Models for Sampled Data Systems

Motivation

Up to this point in the course, we have assumed that the control systems we have studied operate in continuous time and that the control law is implemented in analogue fashion. Certainly in the early days of control, all control systems were implemented via some form of analogue equipment. Typically controllers were implemented using one of the following formats:

- hydraulic
- pneumatic
- analogue electronic

However, in recent times, almost all analogue controllers have been replaced by some form of computer control.

This is a very natural move since control can be conceived as the process of making computations based on past observations of a system's behavior so as to decide how one should change the manipulated variables to cause the system to respond in a desirable fashion.

The most natural way to make these computations is via some form of computer.

A huge array of control orientated computers are available in the market place.

A typical configuration includes:

- some form of central processing unit (*to make the necessary computations*)

- analogue to digital converters (*to read the analogue process signals into the computer*).
  (We call this the process of **SAMPLING**)
- digital to analogue converters (*to take the desired control signals out of the computer and present them in a form whereby they can be applied back onto the physical process*).
  (We call this the process of **SIGNAL RECONSTRUCTION**)

Types of Control Orientated Computer

Depending upon the application, one could use many different forms of control computer. Typical control orientated computers are:

DCS (Distributed Control System) These are distributed computer components aimed at controlling a large plant.

PLC (Programmable Logic Controller) These are special purpose control computers aimed at simple control tasks - especially those having many on-off type functions.

PC (Personal Computer) There is an increasing trend to simply use standard PC's for control. They offer many advantages including minimal cost, flexibility and familiarity to users.

Embedded Controller. In special purpose applications, it is quite common to use special computer hardware to execute the control algorithm. Indeed, the reader will be aware that many commonly used appliances (CD players, automobiles, motorbikes, etc.) contain special microprocessors which enable various control functions.

Why Study Digital Control?

A simple (*engineering*) approach to digital control is to sample quickly and then to make some reasonable approximation to the derivatives of the digital data. For example, we could approximate the derivative of an analogue signal, $y(t)$, as follows:

$$\frac{d}{dt}y(t) \approx \frac{y(t)-y(t-\Delta)}{\Delta}$$

where Δ is the sampling period.

The remainder of the design might then proceed exactly as for continuous time signals and systems using the continuous model.

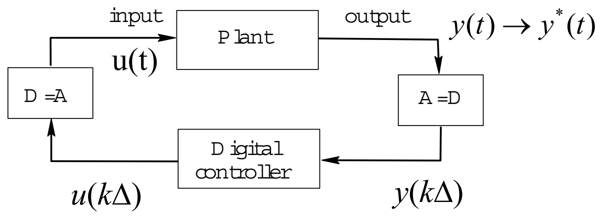
Actually, the above strategy turns out to be quite good and it is certainly very commonly used in practice.

However, there are some unexpected traps for the unwary. These traps have lead to negative experiences for people naively trying to do digital control by simply mimicking analogue methods. Thus it is important to know when such simple strategies make sense and what can go wrong. We will illustrate by a simple example below.

Example of Plant Control

We consider the control of a general system via a computer. This is a very simple example. Yet we will show that this simple example can (when it is fully understood) actually illustrate almost an entire course on digital control.

The set-up for digital control of this system is shown schematically below:



The objective is to cause the output $y(t)$, to follow a given reference signal, $y^*(t)$.

Modelling

Since the control computations will be done inside the computer, it seems reasonable to first find a model relating the sampled output

$\{y(k\Delta); k = 0, 1, \dots \}$

to the sampled input signals generated by the computer, which we denote by

$\{u(k\Delta), k = 0, 1, \dots \}$.

Here Δ is the sample period.

"Let's study digital control"

By the time you have studied the next points you will understand all of the features of the problem, *e.g.*

- how to build the digital control model;
- what are the special features of the one-step-ahead control law we have used; and
- why funny things can (*and sometimes do*) happen between samples.

The current lecture is principally concerned with modelling issues, i.e. how to relate samples of the output of a physical system to the sampled data input.

Specific topics to be covered are:

- Discrete-time signals
- Z-transforms and Delta transforms
- Sampling and reconstruction
- Aliasing and anti-aliasing filters
- Sampled-data control systems

Sampling

The result of sampling a continuous time signal is shown below:

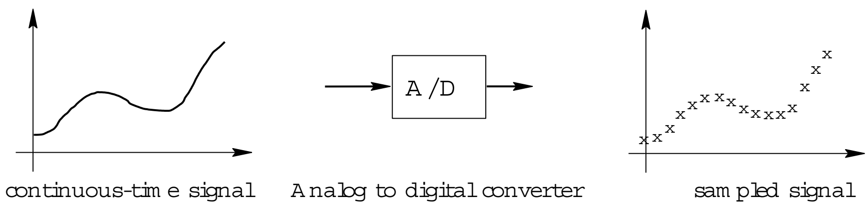


Figure 12.10: *The result of sampling*

There will always be loss of information due to sampling. However, the extent of this loss depends on the sampling method and the associated parameters. For example, assume that a sequence of samples is taken of a signal $f(t)$ every Δ seconds, then the sampling frequency needs to be large enough in comparison with the maximum rate of change of $f(t)$. Otherwise, high frequency components will be mistakenly interpreted as low frequencies in the samples sequence.

Example 12.1

Consider the signal

$$f(t) = 3\cos 2\pi t + \cos\left(20\pi t + \frac{\pi}{3}\right)$$

We observe that if the sampling period Δ is chosen equal to $0.1[s]$ then

$$\begin{aligned} f(k\Delta) &= 3\cos(0.2k\pi) + \cos\left(2k\pi + \frac{\pi}{3}\right) \\ &= 3\cos(0.2k\pi) + 0.5 \end{aligned}$$

from where it is evident that the high frequency component has been shifted to a constant, i.e. the high frequency component appears as a signal of low frequency (*here zero*). This phenomenon is known as ***aliasing***.

This effect is illustrated on the next slide.

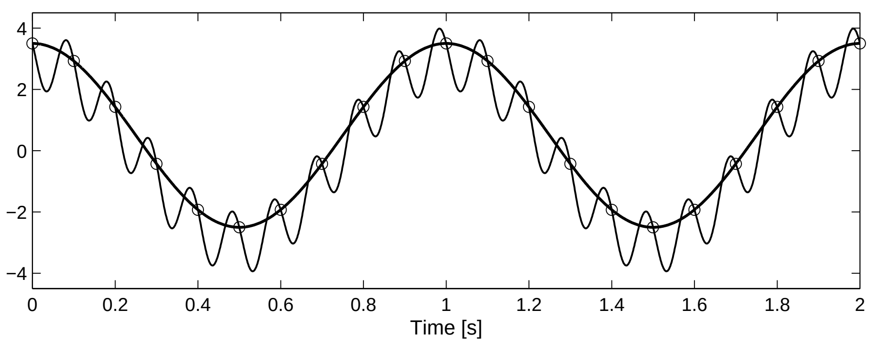


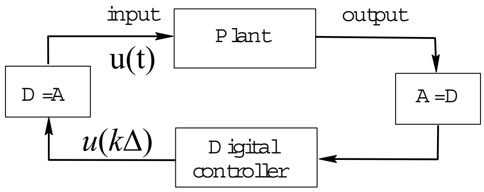
Figure 12.1: *Aliasing effect when using low sampling rate*

Conclusion:

To mitigate the effect of aliasing the sampling rate must be high relative to the rate of change of the signals of interest. A typical rule of thumb is to require that the sampling rate be 5 to 10 times the bandwidth of the signals.

Signal Reconstruction

The output of a digital controller is another sequence of numbers $\{u[k]\}$ which are the sample values of the intended control signal. These sample values need to be converted back to continuous time functions before they can be applied to the plant. Usually, this is done by interpolating them into a staircase function $u(t)$ as illustrated in Figure 12.2.



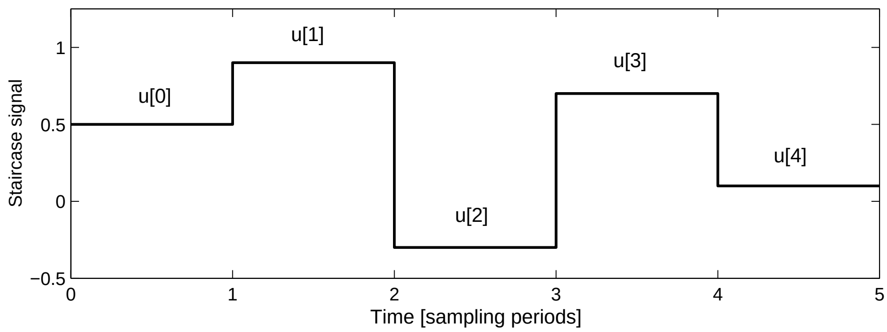


Figure 12.2: *Staircase reconstruction*

Illustration of Signal Reconstruction

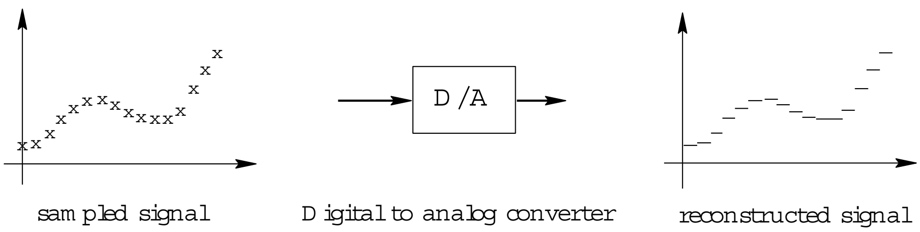


Figure 12.2: *The result of reconstruction*

Modelling

Given the process of signal reconstruction and sampling, we see that the net result is that, inside the computer, the system input and output simply appear as sequences of numbers.

It therefore makes sense to build digital models that relate a discrete time input sequence, $\{u(k)\}$, to a sampled output sequence $\{y(k\Delta)\}$.

Linear Discrete Time Models

A useful discrete time model of the type referred to above is the linear version of the high order difference equation model. In the discrete case, this model takes the form:

$$y[k+n] + \overline{a}_{n-1}y[k+n-1] + \cdots + \overline{a}_0 y[k]$$
$$= \overline{b}_{n-1}u[k+n-1] + \cdots + \overline{b}_0 u[k]$$

Note that we saw a special form of this model in relation to the example presented earlier.

To simplify the way we write the model equations, we will find it useful to have a simple notation to represent a time-shifted output sample, $y\left(\overline{k+m}\Delta\right)$. We introduce a special operator (the shift operator) that allows us to write this very compactly.

The Shift Operator

Forward shift operator

$$\boxed{q(f[k]) \triangleq f[k+1]}$$

In terms of this operator, the model given earlier becomes:

$$q^n y[k] + \overline{a}_{n-1} q^{n-1} y[k] + \cdots + \overline{a}_0 y[k] = \overline{b}_m q^m u[k] + \cdots + \overline{b}_0 u[k]$$

For a discrete time system it is also possible to have discrete state space models. In the shift domain these models take the form:

$$qx[k] = \mathbf{A}_q x[k] + \mathbf{B}_q u[k]$$
$$y[k] = \mathbf{C}_q x[k] + \mathbf{D}_q u[k]$$

Z-Transform

Analogously to the use of Laplace Transforms for continuous time signals, we introduce the Z-transform for discrete time signals.

Consider a sequence $\{y[k]; k = 0, 1, 2, \ldots\}$. Then the Z-transform pair associated with $\{y[k]\}$ is given by

$$\mathcal{Z}\left[y[k]\right] = Y(z) = \sum_{k=0}^{\infty} z^{-k} y[k]$$

$$\mathcal{Z}^{-1}\left[Y(z)\right] = y[k] = \frac{1}{2\pi j} \oint z^{k-1} Y(z) dz$$

A table of Z-transforms of typical sequences is given in Table 12.1 (*see the next slide*).

Also, a table of Z-transform properties is given in Table 12.2 (*see the slide after next*).

Z-transform table

$f[k]$	$\mathcal{Z}[f[k]]$	Region of convergence
1	$\dfrac{z}{z-1}$	$\|z\| > 1$
$\delta_K[k]$	1	$\|z\| > 0$
k	$\dfrac{z}{(z-1)^2}$	$\|z\| > 1$
k^2	$\dfrac{z(z-1)}{(z-1)^3}$	$\|z\| > 1$
a^k	$\dfrac{z}{z-a}$	$\|z\| > \|a\|$
ka^k	$\dfrac{az}{(z-a)^2}$	$\|z\| > \|a\|$
$\cos k\theta$	$\dfrac{z(z-\cos\theta)}{z^2-2z\cos\theta+1}$	$\|z\| > 1$
$\sin k\theta$	$\dfrac{z\sin\theta}{z^2-2z\cos\theta+1}$	$\|z\| > 1$
$a^k\cos k\theta$	$\dfrac{z(z-a\cos\theta)}{z^2-2az\cos\theta+a^2}$	$\|z\| > a$
$a^k\sin k\theta$	$\dfrac{az\sin\theta)}{z^2-2az\cos\theta+a^2}$	$\|z\| > a$
$k\cos k\theta$	$\dfrac{z(z^2\cos\theta-2z+\cos\theta)}{z^2-2z\cos\theta+1}$	$\|z\| > 1$
$\mu[k]-\mu[k-k_o],\quad k_o\in\mathbb{N}$	$\dfrac{1+z+z^2+\ldots+z^{k_o-1}}{z^{k_o-1}}$	$\|z\| > 0$

Z-transform properties. Note that $F_i(z) = Z[f_i[k]]$, $\mu[k]$ denotes, as usual, a unit step, $y[\infty]$ must be well defined and the convolution property holds provided that $f_1[k] = f_2[k] = 0$ for all $k < 0$.

$f[k]$	$\mathcal{Z}[f[k]]$	Names
$\sum_{i=1}^{l} a_i f_i[k]$	$\sum_{i=1}^{l} a_i F_i(z)$	Partial fractions
$f[k+1]$	$zF(z)-zf(0)$	Forward shift
$\sum_{l=0}^{k} f[l]$	$\dfrac{z}{z-1}F(z)$	Summation
$f[k-1]$	$z^{-1}F(z)+f(-1)$	Backward shift
$y[k-l]\mu[k-l]$	$z^{-l}Y(z)$	Unit step
$kf[k]$	$-z\dfrac{dF(z)}{dz}$	
$\dfrac{1}{k}f[k]$	$\int_z^{\infty}\dfrac{F(\zeta)}{\zeta}d\zeta$	
$\lim_{k\to\infty} y[k]$	$\lim_{z\to 1}(z-1)Y(z)$	Final value theorem
$\lim_{k\to 0} y[k]$	$\lim_{z\to\infty} Y(z)$	Initial value theorem
$\sum_{l=0}^{k} f_1[l]f_2[k-l]$	$F_1(z)F_2(z)$	Convolution
$f_1[k]f_2[k]$	$\dfrac{1}{2\pi j}\oint F_1(\zeta)F_2\left(\dfrac{z}{\zeta}\right)\dfrac{d\zeta}{\zeta}$	Complex convolution
$(\lambda)^k f_1[k]$	$F_1\left(\frac{z}{\lambda}\right)$	Frequency scaling

How do we use Z-transforms ?

We saw earlier that Laplace Transforms have a remarkable property that they convert differential equations into algebraic equations.

Z-transforms have a similar property for discrete time models, namely they convert difference equations (expressed in terms of the shift operator q) into algebraic equations.

We illustrate this below for a discrete high-order difference equation model:

Discrete Transfer Functions

Taking Z-transforms on each side of the high order difference equation model leads to

$$A_q(z)Y_q(z) = B_q(z)U_q(z) + f_q(z, x_o)$$

where $Y_q(z)$, $U_q(z)$ are the Z-transform of the sequences $\{y[k]\}$ and $\{u[k]\}$ respectively, and

$$A_q(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_o$$

$$B_q(z) = b_m z^m + b_{m-1}z^{m-1} + \cdots + b_o$$

We then see that (*ignoring the initial conditions*) the Z-transform of the output $Y(z)$ is related to the Z-transform of the input by $Y(z) = G_q(z)U(z)$ where

$$G_q(z) \triangleq \frac{B_q(z)}{A_q(z)}$$

$G_q(z)$ is called the *discrete* (*shift form*) transfer function.

An interesting observation

We see from Table 12.1 that the Z-transform of a unit pulse is 1. Also, we have just seen that Z-transform of the output of discrete linear systems satisfies

$$Y(z) = G_q(z)\mathrm{U}(z)$$

where $G_q(z)$ is the transfer function and $U(z)$ the input.

Hence, the transfer function is the Z-transform of the output when the input is a Kronecker delta.

Example:

Find the unit step response of a system with transfer function given by

$$G_q(z) = \frac{0.5}{z + 0.8}$$

Solution: The Z-transform of the step response, $y[k]$, is given by

$$Y_q(z) = \frac{0.5}{z + 0.5} U_q(z) = \frac{0.5z}{(z + 0.5)(z - 1)}$$

Expanding in partial fractions (use MATLAB command **residue**) we obtain

$$Y_q(z) = \frac{z}{3(z-1)} - \frac{z}{3(z+0.5)} \Longleftrightarrow y[k] \qquad = \frac{1}{3}\left(1 - (-0.5)^k\right)\mu[k]$$

The response is shown on the next slide.

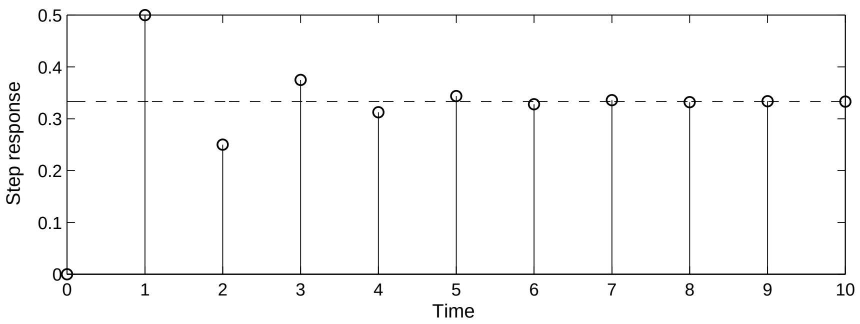


Figure 12.3: *Unit step response of a system exhibiting ringing response*

Note that the response contains the term $(-0.5)^k$, which corresponds to an oscillatory behavior (known as ringing). In discrete time this can occur (as in this example) for a single negative real pole whereas, in continuous time, a pair of complex conjugate poles are necessary to produce this effect.

Discrete Time Models

We next examine several properties of discrete time models, beginning with the issue of stability.

Discrete System Stability

Relationship to Poles

We have seen that the response of a discrete system (in the shift operator) to an input $U(z)$ has the form

$$Y(z) = G_q(z)U(z) + \frac{f_q(z;x_o)}{(z-\alpha_1)(z-\alpha_2)\cdots(z-\alpha_n)}$$

where $\alpha_1 \ldots \alpha_n$ are the poles of the system.

We then know, via a partial fraction expansion, that $Y(z)$ can be written as

$$Y(z) = \sum_{j=1}^{n} \frac{\beta_j z}{z-\alpha_j} + \text{terms depending on } U(z)$$

where, for simplicity, we have assumed non repeated poles.

The corresponding time response is

$$y[k] = \sum \beta_j [\alpha_j]^k + \text{terms depending on the input}$$

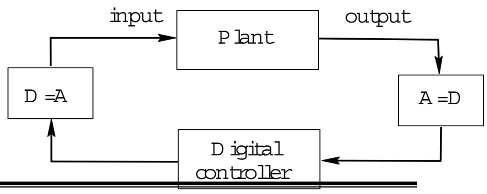
Stability requires that $[\alpha_j]^k \to 0$, which is the case if $[\alpha_j] < 1$.

Hence stability requires the poles to have magnitude less than 1, i.e. to lie inside a unit circle centered at the origin.

Discrete Models for Sampled Continuous Systems

So far in this lecture, we have assumed that the model is already given in discrete form. However, often discrete models arise by sampling the output of a continuous time system. We thus next examine how to obtain discrete time models which link the sampled output of a continuous time system to a sampled input.

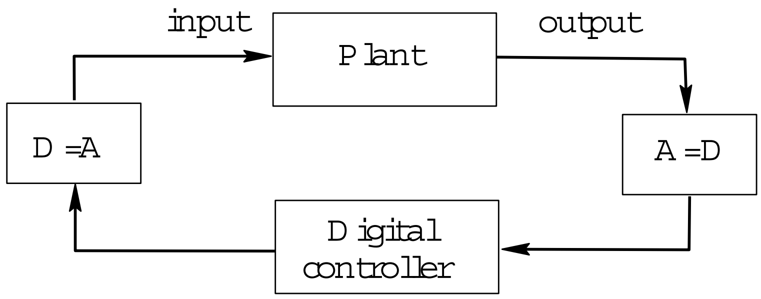
We are thus interested in modelling a continuous system operating under computer control.



A typical way of making this interconnection is shown on the next slide.

The analogue to digital converter (A/D in the figure) implements the process of sampling (at some fixed period Δ). The digital to analogue converter (D/A in the figure) interpolates the discrete control action into a function suitable for application to the plant input.

Digital control of a continuous time plant



Details of how the plant input is reconstructed

When a zero order hold is used to reconstruct $u(t)$, then

$$u(t) = u[k] \qquad \text{for} \quad k \qquad t < (k+1)$$

Note that this is the staircase signal shown earlier in Figure 12.2. Discrete time models typically relate the sampled signal $y[k]$ to the sampled input $u[k]$. Also a digital controller usually evaluates $u[k]$ based on $y[j]$ and $r[j]$, where $\{r(k\Delta)\}$ is the reference sequence and $j \le$ k.

Using Continuous Transfer Function Models

We observe that the generation of the staircase signal $u(t)$, from the sequence $\{u(k)\}$ can be modeled as in Figure 12.5.

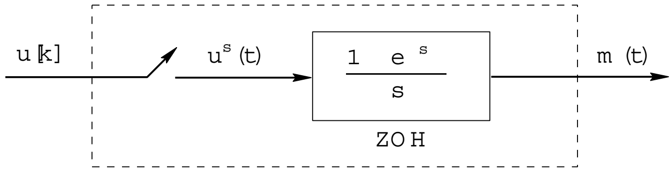
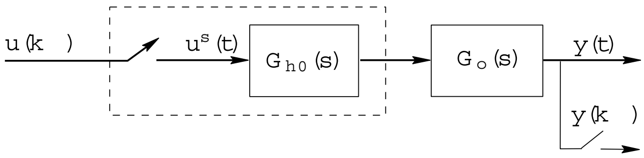


Figure 12.5: *Zero order hold*

Figure 12.6: *Discrete time equivalent model with zero order hold*

Combining the circuit on the previous slide with the plant transfer function $G_0(s)$, yields the equivalent connection between input sequence, $u(k\Delta)$, and sampled output $y(k\Delta)$ as shown below:



We saw earlier that the transfer function of a discrete time system, in Z-transform form is the Z-transform of the output (the sequence $\{y[k]\}$) when the input, $u[k]$, is a Kronecker delta, with zero initial conditions. We also have, from the previous slide, that if $u[k] = \delta_K[k]$, then the input to the continuous plant is a Dirac Delta, i.e. $u^s(t) = \delta(t)$. If we denote by $H_{eq}(z)$ the transfer function from $U_q(z)$ to $Y_q(z)$, we then have the following result.

$$H_{oq}(z) = \mathcal{Z}\ [\text{the sampled impulse response of } G_{h0}(s)G_o(s)]$$
$$= (1 \quad z^{\ 1})\mathcal{Z}\ [\text{the sampled step response of } G_o(s)]$$

Example 12.10

Consider the example used as motivation at the beginning of this lecture. The continuous time transfer function is

$$G_o(s) = \frac{b_0}{s(s + a_0)}$$

Using the result on the previous slide we see that

$$H_{oq}(z) = \frac{(z \quad 1)}{z}\mathcal{Z}\ \frac{b_0}{a_0}(k \quad) \quad \frac{b_0}{a_0^2} + \frac{b_0}{a_0^2}e^{\quad k}$$
$$= \frac{(z \quad 1)}{a_0^2}\ \frac{a_0 b_0 z}{(z \quad 1)^2} \quad \frac{b_0 z}{z \quad 1} + \frac{b_0}{z \quad e^{\ a_0}}$$
$$= \frac{b_0 a_0 \quad + b_0 e^{\ a_0} \quad b_0 \ \ z \quad b_0 a_0 \ \ e^{\ a_0} \quad b_0 e^{\ a_0} \quad + b_0}{a_0^2 (z \quad 1)(z \quad e^{\ a_0} \quad)}$$

This model is of the form:

$$y\left(\overline{k+1\Delta}\right)=\overline{a}_1\,y\left(\overline{k\Delta}\right)+\overline{a}_0\,y\left(\overline{k-1\Delta}\right)+\overline{b}_1u\left(\overline{k\Delta}\right)+\overline{b}_2u\left(\overline{k-1\Delta}\right)$$

Note that this is a second order transfer function with a first order numerator.

The reader may care to check that this is consistent with the input-output model, *i.e.*

$$H_{0q}(z)=\frac{\overline{b}_1z+\overline{b}_0}{Z^2-\overline{a}_1Z-\overline{a}_0}$$

We have thus fulfilled one promise of showing where this model comes from.

Frequency Response of Sampled Data Systems

We evaluate the frequency response of a linear discrete time system having transfer function $H_q(z)$. Consider a sine wave input given by

$$u(k\ ) = \sin(!k\ ) = \sin\ 2\ k\frac{!}{!_s} = \frac{1}{2j}\ e^{j2\ k\frac{!}{!_s}}\quad e^{\ j2\ k\frac{!}{!_s}}$$

where $\omega_s = \frac{2\pi}{\Delta}$.

Following the same procedure as in the continuous time case (see Lecture 4) we see that the system output response to the input is

$$y(k\ ) = \ (!)\sin(!k\ +\ (!))$$

where

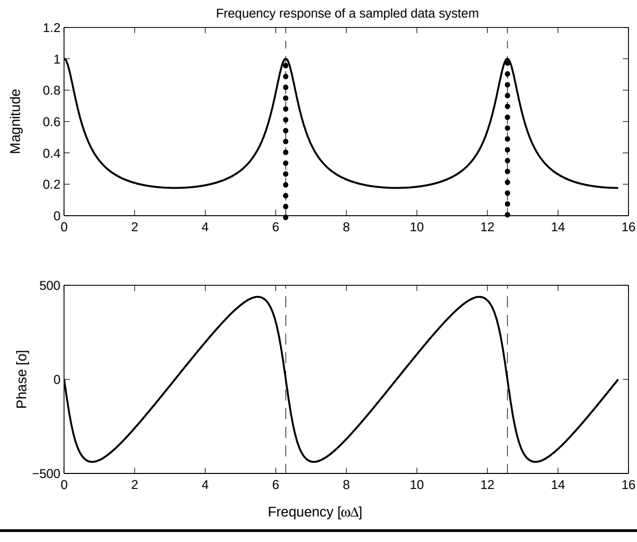
$$H_q(e^{j!}\ ) = \ (!)e^{j\ (!)}$$

The frequency response of a discrete time system depends upon $e^{j\omega\Delta}$ and is thus periodic in ω with period $2\pi/_{\Delta}$.

The next slide illustrates this fact by showing the frequency response of

$$H_q[z] = \frac{0.3}{z - 0.7}$$

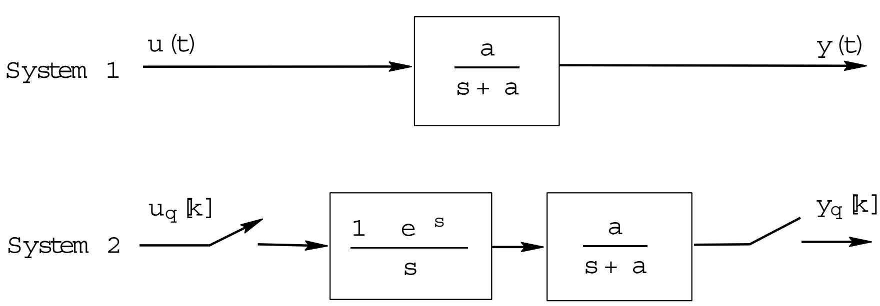
Figure 12.7: *Periodicity in the frequency response of sampled data systems.*



Another feature of particular interest is that the sampled data frequency response converges to its continuous counterpart as $\Delta \rightarrow 0$ and hence much insight can be obtained by simply looking at the continuous version. This is exemplified below.

Example 12.11: Consider the two systems shown in Figure 12.8 on the next slide: Compare the frequency response of both systems in the range $[0, \omega_s]$.

Figure 12.8: *Continuous and sampled data systems*



The continuous time transfer function

$$H(s) = \frac{a}{s+a}$$

The continuous and discrete frequency responses are:

$$H(j\omega) = \frac{Y(j\omega)}{U(j\omega)} = \frac{a}{j\omega + a}$$

$$H_q\left(e^{j\omega\Delta}\right) = \frac{Y_q\left(e^{j\omega\Delta}\right)}{U_q\left(e^{j\omega\Delta}\right)} = Z\left\{G_{h0}(s)\frac{a}{s+a}\right\}\Bigg|_{z=e^{j\omega\Delta}} = \frac{1-e^{-a\Delta}}{e^{j\omega\Delta} - e^{-a\Delta}}$$

Note that for $\omega << \omega_s$ and $a << \omega_s$ *i.e.* $\omega\Delta << 1$ and $a\Delta << 1$, then we can use a first order Taylor's series approximation for the exponentials $e^{-a\Delta}$ and $e^{j\omega\Delta}$ in the discrete case leading to

$$H_q(j\omega\ ) \quad \frac{1 \quad 1+a}{1+j\omega \qquad 1+a} = \frac{a}{j\omega + a} = H(j\omega)$$

The next slide compares the two frequency responses as a function of input frequency for two different values of Δ. Note that for Δ small, the two frequency responses are very close.

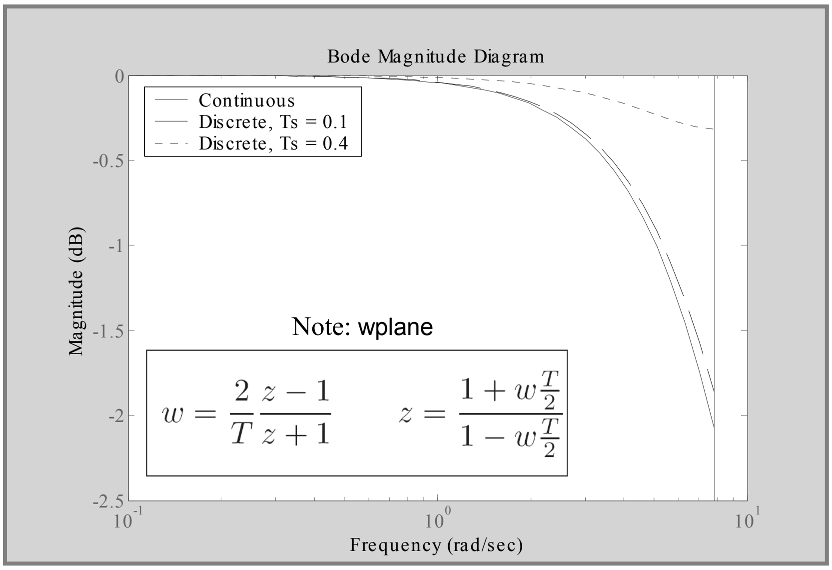


Figure 12.9: *Asymptotic behavior of a sampled data transfer function*

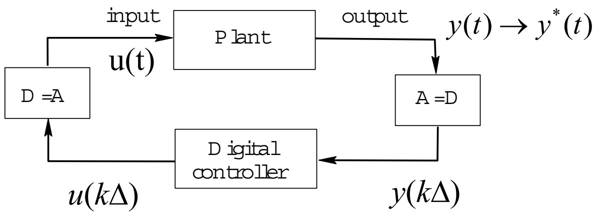
Example of Plant Control

We re-consider the control of the general system via a computer. This is the simple example of:

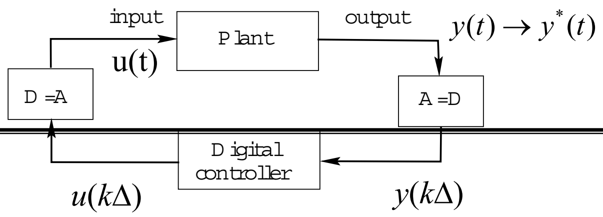
- Continuous time system: $G_o(s) = \frac{b_0}{s(s + a_0)}$
- Discrete time system (HE): $\boxed{H_{0q}(z) = \frac{\bar{b}_1 z + \bar{b}_0}{z^2 - \bar{a}_1 z - \bar{a}_0}}$
- Difference equation:

$$\boxed{y(\overline{k+1}\Delta) = \bar{a}_1 y(\bar{k}\Delta) + \bar{a}_0 y(\overline{k-1}\Delta) + \bar{b}_1 u(\bar{k}\Delta) + \bar{b}_2 u(\overline{k-1}\Delta)}$$

The set-up for digital control of this system is shown schematically below:



The objective is to cause the output $y(t)$, to follow a given reference signal, $y^*(t)$.

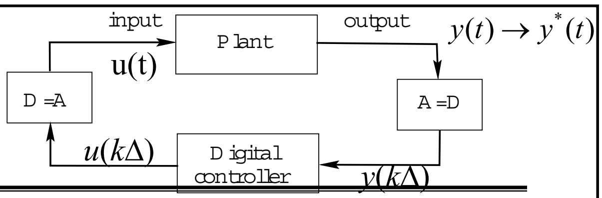


It was shown that the output at time $k\Delta$ can be modelled as a linear function of past outputs and past controls.

Thus the (*discrete time*) model for the control system takes the form:

$$y\left(\overline{k+1}\Delta\right)=\bar{a}_1 y\left(k\Delta\right)+\bar{a}_0 y\left(\overline{k-1}\Delta\right)+\bar{b}_1 u\left(k\Delta\right)+\bar{b}_0 u\left(\overline{k-1}\Delta\right).$$

A Prototype Control Law



Conceptually, we want $y\left(\overline{k+1}\Delta\right)$ to go to the desired value y^*. This suggests that we could simply set the right hand side of the equation

$$y\left(\overline{k+1}\Delta\right) = \overline{a}_1 y(k\Delta) + \overline{a}_0 y\left(\overline{k-1}\Delta\right) + \overline{b}_1 u(k\Delta) + \overline{b}_0 u\left(\overline{k-1}\Delta\right).$$

equal to y^*. Doing this we see that $u(k\Delta)$ becomes a function of $y(k\Delta)$ (as well as $y\left(\overline{k-1}\Delta\right)$ and $u\left(\overline{k-1}\Delta\right)$. At first glance this looks reasonable but on reflection we have left no time to make the necessary calculations. Thus, it would be better if we could reorganize the control law so that $u(k\Delta)$ becomes a function of $y\left(\overline{k-1}\Delta\right)$, … . Actually this can be achieved by changing the model slightly as we show on the next slide.

Model Development

Substituting the model into itself to yield:

$$\begin{aligned} y\left(\overline{k+1}\Delta\right) &= \overline{a}_1 \{y(k\Delta)\} + \overline{a}_0 y\left(\overline{k-1}\Delta\right) \\ &\quad + \overline{b}_1 u(k\Delta) + \overline{b}_0 u\left(\overline{k-1}\Delta\right) \\ &= \overline{a}_1 \{\overline{a}_1 y\left(\overline{k-1}\Delta\right) + \overline{a}_0 y\left(\overline{k-2}\Delta\right) \\ &\quad + \overline{b}_1 u\left(\overline{k-1}\Delta\right) + \overline{b}_0 u\left(\overline{k-2}\Delta\right)\} \\ &\quad + \overline{a}_0 y\left(\overline{k-1}\Delta\right) + \overline{b}_1 u(k\Delta) + \overline{b}_0 u\left(\overline{k-1}\Delta\right) \end{aligned}$$

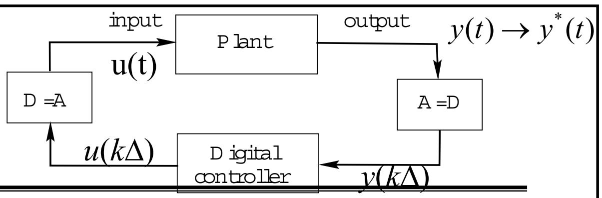
We see that $y\left(\overline{k+1}\Delta\right)$ takes the following form:

$$y\left(\overline{k+1}\Delta\right)=\alpha_1 y\left(\overline{k-1}\Delta\right)+\alpha_2 y\left(\overline{k-2}\Delta\right)$$
$$+\beta_1 u\left(\overline{k}\Delta\right)+\beta_2 u\left(\overline{k-1}\Delta\right)+\beta_3 u\left(\overline{k-2}\Delta\right)$$

where $\alpha_1 = \overline{a}_1^2 + \overline{a}_0$ etc.

Actually, $\alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3$, can be estimated from the physical system. We will not go in to details here.

A Modified Prototype Control Law

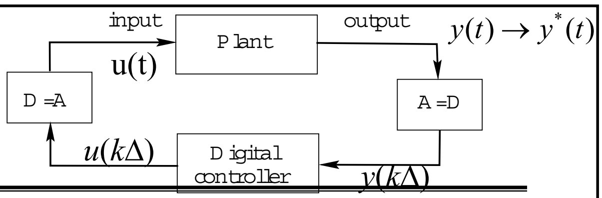


Now we want the output to go to the reference y^*.

Recall we have the model:

$$y\left(\overline{k+1}\Delta\right)=\alpha_1 y\left(\overline{k-1}\Delta\right)+\alpha_2 y\left(\overline{k-2}\Delta\right)$$
$$+\beta_1 u\left(\overline{k}\Delta\right)+\beta_2 u\left(\overline{k-1}\Delta\right)+\beta_3 u\left(\overline{k-2}\Delta\right)$$

This suggests that all we need do is set $y\left(\overline{k+1}\Delta\right)$ equal to the desired set-point $y^*\left(\overline{k+1}\Delta\right)$ and solve for $u(k\Delta)$. The answer is



$$u(k\Delta) = \frac{y^*\left(\overline{k+1}\Delta\right) - \alpha_1 y\left(\overline{k-1}\Delta\right) - \alpha_2 y\left(\overline{k-2}\Delta\right) - \beta_2 u\left(\overline{k-1}\Delta\right) - \beta_3 u\left(\overline{k-2}\Delta\right)}{\beta_1}$$

Notice that the above control law expresses the current control $u(k\Delta)$ as a function of

- the reference, $y^*\left(\overline{k+1}\Delta\right)$
- past output measurements, $y\left(\overline{k-1}\Delta\right), y\left(\overline{k-2}\Delta\right)$
- past control signals, $u\left(\overline{k-1}\Delta\right), u\left(\overline{k-2}\Delta\right)$

Also notice that 1 sampling interval exists between the measurement of $y\left(\overline{k-1}\Delta\right)$ and the time needed to apply $u(k\Delta)$; *i.e.* we have specifically allowed time for the computation of $u(k\Delta)$ to be performed after $y\left(\overline{k+1}\Delta\right)$ is measured!

$$u(k\Delta) = \frac{y^*\left(\overline{k+1}\Delta\right) - \alpha_1 y\left(\overline{k-1}\Delta\right) - \alpha_2 y\left(\overline{k-2}\Delta\right) - \beta_2 u\left(\overline{k-1}\Delta\right) - \beta_3 u\left(\overline{k-2}\Delta\right)}{\beta_1}$$

Recap

All of this is very plausible so far. We have obtained a simple *digital* control law which causes $y\left(\overline{k+1}\Delta\right)$ to go to the desired value $y^*\left(\overline{k+1}\Delta\right)$ in one step !

Of course, the real system evolves in continuous time (*readers may care to note this point for later consideration*).

Sampled Data Systems

Experimental results

However, when we try this on a real system, the results are extremely poor! Indeed, the system essentially goes unstable.

- Can the reader guess some of the causes for the difference between the ideal simulation results and the very poor real results?

Causes of the Poor Response

It turns out that there are many reasons for the poor response. Some of these are:

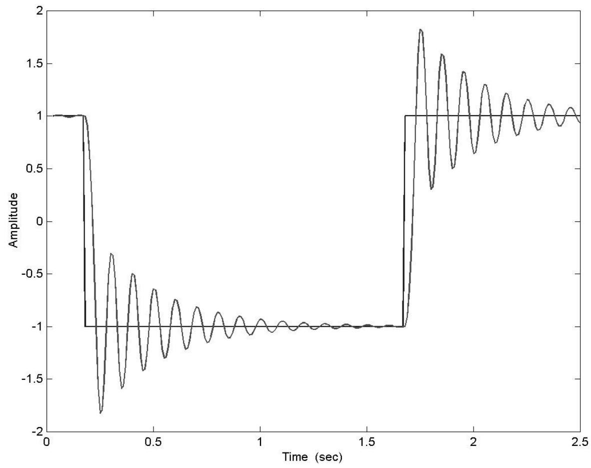
1. *Intersample issues*
2. *Noise*
3. *Timing jitter*

The purpose of this chapter is to understand these issues. To provide motivation for the reader we will briefly examine these issues for a simple example.

1. Intersample Issues

If we look at the output response at a rate faster than the control sampling rate then we see that the actual response is as shown on the next slide.

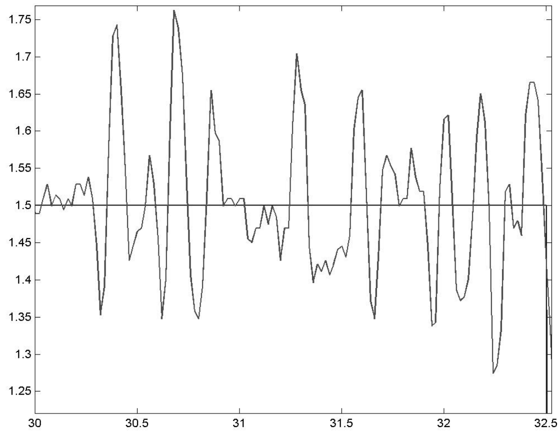
Simulation result showing full continuous output response



This is rather surprising! However, if we think back to the original question, we only asked that the sampled output go to the desired reference. Indeed it has. However, we said nothing about the intersample response!

2. Noise

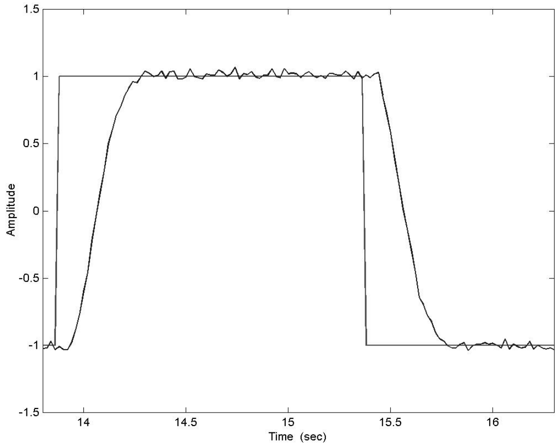
One further point that we have overlooked is that causing $y(t)$ to approach y^* as quickly as possible gives a very wide bandwidth controller. However, it should be clear that such a controller will necessarily magnify noise. Indeed, if we look at the steady response of the system (*see the next slide*) then we can see that noise is indeed causing problems.



3. Timing Jitter

Finally we realize that this particular real controller has been implemented in a computer that does not have a real-time operating system. This means that the true sampling rate actually varies around the design value. We call this *timing jitter*. This can be thought of as introducing modelling errors. Yet we are using a wideband controller. Thus, we should expect significant degradation in performance relative to the idealized simulations.

Finally, we make a much less demanding design and try a simple digital PID controller on the real system. The results are entirely satisfactory as can be seen on the next slide. Of course, the design bandwidth is significantly less than was attempted with the previous design.



Hopefully the above example has motivated the reader to say - "*let's study digital control*".

By the time you have studied the next points you should have understood the features of the above simple problem, *e.g.*

- how to build the digital control model;
- what are the special features of the one-step-ahead control law we have used; and
- why funny things can (*and sometimes do*) happen between samples.

Summary

- Very few plants encountered by the control engineer are digital, most are continuous. That is, the control signal applied to the process, as well as the measurements received from the process, are usually continuous time.
- Modern control systems, however, are almost exclusively implemented on digital computers.
- Compared to the historical analog controller implementation, the digital computer provides
  - much greater ease of implementing complex algorithms,
  - convenient (graphical) man-machine interfaces,
  - logging, trending and diagnostics of internal controller and
  - flexibility to implement filtering and other forms of signal processing operations.

- Digital computers operate with sequences in time, rather than continuous functions in time.
  Therefore,
  - input signals to the digital controller-notably process measurements - must be sampled;
  - outputs from the digital controller-notably control signals - must be interpolated from a digital sequence of values to a continuous function in time.
- Sampling (see next slide) is carried out by A/D (analog to digital converters.

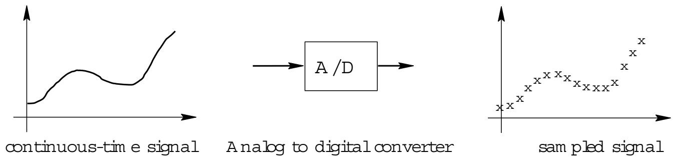


Figure 12.10: *The result of sampling*

- The converse, reconstructing a continuous time signal from digital samples, is carried out by D/A (digital to analog) converters. There are different ways of interpolating between the discrete samples, but the so called zero-order hold (see next slide) is by far the most common.

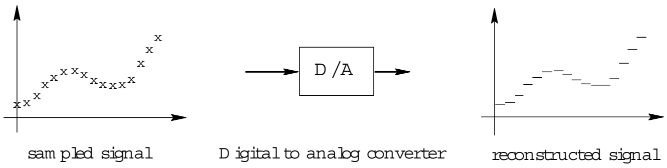


Figure 12.11: *The result of reconstruction*

- When sampling a continuous time signal,
  - an appropriate sampling rate must be chosen
  - an anti-aliasing filter (low-pass) should be included to avoid frequency folding.
- Analysis of digital systems relies on discrete time versions of the continuous operators.

- The chapter has introduced the discrete operator:
  - the shift operator, q, defined by $qx[k] \triangleq x[k+1]$
- The shift operator, q,
  - is the traditional operator;
  - is the operator many engineers feel more familiar with;
  - is used in the majority of the literature.

- Analysis of digital systems relies on discrete time versions of the continuous operators:
  - the discrete version of the differential operator is difference operator;
  - the discrete version of the Laplace Transform is the Z-transform (associated with the shift operator).
- With the help of these operators,
  - continuous time differential equation models can be converted to discrete time difference equation models;
  - continuous time transfer or state space models can be converted to discrete time transfer or state space models in either the shift or δ operators.