9

Automatic Tuning and Adaptation

9.1 Introduction

Automatic tuning, or auto-tuning, is a method where the controller is tuned automatically on demand from a user. Typically, the user will either push a button or send a command to the controller. Automatic tuning of PID controllers can be accomplished by combining the methods for determining process dynamics, described in Chapter 2, with the methods for computing the parameters of a PID controller, described in Chapters 4, 6, and 7. An automatic tuning procedure consists of three steps:

- Generation of a process disturbance.
- Evaluation of the disturbance response.
- Calculation of controller parameters.

This is the same procedure that an experienced engineer uses when tuning a controller manually. The process must be disturbed in some way in order to determine the process dynamics. This can be done in many ways, e.g., by adding steps, pulses, or sinusoids to the process input. The evaluation of the disturbance response may include a determination of a process model or a simple characterization of the response.

Industrial experience has clearly shown that automatic tuning is a highly desirable and useful feature. Automatic tuning is sometimes called tuning on demand or one-shot tuning. Commercial PID controllers with automatic tuning facilities have been available since the beginning of the eighties.

Automatic tuning can be built into a controller. It can also be performed using external devices that are connected to the control loop only during the tuning phase. Controller parameters are displayed when the tuning experiment is finished. Since the tuning devices are supposed to work together with controllers from different manufacturers, they must be provided with a lot of information about the controller in order to give an appropriate parameter suggestion.

Even when automatic tuning devices are used, it is important to obtain a certain amount of process knowledge. This is discussed in the Section 9.2. Automatic tuning is only one way to use the adaptive technique. Section 9.3 gives an overview of several adaptive techniques, as well as a discussion about their use. The automatic tuning approaches can be divided into two categories, namely, model-based approaches and rule-based approaches. In the model-based approaches, a model of the process is obtained explicitly, and the tuning is based on this model. Section 9.4 treats approaches where the model is obtained from transient response experiments, frequency response experiments, and parameter estimation. In the rule-based approaches, no explicit process model is obtained. The tuning is instead based on rules similar to those rules that an experienced operator uses to tune the controller manually. The rule-based approach is treated in Section 9.5. Section 9.7 treats iterative feedback tuning, which is an iterative method to tune the controllers.

A few industrial products with adaptive facilities are presented in Section 9.8. This section illustrates how some of the ideas are used in products. It is not intended as an exhaustive presentation of products. The chapter ends with conclusions and references in Sections 9.9 and 9.10.

9.2 Process Knowledge

In this chapter we will discuss several methods for automatic tuning. Before going into details we must remark that poor behavior of a control loop can not always be corrected by tuning the controller. It is absolutely necessary to understand the reason for the poor behavior.

The process may be poorly designed so that there are long dead times, long time constants, nonlinearities, and inverse responses. Sensors and actuators may be poorly placed or badly mounted, and they may have bad dynamics. Typical examples are thermocouples with heavy casings that make their response slow or on-off valve motors with long travel time. Valves may be over-sized so that they only act over a small region. The sensor span may be too wide so that poor resolution is obtained, or it may also have excessive sensor noise.

There may also be failure and wear in the process equipment. Valves may have excessive stiction. There may be backlash due to wear. Sensors may drift and change their properties because of contamination.

If a control loop is behaving unsatisfactorily, it is essential that we first determine the reason for this before tuning is attempted. It would, of course, be highly desirable to have aids for the process engineer to do the diagnosis. Automatic tuning may actually do the wrong thing if it is not applied with care. For example, consider a control loop that oscillates because of friction in the actuator. Practically all tuning devices will attempt to stabilize the oscillation by reducing the controller gain. This will only increase the period of the oscillation! These important questions are treated in a separate chapter in this book, Chapter 10. Remember that no amount of so called "intelligence" in equipment can replace real process knowledge.

9.3 Adaptive Techniques

Techniques for automatic tuning grew out of research in adaptive control. Adaptation was originally developed to deal with processes with characteristics that were changing with time or with operating conditions. Practically all adaptive techniques can be used for automatic tuning. The adaptive controller is simply run until the parameters have converged, and the parameters are then kept constant. The drawback with this approach is that adaptive controllers may require prior information. There are many special techniques that can be used. Industrial experience has shown that automatic tuning is probably the most useful application of adaptive techniques. Gain scheduling is also a very effective technique to cope with processes that change their characteristics with operating conditions. An overview of these techniques will be given in this section. In this book the phrase *adaptive techniques* will include auto-tuning, gain scheduling, and adaptation.

Adaptive Control

An adaptive controller adjusts its parameters continuously to accommodate changes in process dynamics and disturbances. Adaptation can be applied both to feedback and feedforward control parameters. It has proved particularly useful for feedforward control. The reason for this is that model fidelity is crucial for feedforward control. Adaptive control is sometimes called continuous adaptation to emphasize that parameters are changed continuously.

There are two types of adaptive controllers based on direct and indirect methods. In a direct method, controller parameters are adjusted directly from data in closed-loop operation. In indirect methods, the controller parameters are obtained indirectly by first updating a process model on line, and then determining the controller parameters from some method for control design. The model reference system is a direct adaptive controller. The self-tuning regulator can be implemented both for direct and indirect control. There is a large number of methods available both for direct and indirect methods. They can conveniently be described in terms of the methods used for modeling and control design.

A block diagram of an indirect adaptive controller is shown in Figure 9.1. There is a parameter estimator that determines the parameters of the model based on observations of process inputs and outputs. There is also a design block that computes controller parameters from the model parameters. If the system is operated as a tuner, the process is excited by an input signal. The parameters can either be estimated recursively or in batch mode. Controller parameters are computed, and the controller is commissioned. If the system is operated as an adaptive controller, parameters are computed recursively, and controller parameters are updated when new parameter values are obtained.

Automatic Tuning

By automatic tuning (or auto-tuning) we mean a method where a controller is tuned automatically on demand from a user. Typically, the user will either push a button or send a command to the controller. Industrial experience has clearly indicated that this is a highly desirable and useful feature. Automatic

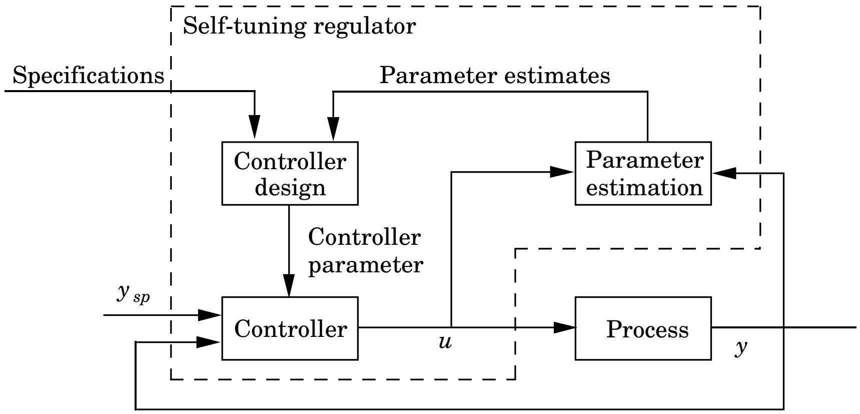

Figure 9.1 Block diagram of an adaptive controller.

tuning is sometimes called tuning on demand or one-shot tuning. Auto-tuning can be built into the controllers. Practically all controllers can benefit from tools for automatic tuning. This will drastically simplify the use of controllers. Single-loop controllers and distributed systems for process control are important application areas. Most of these controllers are of the PID type. Automatic tuning is currently widely used in PID controllers.

Auto-tuning can also be performed with external devices that are connected to a process. Since these systems have to work with controllers from different manufacturers, they must be provided with information about the controller structure in order to give an appropriate parameter suggestion. Such information includes controller structure (standard, series, or parallel form), sampling rate, filter time constants, and units of the different controller parameters (gain or proportional band, minutes or seconds, time or repeats/time).

Gain Scheduling

Gain scheduling is a technique that deals with nonlinear processes, processes with time variations, or situations where the requirements on the control change with the operating conditions. To use the technique it is necessary to find measurable variables, called scheduling variables, that correlate well with changes in process dynamics. The scheduling variable can be, for instance, the measured signal, the control signal, or an external signal. For historical reasons the phrase *gain scheduling* is used even if other parameters than the gain, e.g., derivative time or integral time, are changed. Gain scheduling is a very effective way of controlling systems whose dynamics change with the operating conditions. Gain scheduling has not been used much because of the effort required to implement it. When combined with auto-tuning, however, gain scheduling is very easy to use.

A block diagram of a system with gain scheduling is shown in Figure 9.2. The system can often be viewed as having two loops. There is an inner loop, composed of the process and the controller, and an outer loop, which adjusts the controller parameters based on the operating conditions. There are also

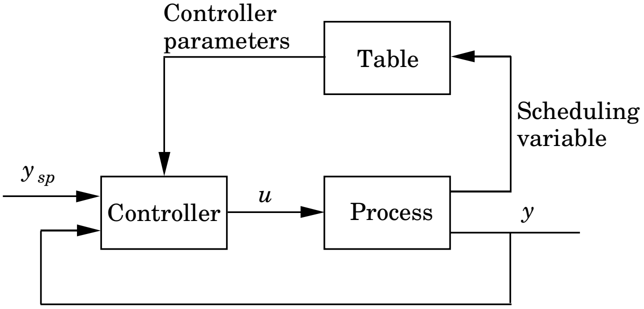

Figure 9.2 Block diagram of a system with gain scheduling.

situations when there is no outer loop, and the scheduling variable is unaffected by the controller output.

The notion of gain scheduling was originally used for flight control systems, but it is being used increasingly in process control. It is, in fact, a standard ingredient in some single-loop PID controllers. For process control applications significant improvements can be obtained by using just a few sets of controller parameters.

Gain scheduling is often an alternative to adaptation. It has the advantage that it can follow rapid changes in the operating conditions. The key problem is finding suitable scheduling variables. Possible choices are the control signal, the process variable, or an external signal. Production rate is often a good choice in process control applications, since time constants and time delays are often inversely proportional to production rate.

Development of a schedule may take a substantial engineering effort. The availability of automatic tuning can significantly reduce the effort because the schedules can then be determined experimentally. A scheduling variable is first determined. Its range is quantified into a number of discrete operating conditions. The controller parameters are then determined by automatic tuning when the system is running in one operating condition. The parameters are stored in a table. The procedure is repeated until all operating conditions are covered. In this way it is easy to install gain scheduling into a computer-controlled system by programming a table for storing and recalling controller parameters and appropriate commands to accomplish this.

Uses of Adaptive Techniques

We have described three techniques that are useful in dealing with processes that have properties changing with time or with operating conditions. In Figure 9.3 is a diagram to guide in choosing the different adaptive techniques.

Controller performance is the first thing to consider. If the requirements are modest, a controller with constant parameters and conservative tuning can be used. With higher demands on performance, other solutions should be considered. If the process dynamics are constant, a controller with constant parameters should be used. The parameters of the controller can be obtained using auto-tuning.

If the process dynamics or the nature of the disturbances are changing, it

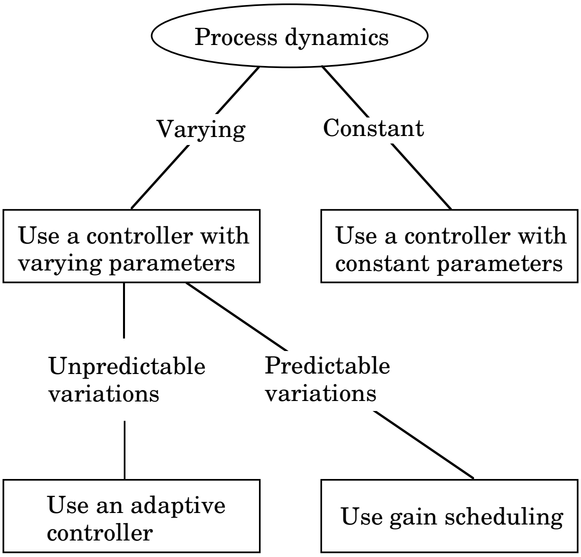

Figure 9.3 When to use different adaptive techniques.

is useful to compensate for these changes by changing the controller. If the variations can be predicted from measured signals, gain scheduling should be used because it is simpler and gives superior and more robust performance than the continuous adaptation. Typical examples are variations caused by nonlinearities in the control loop. Auto-tuning can be used to build up the gain schedules.

There are also cases where the variations in process dynamics are not predictable. Typical examples are changes due to unmeasurable variations in raw material, wear, fouling, etc. These variations cannot be handled by gain scheduling, since no scheduling variable is available, but must be dealt with by adaptation. An auto-tuning procedure is often used to initialize the adaptive controller. It is then sometimes called pre-tuning or initial tuning.

Feedforward control deserves special mentioning. It is a very powerful method for dealing with measurable disturbances. Use of feedforward control, however, requires good models of process dynamics. It is difficult to tune feedforward control loops automatically on demand, since the operator often cannot manipulate the disturbance used for the feedforward control. To tune the feedforward controller it is necessary to wait for an appropriate disturbance. Adaptation, therefore, is particularly useful for the feedforward controller.

9.4 Model-Based Methods

This section gives an overview of automatic tuning approaches that are based on an explicit derivation of a process model. Models can be obtained in many ways, as seen in Chapter 2. In this section we discuss approaches based on transient responses, frequency responses, and parameter estimation. The methods can also be characterized in terms of open and closed loop methods.

Transient Response Methods

Auto-tuners can be based on open-loop or closed-loop transient response analysis. Methods for determining the transient response were discussed in Section 2.7. The most common methods are based on step or pulse responses, but there are also methods that can use many other types of perturbations.

Open-Loop Tuning A simple process model can be obtained from an open-loop transient response experiment. A step or a pulse is injected at the process input, and the response is measured. To perform such an experiment, the process must be stable. If a pulse test is used, the process may include an integrator. It is important that the process be in equilibrium when the experiment is begun.

There are only one or two parameters that must be set *a priori*, namely, the amplitude and the signal duration. The amplitude should be chosen sufficiently large so that the response is easily visible above the noise level. On the other hand, it should be as small as possible in order not to disturb the process more than necessary and to keep the dynamics linear. The noise level can be determined automatically at the beginning of the tuning experiment. However, even if the noise level is known, we cannot decide a suitable magnitude of a step in the control signal without knowing the gain of the process. Therefore, it must be possible for the operator to decide the magnitude.

The duration of the experiment is the second parameter that normally is set *a priori*. If the process is unknown, it is very difficult to determine whether a step response has settled or not. An intuitive approach is to say that the measurement signal has reached its new steady state if its rate of change is sufficiently small. The rate of change is related, however, to the time constants of the process, which are unknown. If a pulse test is used, the duration of the pulse should also be related to the process time constants.

Many methods can be used to extract process characteristics from a transient response experiment. Most auto-tuners determine the static gain, the apparent time constant, and the apparent dead time. The static gain is easy to find accurately from a step-response experiment by comparing the stationary values of the control signal and the measurement signal before and after the step change. The time constant and the dead time can be obtained in several ways, see Section 2.7.

The transient response methods are often used in a pre-tuning mode in more complicated tuning devices. The main advantage of the methods, namely, that they require little prior knowledge, is then exploited. It is also easy to explain the methods to plant personnel. The main drawback with the transient response methods is that they are sensitive to disturbances. This drawback is less important if they are used only in the pre-tuning phase.

Closed-Loop Tuning Automatic tuning based on transient response identification can also be performed in closed loop. The steps or pulses are then added

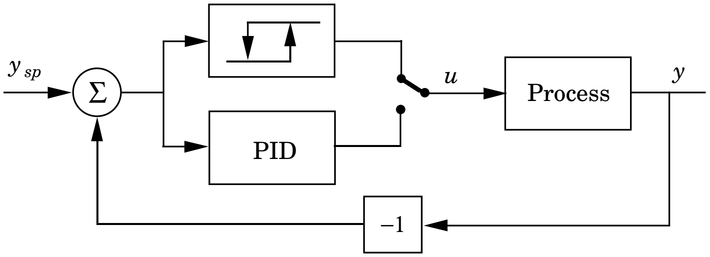

Figure 9.4 The relay auto-tuner. In the tuning mode the process is connected to relay feedback.

either to the set point or to the control signal. There are also auto-tuners that do not introduce any transient disturbances. Perturbations caused by set-point changes or load disturbances are used instead. In these cases it is necessary to detect that the perturbations are sufficiently large compared to the noise level.

Closed-loop tuning methods cannot be used on unknown processes. Some kind of pre-tuning must always be performed in order to close the loop in a satisfactory way. On the other hand, they do not usually require any additional *a priori* information. The magnitude of the step changes in set point are easily determined from the desired, or accepted, change in the measurement signal.

Since a proper closed-loop transient response is the goal for the design, it is appealing to base tuning on closed-loop responses. It is easy to give design specifications in terms of the closed-loop transient response, e.g., damping, overshoot, closed-loop time constants, etc. The drawback is that the relation between these specifications and the PID parameters is normally quite involved. Heuristics and logic are required therefore.

Frequency Response Methods

There are also auto-tuners that are based on frequency response methods. In Section 2.7, it was shown how frequency response techniques could be used to determine process dynamics.

Use of the Relay Method In traditional frequency response methods, the transfer function of a process is determined by measuring the steady-state responses to sinusoidal inputs. A difficulty with this approach is that appropriate frequencies of the input signal must be chosen *a priori*. A special method, where an appropriate frequency of the input signal is generated automatically, was described in Section 2.7. The idea was simply to introduce a nonlinear feedback of the relay type in order to generate a limit cycle oscillation. With an ideal relay the method gives an input signal to the process with a period close to the ultimate frequency of the open-loop system.

A block diagram of an auto-tuner based on the relay method is shown in Figure 9.4. Notice that there is a switch that selects either relay feedback or ordinary PID feedback. When it is desired to tune the system, the PID function is disconnected and the system is connected to relay feedback control. The system then starts to oscillate. The period and the amplitude of the oscillation is determined when steady-state oscillation is obtained. This gives the ultimate period and the ultimate gain. The parameters of a PID controller can then be determined from these values. The PID controller is then automatically switched in again, and the control is executed with the new PID parameters.

The initial amplitude of the relay must be specified in advance. A feedback loop from measurement of the amplitude of the oscillation to the relay amplitude can be used to ensure that the output is within reasonable bounds during the oscillation. It is also useful to introduce hysteresis in the relay. This reduces the effects of measurement noise and also increases the period of the oscillation. With hysteresis there is an additional parameter. This can be set automatically, however, based on a determination of the measurement noise level. Notice that there is no need to know time scales *a priori* since the ultimate frequency is determined automatically from the experiment.

In the relay method, an oscillation with suitable frequency is generated by a static nonlinearity. Even the order of magnitude of the time constant of the process can be unknown. Therefore, this method is not only suitable as a tuning device; it can also be used in pre-tuning. It is also suitable for the determination of sampling periods in digital controllers.

The relay tuning method also can be modified to identify several points on the Nyquist curve. This can be accomplished by making several experiments with different values of the amplitude and the hysteresis of the relay. A filter with known characteristics can also be introduced in the loop to identify other points on the Nyquist curve.

On-Line Methods Frequency response analysis can also be used for on-line tuning of PID controllers. By introducing bandpass filters, the signal content at different frequencies can be investigated. From this knowledge, a process model given in terms of points on the Nyquist curve can be identified and tracked on line. In this auto-tuner the choice of frequencies in the bandpass filters is crucial. This choice can be simplified by using the tuning procedure described above in a pre-tuning phase.

Parameter Estimation Methods

A common tuning procedure is to use recursive parameter estimation to determine a low-order discrete time model of the process. The parameters of the low-order model obtained are then used in a design scheme to calculate the controller parameters. An auto-tuner of this type can also be operated as an adaptive controller that changes the controller parameters continuously. Auto-tuners based on this idea, therefore, often have an option for continuous adaptation.

The main advantage of auto-tuners of this type is that they do not require any specific type of excitation signal. The control signal can be a sequence of manual changes of the control signal, for example, or the signals obtained during normal operation. A drawback with auto-tuners of this type is that they require significant prior information. A sampling period for the identification procedure must be specified; it should be related to the time constants of the closed-loop system. Since the identification is performed on line, a controller that at least manages to stabilize the system is required. Systems based on

Table 9.1 Rules of thumb for the effects of the controller parameters on speed and stability in the control loop.

	Speed	Stability
K increases	increases	reduces
T_i increases	reduces	increases
T_d increases	increases	increases

this identification procedure need a pre-tuning phase, which can be based on the methods presented earlier in this section.

9.5 Rule-Based Methods

This section treats automatic tuning methods that do not use an explicit model of the process. Tuning is based instead on the idea of mimicking manual tuning by an experienced process engineer.

Controller tuning is a compromise between performance and robustness. Table 9.1 shows how stability and speed change when the PID controller parameters are changed. Note that the table only contains rules of thumb. There are exceptions. For example, an increased gain often results in more stable control when the process contains an integrator. The same rules can also be illustrated in tuning maps. See, for example, the tuning map for PI control in Figure 6.7.

The rule-based automatic tuning procedures wait for transients, set-point changes, or load disturbances in the same way as the model-based methods. When such a disturbance occurs, the behavior of the controlled process is observed. If the control deviates from the specifications, the controller parameters are adjusted based on some rules.

Figures 9.5 and 9.6 show set-point changes of control loops with a poorly tuned PI controller. The response in Figure 9.5 is very sluggish. Here, a correct rule is to increase the gain and to decrease the integral time. Figure 9.6 also shows a sluggish response because of a too large integral time. The response is also oscillatory because of a too high gain. A correct rule, therefore, is to decrease both the gain and the integral time.

If graphs like those in Figures 9.5 and 9.6 are provided, it is easy for an experienced operator to apply correct rules for controller tuning. To obtain a rule-based automatic tuning procedure, the graphs must be replaced by quantities that characterize the responses. Commonly used quantities are overshoot and decay ratio to characterize the stability of the control loop and time constant and oscillation frequency to characterize the speed of the loop.

It is rather easy to obtain relevant rules that tell whether the different controller parameters should be decreased or increased. However, it is more difficult to determine *how much* they should be decreased or increased. The rule-based methods are, therefore, more suitable for continuous adaptation

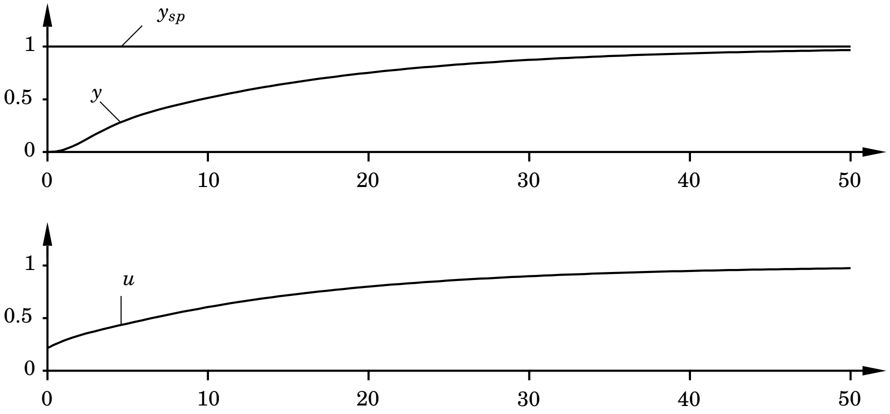

Figure 9.5 A set-point response where a correct rule is to increase the gain and decrease the integral time. The upper diagram shows set-point y_{sp} and process output y, and the lower diagram shows control signal u.

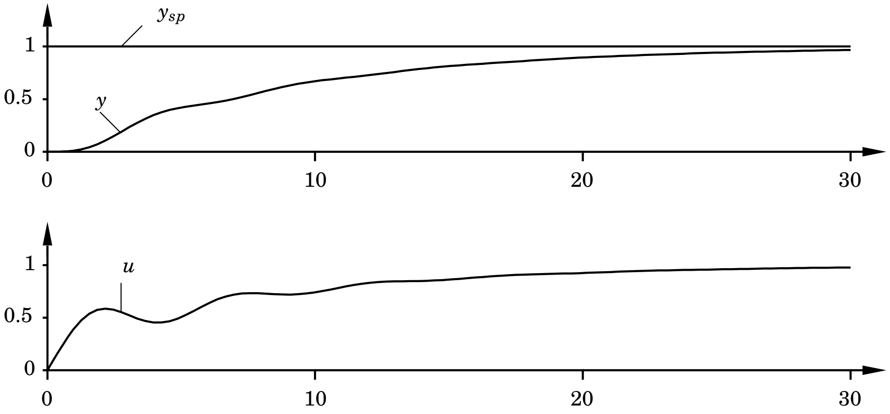

Figure 9.6 A set-point response where a correct rule is to decrease the gain and decrease the integral time. The upper diagram shows set point y_{sp} and process output y, and the lower diagram shows control signal u.

where rather small successive changes in the controller parameters are performed after each transient.

The rule-based methods have a great advantage compared to the model-based approaches when they are used for continuous adaptation, namely, that they handle load disturbances efficiently and in the same way as set-point changes. The model-based approaches are well suited for set-point changes. However, when a load disturbance occurs, the transient response is caused by an unknown input signal. To obtain an input-output process model under such circumstances is not so easy.

A drawback with the rule-base approaches is that they normally assume that the set-point changes or load disturbances are isolated steps or pulses.

Two set-point changes or load disturbances applied shortly after each other may result in a process output that invokes an erroneous controller tuning rule.

9.6 Supervision of Adaptive Controllers

Automatic tuning and gain scheduling have been well accepted by the process industry and are now common both in single-station controllers and distributed control systems. There are many well-engineered auto-tuners that are very easy to use. The industrial use of the "true" adaptive controller is, however, more limited. There are several reasons for this. One is that many controllers that have been tested industrially have not been sufficiently robust. This has tarnished the technique with a somewhat bad reputation. The adaptive algorithms must be provided with a supervisory shell that takes care of those operating conditions that the algorithm is not designed for.

The problem is not unique to adaptive controllers. *Every* controller needs a supervisory shell. The simple PID controller, e.g., has antiwindup functions to treat the situation when the control signal saturates, functions for bumpless transfer at mode switches between manual and automatic control, functions for bumpless transfer at parameter changes, and sometimes dead-zones and control signal rate limitations. This section discusses some supervisory functions for adaptive controllers.

Initialization

The first topic to consider is the initialization of the adaptive controller. Initialization should ensure that suitable controller parameters are used when the adaptation starts. An adaptive controller also requires additional parameters that should be obtained in the initialization phase. For example, the adaptive controllers, both the model based and the rule based, need to know the time scale of the process. It is used to set sampling periods and time constants.

In special-purpose adaptive controllers, the initialization can be performed manually by an experienced user. However, in multi-purpose adaptive controllers, this phase should not be left to unexperienced users. It should be performed automatically. Therefore, almost all industrial multi-purpose adaptive controllers have some kind of automatic tuning or pre-tuning function that initializes the adaptive controller. These procedures may be based on step response experiments, which provide the time scale of the process in terms of the apparent dead time and the apparent time constant. They can also be based on a relay feedback experiment. In this case the time scale of the process is obtained in terms of the ultimate frequency ω_u.

In the following, it is assumed that the time scale of the process has been obtained and is available in the adaptive controller. It is denoted by T_p. It is also assumed that the design calculations are performed in such a way that T_p also is proportional to the closed-loop time constant. The initialization procedure is not only invoked once when the adaptive controller is installed. Parts of the initialization procedure have to be used at mode transitions and parameter changes too.

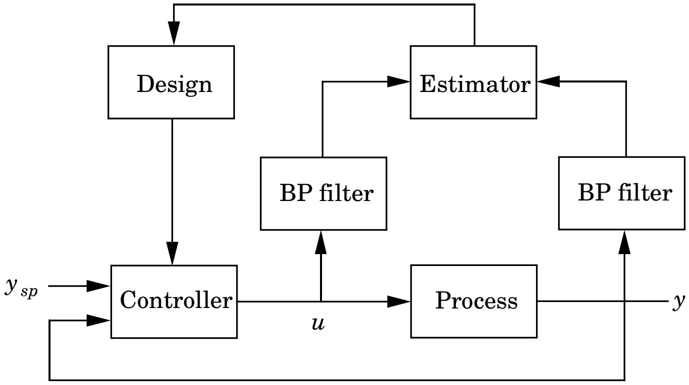

Figure 9.7 Filtering of control and measurement signals.

An important feature of the initialization procedure is to obtain suitable time constants of filters. Low-frequency components of the signals should be reduced in order to eliminate bias terms. High-frequency components are normally corrupted with measurement noise that disturbs the parameter estimator. Therefore, the control signal and the measurement signal should be band-pass filtered before entering the parameter estimator. See Figure 9.7.

It is important that parameter estimation is based on relevant data. If the model order is low, it is particularly important that the model be fitted to data in a frequency region that is suitable for controller design, namely, the frequency range around the ultimate frequency. This frequency range is determined by the choice of time constants in the band-pass filters. The frequency range can be made narrow or wide, depending on the control objective and the estimated model order. In the ECA600 controller, see Section 9.8, a narrow band-pass filter is used, and a process model consisting of only two parameters is identified. Models with more parameters require wider filters.

Excitation Detection

The parameter estimator is the central part of an adaptive controller. A recursive least squares estimator is normally used. This can be described by

$$\begin{aligned}
\hat{\theta}(t) &= \hat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)\\
\varepsilon(t) &= y(t) - \varphi(t)^T\hat{\theta}(t-1)\\
P(t) &= P(t-1) - \frac{P(t-1)\varphi(t)\varphi(t)^T P(t-1)}{1+\varphi(t)^T P(t-1)\varphi(t)},
\end{aligned} \tag{9.1}$$

where $\hat{\theta}$ is the parameter estimates, P is the covariance matrix, and φ is the regression vector, which normally contains delayed measurement and control signals. To be able to track variations in process dynamics, it is necessary to rely more on recent data than on older. This is often ensured by introducing a forgetting factor λ and modifying the covariance matrix according to

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi(t)^T P(t-1)}{\lambda + \varphi(t)^T P(t-1)\varphi(t)}\right). \tag{9.2}$$

A forgetting factor in the range $0 < \lambda < 1$ prevents the covariance matrix from converging to zero. The choice of λ is a compromise between adaptation rate and robustness. Decreasing λ will, e.g., result in an increased adaptation rate but also decreased robustness. The introduction of a forgetting factor may cause problems if the excitation is not good enough. Suppose, e.g., that φ is zero for a certain period. From Equation 9.2 it then follows that the covariance matrix will increase exponentially. There are ways to overcome this problem, e.g., by using a variable forgetting factor or by using directional forgetting. It has also been proposed to reinitialize the covariance matrix periodically to ensure that P stays within certain bounds. This will surely solve the numerical problem but in such a way that the estimation uncertainty is varying periodically, which is unsatisfactory.

The excitation problem is not only a numerical problem. The problem is also to ensure that the parameter estimator is provided with enough relevant data to produce a reliable process model. There are in principle two solutions to this problem:

1. Ensure that excitation always is present by adding excitation signals to the process input.

2. Ensure that estimation is performed only when there is enough natural excitation of the process.

The first approach might seem appealing. An excitation signal that is so small that it is hardly noticeable compared to the normal measurement noise will not do much harm. Unfortunately, such an excitation is not of much help for parameter estimation. The excitation signal must have a significant amplitude to be of any use. Friction or other nonlinearities may otherwise distort or even eliminate the response from the process output, and the excitation is lost. An excitation signal with a significant amplitude causes degradation of the control, and can therefore only be accepted during short periods such as during an automatic tuning experiment. For these reasons, the first approach is seldom used in industrial controllers. Instead, the second approach is used.

To ensure that estimation is only performed after significant changes in set point or load, when there is enough excitation, a procedure that measures the excitation is needed.

A convenient approach for excitation detection that is similar to the one used in the ECA600 controller will now be described. The basic idea is to make a high-pass filtering of the measurement signal. When the magnitude of the filtered variable exceeds a certain threshold, it is concluded that the excitation is high enough for adaptation. The high-pass filter is given by

$$Y_{hp} = \frac{s}{s + \omega_{hp}} Y, \tag{9.3}$$

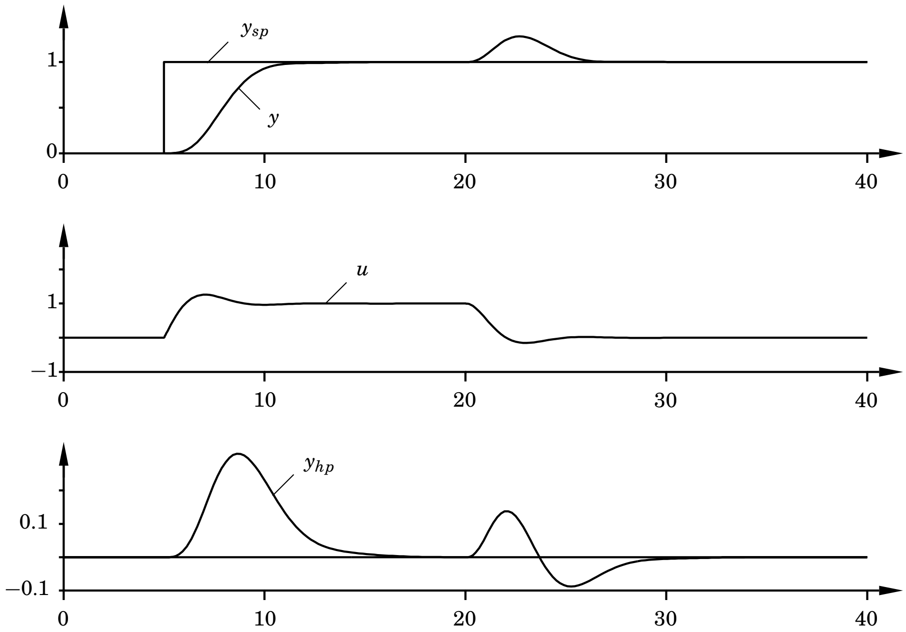

Figure 9.8 Excitation detection using high-pass filtering of the measurement signal. The figure shows responses to a set-point change at $t = 5$ and a load disturbance at $t = 20$.

where Y is the Laplace transform of the process output y, and Y_{hp} is the corresponding high-pass filtered signal. The filter has unit gain at high frequencies. The frequency ω_{hp} is chosen to be inversely proportional to the process time scale T_p.

Figure 9.8 shows a simulation where the measurement signal is passed through the high-pass filter (9.3). From the figure, it is obvious that the output from the high-pass filter is suitable for excitation detection. Excitation is high and adaptation can be initiated when the magnitude of $|y_{hp}|$ becomes large.

The next problem is to decide when the excitation is so low that the estimation should be interrupted again. One approach is to allow adaptation as long as $|y_{hp}|$ remains large. A drawback with this approach is that there are delays in the estimator. This means that even if $|y_{hp}|$ is small, there might still be excitation in the filtered signals in the parameter estimator. A solution to the problem is to simply allow adaptation for a fixed time after excitation has been detected.

Load Disturbance Detection

Model-based adaptive controllers have problems with load disturbances. To see this, consider the block diagram in Figure 9.9. The process output y is given by

$$Y(s) = P(s)\,(U(s) + D(s)) + N(s),$$

where $P(s)$ is the process transfer function, $U(s)$ is the Laplace transform of control signal u, $D(s)$ is the Laplace transform of load disturbance d, and $N(s)$ is the Laplace transform of measurement noise n. It is assumed that the

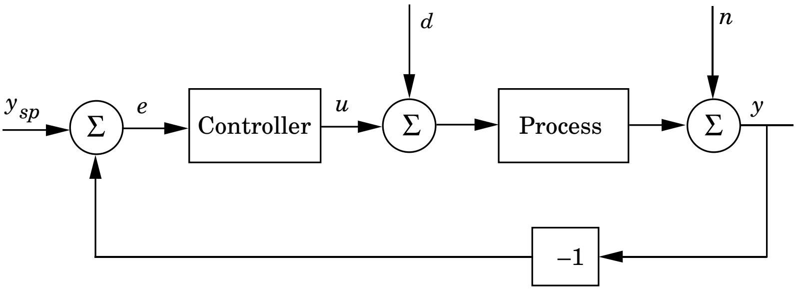

Figure 9.9 Block diagram of a simple feedback loop.

measurement noise only contains high frequencies and that these are filtered out by the filters in the controller. The noise term is therefore not considered in the sequel. The process output can be decomposed into two terms,

$$y(t) = y_u(t) + y_d(t), \tag{9.4}$$

where y_u is caused by the control signal and y_d is caused by the load disturbance.

In Equation 9.1, the prediction error in the least-squares estimator is given by

$$\varepsilon(t) = y(t) - \hat{y}(t) = y(t) - \varphi(t)^T \hat{\theta}(t-1). \tag{9.5}$$

The least-squares estimator tries to minimize $\varepsilon(t)$, i.e., to make the predicted process output $\hat{y}(t)$ equal to the true process output $y(t)$. It is implicitly assumed that

$$y(t) = y_u(t) = \varphi(t)^T \theta(t-1),$$

where $\theta(t)$ are the true process parameters. If this assumption is valid, i.e., if $y(t) = y_u(t)$, parameter estimates $\hat{\theta}(t)$ will converge to the true values $\theta(t)$, provided that the excitation is sufficient. However, if the process output is given by Equation 9.4, and if $y_d(t)$ has frequency components in the estimation region, the parameter estimates will not converge to their true values.

This is a very serious problem in process control applications. In process control, set-point changes are often performed only during production changes. (Exceptions are secondary controllers in cascade configurations.) This means that load disturbances often are the only excitation signals. For rule-based as well as model-based adaptive controllers there are possibilities to obtain useful information provided that the load disturbances come in the form of isolated transients. Such a solution will now be presented.

Figure 9.10 shows the different components of the process output after a step change in the load disturbance. Shortly after the load change, $y(t) \approx y_d(t)$, i.e., the changes in the process output are caused by load d only. After a while, the contribution from the control signal u is the dominating component.

A solution to the identification problem is to avoid adaptation during the first phase of the response, where y_d dominates over y_u. Adaptation should be initiated in the second phase where the major excitation in $y(t)$ is caused by the control signal.

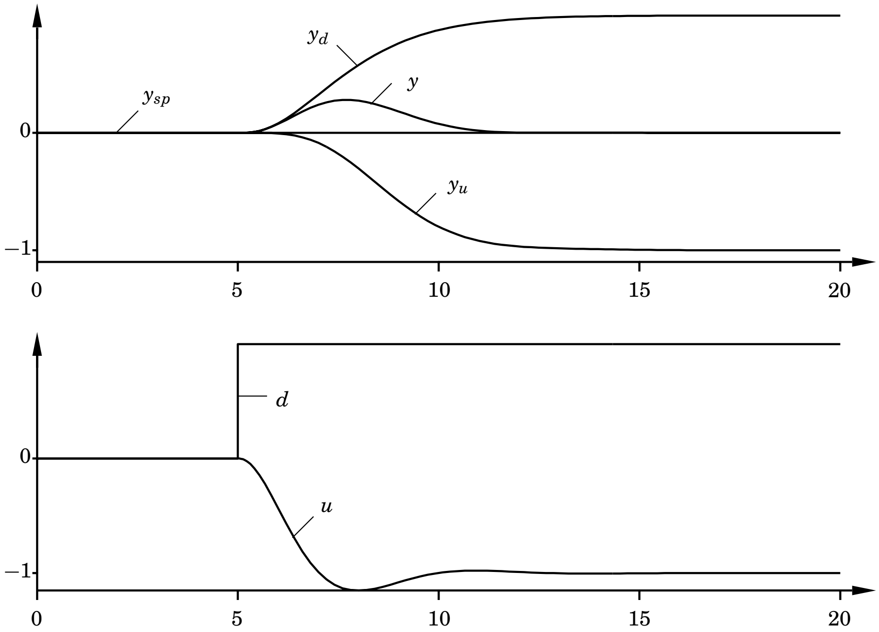

Figure 9.10 The upper diagram shows set point $y_{sp} = 0$, measurement signal y, its load component y_d, and its control signal component y_u. The lower diagram shows load disturbance d and control signal u.

To use this solution, a procedure that detects load disturbances is needed. This detection must be fast, so that adaptation is interrupted as quickly as possible. The detection can be done in the following way. First, the control signal is high-pass filtered in the same way as the measurement signal in Equation 9.3:

$$U_{hp} = \frac{s}{s + \omega_{hp}} U. \tag{9.6}$$

Figure 9.11 illustrates the same experiment as in Figure 9.8, but the high-pass filtered value of the control signal is also presented. In the following, it is assumed that the process has a positive static gain, i.e., $P(0) > 0$, and that all zeros are in the left-half plane. After a set-point change, both y_{hp} and u_{hp} then go in the same direction, whereas they go in opposite directions when a load disturbance occurs. This difference can be used to distinguish between set-point changes and load disturbances. In this way, it is possible to delay the adaptation and avoid adaptation during the first phase of the load disturbance response, and perform adaptation only during the second phase.

Another simpler way to avoid adaptation during the first phase of a load disturbance response can be obtained from the prediction error $\varepsilon(t)$; see Equation 9.5. The prediction error can be written as

$$\begin{aligned} \varepsilon(t) &= y(t) - \varphi(t)^T \hat{\theta}(t-1) \\ &= \big(y_u(t) - \varphi(t)^T \hat{\theta}(t-1)\big) + y_d(t). \end{aligned} \tag{9.7}$$

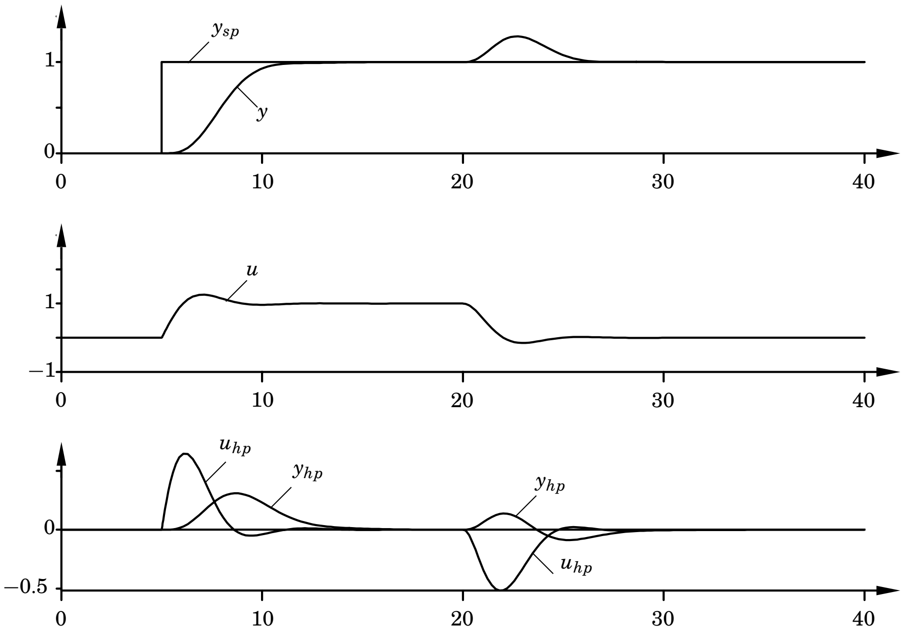

Figure 9.11 Excitation detection using high-pass filters. The figure shows responses to a set-point change at time t=5 and a load disturbance at time t=20.

Hence, $\varepsilon(t)$ consists of two terms, the true prediction error that we want to minimize and the load disturbance component of the process output. If we assume that process dynamics change slowly, the first term will remain bounded, and large load disturbances can be detected through the magnitude of $|\varepsilon(t)|$. If we restrict adaptation to those periods when $|\varepsilon(t)|$ is small, we will be able to track slow variations in the process, and we will also avoid adaptation when $|y_d(t)|$ becomes large.

Oscillation Detection

Oscillations with a high-frequency content near the ultimate frequency form an ideal excitation for adaptive control if they are caused by set-point variations or high controller gains resulting in small stability margins. In these cases, $y(t) = y_u(t)$.

Unfortunately, oscillations in control loops are normally generated by other sources. A common cause of oscillations is stick-slip motion because of valve friction. See Section 10.2. If no precautions are taken, the adaptive controller will interpret these oscillations as caused by a too high loop gain. This means that the controller will be detuned. This happens both for model-based and rule-based adaptive controllers. Stick-slip motion can be modeled as a load disturbance.

Another reason for oscillations in control loops might be that an external oscillating load disturbs the process. This disturbance may, e.g., be caused by a neighboring control loop with stick-slip motion.

In these cases, it is no longer true that $y(t) = y_u(t)$, but the load component

$y_d(t)$ dominates. This means that these disturbances will provide the process estimator with disinformation in the same way as the load disturbances discussed in the previous subsection. To avoid the problem, oscillations have to be detected in the same way as load disturbances, so that adaptation can be inhibited when these disturbances are present. Such detection procedures are presented in Section 10.4.

Signal Saturation

When the process output saturates, it is no longer true that $y(t) = y_u(t)$. Suppose that the process output becomes saturated at the limit y_{limit}. The process output can then described in the following way:

$$y(t) = y_u(t) + (y_{\text{limit}} - y_u(t)).$$

The second term on the right-hand side can be interpreted as a load disturbance component. Hence, we have the same problem as was discussed in the previous subsections. Therefore, the estimation should be interrupted when y saturates. Again, it is useful to have a timer connected to this interrupt, so that the estimation is kept off for a while after the saturation period. This is to avoid erroneous estimates during the transients.

It may also be desirable to interrupt adaptation when control signal u saturates. This might seem confusing, since $y(t) = y_u(t)$ in this case. However, if load disturbances are present it may no longer be true that $y_u(t)$ dominates over $y_d(t)$ in the second phase of a load disturbance response in this case.

Mode Transitions

A constant-parameter controller runs mainly in two modes:

- Manual mode
- Automatic mode

Bumpless transfer between the different operating modes is performed by ensuring that all states are assigned suitable values at the transitions. If this is not done properly, “bumps” will occur in the control signal at the mode transitions. See Section 13.5.

An adaptive controller has three modes of operation:

- Manual mode
- Automatic mode
- Adaptive mode

Here, it is important to ensure bumpless transfer also between the first two modes and the third adaptive mode. The parameter estimation is normally disconnected when the controller is in manual mode and often also when it is in automatic mode without adaptation. It is therefore important to initialize all the additional states that are given in the parameter estimator when the adaptive controller is started.

An erroneous initialization of the parameter estimator will result in “bumps” in the parameter estimates. These bumps are not always immediately visible as a “bump” in the control signal, but they may deteriorate the control in other ways since they provide an erroneous process model.

The most important states of the parameter estimator are given by Equations 9.1 and 9.2. The covariance matrix $P(t)$ should be assigned a large value when the parameter estimates are uncertain. However, when the controller parameters are initialized by an automatic tuning procedure or when they for other reasons are believed to be accurate, $P(t)$ should not be reinitialized to a large value but kept close to its stationary value.

The residual vector $\varphi(t)$ normally contains delayed control and measurement signals. This vector should be initialized by actual values of these signals.

It is also important that filters as well as the supervisory functions be provided with correct states. This can sometimes be accomplished by introducing a delay in the estimator. Suppose, e.g., that the controller is switched from manual to adaptive mode and that the process output is not close to the set point. This means that we immediately get a transient at the mode switch. If the excitation detection procedure is active, the adaptation mechanism may then start before the states have got their appropriate values. This problem can be avoided by delaying the excitation detection procedure at the mode transition.

A re-initialization of the adaptive controller must also be made if parameters related to the adaptation are changed. Suppose, e.g., that the sampling period is changed by the user or that an automatic tuning procedure is run, resulting in new values of the sampling period. This means that a total re-initialization of the adaptive controller must be made, with new filter-time constants, etc.

Another mode transition occurs if the adaptive controller is combined with gain scheduling. In this case, a re-initialization should be performed whenever there is a switch in the gain schedule.

It is also often possible to reset the adaptive controller, so that the parameter estimates $\hat{\theta}(t)$ are reinitialized to some pre-specified values, normally those obtained during the initialization phase.

Bounds on Parameter Estimates

There is a region in the parameter space where the information provided during the initialization phase is relevant. Inside this region, the a priori information about the process time T_p is correct, sampling periods and filter-time constants are suitable. If the process dynamics change so much that the parameter estimates tend to go outside this region, the behavior of the controller might be poor.

It is therefore advantageous to bound the parameter estimates to an allowable region. The adaptation may continue outside the region, but the algorithm should be reinitialized so that new parameters suitable for the new region are obtained. Using gain scheduling it is, e.g., possible to have several regions with different sampling periods and filter-time constants.

It may be difficult to find such regions if the estimated model is of high order. It is easier when the model order is lower, and it perhaps is possible to find physical interpretations of the parameters. In adaptive PID controllers, there are often bounds on the gain, integral time, and derivative time.

There is another reason for bounding the parameter estimates, which is related to the excitation needed for the parameter estimation. Suppose, e.g., that the parameter estimates change so much that a very low closed-loop bandwidth is obtained. The excitation in the interesting frequency band will then be low, and we will get a very slow adaptation.

It may also be advantageous to have bounds on the rate of estimate changes. This is done to decrease the effects of sudden outliers or other errors. This feature can be compared with the rate limiters that often are used in standard controllers.

9.7 Iterative Feedback Tuning

Iterative feedback tuning, IFT, is an iterative on-line method for adjusting controller parameters. The key idea is a clever way of computing the gradient of the controller error with respect to controller parameters.

Consider a standard system with error feedback. Assume that it is desired to minimize the loss function

$$J = \int_0^T f(y(t), u(t))dt$$

for a PID controller with the parametrization

$$C(s) = k + \frac{k_i}{s} + k_d s.$$

To minimize the criterion it is useful to know the gradient of the loss function with respect to the controller parameters. The partial derivative of J with respect to controller gain k is given by

$$\frac{\partial J}{\partial k} = \int_0^T \Big(\frac{\partial f(y(t), u(t))}{\partial y} \frac{\partial y}{\partial k} + \frac{\partial f(y(t), u(t))}{\partial u} \frac{\partial u}{\partial k} \Big) dt. \tag{9.8}$$

To evaluate the right-hand side we need the partial derivatives

$$y_k = \frac{\partial y}{\partial k}, \quad u_k = \frac{\partial u}{\partial k}.$$

They can conveniently be computed from the Laplace transforms. We have

$$\begin{aligned} Y_k &= \frac{\partial Y}{\partial C} \frac{\partial C}{\partial k} = \frac{\partial Y}{\partial C} \\ Y_{k_i} &= \frac{\partial Y}{\partial C} \frac{\partial C}{\partial k_i} = \frac{1}{s} \frac{\partial Y}{\partial C} \\ Y_{k_d} &= \frac{\partial Y}{\partial C} \frac{\partial C}{\partial k_d} = s \frac{\partial Y}{\partial C}. \end{aligned} \tag{9.9}$$

The process output is given by

$$\begin{aligned} Y &= \frac{PC}{1+PC}Y_{sp} + \frac{P}{1+PC}D + \frac{1}{1+PC}N \\ &= \Big(1 - \frac{1}{1+PC}\Big)Y_{sp} + \frac{P}{1+PC}D + \frac{1}{1+PC}N, \end{aligned}$$

and the control error is given by

$$E = Y_{sp} - Y = \frac{1}{1+PC}Y_{sp} - \frac{P}{1+PC}D - \frac{1}{1+PC}N.$$

Using this expression for the error we find

$$\begin{aligned} \frac{\partial Y}{\partial C} &= \frac{P}{(1+PC)^2}Y_{sp} - \frac{P^2}{(1+PC)^2}D - \frac{P}{(1+PC)^2}N \\ &= \frac{P}{1+PC}\Big(\frac{1}{1+PC}Y_{sp} - \frac{P}{1+PC}D - \frac{1}{1+PC}N\Big). \end{aligned}$$

Hence,

$$\frac{\partial Y}{\partial C} = \frac{P}{1+PC}E = \frac{1}{C}\frac{PC}{1+PC}E. \tag{9.10}$$

The partial derivatives of the output with respect to the controller parameters can be computed in a similar way. We have

$$\begin{aligned} U_k &= \frac{\partial U}{\partial C}\frac{\partial C}{\partial k} = \frac{\partial U}{\partial C} \\ U_{k_i} &= \frac{\partial U}{\partial C}\frac{\partial C}{\partial k_i} = \frac{1}{s}\frac{\partial U}{\partial C} \\ U_{k_d} &= \frac{\partial U}{\partial C}\frac{\partial C}{\partial k_d} = s\frac{\partial U}{\partial C}. \end{aligned} \tag{9.11}$$

Straightforward calculations show that the sensitivity derivative of the output is given by

$$\frac{\partial U}{\partial C} = \frac{1}{1+PC}E. \tag{9.12}$$

Equations 9.10 and 9.12 can be used to compute the sensitivity derivatives needed for the optimization. The error E is known, but there is a difficulty because the process transfer function P is not known. This difficulty can be circumvented in the following way:

- Make an experiment, and store the output y_1 and the control error signal e_1.
- Make a second experiment of the same duration where the set point is chosen as the control error e_1 from the first experiment. Store the output y_2 and the control error e_2 of this experiment.

The output and the control error of the second experiment are given by

$$\begin{aligned}
Y_2 &= \frac{PC}{1+PC}E_1 + \frac{P}{1+PC}D_2 + \frac{1}{1+PC}N_2\\
&= \frac{1}{C}\frac{\partial Y_1}{\partial C} + \frac{P}{1+PC}D_2 + \frac{1}{1+PC}N_2\\
E_2 &= \frac{1}{1+PC}E_1 - \frac{P}{1+PC}D_2 - \frac{1}{1+PC}N_2\\
&= \frac{\partial U_1}{\partial C} + \frac{P}{1+PC}D_2 + \frac{1}{1+PC}N_2.
\end{aligned}$$

The terms D_2 and N_2 are uncorrelated with E_1 if the experiments are well separated in time. Their effect can be made arbitrarily small by choosing long data sequences. Hence,

$$\begin{aligned}
\frac{\partial Y_1}{\partial C} &\approx CY_2\\
\frac{\partial U_1}{\partial C} &\approx E_2.
\end{aligned} \tag{9.13}$$

The second experiment thus gives an estimate of the sensitivity derivatives of the input and the output with respect to the controller parameters. Combining this with the input and the output from the first experiment we can now compute the gradient of the loss function with respect to the controller parameter from (9.8). The controller parameters can then be adjusted recursively. Summarizing we obtain the following algorithm.

ALGORITHM 9.1—ITERATIVE FEEDBACK TUNING

1. Make an experiment of fixed duration, and store the output y_1 and the control error signal e_1.
2. Make a second experiment of the same duration where the set point is chosen as the control error e_1 from the first experiment. Store the output y_2 and the control error e_2 of this experiment.
3. Compute the gradient of the loss function from Equations 9.8, 9.9, 9.11, and 9.13.
4. Modify the controller parameters using the gradient.
5. Repeat from 1 until the gradient is sufficiently small.

□

The same idea can be applied to a controller with two degrees of freedom but a third experiment is then required. A nice property of iterative feedback tuning is that it can be used for many different controllers and criteria. It is particularly well suited to optimization with respect to stationary stochastic disturbances.

9.8 Commercial Products

To illustrate how adaptive techniques are used industrially we present some features of industrial controllers. Rather than to give an exhaustive presentation we have selected a few products to show the wide range of techniques, and we have chosen products that have a good track record. We have also selected products where reasonably detailed descriptions are published; more products are described in the book [Van Doren, 2003] and in reviews in trade journals.

Foxboro EXACT$^{\text{TM}}$(760/761)

Foxboro was one of the first companies to announce products using adaptive techniques. The single-loop controller Foxboro EXACT$^{\text{TM}}$(760/761) , which used adaptation based on pattern recognition , was released by Foxboro in October 1984. The controller was later augmented with more features and Foxboro has continued to expand their use of adaptation in a range of products including their DCS system Foxboro I/A$^{\text{TM}}$. The ideas are described in [Bristol *et al.*, 1970] and [Bristol, 1977] and details about the system are found in [Bristol and Kraus, 1984] and [Bristol, 1986]. Foxboro continued the development of adaptation, and auto-tuning and adaptation are now available in their distributed control system under the trade name Exact MV$^{\text{TM}}$. A presentation of the details of the system are found in [Hansen, 2003]. Three function blocks, PIDA, FBTUNE, and FFTUNE, are used to implement the controller. PIDA is an advanced PID controller, FBTUNE, which handles tuning of the feedback gains, has functions for pretuning and adaptation, and FFTUNE has functions for tuning of feedforward gains and gain scheduling.

Controller Structure Foxboro uses a controller structure where integral action is implemented with positive feedback around a lag as illustrated in Figure 3.3. This implementation gives a controller in series form, see (3.8). A controller with a special structure called PIDτ is also available in the system. This controller is a PID controller where the integral action is implemented with positive feedback around a lag with a time delay as shown in Figure 8.10. This arrangement gives a controller with more phase lead than an ordinary PID controller. Since phase lead is also associated with high gain it is necessary to provide good filtering if there is measurement noise. The controller can be interpreted as a controller where the future output is predicted with a combination of past controller inputs and controller outputs, see the discussion of the PPI controller in Section 8.4. The controller PIDτ can also be regarded as a special form of a Smith predictor. The controller PIDτ gives significant improvement of performance for lag-dominated processes but it requires careful tuning.

Pattern Recognition Adaptation based on pattern recognition can be viewed as an automation of the procedure used by an experienced process engineer when he tunes a controller. The following description follows the presentation in [Bristol and Kraus, 1984]. The control error after process perturbations are analyzed and the controller parameters are modified. If the controller parameters are reasonable, a transient error response of the type shown in Figure 9.12

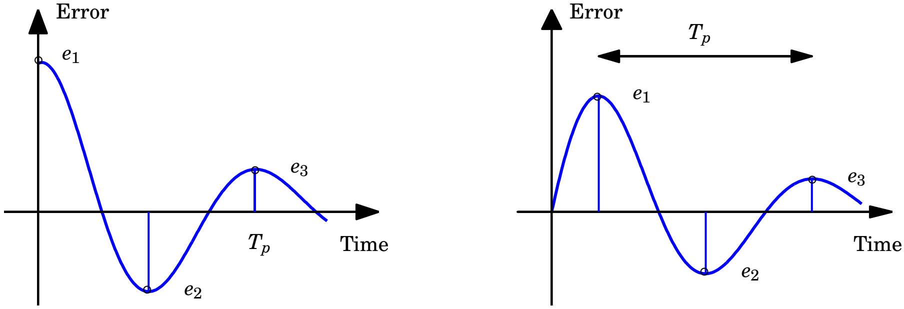

Figure 9.12 Response in control error to a step change of set point (left curve) and load (right curve).

is obtained. Heuristic rules are used to detect that a proper disturbance has occurred and to detect peaks e_1, e_2, e_3, and oscillation period T_p. Heuristics are also used to change the controller parameters if the response is overdamped. The transient is characterized quantitatively in terms of two parameters, overshoot (o) and damping (D), which are defined as

$$o = \left|\frac{e_2}{e_1}\right|, \qquad D = \frac{e_3 - e_2}{e_1 - e_2}, \tag{9.14}$$

where e_1, e_2, and e_3 are the peaks of the transients shown in Figure 9.12. Note that the definition of damping used is equal to the square root of the decay ratio (2.49). Quarter amplitude damping thus corresponds to $D = 0.5$.

The controller parameters are adjusted using heuristic rules to obtain desired damping and overshoot. Some rules are discussed in Section 6.3, and the effect of controller parameters on the transient are illustrated in the tuning map in Figure 6.7.

Pre-Tuning is Foxboro's notation for auto-tuning. The controller has a set of parameters that must be given either by the user from prior knowledge of the loop or estimated using the pre-tune function.

- Initial values of PB, T_i', and T_d'.
- Noise band *(NB)*. The controller starts adaptation whenever the error signal exceeds two times NB.
- Maximum wait time ($W_{\max}$). The controller waits for a time of $W_{\max}$ for the occurrence of the second peak.

If the user is unable to provide the required parameters, a pre-tune function that estimates these quantities can be activated. To activate the pre-tune function, the controller must first be put in manual. When the pre-tune function is activated, a step input is generated. The process parameters static gain K_p, dead time L, and time constant T are then obtained from a simple analysis of the process reaction curve. The controller parameters are calculated using a Ziegler-Nichols-like formula:

$$PB = 120K_pL/T, \quad T_i' = 1.5L, \quad T_d' = T_i/6. \tag{9.15}$$

Maximum wait time, $W_{\max}$, is also determined from the step response by $W_{\max} = 5L$.

The noise band is determined during the last phase of the pre-tune mode. The control signal is first returned to the level before the step change. With the controller still in manual and the control signal held constant, the output is passed through a high-pass filter. The noise band is calculated as an estimate of the peak-to-peak amplitude of the output from the high-pass filter. The estimated noise band is also used to adjust derivative action.

There are a number of optional parameters. If these are not supplied by the user then the default values will be used. The optional parameters are as follows (default values in parenthesis):

- Maximum allowed damping (0.3)
- Maximum allowed overshoot (0.5)
- Derivative factor (1). The derivative term is multiplied by the derivative factor. This allows the derivative influence to be adjusted by the user. Setting the derivative factor to zero results in PI control.
- Change Limit (10). This factor limits the controller parameters to a certain range. Thus, the controller will not set the PB, T_i' and T_d' values higher than ten times or lower than one tenth of their initial values if the default of 10 is used for the change limit.

The pretuning has been improved in the later Foxboro products. The process is excited with a doublet pulse, see Figure 2.33, instead of the step used in the original system. An FOTD or SOTD model is fitted to the data from the experiment, as described in [Shinskey, 1994]. The controller parameters are calculated from the model based on a novel robust analytic design method described in [Hansen, 2003]. Adaptation of feedback gain parameters are still done using the pattern recognition method.

The controllers in the Foxboro DCS system have lead-lag filters for feedforward from measured disturbances. The feedforward gains are tuned by fitting a low-order continuous-time model using the method of moments, see [Hansen, 2003].

ABB

Adaptation in the ABB's systems has its roots in a auto-tuners based on relay feedback first developed by the company NAF in the early 1980's. The first system was part of a small (about 50 loops) DCS system called SDM-20$^{\text{TM}}$ introduced in 1982 and a single-loop controller ECA-40$^{\text{TM}}$ introduced in 1986. These systems also used auto-tuning to build gain schedules. The company NAF went through a series of acquisitions, SattControl, Alfa Laval Automation, and is now part of ABB. The adaptive techniques were developed by adding continuous adaptation, adaptation of feedback and feedforward gains, and diagnostics. These features were all introduced in the ECA600^{TM} which was announced in 1988. The technology is now an integral part of the ABB DCS system Industrial IT System 800xA^{TM}, which also has facilities for fuzzy control and for model predictive control. There are several types of PID controllers; the advanced versions give access to more parameters. There is also a PPI controller for systems with time delay.

Essential parts of the technology are described in [Åström and Hägglund, 1984c; Åström and Hägglund, 1984a; Åström and Hägglund, 1988; Åström and Hägglund, 1990; Åström and Hägglund, 1995a; Åström and Hägglund, 1995b; Hägglund, 1999; Hägglund and Åström, 2000; ABB, 2002].

Bi-directional Data Flow Distributed control systems are traditionally programmed graphically using a block oriented language. One drawback with traditional systems is that data-flow is unidirectional. This leads to unpredictable latency in the system, which is particularly noticeable in large systems and when back calculations to avoid windup are propagated through several loops. An interesting novel feature of the ABB System 800xA is a data structure called *control connection* that permits bi-directional data flow between the control modules. This feature makes it possible to implement windup protection in an elegant way which avoids latency even in complex systems with many cascades.

Relay Auto-Tuning The auto-tuning is performed using the relay method discussed in Section 2.7. The tuner is typically operated as follows. The process is brought to a desired operating point, either by the operator in manual mode or by a previously tuned controller in automatic mode. When the loop is stationary, the operator presses a tuning button. After a short period, when the noise level is measured automatically, a relay with hysteresis is introduced in the loop, and the PID controller is temporarily disconnected (see Figure 9.4). The width of the hysteresis is set automatically, based on measurement of the noise level in the process. The lower the noise level, the lower the amplitude required from the measured signal. The relay amplitude is controlled so that the oscillation is kept at a minimum level above the noise level. When an oscillation with constant amplitude and period is obtained, the relay experiment is interrupted and $P(i\omega_0)$, i.e., the value of the transfer function P at oscillation frequency ω_0, is calculated using describing function analysis.

Control Structures and Controller Design Several PID and PPI controllers are available in the ABB systems. The advanced versions give the user access to many parameters. The PID algorithm in the ECA600^{TM} controller is of series form, the controllers in the DCS system use the parallel form.

The identification procedure provides a process model in terms of one point $P(i\omega_0)$ on the Nyquist curve. There is also a test to determine if process dynamics is lag dominated. The frequency ω_0 depends on the hysteresis in the relay. It is typically less than ω_{180}, which is advantageous; see Section 7.5. By introducing the PID controller $C(i\omega)$ in the control loop, it is possible to move the point corresponding to ω_0 on the Nyquist curve of the loop transfer to a desired location. In the normal case the desired point is

$$P(i\omega_0)C(i\omega_0) = 0.5e^{-i135\pi/180}. \tag{9.16}$$

Since there are three parameters, K, T_i, and T_d, and the design criterion (9.16) only specifies two parameters the additional constraint

$$T_i' = 4T_d'. \tag{9.17}$$

is introduced.

The normal procedure can give very high gains for lag-dominated systems. If this is detected a PI controller with the conservative tuning

$$K = 0.5/|P(i\omega_0)|, \quad T_i = 4/\omega_0 \tag{9.18}$$

is used. A different tuning can also be used for processes which are delay dominated. A PI controller with the tuning rule

$$K = 0.25/|P(i\omega_0)|, \quad T_i = 1.6/\omega_0 \tag{9.19}$$

is then used.

In the early versions of the controller (ECA-40$^{\mathrm{TM}}$ and ECA-600$^{\mathrm{TM}}$) the user can influence tuning by selecting normal/PI/time-delay. Later versions of the controller obtain more information about the process by making a step change after the relay experiment. This gives the static gain of the process and gain ratio κ, and tuning can be improved without any user interaction. In the ABB 800xA$^{\mathrm{TM}}$ system this is accomplished by using the tuning rules in [Åström and Hägglund, 1995a]. Further improvements are possible by using the results in Chapter 7.

Gain Scheduling Gain scheduling was introduced in the early controllers SDM-20$^{\mathrm{TM}}$ and ECA-40$^{\mathrm{TM}}$. It was very easy to build the schedules by using auto-tuning and the feature was well accepted by the users. Gain scheduling therefore became a standard feature of almost all controllers. The users can select the scheduling variable as the control signal, the measured process output, or an external signal. It is important that the scheduling variables do not change too quickly, filtering and hysteresis are used for signals like the process output that can change rapidly. Three sets of parameter values were available in the early systems but larger tables can be used in the later versions. The parameters are obtained by using the auto-tuner once at every operating condition. The ranges of the scheduling variable where different parameters are used can also be given by the user.

Adaptive Feedback Information from the relay feedback experiment is used to initialize the adaptive controller. Figure 9.7 shows the principle of the adaptive controller. The key idea is to track the point on the Nyquist curve obtained by the relay auto-tuner. It is performed in the following way. The control signal u and the measurement signal y are filtered through narrow band-pass filters centered at frequency ω_0. This frequency is obtained from the relay experiment. The signals are then analyzed in a least-squares estimator, which provides an estimate of the point $P(i\omega_0)$.

Adaptive Feedforward Feedforward from measured disturbances can frequently improve performance significantly. Adaptive feedforward has been a feature in all controllers starting with the ECA400$^{\mathrm{TM}}$. Diagnostics for on-line assessment of the potential value of feedforward is an active research topic; see [Petersson *et al.*, 2001], [Petersson *et al.*, 2002], and [Petersson *et al.*, 2003].

The adaptive feedforward control is based on the simple model

$$y(t) = au(t-4h) + bv(t-4h), \tag{9.20}$$

where y is the measurement signal, u is the control signal and v is the disturbance signal that should be fed forward. The sampling interval h is determined from the relay experiment as $h = T_0/8$, where T_0 is the oscillation period. The parameters a and b are estimated recursively by a least-squares algorithm. The feedforward compensator has the simple structure

$$\Delta u_{ff}(t) = k_{ff}(t)\Delta v(t), \tag{9.21}$$

where the feedforward gain k_{ff} is calculated from the estimated process parameters

$$k_{ff}(t) = -0.8\,\frac{\hat{b}(t)}{\hat{a}(t)}. \tag{9.22}$$

The Man/Machine Interface The auto-tuners based on relay feedback can be implemented with very simple man-machine interfaces. In many cases it is sufficient to provide the controllers with just one button to initiate tuning. Gain scheduling can also be implemented in a very user-friendly fashion. Many of the problems normally associated with implementation of adaptive controllers can be avoided because the auto-tuner gives good initial values.

Industrial experience has also indicated that there is a significant advantage to combine adaptation with diagnostics and supervision. For example, it is meaningless to tune a controller if there is a bad actuator in the loop.

Emerson Process Management

The adaptive techniques used in Emerson's systems go back to the DCS systems Provox$^{\mathrm{TM}}$ and RS3$^{\mathrm{TM}}$, where the Fisher-Rosemount Intelligent Tuner and Gain Scheduler were introduced. Use of adaptation has been expanded in the Delta V system. Fairly detailed information about the techniques used is available in the book [Blevins *et al.*, 2003], which also contains many references. The system has facilities for auto-tuning, gain scheduling, and adaptation. There is also software for fuzzy control and for model predictive control.

The automatic tuning is based on relay feedback. The range of the relay oscillation is typically a few percent of the full signal range. An estimate of the apparent time delay is obtained by analysing the initial portion of the first step. When an estimate of the time delay is available it is also possible to obtain an FOTD model. The parameters K_p, T, and L of the FOTD model can be displayed. Tuning is typically accomplished in a few periods of the oscillation. Since a FOTD model is available it is possible to use several tuning techniques. The available options include Ziegler-Nichols tuning, IMC tuning, and Lambda tuning to mention a few. The system is structured so that an inexperienced user has few choices, but an experienced user has many options. There is also a built-in simulator so that tuning can be tested against the process model before committing it to the process.

Adaptive control is based on data from the process during normal operation; excitation can also be provided. The system consists of a supervisor, an excitation generator, adaptors for gain, integration time, and derivative time, and a safety net. The goal of the adaptation is to obtain a well damped slightly oscillatory response. The approach is similar to that used in the Foxboro Exact.

Gain scheduling is done by estimating static process characteristics. Interpolation is done using fuzzy techniques.

Honeywell

Honeywell products using adaptive control started with the single-loop controller UDC 6000$^{\text{TM}}$ which had an adaptive function called Accutune. The adaptive techniques were developed further and they are essential components of Honeywells DCS system TDC 3000$^{\text{TM}}$.

UDC 6000$^{\text{TM}}$ Adaptation in the UDC 6000$^{\text{TM}}$ combines model-based procedures and rule-based procedures. Modeling is based on a step-response experiment. The user brings the process variable to a point some distance away from the desired set point in manual and waits for steady state. Switching to tuning mode initiates an open-loop step response experiment, where the size of the step is calculated to be so large that it is supposed to take the process variable to the set point.

During the experiment, the process variable and its derivative are continuously monitored. Dead time L is calculated as the time interval between the step change and the moment the process variable crosses a certain small limit.

If the derivative of the process variable continuously decreases from the start, it is concluded that the process is of first order and an FOTD model is determined from a few points on the step response. The calculations can be performed before the steady-state is reached, and it is claimed that the process is identified in a time less than one third of the time constant.

If the derivative of the process variable increases to a maximum and then decreases, the process is identified as a second-order process and an SOTD model is determined from the step response. The controller is then switched to automatic mode and controlled to the set point using preliminary controller parameters when the maximum slope of the process output has been reached, bu it is necessary to wait for steady state to obtain the complete model. More details about the modeling procedure are given in [Åström and Hägglund, 1995b].

When the model has been obtained the controller parameters are calculated from the model and the controller is switched to automatic control mode.

The controller used is on series form with the transfer function

$$C(s) = K\frac{(1+sT_i')(1+sT_d')}{sT_i'(1+0.125sT_d')}.$$

Notice that the filter time constant is 1/8th of the derivative time. Controller design is based on pole placement of the Dahlin-Higham type procedure where the process poles are cancelled. There are several different versions depending on system order and time delay. For systems with time delay the closed loop time constant is chosen as $T_{cl} = L + T/3$. The UDC 6000 controller also has continuous adaptation which is activated when the process variable changes more than 0.3 percent from the set point or if the set point changes more than a prescribed value.

Honeywell LOOPTUNE$^{\text{TM}}$ The DCS system TDC3000 has a wide range of controllers; Basic Controllers, Extended Controllers, Multifunction Controllers, Process Managers and Application Modules. LOOPTUNE is a software package in the system that tunes loops with PID controllers.

The tuning algorithm does not rely on any particular model of the system. Performance is evaluated using the quadratic loss function

$$J = \frac{1}{N}\sum_{t=1}^{N}\big((1-\rho)(y(t) - y_{sp}(t))^2 + \rho(u(t) - u(t-1))^2\big), \tag{9.23}$$

where N is the evaluation horizon and ρ a weighting factor that balances control error against actuator changes. The controller parameters are changed one at a time and the loss function is evaluated over a given time horizon N. A large value of N is required to obtain a reliable estimate but the evaluation takes long time. Process knowledge can be used to improve the search for good controller parameters by biasing the search towards higher controller gain and lower integration time.

Yokogawa SLPC-181, 281

The Yokogawa SLPC-181 and 281 both use a process model as a first-order system with dead time for calculating the PID parameters. A nonlinear programming technique is used to obtain the model. The PID parameters are calculated from equations developed from extensive simulations. The exact equations are not published.

Two different controller structures are used.

$$\begin{aligned} &1: \quad u = K\left(-y + \frac{1}{T_i}\int e dt - T_d\frac{dy_f}{dt}\right) \\ &2: \quad u = K\left(e + \frac{1}{T_i}\int e dt - T_d\frac{dy_f}{dt}\right) \end{aligned} \tag{9.24}$$

where y_f is generated by filtering y with a first order filter having time constant T_d/N. The first structure is recommended if load disturbance rejection is most important, and structure 2 if set-point responses are most important. The set point can also be passed through two filters in series:

$$\text{Filter 1: } \frac{1+\alpha_i sT_i}{1+sT_i} \qquad \text{Filter 2: } \frac{1+\alpha_d sT_d}{1+sT_d} \tag{9.25}$$

where α_i and α_d are parameters set by the user, mainly to adjust the overshoot of the set-point response. The effects of these two filters are essentially

Table 9.2 Set-point response specifications used in the Yokogawa SLPC-181 and 281.

Type	Features	Criteria
1	no overshoot	no overshoot
2	5% overshoot	ITAE minimum
3	10% overshoot	IAE minimum
4	15% overshoot	ISE minimum

equivalent to set-point weighting. It can be shown that $\alpha_i = b$, where b is the set-point weighting factor.

The user specifies the type of set-point response performance according to Table 9.2. A high overshoot will, of course, yield a faster response. The controller has four adaptive modes:

Auto mode. The adaptive control is on. PID parameters are automatically updated.

Monitoring mode. In this mode, the computed model and the PID parameters are only displayed. This mode is useful for validating the adaptive function or checking the process dynamics variations during operation.

Auto startup mode. This is used to compute the initial PID parameters. An open loop step response is used to estimate the model.

On-demand mode. This mode is used to make a set-point change. When the on-demand tuning is requested, a step change is applied to the process input in closed loop. The controller estimates the process model using the subsequent closed-loop response.

The controller constantly monitors the performance of the system by computing the ratio of the variances of process output and model output. This ratio is expected to be about 1. If it is greater than 2 or less than 0.5, a warning message for retuning of the controller is given. Dead time and feedforward compensation are available for the constant gain controller, but they are not recommended by the manufacturer to be used in conjunction with adaptation.

Techmation Protuner

The Protuner is a process analyzer from Techmation Inc. It consists of a software package for personal computers and an interface module with cables to be connected to the process output and the control signal of the control loop to be analyzed. The Protuner monitors a step-response experiment, calculates the frequency response of the process based on the experimental data, and suggests controller parameters based on several methods for controller tuning.

Prior Information Before the process analysis is performed, the user must provide some information about the process and the controller. This is done using a couple of “Set-up menus.” The following process information must be given:

- The ranges of the control and the measurement signals.

- It must be determined if the process is stable or if it has integral action.

To be able to set relevant controller parameters, the following data about the controller must be provided:

- P-type (gain or proportional band)
- I-type (seconds, seconds/repeat, minutes, or minutes/repeat)
- Controller structure (ideal, series, or parallel)
- Sampling rate
- Filter time constant (if there is a low-pass filter connected to the measurement signal).

Before the tuning experiment can be performed, the user must also specify a sample time. This is the time during which data will be collected during the experiment. It is important to choose the sample time long enough, so that the step response settles before the sample time has ended. In case of an open-loop experiment of an integrating process, the response must reach a constant rate of change when the experiment ends.

Determining the Process Model The tuning procedure is based on a step-response experiment. It can be performed either in open or closed loop. The open-loop experiment is recommended. When the user gives a start command, the process output and the control signal are displayed on the screen, with a time axis that is given by the sample time defined by the user. The user then makes a step change in the control signal. If the experiment is performed in closed loop a step is instead introduced in the set point.

There are several facilities for editing the data obtained from the step-response experiment. Outliers can be removed, and data can be filtered. These features are very useful because they make it possible to overcome problems that are often encountered when making experiments on industrial processes.

When the data has been edited the Protuner calculates the frequency response of the process. The result can be displayed in a Bode diagram, a Nyquist diagram, or a Nichols diagram. The static gain, the dominant time constant, and the apparent dead time are also displayed, as well as the ultimate gain and the ultimate period.

Design Calculations The controller parameters are calculated from the frequency response. A special technique is used. This is based on cancellation of process poles by controller zeros. The integral time and the derivative time are first determined to perform this cancellation. The gain is then determined to meet predetermined gain and phase margins.

The Protuner provides several design options. Controller parameters are given for the following closed-loop responses slow (critically damped), medium (slightly underdamped) and fast (decay ratio 0.38) responses. The different design options are obtained by specifying different values of the gain and the phase margins. The Protuner provides different controller parameters depending on whether set point or load disturbances are considered. Both P, PI, and

PID controller parameters are provided. The set-point weightings for proportional and derivative action and the high-frequency gain at the derivative part must be supplied by the user.

Evaluation It is possible to evaluate the performance of the closed-loop system in several ways. The combined frequency response, i.e., the frequency response of the loop transfer function $G_l(i\omega) = P(i\omega)C(i\omega)$, can be plotted in a Bode diagram, a Nyquist diagram, or a Nichols diagram. In this way, the phase and amplitude margins or the M_s value can be checked.

The Protuner also has a simulation facility. It is possible to simulate the closed-loop response of the process and the suggested controller. To do this, it is necessary to provide some additional controller parameters, namely, set-point weightings b and c, and derivative gain limitation factor N. Using the simulation facility, it is also possible to investigate the effects of noise and to design filters to reduce these effects.

Some Personal Reflections

Adaptive techniques have been used extensively in industry since the mid 1980s. The techniques are proven useful and the products continue to develop, but there is clearly a potential to improve current products.

Several lessons can be learned from the results of Chapter 7. One observation is that it is useful to characterize process dynamics with three parameters. Dynamics can then be classified as delay dominant, balanced, or lag dominant. Tight control can be obtained by using an FOTD model for systems with balanced and delay-dominated dynamics but control performance can be improved significantly for lag-dominated systems by using a better model. In Section 2.7 it was also shown that it is difficult to obtain an SOTD model from a step response experiment. Hence, it is not possible to design auto-tuners for tight control based on a step response or on knowledge of ultimate gain and ultimate frequency. An indication of this is that Foxboro switched to using a doublet instead of a step. The doublet can actually be regarded as a short version of an experiment with relay feedback.

It is highly desirable to accomplish tuning in a short time, as is illustrated by the Honeywell UDC 6000$^{\mathrm{TM}}$. One advantage of the relay auto-tuner is that tuning often is accomplished in a time that is much shorter than the average settling time of the system. In particular, the time can be much shorter for systems with lag-dominated dynamics. An interesting question is therefore what information can be derived from an experiment with relay feedback. The Emerson experience indicates that at least an FOTD model can be determined from an experiment with relay feedback. To explore in detail the information that can be deduced from a relay experiment is therefore an interesting and useful research task. If improved models are obtained it is also possible to use the algorithms presented in Chapter 7 that give tight control. The potential gains are particularly large for lag-dominated process.

The model-free approaches to adaptive control have many attractive features but their main disadvantage is that tuning takes a long time. Tuning can be made more effective by using iterative feedback tuning, which also computes estimates of the gradient of the loss function; see Section 9.7. Adaptive controllers based on more elaborate models like the self-tuning controller discussed in Section 9.3 is an alternative to model based control. Such controllers work very well but so far they have required very knowledgeable users. There may be a possibility to make them simpler to use by exploiting the information obtained from automatic tuning. This may also be the road to introduce adaptation in the model predictive controllers.

Tools like Techmations Protuner, which permit simulation of a process with different controller settings, are very useful for the advanced user. Many components to build such a system are already available in current DCS systems. It is therefore natural to provide such tools as an integral part of the systems.

9.9 Summary

An essential feature of feedback is that it can be used to design systems that are insensitive to process variations. When there are large variations, performance may be improved by adjusting the controller parameters. Adaptive techniques are therefore increasingly being used in PID controllers to adapt the controller parameters to the changes in process dynamics or disturbances. In this chapter we have given a broad presentation of a variety of adaptive methods covering automatic tuning, gain scheduling, and continuous adaptation. The techniques are used in several ways.

In automatic tuning the controller parameters are adjusted on demand from the user. Gain scheduling can be used when there is a measured scheduling variable that correlate well with the process changes. The controller parameters are obtained from a table, which gives controller parameters as a function of the scheduling variable. Auto-tuning can be used to build the table. Adaptive control can be used when a scheduling variable is not available.

Model based and feature based methods are discussed, particular attention is given to use of relay feedback for auto-tuning, parameter estimation, and iterative feedback tuning.

The adaptive controller derives the knowledge required from the input and output of the process. Adaptive control is less robust than gain scheduling and it requires supervisory functions. Supervision of adaptive controllers is therefore discussed.

A short presentation of some industrial adaptive controller where adaptive methods have been used successfully are also discussed.

9.10 Notes and References

Controllers with automatic tuning grew out of research on adaptive control. Overviews of adaptive techniques are found in [Dumont, 1986; Åström, 1987a; Bristol, 1970; Åström, 1990]. More detailed treatments are found in the books [Harris and Billings, 1981; Hang *et al.*, 1993b; Åström and Wittenmark, 1995]. Overviews of different approaches and different products are found in [Isermann, 1982; Gawthrop, 1986; Kaya and Titus, 1988; Morris, 1987; Yamamoto, 1991; Åström *et al.*, 1993].

Many different approaches are used in the automatic tuners. The systems described in [Nishikawa *et al.*, 1984; Kraus and Myron, 1984; Takatsu *et al.*, 1991] are based on transient response techniques. The paper [Hang and Sin, 1991] is based on cross correlation. The use of orthonormal series representation of the step response of the system is proposed in [Zervos *et al.*, 1988; Huzmezan *et al.*, 2003]. Pattern recognition, which was the basis for Foxboros EXACT^{TM} controller, is discussed in [Bristol, 1967; Bristol, 1970; Bristol *et al.*, 1970; Bristol, 1977; Bristol and Kraus, 1984; Bristol, 1986; Porter *et al.*, 1987; Anderson *et al.*, 1988; Klein *et al.*, 1991; Pagano, 1991; Swiniarski, 1991]. Auto-tuning based on relay feedback is treated in [Åström and Hägglund, 1984b; Åström and Hägglund, 1988; Hägglund and Åström, 1991; Schei, 1992; Hang *et al.*, 1993a; Leva, 1993; Schei, 1994; Voda and Landau, 1995]. Iterative feedback tuning is discussed in [Hjalmarsson *et al.*, 1998]. It is more effective than direct search because gradient information is used.

Traditional adaptive techniques based on system identification and control design have also been applied to PID control. Identification is often based on estimation of parameters in a transfer function model. Examples of this approach are given in [Hawk, 1983; Hoopes *et al.*, 1983; Yarber, 1984a; Yarber, 1984b; Cameron and Seborg, 1983]. There are also systems where the controller is updated directly as in [Radke and Isermann, 1987; Marsik and Strejc, 1989; Rad and Gawthrop, 1991]. Supervision of adaptive controllers is discussed in [Isermann and Lachmann, 1985; Sullivan, 1996; Clarke and Hinton, 1997; Liu, 1998; Hägglund and Åström, 2000].

An overview of several products that use adaptation is given by [Van Doren, 2003]. Several tuning aids are implemented in hand-held computers or as software in PCs where the user is entering the process information through a keyboard; see [Blickley, 1988; Tyreus, 1987; Yamamoto, 1991].

The papers [McMillan *et al.*, 1993b; McMillan *et al.*, 1993a] describe the Fisher Rosemount products for tuning and gain scheduling. The implementations in the Delta V^{TM} DCS system is described in the book [Blevins *et al.*, 2003]. The Yokogawa systems are discussed in [Takatsu *et al.*, 1991] and [Yamamoto, 1991].

There have been comparisons of different auto-tuners and adaptive controllers, but few results from those studies have reached the public domain. Some papers that deal with the issue are [Nachtigal, 1986a; Nachtigal, 1986b; Dumont, 1986; Dumont *et al.*, 1989]. Some operational experience is described in [Higham, 1985; Callaghan *et al.*, 1986].