Chapter 13

Digital Control

Chapter 12 was concerned with building models for systems acting under digital control.

We next turn to the question of control itself.

Topics to be covered include:

- why one cannot simply treat digital control as if it were exactly the same as continuous control, and
- how to carry out designs for digital control systems so that the *at-sample* response is exactly treated.

Having the controller implemented in digital form introduces several constraints into the problem:

(a) the controller sees the output response only at the sample points,

(b) an anti-aliasing filter will usually be needed prior to the output sampling process to avoid folding of high frequency signals (such as noise) onto lower frequencies where they will be misinterpreted; and

(c) the continuous plant input bears a simple relationship to the (sampled) digital controller output, e.g. via a zero order hold device.

A key idea is that if one is only interested in the at-sample response, these samples can be described by discrete time models in either the shift or delta operator. For example, consider the sampled data control loop shown below

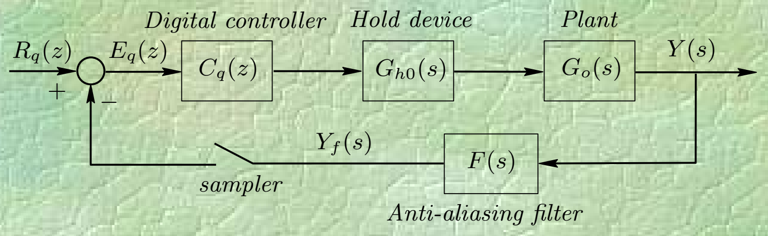


Figure 13.1: *Sampled data control loop*

If we focus only on the sampled response then it is straightforward to derive an equivalent discrete model for the at-sample response of the hold-plant-anti-aliasing filter combination. This was discussed in Chapter 12.

We use the transfer function form, and recall the following forms for the discrete time model:

(a) With anti-aliasing filter F

$$[FG_0G_{h0}]_q(z), \; Z\{\text{sampled impulse response of } F(s)G_0(s)G_{h0}(s)\}$$

(b) Without anti-aliasing filter

$$[G_0G_{h0}]_q(z), \; Z\{\text{sampled impulse response of } G_0(s)G_{h0}(s)\}$$

Control Ideas

Many of the continuous time control ideas studied in earlier chapters carry over directly to the discrete time case. Examples are given below.

The discrete sensitivity function is

$$S_{oq}(z) = \frac{E_q(z)}{R_q(z)} = \frac{1}{\Big(1 + C_q(z)\left[FG_oG_{h0}\right]_q(z)\Big)}$$

The discrete complementary sensitivity function is

$$T_{oq}(z) = \frac{Y_{fq}(z)}{R_q(z)} = \frac{C_q(z)\left[FG_oG_{h0}\right]_q(z)}{\Big(1 + C_q(z)\left[FG_oG_{h0}\right]_q(z)\Big)}$$

These can be used and understood in essentially the same way as they are used in the continuous time case.

Are there special features of digital control models?

Many ideas carry directly over to the discrete case. For example, one can easily do discrete pole assignment. Of course, one needs to remember that the discrete stability domain is different from the continuous stability domain. However, this simply means that the desirable region for closed loop poles is different in the discrete case.

We are led to ask if there are any real conceptual differences between continuous and discrete.

Zeros of Sampled Data Systems

We have seen earlier that open loop zeros of a system have a profound impact on achievable closed loop performance. The importance of an understanding of the zeros in discrete time models is therefore not surprising. It turns out that there exist some subtle issues here as we now investigate.

If we use shift operator models, then it is difficult to see the connection between continuous and discrete time models. However, if we use the equivalent delta domain description, then it is clear that discrete transfer

Functions converge to the underlying continuous time descriptions. In particular, the relationship between continuous and discrete (delta domain) poles is as follows (*See Chapter 12*):

$$p_i^\delta = \frac{e^{p_i \Delta} - 1}{\Delta}; \qquad i = 1, \ldots n$$

where p_i^δ, p_i denote the discrete (delta domain) poles and continuous time poles respectively.

The relationship between continuous and discrete zeros is more complex. Perhaps surprisingly, all discrete time systems turn out to have relative degree 1 irrespective of the relative degree of the original continuous system.

Hence, if the continuous system has n poles and $m(< n)$ zeros then the corresponding discrete system will have n poles and (n-1) zeros. Thus, we have n-m+1 extra discrete zeros. We therefore (*somewhat artificially*) divide the discrete zeros into two sets.

1. **System zeros:** $z_1^\delta, \ldots, z_m^\delta$ Having the property

$$\lim_{\Delta \to 0} z_i^\delta = z_i \qquad\qquad i = 1, \ldots, m$$

where z_i^δ are the discrete time zeros (expressed in the delta domain for convenience) and z_i are the zeros of the underlying continuous time system.

2. **Sampling zeros:** $z^{\delta}_{m+1}, \ldots, z^{\delta}_{n-1}$ Having the property

$$\lim_{\Delta \to 0} \left| z_i^{\delta} \right| = \infty \qquad\qquad i = m+1, \ldots, n-1$$

Of course, if $m = n - 1$ in the continuous time system, then there are no sampling zeros. Also, note that as the sampling zeros tend to infinity for $\Delta \to 0$, they then contribute to the continuous relative degree. This shows the consistency between the two types of model.

We illustrate by a simple example.

Example 13.1

Consider the continuous time servo system of Example 3.4, having continuous transfer function

$$G_o(s) = \frac{1}{s(s+1)}$$

where $n = 2$, $m = 0$. Then we anticipate that discretizing would result in one sampling zero, which we verify as follows.

With a sampling period of 0.1 seconds, the exact shift domain digital model is

$$G_{oq}(z) = K \frac{z - z_o^q}{(z-1)(z-\alpha_o)}$$

where $K = 0.0048$, $z_0^q = -0.967$ and $\alpha_0 = 0.905$.

The corresponding exact delta domain digital model is

$$G_\delta(\gamma) = \frac{K'(\gamma - z_o^\delta)}{\gamma(\gamma - \alpha_o')}$$

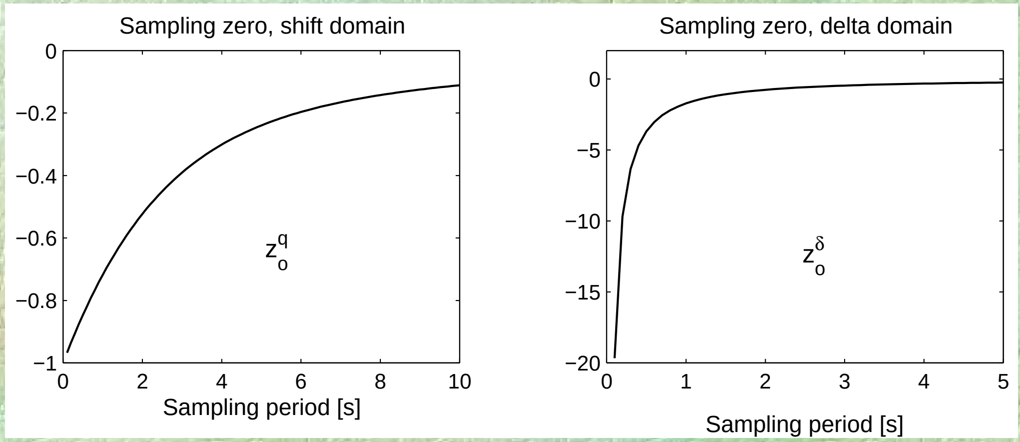
where $K' = 0.0048$, $z_0^q = -19.67$ and $\alpha_0 = -0.9516$.

We see that (*in the delta form*), the discrete system has a pole at $\gamma=0$ and a pole at $\gamma=-0.9516$. These are consistent with the continuous time poles at $s=0$ and $s=-1$.

Note, however, that the continuous system has relative degree 2, whereas the discrete system has relative degree 1 and a *sampling zero* at -19.67 (*in the delta formulation*).

The next slide shows a plot of the sampling zero as a function of sampling period.

Figure 13.2: *Location of sampling zero with different sampling periods. Example 13.1*



In the control of discrete time systems special care needs to be taken with the sampling zeros. For example, these zeros can be non-minimum phase even if the original continuous system is minimum phase. Consider, for instance, the minimum phase, continuous time system with transfer function given by

$$G_o(s) = \frac{s+4}{(s+1)^3}$$

For this system, the shift domain zeros of $[G_0 G_{h0}]_q(z)$ for two different sampling periods are

$$\Delta = 2[s] \qquad \Rightarrow \text{zeros at -0.6082 and -0.0281}$$

$$\Delta = 0.5[s] \qquad \Rightarrow \text{zeros at -1.0966 and 0.1286}$$

Note that $\Delta = 0.5[s]$, the pulse transfer function has a zero outside the stability region.

Thus, one needs to be particularly careful of sampling zeros when designing a digital control system.

Is a Dedicated Digital Theory Really Necessary?

We could well ask if it is necessary to have a separate theory of digital control or could one simply map over a continuous design to the discrete case. Three possible design options are:

1) Design the controller in continuous time, discretize the result for implementation and ensure that the sampling constraints do not significantly affect the final performance.
2) Work in discrete time by doing an exact analysis of the *at-sample* response and ensure that the intersample response is not too surprising, or
3) carry out an exact design by optimizing the continuous response with respect to the (constrained) digital controller.

We will analyze and discuss these 3 possibilities below.

1. Approximate Continuous Designs

Given a continuous controller, $C(s)$, we mention three methods drawn from the digital signal processing literature for determining an *equivalent* digital controller.

1.1 Simply take a continuous time controller expressed in terms of the Laplace variable, s and then replace every occurrence of s by the corresponding delta domain operator γ. This leads to the following digital control law:

$$\overline{C}_1(\gamma) = C(s)\big|_{s=\gamma}$$

where $C(s)$ is the transfer function of the continuous time controller and where $\overline{C}_1(\gamma)$ is the resultant transfer function of the discrete time controller in delta form.

1.2 Convert the controller to a zero order hold discrete equivalent. This is called a *step invariant transformation*. This leads to

$$\overline{C}_2(\gamma) = \mathcal{D}\left[\text{sampled impulse response of}\left\{C(s)G_{h0}(s)\right\}\right]$$

where $C(s)$, $G_{h0}(s)$ and $\overline{C}_2(\gamma)$ are the transfer functions of the continuous time controller, zero order hold and resultant discrete time controller respectively.

1.3 We could use a more sophisticated mapping from s to γ. For example, we could carry out the following transformation, commonly called a *bilinear transformation with pre-warping*. We first let

$$s = \frac{\alpha\gamma}{\frac{\Delta}{2}\gamma + 1} \Longleftrightarrow \gamma = \frac{s}{\alpha - \frac{\Delta}{2}s}$$

The discrete controller is then defined by

$$\overline{C}_3(\gamma) = C(s)|_{s=\frac{\alpha\gamma}{\frac{\Delta}{2}\gamma+1}}$$

We next choose α so as to match the frequency responses of the two controllers at some desired frequency, say ω^*. For example, one might choose ω^* as the frequency at which the continuous time sensitivity function has its maximum value.

We illustrate the above 3 ideas below for a simple system.

Example 13.2

A plant has a nominal model given by

$$G_o(s) = \frac{1}{(s-1)^2}$$

Synthesize a continuous time PID controller such that the dominant closed loop poles are the roots of the polynomial $s^2 + 3s + 4$.

The closed loop characteristic polynomial Acl(s) is chosen as

$$A_{cl}(s) = (s^2 + 3s + 4)(s^2 + 10s + 25)$$

where the factor $s^2 + 10s + 25$ has been added to ensure that the degree of $A_{cl}(s)$ is 4, which is the minimum degree required for an arbitrarily chosen $A_{cl}(s)$.

On solving the pole assignment equation we obtain $P(s) = 88s^2 + 100s + 100$ and $\bar{L}(s) = s + 15$. This leads to the following PID controller

$$C(s) = \frac{88s^2 + 100s + 100}{s(s+15)}$$

We next study the 3 procedures suggested earlier for obtaining an *equivalent* digital control law.

1.1 Method 1 - Here to obtain a discrete time PID controller we simply substitute s by γ. In this case, this yields

$$C_\delta(\gamma) = \frac{88\gamma^2 + 100\gamma + 100}{\gamma(\gamma + 15)}$$

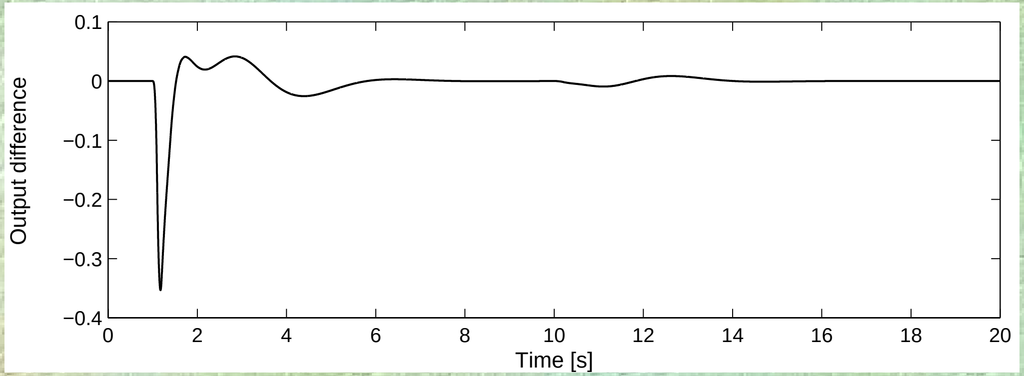
or, in Z transform form

$$C_q(z) = \frac{88z^2 - 166z + 79}{(z-1)(z+0.5)}$$

where we have assumed a sampling period $\Delta = 0.1$.

The continuous and the discrete time loops are simulated with SIMULINK for a unit step reference at $t = 1$ and a unit step input disturbance at $t = 10$. The difference of the plant outputs is shown in Figure 13.3.

Figure 13.3: *Difference in plant outputs due to discretization of the controller (sampling period =0.1*[*s*])



For the above example, we see that method 1.1 (*i.e. simply replace s by γ*) has led to an entirely satisfactory digital control law. However, this isn't always the case as we show by the next example.

Example 13.3

The system nominal transfer function is given by

$$G_o(s) = \frac{10}{s(s+1)}$$

and the continuous time controller is

$$C(s) = \frac{0.416s + 1}{0.139s + 1}$$

Replace the controller by a digital controller with $\Delta = 0.157[s]$ preceded by a sampler and followed by a ZOH using the three approximations outlined earlier.

Three methods for directly mapping a continuous controller to discrete time

1.1 Replacing s by γ in $C(s)$ we get

$$\overline{C}_1(\gamma) = \frac{0.416\gamma + 1}{0.139\gamma + 1}$$

1.2 The ZOH equivalent of $C(s)$ is

$$\overline{C}_2(\gamma) = \frac{0.694\gamma + 1}{0.232\gamma + 1}$$

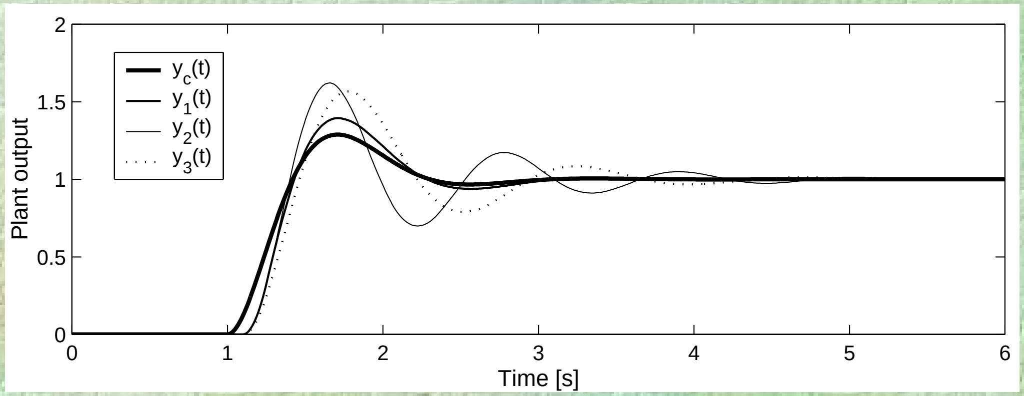
1.3 For the bilinear mapping with pre-warping, we choose $\omega^* = 5.48$. This gives $\alpha = 0.9375$ and the resulting controller becomes

$$\overline{C}_3(\gamma) = C(s)\Big|_{s=\frac{\alpha\gamma}{\frac{\Delta}{2}\gamma+1}} = \frac{0.4685\gamma + 1}{0.2088\gamma + 1}$$

Simulation Results

The above 3 digital controllers were simulated and their performance checked against the performance achieved with the original continuous controller. The results are shown on the next slide.

Figure 13.4: *Performance of different control designs: continuous time ($y_c(t)$), simple substitution ($y_1(t)$), step invariance ($y_2(t)$) and bilinear transformation ($y_3(t)$).*



We see from the figure that none of the approximations exactly reproduces the closed-loop response obtained with the continuous time controller. Actually for this example, we see that simple substitution (Method (1.1)) appears to give the best result and that there is not much to be gained by fancy methods here. However, it would be dangerous to draw general conclusions from this one example.

2. At-Sample Digital Design

The next option we explore is that of doing an exact digital control system design *for the sampled response*.

We recall that the sampled response is exactly described by appropriate discrete-time-models (expressed in either the shift or delta operators).

Time Domain Design

Any algebraic technique (*such as pole assignment*) has an immediate digital counterpart. Essentially all that is needed is to work with z (*or* γ) instead of the Laplace variable, s, and to keep in mind the different region for closed loop stability.

We illustrate below by several special digital control design methods.

Minimal Prototype

The basic idea in this control design strategy is to achieve zero error at the sample points in the minimum number of sampling periods, for step references and step output disturbances (with zero initial conditions). This implies that the complementary sensitivity must be of the form

$$T_o(z) = \frac{p(z)}{z^l}$$

Case 1:

The plant sampled transfer function, $G_{0q}(z)$ is assumed to have all its poles and zeros strictly inside the stability region. Then the controller can cancel the numerator and the denominator of $G_{0q}(z)$ and the pole assignment equation becomes

$$L_q(z)A_{oq}(z) + P_q(z)B_{oq}(z) = A_{clq}(s)$$

where

$$L_q(z) = (z-1)B_{oq}(z)\overline{L}_q(z)$$
$$P_q(z) = K_o A_{oq}(z)$$
$$A_{clq}(s) = z^{n-m}B_{oq}(z)A_{oq}(z)$$

Simplifying, we obtain

$$(z-1)\overline{L}_q(z) + K_o = z^{n-m}$$

This equation can now be solved for K_0 by evaluating the expression at $z = 1$. This leads to $K_0 = 1$, and to a controller and a complementary sensitivity given by

$$C_q(z) = [G_{oq}(z)]^{-1}\frac{1}{z^{n-m}-1}; \qquad \text{and} \qquad T_o(z) = \frac{1}{z^{n-m}}$$

We illustrate this case with an example.

Example 13.4

Consider a continuous time plant with transfer function

$$G_o(s) = \frac{50}{(s+2)(s+5)}$$

Synthesize a minimum prototype controller with sampling period $\Delta = 0.1[s]$.

The sampled transfer function is given by

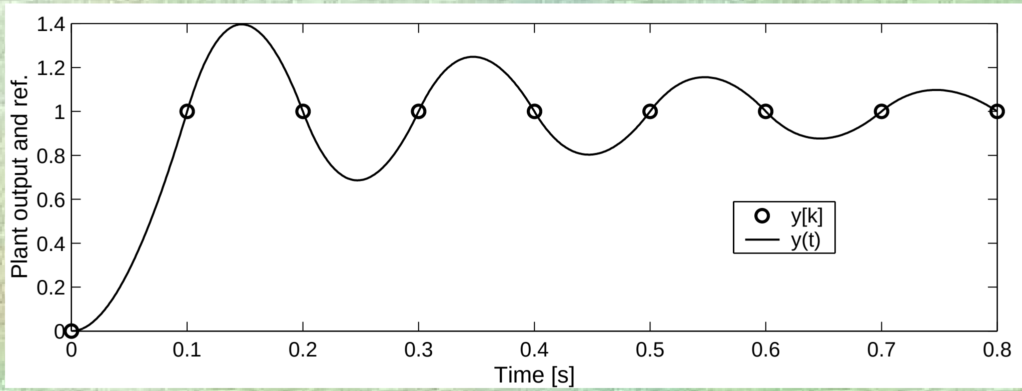
$$G_{oq}(z) = \frac{0.0398(z + 0.7919)}{(z - 0.8187)(z - 0.6065)}$$

Notice that $G_{0q}(z)$ is stable and minimum phase, with $m = 2$ and $n = 3$. The resulting minimal prototype control law is:

$$C_q(z) = \frac{25.124(z - 0.8187)(z - 0.6065)}{(z - 1)(z + 0.7919)} \qquad \text{and} \qquad T_{oq}(z) = \frac{1}{z}$$

The next slide shows a simulation of the closed loop system.

Figure 13.5: *Plant output for a unit step reference and a minimal prototype digital control.*



We see that the *sampled* response settles in exactly one sample period. This is as expected, since $T_{0q}(z) = {}^{1}/_{z}$. However, Figure 13.5 illustrates one of the weaknesses of minimal prototype control: perfect tracking is only guaranteed at the sampling instants!

(*The reader is asked to review the motivating example described in the slides for Chapter 12. Note that exactly the same problem of poor intersample response arose with the earlier example*).

Case 2:

The plant is assumed to be minimum phase and stable, except for a pole at $z = 1$, i.e. $A_{0q}(z) = (z-1)\bar{A}_{0q}(z)$. In this case, the minimal prototype idea does not require that the controller have a pole at $z = 1$. Thus, equations (13.6.6) to (13.6.8) become

$$L_q(z) = B_{oq}(z)\overline{L}_q(z)$$

$$P_q(z) = K_o\overline{A}_{oq}(z)$$

$$A_{clq}(z) = z^{n-m}B_{oq}(z)\overline{A}_{oq}(z)$$

The resulting control is as follows.

$$C_q(z) = [G_{oq}(z)]^{-1}\frac{1}{z^{n-m}-1} = \frac{\overline{A}_{oq}(z)}{B_{oq}(z)}\frac{z-1}{z^{n-m}-1}$$

$$= \frac{\overline{A}_{oq}(z)}{B_{oq}(z)(z^{n-m-1}+z^{n-m-2}+z^{n-m-3}+\ldots+z+1)}$$

$$T_{oq}(z) = \frac{1}{z^{n-m}}$$

Example 13.5

Consider the servo system of Example 3.4. Recall that its transfer function is given by

$$G_o(s) = \frac{1}{s(s+1)}$$

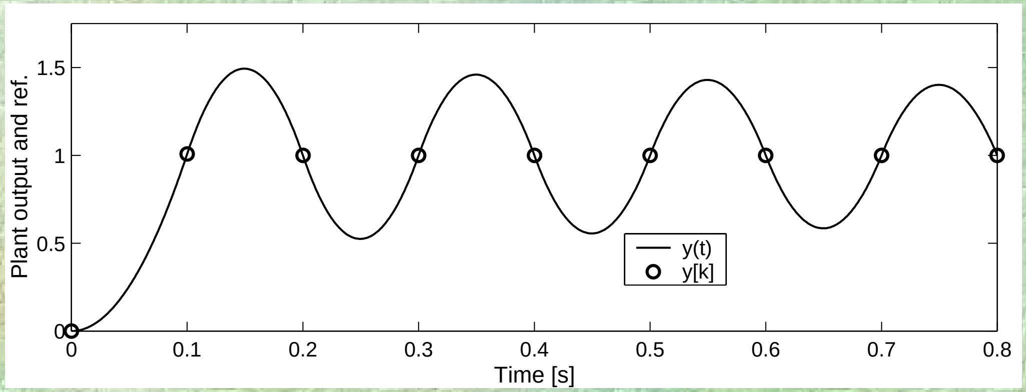
Synthesize a minimal prototype controller with sampling period $\Delta = 0.1[s]$.

$$G_{oq}(z) = 0.0048 \frac{z + 0.967}{(z-1)(z-0.905)}$$

$$C_q(z) = 208.33 \frac{z-0.905}{z+0.967}$$

$$T_{0q}(z) = \frac{1}{z}$$

Figure 13.6: *Plant output for a unit step reference and a minimal prototype digital control. Plant with integration.*



Note that the above results are essentially identical to the simulation results presented for the motivational example given in the slides for Chapter 12.

Minimum Time Dead-Beat Control

The basic idea in dead-beat control design is similar to that in the minimal prototype case: to achieve zero error at the sample points in a finite number of sampling periods for step references and step output disturbances (*and with zero initial conditions*). However, in this case we add the requirement that, for this sort of reference and disturbance, the controller output $u[k]$ also reach its steady state value in the same number of intervals.

The design involves cancelling the open loop poles in the controller. Thus, the system is (*for the moment*) assumed to be stable. We see that the result is achieved by the following control law

$$C_q(z) = \frac{\alpha A_{0q}(z)}{z^n - \alpha B_{0q}(z)}; \quad \alpha = \frac{1}{B_{0q}(1)}$$

The resulting closed loop complementary sensitivity function is

$$T(z) = \frac{C_q(z) G_{0q}(z)}{1 + C_q(z) G_{0q}(z)} = \frac{\alpha B_{0q}(z)}{z^n}$$

Example

Consider the servo system

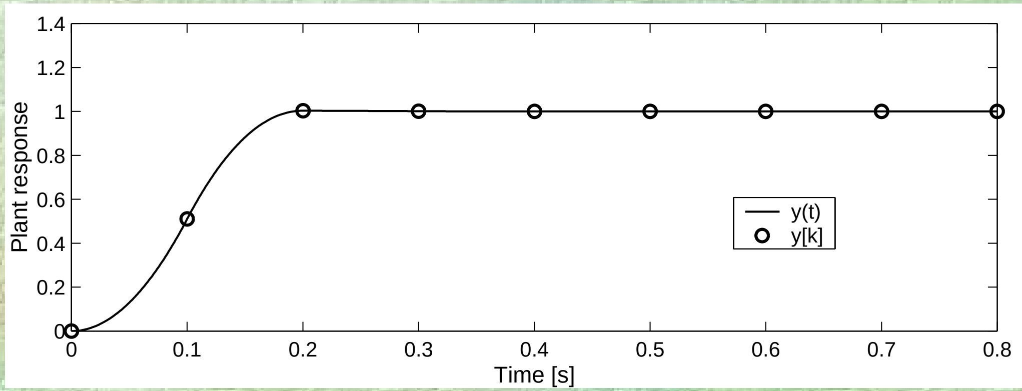
$$G_o(s) = \frac{1}{s(s+1)}$$

Synthesize a minimum time dead-beat control with sampling period $\Delta = 0.1[s]$.

$$C_q(z) = \frac{\alpha A_{0q}(z)}{z^n - \alpha B_{0q}(z)} = \frac{105.49z - 95.47}{z + 0.4910}$$

The next slide shows the simulated response.

Figure 13.7: *Minimum time dead-beat control for a second order plant*



From the above result we see that the intersample problem has been solved by the dead-beat control law.

Note, however, that this is still a very wide-bandwidth control law and thus the other problems discussed in the slides for Chapter 12 (*i.e. noise, input saturation and timing jitter issues*) will still be a problem for the dead-beat controller.

The controller presented above has been derived for stable plants or plants with at most one pole at the origin. Thus cancellation of $A_{0q}(z)$ was allowed. However, the dead-beat philosophy can also be applied to unstable plants, provided that dead-beat is attained in more than n sampling periods. To do this we simply use pole assignment and place all of the closed loop poles at the origin.

Indeed, dead-beat control is then seen to be simply a special case of general pole-assignment. We study the general case below.

Digital Control Design by Pole Assignment

Minimal prototype and dead-beat approaches are particular applications of pole assignment. Indeed, all can be derived by solving the usual pole assignment equation:

$$A(q)L(q) + B(q)P(q) = A_{cl}(q)$$

for particular values of $A_{cl}(q)$.

The general pole assignment problem is illustrated below.

Example

Consider a continuous time plant having a nominal model given by

$$G_o(s) = \frac{1}{(s+1)(s+2)}$$

Design a digital controller, $C_q(z)$, which achieves a loop bandwidth of approximately 3[rad/s]. The loop must also yield zero steady state error for constant references.

We first use the MATLAB program ***c2del.m*** to obtain the discrete transfer function in delta form representing the combination of the continuous time plant and the zero order hold mechanism. This yields

$$\mathcal{D}\left\{G_{ho}(s)G_o(s)\right\} = \frac{0.0453\gamma + 0.863}{\gamma^2 + 2.764\gamma + 1.725}$$

We next choose the closed loop polynomial $A_{cl\delta}(\gamma)$ to be equal to

$$A_{cl\delta}(\gamma) = (\gamma + 2.5)^2(\gamma + 3)(\gamma + 4)$$

The resulting pole assignment equation has the form

$$(\gamma^2 + 2.764\gamma + 1.725)\gamma\overline{L}_\delta(\gamma) + (0.0453\gamma + 0.863)P_\delta(\gamma) = (\gamma + 2.5)^2(\gamma + 3)(\gamma + 4)$$

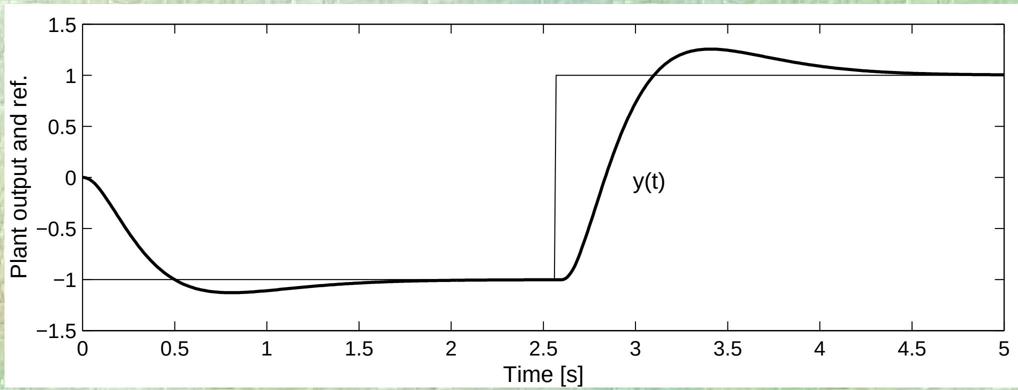
The MATLAB program ***paq.m*** is then used to solve this equation, leading to $C_\delta(\gamma)$, which is finally transformed into $C_q(z)$. The delta and shift controllers are given by

$$C_\delta(\gamma) = \frac{29.1\gamma^2 + 100.0\gamma + 87.0}{\gamma^2 + 7.9\gamma} = \frac{P_\delta(\gamma)}{\gamma \overline{L}_\delta(\gamma)} \qquad \text{and}$$

$$C_q(z) = \frac{29.1z^2 - 48.3z + 20.0}{(z-1)(z-0.21)}$$

Finally, the closed loop response is as shown on the next slide.

Figure 13.8: *Performance of digital control loop*



Internal Model Principle for Digital Control

Most of the ideas presented in previous chapters carry over to digital systems. One simply needs to take account of issues such as the different stability domains and model types.

We illustrate below by the Internal Model Principle which was discussed for Continuous Systems in Chapter 10. In the discrete case, one can choose the internal model to achieve some very interesting results. An example of this is given by repetitive control which we discuss below.

Repetitive Control

An interesting special case of the Internal Model Principle in digital control occurs with periodic signals. It is readily seen that any periodic signal of period N_p samples can be modeled by a discrete time model (*in shift operator form*) using a generating polynomial given by

$$\Gamma_{dq}(q) = \left(q^{N_p} - 1\right)$$

Hence, using the internal model principle, any N_p period reference signal can be exactly tracked (*at least at the sample points*) by including $\Gamma_{dq}(q)$ in the denominator of the controller. This idea is the basis of a technique known as repetitive control aimed at causing a system to learn how to carry out a repetitive (*periodic*) task.

We illustrate by a simple example.

Example 13.10:

Consider a continuous time plant with nominal transfer function $G_0(s)$ given by

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

Assume that this plant has to be digitally controlled with sampling period $\Delta = 0.2[s]$ in such a say that the plant output tracks a periodic reference, $r[k]$, given by

$$r[k] = \sum_{i=0}^{\infty} r_T[k-10i] \Longleftrightarrow R_q(z) = R_{Tq}(z)\frac{z^{10}}{z^{10}-1}$$

Synthesize the digital control which achieves zero steady state at-sample errors.

We observe that the reference generating polynomial $qr(z)$, is given by z^{10} - 1. Thus the IMP leads to the following controller structure

$$C_q(z) = \frac{P_q(z)}{L_q(z)} = \frac{P_q(z)}{\overline{L}_q(z)\Gamma_{qr}(z)}$$

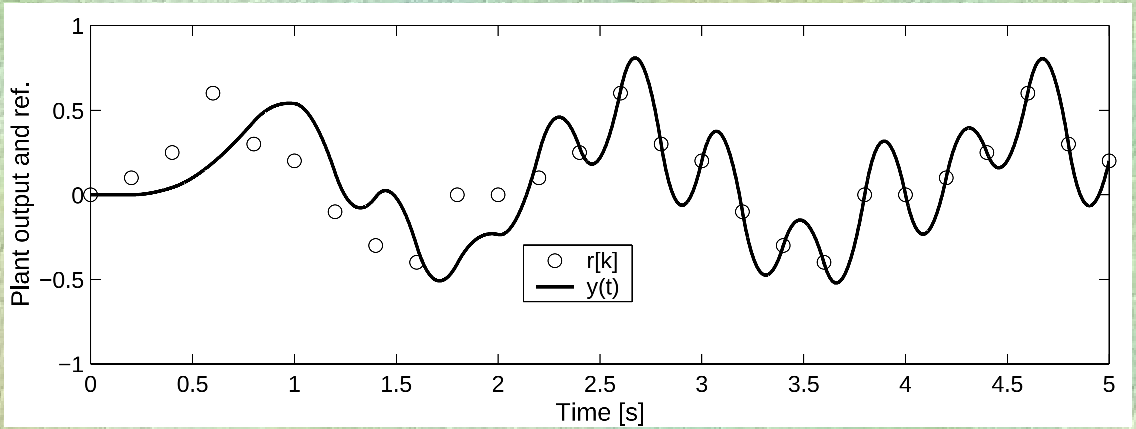
We then apply pole assignment with the closed loop polynomial chosen as $A_{clq}(z) = z^{12}(z - 0.2)$. The solution of the Diophantine equation yields

$$P_q(z) = 13.0z^{11} + 11.8z^{10} - 24.0z^9 + 19.7z^8 - 16.1z^7 + 13.2z^6 - \\ - 10.8z^5 + 8.8z^4 - 7.2z^3 + 36.4z^2 - 48.8z + 17.6$$

$$L_q(z) = (z^{10} - 1)(z + 0.86)$$

Simulating the system with a period 12 input signal leads to the results shown on the next slide.

Figure 13.9: *Repetitive control*



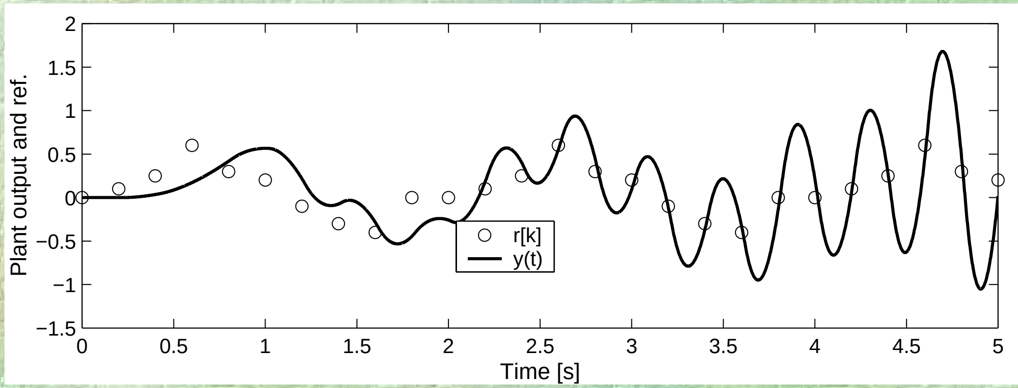
Notice that, after an initial transient, the output tracks the desired periodic reference exactly at the sample points. Unfortunately, we notice that tracking is only guaranteed at the sample points for this form of control law.

Robustness of Repetitive Controllers

Repetitive control is a highly attractive idea since it allows any periodic disturbance to be rejected or any periodic input to be tracked (at the sample points). However, there are costs associated with use of this control law. For example, we have already observed that intersample issues can arise. Another issue is robustness as discussed below.

Perfect tracking in steady state, for reference with *high frequency* harmonics may compromise the robustness of the nominal design. This can be appreciated by introducing a 0.02 [*s*] unmodelled time delay in the control loop designed in Example 13.10. For this case, the loop behavior is depicted in Figure 13.10.

Figure 13.10: *Repetitive control loop in the presence of an unmodelled time delay*



Notice that the inclusion of the small delay has led to closed loop instability.

This behaviour can be readily understood. In particular, perfect tracking requires T_0 to be 1 at all frequencies of interest. However, we know from Theorem 5.3 that robust stability usually requires that $|T_0|$ be reduced at high frequencies. This can be achieved in several ways. For example, one could redefine the internal model to include only the frequency components up to some maximum frequency determined by robustness issues, i.e. we could use

$$\Gamma_{dq}(q) = (q-1)\prod_{i=1}^{N_{max}}\left(q^2 - 2\cos\left(\frac{i2\pi}{N_p}\right)q + 1\right); \qquad N_{max} \leq \frac{N_p - 1}{2}$$

Summary

- There are a number of ways of designing digital control systems:
  - design in continuous time and discretize the controller prior to implementation;
  - model the process by a digital model and carry out the design in discrete time.
- Continuous time design which is discretized for implementation:
  - Continuous time signals and models are utilized for the design;
  - Prior to implementation, the controller is rep0laced by an equivalent discrete time version;
  - Equivalent means to simply map s to δ (where δ is the delta operator);

- Caution must be exercised since the analysis was carried out in continuous time and the expected results are therefore based on the assumption that the sampling rate is high enough to mask sampling effects;
- If the sampling period is chosen carefully, in particular with respect to the open and closed loop dynamics, then the results should be acceptable.

- Discrete design based on a discretized process model:
  - First the model of the continuous process is discretized;
  - Then, based on the discrete process, a discrete controller is designed and implemented;
  - Caution must be exercised with so called intersample behavior: the analysis is based entirely on the behavior as observed at discrete points in time, but the process has a continuous behavior also between sampling instances;

- Problems can be avoided by refraining from designing solutions which appear feasible in a discrete time analysis, but are known to be unachievable in a continuous time analysis (such as removing non-minimum phase zeros from the closed loop!).

- The following rules of thumb will help avoid intersample problems if a purely digital design is carried out:
  - Sample 10 times the desired closed loop bandwidth;
  - Use simple anti-aliasing filters to avoid excessive phase shift;
  - Never try to cancel or otherwise compensate for discrete sampling zeros;
  - always check the intersample response.