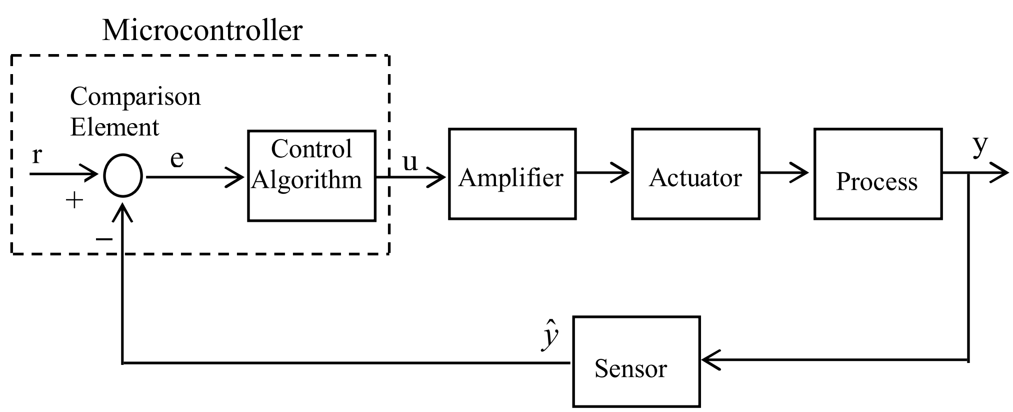
Section 6 Digital Control of Dynamic Systems

6.1 Introduction

Closed-loop process control systems combine the sensors, actuators and mathematical models we have seen in the previous chapters to produce mechatronic products with superior speed and performance. Today the vast majority of process control systems are implemented using digital microprocessors instead of the analog circuitry (and sometimes mechanical means) used in the past. The primary reason for this is the reduced cost. A secondary reason is the programmable nature of the microprocessors makes it possible to implement a wide variety of control algorithms. This greatly increases the design flexibility compared with the prior approaches. In this chapter we will look at ON-OFF, PID and model-based control algorithms. These are used extensively in industrial applications. The term "process" refers to whatever dynamic system we are trying to control. Examples include: the arm of a robot, the engine of your car, and the air in your house.

6.2 Closed-Loop Control System Structure

In this course, we will focus on the unity feedback structure of closed-loop control since it is the most common. This structure is shown in the figure below.



where:

- $r \equiv$ reference input (or setpoint)
- $e \equiv$ error signal
- $u \equiv$ control output
- $y \equiv$ process output
- $\hat{y} \equiv$ measured output

In this chapter we will assume we have a perfect sensor such that $\hat{y} = y$. In reality any sensor inaccuracy will degrade the performance of the control system.

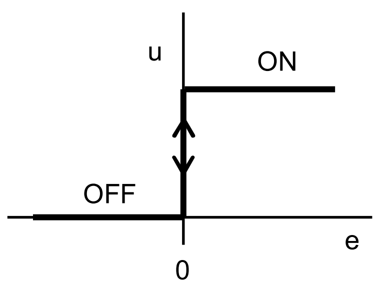
The objectives of a control system may be separated into:

1) **Regulating:** Keeping the process output close to a constant value of the reference input in the presence of external disturbances, *e.g.* temperature control of the air in a room after someone opens the door.
   Such a controller is called a "regulator".
2) **Servoing:** Making the process output respond to changes in the reference input in a specific way, *e.g.* controlling the path of a robot for welding, controlling the table motion of a milling machine. This type of controller is called a "servo".

6.3 Control Algorithms

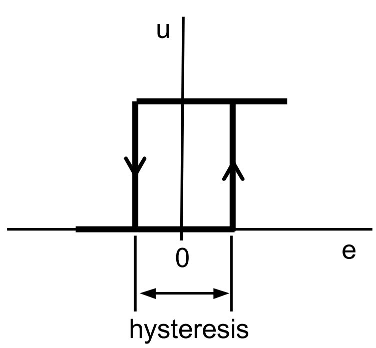
6.3.1 ON-OFF Control

The control output is binary and switches either on or off based on the error signal. The discontinuous control action always produces oscillations in the output.



With "simple ON-OFF control" the control output switches as shown in the figure below.

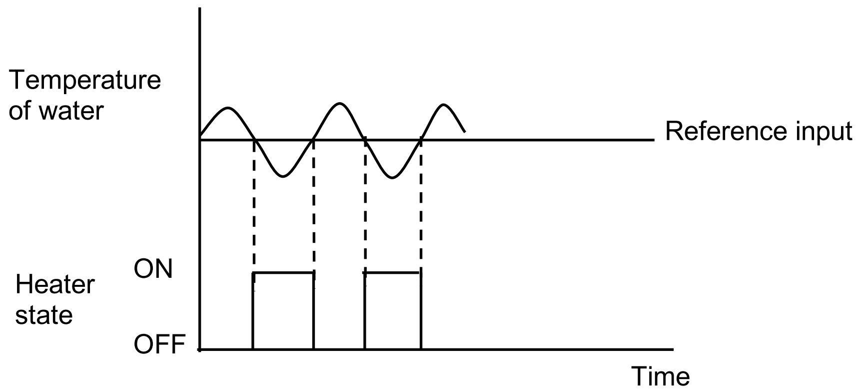
The problem with simple ON-OFF control is that the rapid switching of the actuator as the process output oscillates about the set point causes rapid actuator wear and fatigue. The solution is to include hystereris. The *u* vs. *e* graph for "ON-OFF control with hysteresis" is shown in the figure below.



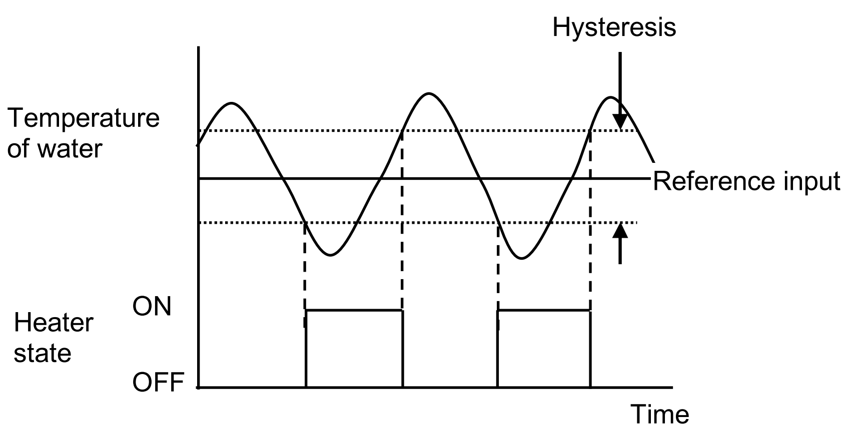
This reduces the switching frequency at the cost of the increasing the magnitude of the process output oscillations.

Example 6.1 Controller for a water heater

Assuming we want to keep the water at a constant temperature this is an example of a regulator. With a simple ON-OFF controller the heater will be turned off whenever $e>0$ (that is whenever the measured temperature is above the reference input) and turned on whenever $e<0$. This produces a continuous oscillation due to sluggish nature of the process dynamics (similar to mechanical inertia). Please see the time response shown below.



When the ON-OFF with hysteresis control algorithm is used the oscillation amplitude will increase, but the frequency will decrease (extending the heater life). The corresponding time response is shown below.



End of example.

Advantages of ON-OFF Control
- Extremely simple to implement.
- Cannot go unstable.

Disadvantages of ON-OFF Control
- Output oscillations are unavoidable.

Applications
Heater control systems, *e.g.* home thermostat.

6.3.2 PID Control

Proportional plus integral plus derivative (PID) control is the most popular control algorithm used in industry since it works well with many processes. Its Laplace transfer function is:

$$\frac{U(s)}{E(s)} = K_P(1+\frac{1}{s}K_I + sK_D) \tag{6.1}$$

where K_P is the proportional gain, K_I is the integral gain and K_D is the derivative gain. Note that the derivative term will amplify any high frequency sensor noise (as we saw in Section 2.2). This can be a problem if we want to use a large derivative gain and the sensor does not have a large signal to noise ratio. One option that is sometimes effective is to low pass filter the derivative term. This filtering has the disadvantage that it increases the phase lag of the controller.

This transfer function must be discretized for use in a digital controller. Normally backwards differencing is used, giving:

$$s = \frac{z-1}{Tz} \tag{6.2}$$

where T is the sampling interval we discussed in Section 4.4. Substituting (6.2) into (6.1) gives the digital PID controller:

$$\frac{U(z)}{E(z)} = K_P(1+K_I\frac{Tz}{z-1} + K_D\frac{z-1}{Tz}) \tag{6.3}$$

or

$$\frac{U(z)}{E(z)} = \frac{K_P\left((K_IT + K_D/T+1)z^2 - (2K_D/T+1)z + K_D/T\right)}{z^2 - z} \tag{6.4}$$

The term "tuning" refers to the selection of controller gains or other parameters to achieve a desired closed-loop performance. Manually tuning a PID controller has the advantage that no mathematical model is required although it requires skill and experience to achieve the desired results. Each of the gains has its own role:

- K_P mainly improves speed of response.
- K_I mainly eliminates steady state error but also adds phase lag to the system.
- K_D mainly adds damping and phase lead to the response.

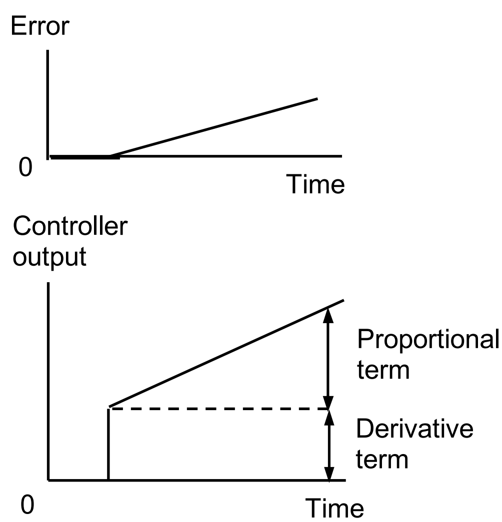
A heuristic approach to PID tuning based on a series of step input tests is as follows:

1) Set all gains equal to zero.
2) Increase K_P until the step response has <u>excessive</u> overshoot.
3) Increase K_D until the overshoot of the step response has the desired value.
4) Increase K_I until the steady state error is eliminated.

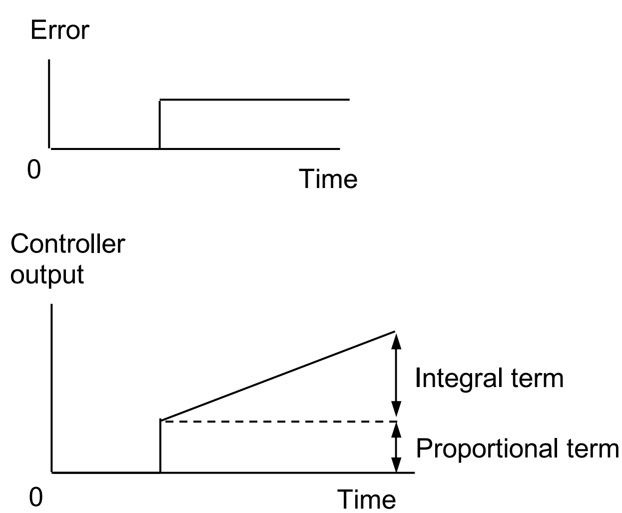
This tuning operation usually involves some iteration and does not apply to every process.

The roles of the three terms of the PID controller are also shown in the figures below.

For a PD controller (with K_I=0):



For a PI controller (with K_D=0):



6.3.3 Model-Based Digital Control

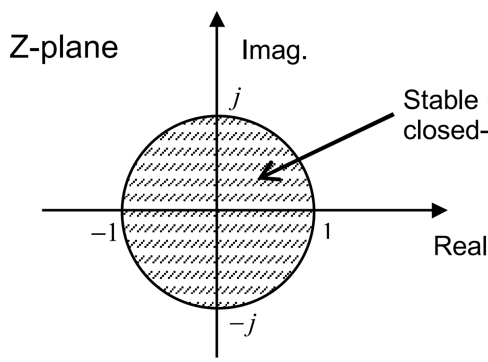
Introduction

With today's flexible digital controllers there is no need to limit the control algorithms to ON-OFF or PID control. Newer model-based digital control algorithms can be developed that outperform these approaches. We will look at a control algorithm whose design is based on the Z transfer functions that were introduced in Section 4.4.

Stability and Digital Control

In the continuous domain a control system is stable if none of the poles of its closed-loop transfer function lie in the right-half of the s-plane.

With digital control the analysis is done on the Z-plane. The Z-plane consists of real and imaginary axes just as the s-plane does. The digital control system is stable if all of the poles of its closed loop transfer function are not outside of the "unit circle". The boundary of the unit circle corresponds to the imaginary axis of the s-plane. The unit circle is centred at the origin and has a radius of one as shown in the figure below.



Example 6.2

For a system with the closed-loop transfer function:

$$\frac{Y(z)}{R(z)} = \frac{2z+3}{z(z^2 - 0.7z + 0.1)} \tag{6.5}$$

We can find the zeros by setting: $2z+3=0$

So we have one zero at $z = -1.5$

We have one pole at $z = 0$. We can find the other poles by setting: $z^2-0.7z+0.1=0$

Applying the quadratic formula we find the other poles are at $z = 0.5$ and $z = 0.2$

Since all poles are not outside the unit circle (that is $|z|\leq 1$ for all poles) the closed-loop is stable.

Model-Based Digital Control (Ragazzini's Method)

There are many methods for designing model-based digital control algorithms. A simple, yet effective, method is that of Ragazzini.

For a standard unity feedback control loop, with a controller transfer function D(z) and a process (or plant) transfer function G(z), the closed-loop transfer function is:

$$Q(z) = \frac{Y(z)}{R(z)} = \frac{D(z)G(z)}{1 + D(z)G(z)} \tag{6.6}$$

If we know the desired closed-loop transfer function H(z) (which defines the kind of output response we are looking for), we simply can equate Q(z) and H(z) and solve for the controller transfer function D(z), as follows:

$$D(z) = \frac{U(z)}{E(z)} = \frac{1}{G(z)} \frac{H(z)}{1 - H(z)} \tag{6.7}$$

Note that several factors must be considered when choosing H(z) otherwise a poor or even unstable controller can be produced.

These are:

1) Causality:

The **causality rule** states that the dead time of H(z) must be greater than or equal to the dead time of G(z). (*i.e.* There is no way for D(z) to make the closed-loop have less dead time than the open-loop). Normally we would pick an H(z) with a dead time equal to the dead time of G(z).

The dead time in sampling periods for a transfer function equals the highest power of z in the denominator minus the highest power of z in the numerator.

Example 6.3

For the transfer function:

$$\frac{2.1z + 3.4}{z^3(z^2 + 1.5z + 2)} \tag{6.8}$$

The highest power of z in the denominator is 5 and the highest power of z in the numerator is 1 so dead time is 5-1= 4 sampling periods.

2) Stability

For stability the rules are:

1) The poles of H(z) must not lie outside the unit circle.
2) H(z) must contain as zeros all of the zeros of G(z) that lie outside the unit circle.
3) 1-H(z) must contain as zeros all of the poles of G(z) that lie outside the unit circle.

Following these stability rules is necessary to prevent the closed-loop from including poles outside of the unit circle.

3) Steady State Error

Normally a control system is designed to be Type 1 (*i.e.* zero steady state error for a step input and a finite steady state error for a ramp input).

Here the error is given by: E(z)=R(z)(1-H(z))
In the Z-domain the final value for a stable system is obtained by setting z=1.
So for zero steady state error to a unit step: E(1) = R(1)(1-H(1)) = 1(1-H(1)) = 0
which has the solution: H(1)=1
i.e., If we design H(z) such that H(1)=1 we will get zero steady state error with a step input and also a finite error to a ramp input.

If the error to a ramp must meet a given numerical specification, then we must do some further work. For a unit ramp input the steady state error is given by:

$$e(\infty)=\sum_{i=1}^{n}\frac{T}{1-p_i}-\sum_{j=1}^{m}\frac{T}{1-q_j} \tag{6.9}$$

Where p_i are the poles of H(z) and q_j are the zeros of H(z). Using equation (6.9) we can achieve the desired steady state error by either moving a pole or zero or by adding an additional zero to H(z).

4) Transient Response

Normally we would design H(z) to have a first or second order transfer function. In continuous control, the speed of response of a first order system is dictated by its time constant, τ, (*i.e.* smaller τ equals faster response) while the speed of response of a second order system is mainly dictated by its natural frequency ω_n (*i.e.* larger ω_n equals faster response). With a second order system the overshoot is a function of the damping ratio ζ (*i.e.* larger ζ equals less overshoot).

We can convert the poles of the desired closed-loop transfer function from the s-plane to the Z-plane using the relation:

$$z = e^{Ts} \tag{6.10}$$

So, if the s-plane pole is $s = a+jb$ then the Z-plane pole is:

$$z = e^{aT+jbT} = e^{aT}\left(\cos(bT) + j\sin(bT)\right) \tag{6.11}$$

Example 6.4

Say we want the closed-loop transfer function to be first order with a time constant of 0.1 seconds. The continuous characteristic equation is then: $\tau s+1=0$, or $0.1s+1=0$. So the s-plane closed-loop pole is s = -10. Using equation (6.11), if the sampling period is T=0.02 seconds, then the pole of H(z) should be located at: $z = e^{(-10)(0.02)} = 0.82$ (*i.e.* we would make the denominator of H(z) equal to: $z - 0.82$).

Example 6.5

If T=5 ms and we want the closed-loop to have a natural frequency of 100 rad/s and a damping ratio of 0.8, what should we use for the denominator of H(z)?

Answer: The desired characteristic polynomial is:

$$s^2 + 2\zeta\omega_n s + \omega_n^{\,2} = s^2 + 2(0.8)(100)s + 100^2 = s^2 + 160s + 10000 \tag{6.12}$$

From the quadratic formula the desired closed-loop poles are: $s = -80\pm 60j$. Applying (6.11) gives the desired Z-plane closed-loop poles:

$$\begin{aligned} z &= e^{aT}\left(\cos(bT) + j\sin(bT)\right) \\ &= e^{(-80)(0.005)}\left(\cos((\pm 60)(0.005)) + j\sin((\pm 60)(0.005))\right) = 0.640 \pm 0.198j \end{aligned} \tag{6.13}$$

So, the denominator of H(z) we should use is:

$$\begin{aligned} &\left(z-(0.640+0.198j)\right)\left(z-(0.640-0.198j)\right) \\ &= z^2 - 2(0.640)z + 0.640^2 + 0.198^2 \\ &= z^2 - 1.28z + 0.449 \end{aligned} \tag{6.14}$$

5) Sensitivity to Modeling Errors

No model is ever perfect so it is important to consider modeling errors when designing a model-based controller. The errors can cause the closed-loop to have a poor performance or even become unstable. This subject is quite complex but a few general observations can be made.

First of all, the closer the poles are located to the origin of the Z-plane the faster the speed of response of the system. Normally we are using closed-loop control to speed up the plant's response so we locate the poles of H(z) closer to the origin than the poles of G(z). There is a limit to how much we can push the system however. In general the greater the difference between the poles of H(z) and the poles of G(z) the more sensitive the controller will be to modeling errors. (So if we are not confident in the accuracy of our model we should not make this make this difference too large).

Implementing a Digital Controller (this includes digital PID)

The inverse Z-transform is used to convert from the Z-domain to the discrete-time domain. Multiplying by a negative power of z is equivalent to adding dead time to a signal. So the inverse Z-transform of $z^{-n}X(z)$ is simply x(k-n) where k is an integer representing the current sample number. (The time corresponding to sample number k is simply equal to kT).

Example 6.6 Given the controller Z transfer function:

$$D(z) = \frac{U(z)}{E(z)} = \frac{5z^2}{z^2 - 0.8z} \tag{6.15}$$

We start by reducing the highest power of z to 0, as follows:

$$D(z) = \frac{U(z)}{E(z)} = \frac{5z^2}{z^2 - 0.8z} \cdot \frac{z^{-2}}{z^{-2}} = \frac{5}{1 - 0.8z^{-1}} \tag{6.16}$$

Then cross multiply to obtain:

$$5E(z) = U(z) - 0.8z^{-1}U(z) \tag{6.17}$$

The inverse Z-transform is:

$$5e(k) = u(k) - 0.8u(k-1) \tag{6.18}$$

To implement this on a microprocessor we just have to write it the form:

$$u(k) = 5e(k) + 0.8u(k-1) \tag{6.19}$$

where k is an integer representing the current sample number, u(k) is the controller output at time kT, u(k-1) is the controller output at time (k-1)T, and e(k) is the error at time kT.

Example 6.7 PID control vs. Model-based control using Ragazinni's method

It is useful to compare the application of a PID controller and a model-based controller to the same plant. For this example the Laplace transfer function of the plant is:

$$\frac{Y(s)}{U(s)} = \frac{1}{s^2 + s + 1} \tag{6.44}$$

So it is second order with $\zeta = 0.5$ and $\omega_n = 1\ rad/s$. The sampling period is T=0.1 seconds and we'll start with the digital PID controller.

Following the heuristic tuning approach given on page 6-5 the three PID gains are initially set equal to zero. Then the proportional gain is gradually increased and step response experiments are conducted to observe the effect of K_P. In this example a unit step is used for *r(t)*. In practice the magnitude of the step should be chosen to be large enough that the signal to noise ratio is good but not so large that the control output (*u(t)*) reaches its saturation limit (recall the definition of actuator saturation on page 4-25). The step response for a small value of K_P is shown in Figure 6.1. Clearly the output response (*y(t)*) is very slow and has a very large steady state error. When K_P is increased to 0.5 the speed and steady state error both improve as shown in Figure 6.2. This trend continues as K_P is further increased while oscillations also appear as shown in Figure 6.3. At some point (based on experience) these oscillations are deemed excessive, K_P is kept constant, and the tuning operation continues with the derivative gain, K_D. A small value of K_D improves the damping of response significantly as shown in Figure 6.4. With K_D= 0.9 (Figure 6.5) the response has a relatively small overshoot. In practice there will always be a point when further increasing K_D will worsen the response either due its amplification of high frequency noise (that we discussed previously) or due to the approximate nature of the digital implementation of the derivative. In this example, K_D values greater than 0.9 (with K_P=6) only worsened the response. Now the tuning operation moves to the integral gain, K_I. With K_I=0.5 (Figure 6.6) the steady state error is eliminated but the settling time is quite slow. With K_I=1 (Figure 6.7) the settling time is much better. Further increasing K_I leads to low frequency oscillations in the output. This step response (Figure 6.7) is close to the best that may be obtained with digital PID control, the given plant and the given sampling interval. The corresponding control output (or plant input) is shown in Figure 6.8. This plot shows that the controller first pushes the plant in the positive direction to produce a rapid response and then pushes it in the negative direction so that it slows down and stops at the steady state value of *r(t)*.

Ideally, we would have liked to have obtained a fast step response with no overshoot. Can the model-based design perform better than PID? If our model of the plant is very accurate then the answer is a definite yes. We can specify H(z) with a desired closed-loop natural frequency of 15 rad/s and a damping ratio of 0.9 and obtain the fast output response with negligible overshoot shown in Figure 6.9. Note that unlike PID control we were able to specify the desired performance beforehand and no iterative tuning was required. One downside of the model-based controller is that the corresponding plant input has a larger magnitude and is more oscillatory than with the PID controller. This might shorten the life of the actuator and/or the plant in practice. Another important issue is the sensitivity of the closed-loop to modeling errors. For the purposes of this example, we will assume the model's parameters are within 20% of their true values. The responses with +20% modeling error (model's $\zeta = 0.4$ and $\omega_n = 0.8\ rad/s$) and with −20% modeling error (model' s $\zeta = 0.6$ and $\omega_n = 1.2\ rad/s$) are shown in Figures 6.11 and 6.12. With this magnitude of modeling error the output response has worsened but is still reasonable.

It should also be noted that in general a smaller value of T results in better closed-loop control performance (for both types of controllers) but requires faster and therefore more expensive hardware.

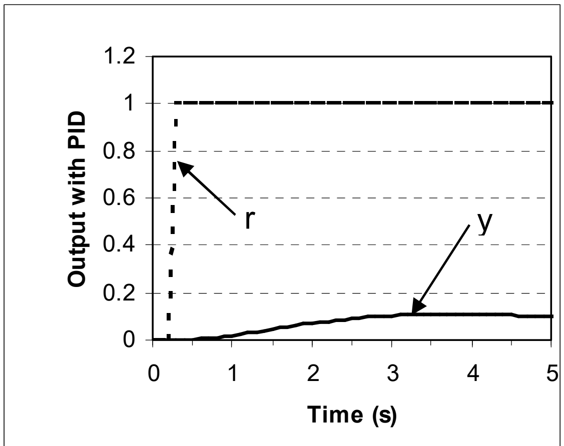


Figure 6.1 K_P=0.1, K_D=0 and K_I=0.

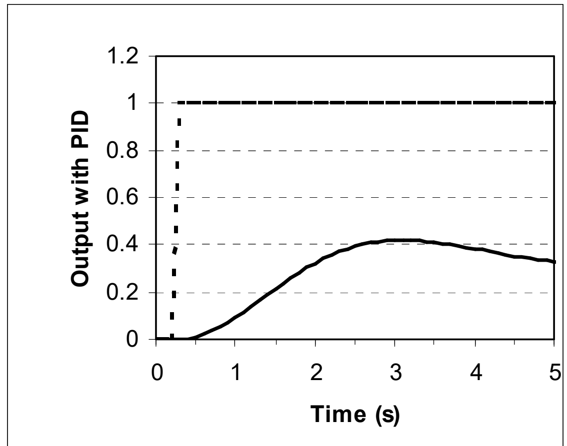


Figure 6.2 K_P=0.5, K_D=0 and K_I=0.

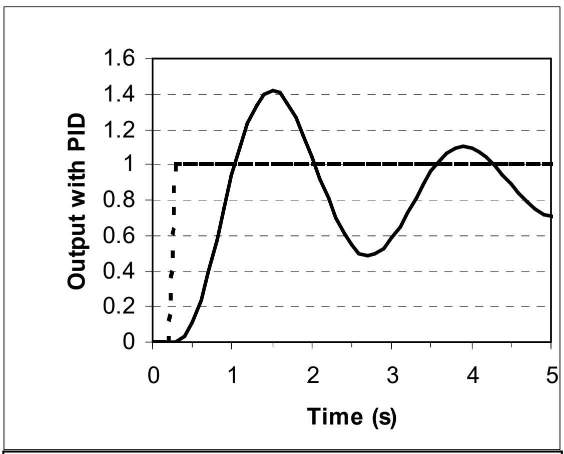


Figure 6.3 K_P=6, K_D=0 and K_I=0.

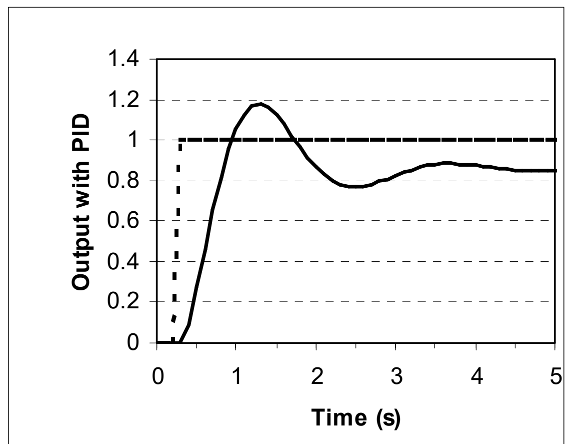


Figure 6.4 K_P=6, K_D=0.2 and K_I=0.

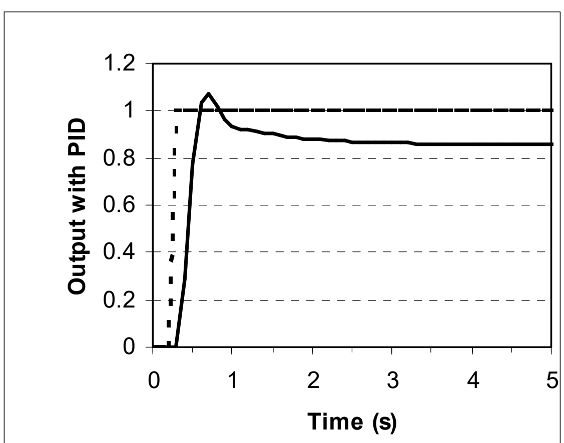


Figure 6.5 K_P=6, K_D=0.9 and K_I=0.

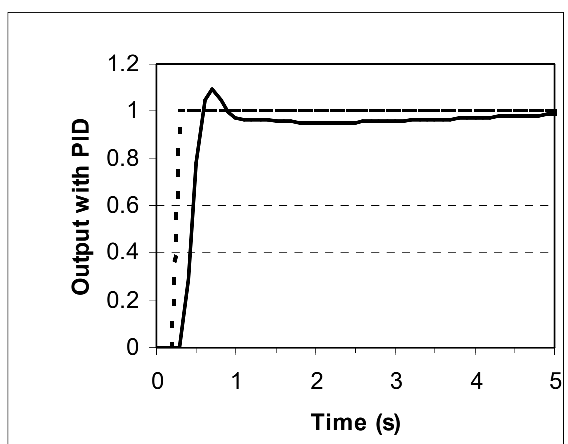


Figure 6.6 K_P=6, K_D=0.9 and K_I=0.5.

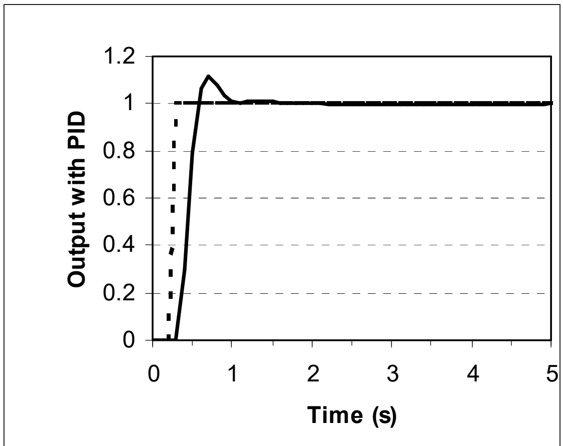


Figure 6.7 K_P=6, K_D=0.9 and K_I=1.

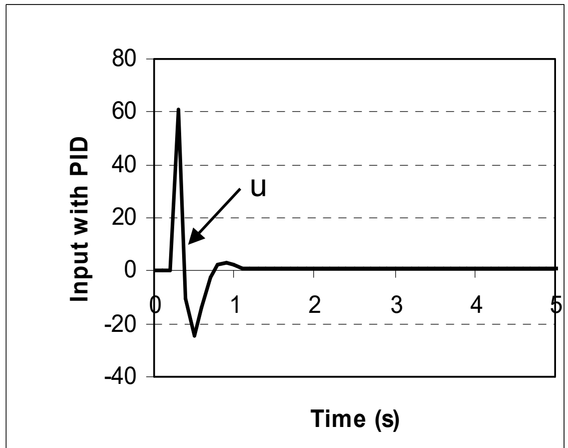


Figure 6.8 Plant input for PID with K_P=6, K_D=0.9 and K_I=1.

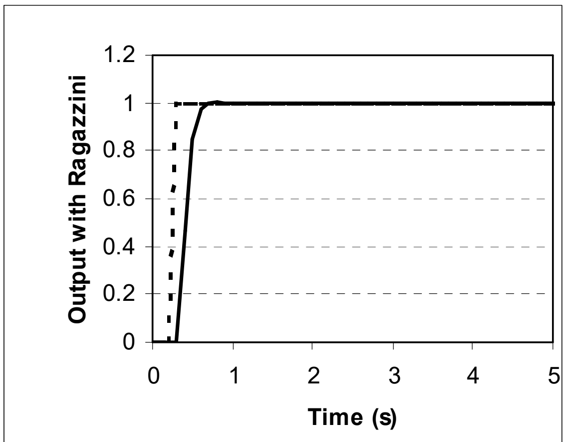


Figure 6.9 Plant output response assuming a perfect model.

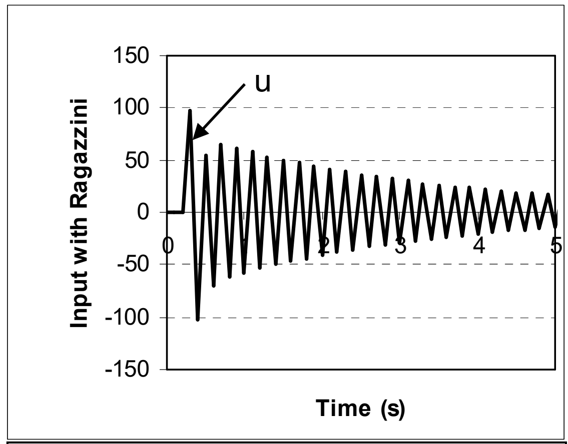


Figure 6.10 Plant input assuming a perfect model.

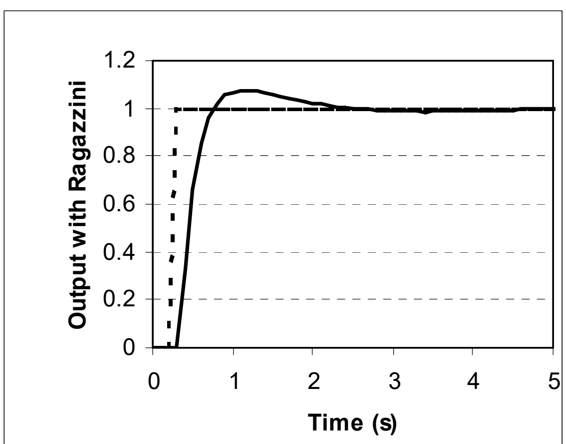


Figure 6.11 Plant output response with a –20% modeling error.

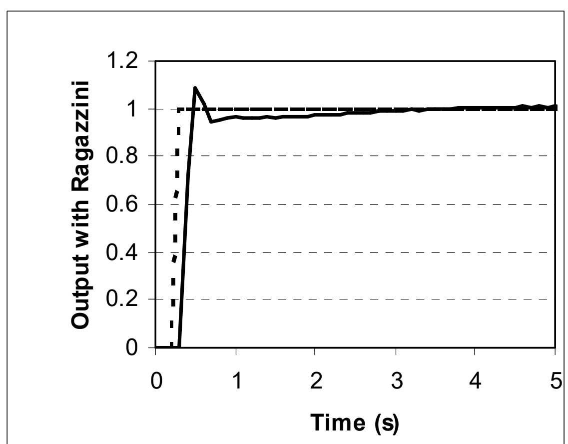


Figure 6.12 Plant output response with a +20% modeling error.

Addendum to Chapter 6

A.1 Another View of the Causality Rule

The dead time of a discrete-time transfer function in sampling periods equals the highest power of z in the denominator minus the highest power of z in the numerator. This was illustrated by example 6.3. The dead time also equals the number of transfer function poles minus the number of transfer function zeros. This is sometimes an easier approach to use with the causality rule.

Example

The transfer function:

$$\frac{2.1z+3.4}{z^3(z^2+1.5z+2)}$$

has five poles (*i.e.*, z = 0, z = 0, z = 0, z = -0.75 + 1.2j, and z = -0.75 - 1.2j) and one zero (*i.e.*, z = -1.62). Therefore, its dead time equals 5 - 1 = 4 sampling periods.

A.2 More Examples of Model-Based Digital Control

Example 6.8

If the sampling period is $T = 0.002$ s and:

$$G(z) = \frac{z-0.7}{z^2(z-2)}$$

then find the $H(z)$ that meets these specifications:
i) The closed-loop transfer function is first order with $\tau = 0.0029$ s and
ii) Zero steady state error to a step input.

Answer

From specification (i) we can calculate the pole of $H(z)$ as follows:

$$z = e^{-\frac{T}{\tau}} = e^{\left(-\frac{0.002}{0.0029}\right)} = 0.50\,.$$

Next, we must check the stability rules. $G(z)$ has two poles at $z = 0$, one pole at z=2 and a zero at z = 0.7. Since $|2|>1$ this pole is outside the unit circle, stability rule 3 applies and we require

$$1 - H(2) = 0 \text{ or}$$

$$H(2) = 1 \tag{1}$$

and to meet specification (ii) we require

$$H(1) = 1. \tag{2}$$

To satisfy equations (1) and (2), we need two parameters, so the numerator of H(z) will have the form: $b_0 + b_1 z$, which has a zero at $z = -\frac{b_0}{b_1}$.

But *G(z)* has three poles and one zero, so to satisfy the causality rule we need to add two more poles to *H(z)*. The best choice is to make these both at $z = 0$. In other words the denominator of H(z) should be: $z^2(z - 0.50)$. So we now have:

$$H(z) = \frac{b_0 + b_1 z}{z^2(z - 0.50)}.$$

From (1):

$$H(2) = \frac{b_0 + b_1(2)}{2^2(2 - 0.50)} = 1$$

$$b_0 + 2b_1 = 6. \tag{3}$$

From (2):

$$H(1) = \frac{b_0 + b_1(1)}{1^2(1 - 0.5)} = 1$$

$$b_0 + b_1 = 0.5 \tag{4}$$

Subtracting (4) from (3) gives $b_1 = 5.5$. Substituting this into (4) gives:

$$b_0 + 5.5 = 0.5$$

$$\therefore b_0 = -5.$$

So, the final answer is: $H(z) = \frac{-5 + 5.5z}{z^2(z - 0.5)}$.

End of example.

Another example on the next page...

Example 6.9

Given $G(z) = \dfrac{0.5(z+1.5)}{z(z-0.6)}$ and $T = 0.001$ s, find the $H(z)$ that meets the specifications:
i) The closed-loop transfer function is first order with $\tau = 0.0045\,\text{s}$ and
ii) Zero steady state error to a step input.

Answer

From specification (i) we can calculate the pole of *H(z)* as follows:

$$z = e^{-\frac{T}{\tau}} = e^{\left(-\frac{0.001}{0.0045}\right)} = 0.80$$

$G(z)$ has poles at z=0 and $z = 0.6$; and a zero at $z = -1.5$. Since $|-1.5| > 1$ this zero is outside of unit circle, so stability rule 2 applies and $H(z)$ must have zero at $z = -1.5$. In equation form:

$$H(-1.5) = 0\,. \tag{1}$$

To satisfy specification (ii) we require:

$$H(1) = 1\,. \tag{2}$$

We can satisfy equations (1) and (2) by making the numerator of *H*(*z*) equal to $b_0(\text{z+1.5})$.

So far, our H(z) has one pole and one zero, but *G(z)* has two poles and one zero, so to satisfy the causality rule we need to add one more pole to *H(z)*. The best choice is to make this pole at $z = 0$. In other words the denominator of *H*(*z*) should be: *z*(*z* - 0.80). So we now have:

$$H(z) = \frac{b_0(z+1.5)}{z(z-0.80)}\,.$$

Next, the value of b_0 can be found using equation (2) as follows:

$$H(1) = \frac{b_0(1+1.5)}{1(1-0.80)} = 1$$

$$2.5b_0 = 0.2$$

$$\therefore b_0 = 0.08\,.$$

The final answer is: $H(z) = \dfrac{0.08(z+1.5)}{z(z-0.8)}$.

End of example.