Advanced Control Methods

Survey and Assessment of Possibilities

Åström, Karl Johan

Published in:
Advanced Control in Computer Integrated Manufacturing

1987

Advanced Control Methods
Survey and Assessment of Possibilities

K J Åström
Department of Automatic Control
Lund Institute of Technology

Abstract
This paper gives an overview of different control methods. PID regulators are first discussed including windup and mode switching. It is then shown how complicated control systems can be designed by combining simple regulators with simple nonlinearities to cope with saturations and different operating modes. It is then shown how PID regulators can be tuned automatically. A new technique for automatic tuning of simple control loops is presented. The method has the advantage of automatically generating test signals whose properties are automatically tuned to the particular process. Adaptation and tuning of more general regulator structures are then discussed. Some key problems that arise in applications of adaptive control to industrial processes, prior knowledge, design based on simplified models, initialization and start up ar adressed. A novel approach to the adaptive problem is proposed where a knowledge-based system orchestrates auto-tuning, conventional adaptive control, on-line diagnosis, and table based gain scheduling.

1. Introduction

Most industrial processes are controlled by simple regulators of the PID type. Such regulators contain several important functions in a primitive form like prediction via derivative action. The regulators typically have three parameters which must be selected, the gain k, the integration time T_i and the derivation time T_d. The paper starts with a discussion of PID control. This includes modifications of the basic PID algorithm to obtain systems with good properties this includes modifications of proportional and derivative parts and feedforward to obtain good response to command signals. It is also shown how complicated control system can be designed by combining PID algorithms with simple nonlinear elements like saturations and selectors. More general linear control algorithms like state feedback and Kalman filters are also treated. These regulators can be viewed as natural extensions where the prediction by derivatives are replaced by model based predictions.

There are many things which have to be taken into account when implementing good industrial control systems. Typical examples are how to include manual control, how to avoid integrator windup, how to avoid transients that occur when switching between different modes are questions related to digital implementation. A lot of knowledge about these issues are known to manufacturers of regulators at least with respect to implementation of PID controllers. It is absolutely essential to consider these issues also for more sophisticated control laws. For this reason this paper devotes considerable space to these issues.

Many regulators in industrial use are poorly tuned. The derivative action is seldom used although it may give better performance. One reason for this is that it is not easy to tune a regulator with three parameters. It is even more difficult to tune regulators with more parameters. Methods of automatic tuning of regulators are therefore very important for the effective use of PID control. Such methods are a necessity for more advanced control. There are in principle two different ways to find suitable values of the parameters of a regulator. One possibility is to use heuristic adjustment rules. Another method is to go through the steps of mathematical modeling and application of systematic design methods. The first method is not very reliable. The second approach requires a substantial engineering effort. For this reasons it has been a long standing goal for control engineers to devise schemes to adjust the regulators automatically (auto-tuners) and techniques where the regulator parameters are continuously adjusted (adaptive control). After much research and experimentation adaptive control is now finding its way to practical use in industry. The first commercial products were introduced in 1981. In 1987 there are several commercial systems among them regulators from Leeds and Northrup, Foxboro, Turnbull Controls and four Swedish products made by Kockumation, ASEA, SattControl Instruments (formerly NAF Control) and First Control Systems. Several thousand control loops are controlled successfully by adaptive regulators.

Different methods for automatic tuning and adaptive control are discussed. Most adaptive control schemes currently used can be characterized as local gradient algorithms. This means that given good initial values they will drive the system towards a very good performance. The effort required to obtain the initial values or the prior knowledge may, however, be quite substantial. Several algorithms therefore have a *pretune mode*. There is also a growing awareness of the need for safeguards to ensure that the adaptive regulators work well under all possible operating conditions to obtain the required prior knowledge. The auto-tuner differs from the adaptive regulator because it requires very little prior knowledge. It also generates the test signals automatically.

A very interesting system can be obtained by combining the ideas. An auto-tuner can be used to arrive at a simple control law in a robust way. The information gathered by the auto-tuner can also be used to derive the prior information required by more sophisticated adaptive control systems. A system which contains several different algorithms is then obtained. To monitor their operation it is then useful to introduce algorithms which supervise the operation of the system and which can initiate switching between algorithms. It is clear that a system of this type will involve a substantial amount of heuristic logic. Expert system methodologies provide a systematic approach for dealing with this logic. The term knowledge-based control or expert control is coined to describe systems of this type.

The paper is organized as follows. PID control is discussed in Section 2. Various extensions and practical issues of control design are treated in Section 3. The bottom up approach to building large control systems by combining PID regulators with nonlinear elements is the topic of Section 4. The general linear regulator is discusssed in Section 5. Auto-tuner is described in Section 6. This system, which was originally designed as a tuner for simple PID regulators, can also be used to initialize more sophisticated algorithms. Adaptive control based on recursive parameter estimation is given in Section 7. In Section 8 the ideas of auto-tuning and adaptive control combined with some ideas from the AI field to obtain knowledge-based control system is described. The paper ends with conclusions and suggestions for further reading.

2. The PID Algorithm

The basic PID algorithm can be expressed as follows

$$u(t) = k\left(e(t) + \frac{1}{T_i}\int^{t} e(s)ds + T_d\frac{de}{dt}\right) \tag{2.1}$$

where u is the control variable and e is the control error $e = r - y$ which is the difference between the set point r and the measured value y. The control variable is thus the sum of three terms called proportional, integral and derivative action.

In practical regulators the algorithm (2.1) is often modified.

Proportional Action

It is advantageous to modify the proportional term to

$$P = k(br - y) \tag{2.2}$$

where b is a constant. This modification can be used to reduce the overshoot to step changes in the command signal.

Derivative Action

The derivative action is often modified to

$$D = -k\frac{pT_d}{1 + pT_d/N}y \tag{2.3}$$

where $p = d/dt$ is the differential operator. This means that the derivation action operates on the output y and not on the command signal. The other modification is that the derivative action only operates on low frequency components. At high frequencies the derivative gain is limited to kN.

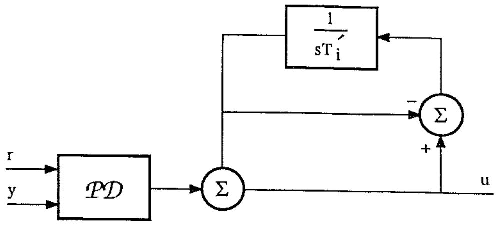

Figure 1. Block diagram of a PID regulator on series form.

Series Form

The algorithm (2.1) is called the parallel form because it can be viewed as a parallel connection of proportional, integral and derivative action. An alternative form is

$$G(s) = k'(1 + \frac{1}{sT_i'})(1 + sT_d') \tag{2.4}$$

This is called the series form because it can be interpreted as a series connection of a PD part with a PI part. The regulator (2.4) can always be represented in the form (2.1) with the coefficients

$$\begin{aligned} k &= k'\frac{T_i' + T_d'}{T_i'} \\ T_i &= T_i' + T_d' \\ T_d &= \frac{T_i'T_d'}{T_i' + T_d'} \end{aligned} \tag{2.5}$$

The parallel form (2.1) can be transformed to the series form only if

$$T_i \geq 4T_d$$

The parameters are then given by

$$\begin{aligned} k' &= \frac{k}{2}\left(1 + \sqrt{1 - \frac{4T_d}{T_i}}\right) \\ T_i' &= \frac{T_i}{2}\left(1 + \sqrt{1 - \frac{4T_d}{T_i}}\right) \\ T_d' &= \frac{T_i}{2}\left(1 - \sqrt{1 - \frac{4T_d}{T_i}}\right) \end{aligned} \tag{2.6}$$

The parallel form is thus more general than the series form. There is, however, a very simple implementation of the series form which is used in many systems. This implementation is shown in Figure 1. The advantages are that it is easy to avoid windup and bumps at mode changes.

Discretization

To implement a continuous time control law like a PID regulator on a digital computer it is necessary to approximate the derivative and the integral which appear in the control law. A few different ways to do this will now be discussed.

Proportional Action

The proportional term is

$$P = k(br - y)$$

This term is implemented simply by replacing the continuous variables by their sampled versions. Hence

$$P(t_k) = k(t_k)(br(t_k) - y(t_k)) \tag{2.7}$$

where $\{t_k\}$ denote the sampling instants, i.e. the times when the computer reads the analog input.

Integral Action

The integral term is given by

$$I(t) = \frac{k}{T_i} \int^t e(s)ds$$

It thus follows that

$$\frac{dI}{dt} = \frac{k}{T_i} e$$

Approximating the derivative by a difference we get

$$\frac{I(t_{k+1}) - I(t_k)}{h} = \frac{k}{T_i} e(t_k)$$

where the sampling period $h = t_{k+1} - t_k$ is assumed constant. This leads to the following recursive equation for the integral term

$$I(t_{k+1}) = I(t_k) + \frac{kh}{T_i} e(t_k) \tag{2.8}$$

For the implementation it is crucial that the wordlength used is sufficiently large so that the term $khe(t_k)/T_i$ is not rounded off when added to $I(t_k)$. This condition is most critical when h is small and T_i is large.

Derivative Action

The derivative term is given by (2.3) i.e.

$$\frac{T_d}{N} \frac{dD}{dt} + D = -kT_d \frac{dy}{dt} \tag{2.9}$$

There are several ways of approximating the derivative.

Forward Differences

Approximating the derivative by a forward difference gives the equation

$$\frac{T_d}{N} \frac{D(t_{k+1}) - D(t_k)}{h} + D(t_k) = -kT_d \frac{y(t_{k+1}) - y(t_k)}{h}$$

This can be rewritten as

$$D(t_{k+1}) = \left(1 - \frac{hN}{T_d}\right) D(t_k) - kN\left(y(t_{k+1}) - y(t_k)\right) \tag{2.10}$$

The approximation is stable only if $T_d > 2Nh$.

Backward Differences

If the derivatives in (2.9) are approximated by backward differences we get

$$\frac{T_d}{N} \frac{D(t_k) - D(t_{k-1})}{h} + D(t_k) = -kT_d \frac{y(t_k) - y(t_{k-1})}{h}$$

This can be rewritten as

$$D(t_k) = \frac{T_d}{T_d + Nh} D(t_{k-1}) - \frac{kT_dN}{T_d + Nh} \left(y(t_k) - y(t_{k-1})\right) \tag{2.11}$$

The approximation is stable for all positive T_d.

Tustin's Approximation

There is yet another approximation proposed by Tustin which is commonly used. This approximation is

$$D(t_k) = \frac{2T_d - hN}{2T_d + hN} D(t_{k-1}) - \frac{2kNT_d}{2T_d + hN} \left(y(t_k) - y(t_{k-1})\right) \tag{2.12}$$

Notice that all approximations have the same form i.e.

$$D(t_k) = a_i D(t_{k-1}) - b_i \left(y(t_k) - y(t_{k-1})\right) \tag{2.13}$$

but with different values of the parameters a_i and b_i. The approximation (2.12) is stable if $T_d > 0$. The value of a_i is, however, negative if $T_d < Nh/2$. This is undesirable because the approximation will then exhibit ringing. This effect can be very significant if $T_d \ll Nh$. Hence only the approximation (2.11) gives good results for all values of T_d.

Incremental Form

The algorithms described so far are called positional algorithms because they give the output of the regulator directly. In digital implementations an incremental form of the algorithms is also used. This form is obtained by computing the time differences of the regulator output and adding the increments. In a continuous time version the time derivative of the output is computed and the derivative is then integrated. This form is particularly useful when the actuator is a stepping motor because the motor can then be used as the summing device.

One advantage with the incremental algorithm is that most of the computations are done using increments only. Short wordlength calculations can often be used. It is only in the final stage where the increments are added that precision is needed. Another advantage with the incremental algorithm is that the regulator output is driven directly from an integrator. This makes it very easy to deal with windup and manual control. A problem with the incremental algorithm is that it can not be used directly for regulators with P or PD action only. Such a regulator cannot keep a proper steady state because the output of the regulator depends on the initial state of the regulator. The problem can be avoided with a regulator which contains a feedback that resets the integrator to a proper value.

3. Windup

Although many aspects of a control system can be understood based on linear theory there are some nonlinear effects that must be accounted for. All actuators have limitations, a motor has limited speed, a valve cannot be more than fully open or fully closed etc. When a control system operates over a wide range of operating conditions it may happen that the control variable reaches the actuator limits. When this happens the feedback loop is effectively broken because the actuator may remain at its limit independently of the process output. If a regulator with integrating action is used, the error may continue to be integrated. This means that the integral term may become very large or colloquially that it "winds up". The consequence is that any regulator with integral action may give large transients when the actuator saturates.

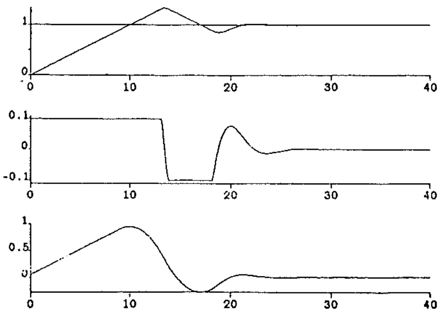

Figure 2. Illustration of integral windup.

An Example

The windup phenomena is illustrated in Figure 2 which shows control of a process with a PI regulator. The initial set-point change is so large that the actuator saturates at the high limit. The integrator increases initially because the error is positive and it reaches its largest value at time $t = 10$ when the error goes through zero. The output remains saturated at this point because of the large value of the integral. It does not leave the saturation limit until the error has been negative for sufficiently long time to let the integral part come down to a small level. The net effect is a large overshoot which is clearly noticeable in the figure.

How to Avoid Windup

There are several ways to avoid integral windup. A convenient way is shown in Figure 3. An extra feedback path is provided in the regulator by measuring the actual actuator output and forming an error signal e_s as the difference between the output of the regulator v and the actuator output u. The signal e_s is fed to the input of the integrator through a gain $1/T_r$. The signal e_s is zero when there is no saturation. It will thus not have any effect on the normal operation when the actuator does not saturate. When the actuator saturates the feedback signal will, however, drive the error e_s to zero. This means that it drives the integrator to a value such that the regulator output is exactly at the saturation limit. This will clearly prevent the integrator from winding up. The rate at which the regulator output is reset is governed by the feedback gain $1/T_r$, where T_r can be interpreted as the time constant which determines how quickly the integral is reset.

It frequently happens that the actuator output cannot be measured. The anti-windup scheme just described can be applied by incorporating a mathematical model of the saturating actuator as is illustrated in Figure 3.

Figure 4 shows what happens when a regulator with anti-windup is applied to the system simulated in Figure 2.

Notice that the output of the integrator is quickly reset to a value such that the regulator output is at the saturation limit and that the integral has a negative value during the initial phase when the actuator is saturated. This behavior is drastically different from that in Figure 2 where the integral has a positive value during the initial transient. Also notice the drastic improvement in performance compared to the ordinary PI regulator used in Figure 2.

The effect of different values of the time constant T_r is illustrated in Figure 5. It may thus seem advantageous to always choose a very small value of the time constant T_r because the integrator is then reset quickly. Some care must, however, be exercised when introducing anti-windup in systems with derivative action. If the time constant T_r is chosen too small it may happen that spurious errors cause saturation of the output due to a large derivative term. This may accidentally reset the integrator. A practical rule is to make T_r proportional to the integration time T_i.

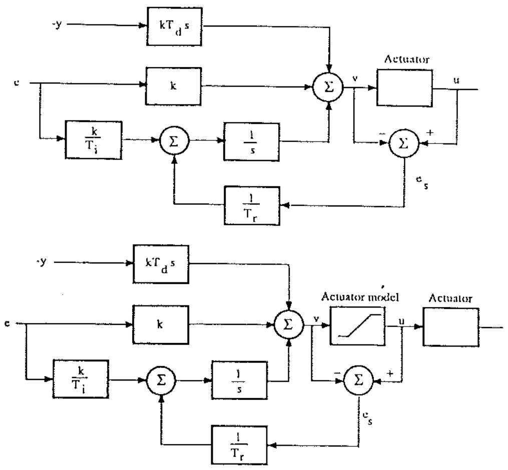

Figure 3. Regulator with anti-windup. A system where the actuator output is measured is shown in A and a system where the actuator output is estimated from a mathematical model is shown in B.

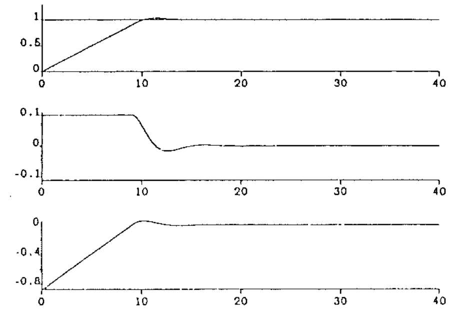

Figure 4. Regulator with anti-windup applied to the system in Figure 2.

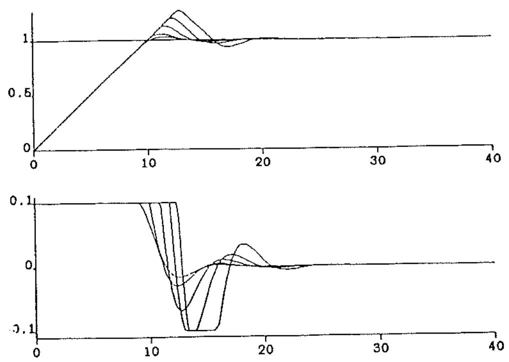

Figure 5. The step response of the system in Figure 4 for different values of the reset time constant T_r.

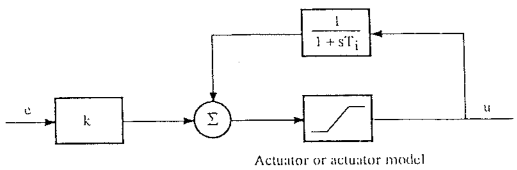

Figure 6. How to provide anti-windup in a regulator where integral action is generated as automatic reset. Compare with Figure 2.

Series Implementation

A similar device for avoiding windup can be applied to the regulator in Figure 1 by incorporating a model of the saturation as is shown in Figure 6. Notice that in this implementation the reset time constant T_r is the same as the integration time T_i.

A Regulator Module

The systems shown in Figure 3 can be conveniently represented if we introduce the module shown in Figure 7. This module has three inputs, the set point, the measured output and a tracking signal. The new input TR is called a tracking signal because it follows from Figure 7 that the regulator output v tries to track this signal. Using such a model the systems shown in Figure 3 can be represented as shown in Figure 8. The parameters are the PID parameter (k, T_i, T_d, b and N) and the reset time constant T_r.

Systems with Selectors

A selector is a device with several inputs and one output. The output is at each time the smallest of the inputs for a minimum selector or the largest input for a maximum selector. Selectors are used to make sure that constraints are satisfied.

When selectors are used to choose among the outputs of several regulators with integral action it is crucial that anti-windup is considered. This is easily handled using the regulator module with a tracking input. Figure 9 shows how the regulators can be connected. When $v_1 < v_2$ the output is $u = v_1$. The output u is thus controlled by regulator R_1. The regulator R_2 will track u since $v_2 \neq u$. A simulation of such a scheme is shown in Figure 10.

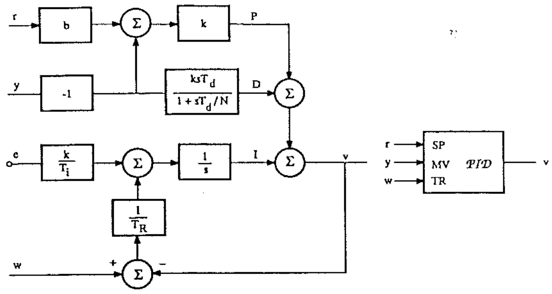

Figure 7. Block diagram and simplified representation of PID regulator with tracking signal.

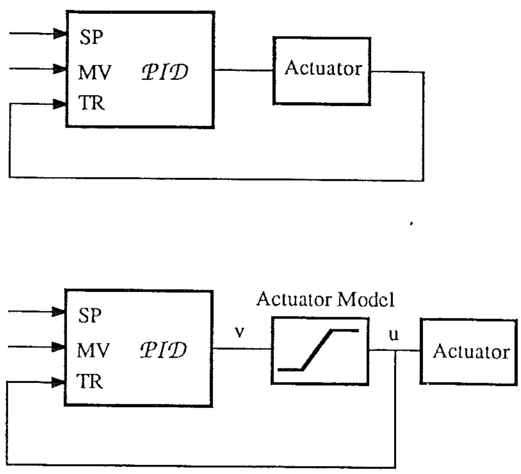

Figure 8. Representation of a regulator with anti-windup using the basic control module with tracking shown in Figure 7.

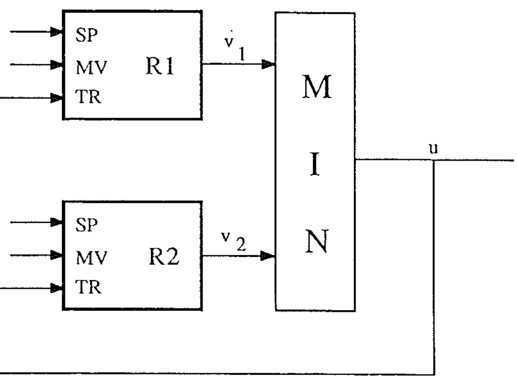

Figure 9. How to avoid windup in circuits with selectors.

Cascade Control

Avoiding windup in cascade control poses special problems. For the secondary regulator windup can be handled in the usual manner. To avoid windup in the primary regulator it is, however, necessary to know that the secondary regulator saturates. One strategy is to put the primary regulator into manual control when

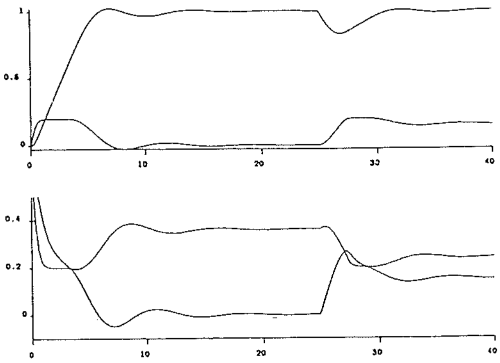

Figure 10. Simulation of a system with selectors.

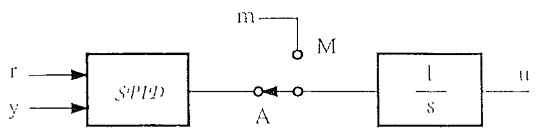

Figure 11. How to introduce manual control in a regulator with incremental output.

the secondary regulator saturates.

4. A Toolbox for Control

Many complex control systems are constructed in a bottom up approach by building a large control system from simple elements. This is actually the way many instrumentation systems have developed. In this section we will describe different modules and show how they can be combined with the PID algorithm.

Manual Control

Most control systems need a facility for manual control. To achieve this it is necessary to have a convenient way to switch off the automatic control action and to change the control variable of the process directly. Manual control is often done using two buttons. The control variable increases when pushing one, it decreases when the other one is pushed. The control variable remains constant if neither button is pushed. A facility of this type is provided even in the most simple regulators. There is typically a mode switch for manual and automatic and increase/decrease buttons. It is of course also necessary to have a smooth transfer between the manual mode and the automatic mode. Since the command buttons only give the changes in the control variables it is necessary to have an internal state which represents the sum of the changes. To ensure a smooth transfer between the manual and automatic modes it is necessary to ensure that the state associated with manual control is updated properly when the regulator is in automatic mode and vice versa.

Incremental Algorithms

A bumpless switch between automatic and manual is particularly easy to do in incremental algorithms when the control variable is driven directly by an integrator. The integrator is provided with a switch so that either the increments from the manual control input or the increments from the PID algorithms are sent to the integrator. See Figure 11.

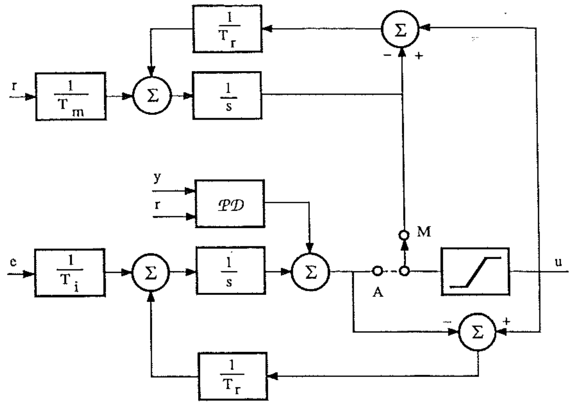

Figure 12. PID regulator with parallel implementation which switches smoothly between manual and automatic control.

Absolute Algorithm with Series Implementation

A similar mechanism can be used in the series implementation of a PID controller shown in Figure 1. In this case there will be a switching transient if the output of the PD part is not zero at the switching instant. Notice that it is necessary to have two switches.

Parallel Implementation

For regulators with parallel implementation the integrator of the PID regulator can be used to add up the changes in manual mode. Such a system gives a smooth transition between manual and automatic mode provided that the switch is made when output of the PD block is zero. If this is not the case there will be a switching transient. This will almost always be the case when the PD action is given by

$$P + D = k\left(br - y - T_d \frac{dy}{dt}\right)$$

with $b \neq 1$.

It is also possible to use a separate integrator to add the incremental changes from the manual control device. To avoid switching transients in such a system it is necessary to make sure that the integrator in the PID regulator is reset to a proper value when the regulator is in manual mode. Similarly the integrator associated with manual control must be reset to a proper value when the regulator is in automatic mode. This can be realized with the circuit shown in Figure 12. With this system the switch between manual and automatic is smooth even if the control error or its derivative is different from zero at the switching instant. When the regulator operates in manual mode as is shown in Figure 12 the feedback from the output v of the PID regulator tracks the output u. With efficient tracking the signal v will thus be close to u at all times. There is a similar tracking mechanism which ensures that the integrator in the manual control circuit tracks the regulator output in manual modules.

To build large automation systems it is useful to have suitable modules. Figure 13 shows the block diagram for the manual control module. It has two inputs, a tracking input and an input for the manual control commands. The system has two parameters, the time constant T_m for the manual control input and the

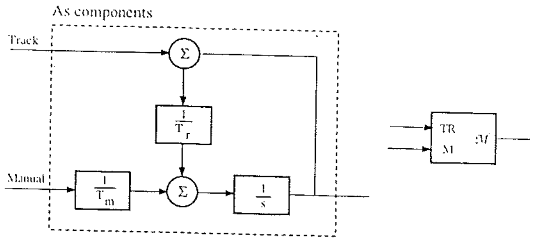

Figure 13. Manual control module.

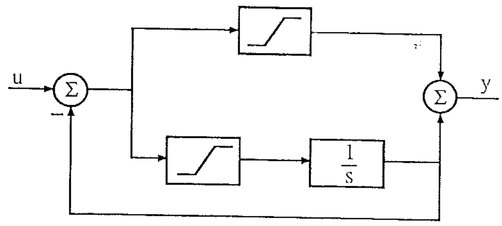

Figure 15. Block diagram of a jump- and rate circuit.

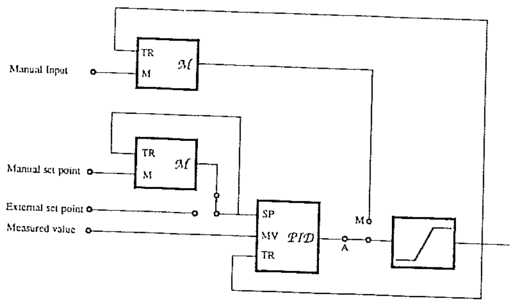

Figure 14. A reasonably complete PID regulator with anti-windup, automatic-manual mode, and manual and external setpoint.

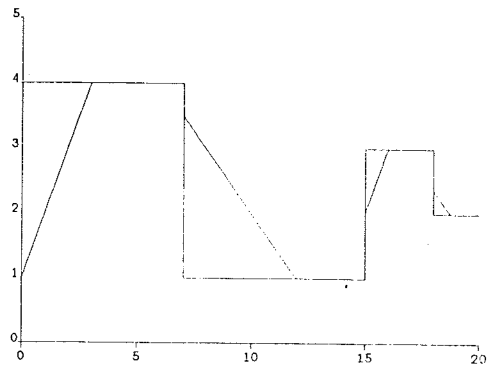

Figure 16. Input and outputs for the jump and rate.

reset time constant T_r. In digital implementations it is convenient to add a feature so that the command signal accelerates as long as one of the buttons increase-decrease buttons are pushed.

Using the module for PID control, introduced in Figure 8, and the manual control module in Figure 13 it is straightforward to construct a complete regulator. Figure 14 shows a PID regulator with internal or external setpoints via increase/decrease buttons and manual automatic mode. Notice that the system only has two switches.

Limiters

We have thus arrived at two modules which are useful for building control systems, a manual control module and a PID module. Both modules have internal states and a tracking input. To design complete systems it is also useful to add modules for selection of maximum and minimum. To model actuators we also need a module for saturating a signal. It is sometimes needed to make sure that command signals do not change too rapidly. This can be ensured by the jump- and rate module shown in Figure 15. The module has one input and one output and four parameters. The properties of the jump and rate circuit are illustrated in Figure 16.

Summarizing we have the following modules.

PID
Manual
Saturation
Min selector
Max selector
Jump and rate

Using these modules we can now construct large control systems that can cope with good set point control, saturations, good responses to large commands.

5. General Digital Control Laws

The derivative action in a PID regulator is a simple way to predict future values of the controlled variable simply by tangential extrapolation of the output T_d time units ahead. This prediction works well in some cases but poorly in others, e.g. when the system has time delays. A much more effective way of making predictions is to use a mathematical model of the system dynamics. Such an approach leads to a control law of the form

$$
\begin{aligned}
u(k) &= L[x_m(k) - \hat{x}(k)] + D_c r(k) \\
\hat{x}(k+1|k) &= A\hat{x}(k) + Bu(k) \\
\hat{x}(k) &= \hat{x}(k|k-1) + K\left[y(k) - C\hat{x}(k|k-1)\right]
\end{aligned}
$$

The regulator can be thought of as comp]osed of a state estimator and a feedback from the estimated state. There are many theories that lead to a regulator of this type, e.g. Smith predictors, state feedback via pole placement or optimal control, Kalman filtering or observers, linear quadratic optimal control and LQG/LTR.

The control law above can also be written in the following form

$$
\begin{aligned}
u(k) &= Cx(k) + Dy(k) + D_c r(k) \\
x(k+1) &= Fx(k) + Gy(k) + G_c r(k)
\end{aligned}
$$

where r is the command signal, y the measured signal and u the control variable.

In an analog realization all operations are executed in parallel. When the control algorithms are implemented on a digital computer the parallel operations have to be realized sequentially. To do this there are two problems that must be taken into account, namely simultaneity and time delays.

Computational Delay

The control program can then be written as

```
1 Adin y r
2 u:=C*x+D*y+Dc*r
3 x:=F*x+G*y+Gc*r
4 Daout u
```

Listing 1. Regulator code.

It is desirable to make the computational delay as small as possible. Notice that the DA conversion can be made after the second statement since the control signal u is then available. Also notice that the product $C*x$ can be precomputed. The code then becomes

```
1 Adin y r
2 u:=u1+D*y+Dc*r
3 ShapeOutput
4 Daout u
5 x:=F*x+G*y+Gc*r
6 u1:=C*x
```

Listing 2. Improved regulator code.

Notice that an extra state variable u_i has been introduced to save computing $C*x$ in the first statement. Also notice that a procedure ShapeOutput which saturates the output, make a min selector etc. is added.

The code shown above can be generalized as follows

```
1 Adin y r
2 ComputeOutput
3 ShapeOutput
4 Daout
5 UpdateState
```

Listing 3. Generalized regulator code.

It would be appealing to make a procedure for each one of the boxes PID, Manual etc. in the Figures 7, 13 and 14. This cannot be done because of the sequential character of the calculations.

In the analog implementation the signals v and u will change simultaneously. This is essential for the antiwindup signal to function properly. In a digital implementation there will always be a delay between u and v. If the PID regulator and the actuator are implemented as separate blocks there is no way to avoid this delay. The tracking signal u will then differ from v and the antiwindup coupling will give an undesired contribution. Although this contribution will be small if the delay is small it is always present. This undesirable effect can be avoided if the code is restructured.

There is also another problem if the blocks in Figure 7 are represented as separate subroutines. If the antiwindup scheme should work properly it is essential that simultaneous values of the tracking signal t_r and the regulator output v are used. This will not be the case if the regulator code is executed first and the actuator model afterwords.

It is thus essential that the computational scheme shown in Listing 3 is used in digital implementations. To obtain the appropriate structure the algorithms for the discrete time PID regulator will be rewritten appropriately.

The regulator output is given by

$$u(t_k) = P(t_k) + I(t_k) + D(t_k) \tag{5.1}$$

where P, I and D are given by the equations (2.7), (2.8) and (2.9) respectively. It follows from (2.7) that the proportional part cannot be precomputed. Equation (2.8) shows that the integral term can be precomputed. For this purpose we introduce I as a state variable. It follows from equation (2.13) that part of the derivative term can be precomputed. A state variable x is introduced to account for those terms. This state variable is defined as

$$x(t_k) = a_i D(t_{k-1}) - b_i y(t_{k-1}) \tag{5.2}$$

The derivative term then becomes

$$D(t_k) = x(t_k) - b_i y(t_k) \tag{5.3}$$

It follows from (5.2) and (5.3) that the state variable x is updated as follows

$$\begin{aligned} x(t_{k+1}) &= a_i\left[x(t_k) - b_i y(t_k)\right] - b_i y(t_k) \\ &= a_i x(t_k) - b_i(1+a_i)y(t_k) \end{aligned} \tag{5.4}$$

Equation (5.1) can now be written as

$$\begin{aligned} u(t_k) &= bkr(t_k) - [k + b_i]\,y(t_k) + I(t_k) + x(t_k) \\ &= \ell_0 r(t_k) - \ell y(t_k) + u_1(t_k) \end{aligned} \tag{5.5}$$

and the equations for updating the states becomes

$$\begin{aligned} I(t_{k+1}) &= I(t_k) + \frac{kh}{T_i}\left[r(t_k) - y(t_k)\right] \\ x(t_{k+1}) &= a_i x(t_k) - b_i(1+a_i)y(t_k) \\ u_1(t_{k+1}) &= I(t_{k+1}) + x(t_{k+1}) \end{aligned} \tag{5.6}$$

The procedure ComputeOutput in Listing 3 is then an implementation of (5.4) and the procedure UpdateState is a procedure which performs the calculations given by (5.6).

6. Automatic Tuning

A novel approach to automatic tuning of PID regulators has been proposed by Åström and Hägglund (1984). It was motivated by the need for a simple robust tuning scheme which requires very little prior information. The method is based on a special technique for system identification which automatically generates an appropriate test signal and a variation of a classical method for adjusting the parameters of a PID regulator.

The Basic Idea

The Ziegler-Nichols method for tuning PID regulators is based on the observation that the regulator parameters can be determined from knowledge of the point where the Nyquist curve of the open loop system intersects the negative real axis. It is traditionally described in terms of the ultimate gain k_c and the ultimate period T_c. In the original scheme, described in Ziegler and Nichols (1943), the ultimate gain and the ultimate period are determined in the following way: A proportional regulator is connected to the system. The gain is gradually increased until an oscillation is obtained. The gain k_c when this occurs is the ultimate gain and the oscillation has the ultimate period. It is difficult to perform this experiment automatically in such a way that the amplitude of the oscillation is kept under control.

The auto-tuner is based on the idea that the ultimate gain and the ultimate period can be determined by introducing relay feedback. A periodic oscillation is then obtained. The critical period T_c is simply the period of the oscillation and the critical gain can be determined from the relay amplitude and the amplitude of the oscillation, see Figure 17. If the process attenuates high frequencies so that the first harmonic component dominates the response it follows that the input and the output are out of phase. Furthermore if the relay amplitude is d it follows from a Fourier series expansion that the first harmonic of the input is $4d/\pi$. If the amplitude of the output is a the process gain is $\pi a/4d$ at the critical frequency and the critical gain becomes

$$k_c = \frac{4d}{\pi a} \tag{6.1}$$

Exact analyses of relay oscillations are also available, see Åström and Hägglund (1984). The period of an oscillation can be determined by measuring the times between zero-crossings. The amplitude may be determined from the peak-to-peak values of the output. These estimation methods are easy to implement because they are based on counting and comparison only. The sensitivity to disturbances can be reduced significantly by filtering the signals adaptively and introducing hysteresis in the relay.

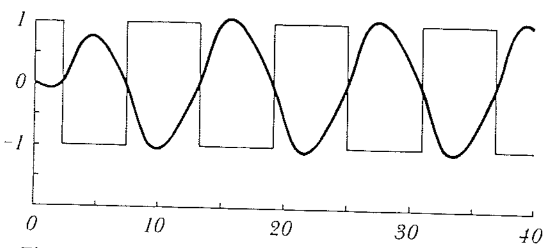

Figure 17. Input and output signals for a system under relay feedback. The linear system has the transfer function $G(s) = 0.51(1-s)/s(s+1)^2$.

Control Design

When the ultimate gain k_c and the ultimate period are known the parameters of a PID regulator can be determined by the Ziegler-Nichols rule which can be expressed as

$$k = \frac{k_c}{2} \qquad T_i = \frac{T_c}{2} \qquad T_d = \frac{T_c}{8} \tag{6.2}$$

This rule gives a closed loop system which is sometimes too poorly damped. There are therefore many modifications of the Ziegler-Nichols rule which give improved performance. A block diagram of the auto-tuner is shown in Figure 18. The tuner is very easy to use. The process is simply brought to an equilibrium by setting a constant control signal in manual mode. The tuning is then activated by pushing the tuning switch. The regulator is automatically switched to automatic mode when the tuning is complete.

Prior Information

A major advantage of the auto-tuner is that it requires little prior information. Only two parameters - the relay amplitude and the hysteresis width of the relay - are required. These parameters are set automatically in the NAF auto-tuner. The relay amplitude is initially set to fixed proportion of the output range. The amplitude is adjusted after one half period to give an output oscillation of specified amplitude. The modified relay amplitude is stored for the next tuning. The hysteresis width is set automatically based on measurements of the measurement noise.

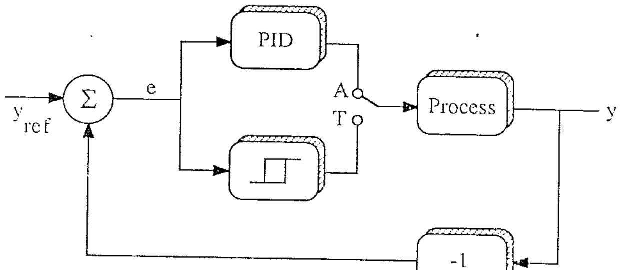

Figure 18. Block diagram of an auto-tuner. The system operates as a relay controller in the tuning mode (T) and as an ordinary PID regulator in the automatic control mode (A).

Practical Experiences

The auto-tuner looks like a conventional standard regulator. The mode switch, which has two positions - manual and automatic control - on a conventional regulator, has a third position called tune. When the regulator is set in this mode a tuning is performed automatically in closed loop.

The experiences of using the auto-tuner have been very good. Operators have found it very easy to use. They are in command since the tuning is made on their request. As a result they also pay more attention to tuning. It has been demonstrated that the commissioning time considerably of new systems can be reduced significantly by using the tuning tool. It has also been found that many loops are poorly tuned.

An Example

The properties of the auto-tuner are illustrated in Figure 19, which shows an application to temperature control in a distillation column. A PI regulator was used originally. The plant personnel had great difficulties to get the control loop to function well using conventional tuning rules. The loop was oscillating when the experiment started as is shown in Figure 19. The

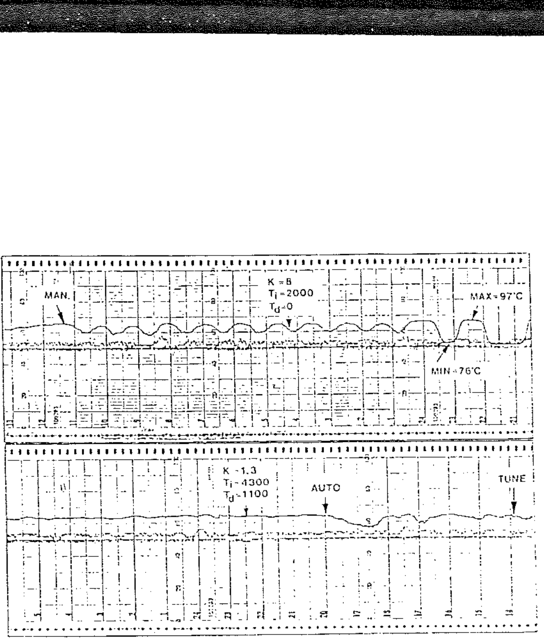

Figure 19. Results obtained applying an auto-tuner to temperature control of a destillation column. The figure is a copy of a strip chart recorder, which explains why time increases from right to left.

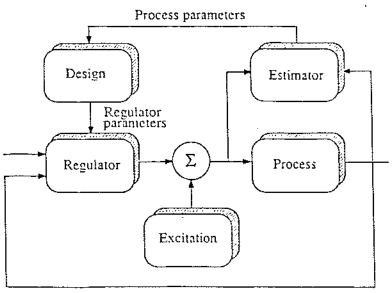

Figure 20. Block diagram of an adaptive regulator.

regulator parameters are also shown in the figure. Notice in particular the absence of derivative action. To perform automatic tuning the regulator was first switched to manual control at time 11.00. The auto-tuning mode was activated at time 14.00. At time 20.00 the tuning is complete and the regulator switches to automatic control mode. Notice the drastic improvement in the behaviour of the closed loop. Also notice that the regulator parameters have been changed significantly.

There are several interesting observations that can be drawn from the experiment. First, notice that tuning of a slow process takes a considerable time. It is not likely that an operator has the time and patience to tune the regulator manually. Secondly, if the tuning is done manually by heuristic methods it is necessary to make several tuning experiments which will increase the time substantially. These are probably the reasons why the loop was poorly tuned to start with. It is also worth observing that there are significant disturbance during the tuning phase, from time 14.00 to 20.00. The adaptive filtering in the tuner can, however, handle the disturbances very well. After one half period of the oscillation a crude estimate of the period is obtained. This value can be used to set the bandwidth of the filters.

7. Adaptive Control

A block-diagram of an adaptive regulator is shown in Figure 20. The regulator can be thought of as composed of two loops. The inner loop consists of the process and an ordinary linear feedback regulator. The parameters of the regulator are adjusted by the outer loop, which performs recursive parameter estimation and control design calculations. Notice that the system automatically performs the tasks of modeling and control design that are normally carried out by an engineer.

The block labeled "regulator design" in Figure 20 represents an on-line solution to a design problem for a system with known parameters. This is called the underlying design problem. When investigating adaptive systems it is useful to exhibit this problem explicitly, because it gives the characteristics of the system under the ideal conditions when the parameters are known exactly.

The parameter estimator attempts to find the process parameters by analysing how the process responds to control signals. Very little useful information can be derived when the process input is constant. In such cases it is useful to introduce small perturbation signals to make sure that useful estimates can be produced. This is done via the block labeled "excitation" in the figure.

The diagram shown in Figure 20 is quite general. It covers common adaptive regulators like model reference adaptive system (MRAS) and self-tuning regulators (STR). Many different design methods and many different parameter estimation schemes can be used. There are adaptive regulators based on phase- and amplitude-margin design methods, pole-placement, minimum variance control, linear quadratic gaussian control and optimization methods. Many different parameter estimation schemes have also been used, for example stochastic approximation, least squares, extended and generalized least squares, instrumental variables, extended Kalman filtering and the maximum likelihood method.

The self-tuner shown in Figure 20 is called an indirect self-tuner, because the regulator parameters are obtained indirectly via estimation of a process model and a control design. It is sometimes possible to reparameterize the process so that it can be expressed in terms of the regulator parameters. This gives a significant simplification of the algorithm because the design calculations are eliminated. Such a regulator is called a direct self-tuner. In terms of Figure 20 the block labelled design calculations disappears and the regulator parameters are updated directly.

Algorithms

EXAMPLE 1

Two examples will be used to illustrate typical adaptive algorithm. Estimate the parameters of the second order model

$$y(t) + a_1 y(t-h) + a_2 y(t-2h) = b_1 u(t-h) + b_2 u(t-2h) \quad (7.1)$$

recursively. Let $\hat{a}_1$ and $\hat{b}_i$ denote the parameter estimates. The control law

$$u(t) = t_0 r(t) - s_0 y(t) - s_1 y(t-h) - r_1 u(t-h)$$

where

$$\begin{aligned}
t_0 &= (1 + p_1 + p_2)/(\hat{b}_1 + \hat{b}_2) \\
r_1 &= \left[(p_1 - \hat{a}_1)\hat{b}_2^2 - (p_2 - \hat{a}_2)\hat{b}_2\hat{b}_2\right]/N \\
s_0 &= \left[(p_1 - \hat{a}_1)(\hat{a}_2\hat{b}_1 - \hat{a}_1\hat{b}_2) + (p_2 - \hat{a}_2)\hat{b}_2\right]/N \\
s_1 &= -\hat{a}_2 r_1/\hat{b}_2 \\
N &= \hat{b}_2^2 - \hat{a}_1\hat{b}_1\hat{b}_2 + \hat{a}_2\hat{b}_1^2
\end{aligned}$$

gives a closed loop system whose pulse transfer function from the command signal to the output is given by

$$H_m(z) = \frac{1 + p_1 + p_2}{b_1 + b_2} \cdot \frac{b_1 z + b_2}{z^2 + p_1 z + P_2}$$

where

$$p_1 = -2e^{-\zeta\omega h} cos\omega h\sqrt{1-\zeta^2}$$

and

$$p_2 = e^{-2\zeta\omega h}$$

The closed loop system will thus retain the open loop zero and the closed loop poles correspond to a sampled second order system with bandwidth ω and relative damping ζ. □

Some minor modifications of the control law in the example are needed to handle bias and integral action. A detailed discussion of these factors is given in Åström (1979). The commercial regulators, Electromax V and TCS 6355 are based on estimation of parameters in the model (7.1). They do, however, use control design methods which are different from the one used in the example.

EXAMPLE 2
The self-tuner discussed in Åström and Wittenmark (1973) is based on the mathematical model

$$\begin{aligned} y(k+d) &= s_0y(k) + s_1(k-1) + \ldots + s_{n_s}y(k-n_s) \\ &+r_0u(k) + \ldots + r_{n_r}u(k-n_r) + \varepsilon(k+d) \end{aligned} \tag{7.2}$$

where u is the control variable, y the measured output and ε is a disturbance. If ε is independent of the terms on the right hand side the minimum variance control law for the plant (7.2) is simply

$$\begin{aligned} u(k) = &-[s_0y(k) + s_1y(k-1) + \ldots + s_{n_s}y(k-n_s) \\ &+r_1u(k-1) + \ldots + r_{n_r}u(k-n_r)]/r_0 \end{aligned} \tag{7.3}$$

The basic self-tuning algorithm can be described as follows:

ALGORITHM 1
Repeat the following steps at each sampling period:

Step 1: Update the estimates of the parameters of the model (7.2), so that a weighted sum of squares of the errors ε are minimal.

Step 2: Compute the control signal $u(k)$ from past data $y(k)$, $y(k-1), \ldots, u(k-1), \ldots$ using (7.3) with the estimates obtained from Step 1. □

Notice that when least square estimation is used the error $\varepsilon(k+d)$ will be uncorrelated with the other terms in the right hand side of (7.2). Also notice that no design calculations are required since the parameters of the regulator (7.3) are obtained directly from the model parameters because of the special model structure used in (7.2). In control system design it is frequently necessary to make a trade-off between the response time and the size of the control signal. In minimum variance control this trade-off is made indirectly via selection of the sampling period. The regulator gain decreases and the response time increases with increasing sampling period. The minimum variance control law cannot handle nonminimum phase system because the process zeros are canceled by the controller. By increasing the sampling period and the delay d used in the adaptive control law the problems with nonminimum phase systems will, however, disappear. See Åström and Wittenmark (1985). Sampling of a stable system, with nonzero steady state gain, always gives a minimum phase sampled system provided the sampling period is sufficiently long. See Åström et al. (1984). This is also true for unstable systems provided that the unstability is caused by a single pole. The quality of the approximation by a low order system will also be improved when the sampling period is increased. The drawbacks with along sampling period are slow responses to disturbances and changes in the set point. Notice that a sampled data system runs open loop between the sampling instants.

Predictive Control

There have recently been a considerable interest in adaptive regulators based on predictive control. Such regulators are based on estimation of models of the type

$$\begin{aligned} &y(k+d) = s_0y(k) + s_1y(k-1) + \ldots + s_{n_s}y(k-n_s) + r_{-d} \\ &u(k+d) + \ldots + r_{-1}u(k+1) + r_0u(k) + \ldots + r_nu(k-n_r) + \varepsilon(k+d) \end{aligned} \tag{7.4}$$

The specifications are often expressed in terms of the desired step response of the closed loop system which is easy to describe to the operator. There are many different algorithms of this type e.g. the extended horizon minimum variance control, Ydstie (1984) and extended prediction self-adaptive controls (de Keyser and Van Cauvenberghe, 1982, 1985, de Keyser et al. 1985). There are also variations based on linear quadratic optimization criteria. See Peterka (1984), the Musmar algorithm Mosca et al. (1982) and Lemos and Mosca (1985). These algorithms are also related to dynamic matrix control (Cutler and Ramaker, (1980) and model predictive control (Richalet et al. 1978), which is dealt with at length in Session III of this meeting. There are also multivariable extensions of the algorithms (Rouhani and Mehra, 1982).

Prior knowledge

The adaptive regulators require significant a priori information. The model structure used must be specified. This includes a description of how the model depend on the parameters. Initial values of the parameters must also be provided. Some schemes require that the initial values are such that the closed loop system is guaranteed to be stable. An estimate of the range of parameter variations must also be given to initialize the parameter estimator. This is however less critical. To track parameter variations properly some information on drift rate of parameters must also be provided. Since practically all implementations of adaptive regulators are based on computer control it is also necessary to specify the sampling period used. This is critical.

The self-tuning regulator given in Example 2 requires the following prior knowledge:

- h sampling period
- d delay in number of sampling periods
- n_r degree of the polynomial R
- n_s degree of the polynomial S
- λ forgetting factor
- θ_0 initial estimate
- p_0 initial covariance
- uh high control limits
- ul low control limits

The sampling period is critical as was discussed above. The integer d is also crucial. The closed loop system will become unstable if h and d are underestimated. The parameters are particularly important. Since the self-tuner is based on minimum variance control they will directly determine the closed loop bandwidth. The parameters n_r and n_s are not particularly critical. A calculation of covariances of inputs and outputs will show if they are too small, see Åström (1970). The parameter λ determines the trade-off between the tracking ability and the steady state variance of the recursive parameter estimator. The parameters

θ_0 and P_0 determine the initial transient of the estimator but are otherwise unessential.

The amount of prior information needed clearly indicates that some expertise is required to commission and run adaptive systems. Several commercial systems have therefore introduced a "pre-tune" mode to help finding the prior information needed. The auto-tuner is a good way to obtain prior information.

Theory

Theory has different and important roles in analysis and design of adaptive control systems. Analysis aimed at understanding specific algorithms is one goal. Creation of new adaptive control laws is another role. Adaptive systems are inherently nonlinear. Their behaviour is also quite complex which makes them difficult to analyse. Progress in theory has been slow and painstaking. Much work remains to be done before a reasonably complete coherent theory is available.

Because of the complex behavior of adaptive systems it is necessary to consider them from several points of view. Theories of nonlinear systems, stability, system identification, recursive estimation, convergence of stochastic algorithms and optimal stochastic control have all contributed to the understanding of adaptive systems.

Many adaptive algorithms are motivated by the assumption that the parameters change slower than the other variables of the system. We can make sure that the parameters change slowly by choosing a small adaptation gain. The variables describing the adaptive system can then be separated into two groups which change at different rates. The adjustable parameters are the slow parameters and the state of the controlled dynamical system are the fast variables. It is possible to derive approximations so that the fast and the slow variables can be treated separately. This idea which originated in analysis of nonlinear oscillations is called averaging.

An Example

An example illustrates use and performance of adaptive control. A ship operates in an environment that changes with wind, waves and currents. The dynamics of a ship depend on trim, loading, ship speed and water depth. A conventional autopilot for a ship is based on the PID algorithm given by Equation 2.1. An adaptive autopilot based on recursive least squares estimation and linear quadratic control theory is manufactured by Kockumation AB in Sweden. This autopilot has a control law, which is more complicated than a PID regulator. Figure 21 shows results of steering experiments with conventional and adaptive control. The experiments are performed under similar conditions. The figure shows clearly the superior performance of the adaptive system. The heading variations are considerably smaller while the rudder motions have a similar magnitude. A closer inspection shows that there are more high frequencies in the control signal for the adaptive autopilot. The reason why the adaptive regulator performs much better is that it is more complex. It has an internal model which describes the dynamics of the ship and of wind, waves and currents. If the parameters were frozen, the performance would not change much in the short run. It would, however, deteriorate when conditions change. There are altogether 8 parameters that are estimated on line. It is quite difficult and tedious to change so many parameters manually. Adaptation is thus a necessity for using a regulator of this complexity. The reduction of deviations in heading can be translated to fuel savings. In the particular case shown in Figure 6 the adaptive autopilot results in fuel savings of 2.7%. There are today about a hundred ships that operate with the autopilot.

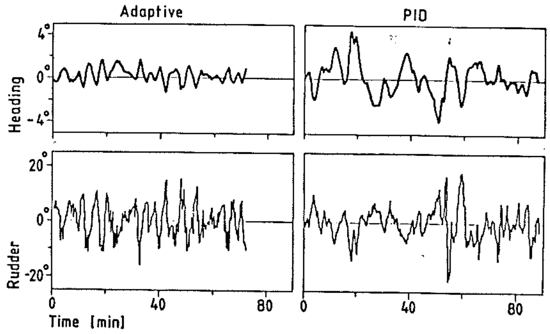

Figure 21. Heading and rudder angles for a conventional and an adaptive autopilot.

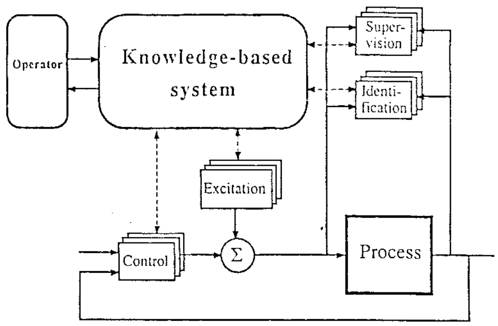

Figure 22. Block diagram of a knowledge-based control system.

8. Knowledge-based Control

The properties of auto-tuners and adaptive regulators are complimentary. The auto-tuner requires little prior information. It is very robust and it can generate good parameters for a simple control law. Adaptive regulators like model reference adaptive controllers or self-tuning regulators can use more complex control laws with potentially better performance. These control laws are local gradient procedures. Starting from reasonably good a priori information on system structure, sampling period, and parameters, the algorithms can adjust the regulator parameters to give a closed loop system with good performance. The algorithms will, however, not work if the prior guesses are too poor. With bad prior data they may even give unstable closed loop systems. The adaptive algorithms are also capable of tracking a system provided that the process parameters change slowly.

It thus seems natural to try to combine auto-tuners and adaptive control algorithm. Pursuing this idea further it seems natural to also include algorithms for monitoring and supervision of the closed loop performance. In Åström, Anton and Årzén (1985) it was proposed to use an expert system to coordinate the different algorithms. The notion of knowledge-based control or expert control has been coined to describe a system of this type. A block diagram of such a system is shown in Figure 22. Knowledge-based control systems are currently being investigated in my laboratory. Such systems have the interesting property that the knowledge about the control problem is represented explicitly and that it can be explored and manipulated. This offers interesting possibilities. We can thus envision a control system that can answer questions like: What is the current knowledge of the process and its environment? Are the fluctuations in the process output normal? What control law is being used? Why was this control law chosen? Why is derivative action not used? List the loops where dead time compensation is needed? List all loops where the regulator parameters have changed significantly during the past two months. What tuning procedure is appropriate for this loop? Monitor the stability margin for this loop. Since the knowledge representation is explicit, it can also be transferred when the hardware is replaced.

9. Conclusions

It is often a tedious task to find suitable parameters of a control law. Control engineers have for a long time been faced with the challenge of doing this automatically. A lot of research work and experimentation are now bearing fruit. Combined with the advances in microelectronics commercial systems for automatic tuning and adaptation are now appearing. The experiences of using such devices have been quite promising. The advanced regulators result in improved product quality and energy savings. The regulators also simplify commissioning and operation of industrial plants. Research work in universities are preparing the ground for the next generation of systems, which also will include AI techniques in the form of knowledge-based systems. Research work which attempts to mimic simple neural networks in silicon is also under way, see Hopfield and Tank (1986). Combined these efforts are pointing towards the appearance of control systems with rudimentary forms of intelligence.

10. References

The following recommendations can be made for further reading. The book Åström and Hägglund (1988) is an up-to date in depth treatment of PID control. Many different digital control algorithms are discussed in Åström and Wittenmark (1985). This book also deals with implementation of digital control systems. Automatic tuning is described in Åström and Hägglund (1984, 1988) and Hägglund and Åström (1985). The books Åström and Wittenmark (1985), Ljung and Söderström (1983) and Goodwin and Sin (1984) are comprehensive sources for the details of the methods of control design, parameter estimation and adaptation. The book Gupta (1986) is a reprint of many classical papers on adaptive control. It is a good source to the original papers in the field. The book by Anderson et al (1986) is an up-to-date account of some current research problems.

ANDERSON, B.D.O., R.R. BITMEAD, C.R. JOHNSON, JR. P.V. KOKOTOVIC, R.L. KOSUT, I.M.Y. MAREELS, L. PRALY and B.D. RIEDLE (1986): *Stability of Adaptive Systems-Passivity and Averaging Analysis*, MIT Press, Cambridge MA.

ÅSTRÖM, K.J., "Adaptive Feedback Control," *Proc. IEEE* **75**, 185-217.

ÅSTRÖM, K.J. AND B. WITTENMARK (1973): "On Self-tuning Regulators," *Automatica* **9**, 185-199.

ÅSTRÖM, K.J. AND T. HÄGGLUND (1984): "Automatic Tuning of Simple Regulators," in in I.D. Landau et al. (Eds.): *Proc. IFAC Workshop*, Pergamon Press, USA.

ÅSTRÖM, K.J., J.J. ANTON AND K.E. ÅRZÉN (1984): "Expert Control," *Automatica* **22**.

ÅSTRÖM, K.J. AND T. HÄGGLUND (1984): "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," *Automatica* **20**.

ÅSTRÖM, K.J. AND B. WITTENMARK (1985): "The Self-runing Regulators Revisited," *Proc. 7th IFAC Symposium*, York, Uk.

ÅSTRÖM, K.J. AND B. WITTENMARK (1985): *Computer Controlled Systems - Theory and Design*, Prentice Hall, Englewood Cliffs N.J..

ÅSTRÖM, K.J. AND T. HÄGGLUND (1988): *Automatic Tuning of PID Regulators*, ISA to appear.

ÅSTRÖM. K.J. AND B. WITTENMARK (1988): *Adaptive Control*, Addison Wesley, M.A..

CUTLER, C.R. AND B.C. RAMAKER (1980): "Dynamic Matrix Control - A Computer Control Algorithm," paper WP5-B , *Joint Automatic Control Conference*, San Francisco.

GOODWIN, G.C. AND K.S. SIN (1984): *Adaptive Filtering, Prediction and Control*, Prentice Hall, Englewood Cliffs, N.J..

GUPTA, M.M. (Ed.) (1986): *Adaptive Methods for Control System Design*, IEEE Press, The Institute of Electrical and Electronics Engineers, Inc., New York.

HÄGGLUND, T. AND K.J. ÅSTRÖM (1985): "Method and an Apparatus in Tuning a PID Regulator," U.S. Patent 4549123.

HOPFIELD, J.J. AND D.W. TANK (1986): "Computing with Neutral Circuits: A Model," *Science* **233**, 625-633.

DE KEYSER AND A.R. VAN CAUWENBERGHE (1982): "Simple Self-tuning Multistep Predictors," in G.A. Bekey, G.N. Saridis: *Proc. 6th IFAC Symposium*, Pergamon Press, Oxford, pp. 1558-1563.

DE KEYSER, R.M.C. AND A.R. VAN CAUWNBERGHE (1985): "Extended Prediction Self-adaptive Control," *Proc. 7th IFAC/IFORS Symp on Identification and System Parameter Estimation*, York, UK.

DE KEYSER, R.M.C., PH, G.A. VAN DE VELDE AND F.A.G. DUMORTIER (1985): "A Comparative Study of Self-adaptive Long-range Predictive Control Methods," *Proc. 7th IFAC/IFORS Symp on Identification and System Parameter Estimation*, York, UK.

LEMOS, J.M. AND E. MOSCA (1985): "A Multipredictor-based LQ Self-tuning Controller," *Proc. 7th IFAC-IFORS Symp on Identification and System Parameter Estimation*, York, UK.

LJUNG, L. AND T. SÖDERSTRÖM (1983): *Theory and Practice of Recursive Identification*, MIT Press, Cambridge MA.

MOSCA, E., G. ZAPPA AND C. MANFREDI (1982): "Progress on Multistep Horizon Self-tuners: The MUSMAR Approach," *Ricerche di Automatica XIII* **XIII**, 85-105.

PETERKA, V (1984): "Predictor-based Self-tuning Control," *Automatica* **20**, 39-50.

RICHALET, J., A. RAULT, J.L. TESTUD AND J. PAPON (1978): "Model Predictive Heuristic Control: Applications to Industrial Processes," *Automatica* **14**, 413-428.

ROUHANI, R. AND R.K. MEHRA (1982): "Model Algorithm Control - Basic Properties," *Automatica* **18**, 401-414.

WITTENMARK, B. AND K.J. ÅSTRÖM (1984): "Practical Issues in the Implementation of Self-tuning Control," *Automatica* **20**, 595-605.

YDSTIE, B. (1984): "Extended Horizon Adaptive Control," paper 14.4/E-4 , *Proc. 9th IFAC World Congress*, Budapest, Hungary.

ZIEGLER, J.G. AND N.B. NICHOLS (1943): "Optimum Settings for Automatic Controllers," *Trans. ASME* **65**, 433-444.