

# Control Systems

Mathias Balling & Mads Thede

Last updated: April 4, 2024

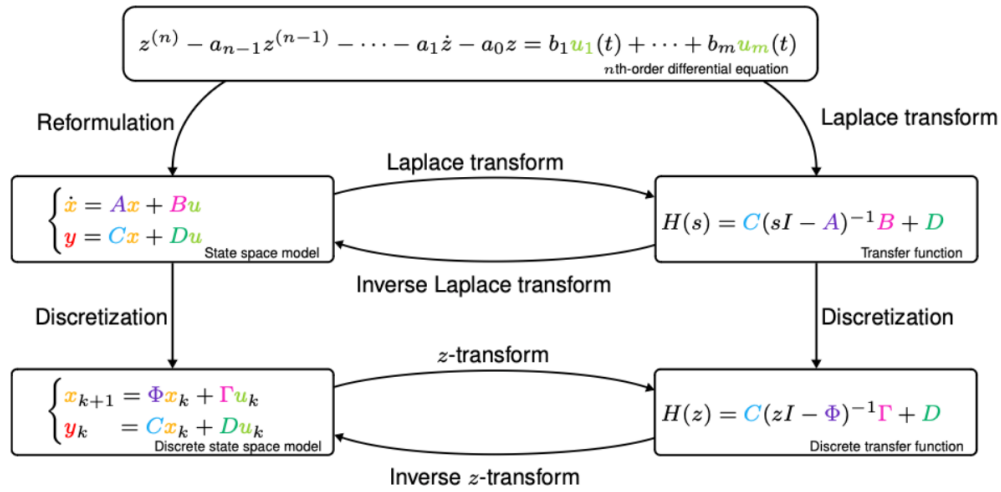
## Contents

<b>1</b>	<b>Linear Time Invariant Systems</b>	<b>3</b>
1.1	Time-Domain models . . . . .	3
1.2	Frequency-Domain models . . . . .	4
1.2.1	State space to transfer function . . . . .	4
1.2.2	Transfer function to state space . . . . .	4
1.2.3	Discrete-time transfer function . . . . .	4
1.3	Examples . . . . .	5
<b>2</b>	<b>Stability and Performance Analysis</b>	<b>6</b>
2.1	Basic System Classes . . . . .	6
2.1.1	First Order Systems . . . . .	6
2.1.2	Second Order Systems . . . . .	6
2.2	Performance specifications . . . . .	7
2.3	Poles and Zeros of Space-State Models . . . . .	8
2.4	Stability . . . . .	9
2.5	Examples . . . . .	9
<b>3</b>	<b>Introduction to Control</b>	<b>10</b>
3.1	Open Loop Control . . . . .	10
3.2	Feedback Control . . . . .	10
3.3	Steady State Tracking . . . . .	12
3.4	Cascade Control . . . . .	12
3.5	DC Motor Dynamics . . . . .	12
3.6	Examples . . . . .	12
<b>4</b>	<b>Design of PID Controllers</b>	<b>13</b>
4.1	P-controller . . . . .	13
4.2	PI-controller . . . . .	13
4.3	PD-controller . . . . .	14
4.4	PID-controller . . . . .	15
4.5	Tuning a PID Controller . . . . .	15
4.5.1	Pole placement . . . . .	15
4.5.2	Ziegler-Nichols method . . . . .	15
4.6	Examples . . . . .	17
<b>5</b>	<b>Root Locus</b>	<b>18</b>
5.1	Examples . . . . .	19
<b>6</b>	<b>The Nyquist Stability Criterion</b>	<b>20</b>
6.1	Frequency response . . . . .	20

6.2	Bode Plots . . . . .	20
6.3	Nyquist Stability Criterion . . . . .	21
6.4	Examples . . . . .	21
<b>7</b>	<b>Dynamic Compensators and Stability Margins</b>	<b>22</b>
7.1	Stability Margins . . . . .	22
7.2	Dynamic compensation . . . . .	22
7.2.1	Lead compensation . . . . .	22
7.2.2	Lag compensation . . . . .	23
7.3	Examples . . . . .	23
<b>8</b>	<b>Implementation</b>	<b>24</b>
8.1	Anti-windup . . . . .	24
8.2	Digitalization . . . . .	25
8.2.1	Emulation . . . . .	25
8.2.2	Numerical Integration Methods . . . . .	26
8.2.3	Discrete Design . . . . .	26
8.3	Examples . . . . .	26

# 1 Linear Time Invariant Systems

## Overview



## 1.1 Time-Domain models

### Linear Map

The map  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is said to be linear if for any  $x, y \in \mathbb{R}^n$  and  $\alpha \in \mathbb{R}$ , the following conditions hold

$$f(x + y) = f(x) + f(y) \quad \text{Super position}$$

$$f(ax) = \alpha f(x) \quad \text{Homogeneity}$$

The function has to go through (0,0) in 2D for it to be linear due to homogeneity.

### Time-Invariant System

Let  $\sigma : \mathbb{R} \times \mathbb{R}^m \rightarrow \mathbb{R}^p$  define the input-output behavior of a system model  $\Sigma$ . The system  $\Sigma$  is time-invariant if for any input signal  $u : \mathbb{R} \rightarrow \mathbb{R}^m$  and any delay  $\tau \in \mathbb{R}$  the following relation holds:

$$y(t - \tau) = \sigma(t, u(t - \tau))$$

for all times  $t \in \mathbb{R}$ , where  $y$  denotes the output signal of the system.

The importance is that the system does not change its behavior due to time. This can be seen as a canon firing at 8am it will not fire different compared to if you do the same at 5pm.

Two types of linear time-domain models.

Continuous-time state space models (based on differential equations)

$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

- $x \in \mathbb{R}^n$  is state. e.g. position or velocity
- $u \in \mathbb{R}^m$  is input
- $y \in \mathbb{R}^p$  is output
- $A \in \mathbb{R}^{n \times n}$  is system matrix

- $B \in \mathbb{R}^{n \times m}$  is input matrix
- $C \in \mathbb{R}^{p \times n}$  is output matrix
- $D \in \mathbb{R}^{p \times m}$  is the direct feedthrough matrix

Discrete-time state space models (based on difference equations)

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = Cx_k + Du_k$$

## 1.2 Frequency-Domain models

Transfer function:

$$G(s) = \frac{Q(s)}{P(s)}$$

where  $Q(s)$  and  $P(s)$  are polynomials in  $s$ .

- The roots of  $P(s)$  are called the **poles** of  $G(s)$
- The roots of  $Q(s)$  are called the **zeros** of  $G(s)$

### 1.2.1 State space to transfer function

Taking Laplace transforms of the system and assuming  $x_0 = 0$ :

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

yields:

$$sX(s) = AX(s) + BU(s)$$

$$Y(s) = CX(s) + DU(s)$$

Which can be rewritten as ( $I$  is the identity matrix):

$$X(s) = (sI - A)^{-1} BU(s)$$

$$Y(s) = \left( C (sI - A)^{-1} B + D \right) U(s)$$

where

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

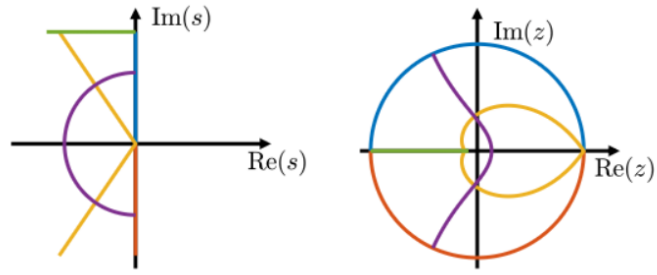
$$G(s) = C (sI - A)^{-1} B + D$$

### 1.2.2 Transfer function to state space

### 1.2.3 Discrete-time transfer function

Discretization from  $s$ -domain to  $z$ -domain can be done using:

- Matched  $z$ -transform
- Bilinear  $z$ -transform
- Impulse invariance  $z$ -transform



### 1.3 Examples

## 2 Stability and Performance Analysis

### 2.1 Basic System Classes

#### 2.1.1 First Order Systems

State-space representation of first order system:

$$\begin{aligned}\dot{x} &= -\frac{1}{\tau}x + \frac{k}{\tau}u \\ y &= x\end{aligned}$$

A first-order system has one pole and is described by:

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

Where  $k$  is the DC-gain and  $\tau$  is the time-constant. The system has a pole in  $s = -\frac{1}{\tau}$  i.e., the smaller time-constant, the faster system response.

#### 2.1.2 Second Order Systems

The transfer function of a second-order system is given by:

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

Where  $\omega_n > 0$  is the natural frequency and  $\zeta > 0$  is the damping ratio and  $k$  is the gain.

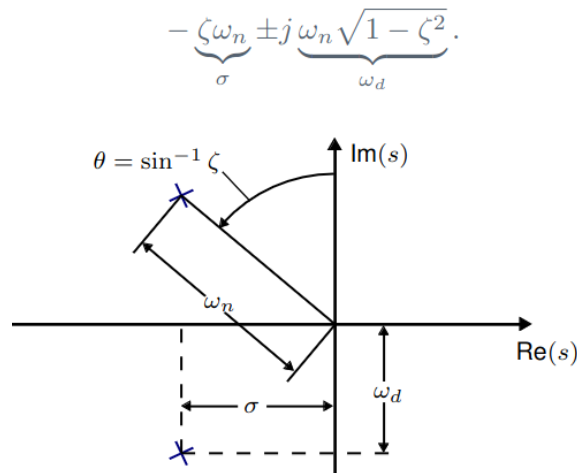
The system has two poles, which are  $s \in \mathbb{C}$  where:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$

The values of  $s$  is given by:

$$s = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$$

When  $\zeta = 1$  the system is critically damped and  $H(s)$  has a double pole in  $s = -\zeta\omega_n$ , when  $0 < \zeta < 1$  the system is underdamped and has complex poles. When  $\zeta > 1$  the system is overdamped and has real and distinct poles.



Impulse response of a underdamped second-order system:

$$h(t) = k \frac{\omega_n}{\sqrt{1 - \zeta^2}} e^{-\sigma t} \sin(\omega_d t) 1(t)$$

The step response of a underdamped second-order system:

$$y(t) = k(1 - e^{-\sigma t}(\cos(\omega_d t) + \frac{\sigma}{\omega_d} \sin(\omega_d t)))$$

Impulse response of a critically damped second order system:

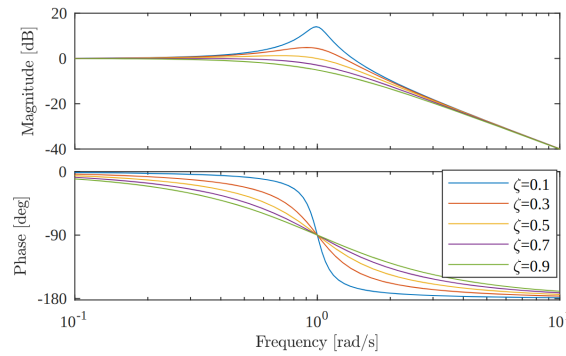
$$h(t) = k\omega_n^2 t e^{-\omega_n t}$$

Having a overdamped system with a damping ration that is greater than one leads to a slower impulse response.

The step response of a critically damped second order system:

$$y(t) = k(1 - e^{-\omega_n t} - \omega_n t e^{-\omega_n t})$$

Looking at the bode plot of a second-order system depends on the damping ratio. For every pole the magnitude of the bode plot decreases by 20dB/decade. And the phase of the bode plot decreases by 90 degrees for every pole.

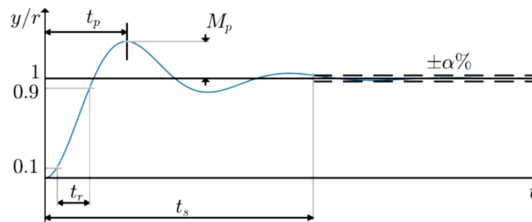


## 2.2 Performance specifications

Three different performance measurements of dynamical systems are:

1. The rise time  $t_r$  (stigetid)
2. The settling time  $t_s$  (indsvingningstid)
3. The overshoot  $M_p$  (oversving)

The rise time is the time it takes for the system to go from 10% to 90% of the final value. The settling time is the time it takes for the system to reach and stay within a certain % of the final value. The overshoot is the maximum percentage that the system goes over the final value.



The peak time is denoted  $t_p$  which is the time where the signal reaches its maximum value.

For a second-order system the rise time can be approximated as:

$$t_r = \frac{1.8}{\omega_n}$$

For a second-order system the settling time can be approximated as:

$$t_s = \frac{-\log(\alpha/100)}{\zeta\omega_n}$$

Log is ln in the equation.

The peak time can be found using:

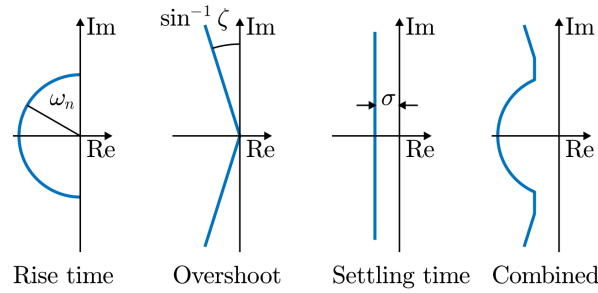
$$t_p = \frac{\pi}{\omega_d}$$

The overshoot is computed from the step response at the peak time. This gives an expression for the overshoot:

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$

for  $0 < \zeta < 1$ . The overshoot is given in percentage.

### Overview for determining performance variables



To obtain a rise time shorter than  $t_r$

$$\omega_n \geq \frac{1.8}{t_r}$$

To obtain an overshoot that is smaller than  $M_p$

$$\zeta \geq \sqrt{\frac{(\frac{\log(M_p)}{-\pi})^2}{1 + (\frac{\log(M_p)}{\pi})^2}}$$

To obtain an  $\alpha\%$ -settling time shorter than  $t_s$

$$\sigma \geq \frac{-\log(\alpha/100)}{t_s}$$

## 2.3 Poles and Zeros of Space-State Models

Using the state-space matrices:

$$G(s) = C(sI - A)^{-1}B + D$$

**Poles** can be found using the following equation:

When  $G(s) \rightarrow \infty$  the system has a pole at  $s \rightarrow p$ .

$$\det(pI - A) = 0$$

Here  $p$  is the eigenvalue of  $A$ .

**Zeros** can be found using the following equation:



When  $G(s) = 0$  the system has a zero at  $s = z$ .

$$\begin{vmatrix} A - zI & B \\ C & D \end{vmatrix} = 0$$

## 2.4 Stability

The stability of the dynamical system can be determined from the eigenvalues of  $A$  in the time domain. This is the equivalent of poles in the frequency domain.

$$\dot{x} = Ax + Bu$$

When the eigenvalues of  $A$  have a negative real part, the system is stable.

**Definition** A linear discrete-time system

$$x_{k+1} = \Phi x_k$$

The state of the next time step is given by  $\phi$  multiplied by  $x_k$ .

Where  $x_k \in \mathbb{R}^{n \times n}$  is asymptotically (asymptotisk) stable if

$$\lim_{k \rightarrow \infty} x_k = 0$$

for any  $x_0 \in \mathbb{R}^n$ .

A linear discrete-time system is stable if all the eigenvalues are inside the unit circle.

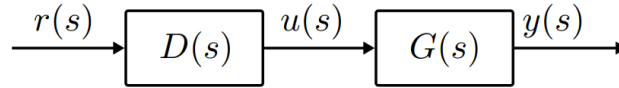
The closer the poles are to origin the more dominating the pole is in the system.

## 2.5 Examples

## 3 Introduction to Control

### 3.1 Open Loop Control

(åbensløjfe regulering)



Steady-State Value of Time Function:

Suppose that  $Y(s)$  is the Laplace transform of  $y(t)$ . Then the final value of  $y(t)$  is either:

- Unbounded. If  $Y(s)$  has any poles in the open right half-plane (unstable)
- Undefined. If  $Y(s)$  has a pole pair on the imaginary axis.
- Constant. If all poles of  $Y(s)$  are in the open left half-plane, except for one at  $s=0$

#### The Final Value Theorem (slutværdi-sætningen)

If all poles of  $sY(s)$  are in the open left half-plane, then:

$$\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} sY(s)$$

The Final Value Theorem determines the constant value that the impulse response of a stable system converges to. The theorem can also be used to determine the DC gain of a system, i.e., the output when a step input is applied to the system.

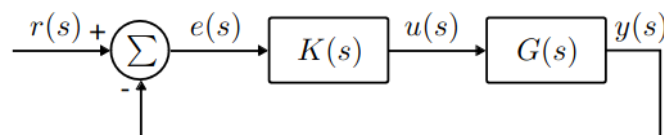
By integrating an impulse response, the step response is obtained. The step response is the integral of the impulse response, and the impulse response is the derivative of the step response.

If the system has a disturbance an open loop control system will not be able to compensate for this.

### 3.2 Feedback Control

(tilbagekobling). Tilbagekobling anvendes for at eliminere forstyrrelsesundertrykkelse.

The connection of a controller  $K(s)$  and a system (also called plant)  $G(s)$  is called a closed-loop system.



$e(s)$  is the error signal, and is the difference between the reference signal  $r(t)$  and the output signal  $y(t)$ . The error signal is used to adjust the input signal  $u(t)$  to the system. The controller  $K(s)$  is used to adjust the input signal.

The closed-loop transfer function is given by:

$$\frac{y(s)}{r(s)} = \frac{G(s)K(s)}{1 + G(s)K(s)}$$

$K(s)$  can be designed to meet certain requirements, such as stability, disturbance rejection, and sensitivity analysis.

## Stability

The closed-loop system is stable if all the closed-loop poles are in the open left half-plane.

The **loop gain (sløjfe-forstærkning)** is defined as:  $L(s) = G(s)K(s)$ . The closed-loop poles are given by:

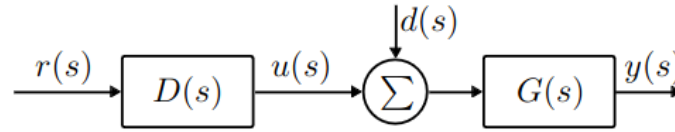
$$1 + L(s) = 0$$

The closed loop poles satisfy:

$$1 + L(s) = 0 \Rightarrow L(s) = -1$$

## Disturbance Rejection

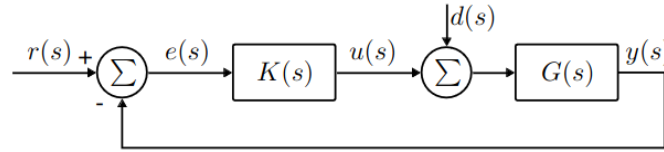
Open loop control system:



The output is given by:

$$y_m(s) = G(s)d(s) + G(s)D(s)r(s)$$

Closed loop control system:



The output is given by (superposition can be applied for deriving the expression):

$$y_m(s) = \frac{G(s)}{1 + G(s)K(s)}d(s) + \frac{G(s)K(s)}{1 + G(s)K(s)}r(s)$$

The larger the amplification of the feedback loop, the smaller the effect of the disturbance on the output.

## Sensitivity Analysis

The sensitivity  $S$  of the open-loop system is given by the ratio of  $\delta T_{ol}/T_{ol}$  to  $\delta G_0/G_0$ . For the open loop system  $S=1$ .  $\delta$  is the relative change in the transfer function.

For the closed loop the sensitivity is given by:

$$S = \frac{1}{1 + G_0 K_0}$$

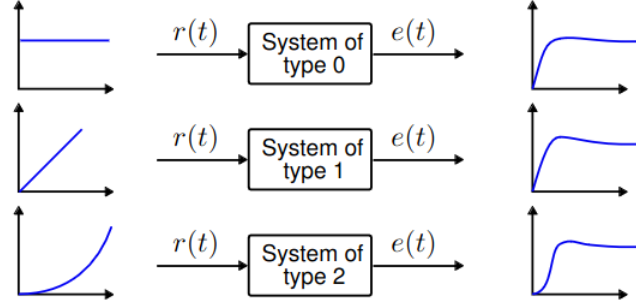
This means that a large amplification leads to a low sensitivity.

**Linearization** Taylor approximation, YES.

### 3.3 Steady State Tracking

#### Definition of system type

Stable systems can be classified according to its system type, defined to be the degree of the polynomial for which the steady-state error is a nonzero finite constant.



An expression for the tracing error is computed to determine the steady-state error. The error is given by:

$$e(s) = \frac{1}{1 + L(s)} r(s)$$

Where  $r(s)$  is the reference signal and  $L(s)$  is the loop gain.

**The amount of poles at  $s=0$  in the denominator of  $L(s)$  determines the system type.**

Final value theorem is used to determine the steady-state error when the reference signal is a polynomial  $r(t) = t^k 1(t)$ .

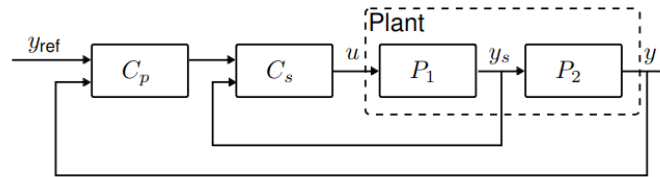
$$e_s s = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} e(s) s$$

$$\lim_{s \rightarrow 0} e(s) s = \lim_{s \rightarrow 0} s \frac{1}{1 + L(s)} r(s)$$

$$\lim_{s \rightarrow 0} s \frac{1}{1 + L(s)} \frac{1}{s^{k+1}}$$

### 3.4 Cascade Control

A cascade control uses the output from one controller as the input to another controller.



The nested control loops are called the inner loop (secondary) and the outer loop (primary). A fundamental reason for applying cascade control is to obtain better disturbance rejection and lower sensitivity to parameter variations.

### 3.5 DC Motor Dynamics

### 3.6 Examples

## 4 Design of PID Controllers

### 4.1 P-controller

The control law for the proportional controller is given by

$$u(t) = K_p e(t)$$

where  $K_p$  is the proportional gain and  $e(t)$  is the error signal.

The controller applies a control signal to the system, which depends linearly on the error signal.

To eliminate the steady state error, feedforward can be added to the P-controller.

$$u(t) = K_p e(t) + u_{ff}$$

Where the term  $u_{ff}$  is also called reset. We choose the feedforward according to the DC gain of the system, i.e.

$$u(t) = K_p e(t) + \frac{1}{G(0)} r(t)$$

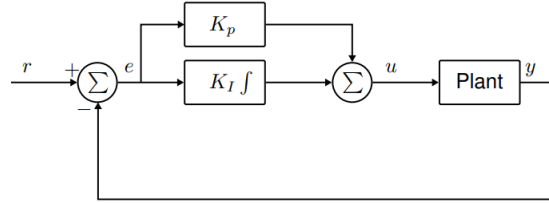
The feedforward does not affect the poles of the closed loop system.

### 4.2 PI-controller

The control law of a proportional-integral feedback controller is given by:

$$u = K_p e + K_I \int_{t_0}^t e(\tau) d\tau$$

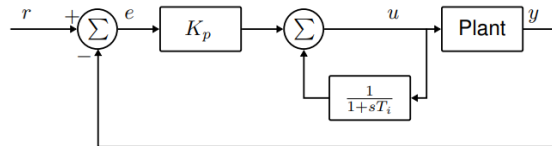
The controller adds a term, which is proportional to the integral of the error  $e$  to the P-control term.



The purpose of adding the integral term is to eliminate the steady state error of the system without the need for feed forward, i.e., integral control is less sensitive to modelling errors. Feed forward cannot eliminate modelling errors, but integral control can.

Comparing the PI-controller to the P-controller, the PI-controller has a better response to disturbances, that are unknown or cannot be measured. This is because the feedforward control can only eliminate the steady state error if the model and disturbances are known.

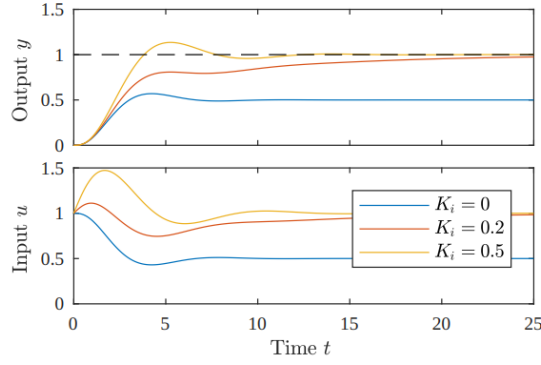
Implementation of a PI-controller is called automatic reset, and is shown in the following figure.



On the figure,  $T_i$  is the integration time (integraltid). The transfer function from  $e$  to  $u$  is given by:

$$T_{ue} = K_p \frac{1 + sT_i}{sT_i} = K_p + \frac{K_p}{sT_i}$$

On the figure below the step responses are shown for  $K_i = 0, 0.2, 0.5$  and  $K_p = 1$



From the unit step response it can be seen that the integral action removes the steady state error. A large  $K_i$  will give a fast response, but also a large overshoot. A small  $K_i$  will give a slow response, but also a small overshoot.

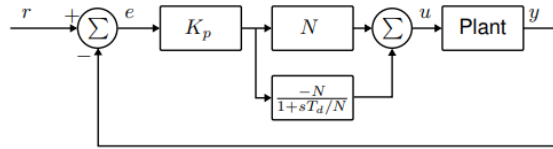
### 4.3 PD-controller

The derivative term provides an anticipatory action to the control, by doing feedback based on the trend of the error, i.e.

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} = K_p \left( e + T_d \frac{de(t)}{dt} \right)$$

Where  $k_d$  is the derivative gain and  $T_d$  is the derivative time constant and  $e_p$  is a prediction of the error ( $T_d$  forwards in time). The parenthesis in the equation on the right is a prediction of the error called  $e_p$ .

#### PD-implementation



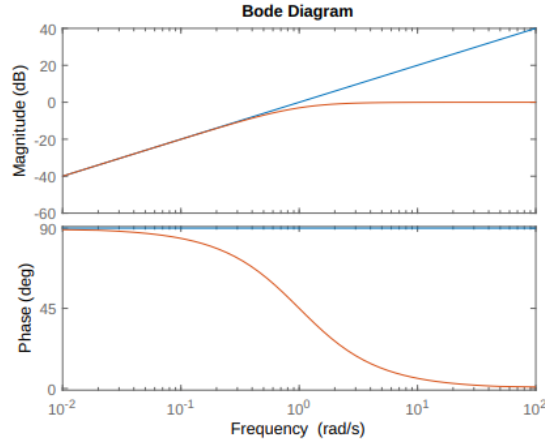
Noise is an issue for controllers that include a derivative term. Therefore, they can be implemented in a low-pass filtered version.

The transfer function of the controller is

$$T_{ue}(s) = K_p \left( N - \frac{N}{1 + sT_d/N} \right) = K_p \frac{sT_d}{1 + sT_d/N}$$

Where  $N$  is a filter constant (typical values of  $N$  are 2 to 20), and  $T_d$  is the derivative time constant.

The bode plot of an ideal PD-controller and a filtered PD controller are similar for low frequencies. Using a low-pass filter, we can at a certain frequency, reduce the gain of the derivative term, and thereby reduce the noise amplification.

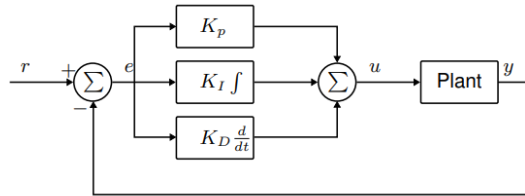


## 4.4 PID-controller

The control law of a PID feedback controller is

$$u(t) = K_p e(t) + K_i \int_{t_0}^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

A block diagram of the controller is given below:



Alternatively, the PID-controller with filter on the D-term is

$$K(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{s T_d}{1 + s T_d / N} \right)$$

## 4.5 Tuning a PID Controller

### 4.5.1 Pole placement

If a model of the system is available, then it is possible to compute an expression for the characteristic polynomial of the closed-loop system. Based on this polynomial, it may be possible to place the poles at desired locations.

### 4.5.2 Ziegler-Nichols method

Sometimes a model of the plant is not available, then the controller should be tuned by only studying the input-output behaviour of the system. Ziegler and Nichols has proposed two methods for tuning PID controllers without explicit use of a plant model.

- Ziegler-Nichols tuning based on step response

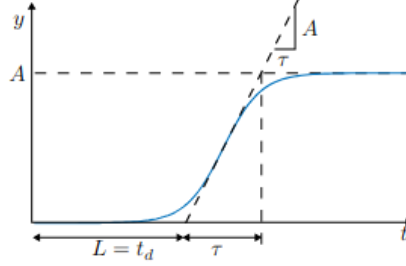
- Ziegler-Nichols tuning - The ultimate sensitivity method

### Tuning based on step response

The considered system is assumed to have the transfer function

$$\frac{y(s)}{u(s)} = \frac{A}{\tau s + 1} e^{-st_d}$$

This is a first-order system with a time delay of  $t_d$ .



The parameters for the PID controller can be found using the following table:

Controller type	Gains
P	$K_p = \frac{1}{RL}$
PI	$K_p = \frac{0.9}{RL}$ $T_i = \frac{L}{0.3}$
PID	$K_p = \frac{1.2}{RL}$ $T_i = 2L$ $T_d = 0.5L$

where  $R = A/\tau$  is the amplitude of the oscillation, and  $L = t_d$ .

### The ultimate sensitivity method

Start using a P-controller, and increase the gain until the system starts to oscillate. The value of  $K_p$  when the output oscillates with a constant amplitude is called the ultimate gain  $K_u$ . The period of the oscillation is called the ultimate period  $P_u$ .

The parameters for the PID controller can be found using the following table:

Controller type	Gains
P	$K_p = 0.5K_u$
PI	$K_p = 0.45K_u$ $T_i = \frac{P_u}{1.2}$
PID	$K_p = 0.6K_u$ $T_i = 0.5K_u$ $T_d = \frac{1}{8}P_u$



## 4.6 Examples

## 5 Root Locus

Rudkurve metoden. Grafisk metode til desgin af regulatorer.

### Rule 1

Consider the characteristic equation:

$$1 + KG(s) = 1 + K \frac{Q(s)}{P(s)} = 0$$

This can be written as:

$$P(s) + KQ(s) = 0$$

This is a polynoial of degree  $N = \max(m, n)$  where m is the number of poles and n is the number of zeros. Lemma. A univariate polynomial of degree d has d roots in  $\mathbb{C}$ .

There are  $N$  lines (loci) where  $N = \max(m, n)$ . Where m is the number of poles and n is the number of zeros.

### Rule 2

Similar to previously, the characteristic equation is rewritten as:

$$P(s) + KQ(s) = 0$$

Let  $K = 0$ , then we observe that the roots of the characteristic equation are the poles of the open-loop system.

Let  $K \rightarrow \infty$ , then we observe that the roots of the characteristic equation are the zeros of the open-loop system.

$$\frac{P(s)}{K} + Q(s) = 0$$

Defintion: As  $K$  increases from 0 to  $\infty$ , the root move from the poles of  $G(s)$  to the zeros of  $G(s)$ .

### Rule 3

When roots are complex they occur in conjugate pairs.

### Rule 4

We study the rewritten characteristic equation:

$$\frac{Q(s)}{P(s)} = -\frac{1}{K}$$

And see that the phase of  $\frac{Q(s)}{P(s)}$  is  $180^\circ$  to satisfy the equation.

The transfer function can be written as:

$$T(s) = \frac{Q(s)}{P(s)} = \frac{(s - z_1)(s - z_2) \cdots (s - z_m)}{(s - p_1)(s - p_2) \cdots (s - p_n)}$$

where  $z_i$  are the zeros and  $p_i$  are the poles. Let  $z \in \mathbb{C}$  then  $zz^* = |z|^2$ ; hence, complex pole pairs and pairs of complex conjugated zeros do not affect the phase of  $T(s)$  for  $s \in \mathbb{R}$ .

The phase of  $(s - z_m)$  when  $s, z_m \in \mathbb{R}$  is

$$\angle(s - z_m) = \begin{cases} 180^\circ & \text{if } s < z_m \\ 0^\circ & \text{otherwise} \end{cases} \quad (1)$$

The portion of the real axis to the left of an odd number of open loop poles and zeros are part of the loci.

#### Rule 5

Lines leave and enter the real axis at  $90^\circ$  angles.

#### Rule 6

For very large values of  $s$  the equation:

$$1 + K \frac{s^m + b_1 s^{m-1} + \dots + b_m}{s^n + a_1 s^{n-1} + \dots + a_n} = 0$$

can be approximated to:

$$1 + K \frac{1}{(s - \alpha)^{n-m}}$$

The phase of this expression should be  $180^\circ$  for this expression to hold. This implies that:

$$(n - m)\phi_l = 180^\circ + 360^\circ(l - 1)$$

where  $\phi_l$  is the phase of the expression.

let  $m < n - 1$  then:

$$-\sum r_i = -\sum p_i$$

where  $r_i$  are the closed-loop poles and  $p_i$  are the open-loop poles.

For  $s$  going to  $\infty$ , it is known that  $m$  closed-loop poles go towards the open-loop zeros, and  $n - m$  closed-loop poles go towards  $\alpha$ , i.e.

$$-\sum r_i = -(n - m)\alpha - \sum z_i = -\sum p_i$$

where  $z_i$  are the open-loop zeros.

Thus,

$$\alpha = \frac{\sum z_i - \sum p_i}{n - m}$$

## 5.1 Examples

## 6 The Nyquist Stability Criterion

### 6.1 Frequency response

The frequency response of a system is the steady-state response of a system to a sinusoidal input. The output of a time-invariant system will have the same frequency as the sinusoidal input, but possibly with a different amplitude and phase.

The output of a time-invariant system is given by:

$$y(t) = \int_{-\infty}^{\infty} h(\tau)u(t - \tau)d\tau$$

Where  $y$  is the output,  $u$  is the input, and  $h$  is the impulse response of the system.

To obtain the frequency response, only sinusoidal inputs are considered.

The frequency response of  $H(s)$  is given by the magnitude and phase of  $H(j\omega)$

$$M = |H(j\omega)|$$

$$\phi = \angle H(j\omega)$$

The  $M$  is the amplitude ratio and  $\phi$  is the phase shift.

The bandwidth of a closed-loop system  $T(s)$  is defined to be the maximum frequency at which the output  $y$  of a system will track a sinusoidal input  $r$  in a satisfactory manner. Output attenuated to  $1/\sqrt{2}$  of the input amplitude. Formally the bandwidth  $\omega_{BW}$  of  $T(s)$  is the maximal frequency such that:

$$|T(j\omega)| \geq 1/\sqrt{2}$$

The maximal value of the frequency response is called the resonant peak,

### 6.2 Bode Plots

Pole decreases the magnitude of the frequency response by 20dB/decade. Zero increases the magnitude of the frequency response by 20dB/decade.

#### Bode Form of transfer function

In the following, we consider the open-loop transfer function

$$KG(s) = K \frac{(s - z_1)(s - z_2) \cdots}{(s - p_1)(s - p_2) \cdots}$$

For convenience, the transfer function is evaluated at  $s = j\omega$  and rewritten into Bode form as follows:

$$KG(j\omega) = K_0 \frac{(j\omega\tau_1 + 1)(j\omega\tau_2 + 1) \cdots}{(j\omega\tau_a + 1)(j\omega\tau_b + 1) \cdots}$$

## Classes of terms in transfer function

**Class 1:**  $K_0(j\omega)^n$  for  $n \in \mathbb{Z}$

Example: For the term  $K_0(j\omega)^n$  we have:

$$\log K_0(j\omega) = \log K_0 + n \log \omega$$

This means that the magnitude plot is a straight line with slope  $n \cdot (20\text{dB/dec})$ . The phase of  $(j\omega)^n$  is constant and given by:  $\phi = n \cdot 90^\circ$ .  $n = -1$  means that there is a pole at the origin. This therefore leads to a slope of  $-20\text{dB/dec}$  and a phase shift of  $-90^\circ$ .

**Class 2:**  $(j\omega\tau + 1)^{\pm 1}$

For  $\omega\tau \ll 1$ ,  $j\omega\tau \approx 1$

For  $\omega\tau \gg 1$ ,  $j\omega\tau + 1 \approx j\omega\tau$

In addition, for  $\omega = 1/\tau$ , the gain is  $\sqrt{2}$  - an increase of 3dB compared to the DC gain. The point  $\omega = 1/\tau$  is called the break point. At the break point the angle is  $45^\circ$ .

**Class 3:**  $((j\omega/\omega_n)^2 + 2\zeta(j\omega/\omega_n) + 1)^{\pm 1}$

For  $\omega \ll \omega_n$ , the amplitude is approximately 1. In addition, at the break point  $\omega = \omega_n$ , the magnitude is  $|G(j\omega)| = 1/(2\zeta)$  and the phase is  $\pm 90^\circ$ .

## Summary of Bode Plot Rules

1. Rewrite the considered transfer function to bode form
2. Determine the value of the  $K_0(j\omega)^n$  term. Plot the low frequency magnitude asymptote through the point  $K_0$  at  $\omega = 1$  and with slope of  $n \cdot 20\text{dB/dec}$
3. Complete the composite magnitude asymptotes by extending the low-frequency asymptote until the first frequency break point. Then change the slope according to the behavior at the break point, and continue the procedure for the remaining break points.
4. Sketch the approximate magnitude curve by increasing the asymptote value by a factor  $\sqrt{2}$  at first-order numerator break and decreasing it by a factor  $1/\sqrt{2}$  at denominator break.
5. Plot the low-frequency asymptote of the phase  $\phi = n \cdot 90^\circ$
6. Change the phase at the phase points, and correct the phase according to the slope at the phase point.

## 6.3 Nyquist Stability Criterion

A contour map of a complex function will encircle the origin Z-P times where Z is the number of zeros and P is the number of poles of the function inside the contour.

To verify the stability of a system, one needs to determine the number of closed-loop poles in the right half plane. Thus, the Nyquist plot is a map of a contour that encircles the entire right-half plane.

The number of closed-loop poles in the right half-plane equals the number of right half-plane zeros of the characteristic equation. The characteristic equation is given by:

$$1 + KG(s) = 0$$

Let N denote the number of clockwise encirclements of -1. Then the number of zeros in the right half plane Z (closed-loop poles) minus the number of open-loop poles in the right half plane P.

$$N = Z - P$$

## 6.4 Examples

## 7 Dynamic Compensators and Stability Margins

### 7.1 Stability Margins

Gain margin: factor by which the gain can be raised before a system becomes unstable.

Phase margin: The amount by which the phase can be increased before the system becomes unstable. (exceeds  $-180^\circ$ )

### 7.2 Dynamic compensation

Lead compensator is equivalent to a d-part for a PID controller. A lead compensator has a positive phase.

Lag compensator is equivalent to an i-part for a PID controller.

Both compensators are given by the transfer function:

$$D(s) = K \frac{s + z}{s + p}$$

Where  $z$  and  $p$  are the zero and pole of the transfer function respectively.

- If  $z < p$ , then  $D(s)$  is called a lead compensation
- If  $z > p$ , then  $D(s)$  is called a lag compensation

#### 7.2.1 Lead compensation

Used for dynamic properties such as rise time, overshoot and settling time. It has relation to the phase margin, where we can lift the phase margin by adding a lead compensator.

A lead compensation is given by:

$$D(s) = \frac{Ts + 1}{\alpha Ts + 1}$$

and  $1/\alpha$  is called the lead ratio.

We have the following properties:

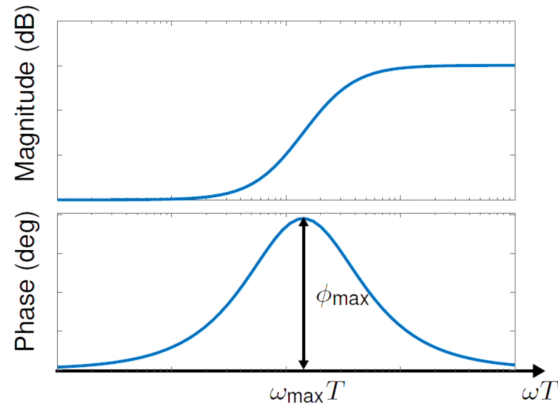
$$\omega_{max} = \frac{1}{T\sqrt{\alpha}} = \sqrt{|z||p|}$$

,

$$T = \frac{1}{\omega_{max}\sqrt{\alpha}}$$

$$\sin(\phi_{max}) = \frac{1 - \alpha}{1 + \alpha}$$

$\phi_{max}$  is the maximum phase lead. Which means that it is the maximum phase that the lead compensator can provide. And  $\omega_{max}$  is the frequency at which the phase lead is maximum.



### 7.2.2 Lag compensation

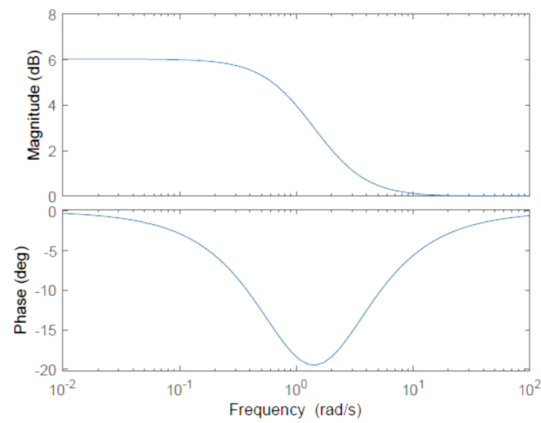
A lag compensator can be used for minimizing the steady state error without affecting the other dynamics. Only looking at stationary error. It doesn't eliminate the error but reduces it.

A lag compensator can be written as:

$$D(s) = K_0 \frac{\alpha Ts + 1}{\alpha Ts + 1}$$

where  $K_0$  is the gain of the compensator,  $z > p \in \mathbb{R}$  and  $\alpha > 1$ .

The gain decreases and the phase gets negative.



## 7.3 Examples

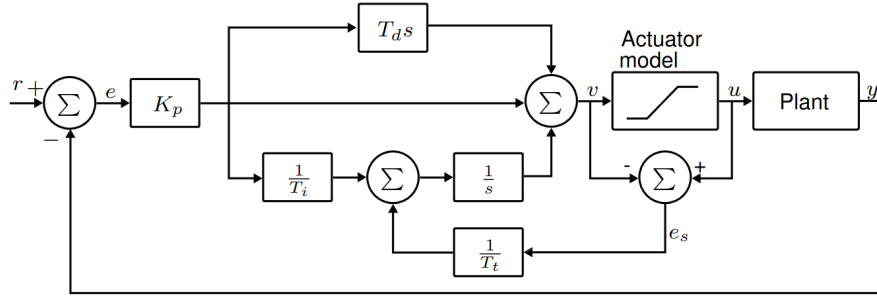
## 8 Implementation

### 8.1 Anti-windup

Real electromechanical systems has saturations on their physical variables. In particular, the output of any actuator is limited from above and below. (Clipping). If you stay between the limits, the system will behave as a linear system.

#### Back-calculation

An anti-windup scheme that recomputes the integral term when the system input is saturated. The value of the integrator output is not changed instantaneously, but it is changed based on the tracking time constant.



The input to the integrator is given by

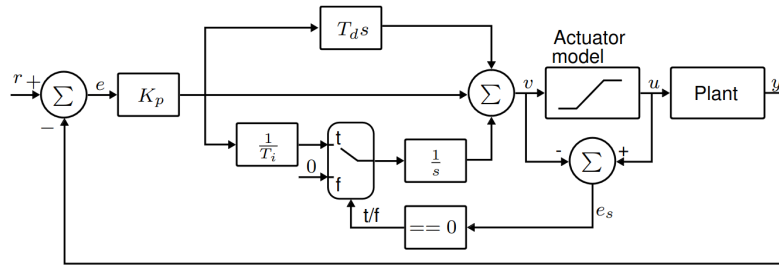
$$\frac{1}{T_t}e_s + \frac{K_p}{T_i}e$$

Where  $e_s$  is zero when the system is not saturated. In steady state, the output of the integrator is constant: hence, its input must be zero, i.e.

$$e_s = -\frac{K_p T_t}{T_i}e$$

#### Conditional integration

Also known as clamping is a bit simpler, and just stops integrating when the system is in saturation.



#### Setpoint weighting

To avoid large control signals when changing the reference rapidly, setpoint weights can be introduced. These modify the PID controller to be:

$$u(t) = K_p(\beta r(t) - y(t)) + K_i \int_0^t (r(\tau) - y(\tau))d\tau + K_d(\gamma \frac{dr}{dt} - \frac{dy}{dt})$$

where  $\gamma$  and  $\beta$  are setpoint weights. Typically  $\gamma = 0$  and  $\beta$  takes values between 0 and 1.



## Filtering

To avoid large control signal noise on the measured output  $y$ , one adds a filter on the derivative term. Often a first order filter is sufficient, but this is application dependent.

$$u_d = k_d s \frac{1}{1 + sT_f}$$

Sometimes it is favorable to filter the control signal directly, thus the controller gets the form

$$C(s) = K_p \left(1 + \frac{1}{sT_i} + sT_d\right) \frac{1}{1 + sT_f + (sT_f)^2/2}$$

## 8.2 Digitalization

Most control systems are sample data systems, i.e., they consist of both discrete and continuous signals. The sampling frequency used for the discrete controller should be above 20 times the closed-loop bandwidth.

### 8.2.1 Emulation

Design continuous controller  $K(s)$  and approximate it with  $K(z)$  obtained via e.g. Tustin's method. Tustin's method also called the trapezoidal rule means to replace the variable  $s$  with

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

The control output of the discrete PID controller can be written using 3 terms.

$$u_p(kT + T) = k_p e(kT + T)$$

$$u_I(kT + T) = u_I(kT) + K_i \frac{T}{2} (e(kT + T) + e(kT))$$

$$u_D(kT + T) = k_D \frac{2}{T} (e(kT + T) - e(kT)) - u_D(kT)$$

$$u(kT + T) = u_p(kT + T) + u_I(kT + T) + u_D(kT + T)$$

To analyze the system, the I-term and D-term are z-transformed:

$$zu_I(z) = u_I(z) + k_i \frac{T}{2} (ze(z) + e(z))$$

$$zu_D(z) = k_D \frac{2}{T} (ze(z) - e(z)) - u_D(z)$$

This gives the following expression for the controller

$$u(z) = (k_P + k_I \frac{T}{2} \frac{z + 1}{z - 1} + k_D \frac{2}{T} \frac{z - 1}{z + 1}) e(z)$$

## Compensation for sampling effects

## Design procedure

A discrete controller can be designed by emulation for the system  $G(s)$  according to the next procedure:

1. Design continuous compensation for the system  $G_d(s)G(s)$ , where  $G_d(s)$  approximates a delay of  $T/2$
2. Derive the discrete controller by applying Tustin's rule or the matched pole-zero method
3. Analyze the design by simulation or experimentally

### 8.2.2 Numerical Integration Methods

#### 8.2.3 Discrete Design

Design the discrete controller directly, without computing  $K(s)$  first.

## 8.3 Examples

### Emulation example of PID controller

$$K(s) = 1.4 \frac{s+6}{s}$$

Use Tustin's method to emulate the controller in discrete time.

$$K(z) = K\left(\frac{2}{T} \frac{z-1}{z+1}\right) = 1.4 \frac{\frac{2}{T} \frac{z-1}{z+1} + 6}{\frac{2}{T} \frac{z-1}{z+1}} = 1.4 \frac{(1+3T)z - (3T-1)}{z-1}$$

In practice you implement the difference equation - not a discrete transfer function.