Control Systems I

Discrete-Time Implementation

Colin Jones & Christophe Salzmann

Laboratoire d'Automatique

2021

Discrete-time concept

Implementation

All controllers developed in this course look something like

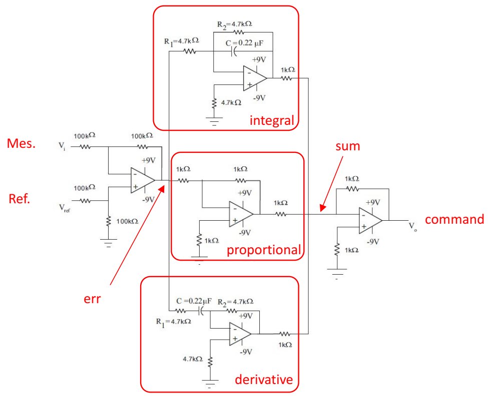
$$K(s) = \frac{U(s)}{E(s)} = \frac{b_0 + sb_1 + s^2 b_2}{1 + sa_1 + s^2 a_2 + s^3 a_3}$$

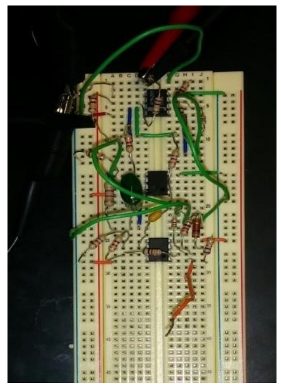
or equivalently in the time domain

$$u(t) = -a_1 \dot{u}(t) - a_2 \ddot{u}(t) - a_3 \dddot{u}(t) + b_0 e(t) + b_1 \dot{e}(t) + b_2 \ddot{e}(t)$$

Challenge: How can we implement this ?

Analog Implementation

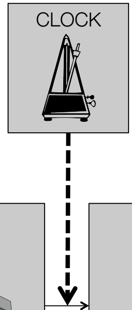


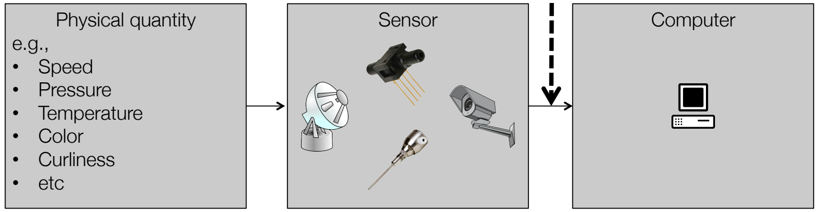


https://neelpmehta.wordpress.com/analog-pid-control-using-op-amps/

Sensors are "Discrete-Time"

- Sensors and computers can only "see" the world at fixed time intervals
- From the computer's perspective the world only exists at these fixed points in time!



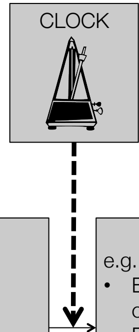


"Analogue": Changes continuously with time and can take any value

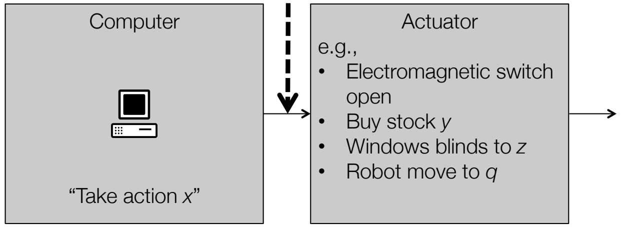
Convert physical quantity into a digital number

"Discrete"
Number representing physical quantity updated at fixed points in time

Control Actions are Updated in "Discrete-Time"



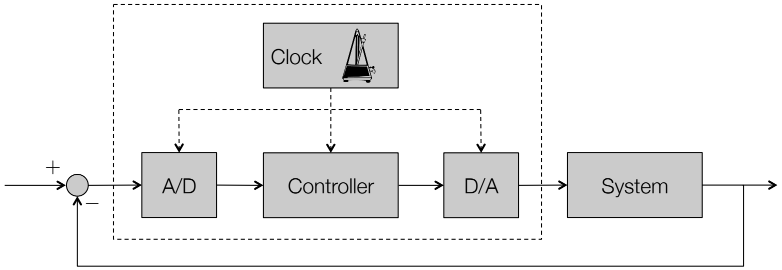
- Interface between computer and actuator runs on a fixed clock
- The action that the computer wants can only be changed at fixed points in time
- The action stays constant between updates



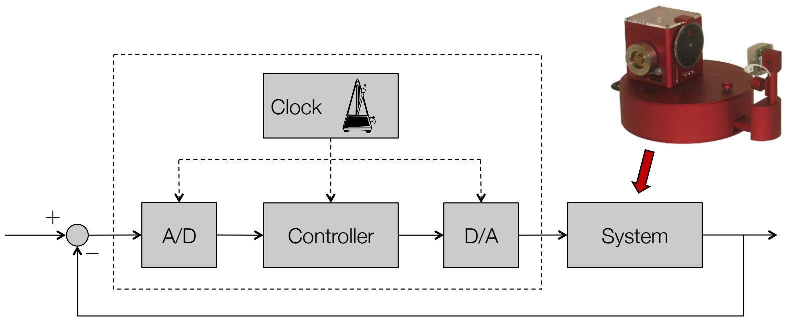
Compute can only take decisions at fixed points in time

Actuators turn 'numbers' into real actions; physical or logical

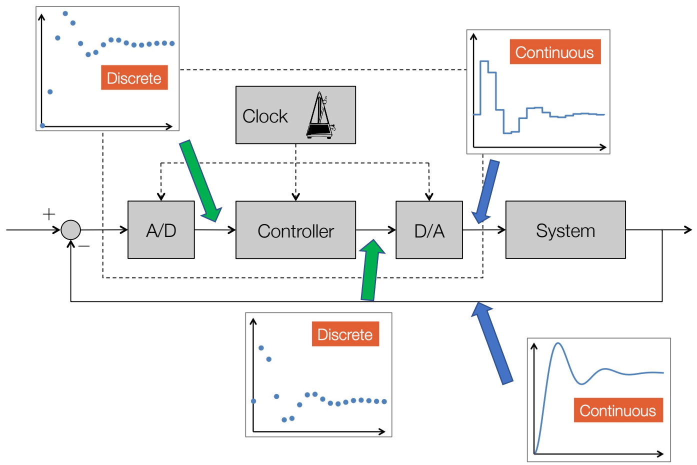
Control Loop with Digital Controller



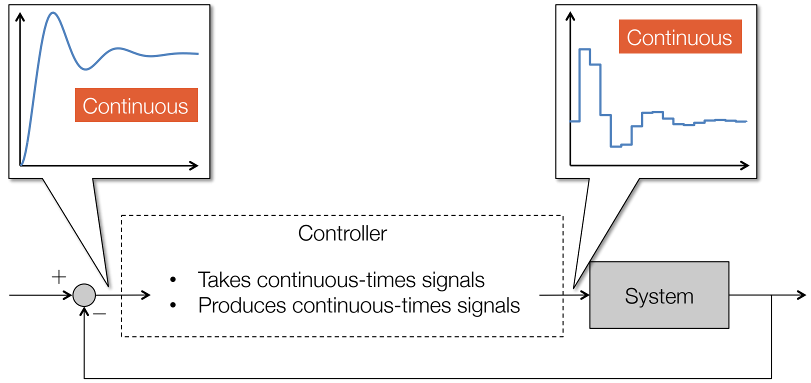
Control Loop with Digital Controller



Control Loop with Digital Controller

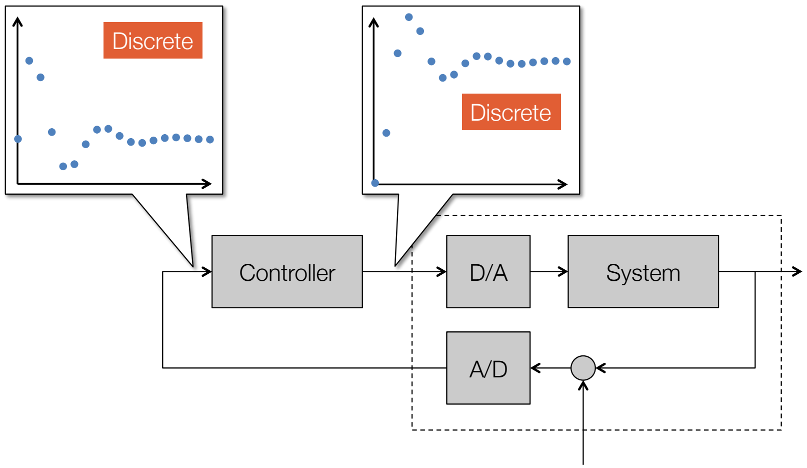


Perspective of the System



The system sees the **discrete-time controller** as a **continuous-time device**

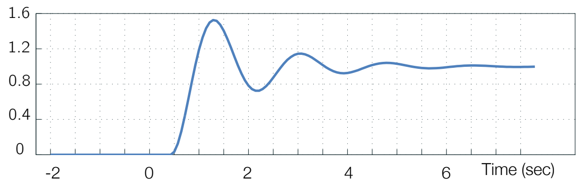
Perspective of the Controller



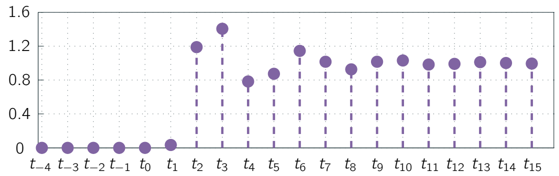
The controller sees the **continuous-time system** as a **discrete-time entity**

Sampling

Continuous-Time vs Discrete-Time Signals



Continuous-time signal: Function $w(t)$ mapping from $\mathbb{R}$ to $\mathbb{R}$.



Discrete-time signal:

- Function $w(t_k)$ mapping from $\{Tk \mid k \in \mathbb{Z}\}$ to $\mathbb{R}$
- Function $w(k)$ mapping from $\mathbb{Z}$ to $\mathbb{R}$

} Equivalent

Sampling - A Few Notes

- Normally sample with a constant **sampling period** T

$$t_k - t_{k-1} = T, \quad \forall k \in \mathbb{Z}$$

- **Sampling frequency**

$$f = \frac{1}{T} \text{ Hz} \qquad\qquad \omega = 2\pi f \text{ rad/sec}$$

- Discrete-time signals and systems are often expressed in terms of the **time index** k, rather than the physical time $t_k = Tk$.
  - We'll often write $w(k)$, $w(Tk)$ or $w(t_k)$ for the sampled signal
  - Controller doesn't care what 'time' it is - it operates on 'clock cycles'

Selection of a Sampling Rate

Nyquist Theorem

A sampled signal contains all information about the continuous signal it was sampled from up to half the sample frequency.

You will learn a lot more about this next year.

This tells us that we need to **sample at least twice as fast as the highest frequency that we care about.**

In practice: Sample $10\times$ to $40\times$ faster than the bandwidth of your system, depending on the cost of sensors, speed of the system, etc.

Demo sampling

Difference Equations

Difference Equations

A **linear difference equation** of order n:

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-d-1) + \cdots + b_m u(k-d-m)$$

or equivalently

$$y(k) = -\sum_{i=1}^{n} a_i y(k-i) + \sum_{i=0}^{m} b_i u(k-d-i)$$

Given an input signal u, the difference equation generates an output signal y.

- d is the system delay
- Represented by a finite number of constants $\{a_i\}$, $\{b_i\}$
- Can compute the value of y at time k given
  - last n outputs $\{y(k-1), \ldots, y(k-n)\}$
  - m inputs from d steps ago $\{u(k-d), u(k-d-1), \ldots, u(k-d-m)\}$

A computer can calculate a difference equation

Example: PI Controller

The PI controller is a dynamic system that takes the error e as an input and produces the system input u as its output

$$u(k) = K_P \left(e(k) + \frac{1}{T_i} \sum_{l=0}^{k-1} e(l)T \right)$$

Not a difference equation - requires a growing input history.

Can re-write:

$$\begin{aligned} u(k) - u(k-1) &= K_P \left(e(k) + \frac{1}{T_i} \sum_{l=0}^{k-1} e(l)T \right) \\ &\quad - K_P \left(e(k-1) + \frac{1}{T_i} \sum_{l=0}^{k-2} e(l)T \right) \\ &= K_P e(k) + K_P \left(\frac{T}{T_i} - 1 \right) e(k-1) \end{aligned}$$

An equivalent representation as a difference equation

Delay operator

Algebraic Representation of Difference Equations

Introduce the **shift operator** z

$$zy(k) = y(k+1) \qquad \text{Foward shift}$$
$$z^{-1}y(k) = y(k-1) \qquad \text{Backward shift}$$

Discretization

Algebraic Representation of Difference Equations

Introduce the **shift operator** z

$$zy(k) = y(k+1) \qquad \text{Foward shift}$$
$$z^{-1}y(k) = y(k-1) \qquad \text{Backward shift}$$

We can now re-write a difference equation as

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-d-1) + \cdots + b_m u(k-d-m)$$
$$(1 + a_1 z^{-1} + \cdots + a_n z^{-n}) y(k) = z^{-d}(b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}) u(k)$$

The next control course will introduce the $\mathcal{Z}$-transform formally, which allows us to define a **discrete time transfer function**

$$\frac{Y(z)}{U(z)} = H(z) = \underbrace{\frac{z^{-d}(b_0 + b_1 z^{-1} + \cdots + b_m z^{-m})}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}}_{\text{Discrete time transfer function}}$$

Approximate an ODE with a Difference Equation

Approximate Discretization

What we have

$$K(s) = \frac{U(s)}{E(s)}$$

$$u(t) + a_1 \dot{u}(t) + \cdots + a_n \frac{\mathrm{d}^p u}{\mathrm{d}t^p}(t) = b_0 e(t) + b_1 \dot{e}(t) + \cdots + b_p \frac{\mathrm{d}^p e}{\mathrm{d}t^p}(t)$$

what we want

$$\bar{u}(k) + a_1 \bar{u}(k-1) + \cdots + a_n \bar{u}(k-n) = b_0 \bar{e}(k-d) + b_1 \bar{e}(k-d-1) + \cdots + b_m \bar{e}(k-d-m)$$

Such that $\bar{u}(k) \approx u(t)$

Tustin Approximation

Write the transfer function in integral form

$$K(s) = \frac{U(s)}{E(s)} = \frac{b_0 s^{-(n-m)} + b_1 s^{-(n-m+1)} + \cdots + b_m s^{-n}}{1 + a_1 s^{-1} + a_2 s^{-2} + \cdots + a_n s^{-n}}$$

Re-writing gives

$$U(s) + a_1 \frac{1}{s} U(s) + a_2 \frac{1}{s^2} U(s) + \cdots + a_n \frac{1}{s^n} U(s)$$
$$= b_0 \frac{1}{s^{n-m}} E(s) + b_1 \frac{1}{s^{n-m+1}} E(s) + \cdots + b_m \frac{1}{s^n} E(s)$$

with the equivalent time-domain representation

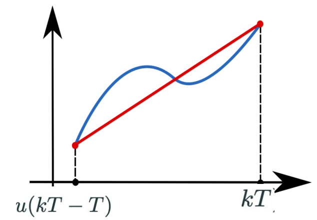
$$u(t) + a_1 \int_0^t u(\tau) d\tau + a_2 \int_0^t \int_0^\tau u(\sigma) d\sigma d\tau + \cdots = \cdots$$

Idea: Approximate the integral

Tustin Approximation

Take a trapezoidal approximation of the integral

$$\int_a^b f(x)\,dx \approx (b-a)\frac{f(a)+f(b)}{2}$$



$$i_1(kT) = \int_0^{kT} u(\tau) d\tau \approx i_1(kT-T) + \frac{T}{2}(u(kT-T) + u(kT))$$

Write in terms of the shift operator

$$I_1(z) = z^{-1} I_1(z) + \frac{T}{2}(z^{-1}+1)U(z) \qquad \rightarrow \qquad I_1(z) = \frac{T}{2}\frac{z+1}{z-1}U(z)$$

Tustin Approximation

We now have

$$I_1(s) = \frac{1}{s}U(s) \qquad \approx \qquad I_1'(z) = \frac{T}{2}\frac{z+1}{z-1}U'(z)$$

More generally, we can approximate the derivative operator s with $\frac{2}{T}\frac{z-1}{z+1}$

$$s \approx \frac{2}{T}\frac{z-1}{z+1}$$

Given our transfer function

$$K(s) = \frac{U(s)}{E(s)} = \frac{b_0 + b_1 s + \cdots + b_n s^n}{1 + a_1 s + \cdots + a_n s^n}$$

we can compute a discrete approximation:

$$K'(z) = \frac{U'(z)}{E'(z)} = \frac{b_0 + b_1 \left(\frac{2}{T}\frac{z-1}{z+1}\right) + \cdots + b_n \left(\frac{2}{T}\frac{z-1}{z+1}\right)^n}{1 + a_1 \left(\frac{2}{T}\frac{z-1}{z+1}\right) + \cdots + a_n \left(\frac{2}{T}\frac{z-1}{z+1}\right)^n}$$

Example - Lead Compensator

$$D(s) = \frac{U(s)}{E(s)} = \frac{T_D s + 1}{\alpha T_D s + 1}$$

Approximate discrete time transfer function

$$D'(z) = \frac{T_D\left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1}{\alpha T_D\left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1} =$$

Example - Lead Compensator

$$D(s) = \frac{U(s)}{E(s)} = \frac{T_D s + 1}{\alpha T_D s + 1}$$

Approximate discrete time transfer function

$$D'(z) = \frac{T_D \left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1}{\alpha T_D \left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1} = \frac{(T + 2T_D)\, z + T - 2T_D}{(T + 2T_D\alpha)\, z + T - 2T_D\alpha}$$

Write in terms of the **delay operator**

$$((T + 2T_D\alpha)z + T - 2T_D\alpha)u(k) = ((T + 2T_D)z + T - 2T_D)e(k)$$
$$((T + 2T_D\alpha) + (T - 2T_D\alpha)z^{-1})u(k) = ((T + 2T_D) + (T - 2T_D)z^{-1})e(k)$$

Example - Lead Compensator

$$D(s) = \frac{U(s)}{E(s)} = \frac{T_D s + 1}{\alpha T_D s + 1}$$

Approximate discrete time transfer function

$$D'(z) = \frac{T_D \left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1}{\alpha T_D \left(\frac{2}{T}\frac{z-1}{z+1}\right) + 1} = \frac{(T + 2T_D)\, z + T - 2T_D}{(T + 2T_D\alpha)\, z + T - 2T_D\alpha}$$

Write in terms of the **delay operator**

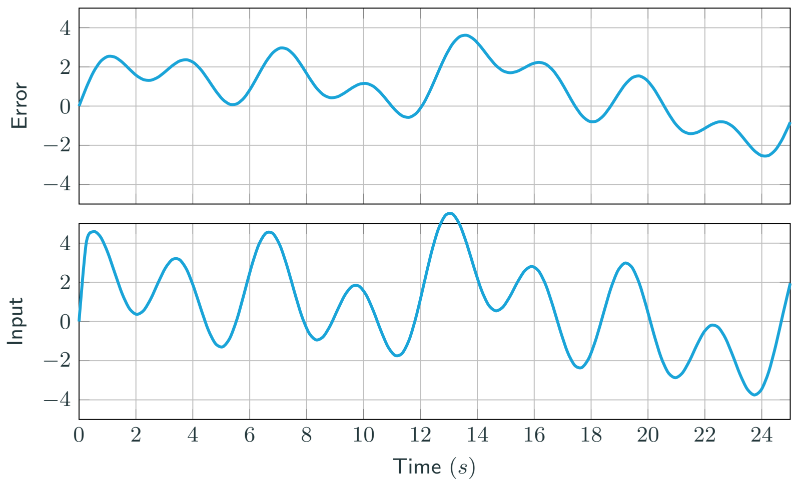
$$((T + 2T_D\alpha)z + T - 2T_D\alpha)u(k) = ((T + 2T_D)z + T - 2T_D)e(k)$$
$$((T + 2T_D\alpha) + (T - 2T_D\alpha)z^{-1})u(k) = ((T + 2T_D) + (T - 2T_D)z^{-1})e(k)$$

Convert to a difference equation

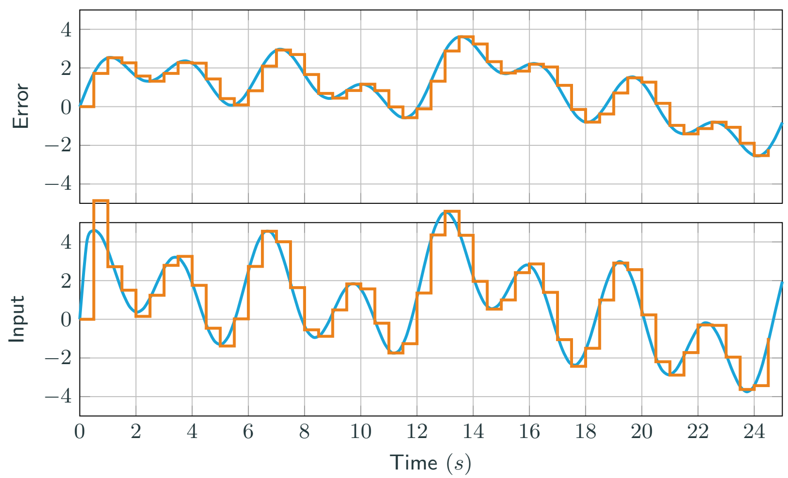
$$u(k) = -\frac{T - 2T_D\alpha}{T + 2T_D\alpha}u(k-1) + \frac{T + 2T_D}{T + 2T_D\alpha}e(k) + \frac{T - 2T_D}{T + 2T_D\alpha}e(k-1)$$

which gives us an expression that we can calculate in a computer

Example



Example



In matlab `c2d(D,T,'Tustin')`

Summary

Given a transfer function for a controller $K(s) = \frac{U(s)}{E(s)}$ and a sample period T, compute a difference equation that can be implemented in a computer.

- Compute an approximate discrete-time transfer function

$$K'(z) = K\left(\frac{2}{T}\frac{z-1}{z+1}\right) = \frac{b_0 z^m + b_1 z^{m-1} + \cdots + b_m}{z^n + a_1 z^{n-1} + \cdots + a_n}$$

- Write in terms of the delay operator z^{-1}

$$K'(z) = \frac{b_0 + b_1 z^{-1} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + \cdots + a_n z^{-n}}$$

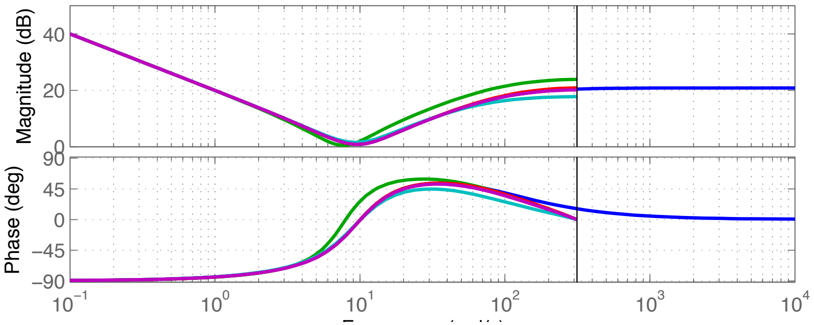
- Write the difference equation

$$u(k) = -a_1 u(k-1) - \cdots - a_n u(k-n) + b_0 e(k) + b_1 e(k-1) + \cdots + b_m e(k-m)$$

Notes

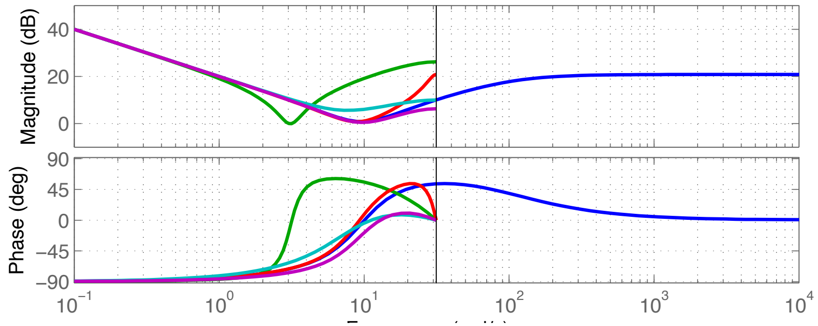
- There are a number of different approximations depending on the system
  - Tustin approximation $\rightarrow$ Matches well in the frequency domain
  - Zero/pole matching $\rightarrow$ Good for controllers based on pole placement
  - Euler approximation $\rightarrow$ Low complexity controller
- All the techniques match well if the sample rate is high enough
- Matlab command for continuous to discrete time conversion `c2d`

Impact of Sample Rate on Frequency Response



- Blue : Continuous time controller
- Green : ZOH approximation sampled at $T = 0.01$
- Red : Tustin approximation sampled at $T = 0.01$
- Cyan : Euler approximation sampled at $T = 0.01$
- Purple : Zero-pole matching sampled at $T = 0.01$

Impact of Sample Rate on Frequency Response



- Blue : Continuous time controller
- Green : ZOH approximation sampled at $T = 0.1$
- Red : Tustin approximation sampled at $T = 0.1$
- Cyan : Euler approximation sampled at $T = 0.1$
- Purple : Zero-pole matching sampled at $T = 0.1$