POLITECNICO DI MILANO

Automatic Control

Digital control systems

Prof. Luca Bascetta (luca.bascetta@polimi.it)

Politecnico di Milano

Dipartimento di Elettronica, Informazione e Bioingegneria

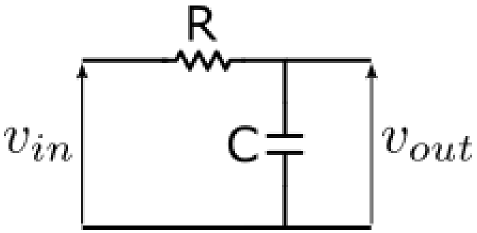
Motivations

All the design techniques we have discussed yield the transfer function $R(s)$ of the regulator.

How can we implement this transfer function on a real control system?

Devising a physical system whose behavior is described by the controller transfer function (analog control system).

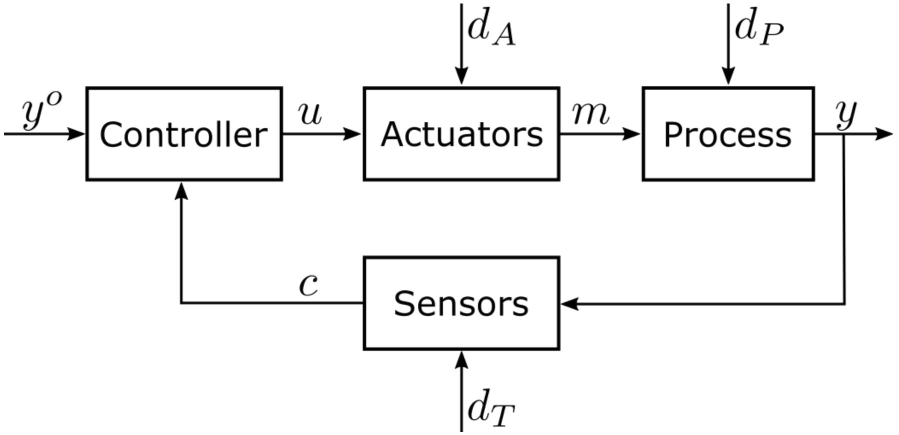
$$R(s) = \frac{1}{1+sT} \quad T = RC$$



Implementing the control system as an algorithm on a microprocessor, i.e., a PC or an embedded system (digital control system).

Let’s introduce the fundamental characteristics of digital control systems.

Introduction to digital control systems (I)



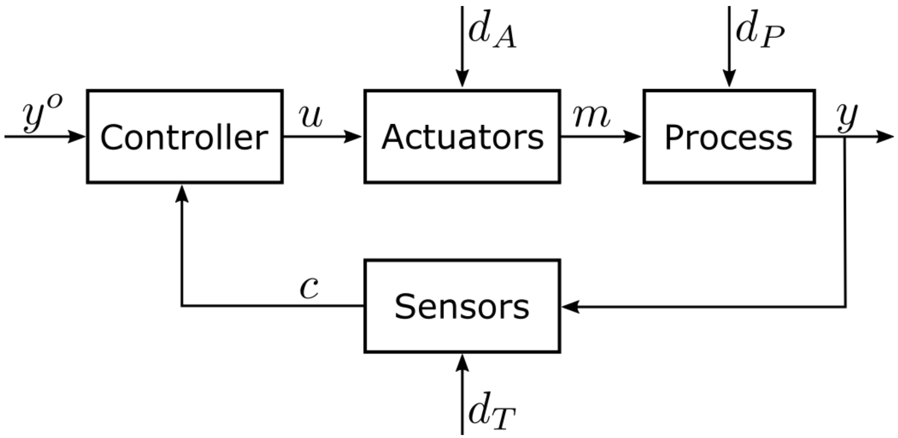
Each signal in this block diagram is a continuous function, i.e. $u(t): \mathbb{R} \to \mathbb{R}$. We call these functions analog signals.

A microprocessor, however, cannot read analog signals… a microprocessor

- cyclically executes an instruction every T milliseconds/microseconds
- cyclically updates its internal time every T milliseconds/microseconds
- does not work as a continuous, but as a discrete time system
- its variables are represented as words with a finite number of bits, i.e. finite precision numbers

We conclude that a microprocessor works on digital signals.

Introduction to digital control systems (II)



Excluding the controller, the “rest of the world” (actuators, sensors, process) works on analog signals.

In order to develop our control system on a microprocessor we have to introduce converters able to translate signals from the analog to the digital domain.

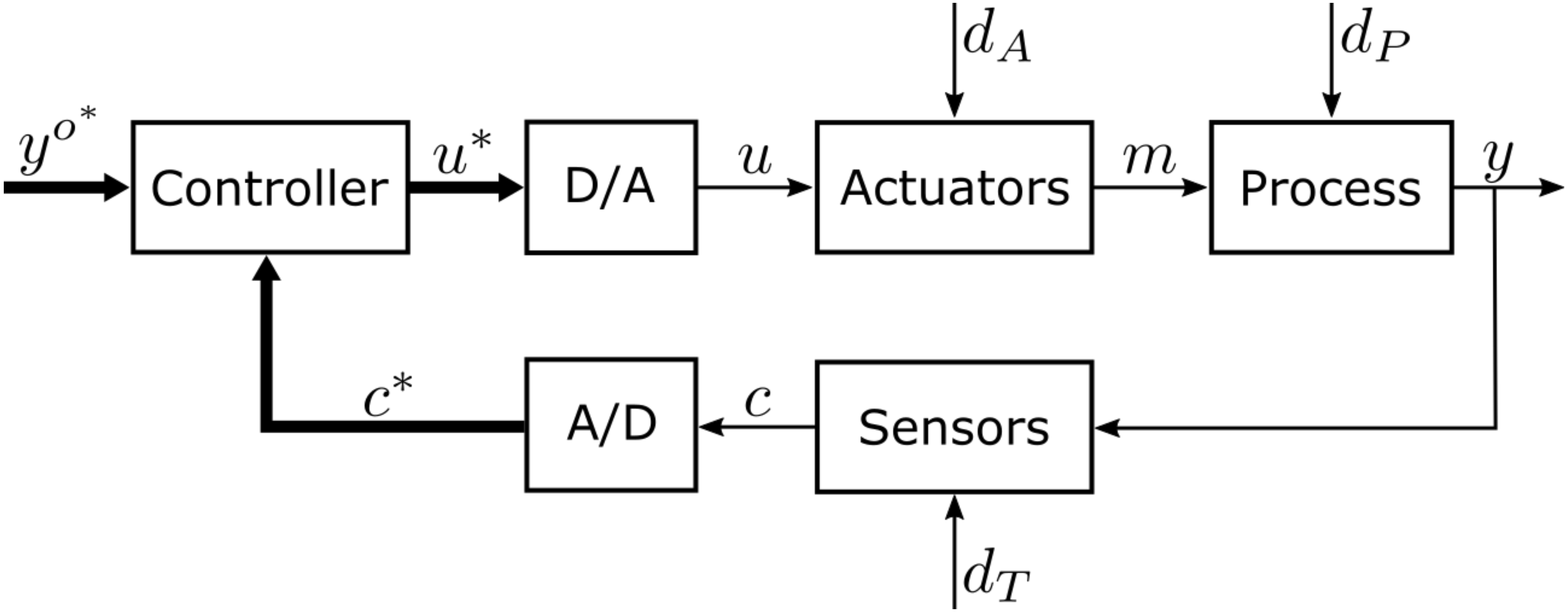
We will call these converters:

- A/D, analog to digital converter
- D/A, digital to analog converter

What happens if we introduce converters in our control architecture?

Introduction to digital control systems (III)

The control architecture becomes…



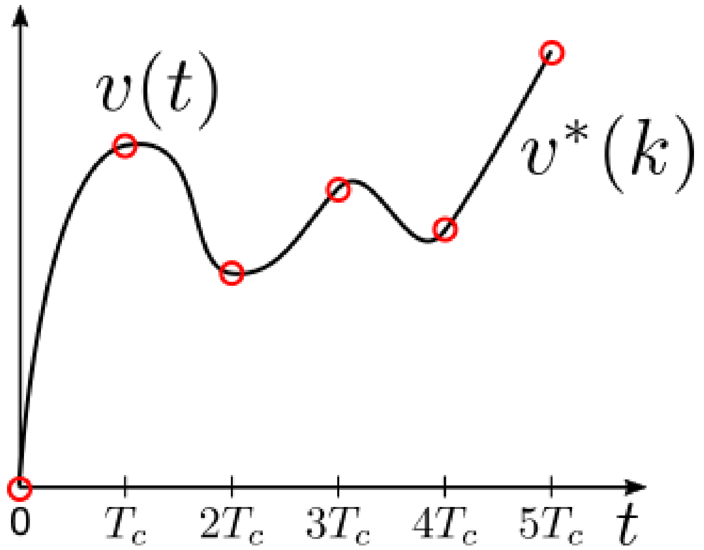
We are now interested to analyze:

- how the converts work
- how the introduction of the converters affects the closed-loop system
- how we can convert the controller transfer function into an algorithm for the microprocessor

A/D converter (I)

Let's consider

- an analog signal $v(t)$
- a sequence of time intervals of length T_c, the first one starting at $t = 0$



If we take a value of $v(t)$ every T_c time instants, we obtain the following sequence

$$v^*(k) = v(kT_c) \quad k = 0, 1, 2, \ldots$$

of <u>samples</u>.

A/D converter (II)

We introduce the following definitions:

- sampling, is the reduction of a continuous time signal $v(t)$ to a discrete time signal $v^*(k)$
- T_c is the sampling interval
- $f_c = 1/T_c$ ($\Omega_c = 2\pi/T_c$) is the sampling frequency or sampling rate
- $f_s = 1/2T_c$ ($\Omega_N = \pi/T_c$) is the Nyquist frequency

A/D conversion entails two different operations:

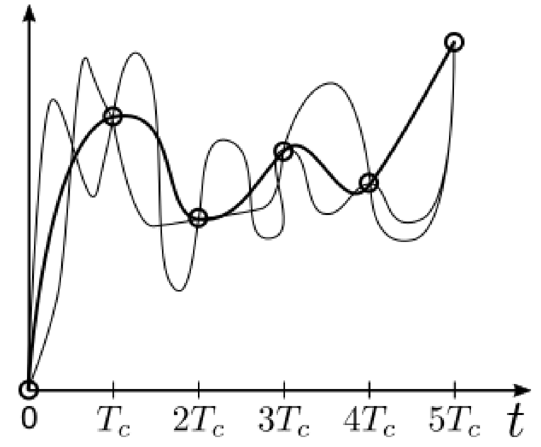
- the reduction of a signal from continuous to discrete time (sampling)
- the conversion of each sample into a finite precision number (quantization)

Quantization is a nonlinear and complex operation. We will neglect it, as a correct design, i.e., selection of the number of bits of the converter, makes it negligible.

We will, instead, focus on the sampling operation.

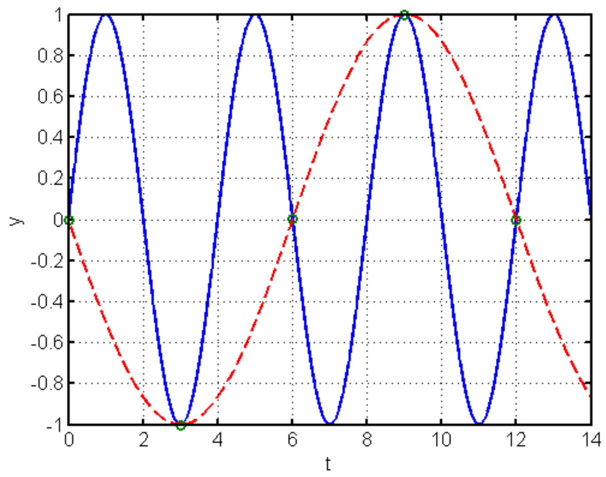
A/D converter (III)

How the introduction of sampling affects the closed-loop system?



Sampling obviously causes a loss of information, the question is how we should select the sampling rate in order to keep this loss negligible, and allow for a reconstruction of the analog signal from samples.

If we consider a sinusoidal signal (remember that each signal can be represented as a summation of sinusoids), we can easily show that the signal reconstructed from few samples can be a sinusoid with a larger period.

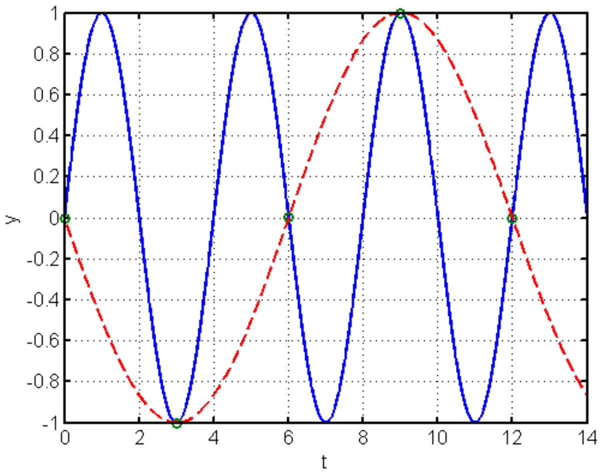


This effect is called aliasing.

We will now investigate how to select the sampling rate in order to avoid the aliasing effect and make the loss of information negligible.

A/D converter (IV)

Consider again the problem of sampling a sinusoidal signal.



In the previous example we selected the sampling period equal to

$$T_c = \frac{3}{4}\bar{T}$$

where $\bar{T}$ is the period of the sinusoidal signal.

Intuitively, we see that, in order to avoid aliasing, we need more than two samples for each period of the sinusoid.

We should thus select the sampling period in such a way that the constraint

$$\bar{T} > 2T_c$$

is satisfied, or equivalently

$$\bar{T} > 2T_c \quad \Rightarrow \quad \frac{2\pi}{\bar{T}} < \frac{\pi}{T_c} \quad \Rightarrow \quad \Omega_N > \bar{\omega}$$

<u>The Nyquist frequency should be higher than the sinusoid frequency.</u>

A/D converter (V)

If the Nyquist frequency is less or equal to the sinusoid frequency, aliasing harmonics arise.

It can be shown that the frequency of the lowest frequency aliasing harmonic is given by

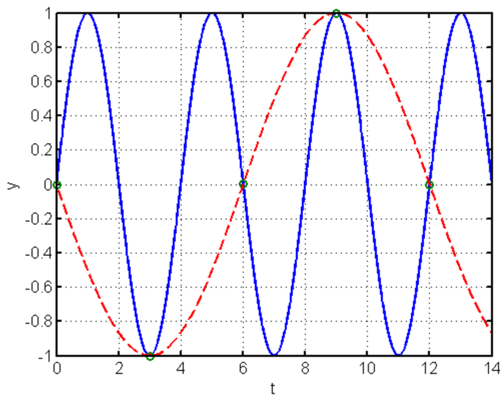
$$\omega_{al} = \left|\Omega_c - \bar{\omega}\right| \qquad \bar{\omega} < 3\Omega_N$$

In the previous example

$$T_c = \frac{3}{4}\bar{T} \quad \Rightarrow \quad \Omega_c = 2\pi\frac{4}{3\bar{T}} = \frac{4}{3}\bar{\omega}$$

and thus

$$\omega_{al} = \frac{4}{3}\bar{\omega} - \bar{\omega} = \frac{\bar{\omega}}{3} \quad \Rightarrow \quad T_{al} = 2\pi\frac{3}{\bar{\omega}} = 3\bar{T}$$



A/D converter (VI)

We can now generalize the previous conclusion.

We call band-limited signal a signal whose Fourier transform or power spectral density has bounded support, i.e., it is almost zero for all the frequencies higher than the maximum frequency Ω_v.

Nyquist-Shannon sampling theorem

Given a band-limited signal $v(t)$, whose maximum frequency is Ω_v, if

$$\Omega_v < \Omega_N$$

we can reconstruct the original signal $v(t)$ from its samples $v^*(k)$, without loss of information.

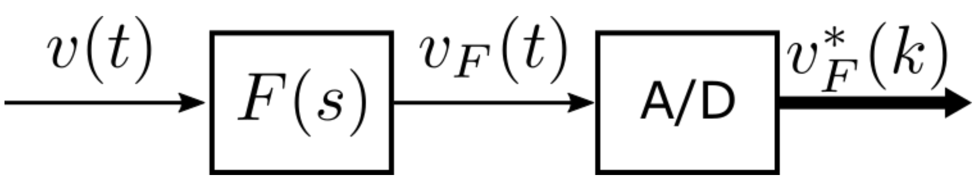
The original signal can be reconstructed using the Whittaker–Shannon interpolation formula

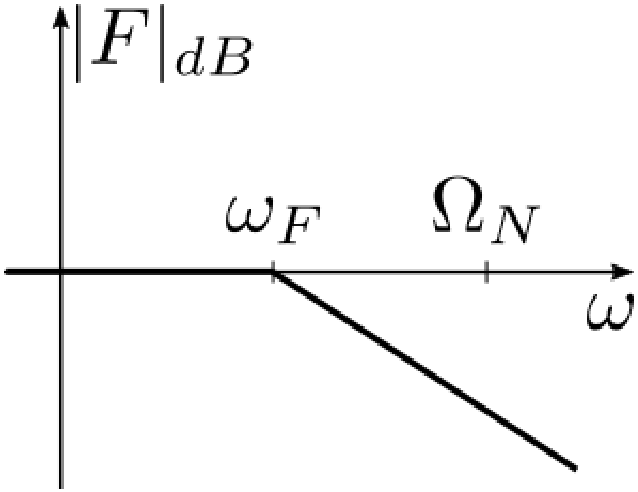
$$v(t) = \sum_{k=-\infty}^{+\infty} \left[v^*(k) \frac{\sin\left(\Omega_N t - k\pi\right)}{\Omega_N t - k\pi} \right]$$

In real applications, due to the presence of noise, no signal is band-limited.

In order to enforce the satisfaction of the Shannon theorem constraint, we need to introduce a low-pass filter called anti-aliasing filter.

The cutoff frequency of this filter should be selected in such a way that the residual harmonics of the filtered signal at the Nyquist frequency are negligible.





D/A converter (I)

D/A conversion is the invers of A/D conversion, i.e., we would like to reconstruct an analog signal from a sequence of samples.

Assuming that the signal has been sampled with a sample rate that satisfies the Shannon theorem, can we use the Shannon interpolation formula?

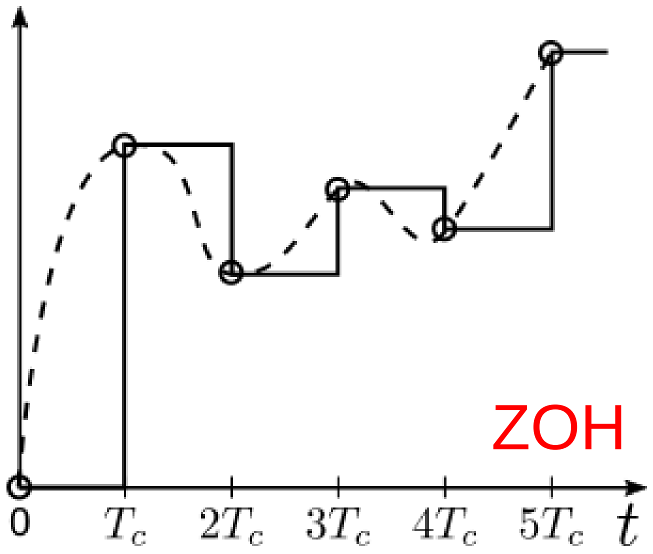
$$v(t) = \sum_{k=-\infty}^{+\infty} \left[v^*(k) \frac{\sin(\Omega_N t - k\pi)}{\Omega_N t - k\pi} \right]$$

We cannot use this formula in a control system, because it is <u>a-causal</u>: to reconstruct a time instant of the analog signal we need all the past and future history of the digital signal.

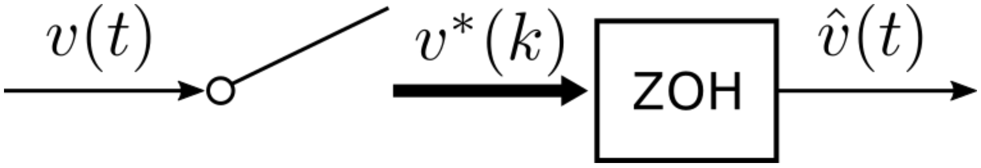
In order to solve this problem we can use an approximation of the Shannon formula based on extrapolation techniques, i.e., given a subset of the past history of the signal the converter computes the output value.

An example of these devices is the <u>Zero-Order Hold (ZOH)</u>.

D/A converter (II)

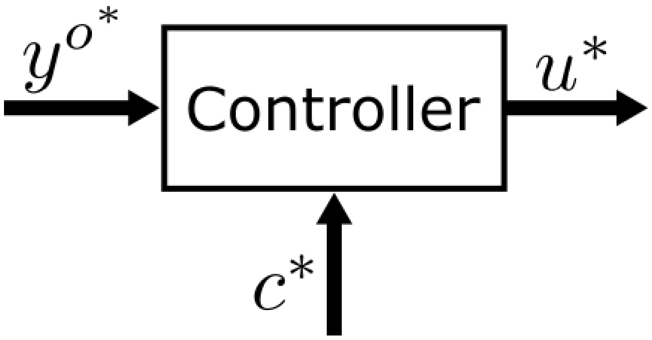


As can be seen from the image the sampling and reconstruction introduce a delay in the signal called sample-and-hold delay. It can be demonstrated that this delay is equal to half of the sampling period.



Digital controller (I)

The digital controller is an algorithm that computes in real-time the control variable from the controlled and reference signals.



This algorithm can be arbitrarily complex,
we will only consider algorithms that represent the input-output relation of a discrete time linear time-invariant dynamical system, i.e.

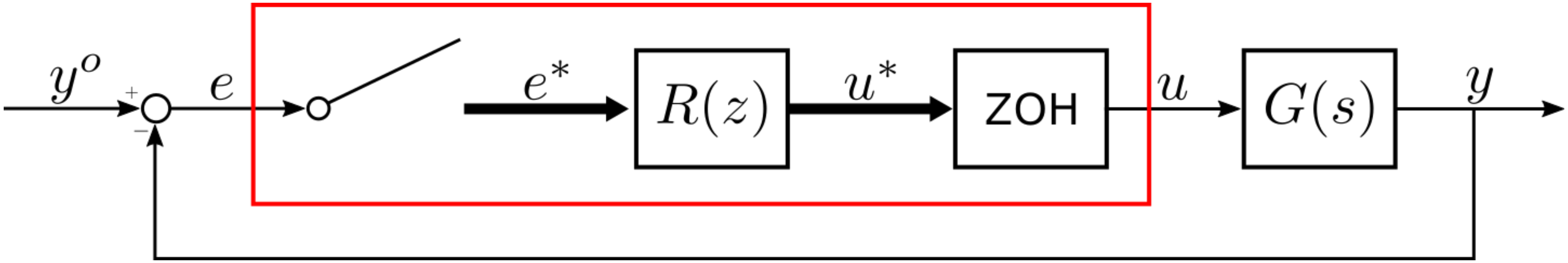
$$R(z) = \frac{U^*(z)}{E^*(z)}$$

We have now to address the last issue, how to compute the transfer function $R(z)$.

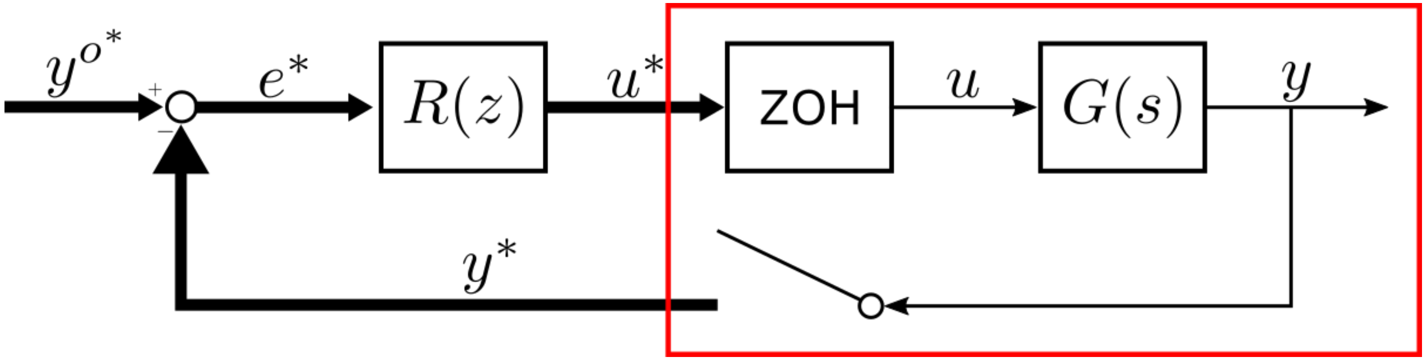
Digital controller (II)

We can approach the design of the digital control system following two ways.

Indirect digital controller design

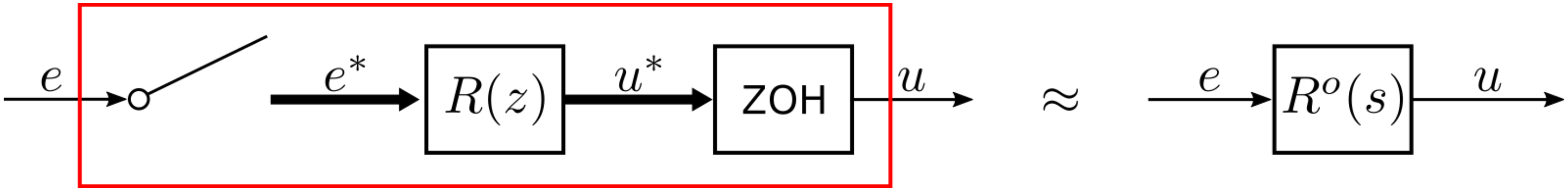


Direct digital controller design



Indirect digital controller design (I)

Assuming that a continuous time regulator $R^o(s)$ has been already designed on $G(s)$, the aim of indirect digital control design technique is to find a discrete time transfer function $R(z)$ in such a way that the input-output relation of the red block is as similar as possible to the one of the continuous time regulator $R^o(s)$.



To achieve this result we have to:

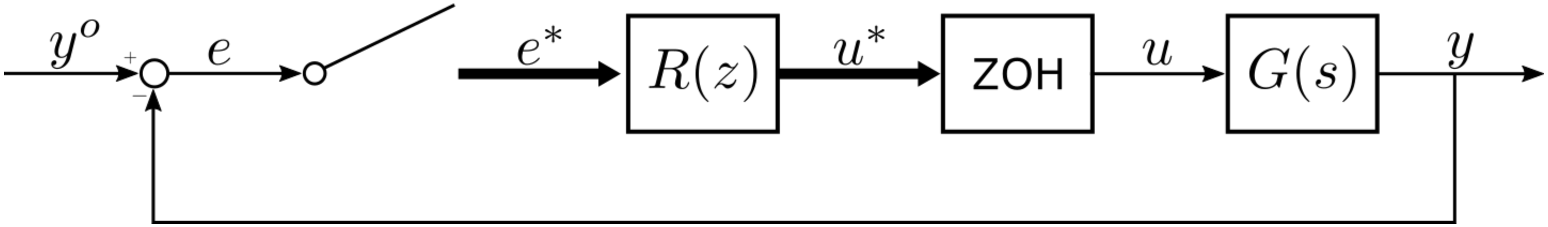
- select a suitable sampling time T_c
- determine the transfer function $R(z)$ from $R^o(s)$

Indirect digital controller design (II)

Let's start from the selection of a suitable sampling time T_c.

From Shannon theorem we know that the Nyquist frequency should be higher than the maximum frequency in the signal we would like to sample. But…

- which signal should we consider in a closed-loop system?
- is it a band-limited signal?
- which is its maximum frequency?



Some answers:

- we sample the error signal
- in a closed-loop system all the signals have a maximum frequency that is well approximated by the crossover frequency ω_c (all the harmonics with a frequency higher than ω_c can be neglected)

Indirect digital controller design (III)

We conclude that, in order to satisfy the Shannon theorem, the sampling frequency should be selected in such a way that

$$\omega_c \ll \Omega_N$$

As a rule of thumb we can adopt the following criterion

$$\Omega_N = 10\omega_c$$

The presence of measurement noise or high frequency components in the reference signal, however, can violate the band-limited constraint.
To avoid this problem we should introduce an anti-aliasing filter before the sampling device.
The filter cut-off frequency should be:

- higher than the crossover frequency
  - no filtering action inside the closed-loop system bandwidth
  - avoid excessive phase margin decrease
- lower than the Nyquist frequency
  - to make the error a band-limited signal

To correctly select the sampling rate, we should also remember that the introduction of the sampler and holder devices is equivalent to a delay in the loop of $T_c/2$.
This delay causes a decrement of the phase margin equal to

$$\Delta\varphi_m = \omega_c \frac{T_c}{2} \frac{180^\circ}{\pi} = 90^\circ \frac{\omega_c}{\Omega_N}$$

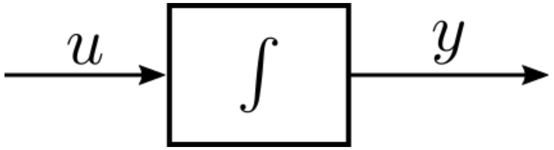
The design of the analog controller $R^o(s)$ should thus ensure an adequate phase margin, so that the introduction of the sampler and holder does not affect too much the stability and performance of the closed-loop system.

The last problem we have to address is the derivation of the transfer function $R(z)$ from $R^o(s)$.

Indirect digital controller design (V)

Consider a continuous time integrator

$$\dot{y}(t) = u(t)$$



If we define

$$u^*(k) = u(kT) \quad y^*(k) = y(kT)$$

where T is the integration interval, we have

$$y^*(k) = y^*(k-1) + u_m(k)T$$

where $u_m(t)$ is the average value of $u(t)$ in the integration interval.

If we approximate $u_m(t)$ with a linear convex combination of the values of $u(t)$ at the first and last time instants

$$u_m(k) = (1-\alpha)\,u^*(k-1) + \alpha u^*(k) \quad \alpha \in [0,1]$$

we obtain

$$y^*(k) = y^*(k-1) + T\left[(1-\alpha)\,u^*(k-1) + \alpha u^*(k)\right]$$

Indirect digital controller design (VI)

Applying now the Z transform to the previous relation

$$y^*(k) = y^*(k-1) + T\left[(1-\alpha)u^*(k-1) + \alpha u^*(k)\right]$$

we obtain

$$Y^*(z) = z^{-1}Y^*(z) + T\left[(1-\alpha)z^{-1} + \alpha\right]U^*(z)$$

We can now compare the continuous time and discrete time transfer functions of an integrator

$$\frac{Y^*(z)}{U^*(z)} = T\frac{(1-\alpha)+\alpha z}{z-1} \qquad \frac{Y(s)}{U(s)} = \frac{1}{s}$$

From this comparison we obtain a relation between s and z (bilinear transformation)

$$s = \frac{1}{T}\frac{z-1}{(1-\alpha)+\alpha z}$$

Indirect digital controller design (VII)

We conclude that the digital controller transfer function can be computed applying the bilinear transformation as follows

$$R(z) = R^o\left(\frac{1}{T}\frac{z-1}{(1-\alpha)+\alpha z}\right)$$

We also introduce the following common transformations

- $\alpha = 0$ (forward Euler method)

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

- $\alpha = 1$ (backward Euler method)

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

- $\alpha = 1/2$ (Tustin method)

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

Indirect digital controller design (VIII)

Caveat: The forward Euler method can give rise to an unstable discrete time system, even if the continuous time system is asymptotically stable, when the sampling time is not sufficiently small.

PI digital regulator (I)

Consider now an analog PI regulator

$$R^o(s) = K_P\left(1 + \frac{1}{sT_I}\right)$$

Using Tustin method we obtain the following discrete time transfer function

$$R(z) = R^o\left(\frac{2}{T}\frac{z-1}{z+1}\right) = K_P\left(1 + \frac{T}{2T_I}\frac{z+1}{z-1}\right)$$

$$= \frac{K_P}{2T_I}\left(\frac{(2T_I + T)z + T - 2T_I}{z-1}\right) = \Gamma_P\frac{z-b}{z-1}$$

where

$$\Gamma_P = K_P\left(1 + \frac{T}{2T_I}\right) \qquad b = \frac{2T_I - T}{2T_I + T}$$

In conclusion

$$\frac{U^*(z)}{E^*(z)} = \Gamma_P\frac{z-b}{z-1} = \Gamma_P\frac{1 - bz^{-1}}{1 - z^{-1}}$$

PI digital regulator (II)

The previous transfer function can be rewritten as

$$\left(1-z^{-1}\right)U^*(z)=\Gamma_P\left(1-bz^{-1}\right)E^*(z)$$

and

$$U^*(z)=z^{-1}U^*(z)+\Gamma_P E^*(z)-\Gamma_P bz^{-1}E^*(z)$$

Applying the inverse Z transform

$$u^*(k)=u^*(k-1)+\Gamma_P e^*(k)-\Gamma_P be^*(k-1)$$

This equation is an algorithm that allows to compute the control variable given the controlled signal and the reference, and it can be easily implemented on a microprocessor.

PI digital regulator (III)

$\Rightarrow$ Waiting for a new clock interrupt

$\Rightarrow$ Acquisition of y_{sp}, acquisition and A/D conversion of y

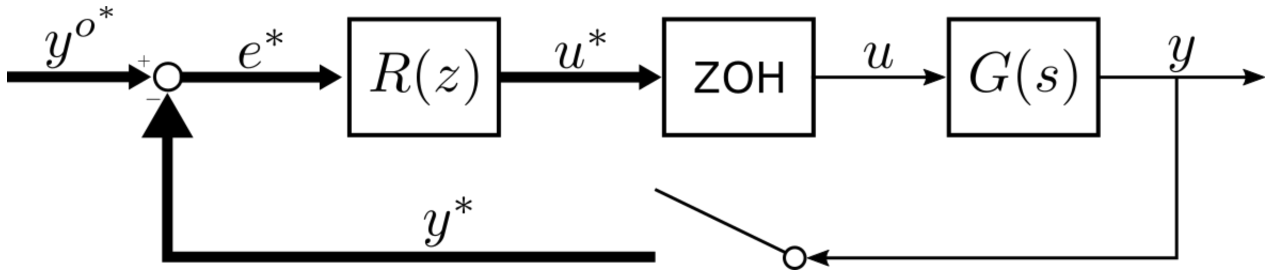
$$e = y_{sp} - y$$

$$u = u + \Gamma_P * e - \Gamma_P * b * e_{old}$$

$\Rightarrow$ D/A conversion of u

$$e_{old} = e$$

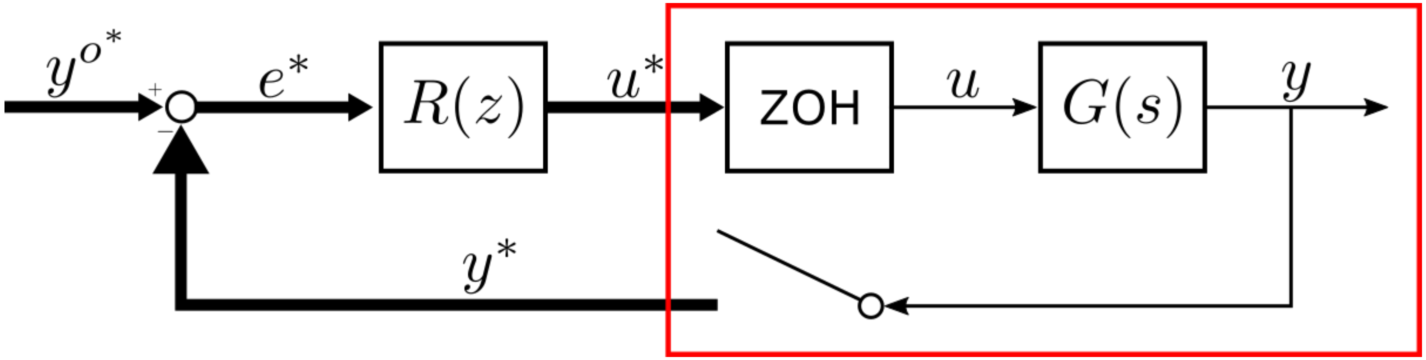
Direct digital controller design (I)



In this case the process, together with the A/D and D/A converters, is considered as a discrete time system, and the regulator is designed using discrete time system design tools.

We will not analyze discrete time system design tools.
We are just interested in the methodology used to compute an equivalent discrete time transfer function of the process, together with the A/D and D/A converters.

Direct digital controller design (II)



It can be easily shown that, if $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ is the state-space realization of a continuous time system of transfer function $G(s)$ (assuming that $G(s)$ has no delays), the discrete time system having as input u^* and output y^* has the following realization

$$\mathbf{A}^* = \mathrm{e}^{\mathbf{A}T} \qquad \mathbf{B}^* = \int_0^T \mathrm{e}^{\mathbf{A}\sigma}\mathbf{B}\mathrm{d}\sigma \qquad \mathbf{C}^* = \mathbf{C} \qquad \mathbf{D}^* = \mathbf{D}$$

This transformation is called zero-order hold method.

Direct digital controller design (III)

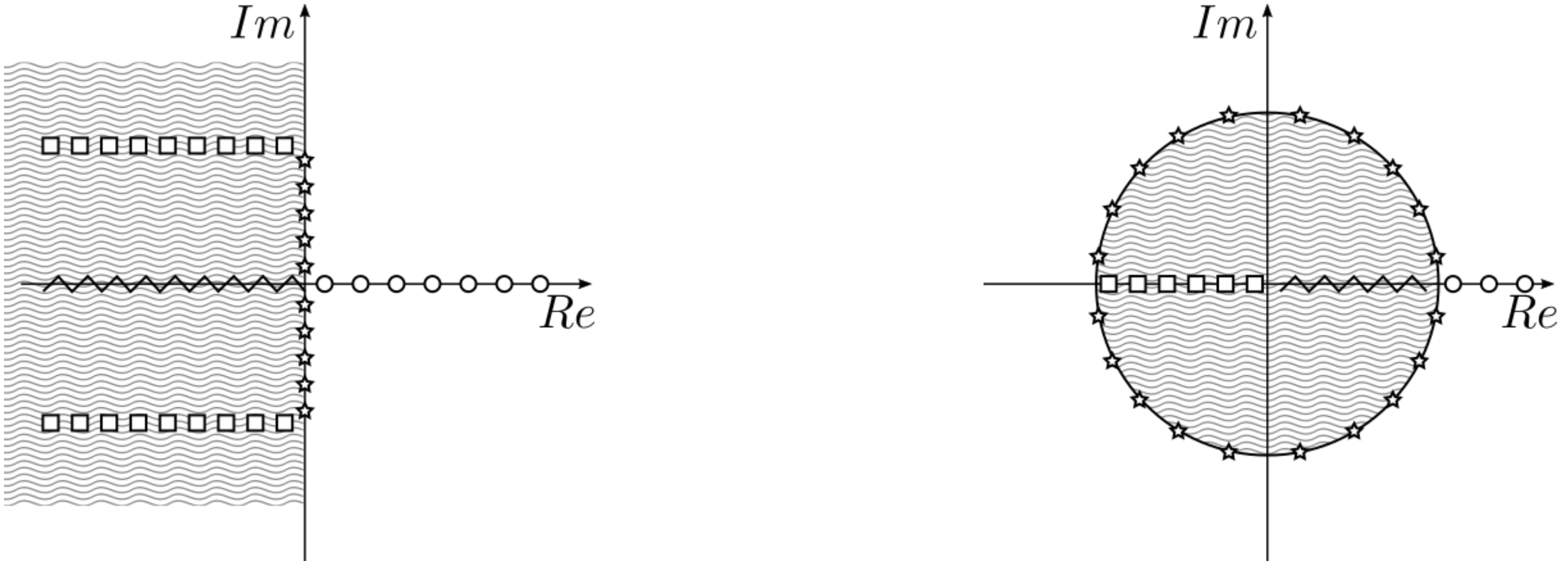
Analyzing the previous result and, in particular

$$\mathbf{A}^* = \mathrm{e}^{\mathbf{A}T}$$

we discover a relation between the eigenvalues of the continuous time and the discrete time system

$$z = \mathrm{e}^{sT}$$

This relation explains how the modes of the continuous time system are transformed into the ones of the discrete time system.



Continuous time | Discrete time