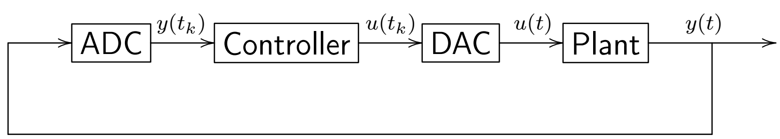
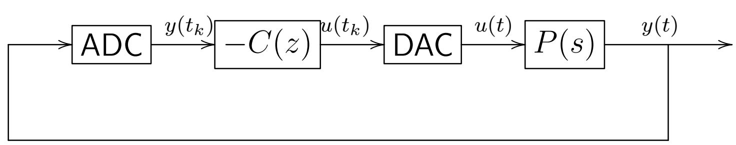
8 Digital Control Systems

8.1 Background of Digital Control Systems

Practically all control systems are implemented on digital computers, meaning that the controller uses *sampled* output of the plant and periodically computes a sequence of commands $\{u[k]\} \triangleq \{u(t_k)\}$ $(k = 0, 1, 2, \dots)$, instead of directly generating a continuous signal $u(t)$. The next block diagram illustrates such a control implementation schemes. The function block that converts $y(t)$ to $y[k] = y(t_k)$ $(k = 0, 1, 2, \dots)$ is called an analog-to-digital converter (ADC). The block that converts $u(t_k)$ to $u(t)$ is called a digital-to analog converter (DAC).

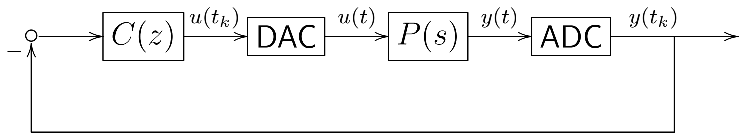


For linear time invariant plants and controllers, the plant and controller can be represented as transfer functions. The block diagram can then be simplified to:



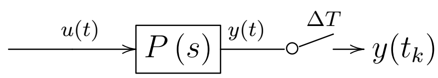
where by convention of negative feedback control, we have added the negative sign in front of the digital controller.

Computer-controlled systems are also called *sampled-data systems*. The mixture of continuous- and discrete-time signals and systems causes multiple difficulties in analysis. Often, it is sufficient to understand and control the behavior of the system at the sampling instances. In that case, the previous block diagram can be re-ordered to



If only the signals at the discrete sampling instances are of interest, the system is called a *discrete-time system*.

For now, we will simply treat the ADC as a sampler. The block diagram can then be represented as:



The most widely used DAC in practice is called a zero order holder (ZOH). Fig. 4 illustrates the idea of the ZOH and digital sampler. Between the discrete-time indices k and $k+1$, the ZOH holds the value of $u(t_k)$; and only the output values $y(t_k)$ $(k = 0, 1, 2, \dots)$ are actually measured and used in the closed-loop system.

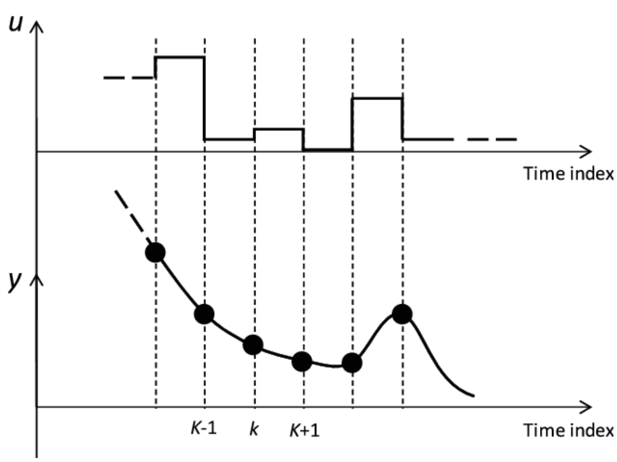


Figure 4 – Discrete-time sampled-data input and output

8.2 Discretization of Continuous-time Systems

Consider the discrete-time controllor implementation scheme where $u(k)$ and

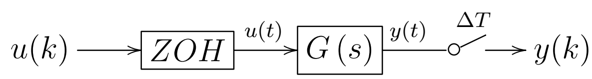


Figure 5 – ZOH-based discrete-time controllor implementation scheme

$y(k)$ have the same sampling time.

To derive the transfer function from $u(k)$ to $y(k)$, we let $u(k)$ be a discrete-time impulse (whose Z transform is 1) and obtain the Z transform of $y(k)$. As $u(k) = 1$ for $k = 0$ and $u(k) = 0$ otherwise, after the zero order hold,

$$u(t) = \begin{cases} 1, & 0 \leq t < \Delta T \\ 0, & \text{otherwise} \end{cases}$$

The Laplace transform of this signal is

$$U(s) = \frac{1 - e^{-s\Delta T}}{s}$$

Hence

$$y(t) = \mathcal{L}^{-1}\left[G(s)\frac{1 - e^{-s\Delta T}}{s}\right] = \mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right] - \mathcal{L}^{-1}\left[G(s)\frac{e^{-s\Delta T}}{s}\right]$$

Sampling this continuous-time signal at ΔT, and performing the Z transform gives:

$$\begin{aligned} G(z) &= \mathcal{Z}\left\{\mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right]\bigg|_{t=k\Delta T} - \mathcal{L}^{-1}\left[G(s)\frac{e^{-s\Delta T}}{s}\right]\bigg|_{t=k\Delta T}\right\} \\ &= \mathcal{Z}\left\{\mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right]\bigg|_{t=k\Delta T}\right\} - z^{-1}\mathcal{Z}\left\{\mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right]\bigg|_{t=k\Delta T}\right\} \end{aligned}$$

where the last equality holds because $e^{-s\Delta T}$ corresponds to exactly one step of discrete-time delay.

Fact 1. *The transfer function from $u(k)$ to $y(k)$ in Fig. 5 is*

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\mathcal{L}^{-1}\left[G(s)\frac{1}{s}\right]\Big|_{t=k\Delta T}\right\}$$

where ΔT is the sampling time.

Example 12. Obtain the ZOH equivalent of

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

Following the discretization procedures we have

$$\frac{G(s)}{s} = \frac{a}{s(s+a)} = \frac{1}{s} - \frac{1}{s+a}$$

and hence

$$\mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\} = 1(t) - e^{-at}1(t)$$

Sampling at ΔT gives $1(k\Delta T) - e^{-ak\Delta T}1(k\Delta T)$, whose Z transform is (from the table of Z transform)

$$\frac{z}{z-1} - \frac{z}{z - e^{-a\Delta T}} = \frac{z(1 - e^{-a\Delta T})}{(z-1)(z - e^{-a\Delta T})}$$

Hence the ZOH equivalent is

$$(1 - z^{-1})\frac{z(1 - e^{-a\Delta T})}{(z-1)(z - e^{-a\Delta T})} = \frac{1 - e^{-a\Delta T}}{z - e^{-a\Delta T}}$$

In MATLAB, the function *c2d.m* computes the ZOH equivalent of a continuous-time transfer function, as well as other discrete equivalents. For

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

and $\Delta T = 1$, the following scripts

```
T=1;
numG=1; denG=[1 0 0];
G = tf(numG,denG);
Gd = c2d(G,T);
```

produces the correct ZOH equivalent. As an exercise, you should derive the analytic formula and verify the MATLAB result.

Exercise 8. Find the zero order hold equivalent of $G(s) = e^{-Ls}$, $2\Delta T < L < 3\Delta T$, where ΔT is the sampling time.

Discretization and Implementation of Continuous-time Design

Big picture
Discrete-time frequency response
Discretization of continuous-time design
Aliasing and anti-aliasing

Outline

1. Big picture

2. Discrete-time frequency response

3. Sampling and aliasing

4. Approximation of continuous-time controllers

Big picture

why are we learning this:

- the majority of controllers are implemented in discrete-time domain
- implementation media: digital signal processor, field-programmable gate array (FPGA), etc
- either: controller is designed in continuous-time domain and implemented digitally
- or: controller is designed directly in discrete-time domain

Frequency response of LTI SISO digital systems

$$a\sin(\omega T_s k) \longrightarrow \boxed{G(z)} \longrightarrow b\sin(\omega T_s k + \phi) \text{ at steady state}$$

- sampling time: T_s
- $\phi\left(e^{j\omega T_s}\right)$: phase difference between the output and the input
- $M\left(e^{j\omega T_s}\right) = b/a$: magnitude difference

continuous-time frequency response:

$$G(j\omega) = G(s)|_{s=j\omega} = |G(j\omega)|\, e^{j\angle G(j\omega)}$$

discrete-time frequency response:

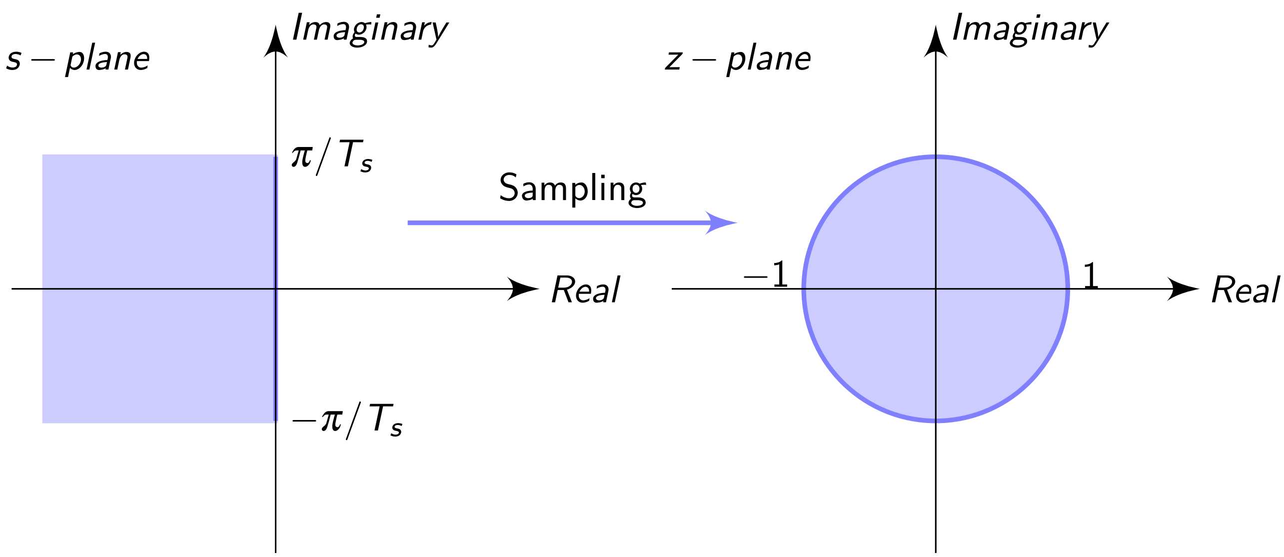
$$G\left(e^{j\omega T_s}\right) = G(z)|_{z=e^{j\omega T_s}} = \left|G\left(e^{j\omega T_s}\right)\right| e^{j\angle G\left(e^{j\omega T_s}\right)}$$
$$= M\left(e^{j\omega T_s}\right) e^{j\phi\left(e^{j\omega T_s}\right)}$$

Sampling and aliasing

sampling maps the continuous-time frequency

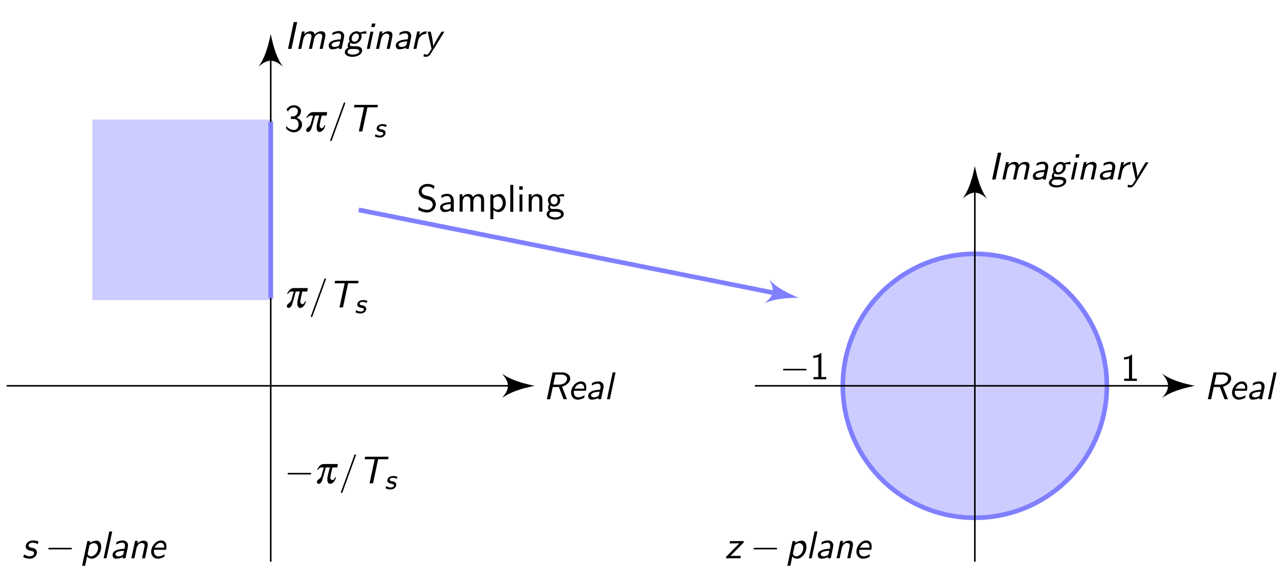
$$-\frac{\pi}{T_s} < \omega < \frac{\pi}{T_s}$$

onto the unit circle



Sampling and aliasing

sampling also maps the continuous-time frequencies $\frac{\pi}{T_s} < \omega < 3\frac{\pi}{T_s}$, $3\frac{\pi}{T_s} < \omega < 5\frac{\pi}{T_s}$, etc, onto the unit circle



Shannon's Sampling Theorem

sufficient samples must be collected (i.e., fast enough sampling frequency) to recover the frequency of a continuous-time sinusoidal signal (with frequency ω in rad/sec)

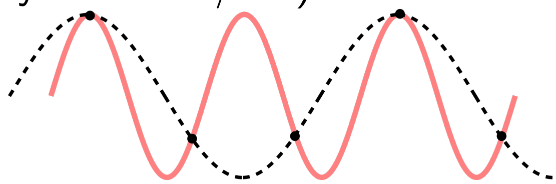


Figure: Sampling example (source: Wikipedia.org)

- the sampling frequency $= \frac{2\pi}{T_s}$
- Shannon's sampling theorem: the Nyquist frequency ($\triangleq \frac{\pi}{T_s}$) must satisfy

$$-\frac{\pi}{T_s} < \omega < \frac{\pi}{T_s}$$

Sampling and aliasing

Example (Sampling and Aliasing)

T_s=1/60 sec (Nyquist frequency 30 Hz).
a continuous-time 10-Hz signal [10 Hz$\leftrightarrow 2\pi\times 10$ rad/sec $\in(-\pi/T_s,\pi/T_s)$]

$$y_1(t) = \sin(2\pi\times 10t)$$

is sampled to

$$y_1(k) = \sin\left(2\pi\times\frac{10}{60}k\right) = \sin\left(2\pi\times\frac{1}{6}k\right)$$

a 70-Hz signal [$2\pi\times 70$ rad/sec $\in(\pi/T_s,3\pi/T_s)$]

$$y_2(t) = \sin(2\pi\times 70t)$$

is sampled to

$$y_2(k) = \sin\left(2\pi\times\frac{70}{60}k\right) = \sin\left(2\pi\times\frac{1}{6}k\right) \equiv y_1(k)!$$

Anti-aliasing

need to avoid the negative influence of *aliasing* beyond the Nyquist frequencies

- sample faster: make π/T_s large; the sampling frequency should be high enough for good control design
- anti-aliasing: perform a low-pass filter to filter out the signals $|\omega| > \pi/T_s$

Approximation of continuous-time controllers

bilinear transform

formula:

$$\boxed{s = \frac{2}{T_s}\frac{z-1}{z+1} \qquad z = \frac{1+\frac{T_s}{2}s}{1-\frac{T_s}{2}s}} \tag{1}$$

intuition:

$$z = e^{sT_s} = \frac{e^{sT_s/2}}{e^{-sT_s/2}} \overset{\text{1st-order Taylor Expansion}}{\approx} \frac{1+\frac{T_s}{2}s}{1-\frac{T_s}{2}s}$$

implementation: start with $G(s)$, obtain the discrete implementation

$$G_d(z) = G(s)|_{s=\frac{2}{T_s}\frac{z-1}{z+1}} \tag{2}$$

Exercise: Show that bilinear transformation maps the closed left half s-plane to the closed unit ball in z-plane

Stability reservation: $G(s)$ stable $\Longleftrightarrow$ $G_d(z)$ stable

Approximation of continuous-time controllers

history

Bilinear transform is also known as Tustin transform.
Arnold Tustin (16 July 1899 – 9 January 1994):

- British engineer, Professor at University of Birmingham and at Imperial College London
- served in the Royal Engineers in World War I
- worked a lot on electrical machines

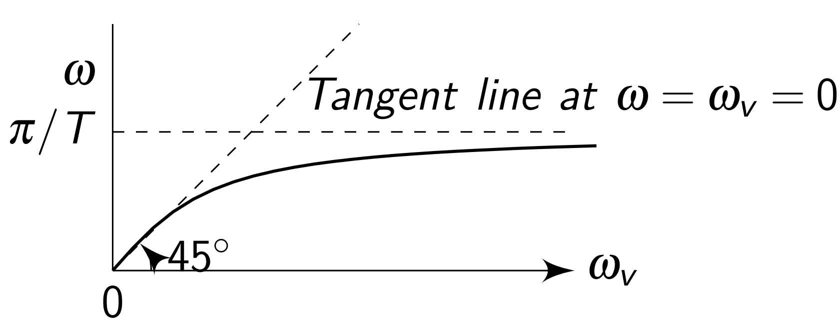
*Approximation of continuous-time controllers

frequency mismatch in bilinear transform

$$\left.\frac{2}{T_s}\frac{z-1}{z+1}\right|_{z=e^{j\omega T_s}}=\frac{2}{T_s}\frac{e^{j\omega T_S/2}\left(e^{j\omega T_S/2}-e^{-j\omega T_S/2}\right)}{e^{j\omega T_S/2}\left(e^{j\omega T_S/2}+e^{-j\omega T_S/2}\right)}=j\overbrace{\frac{2}{T_s}\tan\left(\frac{\omega T_s}{2}\right)}^{\omega_v}$$

$G(s)|_{s=j\omega}$ is the true frequency response at ω; yet bilinear implementation gives,

$$G_d\left(e^{j\omega T_s}\right)=G(s)|_{s=j\omega_v}\neq G(s)|_{s=j\omega}$$



*Approximation of continuous-time controllers

bilinear transform with prewarping

goal: extend bilinear transformation such that

$$G_d(z)\big|_{z=e^{j\omega T_s}} = G(s)\big|_{s=j\omega}$$

at a particular frequency ω_p
solution:

$$\boxed{s = p\frac{z-1}{z+1}, \qquad z = \frac{1+\frac{1}{p}s}{1-\frac{1}{p}s}, \qquad p = \frac{\omega_p}{\tan\left(\frac{\omega_p T_s}{2}\right)}}$$

which gives

$$G_d(z) = G(s)\big|_{s=\frac{\omega_p}{\tan\left(\frac{\omega_p T}{2}\right)}\frac{z-1}{z+1}}$$

and

$$\left.\frac{\omega_p}{\tan\left(\frac{\omega_p T_s}{2}\right)}\frac{z-1}{z+1}\right|_{z=e^{j\omega_p T_s}} = j\frac{\omega_p}{\cancel{\tan\left(\frac{\omega_p T_s}{2}\right)}}\cancel{\tan\left(\frac{\omega_p T_s}{2}\right)}$$

*Approximation of continuous-time controllers

bilinear transform with prewarping

choosing a prewarping frequency ω_p:

- must be below the Nyquist frequency:

$$0 < \omega_p < \frac{\pi}{T_s}$$

- standard bilinear transform corresponds to the case where $\omega_p = 0$
- the best choice of ω_p depends on the important features in control design

example choices of ω_p:

- at the cross-over frequency (which helps preserve phase margin)
- at the frequency of a critical notch for compensating system resonances