Lecture notes on Digital Control

Part of
Control 4 (ENG4042) / Control M (ENG5022) courses
Digital Control module

Dr Matteo Ceriotti
University of Glasgow

Date of latest revision:
01 October 2021

Reference textbooks:

[1] Franklin, Powell, Workmann, “Digital control of dynamic systems” 3rd Edition, Ellis-Kagle press, 2014. ISBN13: 978-0-9791226-1-3
https://www.elliskaglepress.com/

[2] M. Sami Fadali and Antonio Visioli, “Digital Control Engineering Analysis and Design”, 3rd Edition, Elsevier, 2020. ISBN: 978-0-12-814433-6
https://doi.org/10.1016/C2017-0-01563-X

Foreword

These notes were originally typewritten by an anonymous student, to whom I am grateful.
I have since then revised, corrected and updated these notes, and thank all the students who have provided feedback so far.
I cannot guarantee that these notes are free of typos: I would greatly appreciate if you could report any you find to me, matteo.ceriotti@glasgow.ac.uk.

📖 This symbol references the suggested textbooks and points to further reading.

💻 This symbol provides hints for computer-based design (MATLAB/Simulink).

I hope that these notes will help you in learning Digital Control and preparing for the exam.

Thank you

Matteo Ceriotti

Table of Contents

Exercises

1 Review of Continuous/Analogue Systems and Control

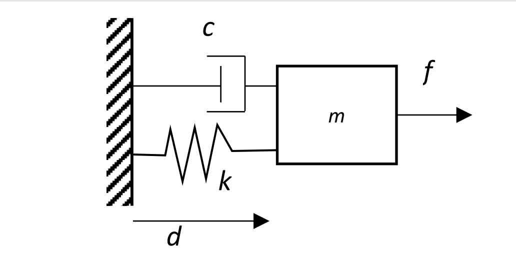
These are dynamical systems where time is the continuous independent variable.
Assume linear system dynamics.
If not → Linearise!
State-Space Form:

$$\begin{cases} \dot{\mathbf{x}} = F\mathbf{x} + G\mathbf{u} \\ \mathbf{y} = U\mathbf{x} + J\mathbf{u} \end{cases} \tag{1.1}$$

$\mathbf{x}$: State vector
$\mathbf{y}$: Output vector (measurements)
$\mathbf{u}$: Input vector (control)

A state-space system is always of the first order, but it is vectorial.
In fact, any ordinary linear differential equation (DE) of order m can be written as a system in the first order, with m equations and m states.

Exercise 1 **Spring-mass-damper**



Find the equations of motion of the system and write them in state-space form, as in Eq. (1.1), assuming the output is the position $y = d$.

Lagrange found a closed form solution:

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0) + \int_0^t e^{\mathbf{F}(t-t_0)}\mathbf{G}u(\tau)dt$$

The matrix exponential is:

$$e^{\mathbf{F}(t-t_0)} \triangleq \mathbf{I} + \mathbf{F}(t-t_0) + \mathbf{F}^2\frac{(t-t_0)^2}{2!} + \ldots + \mathbf{F}^m\frac{(t-t_0)^m}{m!} = \sum_{k=0}^{+\infty}\mathbf{F}^k\frac{(t-t_0)^k}{k!}$$

Note that a second-order system such as the mass-spring-damper in Eq. (1.5) can be generally expressed as:

$$\ddot{r} + 2\zeta\omega_0\dot{r} + \omega_0^{\ 2}r = k_0 u$$

and:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -\omega_0^{\ 2} & -2\zeta\omega_0 \end{bmatrix}\mathbf{x} + \begin{bmatrix} 0 \\ k_0 \end{bmatrix}\mathbf{u}$$

where ω_0 is the natural frequency and ζ is the damping.

1.1 Laplace transformation

By definition:

$$F(s) \triangleq \mathrm{L}\{f(t)\} \triangleq \int_0^{+\infty} f(t)e^{-st}dt$$

$$f(t) = \mathrm{L}^{-1}\{F(s)\} = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}ds$$

It follows that:

$$\mathrm{L}\{\dot{f}(t)\} = sF(s)$$

Derivation in time becomes multiplication by *s*.
Transforming the mass-spring-damper system:

$$\left(s^2 + 2\zeta\omega_0 s + \omega_0^{\,2}\right)Y(s) = k_0 U(s)$$

1.2 Transfer function

The transfer function (TF) is defined as:

$$G(s) \triangleq \frac{\mathrm{L}\{y(t)\}}{\mathrm{L}\{u(t)\}} = \frac{Y(s)}{U(s)}$$

So:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k_0}{s^2 + 2\zeta\omega_0 s + \omega_0^{\,2}}$$

In general, the TF of a linear system is a ratio of two polynomials in *s*:

$$G(s) = \frac{b_1 s^m + b_2 s^{m-1} + \ldots + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \ldots + a_{n+1}}$$

This can also be written as:

$$G(s) = k\frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)}$$

The *m* roots z_i of numerator are called “zeros”. The *n* roots p_i of denominator are called “poles”. *k* is called “gain”.
Transfer functions are function of the complex variable *s* and are said to be defined in the *frequency domain*.

1.2.1 Transfer function from state-space form

To find the transfer function of a system Eq. (1.1) in state-space form, we can transform both equations with Laplace:

$$\begin{cases} s\mathbf{X}(s) = \mathbf{F}\mathbf{X}(s) + \mathbf{G}U(s) \\ Y(s) = \mathbf{H}\mathbf{X}(s) \end{cases}$$

$$\begin{cases} (s\mathbf{I} - \mathbf{F})\mathbf{X}(s) = \mathbf{G}U(s) \\ \mathbf{H}^{-1}Y(s) = \mathbf{X}(s) \end{cases}$$

Substituting to eliminate $\mathbf{X}(s)$:

$$(s\mathbf{I}-\mathbf{F})\mathbf{H}^{-1}Y(s)=\mathbf{G}U(s)$$

$$\frac{Y(s)}{U(s)}=\mathbf{H}(s\mathbf{I}-\mathbf{F})^{-1}\mathbf{G}\equiv G(s) \text{ transfer function}$$

Note that the denominator of $Y(s)/U(s)$ is $\det(s\mathbf{I}-\mathbf{F})$ which is the characteristic polynomial of the transfer function. So, the zeros of characteristic polynomial are poles of the transfer function. Given a transfer function, it is possible to find the corresponding state-space form inverting the procedure.

Exercise 2 State-space form from transfer function

Consider the following transfer function:

$$G(s)=\frac{Y(s)}{U(s)}=\frac{1}{s^2}$$

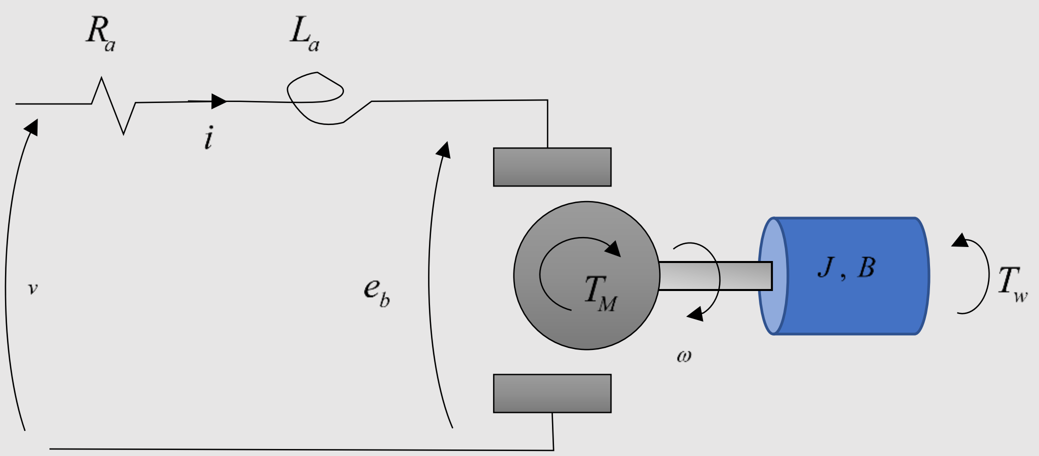
Find the state-space form of the system.

.

Exercise 3 Transfer function of a DC motor

A.

Find the transfer function of the following DC motor, between the applied armature voltage v and the motor's angular velocity ω:



With the following notation:

$v =$ Applied armature voltage;
$e_b =$ Back emf;
i = Current in the armature;
R_a = Armature resistance;
L_a = Armature inductance;
T_M = Torque developed by the motor;
ω = Angular velocity of the motor;
T_w = Disturbance load torque;
J = Moment of inertia of the rotor and attached mechanical load;
B = Viscous friction coefficient of the rotor and attached mechanical load.

Consider the torque generated by a DC motor T_M proportional to the armature current i and the strength of the magnetic field. Assume that the magnetic field is constant and, therefore, that the motor torque is proportional to only the armature current i by a constant factor K_T, called torque constant:

$$T_M = K_T i$$

The back emf, e_b, is proportional to the angular velocity of the shaft by a constant factor K_b

$$e_b = K_b \omega$$

B.

Once you have obtained the TF, find an approximation of it when $L_a \approx 0$.

[3]

1.3 Time Histories

If transfer function of system and input are known, then the output can be computed (in the frequency domain):

$$Y(s) = G(s)U(s)$$

Typical inputs and corresponding transfer functions are:

STEP	$u(t) = R_o 1(t)$	$U(s) = \dfrac{R_o}{s}$
RAMP	$u(t) = V_o t 1(t)$	$U(s) = \dfrac{V_o}{s^2}$
SINE	$u(t) = B_o \sin(\omega_o t) I(t)$	$U(s) = \dfrac{B_o \omega_o}{s^2 + \omega_o^{\ 2}}$

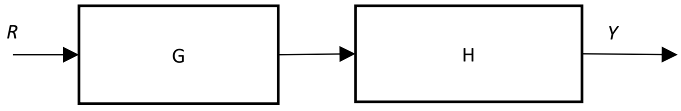
The output can then be expressed as a sum of the fractions, and anti-transformed to get the time history $u(t)$. However, anti-transforming is often difficult.

1.4 Final value theorem

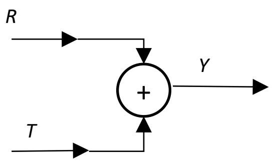
$$\lim_{t \to \infty} x(t) = \lim_{s \to 0^+} sX(s)$$

If system is stable.

1.5 Block Diagrams

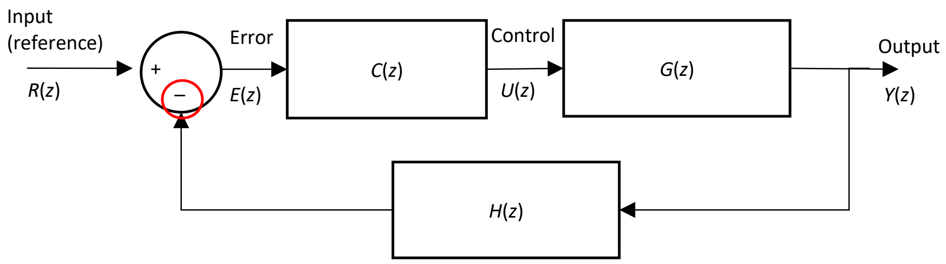


$$Y(s) = GHR(s)$$



$$Y(s) = R(s) + T(s)$$

Feedback loop:



$$\frac{Y(s)}{R(s)} = \frac{CG}{1 + HCG}$$

Note the sign in the denominator is the opposite of that appearing in the feedback loop.

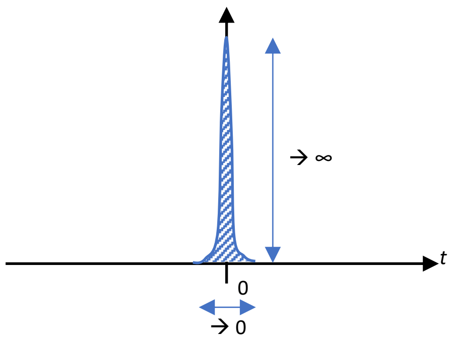
1.6 System response

Impulse:

$$\delta(t) = \begin{cases} +\infty, & \text{at } t = 0 \\ 0, & \text{elsewhere} \end{cases}$$

With:

$$\int_{-\infty}^{+\infty} \delta(t)\, dt = 1$$



Note: $\mathrm{L}\{\delta(t)\} = 1$

So the output $Y(s)$ of a system forced by impulsive input $u(t) = \delta(t)$ is its transfer function:

$$Y(s) = G(s)U(s) = G(s)$$

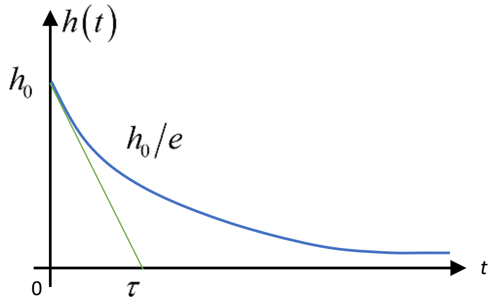
Each pole of the transfer function identifies one response mode of the system.
For a first-order pole:

$$H(s) = \frac{1}{s + \sigma}$$

The impulse response is an exponential.
Anti-transforming:

$$h(t) = e^{-\sigma t} 1(t) \tag{1.6}$$

Where 1(*t*) is the (unit) step function.



$\sigma > 0$ Pole $s < 0$ $\rightarrow$ System is <u>stable</u>
$\sigma < 0$ Pole $s > 0$ $\rightarrow$ System is <u>unstable</u>

Time constant $\tau = \dfrac{1}{\sigma}$

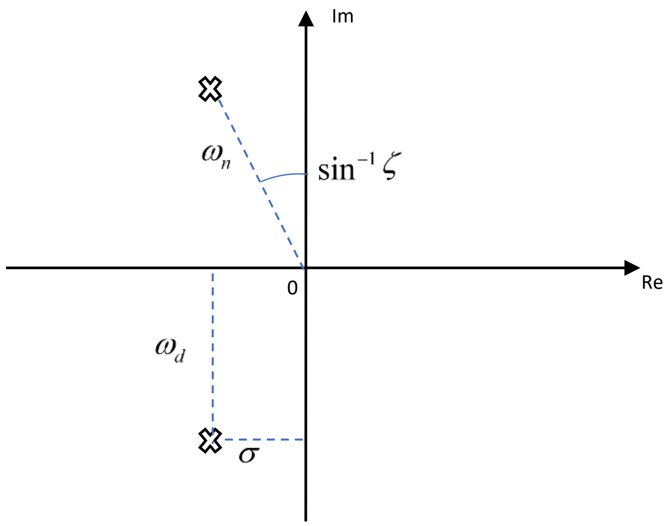
At time τ , response is $\dfrac{1}{e}$ of initial value.

<u>Second-order poles</u> come in complex conjugate pairs as coefficients of polynomial are real.

$$s = -\sigma \pm j\omega_d$$

A second-order transfer function can be written as:

$$H(s) = \frac{\omega_n^{\ 2}}{s^2 + 2\zeta\omega_n s + \omega_n^{\ 2}}$$



Where:

$$\zeta = \frac{\sigma}{\omega_n} \text{ Damping ratio}$$

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} \text{ Undamped natural frequency}$$

$\zeta = 0$ $\rightarrow$ No damping $\rightarrow$ $\omega_n = \omega_d$

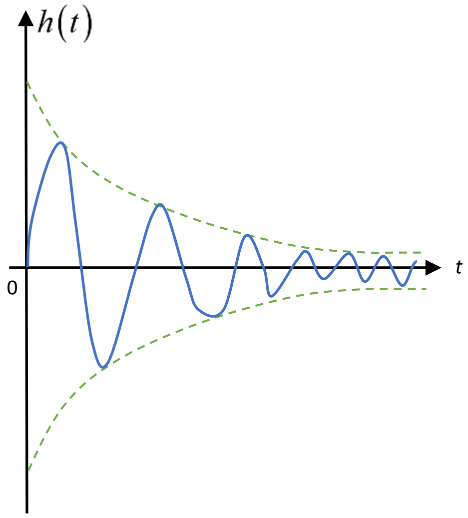
Rewrite the transfer function in the form:

$$H(s) = \frac{\omega_n^2}{(s+\zeta\omega_n)^2 + \omega_n^{\ 2}(1-\zeta^2)}$$

and anti-transform to obtain the impulse response:

$$h(t)=\omega_n e^{-\sigma t}\sin(\omega_d t)I(t) \tag{1.7}$$

which is a damped sinusoid

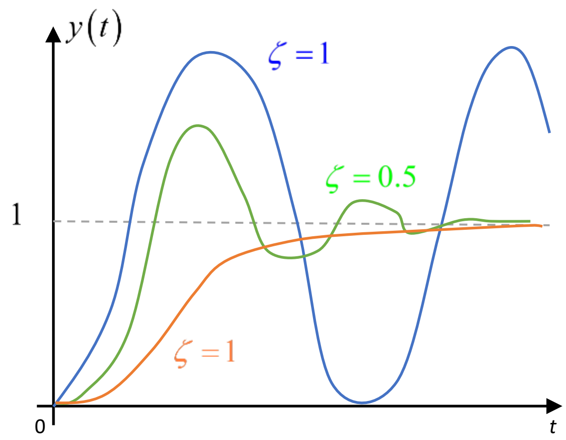


The real part of the pole ($-\sigma$) determines the rate of exponential change, and the imaginary part (ω_d) determines the frequency of oscillations.

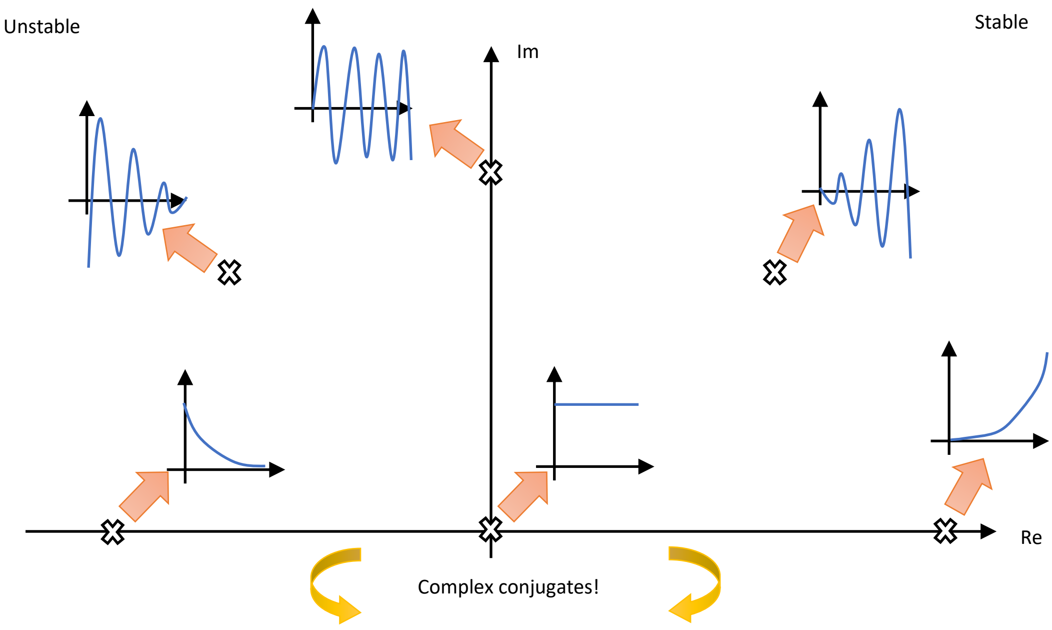
The step response can also be obtained by anti-transforming:

$$Y(s)=H(s)U(s)=\frac{H(s)}{s}$$

$$u(t)=1-e^{-\sigma t}\left(\cos(\omega_d t)+\frac{\sigma}{\omega_d}\sin(\omega_d t)\right) \tag{1.8}$$



Depending on the positions of the poles, the response is different:



[1] 2
[2] 6.2, 5.4

1.7 Design specifications

Design specifications are usually focused on two crucial areas of the response: the *transient*, i.e. the short-time response to a change in the input, and the *steady-state*, i.e. the accuracy of tracking the reference after some time.
The response to a unit step is commonly used to characterise the behaviour of (linear) systems. The transient response to a step input can be quantified with the following parameters.

1. Time constant τ : time required to reach 63% of the final response value. Approximating the initial transient of the second order system with one of the first order with exponential decay, comparing Eqs. (1.6):

$$\tau \approx \frac{1}{\sigma} = \frac{1}{\zeta\omega_n} \tag{1.9}$$

2. Rise time T_r : Time to go from 10% to 90% of the final response value
3. Percentage overshoot P_O : Percentage of the peak response value above the final. Only existing for oscillatory response (i.e. complex conjugate poles).

$$P_O = 100\frac{peak - final}{final} \tag{1.10}$$

4. Peak time T_p : time at the first peak of the response. Only existing for oscillatory response (i.e. complex conjugate poles).
5. Settling time T_s : Time after which the response remains within a given distance (usually 2%) of the final response value.

These values can be calculated or estimated easily for a second order system:

$$H(s) = \frac{\omega_n^{\ 2}}{s^2 + 2\zeta\omega_n s + \omega_n^{\ 2}}$$

$$P_O = 100e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \tag{1.11}$$

$$T_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{1.12}$$

$$T_s \approx \frac{4}{\zeta\omega_n} \text{ (2\%)} \qquad T_s \approx \frac{3}{\zeta\omega_n} \text{ (5\%)} \tag{1.13}$$

We can notice that the damping ratio ζ is related to the oscillatory nature of the system (higher oscillations for low ζ), and the natural frequency ω_n is related to the speed of response (faster response for higher ω_n)

For systems with zeros, the percentage overshoot P_O is higher than predicted.

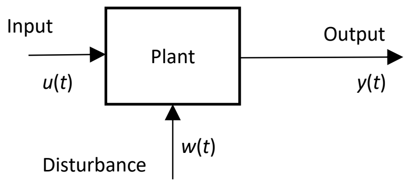
For higher-order systems, these formulas provide approximate values, if the time response is dominated by a single pair of complex conjugate poles (if additional poles and zeros are far in the left half plane or almost cancel).

Based on these and other similar considerations, the design process can be reduced to the positioning of the poles to obtain the desired response in the time domain.

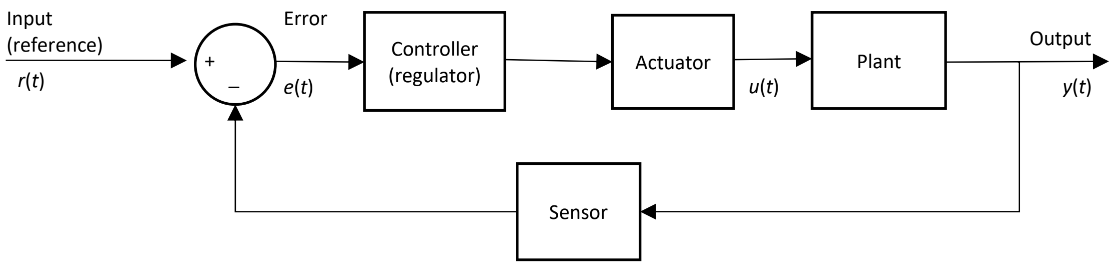
[2] 5.3

2 Introduction to Digital Control

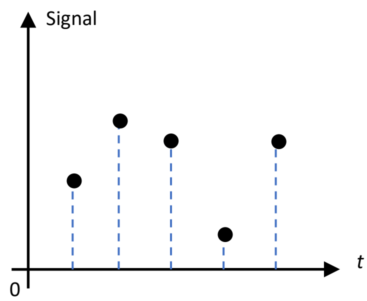
The process to be controlled, described by a set of differential equations (DEs), is called plant. By "control" we mean the process of forcing the plant output $y(t)$ to follow a reference input $v(t)$ by means of an input $\mu(t)$



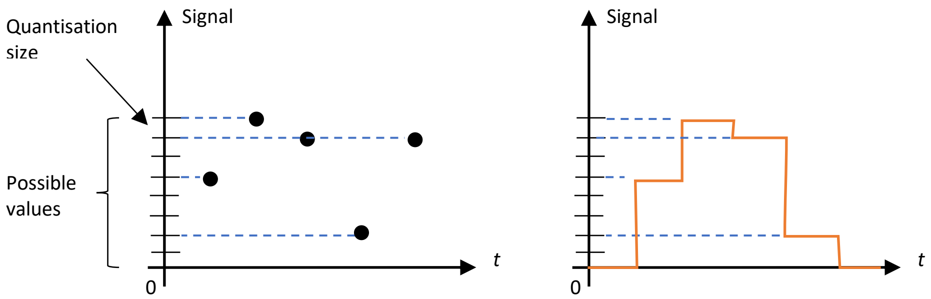
Robust =
Good disturbance rejection +
Low sensitivity to unknown parameters
It was found that a feedback control has many advantages in terms of robustness:



In this course we deal with digital control, i.e The regulator/controller is a digital computer. Discrete sensors are defined at each sample period T
$u(kT), \hat{y}(kT)$, ...

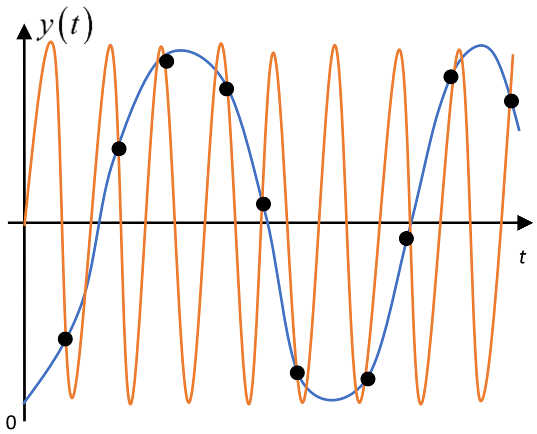


The A/D provides a discrete signal that is also quantized, $\rightarrow$ finite number of digits (or bits).



Digital signal =Discrete + Quantized

Digital control is about taking into account the effects of quantization and discretization in the design of the controller.
If both q and T are small, then digital signals behave like continuous signals, and continuous design methods can be used.
Limitations in cost, power, design do not allow using a powerful computer, hence T and q can be large.
We deal with the effects of discretization only assuming $q = 0$.
Ripple: Oscillations and anomalies that happen between two samples (if T is large)
Aliasing: Ambiguity between continuous and discrete signals. More than one continuous signal result in same sample waves.



[1] 3.1

2.1 Introduction to digital computers and difference equations

Continuous controllers are built using analogue electronic components like resistors, capacitors, op-amp. Today most controllers are implemented in a digital computer or controller (integrated circuit).
Advantages:

- **Flexibility** Computers can be programmed easily
- **Multi-Purpose** One computer can be used for more than one problem
- **Cost** Integrated digital circuits are cheaper than corresponding analogue

Disadvantages:

- **Cannot integrate or differentiate** Differential equations are solved approximately reducing them to algebraic equations with ⊕ and ⊗ operations only: numerical integration

Approximating a continuous transfer function $D(s)$ with difference equations is called emulation.

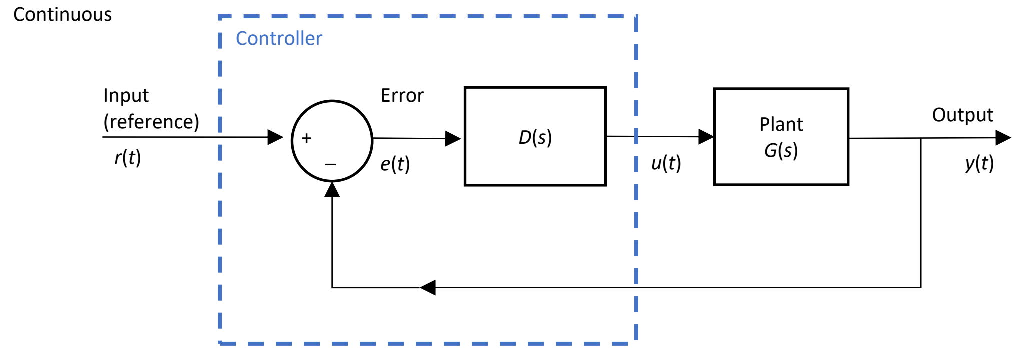


Figure 1. Feedback control system with continuous, analogue control.

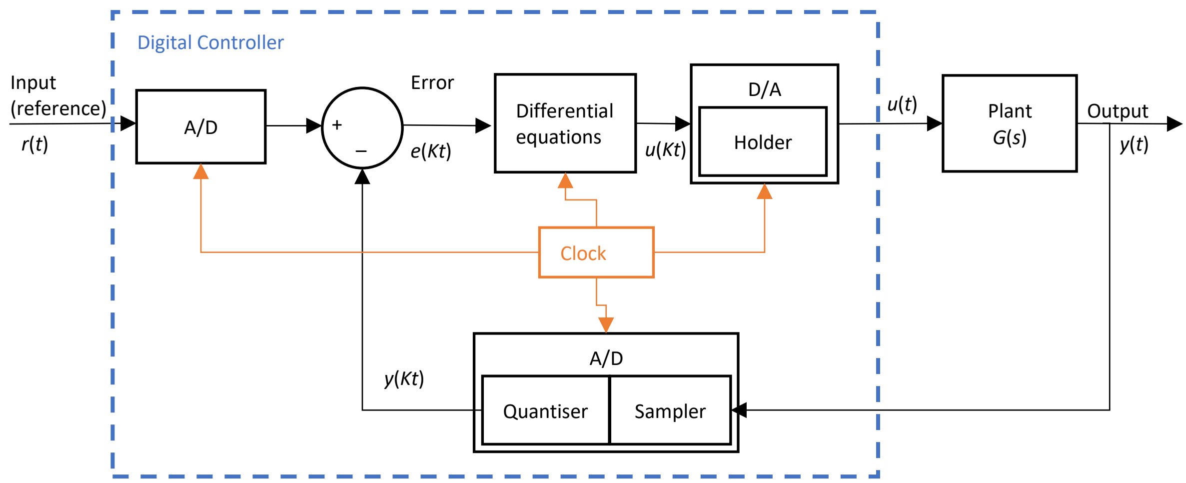
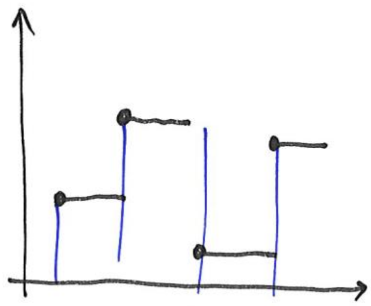


Figure 2. Feedback control system with digital control.

The D/A converts a binary number (quantised and discretised) to an analogue voltage (continuous signal).
The A/D converts a voltage to the closest binary number (adds quantisation and discretisation)
The zero-order hold maintains the same voltage throughout the sample period.



Let's see how the Euler's Method is implemented in a digital computer in practice:

$$\dot{x} \triangleq \lim_{\partial t \to 0} \frac{\partial x}{\partial t}$$

Approximating ∂t to be small:

$$\dot{x}(k) \approx \frac{x(k+1) - x(k)}{T}$$

Or removing the approximation:

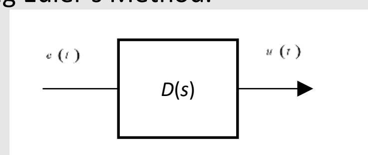
$$\dot{x}(k) = \frac{x(k+1) - x(k)}{T}$$

This is a <u>difference equation</u> (ΔE, to distinguish from differential equation, DE) and it allows to solve the time history of a dynamical system approximately by iterating with time steps of length *T*.
Set initial condition:

$$x(0)$$
$$\downarrow$$
Set $k = 1$
$$\downarrow$$
Read $x(k+1)$ (input)
$$\downarrow$$
Compute $\dot{x}(k)$ (output)
$$\downarrow$$
$$k \leftarrow k + 1$$
$$\downarrow$$
Iterate

Exercise 4 From continuous TF to difference equation with Euler's method

Implement *D*(*s*) in digital form using Euler's Method.



$$D(s) = \frac{U(s)}{E(s)} = K\frac{s+a}{s+b}$$

Hint: write the corresponding DE (in the time domain) and then approximate its derivatives to obtain a ΔE.

University of Glasgow | James Watt School of Engineering

Exercise 5 ΔE of a PID controller

A. Find the difference equation (ΔE) implementing a PID controller, using the Euler forward rule:

$$u(t) = Ke(t) + \frac{K}{T_I}\int_0^k e(t)\,dt + KT_D\dot{e}(t)$$

B. Discuss a potential issue in implementing this controller in a real-world digital computer.

[1] 3.1

2.2 Linear Difference Equations

Equation in previous exercise is a particular case of <u>difference equation</u> (ΔE) or <u>recurrence equation</u>. In general, given the time sequences:

$e_0, e_1, e_2 \ldots e_k$ = Input (Read from sensor)

$u_0, u_1, u_{2\ldots} u_{k-1}$ = Output, forcing term (Computed previously and stored)

The computer can compute:

$$u_k = f\left(e_0, \ldots, e_k, u_0, \ldots, u_{k-1}\right)$$

At each sample period *T*. *k* is an integer sequentially numbering each sample. The time of each sample is *t* = *kT*.
In the case the difference equation represents a controller, the forcing term (output) *u* is the control variable, that feeds into the plant, while *e* is the input, read from a sensor.

We will deal only with linear, constant-coefficient, finite ΔE:

$$\begin{aligned} u_k &= -a_1 u_{k-1} - a_2 u_{k-2} - \ldots - a_m u_{k-m} \\ &+ b_0 e_k + b_1 e_{k-1} + \ldots + b_n e_{k-n} \end{aligned}$$

2.3 Difference equations with no input

We will start studying difference equations which do not have any input (b_i = 0): these are called *autonomous*, and their output is defined by the initial conditions only.

Exercise 6 Numerical solution of difference equation

Solve $u_k = u_{k-1} + u_{k-2}$ for several values of *k*., using initial conditions $u_0 = u_1 = 1$. What sequence do you get?

If response to finite initial conditions in unbounded, the system is <u>unstable</u>.
Question: How can we determine if a DS is stable, without trying to compute a solution for $k = +\infty$ for all possible initial conditions? We need to find a general solution.
We know that for continuous, linear, constant-coefficient, ordinary DEs, a general solution is a linear combination of terms like e^{st}.
For linear constant-coefficient ΔEs, general solutions are a linear combination of term like:

$$u(k) = Az^k$$

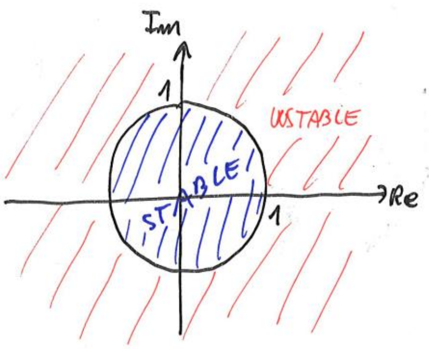
Exercise 7 General solution of Fibonacci difference equation

Find the closed-form solution of the Fibonacci sequence:

$$u_k = u_{k-1} + u_{k-2}$$

Hint: substitute the general solution to find the characteristic equation and solve for z. Create a solution with linear combinations of the terms found above, apply the initial conditions and solve for the coefficients.

Note that in previous example $z_1 > 1$, hence $z_1^k \to \infty$ for $k \to \infty$.
Also note that z_1, z_2 do not depend on the initial conditions, so this result is general for any initial condition.
This result can be generalized: If any of the solutions $|z_i| > 1$, the ΔE is unstable.
z_i can be complex!



The equation found substituting $u_k \to z^k$ is called the characteristic equation, and is a polynomial in z:

$$u_k = u_{k-1} + u_{k-2}$$
$$\updownarrow$$
$$z^k = z^{k-1} + z^{k-2} \text{ Characteristic Equation}$$

The roots of this equation determine the stability of the ΔE. If the magnitude of all roots < 1, ΔE is stable.

Exercise 8 Stability of autonomous difference equation

Determine the stability of $u_k = 0.9u_{k-1} - 0.2u_{k-2}$.

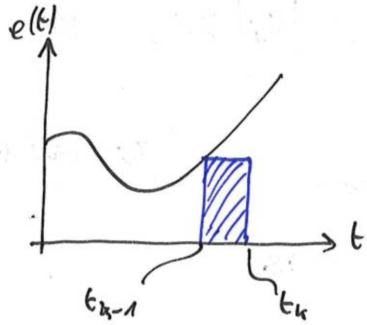
[1] 4.1
[2] 2.2

2.4 Difference equations with input: Numerical Integration

We now move on to difference equations with an input, e. As an example, let us consider the case of numerical integration.
Given a continuous signal $e(t)$, we want to approximate with a ΔE:

$$I = \int_0^t e(t)\,dt$$

Using only the values $e(0),\ e(t_1),\ \dots\ e(t_{k-1}),\ e(t_k)$.



Suppose we know the value of the integral up to t_{k-1} at t_k, we need to add the area between t_{k-1} and t_k.

If we approximate this area with a rectangle, and take the value $e(k-1)$ as height of the rectangle, we have:

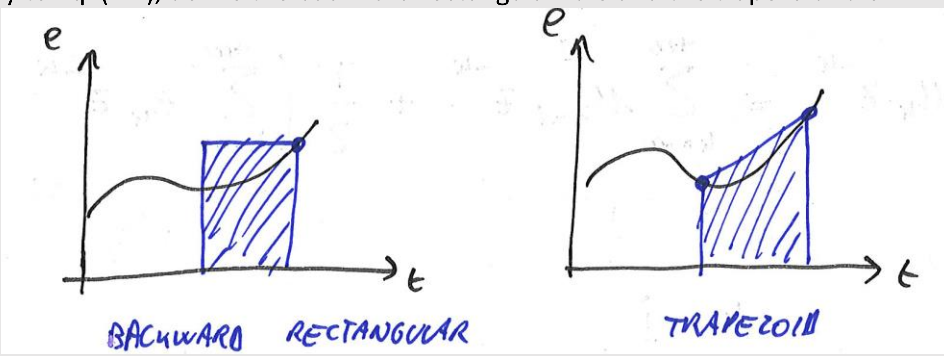
$$u_k = u_{k-1} + (t_k - t_{k-1})e_{k-1} = u_{k-1} + Te_{k-1} \quad \text{(forward rectangular rule)} \tag{2.1}$$

Constant sampling period = *T*

This is the ΔE of the **forward rectangular** rule.

Exercise 9 Backward and trapezoid integration rules

Analogously to Eq. (2.1), derive the backward rectangular rule and the trapezoid rule:



.

Exercise 10 Realizable ΔE of a PID controller

Consider again the PID controller in Exercise 5:

$$u(t) = Ke(t) + \frac{K}{T_I}\int_0^k e(t)\,dt + KT_D\dot{e}(t)$$

Find a realizable (causal) ΔE for it.

[1] 4.1

3 Z-transform and Discrete Transfer Function

3.1 Z-transform

Given a discrete signal e_k we define the **z-transform** as:

$$E(z) \triangleq \mathrm{Z}\{e(k)\} \triangleq \sum_{k=-\infty}^{+\infty} e_k z^{-k} \tag{3.1}$$

This has the same role as the Laplace transform of continuous systems.
So for a time sequence: $e_k = \{e_0, e_1, e_2, \ldots, e_k\}$, its z-transform is:

$$E(z) = e_0 + e_1 z^{-1} + e_2 z^{-2} + \ldots + e_k z^{-k}$$

The variable z^{-m} can be interpreted as a time delay of *m* steps.

Let's consider a ΔE with input and output, such as the trapezoid integration rule from Exercise 9:

$$u_k = u_{k-1} + \frac{T}{2}(e_k + e_{k-1}) \tag{3.2}$$

We multiply by z^{-k} and sum over *k*:

$$\underbrace{\sum_{k=-\infty}^{+\infty} u_k z^{-k}}_{U(z)} = \sum_{k=-\infty}^{+\infty} u_{k-1} z^{-k} + \frac{T}{2}\left(\underbrace{\sum_{k=-\infty}^{+\infty} e_k z^{-k}}_{E(z)} + \sum_{k=-\infty}^{+\infty} e_{k-1} z^{-k}\right)$$

By direct comparison with Eq. (3.1), we can see that the first and third summations are the definition of *U*(*z*) and *E*(*z*) respectively. For the remaining terms, let's consider for example the second summation, $\sum_{k=-\infty}^{+\infty} u_{k-1} z^{-k}$: we can change the variable introducing $j = k-1$:

$$\sum_{k=-\infty}^{+\infty} u_{k-1} z^{-k} \underset{\substack{\uparrow \\ j=k-1}}{=} \sum_{j=-\infty}^{+\infty} u_j z^{-(j+1)} = z^{-1} \sum_{j=-\infty}^{+\infty} u_j z^{-j} = z^{-1} U(z)$$

And analogously for $\sum_{k=-\infty}^{+\infty} e_{k-1} z^{-k}$, to obtain:

$$U(z) = z^{-1} U(z) + \frac{T}{2}\left(E(z) + z^{-1} E(z)\right)$$

This is the frequency-domain version of Eq. (3.2).
Note: A back shift of one sample $[u_{k-1}]$ corresponds to a multiplication by z^{-1} in the *z*-transform!
The z-transform is a linear operation.
The *z*-transform of a linear, constant coefficient ΔE is a polynomial in *z*. So, we can always reduce it as a fraction of two polynomials in *z*:

$$U(z) = \underbrace{\frac{T}{2}\frac{1+z^{-1}}{1-z^{-1}}}_{H(z)} E(z) \tag{3.3}$$

We will discuss z-transforms of basic signals in Sec. 4.1.

[1] 4.2, 4.2.1
[2] 2.3

3.1.1 Shift theorem

The shift theorems allow to easily find the z-transform of a sequence shifted in time, when initial conditions are not zero. Given the z-transform of a signal e_k:

$$Z\{e_k\} \triangleq E(z) \triangleq \sum_{k=-\infty}^{+\infty} e_k z^{-k}$$

Then a positive shift of one sample gives:

$$Z\{e_{k+1}\} = \sum_{k=-\infty}^{+\infty} e_{k+1} z^{-k} = zE(z) - ze_0 \quad \text{(left shift theorem)} \tag{3.4}$$

Two positive shifts in time give:

$$Z\{e_{k+2}\} = z^2 E(z) - z^2 e_0 - ze_1 \tag{3.5}$$

Similarly, one negative shift gives:

$$Z\{e_{k-1}\} = e_{-1} + z^{-1} E(z) \quad \text{(right shift theorem)} \tag{3.6}$$

And for two negative shifts:

$$Z\{e_{k-2}\} = e_{-2} + e_{-1} z^{-1} + z^{-2} E(z) \tag{3.7}$$

For a causal signal, the signal/input at negative time is zero, hence the right shift theorem reduces to:

$$Z\{e_{k-m}\} = z^{-m} E(z) \quad \text{(for causal systems, } e_{-1} = e_{-2} = \ldots = 0\text{)}$$

https://learn.lboro.ac.uk/archive/olmp/olmp_resources/pages/workbooks_1_50_jan2008/Workbook21/21_2_bscs_z_trnsfm_thry.pdf

3.1.2 Multiplication by a^k

Consider the sequence:

$$f_k = a^k e_k$$

Applying the z-transform we get:

$$\begin{aligned} Z\{f_k\} &= f_0 + f_1 z^{-1} + f_2 z^{-2} + \ldots \\ &= e_0 + ae_1 z^{-1} + a^2 e_2 z^{-2} + \ldots \\ &= \sum_{k=0}^{\infty} a^k e_k z^{-k} = \sum_{k=0}^{\infty} e_k \left(\frac{z}{a}\right)^{-k} = E\left(\frac{z}{a}\right) \end{aligned}$$

Hence we have demonstrated the following theorem:

$$Z\{a^k e_k\} = E\left(\frac{z}{a}\right) \tag{3.8}$$

https://learn.lboro.ac.uk/archive/olmp/olmp_resources/pages/workbooks_1_50_jan2008/Workbook21/21_2_bscs_z_trnsfm_thry.pdf

3.1.3 Final value theorem

The final value theorem allows to calculate the value of a sequence as k tends to infinity, starting from the z-transform, under the assumption that this limit exists (i.e. no unbounded or oscillatory sequences).

$$\lim_{N \to +\infty} f(N) = f(\infty)$$

Let's start from the definition of the z-transform, Eq. (3.1), assuming $f(k) = 0, k < 0$:

$$F(z) \triangleq \mathrm{Z}\{f(k)\} \triangleq \sum_{k=0}^{+\infty} f(k) z^{-k} = \lim_{N \to +\infty} \sum_{k=0}^{N} f(k) z^{-k}$$

We now apply a shift of one time step:

$$zF(z) - zf(0) = \lim_{N \to +\infty} \sum_{k=0}^{N} f(k+1) z^{-k}$$

Subtracting the last two equations:

$$(z-1)F(z) - zf(0) = \lim_{N \to +\infty} \sum_{k=0}^{N} f(k+1) z^{-k} - f(k) z^{-k}$$

Taking the limit at z → 1:

$$\lim_{z \to 1}\left[(z-1)F(z) - zf(0)\right] = \lim_{z \to 1} \lim_{N \to +\infty} \sum_{k=0}^{N} \left[f(k+1) z^{-k} - f(k) z^{-k}\right]$$

$$\lim_{z \to 1}\left[(z-1)F(z)\right] - f(0) = \lim_{N \to +\infty} \sum_{k=0}^{N} \left[f(k+1) - f(k)\right]$$

$$\lim_{z \to 1}\left[(z-1)F(z)\right] - f(0) = \lim_{N \to +\infty} \left[\overbrace{-f(0) + f(1)}^{k=0} \overbrace{-f(1) + f(2)}^{k=1} \ldots \overbrace{-f(N) + f(N+1)}^{k=N}\right]$$

$$= -f(0) + \lim_{N \to +\infty} f(N+1)$$

$$= -f(0) + f(\infty)$$

$$\lim_{z \to 1}\left[(z-1)F(z)\right] = f(\infty) \tag{3.9}$$

Which is also sometimes expressed as (equivalent in the limit *z* → 1):

$$\lim_{z \to 1}\left[\left(1 - z^{-1}\right)F(z)\right] = f(\infty) \tag{3.10}$$

📖 [1] 4.6.1
📖 [2] 2.3.4

3.2 Discrete transfer function

We can define the **discrete transfer function** as the ratio between output and input [z-transformed]:

$$H(z) \triangleq \frac{U(z)}{E(z)} \tag{3.11}$$

For the specific case of the trapezoid rule, the transfer function is:

$$H(z) = \frac{U(z)}{E(z)} = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \leftarrow \text{TF of trapezoid rule}$$

Or in general for a discrete linear system:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \ldots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} \ldots + a_n z^{-n}} = \frac{b(z)}{a(z)} \text{ Polynomials}$$

$$z_i \mid b(z_i) = 0 \rightarrow \text{Zeros}$$

$$p_i \mid a(p_i) = 0 \rightarrow \text{Poles}$$

TF can be factored as:

$$H(z) = K\frac{\prod_{i=1}^{m}(z-z_i)}{\prod_{i=1}^{n}(z-p_i)}$$

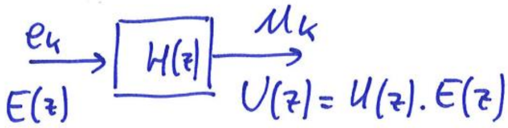
For understanding the meaning of the discrete transfer function, consider:

$$u(k) = e_{k-1} \rightarrow U(z) = z^{-1}E(z)$$

z^{-1} in the transfer function corresponds to a delay of a time sample.

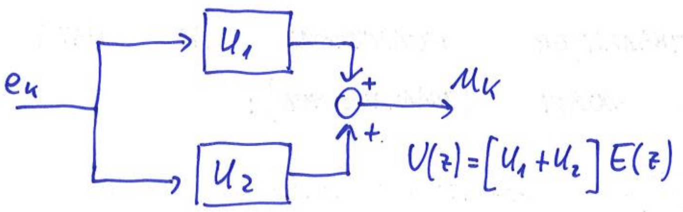
[1] 4.2, 4.2.2
[2] 2.6.2

3.3 Block Diagrams and linearity



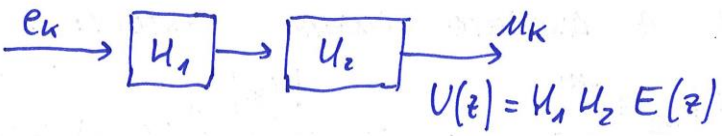
$$U(z) = HE(z)$$

Parallel connection:



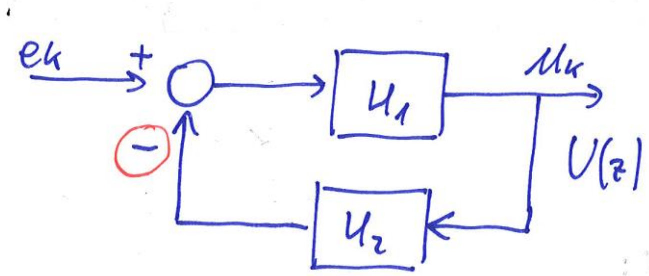
$$U(z) = (H_1 + H_2)E(z)$$

Serial connection:



$$U(z) = H_1H_2E(z)$$

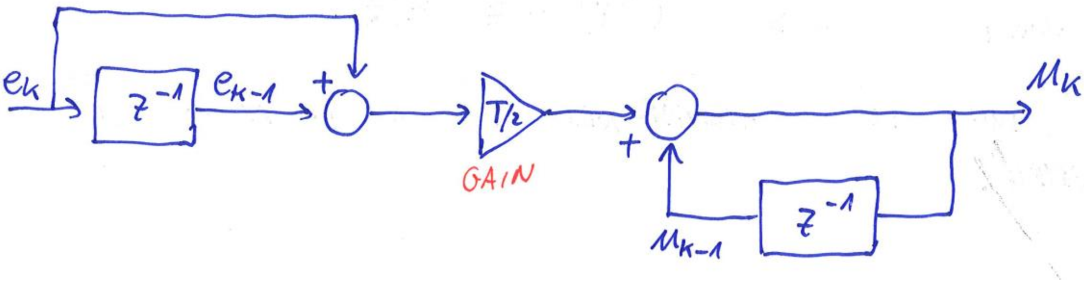
Feedback loop:



$$U(z) = \frac{H_1}{1+H_1H_2}E(z)$$

Any linear transfer function can be thought as a linear combination of z^{-1} blocks. For example, for the trapezoid integration rule:

$$u_k = u_{k-1} + \frac{T}{2}\left(e_k + e_{k-1}\right)$$



Exercise 11 Difference equation in Simulink

Use a computer tool (Simulink) to create a discrete model of the following difference equation, using basic building blocks such as: delay z^{-1} block, gain and sum:

$$u(k) = au(k-1) + bu(k-2), \quad k = 3,4,5...$$

$$\begin{cases} u(1) = u_1 \\ u(2) = u_2 \end{cases}$$

Use this model to find the first 10 points of the Fibonacci difference equation.

.

Exercise 12 Integrator models in Simulink

Use a computer tool (Simulink) to create a discrete models encoding the three integrators (forward, backward and rectangular) using basic building blocks such as: delay z^{-1} blocks, gains and sums.
Use these models to integrate a discrete sine wave (of unit frequency and amplitude for simplicity) and compare the results.

📖 [1] 4.2.3
📖 [2] 2.3.2.1

3.4 Inversion of the z-transform

The inverted z-transform can be used to reconstruct the time domain of a signal whose z-transform is known. A complex integral can be used, but this very rarely can be solved in closed form, and it is rarely useful in real applications.
A more informative approach is that of trying to reduce the expression in the z-domain to a sum of parts that can be easily anti-transformed using tables, such as Table 1 (thanks to the linearity property).

Table 1. Some z-transforms. It is assumed f(t) = 0 for t < 0.

	Continuous frequency-domain (s-domain) $F(s)$	**Discrete time domain** $f(kT)$	**Discrete frequency-domain (z-domain)** $F(z)$
Impulse	-	$\delta(k) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$	1

Time-shifted impulse	-	$\delta(k-m)=\begin{cases}1, & k=m\\0, & k\neq m\end{cases}$	z^{-m}
Step	$\frac{1}{s}$	$1(kT)$	$\frac{z}{z-1}$
Ramp	$\frac{1}{s^2}$	kT	$\frac{Tz}{(z-1)^2}$
Parabola	$\frac{2!}{s^3}$	$(kT)^2$	$T^2\frac{z(z+1)}{(z-1)^3}$
Exponential	$\frac{1}{s+a}$	e^{-akT}	$\frac{z}{z-e^{-aT}}$
	$\frac{1}{(s+a)^2}$	kTe^{-akT}	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
	$\frac{a}{s(s+a)}$	$1-e^{-akT}$	$\frac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$

The following sections show two techniques that can be handy to reduce an expression in the z-domain to a sum of elementary fractions.

📖 [2] 2.3.3, A-I

3.4.1

We can use the “long division” algorithm to express any fraction in *z* as an expansion of decreasing powers of *z*, up to a certain order:

$$F(z)=f_0+f_1z^{-1}+f_2z^{-2}+\ldots+f_iz^{-i}+\ldots$$

Which implies a time sequence of *i* elements:

$$\{f_0, f_1, f_2, \ldots, f_i\}$$

Exercise 13 Long division

Solve the following fraction:

$$F(z)=\frac{6z^2+7z-20}{2z+5}$$

Using the long division algorithm.

.

Exercise 14 Long division

Find the first 3 elements of the time sequence f_k corresponding to the following function:

$$F(z)=\frac{2z+5}{6z^2+7z-20}$$

Using the long division algorithm.

.

3.4.2

If the function has distinct real roots on the denominator, then the partial expansion has the form:

$$F(z)=\sum_{i=1}^{n}\frac{A_i}{z-z_i} \tag{3.12}$$

Where z_i are the roots of the denominator, and A_i are coefficients to be determined.

Exercise 15 Partial fraction expansion

Find the time domain expression f_k of:

$$F(z)=\frac{7z^2-23z}{z^2-7z+10}$$

Using partial fraction expansion.

.

If the function has roots with multiplicity $r > 1$ on the denominator, then it can be expanded as:

$$F(z)=\sum_{i=1}^{r}\frac{A_i}{(z-z_i)^{r+1-i}}+\sum_{j=r+1}^{n}\frac{B_j}{z-z_j} \tag{3.13}$$

Because most of the basic z-transforms in the tables have a *z* in the numerator, it is sometimes easier to obtain the partial fraction expansion of $\frac{F(z)}{z}$ and then multiply by *z*.

Exercise 16 Partial fraction expansion

Find the time domain expression f_k of:

$$F(z)=\frac{-z^2+2z+18}{(z-3)^2}$$

Using partial fraction expansion.

.

Similar expansions exist for complex conjugate roots, and a combination of real and complex roots.

[1] 4.6.1
[2] 2.3.3.2

3.5 Using z-transform to solve difference equations

The z-transform and its inverse operation can be used to solve difference equations analytically in some cases. The procedure is to start from the difference equation, transform it into the z domain, substitute the initial conditions, solve the algebraic equation for the complex variable, and then anti-transform.

Exercise 17 Solution of Fibonacci sequence with z-transform

Using the z-transform, solve the following ΔE (Fibonacci sequence):

$$y_{k+2}=y_{k+1}+y_k$$
$$\begin{cases} y_0=1 \\ y_1=1 \end{cases}$$

Using partial fraction expansion.

.

Exercise 18 Solution of second-order difference equation with z-transform

Using the z-transform, solve (assume *T*=1):

$$y_k - 6y_{k-1} + 9y_{k-2} = 0$$
$$\begin{cases} y_{-1} = 1 \\ y_{-2} = 0 \end{cases}$$

.

[2] 2.5

https://learn.lboro.ac.uk/archive/olmp/olmp_resources/pages/workbooks_1_50_jan2008/Workbook21/21_3_z_trnsfm_n_difrnce_eqn.pdf

3.6 Pulse response and transfer function

Consider a unit pulse (or impulse) signal, defined as:

$$e_k = \delta_k \triangleq \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$$

The z-transform of the impulse signal is unity, in fact:

$$E(z) = \mathrm{Z}\{\delta_k\} = \sum_{k=-\infty}^{+\infty} \delta_k z^{-k} = 1$$

Now suppose we feed a transfer function H with an impulse input. The z-transformed output, thanks to Eq. (3.11), is:

$$U(z) = E(z)H(z) = H(z) \leftarrow \text{for impulse input} \qquad (3.14)$$

So the transfer function can also be seen as the transform of the response to a unit-pulse input.

[1] 4.2.4

[2] 2.6.1

3.7 Convolution

$$\begin{aligned} U(z) &= E(z)H(z) \\ &= \left[e_0 + e_1 z^{-1} + e_2 z^{-2} + e_3 z^{-3} + \ldots\right]\left[h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + \ldots\right] \\ &= e_0 h_0 + \left(h_0 e_1 + h_1 e_0\right) z^{-1} + \left(e_0 h_2 z^{-2} + e_1 h_1 z^{-2} + h_0 e_2 z^{-2}\right) + \ldots \end{aligned}$$

This must be equal to $U(z)$:

$$U(z) = u_0 + u_1 z^{-1} + u_2 z^{-2} + u_3 z^{-3} + \ldots$$

Equating each coefficient of z:

$$\begin{aligned} u_0 &= e_0 h_0 \\ u_1 &= h_0 e_1 + h_1 e_0 \\ u_2 &= e_0 h_2 + e_1 h_1 + h_0 e_2 \\ &\ldots \end{aligned}$$

This can be written as:

$$u_k = \sum_{j=0}^{k} e_j h_{k-j}$$

Extending to ∞:

$$u_k = \sum_{j=-\infty}^{+\infty} e_j h_{k-j} \qquad \text{Discrete convolution} \tag{3.15}$$

Note: Negative values of j correspond to inputs applied before time = 0. Assuming that $e_j = 0$ for $j < 0$ (Input starts at time 0), values of $k < j$ occur if the pulse response is non-zero at negative times.
It means that the system is "responding" to any impulse that only happens in the future!
These types of systems are called <u>non-causal</u> and cannot be practically realized.

In principle, the convolution (3.15) can be used to obtain the response of a system to an input. This is however rarely useful in practice because the time response of the transfer function is not known, and it involves an infinite summation.

📖 [1] 4.2.4
📖 [2] 2.6.1, 2.6.2

4 Signal Analysis and Dynamic Response

A method to study linear constant discrete systems could be:

1. Compute discrete transfer function $H(z)$;
2. Compute z-transform of input $E(z)$;
3. Compute output z-transform $U(z)=H(z)E(z)$;
4. Invert $U(z)$ to find $u(kT)$.

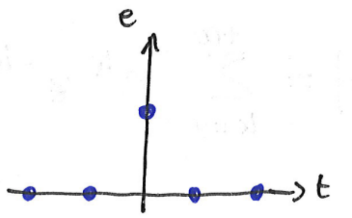
Step 4 can be tricky. We can guess features in the time domain of the output by studying transforms of elementary signals.

📖 [1] 4.4
📖 [2] 2.6.2

4.1 Z-transforms of basic signals

4.1.1 Unit pulse

$$e_k=\delta(k)=\begin{cases}1 & for\ \ k=0\\ 0 & for\ \ k\neq 0\end{cases}$$

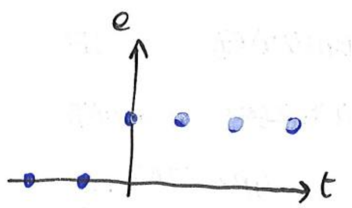


$$E(z)=\sum_{-\infty}^{+\infty}\delta_k z^{-k}=z^0=1$$

📖 [1] 4.4.1
📖 [2] 2.3.1

4.1.2 Unit Step

$$e_k=1(k)=\begin{cases}1 & for\ \ k\geq 0\\ 0 & for\ \ k<0\end{cases}$$

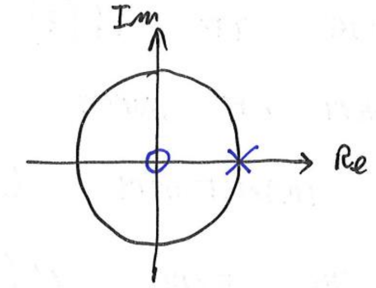


$$E(z)=\sum_{-\infty}^{+\infty}e_k z^{-k}=\sum_{k=0}^{+\infty}z^{-k}=$$

$$=\frac{1}{1-z^{-1}}\quad\left(s.t.\left|z^{-1}\right|<1\right)$$ (geometric series)

$$= \frac{z}{z-1} \quad (s.t. |z| > 1) \tag{4.1}$$

This was a zero at $z = 0$ and pole at $z = 1$. The continuous step has transform $\frac{1}{s}$ so note that a pole at $s = 0$ maps into a pole at $z = 1$ in discrete environment.



[1] 4.4.2

Exercise 19 Z-transform of ramp

Derive the z-transform of the unit ramp (see Table 1):

$$e_k = kT\,1(k)$$

.

Exercise 20 Difference equation response to ramp

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

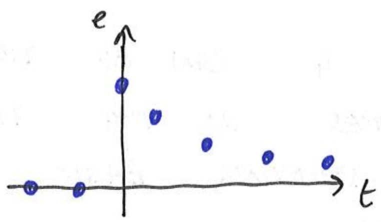
Find the response to a discrete unit ramp ($u_k = k\,T\,1(k)$) using the z-transform (follow the process described at the beginning of Chapter 4.

Hint: z-transform, substitute z-transform of input, solve, invert with the anti-z-transform.

.

4.1.3 Exponential

$$e_k = \begin{cases} r^k, & \text{for } k \geq 0 \\ 0, & \text{for } k < 0 \end{cases}$$
$$= r^k 1(k)$$

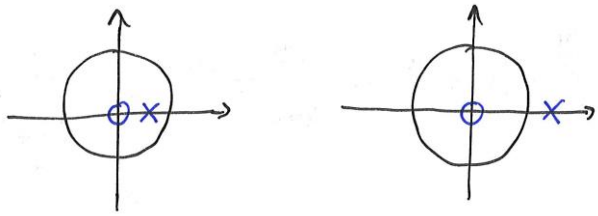


$$E(z) = \sum_{-\infty}^{+\infty} r^k z^{-k} = \Sigma\left(rz^{-1}\right)^k$$

$$= \frac{1}{1 - rz^{-1}} \quad \left(s.t. \left|rz^{-1}\right| < 1\right) \text{ (geometric series)}$$

$$= \frac{z}{z-r} \quad (s.t. |z| > |r|)$$

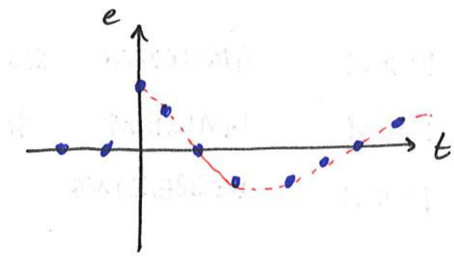
Pole is at $z = r$. The signal e_k grows unbounded if $|r| > 1$ hence if the transform converges and has a pole $|z| > 1$ the signal is growing.



[1] 4.4.3

4.1.4 Sinusoid

$$e_k = r^k \cos(k\theta) 1(k) = $$
$$= r^k \frac{e^{jk\theta} + e^{-jk\theta}}{2} 1(k)$$



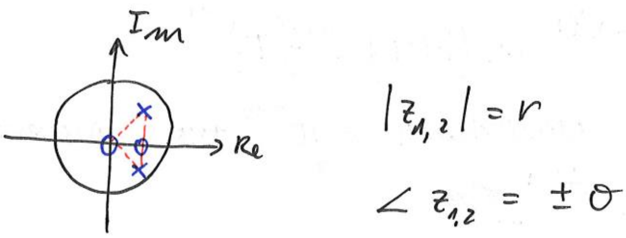
Taking the z-transform:

$$E(z) = \frac{1}{2}\left[\sum_0^{+\infty} r^k e^{j\theta k} z^{-k} + \sum_0^{+\infty} r^k e^{-j\theta k} z^{-k}\right]$$
$$= \frac{1}{2}\left[\sum \left(re^{j\theta} z^{-1}\right)^k + \sum \left(re^{-j\theta} z^{-1}\right)^k\right]$$

These are two geometric series with radius $\left(re^{\pm j\theta} z^{-1}\right)$, hence:

$$= \frac{1}{2}\left[\frac{1}{1 - re^{j\theta} z^{-1}} + \frac{1}{1 - re^{-j\theta} z^{-1}}\right] = \qquad \left(s.t. \; |z| > r\right)$$
$$= \frac{1}{2}\left[\frac{z}{z - re^{j\theta}} + \frac{z}{z - re^{-j\theta}}\right] =$$
$$= \frac{1}{2}\left[\frac{z\left(z - re^{-j\theta} + z - re^{j\theta}\right)}{\left(z - re^{j\theta}\right)\left(z - re^{-j\theta}\right)}\right] =$$
$$= \frac{1}{2}\left[\frac{2z\left(z - r\cos\theta\right)}{\left(z^2 - 2r\cos\theta z + r^2\right)}\right] = \frac{z\left(z - r\cos\theta\right)}{\left(z^2 - 2r\cos\theta z + r^2\right)} \qquad (4.2)$$

The poles are $z_{1,2} = r\left(\cos\theta + j\sin\theta\right) = re^{\pm j\theta}$

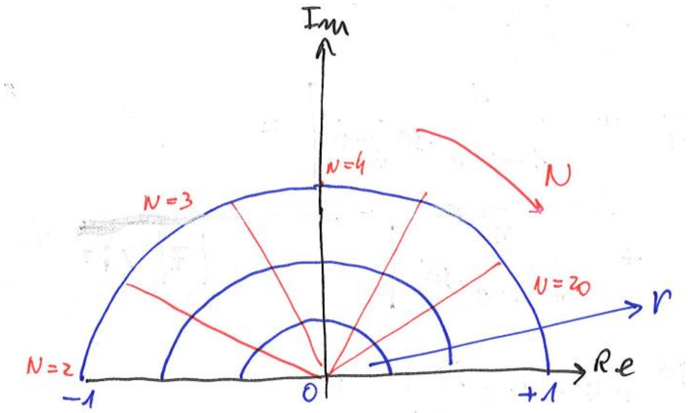


Settling time:

$r>1$ Growing signal
$r=1$ Constant amplitude
$r<1$ Decaying signal

Frequency:

θ determines the frequency of oscillation as $N=\frac{2\pi}{\theta}[rad]$ samples/cycle.



Let us now compare this with a continuous sinusoid:

$$y(t)=e^{-\sigma t}\cos(\omega t)1(t)$$

If we take samples:

$$y(kT)=e^{-\sigma kt}\cos(\omega kT)1(kT)$$

Which is equivalent to the discrete sinusoid as long as:

$$r=e^{-\sigma T}$$

$$\theta=\omega T$$

The poles of the continuous sinusoid are:

$$s_{1,2}=-\sigma\pm j\omega$$

While for the discrete one:

$$z_{1,2}=re^{\pm j\theta}=e^{-\sigma T}e^{\pm j\omega T}=e^{(-\sigma\pm j\omega)T}=e^{s_{1,2}T}$$

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

4.2 z-plane to s-plane correspondence

This relation maps the poles of discrete and continuous signal with same time features. It can be used to transfer the poles from continuous (s-plane) to discrete (z-plane) domain. We can generalise to a function:

If the Laplace transform *F*(*s*) of a continuous-time function *f*(*t*) has a pole p_s, then the z-transform *F*(*z*) of its discretised counterpart *f*(*kT*) has a pole in:

$$p_z=e^{p_sT} \tag{4.3}$$

This relationship in general does not apply to the zeros.

Note the correspondence is one-to-many as:

$$s_2=s_1+j\frac{2\pi}{T}N,\qquad \forall N\in\mathbb{Z}$$

So: $e^{s_1T}=e^{s_2T}$

Remind, for a pole in the continuous domain:

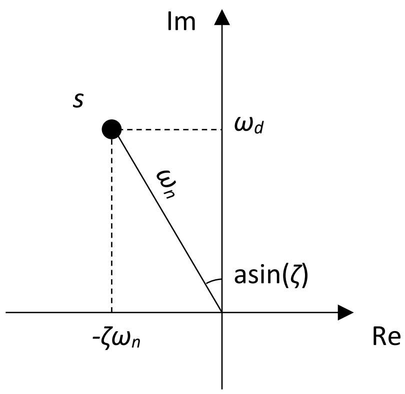
$$s=-\sigma\pm j\omega_d$$

$$\sigma = \zeta\omega_n$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2} \qquad \text{(damped frequency)} \tag{4.4}$$

ω_n = Natural frequency (rad/s)

ζ = (Greek zeta) = Damping ratio



Then according to Eq. (4.3), the corresponding discrete poles are:

$$p_z = e^{-\sigma T}e^{j\omega_d T} = e^{-\sigma T}e^{j(\omega_d T + 2\pi N)}, \quad N \in \mathbb{Z} \tag{4.5}$$

So the pole locations in the z-domain are periodic function of the damped frequency ω_d, with period $\frac{2\pi}{T} = \omega_s$, i.e. the sampling frequency.

In the z-domain, the characteristic equation for second-order (complex-conjugate) poles is:

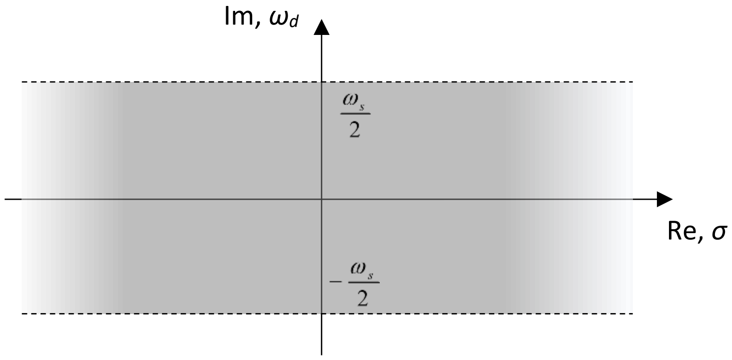
$$\left(z - e^{(-\sigma + j\omega_d)T}\right)\left(z - e^{(-\sigma - j\omega_d)T}\right) = z^2 - 2\cos\left(\omega_d T\right)e^{-\sigma T}z + e^{-2\sigma T} \tag{4.6}$$

Which is essentially another way of rewriting the denominator of Eq. (4.2). By comparison, the magnitude and argument of the poles are:

$$\begin{aligned} r &= \left|z_{1,2}\right| = e^{-\sigma T} = e^{-\zeta\omega_n T} \\ \theta &= \angle z_{1,2} = \pm\omega_d T \end{aligned} \tag{4.7}$$

The implication of this is that the continuous-discrete correspondence is one to many: a single pole in the z-domain maps to infinite poles in the s-domain. Equivalently, infinitely many functions in the s-domain correspond to a single function in the z-domain.

In order to remove this ambiguity, we can limit the poles to a strip in the s-plane, called the *primary strip*, which limits the frequencies (imaginary axis of the poles) to $\left[-\frac{\omega_s}{2}, \frac{\omega_s}{2}\right]$:

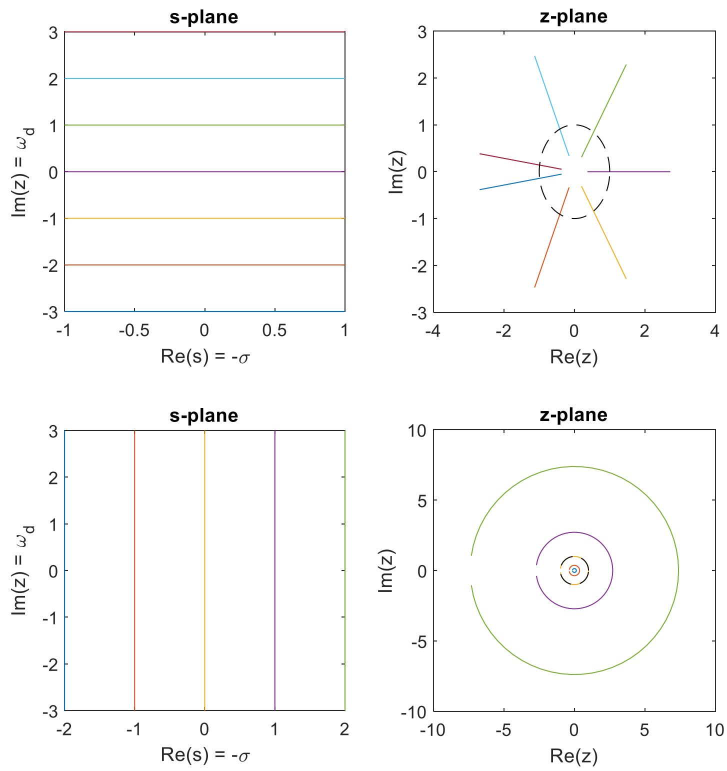


The width of the strip is related to the sampling frequency, and therefore it can be expanded by using a larger sampling frequency (smaller sample time *T*). This is usually constrained by availability of hardware and/or costs.

We can continue the investigation into the mapping of continuous and discrete poles by studying Eq. (4.3). For example, the imaginary axis in the s-plane maps into the unit circle in the z-plane (stability boundary).
The same relation allows to map lines of constant damping ζ and constant natural frequency ω_n

Other correspondences are in the following table and figures:

s-plane	z-plane
Imaginary axis (stability boundary)	Unit circle (stability boundary)
Negative real axis	[0, 1]
Lines of constant ω_d ("horizontal")	Radial lines
Lines of constant σ ("vertical")	Circumferences



We can finally draw contours of constant natural frequency ω_n and damping coefficient ζ in the z-plane:

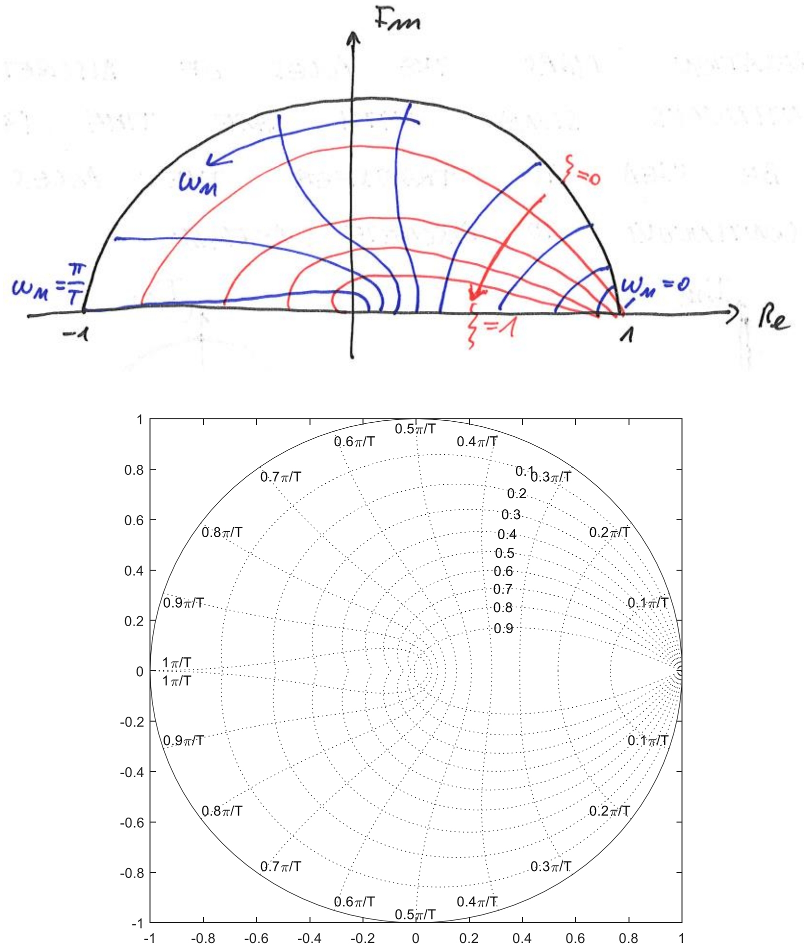


Figure 3. Contours of constant frequency and damping in the z-plane

The analysis applies to signals and also to impulse response of discrete systems! These contours will become useful for a visual assessment of the features of a function, based on its poles in the z-domain, be it a signal or a control loop (see following Sec. 7.2).

📖 [1] 4.4.4, 4.4.5
📖 [2] 6.2, 6.2.1

💻 MATLAB functions: zgrid, sgrid, pzmap

4.3 Frequency response

Remind that continuous linear systems excited with a sinusoid respond, after the transient, with a sinusoid of the same frequency. In addition, their transfer function is:

$$H(s) = H(j\omega) = A(\omega)e^{j\varphi(\omega)}$$

Then the steady-state response to a unit sinusoid of frequency ω is a sinusoid of amplitude $A(\omega)$ and phase $\varphi(\omega)$, both of which depend on the frequency.

A linear discrete system has a similar behaviour. At steady state (after the transient), the response to a signal of frequency ω depends on the phase and amplitude of the transfer function, evaluated at $z = e^{j\omega T}$:

$$H\left(e^{j\omega T}\right)=A(\omega T)e^{j\varphi(\omega T)}$$

Note that we evaluate the transfer function on the unit circle ($z=e^{j\omega T}$) instead of the imaginary axis ($s=j\omega$)!

When excited with a sine of unit amplitude and frequency ω , the steady-state response (after transient) is a sine of same frequency ω , amplitude $A(\omega T)$, and phase shift $\varphi(\omega T)$

$$A=\left|H\left(e^{j\omega T}\right)\right|$$

$$\varphi=\angle H\left(e^{j\omega T}\right)$$

Exercise 21 Steady-state response to sinusoidal input

Determine the steady-state response of:

$$H=\frac{1}{(z-0.2)(z-0.9)}$$

Due to the input:

$$u_k=5\cos(0.3k+0.1)$$

.

📖 [1] 4.5
📖 [2] 2.8

4.3.1 DC gain

DC gain is the gain of the system at steady state for a step input. A step input is essentially a sinusoid where its frequency tends to zero, therefore:

$$\left.H\left(e^{j\omega T}\right)\right|_{\omega\to 0}=H(1)$$

📖 [2] 2.8.1

4.3.2 Periodicity and aliasing

Note: Consider a discrete sinusoid of frequency ω . Now, all other discrete sinusoids of frequency $\omega+m\frac{2\pi}{T},\ m\in\mathbb{Z}$ are essentially the same in the discrete domain, in fact:

$$\begin{aligned}e^{j\left(\omega+m\frac{2\pi}{T}\right)T}&=e^{j\omega T}e^{jm2\pi}\\&=e^{j\omega T}\left(\cos(m2\pi)+j\sin(m2\pi)\right)\\&=e^{j\omega T}\end{aligned}$$

And hence:

$$H\left(e^{j\omega T}\right)\equiv H\left(e^{j\left(\omega+m\frac{2\pi}{T}\right)T}\right),\qquad m\in\mathbb{Z}$$

This is equivalent to say that all sinusoids of frequency $\omega+m\frac{2\pi}{T},\ m\in\mathbb{Z}$ pass through the same sample points, and therefore are indistinguishable in the discrete domain, and therefore they are seen as the same signal by the system, and therefore they generate the same response! This phenomenon is called **aliasing**.

📖 [2] 2.8.1

4.3.3 Symmetry

Consider a transfer function with real coefficients (essentially all real-world scenarios). We have:

$$\overline{H\left(e^{j\omega T}\right)} = H\left(\overline{e^{j\omega T}}\right)$$

In addition, for negative frequencies:

$$H\left(e^{j(-\omega)T}\right) = H\left(\overline{e^{j\omega T}}\right)$$

And therefore:

$$\overline{H\left(e^{j\omega T}\right)} = H\left(e^{j(-\omega)T}\right)$$

Which means that the transfer function is even in magnitude, and odd in phase:

$$\left|H\left(e^{j(-\omega)T}\right)\right| = \left|H\left(e^{j\omega T}\right)\right|$$

$$\angle H\left(e^{j(-\omega)T}\right) = -\angle H\left(e^{j\omega T}\right)$$

These symmetry properties, combined with the periodicity discussed in Sec. 4.3.2, imply that we only need to obtain the frequency response in the range $\omega \in \left[0, \frac{\omega_s}{2}\right]$ where $\omega_s = \frac{2\pi}{T}$ is the sampling frequency. The frequency response for negative frequencies is obtained by symmetry, and for frequencies above $\frac{\omega_s}{2}$ the frequency response is periodically repeated. This will be shown in more detail in Sec. 8.2 and 8.3. For an idea, see Figure 6 and Figure 7.

5 Modelling of Systems with Digital Control

Since most dynamic systems (plants) to be controlled are continuous, we need to describe the dynamics at the interface between continuous and discrete systems.
This is implemented in the D/A (DAC) and A/D (ACD) converters.

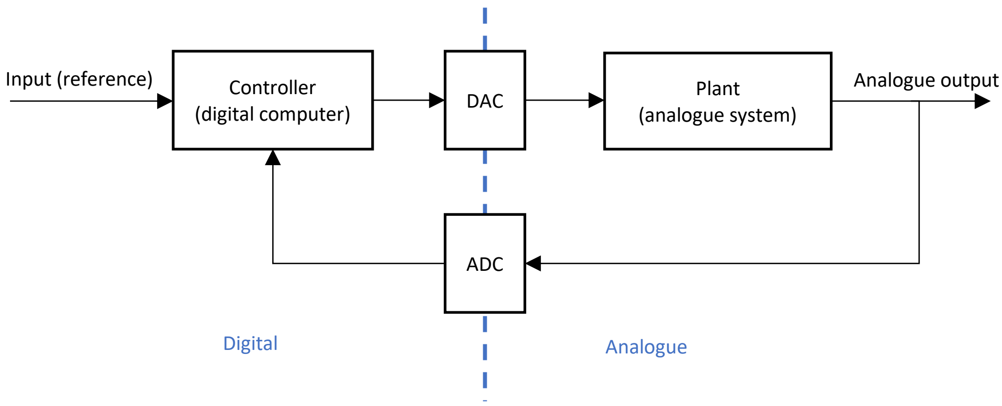
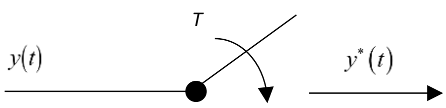


Figure 4. Main components of a digital control loop (with analogue plant)

5.1 Analogue-to-digital converter (ADC)

If, as we stated in the introduction, we neglect the effects of quantisation, and assume that the analogue-to-digital converter (ADC or A/D) responds instantaneously, then the ADC can be modelled as an ideal sampler, e.g. a device that "closes the circuit" for an infinitesimal amount of time, and let an ideal impulse pass, with magnitude equal to the magnitude of the continuous signal being sampled, at the time of sample (i.e. at each $t = kT$):



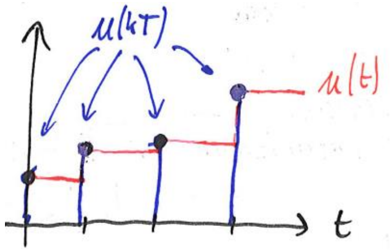
In practice, a sampled version $y^*(kT)$ of a continuous signal $y(t)$ can be obtained by "extracting" the values of the signal at each sample:

$$y(t) \to y^*(kT), \; k = 0,1,2,\ldots$$

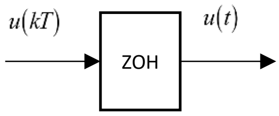
📖 [1] 5.1
📖 [2] 3.2

5.2 Digital-to-analogue converter (DAC): the zero-order hold (ZOH)

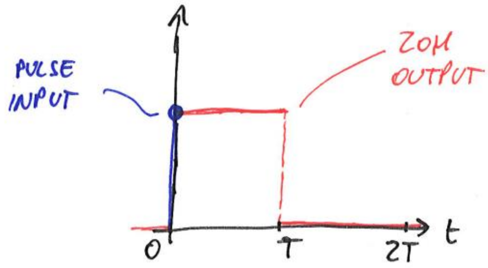
Even assuming ideal conditions (no errors in magnitude and instant action), a digital-to-analogue converter (DAC) is more complex than its inverse counterpart, if only because there are infinite ways to recreate a continuous signal from its samples (i.e. “filling the gaps” between samples).
The simplest implementation is the zero-order hold (ZOH), a DAC that takes a sample $u(kt)$ and holds this value (constant) until time $t = kt + t$, when next sample arrives:



5.2.1 Transfer function of the ZOH



We know that the discrete TF is the z-transform of the output for an impulse input at $k = 0$:



In the time domain, the response of the ZOH to a unit impulse can be written as

$$u(t) = 1(t) - 1(t - T)$$

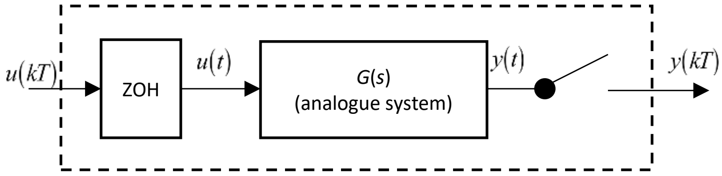
Remembering that $\mathcal{L}\{1(t)\} = \frac{1}{s}$ and that a shift in time of *T* corresponds to a multiplication by e^{-Ts} in the frequency domain:

$$G_{ZOH}(s) = \frac{1}{s} - \frac{e^{-sT}}{s} = \frac{1 - e^{-sT}}{s}$$

📖 [1] 5.1
📖 [2] 3.3

5.2.2 Continuous system preceded by a ZOH and followed by a sampler

We are now interested in modelling a continuous transfer function preceded by a ZOH. This is often the case for the plant of the system. We seek to obtain a “discrete equivalent” of the subsystem “(continuous) system preceded by a ZOH” in the discrete domain. In this way, we can substitute the “ZOH+continuous system+sampler” subsystem in the mixed continuous/digital model in Figure 4, and obtain a fully-digital model, which can be more easily analysed.



Discrete equivalent preceded by a ZOH: G_{ZAS}

The sampled signal in z-domain can be modelled with a sequence of inverse Laplace and (direct) z-transform of the continuous signal in the s-domain.

$$Y(z) = \mathrm{Z}\{y(kt)\} = \mathrm{Z}\{\mathrm{L}^{-1}\{Y(s)\}\}$$

The combination of ZOH, *G*(*s*) and sampler can be modelled as:

$$G_{ZAS}(z) = \mathrm{Z}\left\{\mathrm{L}^{-1}\left\{\frac{1-e^{-sT}}{s}G(s)\right\}\right\} \tag{5.1}$$

Where, in the subscript: Z = ZOH, A = analogue transfer function (usually the plant), S = sampler.
Noting that e^{-sT} is a time-shift of one sample (*T*), and therefore can be taken out of the (anti-)transformations:

$$\mathrm{Z}\left\{\mathrm{L}^{-1}\left\{e^{-sT}\frac{G(s)}{s}\right\}\right\} = z^{-1}\mathrm{Z}\left\{\mathrm{L}^{-1}\left\{\frac{G(s)}{s}\right\}\right\}$$

Eq. (5.1) becomes:

$$G_{ZAS}(z) = (1-z^{-1})\mathrm{Z}\left\{\mathrm{L}^{-1}\left\{\frac{G(s)}{s}\right\}\right\} \tag{5.2}$$

Or with slight abuse of notation, we can simply say:

$$G_{ZAS}(z) = (1-z^{-1})\mathrm{Z}\left\{\frac{G(s)}{s}\right\} \tag{5.3}$$

We will show that the effect of the ZOH can be taken into account if designing the controller in the continuous domain (by emulation), by introducing a time delay of $\frac{T}{2}$ (see Sec. 8.4).

Exercise 22 Discrete equivalent of second-order system with ZOH

Find the discrete TF of $G(s) = \frac{1}{s^2}$ preceded by a ZOH.

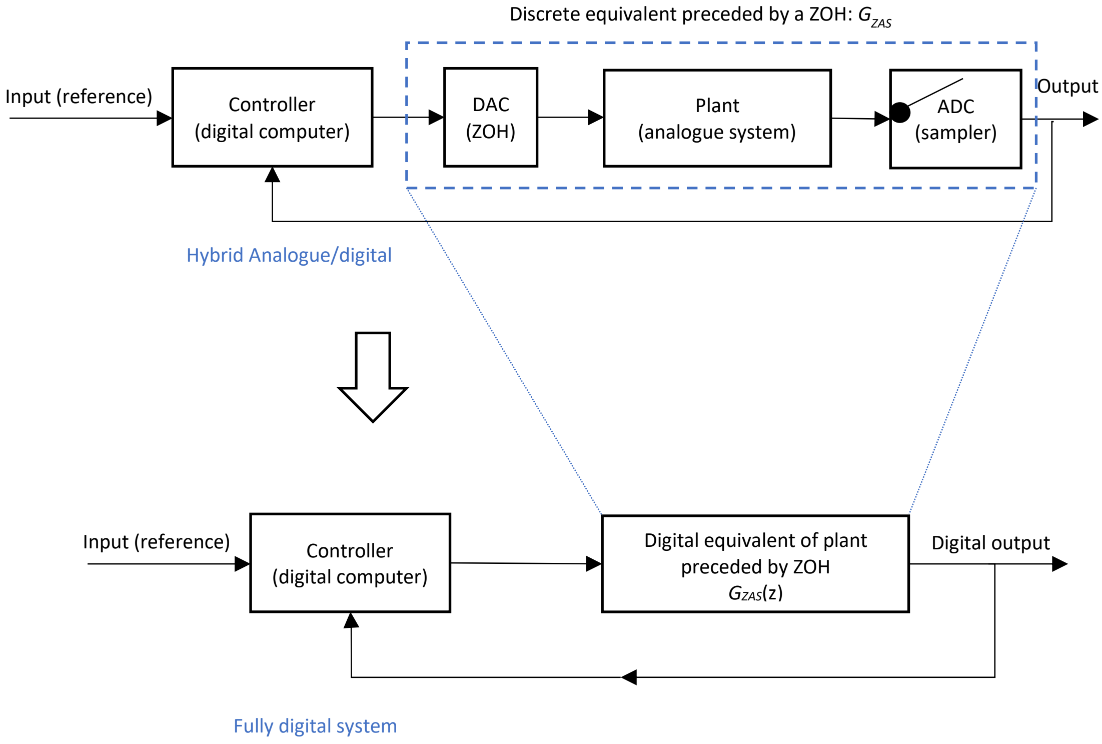
.

Exercise 23 Discrete equivalent of low-pass filter with ZOH

Find the ZOH equivalent of:

$$H(s) = \frac{a}{s+a}$$

By using this equivalence, we can transform a mixed continuous-digital system into a fully-digital one:



Exercise 24 Computer-aided verification of discrete equivalents

Use a computer tool (MATLAB/Simulink) to show numerically that the discrete equivalent of the plant calculated with Eq. (5.3) in a fully-digital system produces the same output as the combined system: ZOH+plant+sampler in a continuous system.
Consider for example the plant:

$$G(s) = \frac{10}{s + 0.1}$$

And use a sine wave of unit frequency for simplicity.

.

📖 [1] 4.3.1
📖 [2] 3.5

💻 MATLAB functions: c2d

5.2.3 State-Space form

Computing the z-transform using Eq. (5.2) requires transforming and anti-transforming, which can be tricky for complicated $G(s)$.
There is another method to find the discrete equivalent of a TF preceded by a ZOH which does not depend on Laplace (anti-) transformations and therefore it is suitable to be automatized. This requires the TF to be written in the state-space form.

Remember: Continuous systems can be written as linear, time-invariant

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}u \\ y &= \mathbf{H}\mathbf{x} + \mathbf{J}u\end{aligned} \quad \text{State space representation}$$

Note about notation: **G** is a matrix and has nothing to do with the transfer function $G(s)$ or $G(z)$.

A similar state-space form exists for discrete systems:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k) \\ y(k) &= \mathbf{H}\mathbf{x}(k) \end{aligned} \qquad \text{(Discrete state-space form)}$$

Note that the derivative of the state has been replaced by the state at the next step (*k*+1). We use the corresponding Greek letters $\mathbf{\Phi}, \mathbf{\Gamma}$ for the matrices of discrete system.

The transfer function can also be computed in a very similar way. Taking the z-transform of the discrete state-space form:

$$\begin{cases} (z\mathbf{I} - \mathbf{\Phi})\mathbf{X}(z) = \mathbf{\Gamma}U(z) \\ Y(z) = \mathbf{H}\mathbf{X}(z) \end{cases}$$

$$\frac{Y(z)}{U(z)} = \mathbf{H}[z\mathbf{I} - \mathbf{\Phi}]^{-1}\mathbf{\Gamma}$$

Note that the denominator of $Y(z)/U(z)$ is $\det(z\mathbf{I} - \mathbf{\Phi})$ which is the characteristic polynomial of the transfer function. So, the zeros of characteristic polynomial are poles of the transfer function.

It is possible to find the discrete equivalent of a system directly in state-space form when preceded by a ZOH. It can be demonstrated that the digital equivalent of a continuous system, expressed in state-space form through matrices **F**, **G** and **H**, preceded by a ZOH, can be expressed with the following matrices:

$$\mathbf{\Phi} = e^{\mathbf{F}T} = \mathbf{I} + \mathbf{F}T + \frac{\mathbf{F}^2 T^2}{2!} + \ldots + \frac{\mathbf{F}^m T^m}{m!} + \ldots$$

$$\mathbf{\Gamma} = \int_0^t e^{\mathbf{F}\tau} d\tau \mathbf{G}$$

The discrete transfer function of G(s) preceded by a ZOH and sampled is therefore given by:

$$G_{ZAS}(z) = \frac{Y(z)}{U(z)} = \mathbf{H}[z\mathbf{I} - \mathbf{\Phi}]^{-1}\mathbf{\Gamma}$$

So in conclusion we have:

	Continuous		Discrete equivalent (preceded by ZOH)
State-space form	$\begin{cases} \dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u \\ y = \mathbf{H}\mathbf{x} \end{cases}$	→	$\begin{cases} \mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathrm{u}(k) \\ y(k) = \mathbf{H}\mathbf{x}(k) \end{cases}$
↕	$G(s) = \mathbf{H}(s\mathbf{I} - \mathbf{F})^{-1}\mathbf{G}$		$G_{ZAS}(z) = \mathbf{H}(z\mathbf{I} - \mathbf{\Phi})^{-1}\mathbf{\Gamma}$
Transfer function	$G(s)$	→	$G_{ZAS}(z) = (1 - z^{-1})\mathrm{Z}\left\{\frac{G(s)}{s}\right\}$

Exercise 25 Discrete equivalent with ZOH in state-space

Find the discrete equivalent of the same TF in Exercise 22, $G(s) = \frac{1}{s^2}$, preceded by a ZOH in the following way:

$G(s) \rightarrow$ Continuous state-space form $\rightarrow$ Discrete state-space form

And show that it leads to the same result in Exercise 22.

.

[1] 4.3.3

5.3 Closed-loop systems

After converting the plant, ZOH and sampler into a discrete equivalent, we can model a canonical digital feedback control loop from Figure 4 as follows:

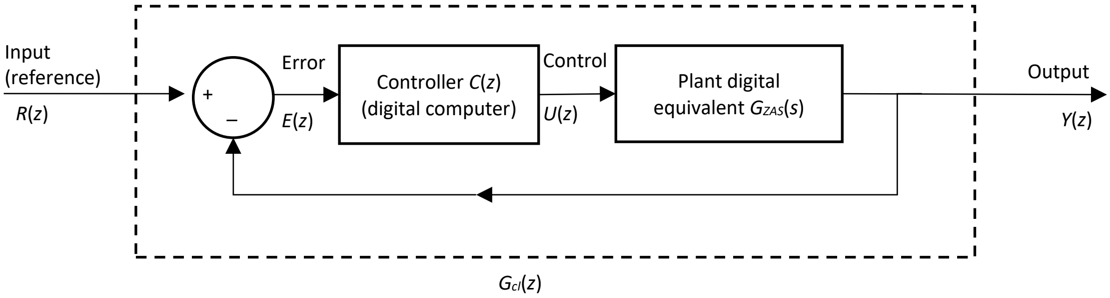


Figure 5. Digital feedback loop

We have replaced the plant, the ZOH and the sampler with the digital equivalent, and the system is entirely digital now. For this reason, it is possible to model it with a single discrete transfer function, which is the *closed-loop transfer function* (remember Sec. 3.3):

$$G_{cl}(z) = \frac{C(z)G_{ZAS}(z)}{1+C(z)G_{ZAS}(z)} \tag{5.4}$$

We can define the *open loop transfer function* (or *open loop gain*) of the controlled system:

$$L(z) \triangleq C(z)G_{ZAS}(z)$$

The roots of the characteristic equation:

$$1+L(z)$$

are the closed-loop system poles, and therefore play a fundamental role in the design of the controller, and the stability and response of the controlled system.

[2] 3.7

5.4 Steady-state error

The error with respect to the reference, from Figure 5, can be expressed as:

$$\begin{aligned} E &= R - Y \\ &= R - G_{cl}R \\ &= (1-G_{cl})R \end{aligned}$$

The steady-state error (in the time domain) can be obtained with the final value theorem, Eq. (3.9):

$$\begin{aligned} e(\infty) &= \lim_{z\to 1}\left[(z-1)E(z)\right] \\ &= \lim_{z\to 1}\left[(z-1)(1-G_{cl})R\right] \\ &= \lim_{z\to 1}\left[(z-1)\left(1-\frac{L}{1+L}\right)R\right] \\ &= \lim_{z\to 1}\frac{(z-1)R}{1+L} \end{aligned} \tag{5.5}$$

It is clear that this limit exists if all terms (*z* - 1) at denominator cancel out, and this depends both on *L* and *R*. Starting from *L*, we can isolate all terms (*z* - 1) from numerator and denominator:

$$L = \frac{N_L^*}{(z-1)^n D_L^*}$$

Where N^*_L and D^*_L are numerator and denominator of *L*, without any factor (*z* - 1), and *n* is called *type number*. Poles at *z* = 1 are extremely important for digital systems, and play an analogous role as poles at *s* = 0 for continuous systems.
Substituting this expression for *L*:

$$e(\infty) = \lim_{z\to 1} \frac{(z-1)R}{1+L(z)} = \lim_{z\to 1} \frac{(z-1)R(z)}{1+\dfrac{N_L^*(z)}{(z-1)^n D_L^*(z)}} = \lim_{z\to 1} \frac{(z-1)^{n+1} D_L^*(z) R(z)}{N_L^*(z)+(z-1)^n D_L^*(z)} = \lim_{z\to 1} \frac{(z-1)^{n+1} D_L^*(1) R(z)}{N_L^*(1)+(z-1)^n D_L^*(1)}$$

Where N^*_L and D^*_L have been evaluated in *z* = 1 because there will not be any zero in that point.

5.4.1 Step input

Let's now assume a unit step input, Eq. (4.1):

$$R(z) = \frac{z}{z-1} \qquad \text{(Unit step)}$$

Substituting:

$$e(\infty) = \lim_{z\to 1} \frac{(z-1)^{n+1} D_L^*(1) \dfrac{z}{z-1}}{N_L^*(1)+(z-1)^n D_L^*(1)} = \lim_{z\to 1} \frac{(z-1)^n D_L^*(1)}{N_L^*(1)+(z-1)^n D_L^*(1)}$$

From inspection, we can conclude that, depending on the type number:

$$e(\infty) = \begin{cases} \dfrac{D_L^*(1)}{N_L^*(1)+D_L^*(1)} = \dfrac{1}{1+L(1)}, & n=0 \\ 0, & n>0 \end{cases} \tag{5.6}$$

Similar analysis can be done for other inputs, such as ramp and parabola.

5.4.2 Ramp input

The ramp input is significant as it represents a continuously-varying reference.

$$R(z) = \frac{Tz}{(z-1)^2} \qquad \text{(unit ramp)}$$

Starting again from Eq. (5.5) and substituting the Ramp:

$$e(\infty) = \lim_{z\to 1} \frac{(z-1)\dfrac{Tz}{(z-1)^2}}{1+L(z)} = \lim_{z\to 1} \frac{Tz}{(z-1)(1+L(z))} = \lim_{z\to 1} \frac{T}{(z-1)L(z)}$$

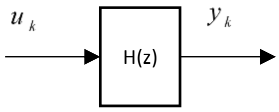
If type *n* = 1, then the term (z – 1) vanishes and the expression has a finite value. Conversely, if type *n* = 0 or *n* > 1, then the term (z – 1) remains and the error goes to infinity or zero:

$$e(\infty) = \begin{cases} \infty, & n=0 \\ \lim\limits_{z\to 1} \dfrac{T}{(z-1)L(z)}, & n=1 \\ 0, & n\ge 2 \end{cases} \tag{5.7}$$

📖 [2] 3.9, 3.9.1, 3.9.2

6 Stability

Consider a digital system with output *y* and input *u* (e.g. the plant):



6.1 Asymptotic stability

A system is *asymptotically stable* if its steady-state (i.e. at infinite time) response to any initial condition decays to zero, that is:

$$y(\infty)=0, \quad \forall y(0) \quad \text{(asymptotic stability)} \tag{6.1}$$

We now try to find a condition on the transfer function for asymptotic stability. Consider a system with the following difference equation:

$$a_0 y_k + a_1 y_{k+1} + \dots + a_n y_{k+n} =$$
$$= b_0 u_k + b_1 u_{k+1} + \dots + b_m u_{k+m}$$

Subject to the *n* initial conditions:

$$y_0, y_1, \dots, y_{n-1}$$

By applying the z-transform and remembering the left shift theorem, Eqs. (3.4) and (3.5):

$$a_0 Y(z) + a_1\left[zY(z) - zy_0\right] + \dots + a_n\left[z^n Y(z) - z^n y_0 - z^{n-1} y_1 - \dots\right] =$$
$$= b_0 U(z) + b_1 z U(z) + \dots + b_m z^m U(z)$$

As we are interested in response to initial conditions, we can assume zero input. Rearranging the equation for the output *Y*(*z*):

$$Y(z) = \frac{-a_1 y_0 z - \dots - a_n y_0 z^n - a_n y_1 z^{n-1} - a_n y_n}{a_0 + a_1 z + \dots + a_n z^n}$$

Compare this to the transfer function of the same system:

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 + b_1 z + \dots + b_m z^m}{a_0 + a_1 z + \dots + a_n z^n}$$

It can be seen that the numerator of the transfer function *H* depends on the coefficients of the input terms b_i, we can state that the zeros of the transfer function have no influence on the response due to initial conditions.

Instead, the denominator of the transfer function *H* is the same as the denominator of the z-transformed output *Y*, <u>if there are no zero/pole cancellations in the transfer function</u>. Therefore, we can conclude that the output *Y* due to initial conditions is bounded for system poles in the unit circle ($|z|<=1$), and decays exponentially for system poles inside the unit circle ($|z|<1$).

Note that this condition also applies in case of pole/zero cancellation, <u>if the cancelled poles/zeros are stable</u> ($|z|<1$). However, a cancellation of an unstable pole/zero would invalidate the condition.

📖 [2] 4.1, 4.3.1

6.2 BIBO stability

As opposed to asymptotic stability, which refers to initial conditions, external stability or bounded-input/bounded-output (BIBO) stability refers to the fact that the response of a system with bounded inputs is also bounded (regardless of its internal states), that is:

$$|y_k| \le N, \quad \forall |u_j| \le M$$

Suppose the input is bounded:

$$|u_j| \le M < +\infty \ \forall j$$

The response (in magnitude) is, remembering the discrete convolution (Eq. (3.15)):

$$|y_k| \underset{\substack{\uparrow \\ \text{Convolution}}}{=} \left|\sum_j u_j h_{k-j}\right| \le \sum_j |u_j| |h_{k-j}| \underset{\substack{\uparrow \\ \text{Bounded input}}}{\le} M \sum_j |h_{k-j}|$$

So output is bounded if, for any bounded input:

$$\sum_{j=-\infty}^{+\infty} |h_{k-j}| < +\infty$$

Or, removing unnecessary indexes:

$$\sum_k |h_k| < +\infty \qquad \forall |u_j| \le M \tag{6.2}$$

So, a system is BIBO-stable if the time history of its transfer function (h), or equivalently the time history of its impulse response, is absolutely summable, i.e. the sum converges to a finite value. It can be shown that this condition is also necessary.

Exercise 26 BIBO stability of difference equation

Determine BIBO stability of:

$$u_k = au_{k-1} + be_k$$

.

[1] 4.2.5
[2] 4.3.2

6.2.1 BIBO stability condition in frequency domain

We found a condition for BIBO stability in the time domain, Eq. (6.2), i.e. a condition on the time history of the transfer function, h_k (or equivalently impulse response). Is there an analogue condition in the frequency domain, i.e. using the expression $H(z)$?

Expand the impulse response in a series:

$$|h_k| = \left|\sum_{i=1}^{n} A_i p_i^k\right| \le \sum_{i=1}^{n} |A_i| |p_i^k| \le n|A_{\max}| |p_{\max}|^k$$

Where $p_{\max}$ and A_{max} are the pole and coefficient with largest magnitude. Using the BIBO stability condition Eq. (6.2):

$$\sum_{j=0}^{\infty} |h_j| \le n|A_{\max}| \sum_{k=0}^{\infty} |p_{\max}|^k$$

The summation is a geometric series, hence:

$$\sum_{j=0}^{\infty} |h_j| \le n|A_{\max}| \frac{1}{1-|p_{\max}|} < +\infty \qquad if \quad |p_{\max}| < 1$$

Meaning the largest pole must be inside the unit circle, and hence all poles must also be. If can be shown that this is a necessary and sufficient condition.

In conclusion, a digital (linear) system is BIBO-stable if (and only if) all poles are inside the unit circle. Therefore, the BIBO stability of the system can be determined computing the locations of the poles (roots of the characteristic equation) and determining whether they are inside the unit circle. If the system has any pole outside of the unit circle, then it is unstable.

If there are no zero/pole cancellations, then the two definitions of stability – asymptotic and BIBO – coincide, and all poles inside the unit circle is a condition that ensures both. In such case, it is common to simply refer to *stability* in general.

Exercise 27 BIBO stability of numerical integration

Determine the BIBO stability of backward, forward and trapezoid integration rules, using both conditions in the time domain and frequency domain. Explain results.

.

[1] 4.2.5

[2] 4.3.2

6.3 Nyquist stability criterion

The Nyquist stability criterion for digital systems is very similar to the one for continuous systems. It is assumed here that the reader is familiar with that.

By considering the closed loop TF:

$$G_{cl} = \frac{L(z)}{1+L(z)} = \frac{CG}{1+CG}$$

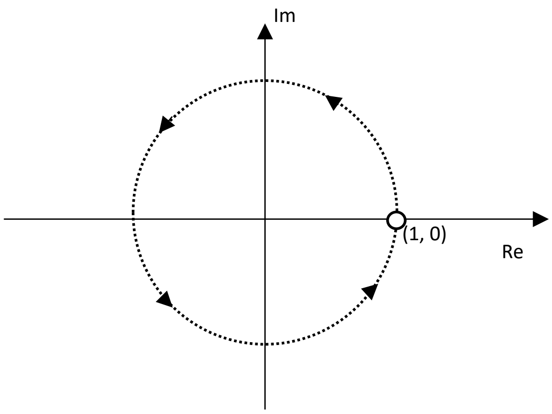
The characteristic polynomial of the closed loop is:

$$p_{cl} = 1 + L = 1 + \frac{N_L}{D_L} = \frac{N_L + D_L}{D_L}$$

We note that:

- The poles of the closed loop TF G_{cl} are the zeros of p_{cl}, i.e. the zeros of $N_L + D_L$
- The poles of the open-loop TF L are the poles of p_{cl}

We now consider a contour in the complex plane that is the unit circle, starting from (1,0) (zero frequency) and circling anticlockwise to return to the same point. This path is encircling the stability region, and it is essentially equivalent to the negative semi-plane that is used for continuous systems.

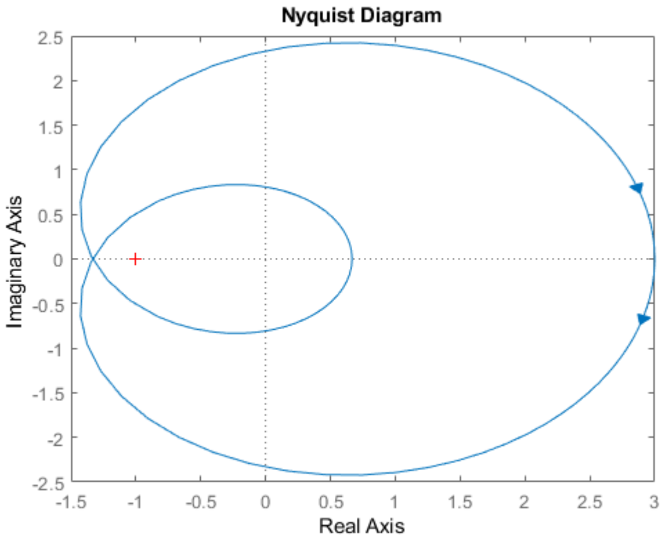


To draw the Nyquist diagram, we evaluate the open-loop transfer function $L(z)$ for z moving on the contour described, i.e.:

$$z = e^{j\omega T}, \omega = 0 \ldots \frac{\pi}{T}$$

$$z = e^{j\omega T}, \omega = -\frac{\pi}{T} \ldots 0$$

We can then observe the Nyquist diagram, for example:



We can count the number of (clockwise) encirclements of the Nyquist plot around the point (-1,0), and denote this with *N*.
Let's now call:

- *P* = number of open-loop poles outside the unit circle (unstable poles of *L*, open-loop)
- *Z* = number of closed-loop poles outside the unit circle (unstable poles of G_{cl}, closed-loop)

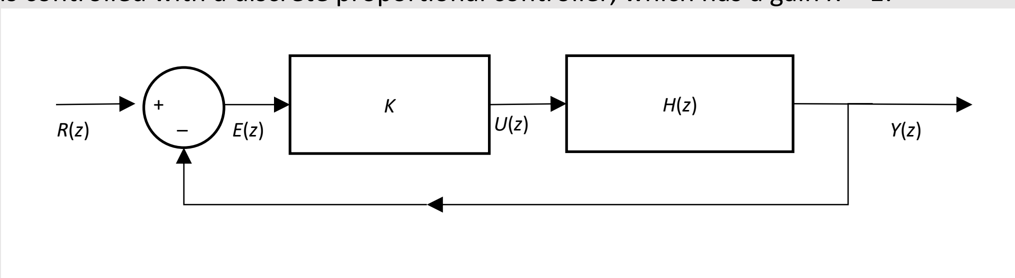
If we know the number *P*, then we can quickly compute Z as:

$$Z = P - N \quad \text{(Nyquist criterion)} \tag{6.3}$$

If the open-loop system is stable (no unstable poles, *P* = 0), then the closed-loop system is stable if and only if the Nyquist plot does not encircle (-1, 0) (*N* = 0).

Exercise 28 Closed-loop stability of proportional controller with Nyquist criterion

A plant is controlled with a discrete proportional controller, which has a gain *K* = 1:



$$H(z) = \frac{0.4}{(z-0.5)(z-0.2)}$$

Evaluate the closed-loop stability using the Nyquist diagram/criterion.

.

Note that the Nyquist plot must be a closed path, and in certain cases, it is closed "at infinity". See the following example:

Exercise 29 Stability with Nyquist plot

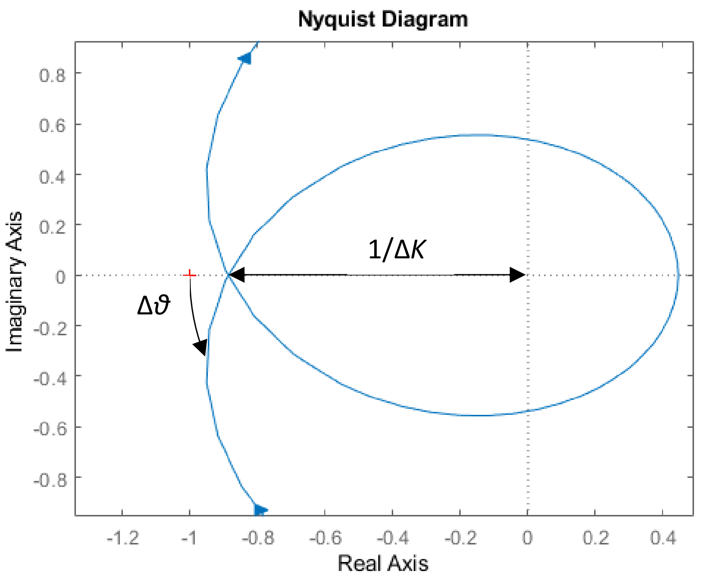
Use the Nyquist plot to determine the stability of the closed-loop system where the plant is:

$$G(z)=\frac{10}{(z-1)(z-0.1)}$$

[2] 4.3.2
https://mycourses.aalto.fi/pluginfile.php/134622/mod_folder/content/0/Chapter_3%20(print).pdf

6.3.1 Gain and phase margin

The *gain margin* is the gain change that makes the system marginally stable.
The *phase margin* is the negative phase perturbation that makes the system marginally stable.
The idea here is exactly the same as for continuous systems: by looking at the Nyquist plot, identify the change in gain and phase that make the plot to "touch" the point (-1, 0).
The effect of a gain ΔK is to scale the Nyquist plot without rotation.
The effect of a factor $e^{-j\Delta\theta}$ is to rotate the plot clockwise without changing its magnitude.
Therefore, the gain and phase margins can be represented graphically as such:

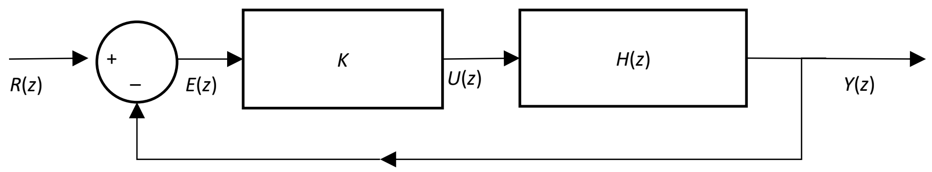


The gain margin can be obtained analytically by finding the point the Nyquist plot intersects the real axis, equating the imaginary part of the frequency response to zero, solving for the frequency, then solving the frequency response at that frequency for the real part.
The phase margin can be obtained by equating the magnitude of the frequency response to unity and solving for angle, and then adding 180 degrees.

Because these techniques are essentially the same as for continuous systems, a full description for discrete systems is unnecessary.

Exercise 30 Gain margin

Find the gain margin for the system in the previous Exercise 29 (K = 1):



$$H(z) = \frac{0.4}{(z-0.5)(z-0.2)}$$

.

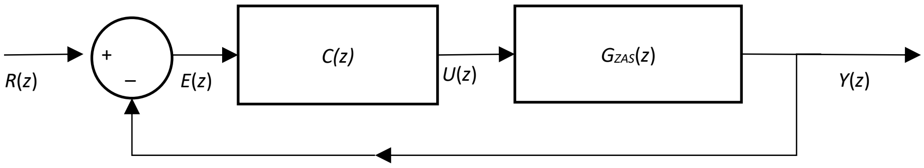
[2] 4.6.1

https://mycourses.aalto.fi/pluginfile.php/134622/mod_folder/content/0/Chapter_3%20(print).pdf

7 Design of digital controllers

7.1 Root locus

The root locus is a technique that allows to map the position of the closed-loop poles in a feedback system, as the proportional gain of the controller varies, given the open-loop transfer function of the same system.



The root locus technique assumes that either a controller has already been designed and its gain needs be tuned, or that the controller is a simple proportional controller only (and therefore the proportional gain is the only parameter).
The closed loop characteristic equation of the controlled system is:

$$1+C(z)G_{ZAS}(z)=0$$

This can be rewritten making the controller gain *K* explicit:

$$1+KL(z)=0 \tag{7.1}$$

Where *L*(*z*) is the open loop gain (or transfer function), which for the purposes of the root locus, can include the zeros and poles of the controller *C*(*z*), if any.
Equation (7.1) (a complex equality) is equivalent to the following two (real) conditions:

$$\angle L(z)=(2m+1)180°, \quad m\in\mathbb{Z} \quad \text{(angle condition)} \tag{7.2}$$

$$K\left|L(z)\right|=1 \quad \text{(magnitude condition)} \tag{7.3}$$

Following Eq. (7.1), the closed-loop characteristic equation therefore is:

$$1+K\frac{N_L(z)}{D_L(z)}=0 \ \rightarrow \ D_L(z)+KN_L(z)=0 \tag{7.4}$$

The root locus can be plotted with a computer-based tool (MATLAB), or manually: all the techniques for plotting the root locus in the continuous (s-) domain apply to the digital (z-) domain.

Exercise 31 Root locus and critical gain

Find the root locus plot and the critical gain for the open-loop system:

$$L(z)=\frac{2}{(z-1)(z+0.1)}$$

.

[2] 6.1

[1] Simons, Stuart, "Alternative approach to complex roots of real quadratic equations", Mathematical Gazette 93, March 2009, 91–92.

7.2 Z-domain design

The poles can be positioned directly in the z-plane, using the techniques and insights we got in Sec. 4.2. In particular, the “grid” in the z-plane (Figure 3) can be used to place complex conjugated poles to obtain a certain frequency and damping response.
Control system design specifications can be formulated in a similar way to what done for continuous systems (see Sec. 1.7), however these are defined in terms of the continuous envelopes of the discretised signal:

1. **Time constant** τ : time required for the continuous envelope to reach 63% of the final response value

$$\tau = \frac{1}{\zeta\omega_n} \tag{7.5}$$

2. **Settling time** T_s : Time after which the continuous envelope response remains within 2% of the final response value.

$$T_s = \frac{4}{\zeta\omega_n} \tag{7.6}$$

3. **Frequency of oscillations** ω_d : Angle of the dominant complex conjugate poles divided by the sampling period *T*.

Exercise 32 Digital proportional controller

Design a proportional controller for the open-loop transfer function in the previous Exercise 31:

$$L(z) = \frac{2}{(z-1)(z+0.1)}$$

With a sample time $T = 0.1\,s$, for the following design specifications:

A. Damped frequency $\omega_d = 6\,rad/s$

B. Time constant $\tau = 0.3s$

C. Damping ratio $\zeta = 0.8$

.

Exercise 33 Digital proportional controller for position control with viscosity

Design a digital proportional controller for the position control system with viscosity:

$$G(s) = \frac{1}{s(s+10)}$$

With a sample time $T = 0.05$, for the following design specifications:

A. Steady-state error due to ramp of 10%, $e(\infty) = 0.1$.

B. Damping ratio $\zeta = 0.8$ (Use CAD as required). What is the steady-state error in this case?

C. What would you do to achieve a zero steady-state error?

.

MATLAB functions: ss, tf, step, rlocus, zmap

[2] 6.1

7.3 Direct z-domain PID design

With this technique, we select a sample time, convert the plant to its digital equivalent, and place a digital controller directly in line with the discrete equivalent of the plant and tune it. The fundamental digital controllers are:

$$C(z) = K_P + K_I T \frac{z}{z-1} = K \frac{z-a}{z-1} \qquad \text{(PI)} \tag{7.10}$$

$$C(z) = K_P + \frac{K_D}{T} \frac{z-1}{z} = K \frac{z-a}{z} \qquad \text{(PD)} \tag{7.11}$$

$$C(z) = K_P + K_I T \frac{z}{z-1} + \frac{K_D}{T} \frac{z-1}{z} = K \frac{(z-a)(z-b)}{z(z-1)} \qquad \text{(PID)} \tag{7.12}$$

The type of controller is usually selected to achieve a design specification (e.g. zero steady-state error due to step or ramp); the zero(s) is (are) usually selected to cancel unstable or undesired poles of the plant; the gain is then tuned as usual with the root locus or other techniques.

Exercise 34 Digital DC motor controller in the z-domain

Consider the DC motor modelled in Exercise 3:

$$G(s) = \frac{K_m}{\tau_m s + 1}$$

with $\tau_m = 0.1\,\mathrm{s}$, $K_m = 1$. Design a digital controller directly in the z domain with the following requirements for the closed-loop system: $\zeta = 0.7$, $\omega_n = 25\ \mathrm{rad/s}$. Assume that, due to hardware limitations, the sample time cannot be higher than $T = 0.0341\ \mathrm{s}$.

.

Exercise 35 Digital controller of DC motor for speed with specifications

Design a digital controller for a DC motor speed with plant:

$$G(s) = \frac{1}{(s+0.5)(s+5)}$$

With zero steady-state error due to step, percentage overshoot less than 10% ($P_O < 0.1$), using a sample time of T = 0.1 s.
Use computer aided design for solving non-linear equations.
Verify whether a settling time of approximately 1.5 s ($T_s = 1.5\,s$) can be achieved.

.

Exercise 36 Digital controller direct design for speed control plant

Design a digital controller directly for the speed control system, with plant:

$$G(s) = \frac{1}{(s+1)(s+3)}$$

With zero steady state error due to step, percentage overshoot less than 5%, time constant of less than 0.3 s ($\tau < 0.3\,s$). Use a sampling time $T = 0.025\,s$.
Use computer aided design for solving non-linear equations.

.

[2] 6.4

https://msc.berkeley.edu/assets/files/PID/modernPID4-digitalPID.pdf

7.4 Design of discrete equivalents (design by emulation)

In this section we focus on emulation, i.e. finding a discrete equivalent to a continuous transfer function such to have similar characteristics, for example with respect to bandwidth.
Applications:

- Bandpass filters used in radios to select one specific frequency and reject the others
- Rejection band filters to remove one particular frequency causing noise (e.g. 50 – 60 Hz of power-line)
- Equalizers to adjust some amplitude or phase unwanted distortion introduced (e.g. Radio transmission)
- Compensators to adjust the dynamic response of a system (e.g. Control)

The emulation will allow to design these filters in continuous time domain and then implement them in digital preserving their frequency features. In general, it is important that the digital controller (or filter) preserves the stability and frequency response (in the range of useful frequencies of the digital controller [0, $\left[0, \omega_s/2\right]$) of the analogue one; this is however not always true, or possible.

Design by emulation requires designing the continuous (analogue) controller first, then converting it to its digital counterpart and then tuning it, if needed.

1. Design a continuous controller for the analogue plant to meet the design specifications.
2. Map the controller to a discrete equivalent using a suitable transformation (see one of the next subsections)
3. Tune the digital controller in combination with the digital equivalent of the plant (G_{ZAS})
4. Check the response and iterate if necessary

[2] 6.3

7.4.1 Numerical integration

We use the numerical integration rules (see Section 2.4) to convert a continuous transfer function to a digital one.
Using the forward rectangular rule, any differentiation can be approximated as:

$$\dot{y}_k \simeq \frac{y_{k+1} - y_k}{T} \tag{7.13}$$

This can be applied directly to the differential equation of the system, to find its digital equivalent. The digital transfer function can then be retrieved through z-transform.
Alternatively, we can apply the Laplace transform to the left-hand side of Eq. (7.13) (continuous side) and z-transform to the right-hand side (discrete side) to find the following correspondence:

$$sY(s) \leftrightarrow \frac{z-1}{T} Y(z)$$

We can then infer that a direct transformation of a continuous transfer function can be obtained through the following substitution:

$$s \leftarrow \frac{z-1}{T} \quad \text{(forward rectangular)} \tag{7.14}$$

Exercise 37 Discrete equivalent with forward rectangular rule

Let's consider a generic first-order filter transfer function:

$$\frac{U(s)}{E(s)} = H(s) = \frac{a}{s+a}$$

Find the discrete equivalent with direct numerical integration, forward rectangular rule, explicitly using the integration rule in Eq. (2.1).

.

Exercise 38 Derivation of backward and trapezoid differentiation rules

Derive the direct substitutions for the backward and trapezoid differentiation rules, in an analogous way as done in Sec. 7.4.1, Eqs. (7.13)-(7.14).

.

Exercise 39 Discrete equivalent of LP filter by direct numerical integration

Derive the discrete equivalent transfer function of filter $U(s) = \frac{a}{s+a}$ by direct numerical integration using backward rule and trapezoid rule.

.

It can be shown that the following substitutions apply:

$$s \leftarrow \frac{z-1}{T} \quad \text{(forward rectangular)}$$
$$s \leftarrow \frac{z-1}{Tz} \quad \text{(backward rectangular)}$$
$$s \leftarrow \frac{2}{T}\frac{z-1}{z+1} \quad \text{(trapezoid/Tustin/bilinear)} \tag{7.15}$$

The trapezoid rule substitution is also called Tustin's method or bilinear transformation.

[1] 6.1

7.4.2 Stability

These transformations are maps from s-plane to z-plane. Let's see how the poles and the stability line $j\omega$ (stability boundary) transform into the z-plane. Inverting the transformations:

$$\begin{aligned} Forward &: \quad z = 1+Ts \\ Backward &: \quad z = \frac{1}{1-Ts} \\ Bilinear &: \quad z = \frac{1+\frac{T}{2}s}{1-\frac{T}{2}s} \end{aligned}$$

If we substitute $s = j\omega$ (imaginary axis) we see where the imaginary axis (the stability boundary) is mapped in the z-plane.

For the forward rule:

$$z = 1 + Tj\omega \;\rightarrow \text{Shifted vertical line}$$

For the backward rule:

$$z = \frac{1}{1-Tj\omega} = \frac{1}{2} + \frac{1}{1-Tj\omega} - \frac{1}{2} =$$

$$= \frac{1}{2} + \frac{2-1+Tj\omega}{2(1-Tj\omega)} = \frac{1}{2} + \frac{1+Tj\omega}{2(1-Tj\omega)} = \frac{1}{2} + \frac{1}{2}\frac{1+Tj\omega}{1-Tj\omega}$$

Hence:

$$z - \frac{1}{2} = \frac{1}{2}\frac{1+Tj\omega}{1-Tj\omega}$$

$$\left|z - \frac{1}{2}\right| = \left|\frac{1}{2}\frac{1+Tj\omega}{1-Tj\omega}\right| = \frac{1}{2}$$

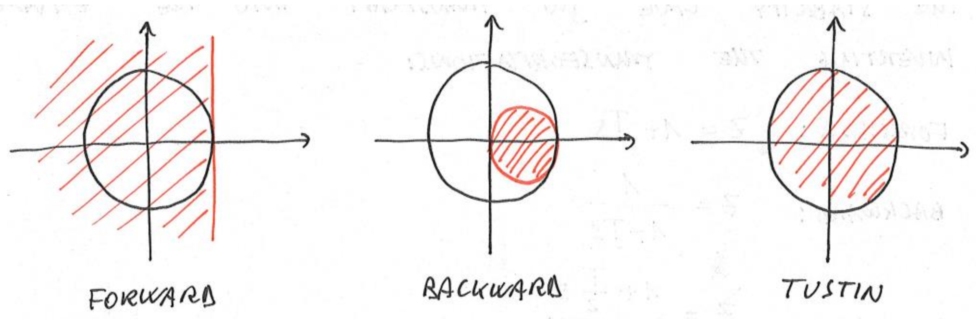
This is the equation of a circle with radius $r = \frac{1}{2}$ and centre $= \frac{1}{2}$.

For the bilinear transformation (Tustin):

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

$$|z| = 1$$

This represents the unit circle.



From here it can be seen that the forward rule maps the stable semiplane $\mathrm{Re}(s) < 0$ into a semiplane $|z| < 1$ and hence out of the unit circle. This rule can make a stable continuous filter unstable in the digital domain.

The backward rule guarantees that a stable continuous controller remains stable, however it does not cover the full range of frequencies and damping in the digital domain, as poles can only be placed in a subset of the unit circle.

Finally, the Tustin rule preserves stability and maps the entire stability region in the whole unit circle.

The following few examples will show design of PI/PD controllers by emulation, using the Tustin rule, for typical plants.

Exercise 40 Digital controller by emulation for DC motor (speed)

Design a digital controller by emulation (Tustin rule) for a DC motor speed with plant:

$$G(s) = \frac{1}{(s+0.5)(s+5)}$$

With zero steady-state error due to step, percentage overshoot less than 10% ($P_O < 0.1$) and settling time of approximately 1.4 s ($T_s = 1.4\ s$) – if possible. Select a suitable sample time.

.

Exercise 41 Digital controller by emulation for DC motor (position)

Design a digital controller by emulation (Tustin rule) for a DC motor position with plant:

$$G(s)=\frac{1}{s(s+0.5)(s+5)}$$

With percentage overshoot less than 10% ($P_O<0.1$) and settling time of approximately 1.4 s ($T_s=1.4\,s$).

.

Exercise 42 Digital controller by emulation for speed control plant

Design a digital controller by emulation (Tustin rule) for the same speed control system used in Exercise 36:

$$G(s)=\frac{1}{(s+1)(s+3)}$$

With time constant of less than 0.3 s ($\tau<0.3\,s$), dominant pole damping ratio of at least 0.7, and zero steady state error due to step.

.

Exercise 43 Digital DC motor controller by emulation with Tustin

Consider the DC motor modelled in Exercise 3:

$$G(s)=\frac{K_m}{\tau_m s+1}$$

with $\tau_m=0.1\,\mathrm{s}$, $K_m=1$.

A.

Design a continuous time controller with the following requirements for the closed-loop system: $\zeta=0.7$, $\omega_n=25\ \mathrm{rad/s}$.

B.

Design a digital controller by emulation (Tustin), using a sample frequency that is 6 times the crossover frequency. Compare the poles and the step response with the analogue controller.

C.

Repeat the design using a sample frequency that is 20 times the crossover frequency. Compare the poles and the step response with the analogue controller.

.

MATLAB functions: tf, c2d, bode, feedback, pzmap, step

[1] 6.1

[2] 6.3.1, 6.3.3

7.4.3 Tustin with pre-warping

7.4.3.1 Distortion

The Tustin rule maps the entire semiplane $\mathrm{Re}(s)<0$ into the unit circle $|z|<1$ and hence the imaginary axis $j\omega$ into the circumference $z=1$. While tustin rule preserves stability or instability, a large distortion in gain (or phase) might be introduced.
To show this distortion, we assess the frequency response of a generic analogue transfer function $H_a(s)$ and its digital counterpart obtained with the Tustin rule $H_d(z)$:

$$H_d(z)\Big|_{z=e^{j\omega T}} = H_a(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}\Bigg|_{z=e^{j\omega T}} = H_a\left(\frac{2}{T}\frac{e^{j\omega T}-1}{e^{j\omega T}+1}\right) =$$

$$= H_a\left(\frac{2}{T}\frac{\left(e^{j\omega T}-1\right)\times e^{-j\omega\frac{T}{2}}}{\left(e^{j\omega T}+1\right)\times e^{-j\omega\frac{T}{2}}}\right) = H_a\left(\frac{2}{T}\frac{e^{j\omega\frac{T}{2}}-e^{-j\omega\frac{T}{2}}}{e^{j\omega\frac{T}{2}}+e^{-j\omega\frac{T}{2}}}\right)$$

Remember that:

$$\frac{1}{2}\left[e^{j\theta}+e^{-j\theta}\right]=\cos\theta$$

$$\frac{1}{2j}\left[e^{j\theta}-e^{-j\theta}\right]=\sin\theta$$

$$H_d\left(e^{j\omega T}\right) = H_a\left(\frac{2}{T}\frac{e^{j\omega\frac{T}{2}}-e^{-j\omega\frac{T}{2}}}{e^{j\omega\frac{T}{2}}+e^{-j\omega\frac{T}{2}}}\right) = H_a\left(j\frac{2}{T}\tan\left(\frac{\omega T}{2}\right)\right)$$

This means that the Tustin rule maps every frequency in the analogue filter $\omega_a = \frac{2}{T}\tan\frac{\omega_d T}{2}$ into ω_d. This means the discrete filter will behave at ω_d in the same way as the analogue at ω_a. Gain and phase shift of the discrete filter at ω_d are the same as those of the analogue filter at ω_a.
We can also assess the Tustin transformation on the analogue folding frequency $\omega_s/2=\pi/T$:

$$H_d\left(e^{j\omega_s T/2}\right) = H_a\left(j\frac{2}{T}\tan\left(\frac{\omega_s T}{4}\right)\right) = H_a\left(j\frac{2}{T}\tan\left(\frac{\pi}{2}\right)\right) = H_a(j\infty)$$

This means that infinitely high frequencies in the analogue domain are transformed into frequencies $\omega_s/2=\pi/T$ in the digital domain. This is the highest possible frequency in the digital domain with sample time T, and it therefore implies that the Tustin rule does not introduce ambiguity due to aliasing. The downside is that distortion must happen, as infinite bandwidth is compressed in a finite interval:

$$\omega_a \in [0,\infty) \leftrightarrow \omega_d \in \left[0, \frac{\omega_s}{2}\right)$$

If the sampling time is small enough, then:

$$\tan\left(\frac{\omega T}{2}\right) \approx \frac{\omega T}{2}$$

and the distortion can be neglected. However, as discussed before, it is not always possible to obtain a small-enough sample time due to hardware limitations.
The next subsection will show a method to eliminate the distortion at one specific frequency.

📖 [2] 6.3.3

7.4.3.2 Pre-warping

It is possible to "pre-warp" the analogue transfer function and consider the following:

$$H(s') = H\left(s\frac{\omega_d}{\omega_a}\right) = H\left(s\frac{\omega_d}{\frac{2}{T}\tan\frac{\omega_d T}{2}}\right)$$

The Tustin rule applied to this "pre-warped" transfer function will now map analogue frequency ω_d into digital frequency ω_d and therefore preserve gain and phase shift at ω_d.

Combining the Tustin rule and the pre-warping we get:

$$H(z) = H\left(s\frac{\omega_d}{\omega_a}\right)\Bigg|_{s=\frac{2}{T}\frac{z-1}{z+1}} = H(s')\Big|_{s'=\frac{\omega_d}{\omega_a}\frac{2}{T}\frac{z-1}{z+1}} = H(s')\Big|_{s'=\frac{\omega_d}{\frac{2}{T}\tan\frac{\omega_d T}{2}}\frac{2}{T}\frac{z-1}{z+1}} = H(s')\Big|_{s'=\frac{\omega_d}{\tan\frac{\omega_d T}{2}}\frac{z-1}{z+1}}$$

We can apply this transformation to the original analogue filter $H(s)$, knowing that we can choose a frequency ω_d which will be preserved, i.e. gain and phase shift of the digital filter at ω_d will match those of the analogue filter at the same frequency ω_d

Summarizing the Tustin rule with pre-warping, we can write:

$$H_p(z) = H(s')\Big|_{s'=\frac{\omega_d}{\tan\left(\frac{\omega_d T}{2}\right)}\frac{z-1}{z+1}}$$

At $\omega = \omega_d$, $H_p(z)$ (digital) has the same gain as $H(s)$ (analogue).

Note also that the pre-warping vanishes for $\omega_d \to 0$, so frequency $\omega \to 0$ in analogue is always mapped to $\omega \to 0$ in digital.

The obvious choice of ω_d for a low-pass filter would be the cut-off frequency ω_c (but it does not have to be). See the following example.

Exercise 44 Tustin with pre-warping on a LP filter

Consider a continuous/analogue first-order low-pass filter with cut-off frequency (actually a pulsation) ω_c:

$$H(s) = \frac{\omega_c}{s+\omega_c}$$

Design a digital equivalent of it, using Tustin with pre-warping, such that the gain at the cut-off frequency is preserved.

Verify the result for $\omega_c = 10\,rad/s, T = 0.2\,s$.

.

Exercise 45 Tustin with pre-warping on a band-pass filter

Consider a second-order band-pass filter with a gain of 0 dB at $\omega_f = 100$.

Design the discrete equivalent of it using the Tustin rule for the following cases and compare:

A. T = 0.01 s

B. T = 0.03 s

C. T = 0.03 s and pre-warping using the passing frequency as frequency to be preserved in the transformation.

Hint: a band-pass filter can be created combining a low-pass and a high-pass filter.

.

[2] 6.3.3, 5.4.2

http://home.ku.edu.tr/~alperdogan/elec304/SteadyStateErrors.pdf

7.4.4 Pole-zero matching

We have seen the correspondence of poles between s-plane and z-plane follows:

$$z = e^{sT}$$

The pole-zero (PZ) matching technique relies on this correspondence for both poles and zeros. A set of heuristic rules locates poles, zeros and gain of a z-transform that describes a discrete equivalent transfer function that approximates a given $H(s)$.

The discrete equivalent can be constructed following these four steps:

1. All poles of $H(s)$ are mapped following:

$$z = e^{sT}$$

For example, for each pole of H(s) at $s = a + jb$, H(z) will have a pole at $z = re^{j\theta}$, where:

$$r = e^{aT}$$
$$\theta = bT$$

2. All <u>finite</u> zeros are mapped with the same rule: $z = e^{sT}$
3. Every zero at $s = \infty$ is mapped to $z = -1$, and therefore adds a term $(z+1)$ to the numerator of H(z).

To determine the number of zeros at infinity, one calculates the limit:

$$\lim_{s \to \infty} H(s)$$

If this limit goes to zero, then there is at least one zero at infinity. The degree of the polynomial remaining after the limit determines the multiplicity of the zeros at infinity. In practice, the number of zeros at infinity is obtained by the difference between the highest degree of the denominator (number of poles) and the highest degree of the numerator (number of zeros). For example,

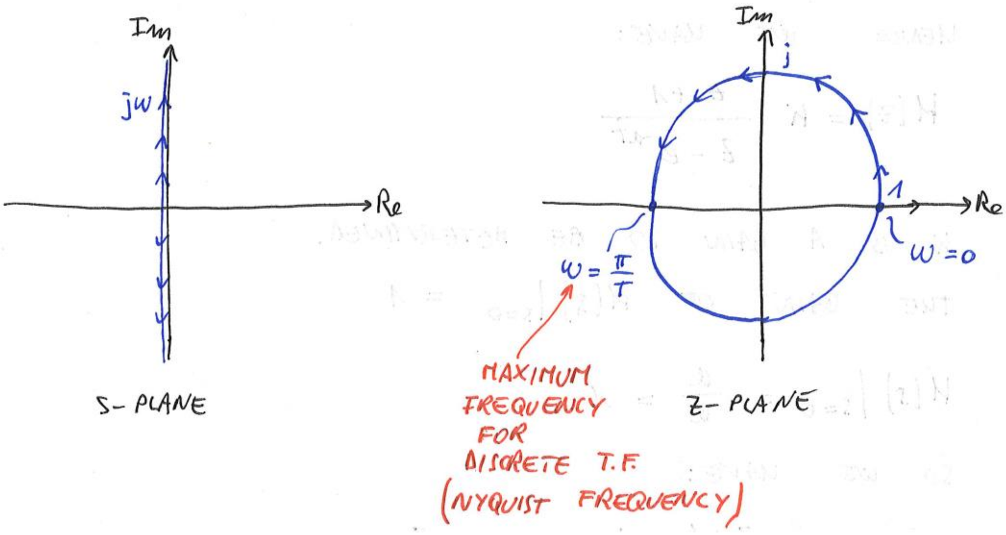
$$H(s) = \frac{s^2 + s + 1}{s^2 + 3}; \quad H(s) = \frac{s^2 + s + 1}{s + 3}$$

Have no zeros at infinity, while:

$$H(s) = \frac{s^2 + s + 1}{s^3 + 3}; \quad H(s) = \frac{s^2 + s + 1}{s^4 + 3}$$

Have one and two zeros at infinity respectively.

The idea behind this rule is that $s = \infty$ represents the maximum frequency of the continuous transfer function, and this is at $z = -1$ for the discrete transfer function, limited by the digital filter.



4. The gain of the digital transfer function is selected such to match the gain of $H(s)$ at a specific frequency. For a low-pass filter and in most applications the critical frequency is $s=0$ hence a gain is selected such that: $H(s)\big|_{s=0} = H(z)\big|_{z=1}$ (DC gain preserved). For high-pass filters, one can decide to preserve the gain at high frequencies, hence $H(s)\big|_{s=\infty} = H(z)\big|_{z=-1}$.

Because of the ease of preserving the gain at a specific frequency, this technique of design by emulation is particularly useful for filters.

Exercise 46 Discrete equivalent of LP filter by PZ matching

Compute discrete equivalent of

$$H(s) = \frac{a}{s+a}$$

By zero-pole matching. Match the steady-state gain.

.

[2] 6.3.2

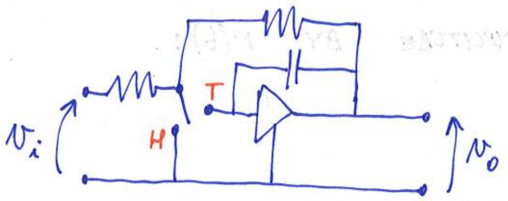
MATLAB functions: c2d

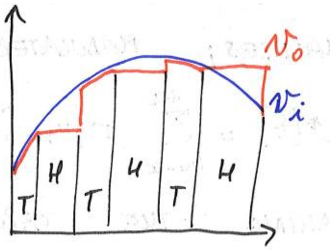
8 Sampled-data systems

We have seen digital systems, however most of real systems are continuous and the digital signals come from sampling of these continuous systems/signals. Systems with both continuous and discrete signals are called sampled-data systems.
We now study the effect of sampling on continuous signals.

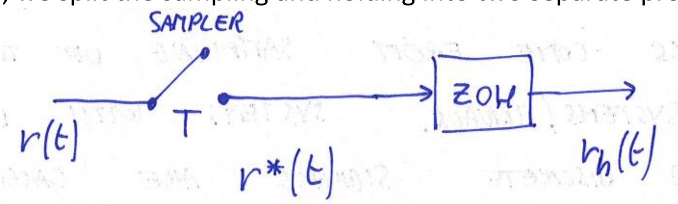
8.1 Sample and hold

A commonly used A/D converter is the sample and hold. This can be seen as a sampler followed by a ZOH.
The sample and hold can be realized in the following way.

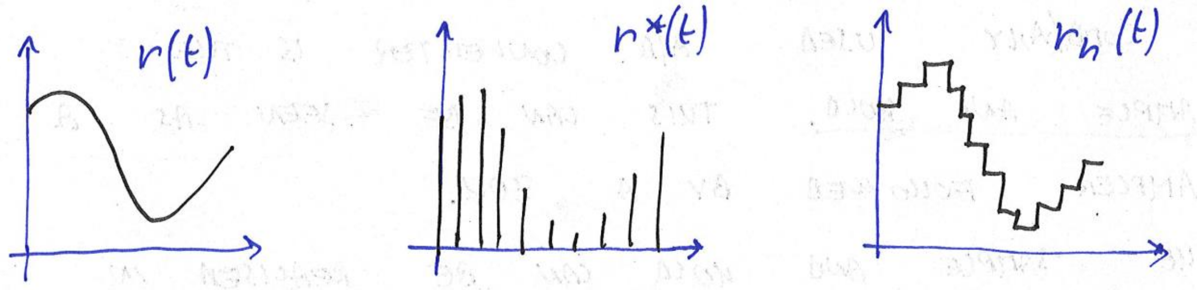




For studying purposes, we split the sampling and holding into two separate processes.



Sampling is represented by <u>impulse modulation</u> while holding is done through a linear filter. So we can apply the Laplace transform for signal analysis.



8.1.1 Sampling

r^* can be represented as a train of impulses, modulated in amplitude by $r(t)$:

$$r^*(t) = \sum_{k=-\infty}^{+\infty} r(t)\delta(t-kT)$$

Taking the Laplace transform:

$$R^*(s) = \mathrm{L}\{r^*(t)\} = \int_{-\infty}^{+\infty} r^*(\tau)e^{-s\tau}d\tau$$

$$= \int_{-\infty}^{+\infty} \sum r(\tau)\delta(\tau - kT)e^{-sT}d\tau$$

Because $\delta \neq 0$ only at $t=0$ and $\int \delta(\tau)d\tau = 1$,

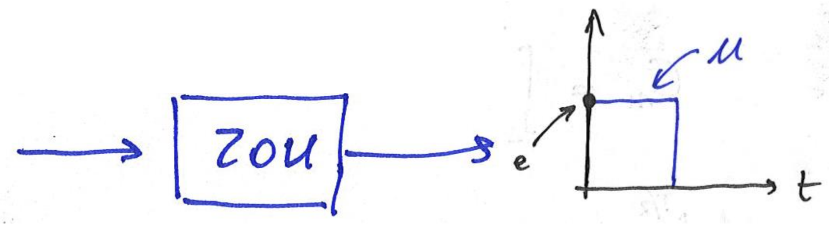
$$R^*(s) = \sum_{-\infty}^{+\infty} r(kT)e^{-skT}$$

8.1.2 Holding

The holding process through a zero-order hold can be studied in the same way. The held signal is:

$$r_h = r(kT) \qquad \text{for} \quad kT \leq t < kT + T$$

The transfer function of the ZOH, ZOH(s) can be found computing the Laplace transform of its impulse response.



$$e = \delta(t) \qquad\qquad u(t) = 1(t) - 1(t-T)$$

$$ZOH(s) = \mathrm{L}\{u(t)\} = \mathrm{L}\{1(t)\} - \mathrm{L}\{1(t-T)\}$$

$$= \frac{1-e^{-sT}}{s}$$

8.2 Spectrum of a sampled signal

We want to compare the spectrum of the sampled signal r^* to that of the original signal. We use Fourier analysis.

Let us consider the sampled signal:

$$r^*(t) = \sum_{-\infty}^{+\infty} r(t)\delta(t-kT) = r(t)\sum_{-\infty}^{+\infty}\delta(t-kT) \tag{8.1}$$

The last summation is periodic of period *T*. Therefore, $\sum \delta(t-kT)$ can be expressed as a Fourier series.

Remember Fourier series, for a periodic function *f* of period *T*:

$$f(t) = \sum_{n=-\infty}^{+\infty} C_n e^{j\frac{2\pi n}{T}t}$$

$$\text{with: } C_n = \frac{1}{T}\int_{-T/2}^{+T/2} f(t)e^{-jn\frac{2\pi}{T}t}dt$$

Hence:

$$\sum_{k=-\infty}^{+\infty}\delta(t-kT) = \sum_{n=-\infty}^{+\infty} C_n e^{-jn\frac{2\pi t}{T}}dt$$

$$\text{with: } C_n = \frac{1}{T}\int_{-T/2}^{+T/2}\sum_{k=-\infty}^{+\infty}\delta(t-kT)e^{-jn\frac{2\pi t}{T}}dt$$

The only non-zero term in the integral is for $k=0$!

$$C_n = \frac{1}{T}\int_{-T/2}^{T/2}\sum_{n=-\infty}^{\infty}\delta(t)e^{-jn\frac{2\pi t}{T}}dt = \frac{1}{T}$$

Because $\int_{-T/2}^{T/2}\sum_{n=-\infty}^{\infty}\delta(t)e^{-jn\frac{2\pi t}{T}}dt = 1$

Substituting in $r^*(t)$, Eq. (8.1):

$$r^*(t) = r(t)\sum_{n=-\infty}^{+\infty}\frac{1}{T}e^{j\frac{2\pi n}{T}t}$$

Defining the sample pulsation $\omega_s \triangleq \frac{2\pi}{T}$

$$r^*(t) = r(t)\sum_{n=-\infty}^{+\infty}\frac{1}{T}e^{jn\omega_s t}$$

We now take the Laplace transform:

$$\begin{aligned}
R^*(s) = \mathrm{L}\{r^*(t)\} &= \int_{-\infty}^{+\infty} r(t)\frac{1}{T}\sum_n e^{jn\omega_s t}e^{-st}dt \\
&= \frac{1}{T}\sum_n\int_{-\infty}^{+\infty} r(t)e^{jn\omega_s t}e^{-st}dt \\
&= \frac{1}{T}\sum_n\int_{-\infty}^{+\infty} r(t)e^{-(s-jn\omega_s)t}dt \\
&= \frac{1}{T}\sum_n\int_{-\infty}^{+\infty} r(t)e^{-s't}dt
\end{aligned}$$

Let us define $s - jn\omega_s \triangleq s'$; we recognise that the previous integrand is equivalent to the Laplace transform of $r(t)$, but in the variable s' instead of s:

$$R^*(s) = \frac{1}{T}\sum_{n=-\infty}^{+\infty}\mathrm{L}\{r(t)\}\Big|_{s'=s-j\omega_s n}$$

$$R^*(s) = \frac{1}{T}\sum_{n=-\infty}^{+\infty}R(s - jn\omega_s) \tag{8.2}$$

This shows that the spectrum of R* is composed by an infinite train of sidebands, in addition to the standard signal R, which are given by all elements in the summation for which $n \neq 0$.
Considering an arbitrary spectrum for *R* and in particular its magnitude $|R|$ we have:

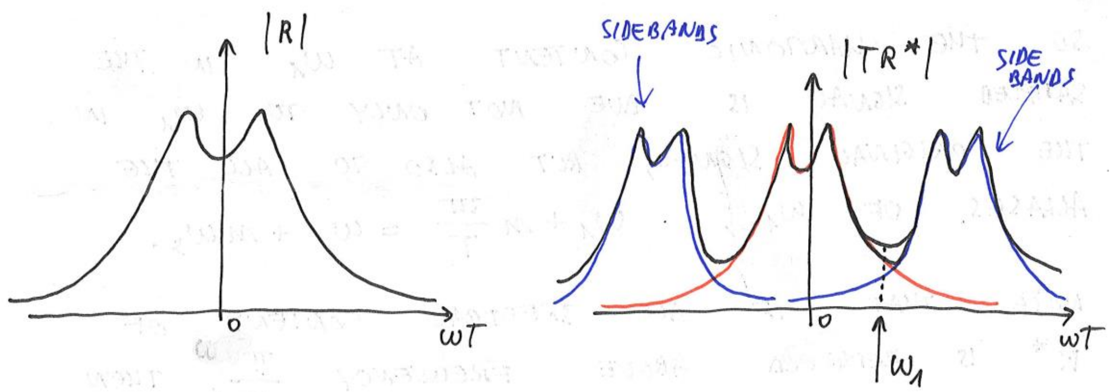


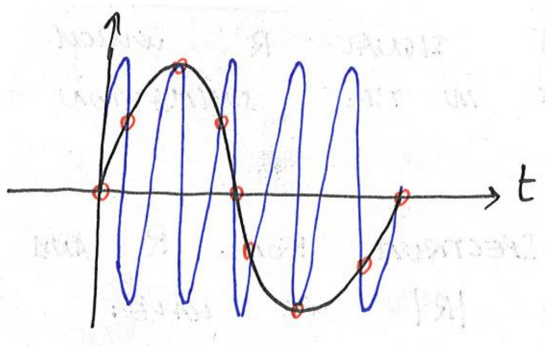
Figure 6. Spectrum of a generic signal, and counterpart after sampling

The amplitude at frequency ω_1 of $|TR^*|$ is the sum of infinite components.

The larger contribution to the amplitude comes from $R(j\omega_1)$, i.e. the main band. A smaller contribution comes from $R\left(j\omega_1 - j\frac{2\pi}{T}\right)$, that is the first sideband from Eq. (8.2) for $n=1$. In the same way, there are other infinite sidebands than provide smaller and smaller contributions to the amplitude of $R^*(j\omega_1)$.

The copied sideband with a frequency shift $\omega_n = \omega_1 - \frac{2\pi}{T}n,\ \forall n \in \mathbb{Z}$ is called an *alias* of $R^*(j\omega_1)$ and this phenomenon is called *aliasing*.

In time domain, this means that there are infinite sinusoids at various frequencies that produce the same samples:



So the harmonic content at ω_1 in the sampled signal is due not only to ω_1 in the original signal, but also to all the aliases of R^*, namely $R^*\left(\omega_1 - \frac{2\pi}{T}n\right),\ \forall n \in \mathbb{Z}$.

[2] 2.9

8.3 Anti-aliasing

How can we get rid of this phenomenon? Note that if all spectral content of R^* is removed above frequency $\frac{\omega_s}{2} = \frac{\pi}{T}$, then sampling does not introduce aliasing, because each alias is shifted by twice this frequency, and therefore aliases cannot overlap:

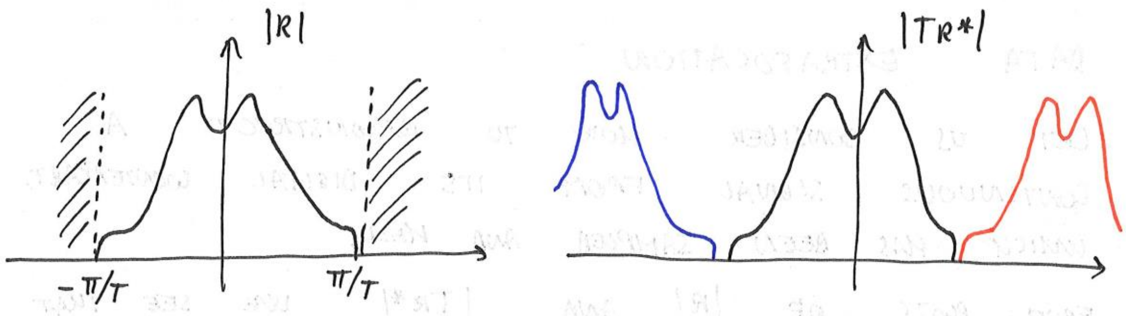


Figure 7. Effect of anti-aliasing filter on continuous signal, and full spectrum after sampling.

A spectrum such as the one plotted, with cut-off at $\frac{\omega_s}{2} = \frac{\pi}{T}$, can be obtained applying a low-pass (or *anti-aliasing*) filter to R, before sampling.

Sampling theorem (Nyquist): A continuous signal can be uniquely reconstructed from its samples if the sampling frequency $\omega_s = \dfrac{2\pi}{T}$ is at least twice the highest frequency in the signal, which shall therefore limited to $\omega_N = \dfrac{\pi}{T}$ (Nyquist frequency).

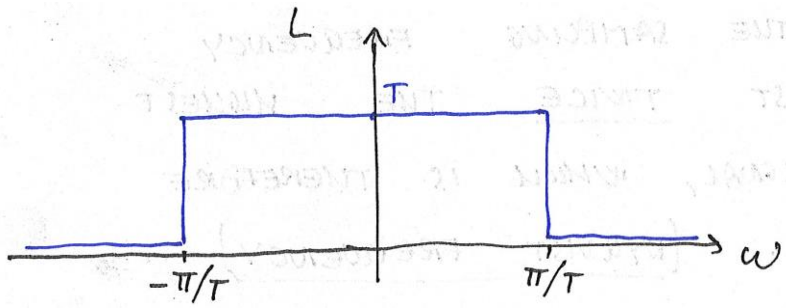
[2] 2.9

8.4 Data extrapolation

Data extrapolation is the process of reconstructing a continuous signal starting from its digital counterparts, i.e. samples $r^*(t)$.

From plots of $|R|$ and $|TR^*|$, we note that the spectrum of R is entirely contained in the lower band of $|R^*|$ (provided that there is no alias). So if we pass R^* through an appropriate low-pass filter and multiply by T, we have R.

Consider an ideal low-pass filter $L(s)$, its transfer function has spectrum:



Then the original signal can be reconstructed by applying the filter: $R = L(s)R^*$. We will apply the convolution to obtain $r(t)$, so we need to obtain time history of $l(t)$ ($r^*(t)$ are the samples).

Remember the (continuous-time) convolution:

$$U(s) = G(s)E(s)$$

$$u(t) = \int_{-\infty}^{+\infty} g(\tau)e(t-\tau)d\tau \qquad (8.3)$$

The filter time history $l(t)$ can be found by anti-transforming its transfer function $L(s)$ with Laplace.

Remember the Laplace transform and its inverse:

$$F(s) = \mathrm{L}\{f(t)\} = \int_0^{+\infty} f(t)e^{-st}dt$$

$$f(t) = \mathrm{L}^{-1}\{F(s)\} = \frac{1}{2\pi j}\int_{\sigma-j\infty}^{\sigma+j\infty} F(s)e^{st}ds$$

In the specific case we can take $\sigma = 0$ and $s \equiv j\omega$:

$$l(t) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} L(j\omega)e^{j\omega t}d\omega \qquad (8.4)$$

In our case, $L(j\omega)$ is the spectrum plotted above, or in formulas:

$$L(j\omega)=\begin{cases}T, & \text{for } -\dfrac{\pi}{T}<\omega<\dfrac{\pi}{T}\\ 0, & \text{elsewhere}\end{cases}$$

Inserting into Eq. (8.4):

$$\begin{aligned}l(t)&=\frac{1}{2\pi}\int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}}Te^{j\omega t}d\omega=\frac{T}{2\pi}\frac{1}{jt}e^{j\omega t}\Big|_{-\frac{\pi}{T}}^{+\frac{\pi}{T}}=\\ &=\frac{T}{2\pi jt}\left(e^{j\frac{\pi t}{T}}-e^{-j\frac{\pi t}{T}}\right)=\frac{\sin\left(\frac{\pi t}{T}\right)}{\frac{\pi t}{T}}\triangleq\operatorname{sinc}\left(\frac{\pi t}{T}\right)\end{aligned}\tag{8.5}$$

Now we have $l(t)$ and $r^*(t)$. Since $R=L(j\omega)R^*$, we can use the convolution formula (8.3) for obtaining the time history of R, that is the original signal $r(t)$:

$$r(t)=\int_{-\infty}^{+\infty}r^*(\tau)l(t-\tau)d\tau$$

Inserting Eq. (8.1) for $r^*(\tau)$ and Eq. (8.5) for $l(t-\tau)$:

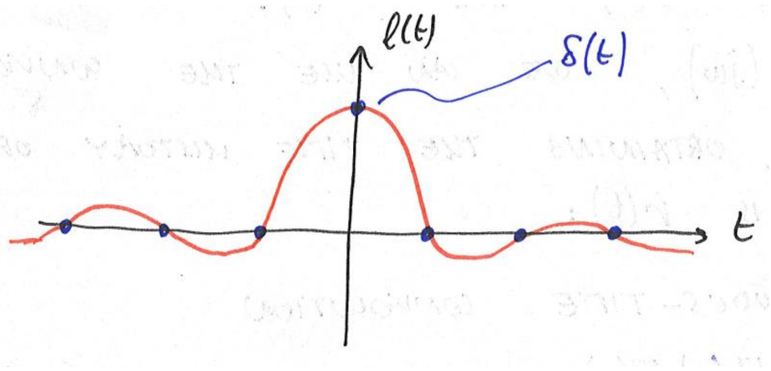
$$r(t)=\int_{-\infty}^{+\infty}\sum_{k=-\infty}^{+\infty}r(\tau)\delta(\tau-kT)\operatorname{sinc}\left(\frac{\pi(t-\tau)}{T}\right)d\tau$$

The train of impulses δ integrates to 1 only at $\tau=kT$ hence:

$$r(t)=\sum_{-\infty}^{+\infty}r(kT)\operatorname{sinc}\left(\frac{\pi(t-kT)}{T}\right)\tag{8.6}$$

This tells us how to build $r(t)$ from its samples $r(kT)$. The $\operatorname{sinc}(.)$ functions are used to interpolate $r(kT)$ in between the samples, to generate $r(t)$ at any time point.

We can plot the time history of $l(t)$ from Eq. (8.5):



This is also the response to the impulse of the filter $L(j\omega)$.

This shows that this ideal low-pass filter $L(j\omega)$ is non-causal, because $l(t)\neq 0$ for $t<0$, when impulse only occurs at $t=0$. This means that it cannot be implemented in practice.
If the interpolated signal is not needed in real time (as samples arrive) then a delay can be added, to save a number of samples before attempting interpolation.
In feedback systems a large delay has serious impact on stability so this filter cannot be used. This is why we use the polynomial holds, and the ZOH in particular. We found its (continuous) transfer function to be:

$$ZOH(s)=\frac{1-e^{-sT}}{s}$$

We can calculate gain and phase shift of the ZOH, to understand how it will condition the digital (sampled) signal:

$$ZOH(j\omega)=\frac{1-e^{-j\omega T}}{j\omega}=e^{-j\omega\frac{T}{2}}\left[\frac{e^{j\omega\frac{T}{2}}-e^{-j\omega\frac{T}{2}}}{j\omega}\right]\frac{2}{2}=$$

$$=e^{-j\omega\frac{T}{2}}\left[\frac{e^{j\omega\frac{T}{2}}-e^{-j\omega\frac{T}{2}}}{2j}\right]\frac{2}{\omega}=$$

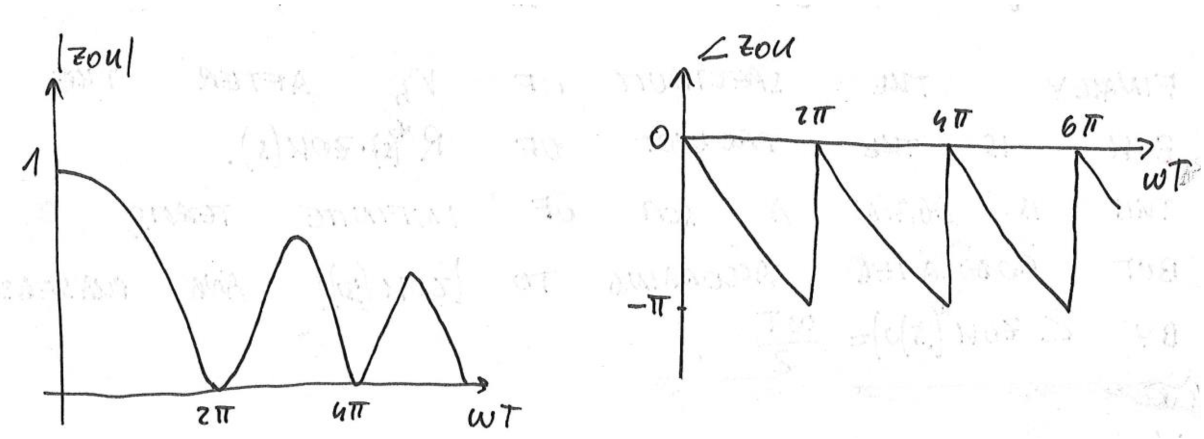
$$=e^{-j\omega\frac{T}{2}}\sin\left(\frac{\omega T}{2}\right)\frac{2}{\omega}\frac{T}{T}=Te^{-j\omega\frac{T}{2}}\frac{\sin\left(\frac{\omega T}{2}\right)}{\frac{\omega T}{2}}=$$

$$=Te^{-j\omega\frac{T}{2}}\operatorname{sinc}\frac{\omega T}{2}$$

This expression is handy to identify phase and magnitude of ZOH:

$$|ZOH(j\omega)|=T\left|\operatorname{sinc}\frac{\omega T}{2}\right|$$

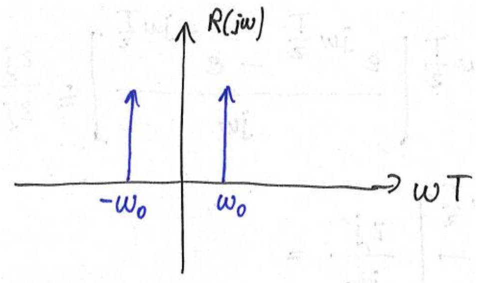
$$\angle ZOH(j\omega)=-\frac{\omega T}{2}$$



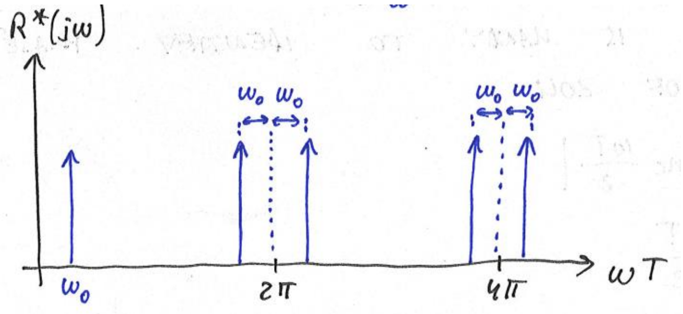
Assume we feed the ZOH with a sinusoidal input:

$$r(t)=A\cos(\omega_0 t+\varphi)$$

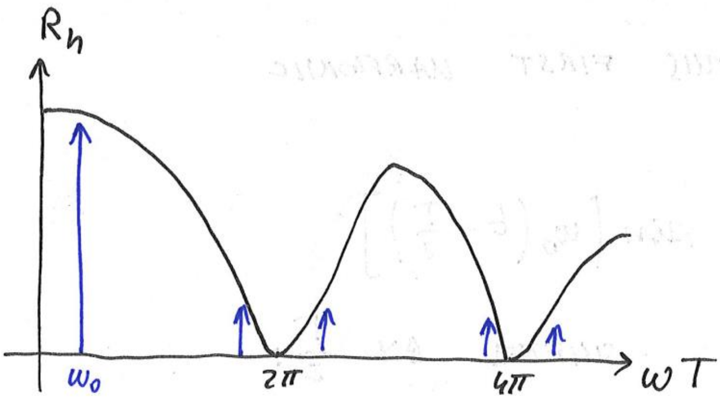
The spectrum of the sinusoidal input is:



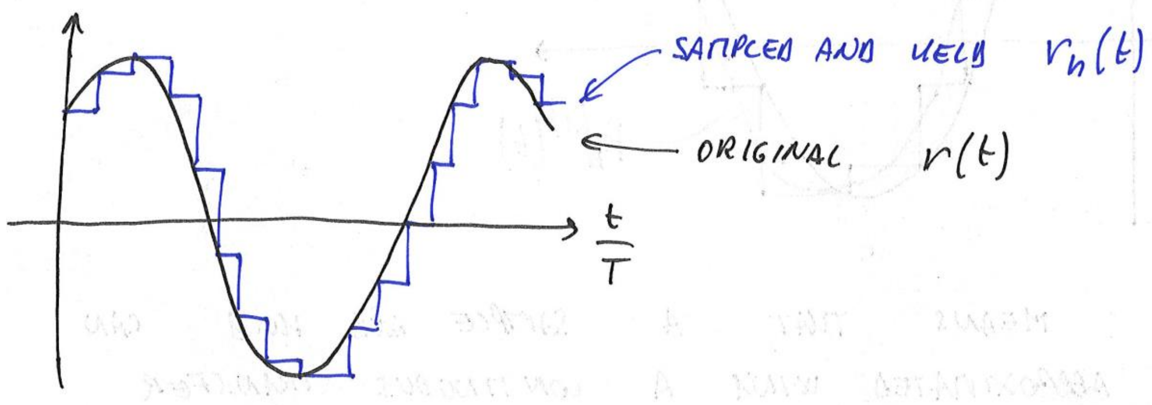
We saw that the effect of sampling is to create $R^*(\omega T)$ which is composed by an unlimited sum of copies of $R(\omega T)$ shifted by $m\dfrac{2\pi}{T}$ in frequency and each multiplied by $\dfrac{1}{T}$ (see Eq. (8.2))



Finally the spectrum of r_h after the ZOH is the product of $R^*(s)ZOH(s)$. This is again a sum of infinite terms, but modulated according to $|ZOH(j\omega)|$ (plotted above) and phase-shifted by $\angle ZOH(j\omega)=\dfrac{\omega T}{2}$:



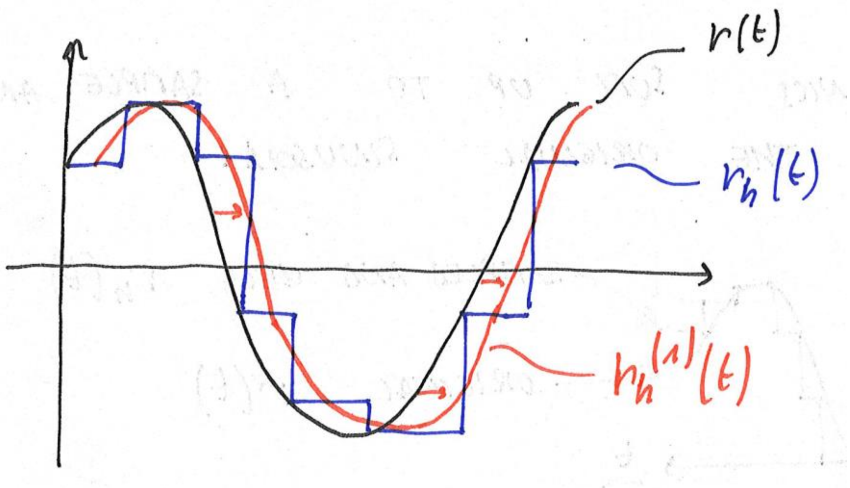
All these harmonics sum up to a sample and hold signal of the original sinusoid:



This shows, both in time and frequency domain that the output sampled and held signal $r_h(t)$ is not a sinusoid, and therefore the sample and hold is not a linear system.
However, we can extract a sinusoidal approximation to $r(t)$ from $r_h(t)$ by considering only the first harmonic. This has amplitude $A\,\mathrm{sinc}\dfrac{\omega T}{2}$ and phase shift of $\dfrac{\omega_0 T}{2}$
In time domain this first harmonic corresponds to:

$$r_h(t) = A\operatorname{sinc}\frac{\omega_0 T}{2}\cos\left[\omega_o\left(t-\frac{T}{2}\right)+\varphi\right]$$

This is a sinusoid delayed by $\frac{T}{2}$ in time:



This means that a sample and hold (ZOH) could be approximated with a continuous transfer function corresponding to a delay of $\frac{T}{2}$.

When designing in the continuous domain, with the intention of implementing the controller digitally, then the presence of a ZOH can be (approximately) accounted for by introducing this delay. The simplest first-order linear approximation that can be used to simulate the delay in a continuous system is:

$$G(s) = \frac{2/T}{s+2/T}$$

Exercise 47 Reconstruction of a signal from its samples

Consider the following continuous time sequence:

$$f(t) = \sin(t) + \sin(1.5t + \pi/3)$$

And its corresponding samples with $T = 1$:

$$f_k = \sin(k) + \sin(1.5k + \pi/3)$$

Write a machine code to reconstruct $f(t)$ (and plot it) in the time window $t \in [-10, 5]$ using a finite subset of its samples f_k, for example $k = -5, \dots -1$ and $k \in -10, \dots, -1$. Graphically compare the reconstruction with the original function, and show that the more samples are used, the better the reconstruction becomes.

.

📖 [1] 3.2
📖 [2] 3.3

Appendix A: Table of Laplace and z-transforms

Laplace and Z Transforms for Causal Functions

f(t) $t \geq 0$ *(causal)*	***F(s)***	***F(z)*** *(t=kT, T = Sample Time, k = Index)*
$\delta\ (t)$	1	$1 = z^{-0}$
$\delta\ (t-kT)$	e^{-kTs}	z^{-k}
$u(t)$	$\frac{1}{s}$	$\frac{z}{z-1}$
t	$\frac{1}{s^2}$	$\frac{T \cdot z}{(z-1)^2}$
$\frac{t^k}{k!}$	$\left(\frac{1}{s}\right)^{k+1}$	$\underset{\alpha \to 0}{Limit} \frac{(-1)^k}{k!} \cdot \frac{\partial^k}{\partial \alpha^k}\left(\frac{z}{z-e^{-\alpha T}}\right)$
e^{-at}	$\frac{1}{s+a}$	$\frac{z}{z-e^{-aT}}$
$t \cdot e^{-at}$	$\frac{1}{(s+a)^2}$	$\frac{T \cdot z \cdot e^{-aT}}{(z-e^{-aT})^2}$
$e^{-at} \cdot \cos(bt)$	$\frac{(s+a)}{(s+a)^2+b^2}$	$\frac{z \cdot (z-e^{-aT} \cdot \cos(bt))}{z^2 - z \cdot (2 \cdot e^{-aT} \cdot \cos(bt)) + e^{-2aT}}$
$e^{-at} \cdot \sin(bt)$	$\frac{b}{(s+a)^2+b^2}$	$\frac{z \cdot e^{-aT} \cdot \sin(bt)}{z^2 - z \cdot (2 \cdot e^{-aT} \cdot \cos(bt)) + e^{-2aT}}$

University of Glasgow | James Watt School of Engineering

References

1. Franklin, G.F., D.J. Powell, and M. Workmann, *Digital control of dynamic systems, 3rd Editon*. 2014: Ellis-Kagle press.
2. Fadali, M.S. and A. Visioli, *Digital control engineering analysis and design, 3rd Edition*. 2020: Elsevier.
3. Gopal, M., *Control Systems: Principles and Design*. 2006: McGraw-Hill.