Digital Control

The previous lecture 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 the delta operator. For example, consider the sampled data control loop shown below

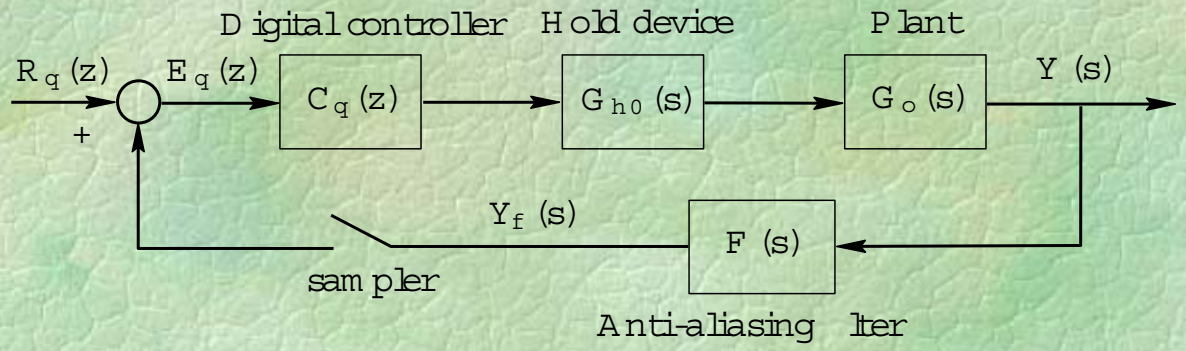


Figure 13.1: *Sampled data control loop*

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.

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 lectures carry over directly to the discrete time case. Examples are given below.

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.

Continuous-Discrete Poles

Functions converge to the underlying continuous time descriptions. In particular, the relationship between continuous and discrete poles is as follows:

$$p_i^d = e^{p_i T} \quad \text{or} \quad p_i^d \cong p_i T + 1, \quad i = 1, \cdots, n$$

where p_i^d, p_i denote the discrete (z-domain) poles and continuous time poles, respectively.

Continuous-Discrete Zeros

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 $(n\text{-}1)$ zeros. Thus, we have n-m+1 extra discrete zeros. We therefore (*somewhat artificially*) divide the discrete zeros into two sets.

Example 13.1

Consider the continuous time system having continuous transfer function

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

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

With a sampling period of T=0.1 seconds, the shift domain digital model is:

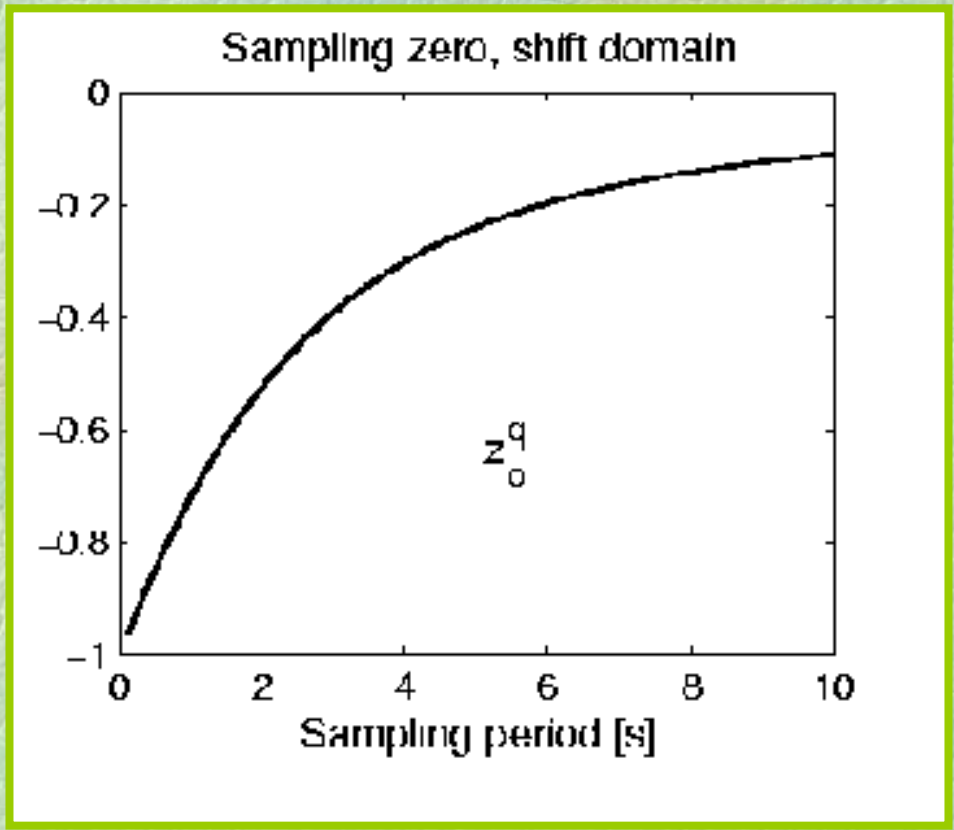
$$G_{oq}(z) = K \frac{z \quad z_o^q}{(z \quad 1)(z \quad _o)}$$

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

Note that the continuous system has relative degree 2, whereas the discrete system has relative degree 1 and a *sampling zero*.

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.*



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

$$\Delta = 2[s] \quad \Rightarrow \text{zeros at } -0.6082 \text{ and } -0.0281$$
$$\Delta = 0.5[s] \quad \Rightarrow \text{zeros at } -1.0966 \text{ and } 0.1286$$

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. The possible design options are:

1) Design the controller in continuous time, discretise 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.

We will analyze and discuss these possibilities below.

1. Approximate Continuous Designs

Given a continuous controller, $C(s)$, we mention the 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 shift domain operator z. This leads to the following digital control law:

$$\overline{C}_1(z) = C(s)\Big|_{s=\frac{z-1}{T}}$$

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

1.2 Convert the controller to a zero order hold discrete equivalent. This is called a *step invariant transformation*, Hold Equivalence (HE). This leads to

$$\overline{C}_2(\ ) = \mathrm{D}\ [\mathrm{sampled\ impulse\ response\ of} C(s)G_{h0}(s)]$$

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.

1.2.1 Convert the controller to a simple discrete equivalent. This leads to

$$\overline{C}_{2.1}(\gamma) = \mathrm{D}\left[\text{sampled impulse response of } \{\mathrm{C(s)}\}\right]$$

where $C(s)$ and $\overline{C}_{2.1}(\gamma)$ are the transfer functions of the continuous time controller and the 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

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

The discrete controller is then defined by

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

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 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 $A_{cl}(s)$ 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 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 z. In this case, with

$$s = \frac{z-1}{T}, \quad T = 0.1$$

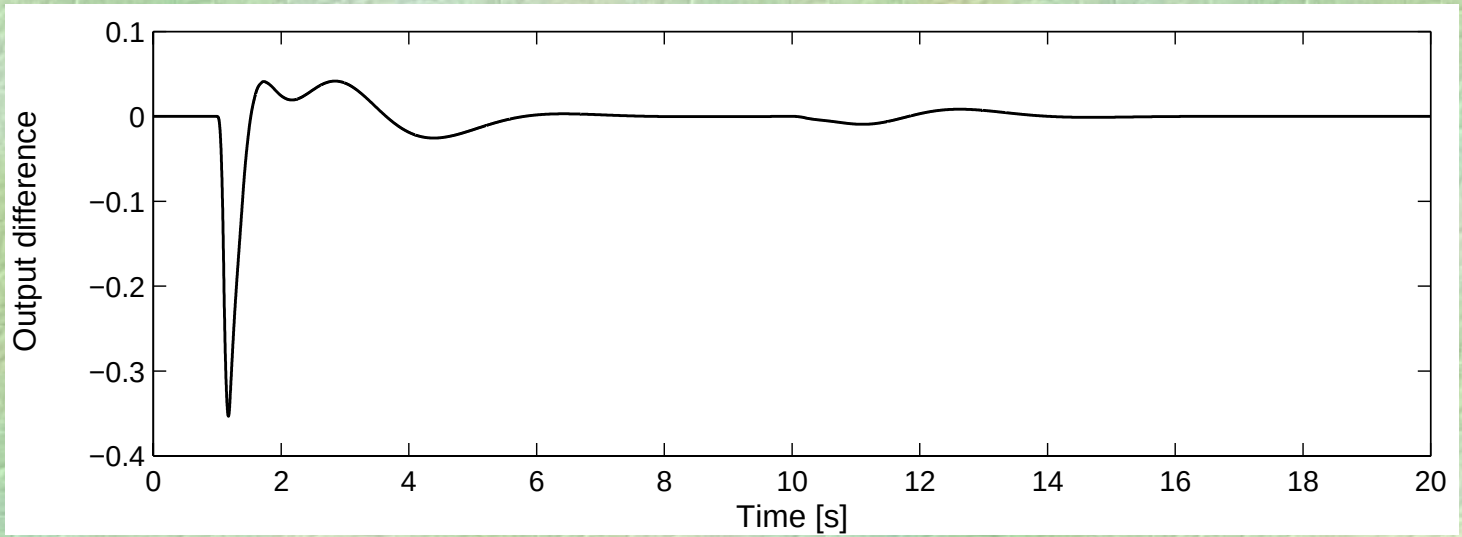
or, in Z transform form

$$C_q(z) = \frac{88z^2 \quad 166z + 79}{(z \quad 1)(z + 0{:}5)}$$

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 discretisation of the controller (sampling period =0.1[s])*



For the above example, we see that method 1.1 (*i.e. simply replace s by z*) 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.

The methods for directly mapping a continuous controller to discrete time

1.1 Replacing s by z (γ) in $C(s)$ we get

$$\overline{C}_1(\gamma) = 1.8633\frac{1.6062\,z-1}{z-1}$$

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

$$\overline{C}_2(\ ) = \frac{0{:}694\ \ +\ 1}{0{:}232\ \ +\ 1}$$

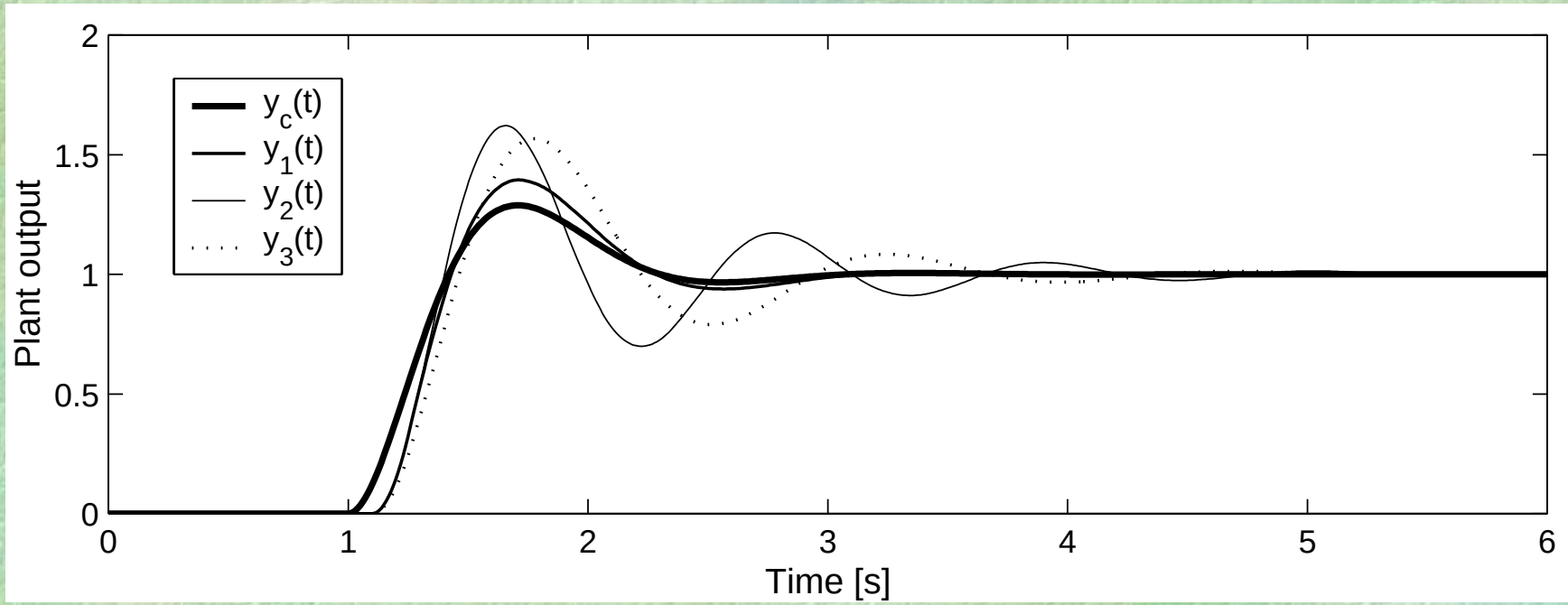
1.3 For the bilinear mapping with pre-warping, we choose $\omega^* = 5.48$. This gives $\alpha = 0.9375$ and the resulting controller becomes

$$\overline{C}_3(\ ) = C(s)\Big|_{s=\frac{\ }{\overline{2}\ +1}} = \frac{0{:}4685\ \ +\ 1}{0{:}2088\ \ +\ 1}$$

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 operator z).

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.

Frequency Domain Design

Automatic design techniques can be exploited for frequency domain design.

Common frequency domain design tools are:

- Bode plots;
- Root locus;
- Nyquist diagrams.

- See laboratory experiences and practical applications…

Summary

- There are a number of ways of designing digital control systems:
  - design in continuous time and discretise 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 discretised for implementation:
  - Continuous time signals and models are utilized for the design;
  - Prior to implementation, the controller is replaced by an equivalent discrete time version;
  - Equivalent means to simply map s to z (where z is the shift 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 discretised process model:
  - First the model of the continuous process is discretised;
  - 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.

- 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;
  - always check the intersample response.