Essentials of Control Systems

Discretization and Implementation of Continuous-time Design

Xu Chen

University of Washington

ver. May 26, 2020

Outline

1. Big picture

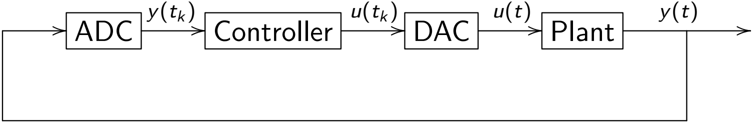
2. Discrete-time frequency response

3. Approximation of continuous-time controllers

4. Sampling and aliasing

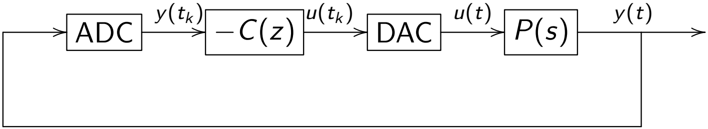
Background of Digital Control Systems

- Practically all control systems are implemented on digital computers: 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 $u(t)$. e.g.,

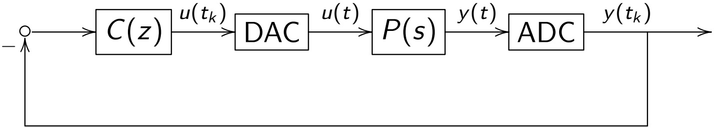


- analog-to-digital converter (ADC): converts $y(t)$ to $y[k] = y(t_k)$ $(k = 0, 1, 2, \dots)$
- digital-to analog converter (DAC): converts $u(t_k)$ to $u(t)$
- the overall system is known as a *sampled-data system*

From Sampled-Data to Discrete-Time Systems



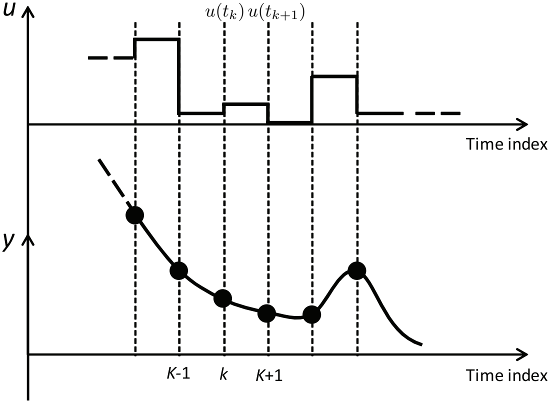
- Mixing continuous- and discrete-time signals and systems causes multiple difficulties in analysis.
- Often, it is sufficient to understand and control the behavior of the system at the sampling instances.
- Then the previous block diagram can be re-ordered to



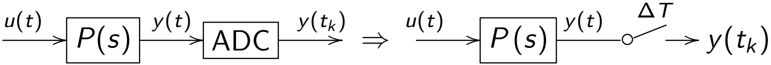
If only the signals at t_k's are of interest, the system is called a *discrete-time system*.

ADC and DCA

- the most widely used DAC is the zero order holder (ZOH)



- we often treat the ADC as a sampler



Big picture

- implementation platform of digital control: 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:

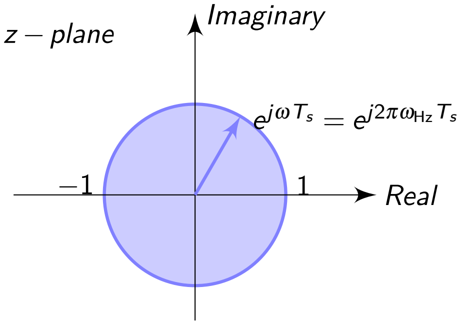
$$G(j\omega) = G(s)|_{s=j\omega} = |G(j\omega)|\, e^{j\angle G(j\omega)}$$

discrete-time frequency response:

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

The units of frequency

- sampling time: T_s
- $G(j\omega)$: default unit of ω is radians/second or rad/sec in Matlab
- $G\left(e^{j\omega T_s}\right)$: ω is in rad/sec; $\Omega \triangleq \omega T_s$ in radians or randians/sample
- Hz as the other common unit of frequency: $\underbrace{\omega}_{\text{in rad/s}} = 2\pi\omega_{\text{Hz}}$
- a little abuse of notation in literature: $\sin(\omega k)$ as a discrete signal (ω in radians/sample here) and $G\left(e^{j\omega}\right)$ also used for discrete0time frequency response



Sampling

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)

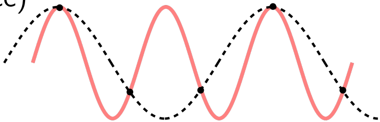


Figure: Sampling example (source: Wikipedia.org)

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

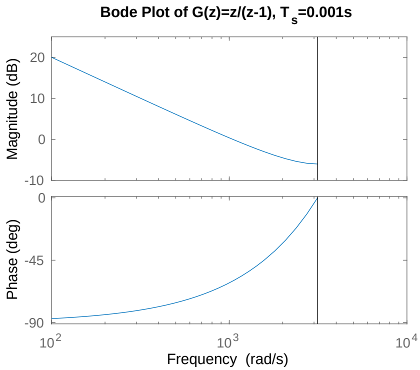
$$-\frac{\pi}{T_s} < \omega < \frac{\pi}{T_s}$$

Sampling

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

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

- The Nyquist frequency is the maximum frequency in the bode plot of a discrete-time system in Matlab: e.g.,



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:

$$z = \mathrm{e}^{sT_s} = \frac{\mathrm{e}^{sT_s/2}}{\mathrm{e}^{-sT_s/2}} \approx \frac{1+\frac{T_s}{2}s}{1-\frac{T_s}{2}s}$$

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

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

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

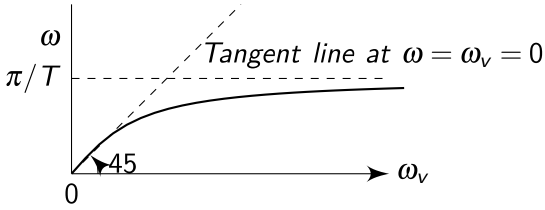
Approximation of continuous-time controllers

frequency mismatch in bilinear transform

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

$G(s)|_{s=j\omega}$ 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:

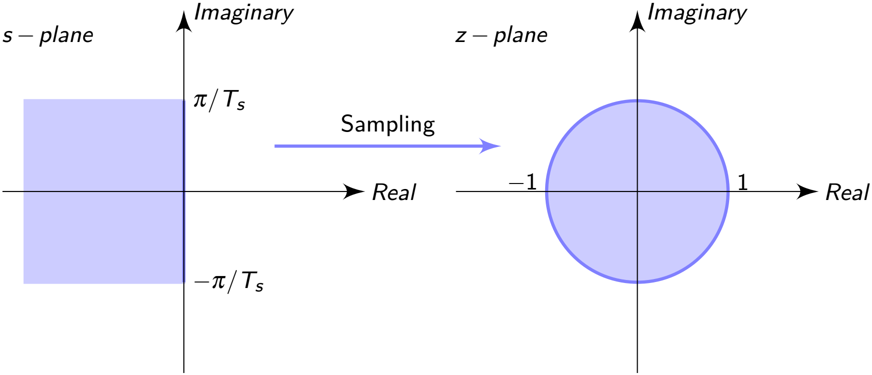
- at the cross-over frequency (which helps preserve phase margin)
- at the frequency of a critical notch for compensating system resonances

Sampling and aliasing

sampling maps the continuous-time frequency

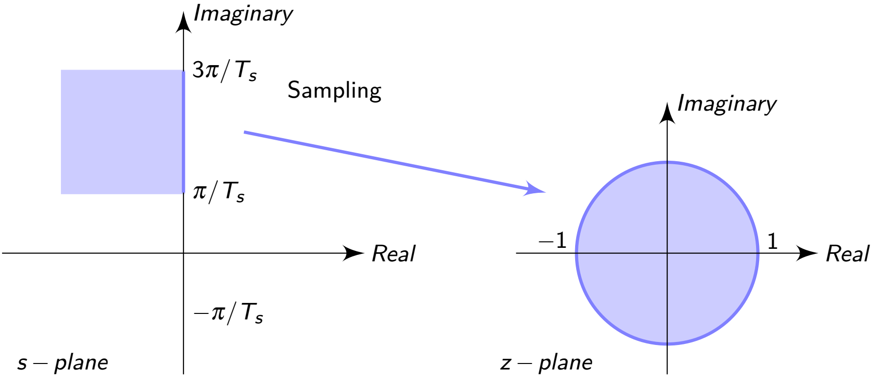
$$-\frac{\pi}{T_s} < \omega < \frac{\pi}{T_s}$$

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



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 $|\omega| > \pi/T_s$

Sampling example

- continuous-time signal

$$y(t) = \begin{cases} e^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}, \ a > 0$$

$$\mathscr{L}\{y(t)\} = \frac{1}{s+a}$$

- discrete-time sampled signal

$$y(k) = \begin{cases} e^{-aT_s k}, & k \geq 0 \\ 0, & k < 0 \end{cases}$$

$$\mathscr{Z}\{y(k)\} = \frac{1}{1 - z^{-1} e^{-aT_s}}$$

- sampling maps the continuous-time pole $s_i = -a$ to the discrete-time pole $z_i = e^{-aT_s}$, via the mapping

$$z_i = e^{s_i T_s}$$