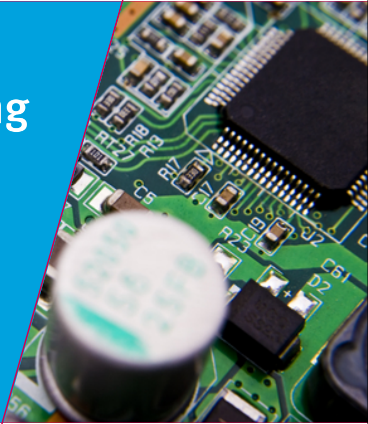
4CM00: Control Engineering

Digital filters

Dr.ir. Gert Witvoet



Technische Universiteit
Eindhoven
University of Technology

September 2020

Controller implementation

- Derive model / measurement of the plant
- Design a controller
- Implement controller in real-time environment...

Controller implementation

- Derive model / measurement of the plant
- Design a controller
- Implement controller in real-time environment...

$\Downarrow$

unexpected problems,

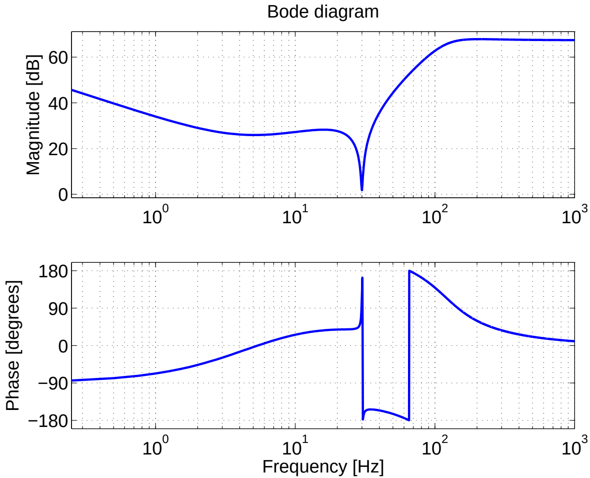
maybe even instability!

$\Downarrow$

WHY ??

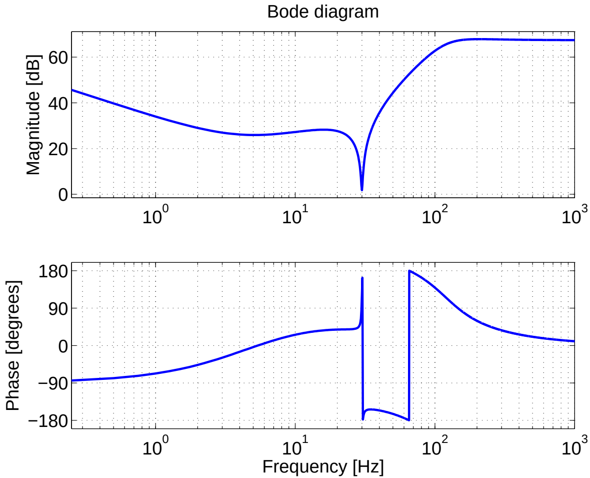
Controller implementation: example

Designed controller $C(s)$:

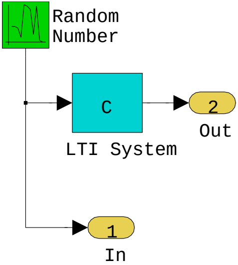


Controller implementation: example

Designed controller $C(s)$:

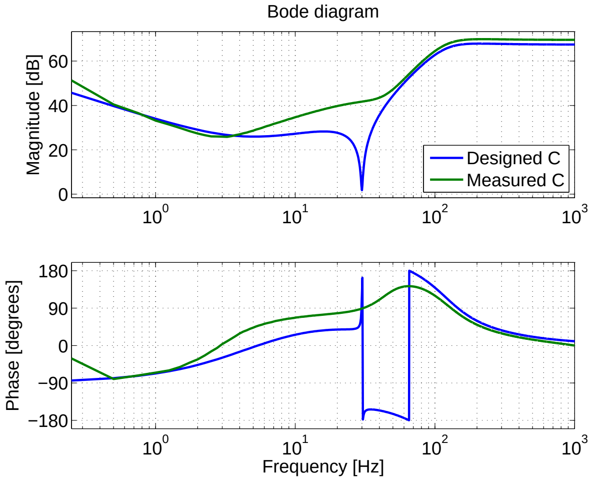


Measurement:

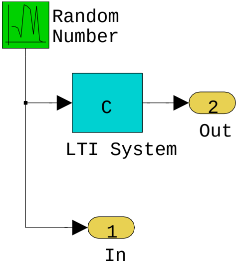


Controller implementation: example

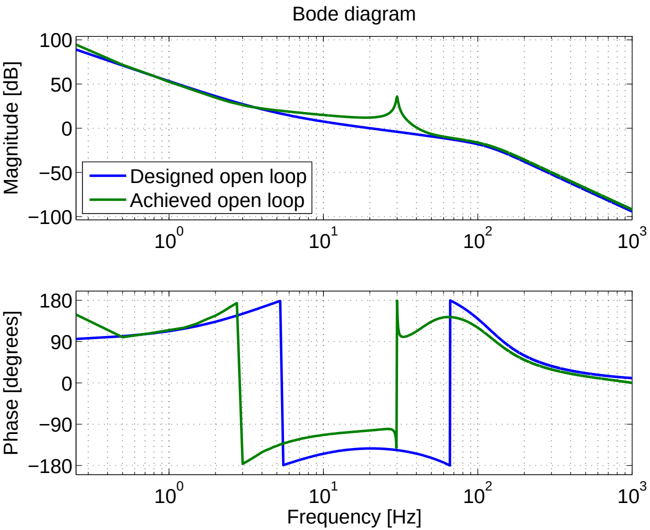
Designed controller $C(s)$:



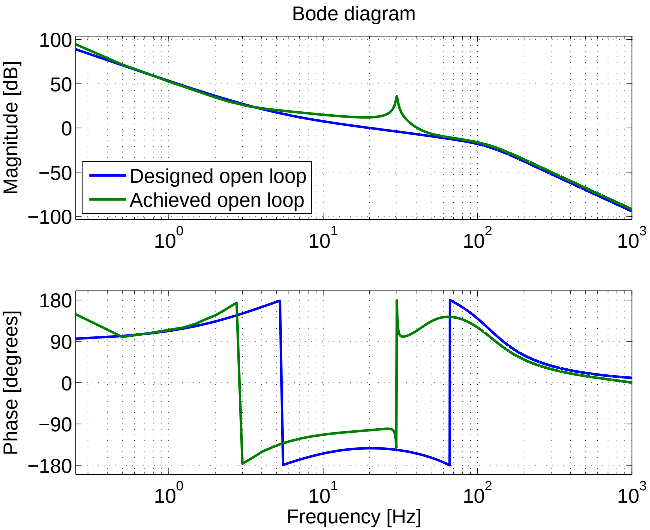
Measurement:



Controller implementation: example



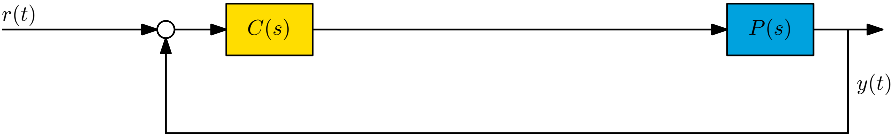
Controller implementation: example



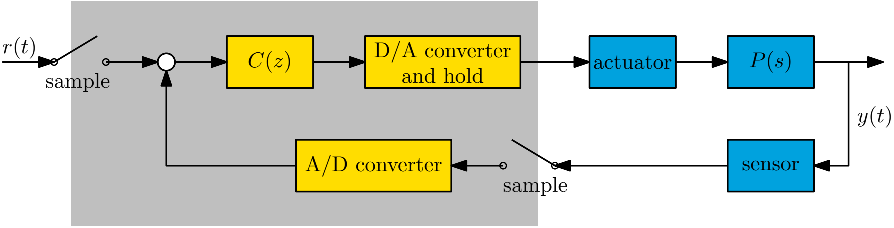
Achieved closed loop system is unstable!

Real-time implementation

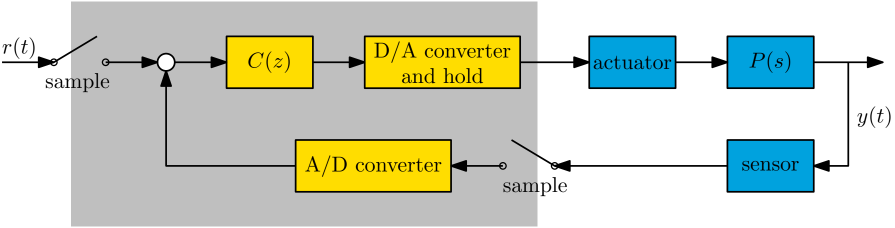
Real-time implementation



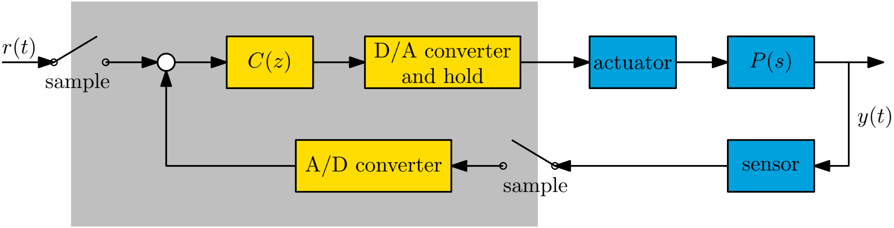
Real-time implementation



Real-time implementation



- Measurement $y(t)$ is sampled every T seconds: $y(t) \rightarrow y(kT)$
- Analog value of $y(kT)$ is digitized
  - e.g. by an encoder
  - e.g. by a 10, 12 or 16 bits A/D converter
- Controller $C(z)$ computes new output $u(kT)$ based on measurements $y(kT)$, $y((k-1)T)$, $y((k-2)T)$, etc.
- D/A converter creates analog continuous signal by holding the output $u(kT)$ until the next sample



Continuous time:

Controller $C(s)$ computes $u(t)$ based on continuous time signals $e(t), \dot{e}(t), \ddot{e}(t), \ldots, \int e(t), \int\int e(t), \ldots$

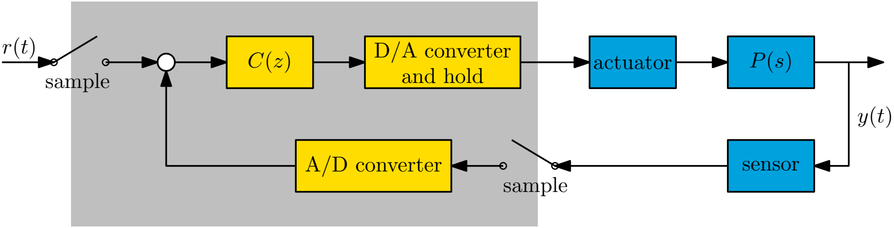
Discrete time:

Controller $C(z)$ computes $u(k)$ based on sampled signals $e(k), e(k-1), e(k-2), \ldots, u(k-1), u(k-2), \ldots$

Consequence: $C(s) \neq C(z)$!!!

Technische Universiteit Eindhoven University of Technology

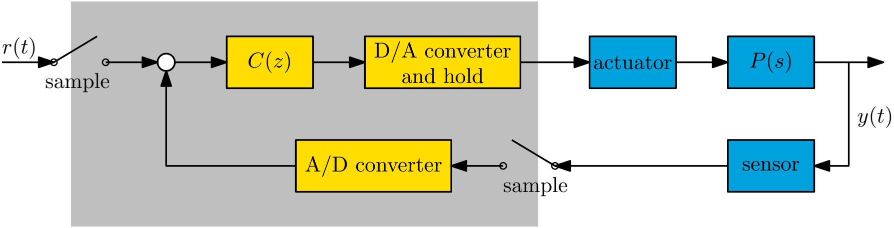
Real-time implementation



How to make correct controller $C(z)$?

- Discrete filter design
  - Design $C(z)$ from scratch in z-domain

Real-time implementation

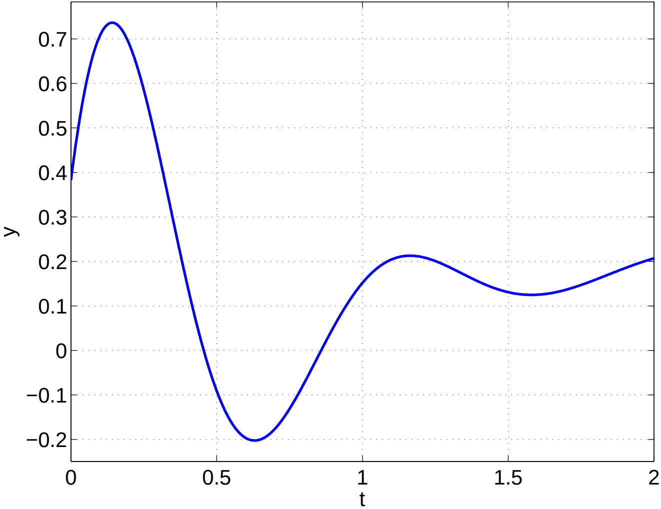


How to make correct controller $C(z)$?

- Discrete filter design
  - Design $C(z)$ from scratch in z-domain
- Emulation: discretize continuous time controller $C(s)$
  - Various discretization methods possible
  - Accuracy depends on method and sample time

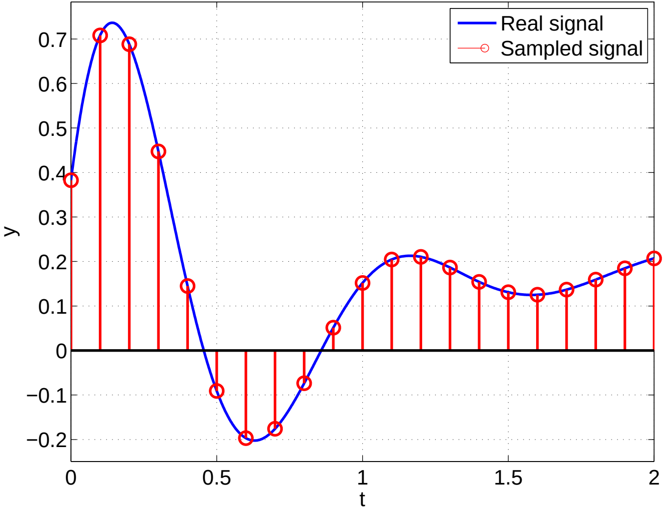
Effect of sampling

Measured plant output $y(t)$:



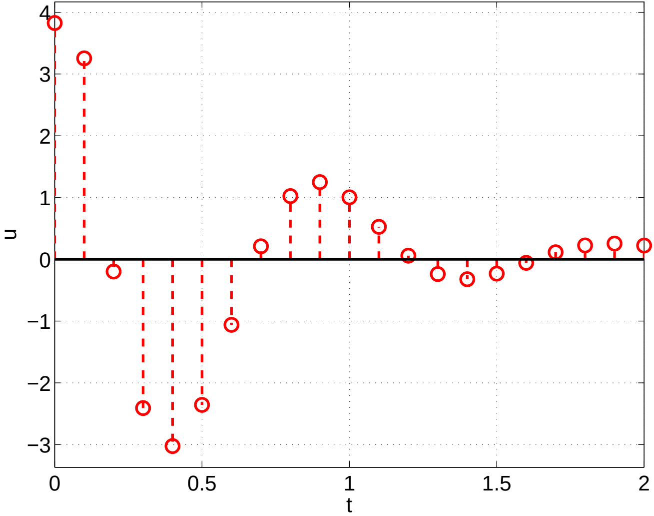
Effect of sampling

Measured plant output $y(t)$:



Effect of sampling

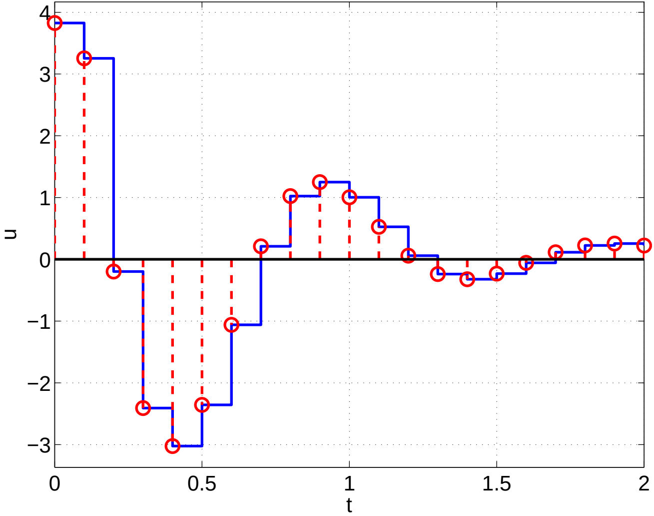
Calculated and applied controller output $u(t)$:



- Calculated signal

Effect of sampling

Calculated and applied controller output $u(t)$:

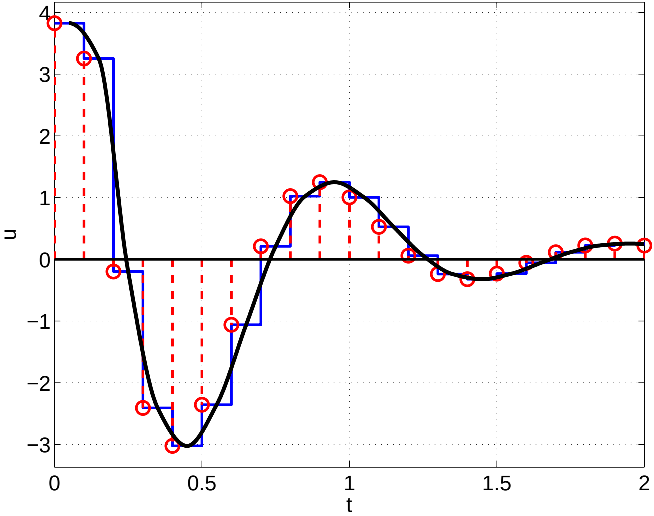


- Calculated signal
- Applied signal

Technische Universiteit Eindhoven University of Technology

Effect of sampling

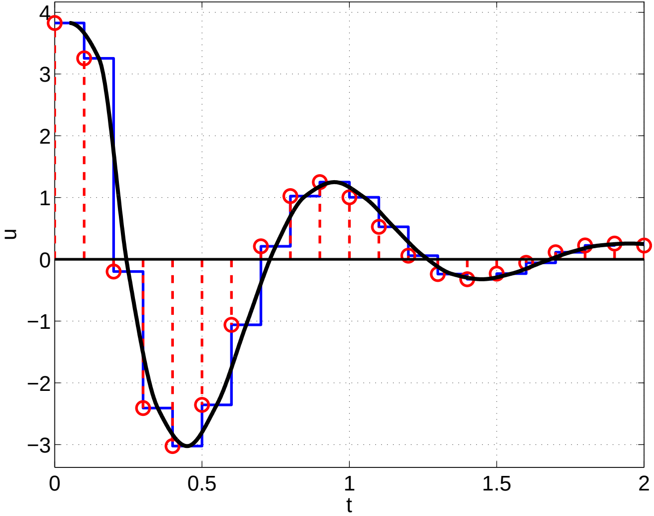
Calculated and applied controller output $u(t)$:



- Calculated signal
- Applied signal
- Equivalent signal

Effect of sampling

Calculated and applied controller output $u(t)$:



- Calculated signal
- Applied signal
- Equivalent signal

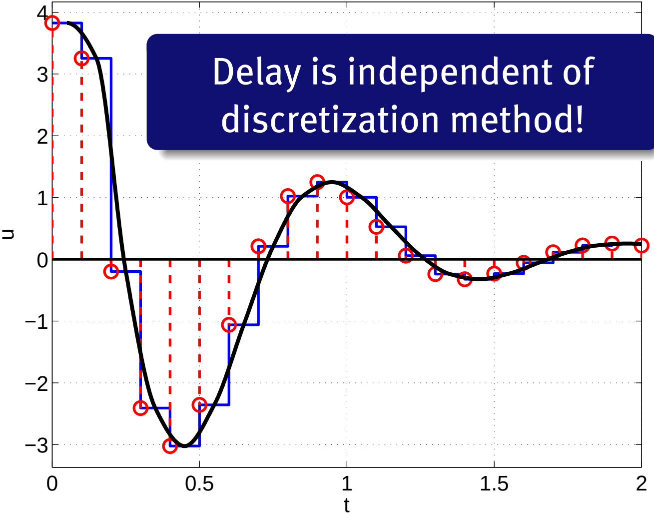
$\Downarrow$

delay: $\frac{1}{2}T$

Technische Universiteit Eindhoven University of Technology

Effect of sampling

Calculated and applied controller output $u(t)$:



Delay is independent of discretization method!

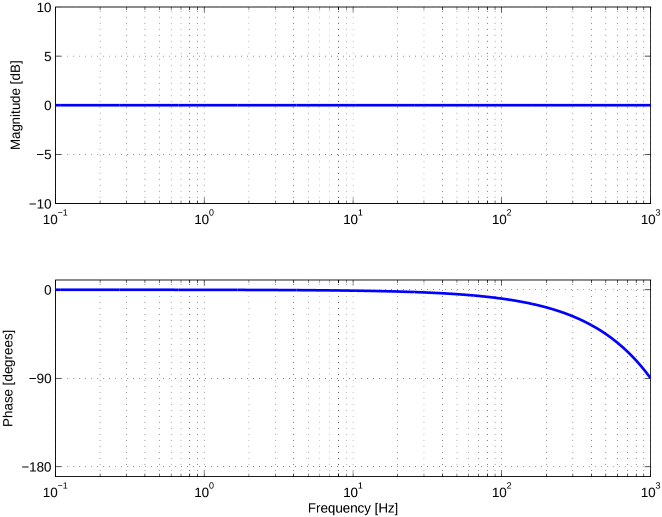
- Calculated signal
- Applied signal
- Equivalent signal

$\Downarrow$

delay: $\frac{1}{2}T$

Effect of sampling

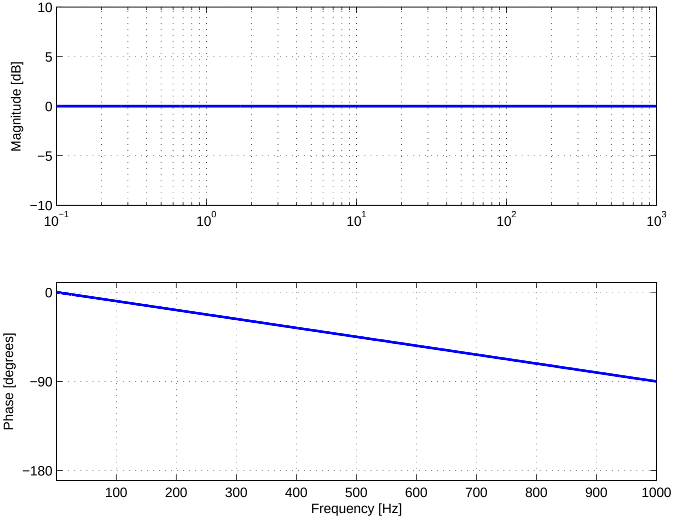
Phase delay is frequency dependent!



Technische Universiteit
Eindhoven
University of Technology

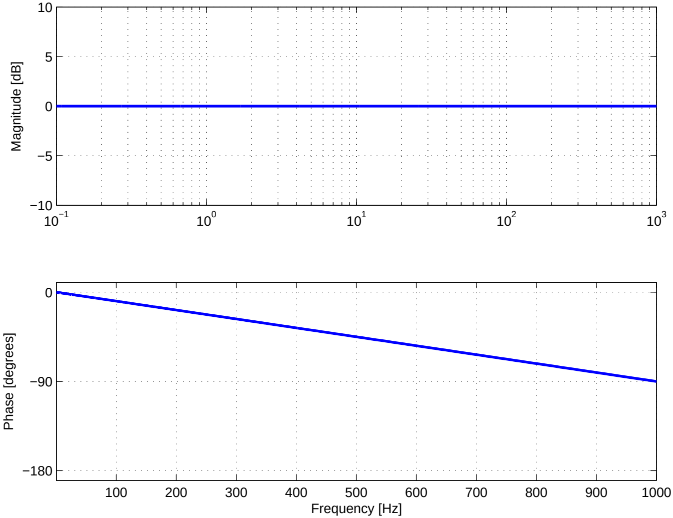
Effect of sampling

Phase delay is frequency dependent!



Effect of sampling

Phase delay is frequency dependent! $\Rightarrow$ $H_d(j\omega) = e^{-j\omega T_d}$



Technische Universiteit Eindhoven University of Technology

Introduction to discrete time systems

Discrete time systems

Discrete time systems produce sampled outputs based on sampled inputs.

Continuous:		Discrete:
time t	$\leftrightarrow$	time step k
signal $x(t)$	$\leftrightarrow$	sample $x(k)$
$\dot{x}(t) = f(x(t))$	$\leftrightarrow$	$x(k+1) = f(x(k), x(k-1), \ldots)$

Technische Universiteit Eindhoven University of Technology

Discrete time systems

Discrete time systems produce sampled outputs based on sampled inputs.

$$\begin{array}{rcl} \underline{\text{Continuous:}} & \leftrightarrow & \underline{\text{Discrete:}} \\ \text{time } t & \leftrightarrow & \text{time step } k \\ \text{signal } x(t) & \leftrightarrow & \text{sample } x(k) \\ \dot{x}(t) = f(x(t)) & \leftrightarrow & x(k+1) = f(x(k), x(k-1), \ldots) \end{array}$$

Past, present and future values of $x(k)$ are denoted by the shift operator z:

$$\begin{aligned} x(k+1) &= zx(k) \\ x(k-1) &= z^{-1}x(k) \\ x(k-N) &= z^{-N}x(k) \end{aligned}$$

The z transform

Shift operator z is discrete counterpart of Laplace variable s:

$$\dot{x}(s) = sx(s) \quad \leftrightarrow \quad x(k+1) = zx(k)$$

The z transform can be used to derive I/O models.

The z transform

Shift operator z is discrete counterpart of Laplace variable s:

$$\dot{x}(s) = sx(s) \quad \leftrightarrow \quad x(k+1) = zx(k)$$

The z transform can be used to derive I/O models.

Example:

$$\begin{aligned}
y(k) &= -a_1 y(k-1) + b_0 u(k) + b_1 u(k-1) \\
y(k) &= -a_1 z^{-1} y(k) + b_0 u(k) + b_1 z^{-1} u(k) \\
\left(1 + a_1 z^{-1}\right) y(k) &= \left(b_0 + b_1 z^{-1}\right) u(k) \\
H(z) = \frac{y(k)}{u(k)} &= \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}
\end{aligned}$$

or equivalently: $H(z) = \dfrac{b_0 z + b_1}{z + a_1}$.

Technische Universiteit Eindhoven University of Technology

Backwards derivation:

- derive the difference equation of the transfer function

$$H(z) = \frac{z}{z-1}$$

The z transform

Backwards derivation:

- derive the difference equation of the transfer function

$$H(z) = \frac{z}{z-1}$$

Answer:

$$\begin{aligned} \frac{y(k)}{u(k)} &= \frac{z}{z-1} \\ (z-1)y(k) &= zu(k) \\ y(k+1) - y(k) &= u(k+1) \\ y(k+1) &= u(k+1) + y(k) \end{aligned}$$

or equivalently: $y(k) = u(k) + y(k-1)$.

Stability

	Continuous time	Discrete time
Transfer function	$H(s) = \frac{\mathrm{num}(s)}{\mathrm{den}(s)}$	$H(z) = \frac{\mathrm{num}(z)}{\mathrm{den}(z)}$
Poles λ_i	$\mathrm{den}(s = \lambda_i) = 0$	$\mathrm{den}(z = \lambda_i) = 0$
Stable if	$\mathrm{Re}(\lambda_i) < 0 \quad \forall \lambda_i$	$\lvert\lambda_i\rvert < 1 \quad \forall \lambda_i$
Stability boundary	imaginary axis:	unit disc:
	$s = j\omega$	$z = \cos\omega + j\sin\omega = e^{j\omega}$

Stability

	Continuous time	Discrete time
Transfer function	$H(s) = \frac{\mathrm{num}(s)}{\mathrm{den}(s)}$	$H(z) = \frac{\mathrm{num}(z)}{\mathrm{den}(z)}$
Poles λ_i	$\mathrm{den}(s = \lambda_i) = 0$	$\mathrm{den}(z = \lambda_i) = 0$
Stable if	$\mathrm{Re}(\lambda_i) < 0 \;\; \forall \lambda_i$	$\lvert\lambda_i\rvert < 1 \;\; \forall \lambda_i$
Stability boundary	imaginary axis:	unit disc:
	$s = j\omega$	$z = \cos\omega + j\sin\omega = e^{j\omega}$

Discrete time frequency ω:

Frequency ω is normalized, such that $\omega \in [-\pi, \pi]$.
This corresponds to a sampling time $T = 1$.
Actual frequency $\omega_s = \frac{\omega}{T_s}$, where T_s is actual sample time.

Stability

Example 1:

$$y(k+1) = ay(k)$$

Consider a time sequence:

$$\begin{aligned} y(k+2) &= ay(k+1) = a^2 y(k) \\ y(k+3) &= ay(k+2) = a^3 y(k) \\ y(k+N) &= a^N y(k) \end{aligned}$$

Note: $y(k+N) \to 0$ for $N \to \infty$ if and only if $a^N \to 0$, i.e. $|a| < 1$.

Stability

Example 2:

$$H(z) = \frac{z^3 + 0.9z^2 + 0.16z - 0.9}{z^3 - 1.3z^2 + 0.2z + 0.2}$$

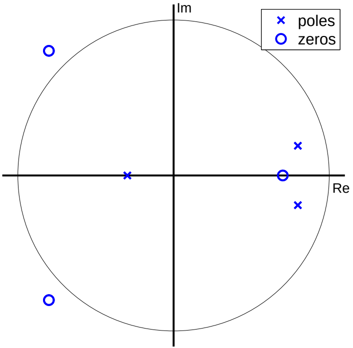
Poles: $\lambda^3 - 1.3\lambda^2 + 0.2\lambda + 0.2 = 0 \quad \Rightarrow \quad \lambda = \begin{bmatrix} 0.80 + 0.19j \\ 0.80 - 0.19j \\ -0.30 \end{bmatrix}$

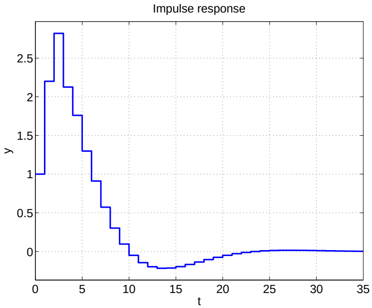
Stability

Example 2:

$$H(z) = \frac{z^3 + 0.9z^2 + 0.16z - 0.9}{z^3 - 1.3z^2 + 0.2z + 0.2}$$

Poles: $\lambda^3 - 1.3\lambda^2 + 0.2\lambda + 0.2 = 0 \quad \Rightarrow \quad \lambda = \begin{bmatrix} 0.80 + 0.19j \\ 0.80 - 0.19j \\ -0.30 \end{bmatrix}$





Discrete frequency response

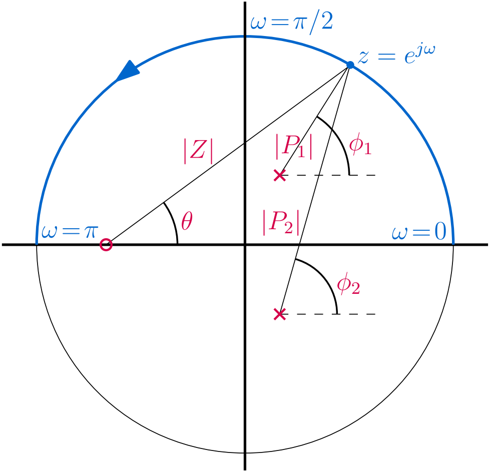
Continuous: Substitute $s = j\omega$ into $H(s)$ to obtain $H(j\omega)$

Discrete: Substitute $z = e^{j\omega}$ into $H(z)$ to obtain $H(j\omega)$

Discrete frequency response

Continuous: Substitute $s = j\omega$ into $H(s)$ to obtain $H(j\omega)$

Discrete: Substitute $z = e^{j\omega}$ into $H(z)$ to obtain $H(j\omega)$



Example:

$$H(z) = \frac{z+b}{(z+a_1)(z+a_2)}$$

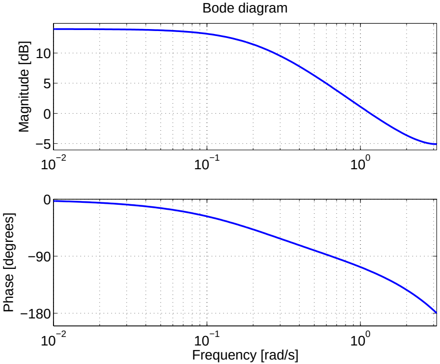
Magnitude and phase:

$$\begin{aligned} |H(j\omega)| &= \frac{|e^{j\omega}+b|}{|e^{j\omega}+a_1||e^{j\omega}+a_2|} \\ &= \frac{|Z|}{|P_1|\cdot|P_2|} \\ \angle H(j\omega) &= \theta - \phi_1 - \phi_2 \end{aligned}$$

Discrete frequency response

- Stable poles and zeros can have 180° phase change

$$H(z) = \frac{1}{z - 0.8}$$

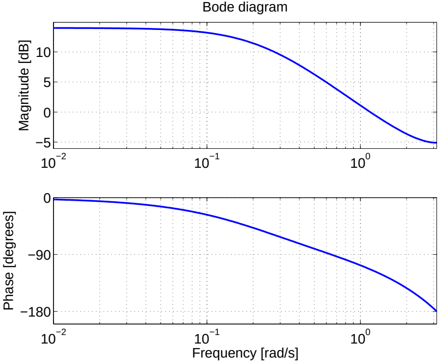


Information up to Nyquist frequency

Discrete frequency response

- Stable poles and zeros can have 180° phase change

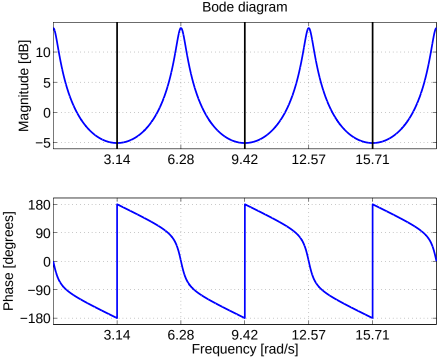
$$H(z) = \frac{1}{z-0.8}$$



Information up to Nyquist frequency

- Frequency response repeats itself every 2π rad/s

$$H(z) = \frac{1}{z-0.8}$$



Plotted on linear frequency scale

Discrete frequency response

Is there a relation between continuous and discrete poles / zeros?

Discrete frequency response

Is there a relation between continuous and discrete poles / zeros?

Answer: Yes, $\lambda_D = e^{T_s \lambda_C}$

- λ_D: discrete time pole / zero
- λ_C: continuous time pole / zero
- T_s: sample time

Discrete frequency response

Is there a relation between continuous and discrete poles / zeros?

Answer: Yes, $\lambda_D = e^{T_s \lambda_C}$

- λ_D: discrete time pole / zero
- λ_C: continuous time pole / zero
- T_s: sample time

- Stable poles / zeros are mapped inside the unit disc
  $\mathrm{Re}(\lambda_C) < 0 \quad \Rightarrow \quad |\lambda_D| < 1$
- Unstable poles / zeros are mapped outside the unit disc
  $\mathrm{Re}(\lambda_C) > 0 \quad \Rightarrow \quad |\lambda_D| > 1$

Discrete frequency response

Is there a relation between continuous and discrete poles / zeros?

Answer: Yes, $\lambda_D = e^{T_s \lambda_C}$

- λ_D: discrete time pole / zero
- λ_C: continuous time pole / zero
- T_s: sample time

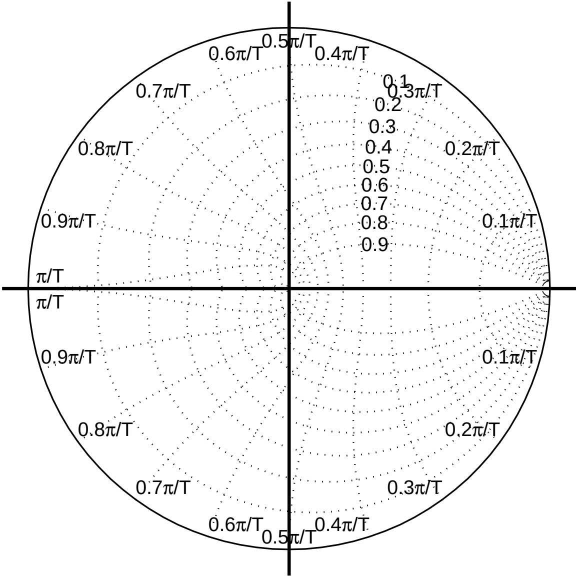
- Stable poles / zeros are mapped inside the unit disc
  $\text{Re}(\lambda_C) < 0 \quad \Rightarrow \quad |\lambda_D| < 1$
- Unstable poles / zeros are mapped outside the unit disc
  $\text{Re}(\lambda_C) > 0 \quad \Rightarrow \quad |\lambda_D| > 1$

To find frequency of discrete pole / zero, use inverse relation:

$$f = \left| \frac{\ln(\lambda_D)}{2\pi T_s} \right|$$

E.g. when $f_s = 1000$Hz we have that $\lambda_D = 0.8$ is located at $f = 35.5$Hz.

Discrete frequency response



- Stability boundary: $s = j\omega \;\Rightarrow\; |z| = 1$
- 'Origin': $s = 0 \;\Rightarrow\; z = +1$
- 'Infinite' pole: $s = -\infty \;\Rightarrow\; z = 0$
- Vertical lines in s-plane $\Rightarrow$ circles in z-plane
- Horizontal lines in s-plane $\Rightarrow$ radial lines in z-plane

Impulse response

It is easy to derive the impulse response from the z-transform:

- write $H(z)$ in the form $H(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3} + \dots$
- i.e. eliminate denominator term

Impulse response

It is easy to derive the impulse response from the z-transform:

- write $H(z)$ in the form $H(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3} + \dots$
- i.e. eliminate denominator term

Difference equation then becomes:

$$y(k) = c_0 u(k) + c_1 u(k-1) + c_2 u(k-2) + c_3 u(k-3) + \dots$$

Impulse input: $u(k) = 1$ for $k = 0$, and $u(k) = 0$ for $k \neq 0$.

- At time step $k = i$ only $u(k-i)$ contributes to output
- So coefficient c_i determines output: $y(i) = c_i$

Impulse response

It is easy to derive the impulse response from the z-transform:

- write $H(z)$ in the form $H(z) = c_0 + c_1 z^{-1} + c_2 z^{-2} + c_3 z^{-3} + \dots$
- i.e. eliminate denominator term

Difference equation then becomes:

$$y(k) = c_0 u(k) + c_1 u(k-1) + c_2 u(k-2) + c_3 u(k-3) + \dots$$

Impulse input: $u(k) = 1$ for $k = 0$, and $u(k) = 0$ for $k \neq 0$.

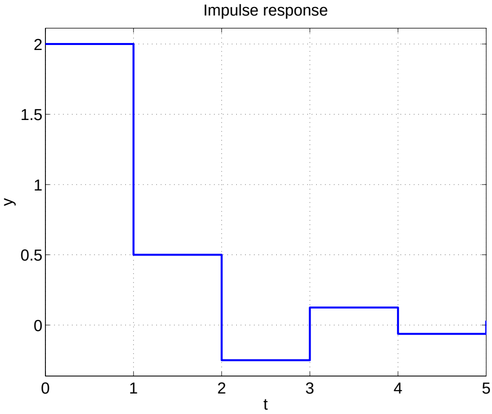
- At time step $k = i$ only $u(k-i)$ contributes to output
- So coefficient c_i determines output: $y(i) = c_i$

Note: In general, expansion of $H(z)$ is *infinite*. However, in some cases $H(z)$ has a Finite Impulse Response. $H(z)$ is then a FIR filter.

Technische Universiteit Eindhoven University of Technology

Impulse response: example

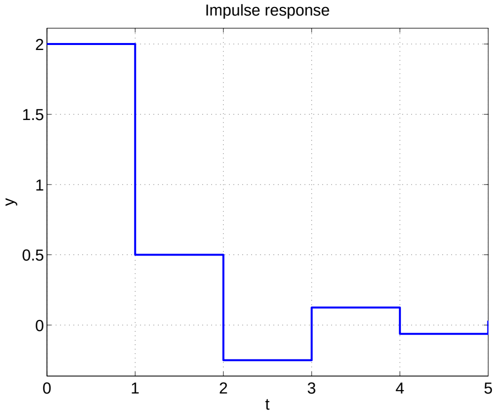
$$H(z) = \frac{4 + 3z^{-1}}{2 + z^{-1}}$$



Impulse response: example

$$H(z) = \frac{4+3z^{-1}}{2+z^{-1}} = 2 + \frac{1}{2}z^{-1} - \frac{1}{4}z^{-2} + \frac{1}{8}z^{-3} - \dots$$

$$y(k) = 2u(k) + \frac{1}{2}u(k-1) - \frac{1}{4}u(k-2) + \frac{1}{8}u(k-3) - \dots$$



Anti-causal systems

Ordinary causal systems:

- output only depends on present and past input values:
  $y(k) = f(u(k), u(k-1), u(k-2), \ldots)$

Anti-causal systems:

- output also depends on future input values
- 'predicts' future
- do not occur in reality

Anti-causal systems

Ordinary causal systems:

- output only depends on present and past input values:
  $y(k) = f(u(k), u(k-1), u(k-2), \ldots)$

Anti-causal systems:

- output also depends on future input values
- 'predicts' future
- do not occur in reality

How to recognize recognize anti-causal systems?

- Impulse response of $H(z)$:
  contains positive powers of shift operator, i.e. z, z^2, etc.
- Transfer function $H(z)$:
  numerator has higher degree than denominator,
  i.e. $H(z)$ has more zeros than poles.

Controller design

Controller emulation

Implementation procedure

- Design continuous time controller $C(s)$
- Discretize the controller
  - Time domain approximation
    - Forward Euler (Zero-order hold)
    - Backward Euler
    - Tustin
  - Frequency domain approximation
    - Tustin with prewarping
    - Pole-zero matching
- Measure obtained controller $C(z)$ in open loop
  - Compare with design!
  - Adjust controller if necessary
- Implement $C(z)$ in closed loop

Technische Universiteit Eindhoven University of Technology

Time domain approximations

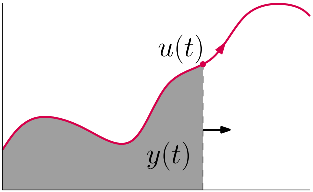
- Imitate time domain effect of controller $C(s)$
- Make assumptions on behavior of $u(t)$ between samples:
  - hold: take $u(t)$ constant between samples (Euler)
  - linear: assume linear change from $u(k-1)$ to $u(k)$ (Tustin)

Time domain approximations

- Imitate time domain effect of controller $C(s)$
- Make assumptions on behavior of $u(t)$ between samples:
  - <u>hold:</u> take $u(t)$ constant between samples (Euler)
  - <u>linear:</u> assume linear change from $u(k-1)$ to $u(k)$ (Tustin)

Consider continuous time integrator: $C(s) = \frac{1}{s}$

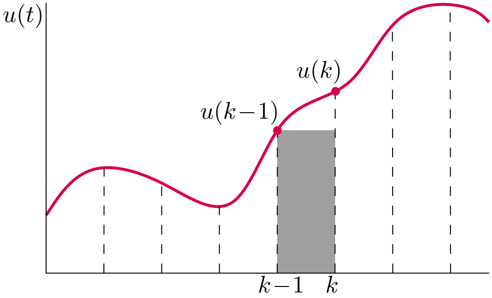
$$y = \int u\, dt \qquad \text{or} \qquad \dot{y} = u$$



- $y(t)$ is surface beneath curve $u(t)$
- $y(k)$ is previous value $y(k-1)$ + additional surface
- Sample time: T

Forward Euler

Assume $u(t)$ constant from $k-1$ to k.



Forward Euler:

$$y(k) = y(k-1) + Tu(k-1)$$

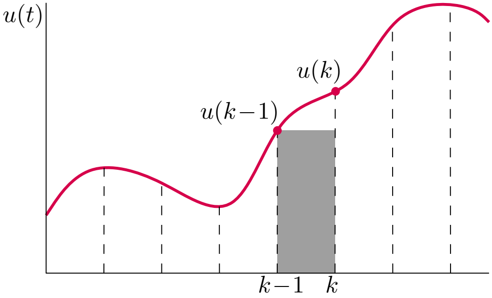
Approximation of $C(s)$:

$$C(z) = \frac{Tz^{-1}}{1 - z^{-1}} = \frac{T}{z-1}$$

Forward Euler: $\frac{1}{s} \to \frac{T}{z-1}$ and $s \to \frac{z-1}{T}$

Forward Euler

Assume $u(t)$ constant from $k-1$ to k.



Forward Euler:

$$y(k) = y(k-1) + Tu(k-1)$$

Approximation of $C(s)$:

$$C(z) = \frac{Tz^{-1}}{1 - z^{-1}} = \frac{T}{z-1}$$

Forward Euler: $\frac{1}{s} \rightarrow \frac{T}{z-1}$ and $s \rightarrow \frac{z-1}{T}$

Example:

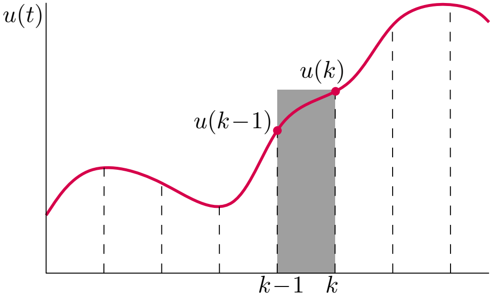
$$C(s) = \frac{s+a}{s+b} \quad \Rightarrow \quad C(z) = \frac{z-1+aT}{z-1+bT}$$

Drawback: $C(z)$ is anti-causal for non-proper $C(s)$ (derivative-terms)!

Technische Universiteit Eindhoven University of Technology

Backward Euler

Assume $u(t)$ was constant from k to $k-1$.



Backward Euler:

$$y(k) = y(k-1) + Tu(k)$$

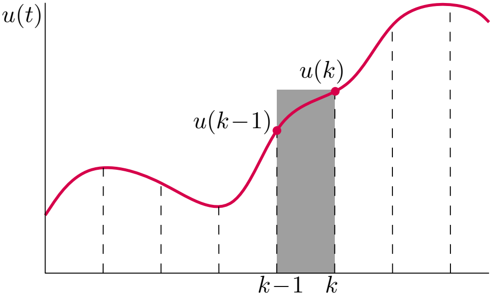
Approximation of $C(s)$:

$$C(z) = \frac{T}{1-z^{-1}} = \frac{Tz}{z-1}$$

Backward Euler: $\frac{1}{s} \rightarrow \frac{Tz}{z-1}$ and $s \rightarrow \frac{z-1}{Tz}$

Backward Euler

Assume $u(t)$ was constant from k to $k-1$.



Backward Euler:

$$y(k) = y(k-1) + Tu(k)$$

Approximation of $C(s)$:

$$C(z) = \frac{T}{1 - z^{-1}} = \frac{Tz}{z-1}$$

Backward Euler: $\frac{1}{s} \to \frac{Tz}{z-1}$ and $s \to \frac{z-1}{Tz}$

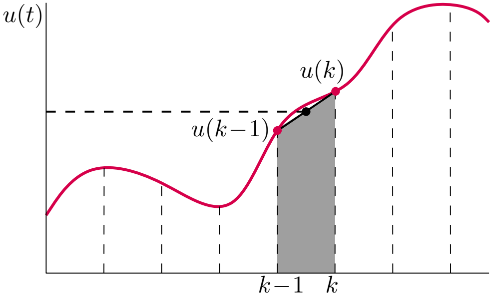
Example:

$$C(s) = \frac{s+a}{s+b} \quad \Rightarrow \quad C(z) = \frac{(1+aT)z-1}{(1+bT)z-1}$$

Technische Universiteit Eindhoven University of Technology

Tustin

Assume linear change from $u(k-1)$ to $u(k)$



Tustin:

$$y(k) = y(k-1) + T\frac{u(k)+u(k-1)}{2}$$

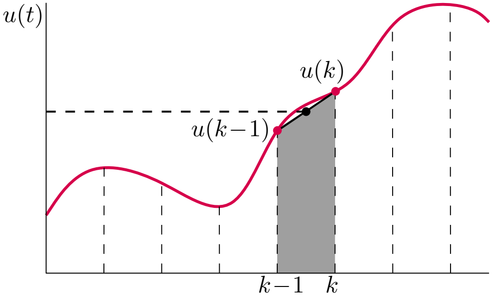
Approximation of $C(s)$:

$$C(z) = \frac{T(1+z^{-1})}{2(1-z^{-1})} = \frac{T(z+1)}{2(z-1)}$$

Tustin: $\frac{1}{s} \to \frac{T(z+1)}{2(z-1)}$ and $s \to \frac{2(z-1)}{T(z+1)}$

Tustin

Assume linear change from $u(k-1)$ to $u(k)$



Tustin:

$$y(k) = y(k-1) + T\frac{u(k)+u(k-1)}{2}$$

Approximation of $C(s)$:

$$C(z) = \frac{T(1+z^{-1})}{2(1-z^{-1})} = \frac{T(z+1)}{2(z-1)}$$

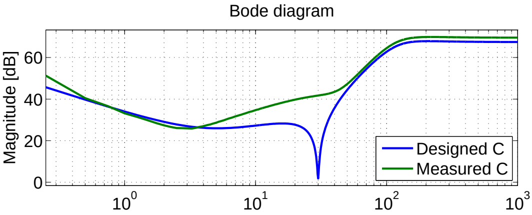
Tustin: $\frac{1}{s} \rightarrow \frac{T(z+1)}{2(z-1)}$ and $s \rightarrow \frac{2(z-1)}{T(z+1)}$

Example:

$$C(s) = \frac{s+a}{s+b} \quad \Rightarrow \quad C(z) = \frac{2(z-1)+aT(z+1)}{2(z-1)+bT(z+1)}$$

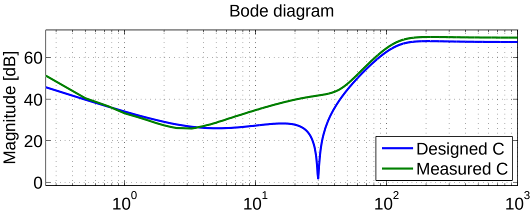
Frequency domain approximation

Note: Time domain approximations (especially Euler) do not take frequency response information into account!
Especially 'fast' dynamics in $C(s)$ can be 'missed' by the discretization, e.g. (anti-)resonances.



Frequency domain approximation

Note: Time domain approximations (especially Euler) do not take frequency response information into account!
Especially ‘fast’ dynamics in $C(s)$ can be ‘missed’ by the discretization, e.g. (anti-)resonances.



Solution:

- Include frequency information into discretization:
  - Tustin with prewarping
  - Pole-zero matching

Tustin with prewarping

Extension of the Tustin discretization:

- Prewarping matches frequency responses of $C(z)$ and $C(s)$ at a specified frequency ω_p
- Combination of time and frequency domain techniques

Tustin substitution changes into:

$$s \to \frac{\omega_p}{\tan(\omega_p T/2)} \cdot \frac{z-1}{z+1}$$

As a consequence:

$$C\left(s = j\omega_p\right) \ = \ C\left(z = e^{j\omega_p T}\right)$$

Pole-zero matching

- Completely frequency based
- Maps all poles and zeros to the discrete domain

Procedure:

- Determine poles p_i and zeros q_i of $C(s)$
- Map p_i and q_i to discrete domain:
  - discrete poles at $z = e^{p_i T}$
  - discrete zeros at $z = e^{q_i T}$
- Match low-frequent DC-gain of $C(z)$ to $C(s)$

Pole-zero matching

- Completely frequency based
- Maps all poles and zeros to the discrete domain

Procedure:

- Determine poles p_i and zeros q_i of $C(s)$
- Map p_i and q_i to discrete domain:
  - discrete poles at $z = e^{p_i T}$
  - discrete zeros at $z = e^{q_i T}$
- Match low-frequent DC-gain of $C(z)$ to $C(s)$

Example:

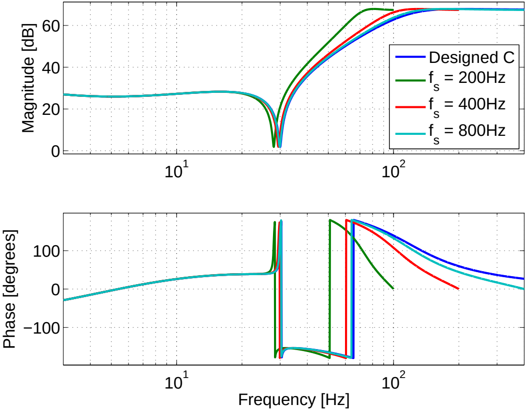
$$C(s) = \frac{s+a}{s+b} \quad \Rightarrow \quad C(z) = \frac{a}{b} \cdot \frac{1-e^{-bT}}{1-e^{-aT}} \cdot \frac{z-e^{-aT}}{z-e^{-bT}}$$

Technische Universiteit Eindhoven University of Technology

Influence of sample time

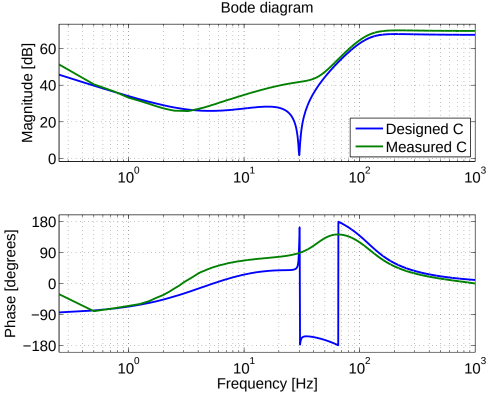
For all discretization methods:

- The smaller the sample time T, the better the approximation
- In practice, choose sample frequency at least 20 times bandwidth



Simulink

Back to our original problem:



Simulink chooses its own discretization method for $C(s)$:

- uses zero-order hold (ZOH)
- same as forward Euler for small sample times T

DCtools

ShapeIt comes with the DCtools blockset.

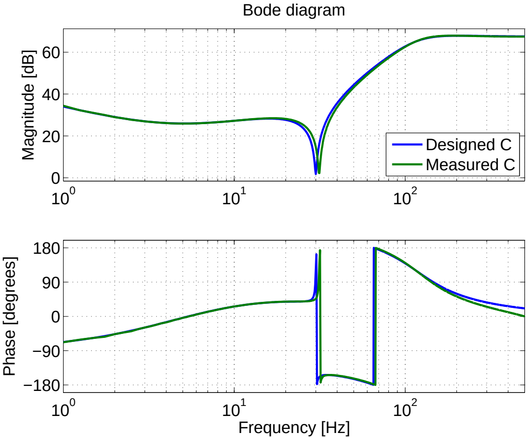
Controller blocks are discretized versions of continuous filters:

- using Tustin with prewarping
  (prewarp frequency differs per block)

DCtools

ShapeIt comes with the DCtools blockset.
Controller blocks are discretized versions of continuous filters:

- using Tustin with prewarping
  (prewarp frequency differs per block)



Technische Universiteit Eindhoven University of Technology

The best alternative. . . ?

Alternatively, discretize $C(s)$ yourself

- choose an appropriate method
- choose a sample frequency
- compare $C(s)$ and $C(z)$
- make adjustments in $C(s)$, f_s or the method if necessary
- implement the obtained $C(z)$

Note: If $C(s)$ and $C(z)$ differ, try to reason how it would alter the open-loop. Will $C(z)$ make it worse or better?

Always measure your controller before implementing!