oscillation. A more accurate estimate is obtained by Fourier analysis.

Mathematical models. A complete mathematical model is a well-known representation of the dynamics. Simple cases that are common in process control are

$$G(s) = k_p \frac{e^{-sL}}{1 + sT} \tag{4.1}$$

and

$$G(s) = k_v \frac{e^{-sL}}{s(1 + sT)} \tag{4.2}$$

More elaborate models are, of course, also possible. When specifying models it is also desirable to give a validity region. When detailed specifications are given, control theory can be used. Models can be determined using system identification methods. Notice that there are simple methods to determine model (4.1) from a relay experiment.

System identification techniques can be used to obtain more complicated models. Notice that there is a significant advantage to have a crude model to plan the experiments and to choose appropriate excitation signals.

Levels of knowledge about the process. When developing a knowledge-based system it is useful to define different levels of process knowledge. The following classification is useful:

Level 0 Qualitative characterization

Level 1 Level 0 and α and L or k_{180} and ω_{180}

Level 2 Level 1 and k_p

Level 3 Level 2 and more points on Nyquist curve, possibly with uncertainty regions

Level 4 Complete mathematical model with uncertainty regions

Level 4A Process with known dynamics that is SPR or of first or second order with known model.

4.3 Disturbances

Disturbances are important aspects of a control problem. In some cases the disturbances are key factors in control system design. The trade-off between rejection of load disturbances and measurement noise is a key question. Unfortunately, there are no simple rules, like the Ziegler–Nichols rules, to find a controller that makes this trade-off. In simple controllers it is often the nature of the disturbances that determines if derivative action should be used.

It is important to know the origin of the disturbances, i.e., whether they are due to measurement noise, load disturbances, set point changes or parameter variations.

Qualitative classification. Disturbances can be classified as transient, stationary or a combination. The transient disturbances are occasional upsets such as steps, pulses, ramps and drift. The stationary disturbances can be periodic, narrow-band or wide-band.

Quantitative description. To describe disturbances quantitatively, it is necessary to give both their amplitude and time characteristics. A simple description of the amplitude distribution can be given in terms of mean, variance, maximum and minimum. A more elaborate description is to give the amplitude distribution.

Time variations can be described in many ways, e.g. as a spectral distribution or in terms of a filter. Crude properties of the filter, e.g. time constants or frequencies, can also be used. To make a useful assessment it is necessary to know the disturbance levels below and above the bandwidth of the system. This means that it is necessary to know the *time scale* for a proper classification. For simplicity we label the high-frequency disturbances as measurement noise and the rest as load disturbances.

If a PID controller has been found it is possible to make a simple assessment of the high-frequency measurement noise simply by measuring the mean square value of the derivative part. Similarly the need for additional filtering can be estimated by also measuring the mean squared value of a low-pass filtered version of the derivative term.

Levels of knowledge about disturbances. Different levels of knowledge of disturbances can be summarized as follows:

Level 0 Qualitative knowledge

Level 1 Level 0 and magnitudes of measurement noise and load disturbances

Level 2 Level 1 and time constants associated with the disturbances

Level 3 Mathematical models of disturbances.

Knowledge of the type discussed in this section is easily encoded by simple rules and algorithms.

5. EXAMPLES

In this section we will give examples of how the expert control paradigm fits different types of feedback systems. The algorithms and the heuristics used will be described briefly.

5.1 Single Loop Controllers

Simple process controllers of the PID type are currently going through an interesting development. Features like automatic tuning, adaptation and gain scheduling are currently being incorporated even in single-loop controllers. To achieve this, it is necessary to automate modeling as well as control design. Modeling has been automated both by conventional system identification methods [10], [11], [12], [13], and with heuristical approaches based on pattern recognition [14], [15], [16], [17] and [18]. A controller that combines an expert system with a neural net is discussed in [19], see also [20]. Control design has also been automated using both algorithmic and heuristic methods. The traditional way of tuning controllers is often based on heuristic rules of the Ziegler–Nichols type [21]. Recently there have been significant efforts to improve and extend these methods. See [16], [9], [22].

A result of this development is that the instrument engineers now have algorithms that will help them tune the controller or will even tune the controller automatically. An interesting side effect is that it has also made gain scheduling easy to use. It is straightforward to generate a gain schedule semi-automatically by using an auto-tuner.

Many of the design choices in existing systems are restricted by the computational power that was economically feasible. With the current rate of increase of computational power and techniques, that have already been proven in laboratory tests, one can extrapolate the characteristics of future controllers. Natural next steps are to include diagnostics and loop auditing [23]. With such features we can talk about autonomous controllers. When such a controller is connected to an unknown process, it will explore the features of the process and the disturbances to decide upon a suitable controller structure and perform the control functions. A system may have the following key components for analysis of the process.

Static Analyzer
Transient Response Analyzer
Relay Feedback Analyzer
Frequency Response Analyzer
Parameter Estimator
Noise Analyzer

The static analyzer gives the relation between the process inputs and outputs in steady state. It tells if the relation is linear or essentially nonlinear. It also gives an indication if the primary function of the feedback loop is regulation or servo following, see [24].

The transient response analyzer investigates the transient responses both passively, when there are natural disturbances, and actively by introducing perturbations. The analyzer determines the period and the damping of the dominant mode, static gain, settling time, rise time, etc.

The relay feedback analyzer determines the frequency ω_{180} where the system has 180° phase lag and the gain k_{180} at that frequency. When applied to an open-loop system this data can be used to determine parameters of a PID controller, see [10]. When applied to a closed-loop system with error feedback, relay feedback can be used to determine the amplitude margin. Simple calculations show that this is given by

$$a_m = \frac{1 + k_{180}}{k_{180}}$$

where k_{180} is the gain obtained from the relay analyzer.

The frequency response analyzer determines the frequency response of a system by active perturbation. This is useful in order to determine dynamics accurately. The parameter estimator determines a parametric model of the system and the noise analyzer characterizes the disturbances acting on the system.

Control design can be based on two subsystems:

- Controller Assessment
- Control Design

The controller assessment determines the structure of a controller that will satisfy the specifications based on data obtained from the analysis and interaction with the operator. Different ways of doing this are discussed in [24]. Several control designs may be used depending on the control algorithm. For PID control there are essentially two types of algorithms that are used. One gives suggestions for modification of the controller parameters based on features obtained from the process analyzers. This is often expressed in terms of rules, see [16] and [15]. The other type of algorithms gives the controller parameters directly as functions of data obtained from the process analyzers, see [10], [22], [9]. In spite of all work that has been done it appears, however, that good simple tuning rules for PID controllers that cover the full range of interesting processes and operating conditions are still missing.

When the system is running it is useful to have various monitoring routines like:

Stability Assessment
Performance Assessment
Actuator Monitor

The stability assessment monitors the stability margins continuously. This can be done passively by tracking the loop transfer function at a few frequencies as is discussed in [11] and [12]. The performance monitor measures means, variances, max and mean of the control signal and the error. Actuators like valves are critical components in many cases. The actuator monitor determines if the actuator deteriorates by determining backlash and hysteresis.

Several of the ideas discussed above have been implemented in existing industrial controllers although in a limited form. Tuners for PID controllers have been implemented as expert systems [15] and [25]. Supervision is a task that is conveniently implemented as an expert system, see [26], [27], [28], [29], [30]. Applications of expert control are found in [31], [32], and [33]. Another interesting application is described in [34] and [35]. In this application an expert system is used to reconfigure and redesign a flight control in case of damages to the airframe. Testing and validation of rule-based systems is a difficult task. An attempt to do this is described in [36].

5.2 Fuzzy Control

Fuzzy control [37], [38], [39], and [40] is conceptually quite different from expert control. In expert control it is attempted to determine and refine as much knowledge as possible about the feedback loop. The goal of fuzzy control is instead to find a framework to deal with imprecision and to design controllers based on inaccurate knowledge of a system. In spite of this there is a strong similarity between the structure of fuzzy and expert controllers. To see this consider the block diagram of a fuzzy controller in Figure 5.1. The sensor signals are converted to linguistic variables characterized by their membership functions. This process called fuzzification, is equivalent to a quantization of the signals. The number of quantization levels are typically small, three to seven. A fuzzy control law may be viewed as a state feedback that gives a representation of the control signal as a fuzzy variable in terms of the state where it is assumed that all state variables are available as linguistic variables. For example a control law where the state variables, in terms of the error (e) and its rate

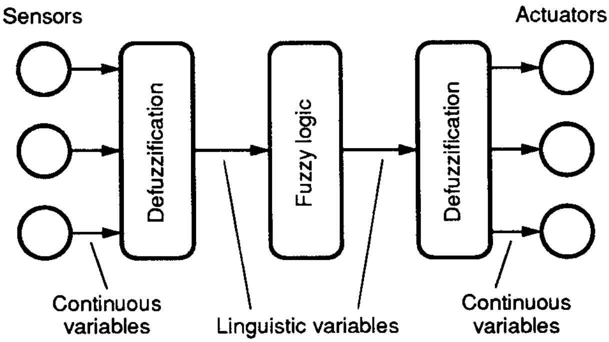

Figure 5.1 A fuzzy controller.

of change (v), are available as linguistic variables N (negative), Z (zero) and P (positive) and the control signal is also quantized in three levels. A fuzzy control law can then be expressed by the three rules

If (e not N and v P) or (e P and v Z) then u N
If (e and v Z) or (e and v of opposite sign) then u Z
If (e not P and v N) or (e N and v Z) then u P

which gives the conditions for the linguistic variable that represents the control signal to be negative, zero or positive. The fuzzy control law gives the control signal as a linguistic variable. The linguistic variable is then mapped into a real number by an operation called "defuzzification". This corresponds to taking the mean or the median of the membership function of the variable.

The fuzzy controller described above may be regarded as a nonlinear PD controller. Sometimes the controller computes the change in the control variable. This means that the controller would be a nonlinear PI controller.

The control law described can be implemented as an expert control system of the type shown in Figure 3.1. Doing so we find that the fuzzy controller has two algorithms, one for fuzzification and one defuzzification. The heuristics is captured by a few rules of the **if-then-else** type. This shows that the system configuration in Figure 3.1 is quite flexible.

6. IMPLEMENTATION

Several experimental expert control systems have been implemented. It was shown in [41] that a blackboard system is a convenient architecture. A detailed description of an implementation is given in [41]. Alternative implementations are also found in [42] and [43]. Comparisons between different tools for implementing such systems are given in [44] and [45].

6.1 Distributed control systems

There is a substantial activity in the process industry aimed at investigating the potential of combining distributed control systems and knowledge-based systems. The majority of the work is focussed on supervisory applications, e.g. alarm analysis, process monitoring, and diagnosis. The systems are typically used as operator assistants. Use of knowledge-based systems for plant wide control is outlined in [46].

A number of commercial real-time knowledge-based system tools have been developed. The most widely used and most sophisticated system is G2 developed by Gensym Corporation [47], [48]. Other examples are RTWorks from Talarian Corporation, RTAC from Mitech, Cogsys from Cogsys Ltd, Muse from Cambridge Consultants, and Chronos from Sagem. Systems of this kind are usually used as an add-on on top of existing distributed control systems. For example, interfaces to G2 have been developed for the major distributed control systems (ABB, Alan Bradley, Fisher Controls International, Honeywell, Taylor, Yokogawa).

A problem with the real-time knowledge-based system above is that they are only loosely interfaced to the distributed control systems. This can potentially create problems with multiple operator consoles, communication bottlenecks, and data duplication. As a response to this several system vendors have developed their own tools. Bailey Controls have developed Expert 90, a small rule-based software module that can be embedded in their distributed control system [49]. Honeywell have developed TDC 3000 Expert, an on-line monitoring system that provides its operator output on the standard operator console. Interesting views on integrating knowledge-based systems into distributed control systems are given in [50], [51], and [52].

It is quite clear that many of the existing real-time knowledge-based system tools have the functionality that is required for expert control applications. It is also clear that, due to their size, it is not realistic to embed them in stand-alone controllers. However, in a distributed control system it is quite feasible to let an integrated real-time knowledge-based system take care of the tuning and monitoring of several control loops, either decoupled or coupled.

Related works. Several works have been reported concerning different aspects of expert control. A complete approach to the whole problem is, however, still missing. Two areas, where most work has been done, deal with automatic tuning of fixed, typically PID, controllers and supervision of adaptive controllers. The work done also differs with respect to the usage of expert system techniques. One approach concentrates on the development of good control heuristics, often expressed in terms of **if-then-else** rules, which then are implemented with standard programming techniques. Another approach also uses expert system techniques in the implementation.

An example of a system focused on control heuristics but implemented with conventional techniques is the Foxboro EXACT [14], [15]. EXACT is a performance adaptive PID controller based on pattern identification of transients in the control error caused by load disturbances or set-point changes. Heuristics and theoretical knowledge is used to adjust the PID parameters to achieve desired damping and overshoot. Similar work is reported in [53]. Heuristic rules for supervision of adaptive controllers are described in [29] and [25]. In the following examples expert system techniques are also used in the implementation. The paper [54] describes an expert system based tuner for PI controllers based on step-response analysis. The process is classified according to its qualitative transient-response characteristics, e.g. monotone, oscillatory, no overshoot, medium delay, etc., for both the open- and closed-loop case. The parameters in a PI controller are then adjusted by heuristic tuning rules for each process class.

Sanoff and Wellstead have combined a rule-based expert system with a self-tuning regulator [55]. The system consists of two parts, one off-line configuration system that determines the parameter settings and one run-time system that monitors the control. In [34], Expert System Adaptive Control (ESAC) is described. The system consists of a self-tuning regulator augmented with three different expert system modules: the system identifier, the control system designer and the control implementation supervisor. A real-time version of the system has not been implemented.

Several off-line consultation tools for controller tuning exist. A system for choosing the parameter settings of an adaptive controller has been implemented by [56] Off-line expert systems for tuning of PID controllers also exist [25] and [57].

The above examples all consider control of general processes. Expert control ideas have also been applied to specific processes. In [58] a rule-based is combined with a neural network for control of a two-link robotic manipulator. Real-time control of a mobile robot is also the topic in [59], who proposes to use a blackboard architecture.

7. CONCLUSIONS

It is straightforward to extract a more general pattern from the examples in Section 5. To solve a control problem a number of design approaches are first determined that may be appropriate for the problem. The design methods are analyzed carefully to determine the conditions under which they perform satisfactorily and those when they do not. Next, criteria are sought delineating these conditions. Finally, an expert system is used to decide when and how to apply the different methods. This approach, which can be applied to a wide variety of problems, seems to offer interesting possibilities for combining analytical and heuristic approaches.

For simplicity, the use of AI techniques has been applied here to single-loop control. This allows an uncluttered presentation of some of the elemental concepts that arise when AI and control technologies are merged. A heuristic component has been added to 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 systems based on selectors and safety jackets currently in use in standard hard-wired logic.

Experience from experiments with systems of this type has shown that the approach is useful in several respects. It is very effective as a test bench for defining the logic required for safe operation of potential control schemes even if this logic is later implemented differently. The experiments point toward the conclusion that powerful control laws can be obtained by combining conventional control algorithms with an expert system. The approach taken in this chapter also emphasizes the need for new theoretical results. Design of a stability margin supervisor is a typical example.

Experience from building expert systems for real applications has shown that their power is most apparent when the problem considered is sufficiently complex. Process control problems are admittedly complex. Plant operators run systems with multiple loops, unpredictable material variations, etc. Over time, and with experience, operators generate rules of thumb that help them deal with this complexity. This chapter has pointed out that an expert system can provide a framework for blending numerical algorithms with this detailed expertise of the plant operator.

Control systems are currently undergoing an interesting phase of development. The driving force is primarily the drastically increased computation ability offered by the microprocessor. This has the potential of making systems more efficient and easy to use. A number of ideas discussed in literature have been reviewed. This includes fuzzy control, neural networks, knowledge-based systems and qualitative reasoning. There are undoubtedly opportunities to make control systems with significantly increased capability. In a simple setting this has been demonstrated by recently announced single-loop controllers, with capabilities for automatic tuning, gain scheduling and adaptation, that are easy to use.

REFERENCES

[1] K. J. Åström, J. J. Anton, and K.-E. Årzén. "Expert control." *Automatica*, **22:3**, pp. 277–286, 1986.

[2] P. R. Prasad and J. F. Davis. "A framework for knowledge-based diagnosis in process operations." In Passino and Antsaklis, Ed, *An Introduction to Intelligent and Autonomous Control.* Kluwer Academic Publishers, 1992.

[3] A. Barr and E. A. Feigenbaum, Eds. *The Handbook of Artificial Intelligence.* William Kaufmann, Los Altos, California, 1982.

[4] F. Hayes-Roth, D. Watermann, and D. Lenat. *Building Expert Systems.* Addison-Wesley, Reading, Massachusetts, 1983.

[5] D. A. Waterman and F. Hayes-Roth, Eds. *Pattern-Directed Inference Systems,* New York, 1983. Academic Press.

[6] T. J. Laffey, P. A. Cox, J. L. Schmidt, S. M. Kao, and J. Y. Read. "Real-time knowledge-based systems." *AI Magazine,* **9:1**, pp. 27–45, 1988.

[7] E. Sacerdoti. *The Structure of Plans and Behavior.* Elsevier, New York, 1977.

[8] K. M. Passino and P. J. Antsaklis. "Modeling and analysis of artificially intelligent planning systems." In Passino and Antsaklis, Ed, *An Introduction to Intelligent and Autonomous Control.* Kluwer Academic Publishers, 1992.

[9] C. C. Hang and K. J. Åström. "Refinements of the Ziegler-Nichols tuning formula for PID auto-tuners." In *Proceedings of the ISA 88 International Conference and Exhibit,* pp. 1021–1030, Houston, Texas, 1988.

[10] K. J. Åström and T. Hägglund. *Automatic Tunings of PID Controllers.* Instrument Society of America, Research Triangle Park, North Carolina, 1988.

[11] T. Hägglund and K. J. Åström. "An industrial adaptive PID controller." In *IFAC Symposium on Adaptive Control and Signal Processing, ACASP '89,* Glasgow, UK, 1989.

[12] T. Hägglund and K. J. Åström. "A frequency domain approach to adaptive control." In *Preprints 11th IFAC World Congress,* volume 4, pp. 265–270, Tallinn, Estonia, 1990.

[13] T. Hägglund and K. J. Åström. "Industrial adaptive controllers based on frequency response techniques." *Automatica,* **27**, pp. 599–609, 1991.

[14] E. H. Bristol. "Pattern recognition: An alternative to parameter identification in adaptive control." *Automatica,* **13**, pp. 197–202, 1977.

[15] T. W. Kraus and T. J. Myron. "Self-tuning PID controller uses pattern recognition approach." *Control Engineering*, June, pp. 106–111, June 1984.

[16] K. L. Anderson, G. L. Blankenship, and L. G. Lebow. "A rule-based PID controller." In *Proc. IEEE Conference on Decision and Control*, Austin, Texas, 1988.

[17] M. Klein. "An elementary pattern recognition self-tuning PI-controller." In *IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore, 1991.

[18] R. W. Swiniarski. "Neuromorphic self-tuning PID controller uses pattern recognition approach." In *IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore, 1991.

[19] C. C. Lee. "A self-learning rule-based controller with approximate reasoning and neural nets." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[20] H. Takatsu. "Intelligent self-tuning PID controller." In *IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91)*, Singapore, 1991.

[21] P. B. Deshpande and R. H. Ash. *Elements of Computer Process Control with Advanced Control Applications.* Instrument Society of America, Research Triangle Park, NC, 1981.

[22] K. J. Åström, C. C. Hang, and P. Persson. "Towards intelligent PID control." In *Preprints IFAC Workshop on Artificial Intelligence in Real Time Control, Shenyang, P. R. China*, pp. 38–43, Oxford, UK, 1989. Pergamon Press.

[23] D. Brandl and S. A. Jeffreys. "The computer assistant for control systems engineering in a historical perspective." In *Proceedings of ISA Annual Conference*, pp. 401–410, Houston, Texas, 1988.

[24] K. J. Åström. "Assessment of achievable performance of simple feedback loops." *International Journal of Adaptive Control and Signal Processing*, **5**, pp. 3–19, 1991.

[25] J. Lieslehto, J. T. Tanttu, and H. N. Koivo. "An expert system for tuning PID controllers." In *Preprints 4th IFAC Symposium on Computer Aided Design in Control Systems, CADCS'88*, Beijing, P. R. China, 1988. Chinese Association of Automation.

[26] P. Caloud. "Towards continuous process supervision." In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pp. 1086–1089, Milan, 1987.

[27] G. Chang and J. D. Birdwell. "An expert system-based supervisor in real-time control." In *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, Hawaii, 1990.

[28] A. Cinar and M. R. Basila, Jr. "Reliable process control with a supervisory expert." In *Proceedings of the 29th IEEE Conference on Decision and Control*, Honolulu, Hawaii, 1990.

[29] R. Sanz, J. Aguilar-Martin, C. Sierra, L. Godo, and A. Ollero. "Adaptive control with a supervisor level using a rule based inference system with approximate reasoning." In *IMACS World Congress*, Paris, France, 1988.

[30] Z. Zhang, M. J. H. Yang, L. Chen, and X. Zhang. "Application of expert control in plasma surface alloying furnace." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[31] S.-L. Jamsa-Jounela and E.-K. Karki. "Expert control systems for rougher flotation of phosphate." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[32] Z. Li. "Expert control strategy used for CFB boiler control system." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[33] N. Tamura, M. K. K. Matsuda, M. Takami, and K. Kadoguchi. "Application of artificial intelligence to operation control of blast furnace." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[34] T. L. Trankle and L. Z. Markosian. "An expert system for control system design." In *Proceedings of IEE Control Conference '85*, pp. 495–499, Cambridge, UK, 1985.

[35] T. L. Trankel, L. Z. Markosian, and P. Sheu. "Expert system architecture for control system design." In *Proceedings of the American Control Conference*, Seattle, Washington, 1986.

[36] J. Tepandi. "Verification, testing, and validation of rule-based expert systems." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[37] E. H. Mamdani. "Application of fuzzy algorithm for control of simple dynamic plant." *Proc. IEE*, **121**, pp. 1585–1588, 1974.

[38] E. H. Mamdani and S. Assilian. "A case study on the application of fuzzy set theory to automatic control." In *Proceedings IFAC Stochastic Control Symposium*, Budapest, Hungary, 1974.

[39] E. H. Mamdani and S. Assilian. "An experiment in linguistic synthesis with a fuzzy logic controller." *Int. J. Man-Machine Stud.*, **7**, pp. 1–13, 1975.

[40] H. R. Berenji. "Fuzzy and neural control." In Passino and Antsaklis, Ed, *An Introduction to Intelligent and Autonomous Control.* Kluwer Academic Publishers, 1992.

[41] K.-E. Årzén. "An architecture for expert system based feedback control." *Automatica,* **25:6**, pp. 813–827, 1989.

[42] D. A. Handelman and R. F. Stengel. "An architecture for real-time rule-based control." In *Proceedings of the American Control Conference,* pp. 1636–1642, Minneapolis, Minnesota, 1987.

[43] S. E. Sallé and K. J. Åström. "Synthesis of a smart PID controller using expert system techniques." In *Proceedings IFAC International Symposium on Intelligent Tuning and Adaptive Control (ITAC 91),* Singapore, 1991.

[44] S. E. Sallé and K.-E. Årzén. "A comparison between three development tools for real-time expert systems: Chronos, G2, and Muse." In *IEEE Control Systems Society Workshop on Computer-Aided Control System Design (CACSD),* Tampa, Florida, 1989.

[45] K.-E. Årzén. "A survey of commercial real-time expert systems environments." In *IFAC/IFIP/IMACS Symposium on Artificial Intelligence in Real-Time Control,* Delft, The Netherlands, 1992.

[46] G. Stephanopoulos, J. E. Johnston, and R. Lakshmanan. "Artificial intelligenve in plantwide control – application to a priori design and on-line implementation." In *Workshop on Artificial Intelligence for Process and Instrument Monitoring and Control, AAAI '86,* Philadelphia, Pennsylvania, 1986.

[47] L. R. Moore, L. B. Hawkinson, M. Levin, A. G. Hoffman, B. L. Matthews, and M. H. David. "Expert system methodology for real-time process control." In *Preprints, IFAC Workshop on AI in Real-Time Control, Swansea, UK,* Oxford, UK, 1987. Pergamon Press.

[48] R. Moore, H. Rosenof, and G. Stanley. "Process control using a real time expert system." In *Preprints 11th IFAC World Congress,* Tallinn, Estonia, 1990.

[49] R. A. Oyen, M. A. Keyes, and M. P. Lukas. "An expert system shell embedded in the control system." In *Preprints, IFAC Workshop on AI in Real-Time Control, Swansea, UK,* Oxford, UK, 1988. Pergamon Press.

[50] K.-E. Årzén. "Knowledge-based control systems: Aspects on the unification of conventional control systems and knowledge-based systems."

In *Proceedings of the 1989 American Control Conference (ACC '89)*, pp. 2233–2238, Pittsburgh, Pennsylvania, 1989.

[51] K.-E. Årzén. "Knowledge-based control systems." In *American Control Conference (ACC '90)*, San Diego, California, 1990.

[52] T. Beck and R. J. Lauber. "Integration of an expert system into a real-time software system." In *Preprints 11th IFAC World Congress*, Tallinn, Estonia, 1990.

[53] J. Marsik and V. Strejc. "Application of identification-free algorithms for adaptive control." In *Proc. 10th World Congress on Automatic Control.* Pergamon Press, Munich, 1987.

[54] B. Porter, A. H. Jones, and C. B. McKeown. "Real-time expert tuners for PI controllers." *IEE Proceedings Part D*, **134:4**, pp. 260–263, 1987.

[55] S. P. Sanoff and P. E. Wellstead. "Expert identification and control." In *Proc. IFAC Identification and System Parameter Estimation*, pp. 1273–1278, York, UK, 1985.

[56] B. A. Foss and O. Øgård. "Choosing the settings of an adaptive controller using an expert system." In *4th IFAC Symposium on Computer Aided Design in Control Systems (CADCS' 88)*, pp. 518–523, Beijing, P. R. China, 1988.

[57] B. Tyreus. "TUNEX – an expert system for controller tuning." Technical report, du Pont, 1987.

[58] D. A. Handelman, S. H. Lane, and J. J. Gelfand. "Integration of knowledge-based system and neural network techniques for robotic control." In *Proc. of the IFAC workshop on Artificial Intelligence in Real-Time Control*, Swansea, Wales, UK, 1988.

[59] G. K. H. Pang. "A blackboard control architecture for real-time control." In *Proc. of the American Control Conference*, pp. 221–226, 1988.