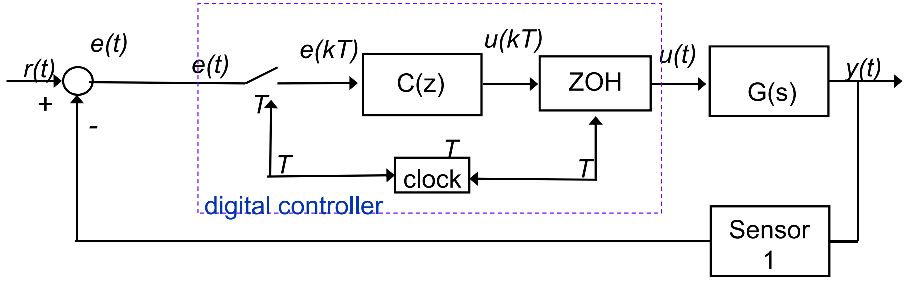
Automatic Control

Introduction to digital control

- **Digital control design by emulation**

Digital control design by emulation: preliminary steps

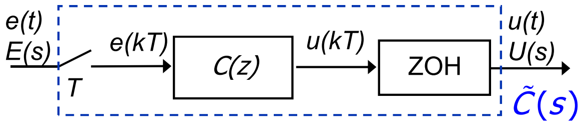
Now, the problem of designing the controller C(z) for the sampled data control system structure below is considered.



A possible procedure is suggested by the structure of the digital controller → …

Digital control design by emulation: preliminary steps

The structure of the digital controller is



and it is equivalent to an analog controller with transfer function

$$\tilde{C}(s) = G_{ZOH}(s)C(e^{sT})\frac{1}{T}$$

Control performance depends on the frequency response (e.g. the Nichols plot) of the (analog) loop function

$$\tilde{L}(s) = \tilde{C}(s)G(s), \tilde{C}(s) = \underbrace{G_{ZOH}(s)}_{D/A}C(e^{sT})\underbrace{\frac{1}{T}}_{A/D}$$

$$\rightarrow \tilde{L}(j\omega) = \tilde{C}(j\omega)G(j\omega) = G_{ZOH}(j\omega)C(e^{j\omega T})\frac{1}{T}G(j\omega)$$

Digital control design by emulation: preliminary steps

Thus, $C(z)$ should be designed so that the frequency response of the analog controller

$$\tilde{C}(s) = G_{ZOH}(s)C(e^{sT})\frac{1}{T}$$

$$\rightarrow \tilde{C}(j\omega) = G_{ZOH}(j\omega)C(e^{j\omega T})\frac{1}{T} = \frac{1-e^{-j\omega T}}{j\omega}C(e^{j\omega T})\frac{1}{T} =$$

$$\simeq Te^{-j\omega T/2}C(e^{j\omega T})\frac{1}{T} = e^{-j\omega T/2}C(e^{j\omega T}), \omega \in [0, \omega_N]$$

satisfies frequency domain constraints imposed, e.g., on the Nichols plane.

Digital control design by emulation: definition

$$\tilde{C}(j\omega) \simeq e^{-j\omega T/2} C(e^{j\omega T}), \omega \in [0, \omega_N]$$

… on the other hand, the loop-shaping continuous-time approach is a well established control design procedure.

In this regard, a suitable procedure for the design of the digital tf $C(z)$ consists in

1. looking for an analog controller $C_0(s)$ such that the loop function frequency response
$$\tilde{L}_0(j\omega) = \tilde{C}(j\omega)G(j\omega) \simeq e^{-j\omega T/2} C_0(j\omega)G(j\omega), \omega \in [0, \omega_N]$$
satisfies the control requirements
2. computing a discrete equivalent $C(z)$ of $C_0(s)$ so that
$$C(e^{j\omega T}) \simeq C_0(j\omega), \omega \in [0, \omega_N]$$

This design procedure is referred to as **emulation.**

Digital control design by emulation: overview

The main steps of the **emulation** design method are

1. Choose a suitable sampling period T
2. Design an analog controller $C_0(s)$ using any method (e.g. frequency response loop-shaping)
3. Obtain through a suitable discretization procedure the digital controller $C(z)$
4. Verify through simulation time domain performance
5. If needed, design anti-aliasing filter
6. If needed, perform modifications of $C_0(s)$ in order to meet the requirements

Sampling time selection

Digital controllers design by emulation: details

Choice of a suitable sampling period T

Several aspects have to be taken into account.

- Sampling Theorem

  the sampling frequency must at least equal twice the value of the highest significant frequency in the signal.

  → in a feedback system the highest significant frequency of all the signals in the loop is the system bandwidth ω_B , therefore a lower bound for the sampling frequency ω_s is

$$\omega_s > 2\,\omega_B \rightarrow \omega_s > 3\,\omega_c$$

(for a well damped system: $\omega_B \geq 1.5\,\omega_c$)

Digital controllers design by emulation: details

Choice of a suitable sampling period T

- ZOH filter:

the D/A converter introduces a phase lag of

$$\angle G_{ZOH}(j\omega) = -\frac{\omega T}{2}$$

In order to limit the phase lag at the cross-over frequency to small values (e.g. – 10° to - 5°) the following bound should be considered

$$\angle G_{ZOH}(j\omega) = -\frac{\omega T}{2} = -\frac{\omega \pi}{\omega_s}$$

$$-10^\circ < -\frac{\omega\pi}{\omega_s} < -5^\circ \rightarrow 0.087 \text{ rad} < \frac{\omega\pi}{\omega_s} < 0.17 \text{ rad}$$

$$\underset{\omega=\omega_c}{\rightarrow} 18\omega_c < \omega_s < 36\omega_c$$

Digital controllers design by emulation: details

Choice of a suitable sampling period T

- Suitable sampling of the transient:

a suitable number of samples must considered for describing the behavior of the transient phase

typically 10 to 50 samples can be employed for a suitable description of the transient behavior

for a well damped system (e.g. ζ = 0.6) we have

$$t_{s,1\%}\omega_c \approx 5.5 \rightarrow \frac{t_{s,1\%}}{50} < T < \frac{t_{s,1\%}}{10}$$

$$\frac{0.11}{\omega_c} < T < \frac{0.55}{\omega_c} \rightarrow \frac{0.11}{\omega_c} < \frac{2\pi}{\omega_s} < \frac{0.55}{\omega_c}$$

$$\underset{\omega=\omega_c}{\rightarrow} 11.42\omega_c < \omega_s < 57.2\omega_c$$

Digital controllers design by emulation: details

Choice of a suitable sampling period T

- Limitations → “small” T
  - improve conversion accuracy and destabilizing effects
  - increase the cost of A/D and D/A devices (more than linearly)
  - emphasizes quantization effects

  a trade-off among such criteria is made for choosing T

A final practical rule of thumb is to choose T within the interval

$$20\ \omega_c < \omega_s < 50\ \omega_c \rightarrow 0.12/\omega_c < T < 0.3/\omega_c$$

provided that the chosen values satisfies HW and cost limitations

Analog controller design

Digital controllers design by emulation: details

Design an analog controller $C_0(s)$ using any method

If it is known in advance that the controller has to be realized through a digital computer, the design of $C_0(s)$ is performed taking into account the dynamics introduced by the A/D and the ZOH D/A transfer function

$$G_{A/D}(s) = \frac{1}{T} \qquad G_{ZOH}(s) = \frac{1-e^{-Ts}}{s}$$

Thus, the analog controller design should be performed considering as plant transfer function:

$$G'(s) = \frac{1}{T} G(s) \frac{1-e^{-Ts}}{s}$$

Remark : in MatLab environment, use the 1st order Padé approximation:

$$G_{ZOH}(s) = \frac{1-e^{-Ts}}{s} \approx \frac{T}{1+sT/2} \rightarrow G'(s) = G(s)\frac{1}{1+sT/2}$$

Discretization of continuous-time LTI systems

Introduction to discretization methods

In order to derive a discrete equivalent of the analog controller $C_0(s)$, the sampling transformation defined by $z = e^{sT}$, can be employed

In particular, the discrete time controller tf $C(z)$, can be found using the inverse of the sampling transformation:

$$s = \frac{1}{T}\log(z) \quad \Rightarrow \quad C(z) = C_0\left(\frac{1}{T}\log(z)\right)$$

However, the obtained $C(z)$ is not provided by a real rational function, and thus it does not represent a finite dimensional discrete time system

Introduction to discretization methods

A real rational form for $C(z)$ can be obtained through a procedure which aims at computing a linear discrete time system whose input-output relationship is made up by a finite difference equation of the form

$$u(k) = -a_{n-1}u(k-1) - a_{n-2}u(k-2) - \ldots - a_0 u(k-n) + \\ + b_n e(k) + b_{n-1}e(k-1) + \ldots + b_0 e(k-n)$$

which approximates the behavior of the continuous time dynamical system represented by the tf $C_0(s)$

Such procedure is referred to as **discretization** of continuous time dynamical systems

Discretization through the bilinear transformation

The simplest and most effective way to discretize a continuous time controller $C_0(s)$ is to consider its state space representation

$$\begin{cases}\dot{x}(t)=Ax(t)+Be(t)\\ u(t)=Cx(t)+De(t)\end{cases}\rightarrow C_0(s)=\frac{U(s)}{E(s)}=C(sI-A)^{-1}B+D$$

and suppose to integrate it through a numeric method, such that:

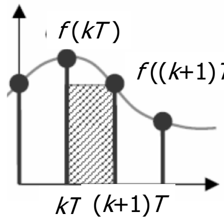
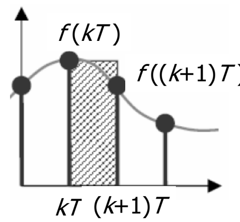
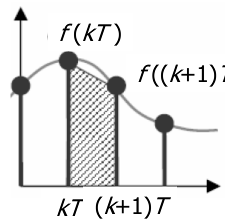
$$\int_{kT}^{(k+1)T}\dot{x}(t)dt=\int_{kT}^{(k+1)T}Ax(t)dt+\int_{kT}^{(k+1)T}Be(t)dt$$

$$\underset{\substack{\uparrow\\ x(kT)=x^*(k)\\ x((k+1)T)=x^*(k+1)}}{\rightarrow}\quad x^*(k+1)-x^*(k)=A\int_{kT}^{(k+1)T}x(t)dt+B\int_{kT}^{(k+1)T}e(t)dt$$

Discretization through the bilinear transformation

The integral $\int_{kT}^{(k+1)T}f(t)dt$ represents the area under the function $f(t)$ between $t=kT$ and $t=(k+1)T$. Such an area can be approximated as

$$\int_{kT}^{(k+1)T}f(t)dt\simeq[(1-\alpha)f(kT)+\alpha f((k+1)T)]T,\ 0\le\alpha\le 1$$

$\alpha = 1$ → Bacward Euler	$\alpha = 0$ → Forward Euler	$\alpha = 0.5$ → Tustin
		
Area = $Tf((k+1)T)$	Area = $Tf(kT)$	Area = $0.5\,T(f((k+1)T)+f(kT))$

Discretization through the bilinear transformation

Thus we have

$$x^*(k+1)-x^*(k)\simeq A[(1-\alpha)x^*(k)+\alpha x^*(k+1)]T+$$
$$+B[(1-\alpha)e^*(k)+\alpha e^*(k+1)]T$$

$$\rightarrow\left(\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}I-A\right)X^*(z)=BE^*(z)$$

$$\rightarrow X^*(z)=\left(\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}I-A\right)^{-1}BE^*(z)$$

$$u^*(k)=Cx^*(k)+De^*(k)\rightarrow U^*(z)=CX^*(z)+DE^*(z)$$

$$\rightarrow U^*(z)=\underbrace{\left[C\left(\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}I-A\right)^{-1}B+D\right]}_{C(z)}E^*(z)$$

Discretization through the bilinear transformation

Then:

$$\left.\begin{aligned}&C_0(s)=C(sI-A)^{-1}B+D\\ &C(z)=C\left(\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}I-A\right)^{-1}B+D\\ &\rightarrow C(z)=C_0(s)\Big|_{s=\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}}\end{aligned}\right\}\rightarrow\quad s=\frac{1}{T}\frac{z-1}{\alpha z+1-\alpha}$$

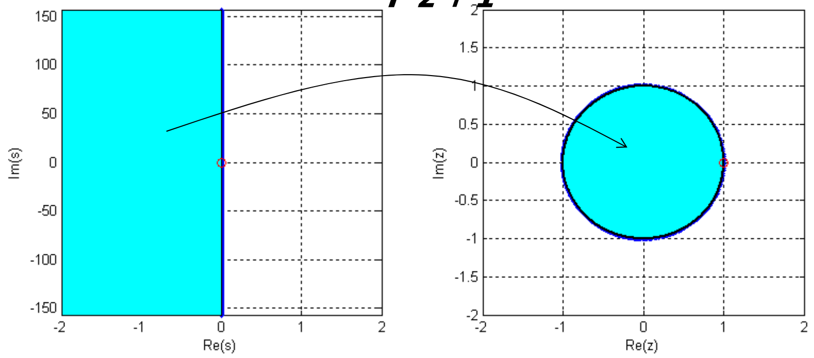
bilinear transformation

$$\text{In particular}\rightarrow\begin{cases}\alpha=0.5\ \rightarrow\ \textbf{Tustin:} & s=\frac{2}{T}\frac{z-1}{z+1}=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\\ \alpha=1\ \rightarrow\ \textbf{Backward Euler:} & s=\frac{1}{T}\frac{z-1}{z}=\frac{1-z^{-1}}{T}\\ \alpha=0\ \rightarrow\ \textbf{Forward Euler:} & s=\frac{z-1}{T}=\frac{1}{T}\frac{1-z^{-1}}{z^{-1}}\end{cases}$$

Discretization through the bilinear transformation

Mapping properties of the Tustin approximation

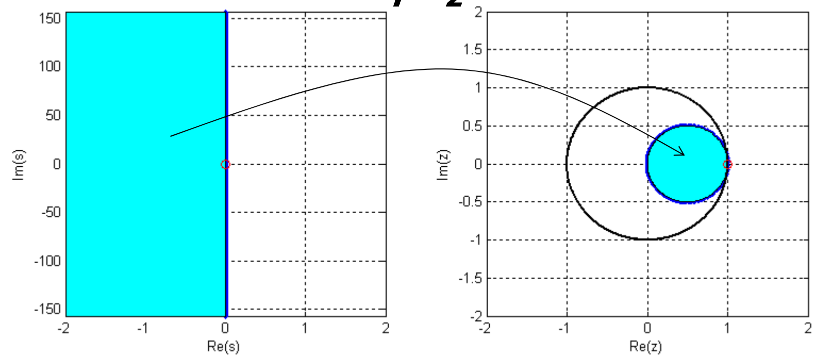
$$s = \frac{2}{T}\frac{z-1}{z+1}$$



Discretization through the bilinear transformation

Mapping properties of the backward Euler approximation

$$s = \frac{1}{T}\frac{z-1}{z}$$

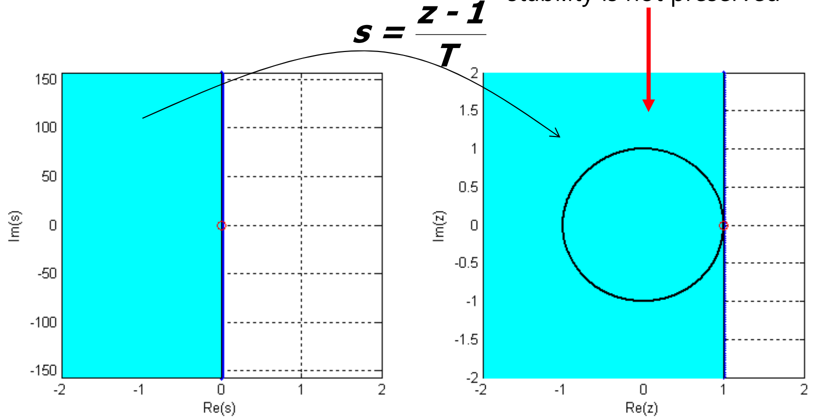


Discretization through the bilinear transformation

Mapping properties of the forward Euler approximation

$$s = \frac{z-1}{T}$$

stability is not preserved



Discretization through the bilinear transformation

Example: discretize the following analog controller with $T = 1$ s

$$C_0(s) = 2\frac{1+\frac{s}{0.1}}{1+\frac{s}{10}}, K_c = \lim_{s \to 0} C_0(s) = 2$$

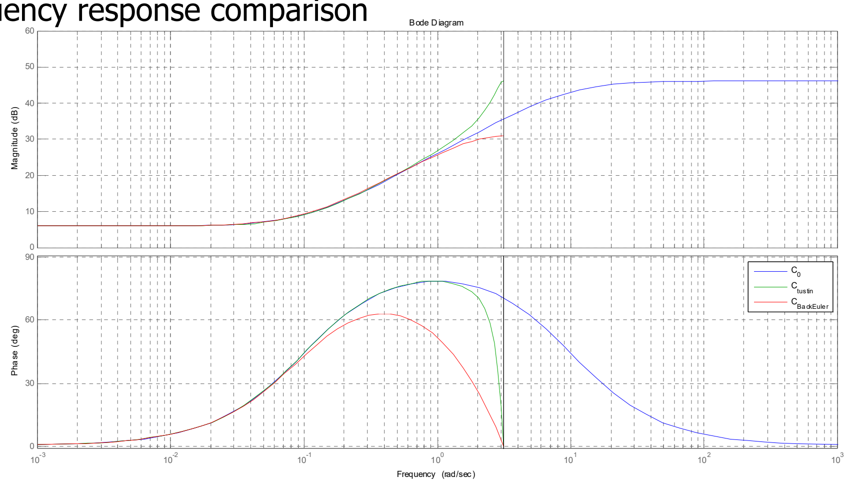
$\to$ **Tustin :** $\quad C(z) = \dfrac{35z - 31.67}{z+0.6}, K_c = \lim_{z \to 1} C(z) = 2$

$\to$ **Backward Euler :** $\quad C(z) = \dfrac{20z - 18.18}{z-0.09}, K_c = \lim_{z \to 1} C(z) = 2$

$\to$ **Forward Euler :** $\quad C(z) = \dfrac{200z - 180}{z+9} \to$ unstable

Discretization through the bilinear transformation

Frequency response comparison



The Tustin approximation provides the best frequency response matching of the analog controller

Discretization through zero-pole-gain matching

An alternative approach for analog controller discretization can be obtained through the application of the sampling transformation $z = e^{sT}$ to zeros and poles of the analog controller $C_0(s)$

Given the zpk form

$$C_0(s) = K\frac{(s-q_1)(s-q_2)\ldots(s-q_n)}{s^r(s-p_1)(s-p_2)\ldots(s-p_{n-r})}$$

The **matched pole-zero (MPZ)** method: the discretized digital controller $C(z)$ is obtained as:

$$C(z) = K_d\frac{(z-e^{q_1T})(z-e^{q_2T})\ldots(z-e^{q_nT})}{(z-1)^r(z-e^{p_1T})(z-e^{p_2T})\ldots(z-e^{p_{n-r}T})}, K_d = K\frac{\Pi_{i=1}^{n} q_i \Pi_{i=1}^{n-r}(1-e^{p_iT})}{\Pi_{i=1}^{n-r} p_i \Pi_{i=1}^{n}(1-e^{q_iT})}$$

Remark: K_d is computed in order to guarantee that $C_0(s)$ and $C(z)$ have the same (generalized) DC gain

Discretization through zero-pole-gain matching

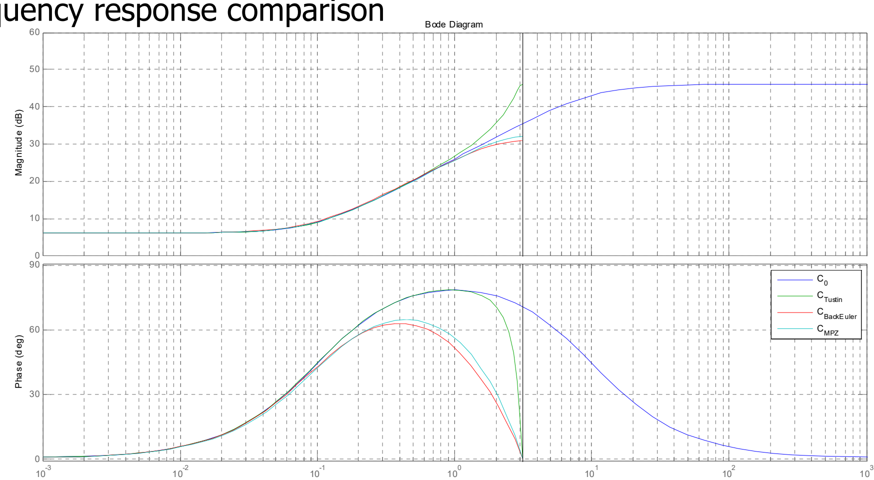
Example: ($T = 1$ s)

$$C_0(s) = 2\frac{1+\dfrac{s}{0.1}}{1+\dfrac{s}{10}} = 200\frac{s+0.1}{s+10}, K_c = \lim_{s\to 0} C_0(s) = 2$$

$$\textbf{MPZ:}\quad C(z) = 21.0157\frac{z-e^{-0.1}}{z-e^{-10}} = 21.0157\frac{z-0.9048}{z-4.54\cdot 10^{-5}}, \lim_{z\to 1} C(z) = 2$$

Discretization through the bilinear transformation

Frequency response comparison



The Tustin approximation still provides the best frequency response matching of the analog controller

Discretization methods

Discretization of a transfer function with MatLab

```
>> C = c2d(C_0, T, METHOD)

  METHOD:
     'zoh'            Zero-order hold
     'foh'            First-order hold
     'imp'            Impulse-invariant
   →'tustin'          Tustin
     'prewarp'        Tustin approximation with frequency
                      prewarping
    →'matched'        Matched pole-zero
```

The default is 'zoh' when METHOD is omitted

Anti-aliasing filter design

Digital controllers design by emulation: details

Antialising filter design

- A Butterworth filter of the form

$$F(s) = \frac{1}{B_n(s)} \rightarrow s' = \frac{s}{\omega_f}$$

order n	$B_n(s)$
1	$(s'+1)$
2	$(s'^2+1.414s'+1)$
3	$(s'^2+s'+1)(s'+1)$
4	$(s'^2+0.765s'+1)(s'^2+1.848s'+1)$

can be employed as antialising filter

Digital controllers design by emulation: details

Antialising filter design

It is required that the cut-off feauency ω_f satisfies:

$$\omega_B < \omega_f < \frac{\omega_s}{2}$$

The filter can be designed by imposing a given attenuation γ (e.g. γ = 0.1) for all the frequencies greater than a given ω_h (e.g. $\omega_h = \omega_s/2$):

$$|F(j\omega)| \leq \gamma, \forall \omega \geq \omega_h$$

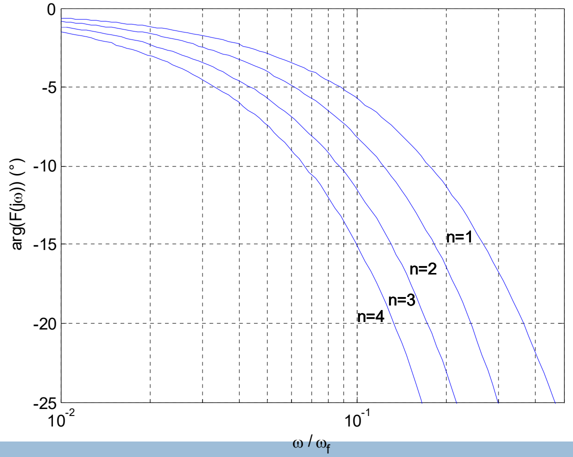
We have

$$\omega_f = \omega_h \left(\frac{\gamma^2}{1-\gamma^2} \right)^{\frac{1}{2n}} \underset{\gamma \ll 1}{\underset{\uparrow}{\cong}} \omega_h \gamma^{\frac{1}{n}}$$

Digital controllers design by emulation: details

Antialiasing filter design

The order n of the filter is chosen in order to limit the phase lag introduced at ω_c using the normalized filter phase diagram



Digital controllers design by emulation: details

Example: Design a Butterworth filter to introduce an attenuation $\gamma = 0.1$ at $\omega_h = 10$ rad/s for a control system with $\omega_c = 0.2$ rad/s

$n = 2 \rightarrow$

$\omega_f = \omega_h \gamma^{1/n} = 10 \cdot 0.1^{1/2} = 3.16$

$\omega_c / \omega_f = 0.2/3.16 = 0.063$

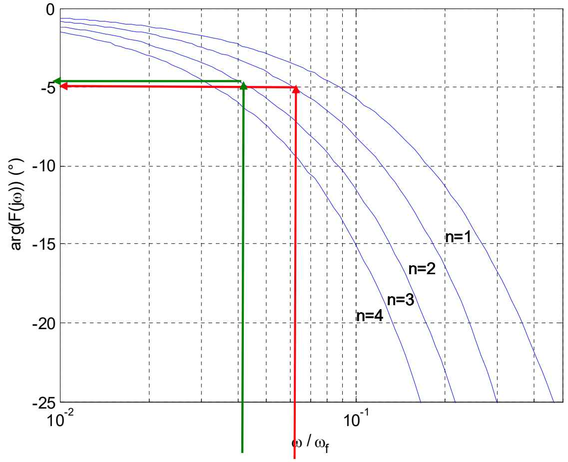
$\rightarrow \varphi_{lag} \simeq -5°$

$n = 3 \rightarrow$

$\omega_f = \omega_h \gamma^{1/n} = 10 \cdot 0.1^{1/3} = 4.64$

$\omega_c / \omega_f = 0.2/4.64 = 0.043$

$\rightarrow \varphi_{lag} \simeq -4.9°$



Butterworth filters

Butterworth filter computation with MatLab

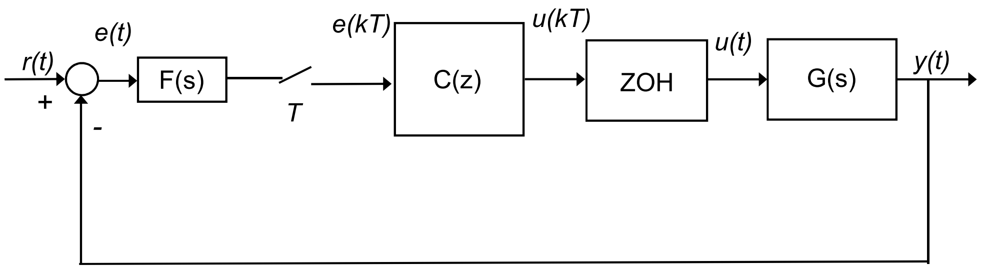
```
BUTTER Butterworth digital and analog filter design.

[B,A] = BUTTER(N,Wf) designs an Nth order lowpass digital
Butterworth filter and returns the filter coefficients in
length  N+1 vectors B (numerator) and A (denominator).
The coefficients are listed in descending powers of z.
The cutoff frequency Wf must be 0.0 < Wf < 1.0, with 1.0
corresponding to half the sample rate.

[B,A] = BUTTER(N,Wf,'s'), design analog Butterworth
filters.  In this case, Wf is in [rad/s] and it can be
greater than 1.0.
```


Performance evaluation through simulation

Digital controllers design by emulation: details

Verify through simulation time domain performance

- Simulink simulation is needed to verify time domain performance



Remark: in the Simulink scheme, blocks corresponding to A/D and D/A may be omitted