9. State-Space Methods

9.1 State-space models
   9.1.1 Introduction
   9.1.2 Basic concepts and definitions
   9.1.3 Building a state-space model
   9.1.4 Linearization of a state-space model

9.2 Linear state-space models
   9.2.1 From state-space model to transfer function
   9.2.2 Controllability and observability
   9.2.3 Linear state-vector transformations
   9.2.4 Canonical state-space forms for SISO systems
   9.2.5 Stability, poles, and zeros

9.3 Control design by state feedback
   9.3.1 Pole placement
   9.3.2 Linear-quadratic control
   9.3.3 State estimation
9. State-Space Methods

9.1 State-space models

9.1.1 Introduction

- For systems with multiple input signals and multiple output signals (MIMO systems), it becomes difficult to work with transfer functions between each input and output.
- A more compact and clearer model form is desired.
- Such a model form is a state-space model.
- A state-space model is based on the concept of state variables and uses formalism derived from linear algebra (“matrices and vectors”).
- The concept of state variables is a natural concept. The state variables are so-called intensive quantities, which describe the state of a system.
- The state variables are often (but not always) natural process variables such as pressure, temperature, concentration, etc.
9. State-Space Methods

9.1 State-space models

### 9.1.2 Basic concepts and definitions

Assume that a *system state* at time $t$ can be expressed by a finite set of variables $x_1(t), x_2(t), ..., x_n(t)$.

- These variables are called the *state variables* of the system.
- To enable a compact description of the system, the state variables are collected into a *state vector*

$$
\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = [x_1(t) \ x_2(t) \ \cdots \ x_n(t)]^T \tag{9.1}
$$

- Normally, there are a number of *input signals* $u_1(t), u_2(t), ..., u_m(t)$, which are collected into an *input vector*

$$
\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix} = [u_1(t) \ u_2(t) \ \cdots \ u_m(t)]^T \tag{9.2}
$$
9.1 State-space models

- Usually, there are a number of output signals \(y_1(t), y_2(t), \ldots, y_p(t)\), which are collected into an output vector

\[
y(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{bmatrix} = [y_1(t) \ y_2(t) \ \cdots \ y_p(t)]^T
\]  

For a continuous-time state-space system the relationship between these variables can be written as a system of first order differential equations

\[
\frac{dx(t)}{dt} \equiv \dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0
\]

\[
y(t) = h(x(t), u(t))
\]  

(9.4)

On the basis of (9.4), the state variables can be defined as follows:

- A system state at time \(t = t_0\) is given by the minimum number of variables \(x_1(t), x_2(t), \ldots, x_n(t)\), which together with the input signals to the system for \(t \geq t_0\) are necessary to describe the behaviour of the system for all \(t \geq t_0\).

The state of a system state has a nice physical interpretation:

- The state of a system represents the minimum information about the system which is necessary to predict the future behaviour of the system.
Example 9.1. State-space model from third-order transfer function.

Consider a system with the transfer function

\[ G(s) = \frac{Y(s)}{U(s)} = \frac{k}{s^3 + a_1 s^2 + a_2 s + a_3} \]  

(Ex 9.1.1)

or

\[(s^3 + a_1 s^2 + a_2 s + a_3)Y(s) = kU(s)\]  

(Ex 9.1.2)

The inverse Laplace transform yields the third-order differential equation

\[ \ddot{y}(t) + a_1 \dot{y}(t) + a_2 \dot{y}(t) + a_3 y(t) = ku(t) \]  

(Ex 9.1.3)

where the initial conditions are zero because it is inherent in (Ex 9.1.2).

A state-space model allows only first-order differential equations. To achieve this, new variables are introduced according to (other choices are possible)

\[ x_1(t) = \dot{y}(t), \quad x_2(t) = \dot{y}(t), \quad x_3(t) = y(t) \]  

(Ex 9.1.4)

which substituted into (Ex 9.1.3) yields

\[ \dot{x}_1(t) + a_1 x_1(t) + a_2 x_2(t) + a_3 x_3(t) = ku(t) \]  

(Ex 9.1.5)

Note that the highest derivative of \( y(t) \) is not redefined — it is expressed by the first derivative of \( x_1(t) \).
9.1.2 Basic concepts and definitions

Equations (Ex 9.1.4–5), including the initial conditions, can be expressed as the equation system

\[
\begin{align*}
\dot{x}_1(t) &= -a_1 x_1(t) - a_2 x_2(t) - a_3 x_3(t) + ku(t), \quad x_1(0) = 0 \\
\dot{x}_2(t) &= x_1(t), \quad x_2(0) = 0 \quad \text{(Ex 9.1.6)} \\
\dot{x}_3(t) &= x_2(t), \quad x_3(0) = 0 \\
y(t) &= x_3(t)
\end{align*}
\]

which is in the general form of (9.4).

Equation (Ex 9.1.6) can be written more neatly with vectors and matrices as

\[
\begin{bmatrix}
\dot{x}_1(t) \\
\dot{x}_2(t) \\
\dot{x}_3(t)
\end{bmatrix} = 
\begin{bmatrix}
-a_1 & -a_2 & -a_3 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x_1(t) \\
x_2(t) \\
x_3(t)
\end{bmatrix} +
\begin{bmatrix}
k \\
0 \\
0
\end{bmatrix} u(t), \quad
\begin{bmatrix}
x_1(0) \\
x_2(0) \\
x_3(0)
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix} \quad \text{(Ex 9.1.7)}
\]

\[
y(t) =
\begin{bmatrix}
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1(t) \\
x_2(t) \\
x_3(t)
\end{bmatrix}
\]
9.1.2 Basic concepts and definitions

Equation (Ex 9.1.7) can be written more compactly as

\[ \dot{x}(t) = Ax(t) + bu(t), \quad x(0) = 0 \]
\[ y(t) = c^T x(t) \]  

(Ex 9.1.8)

with

\[ A = \begin{bmatrix} -a_1 & -a_2 & -a_3 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} k \\ 0 \\ 0 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]  

(Ex 9.1.9)

where \( A \) is called the system matrix.

- It can be concluded that a transfer function of \( n^{th} \) order is equivalent to \( n \) differential equations of first order. This means that the system has \( n \) number of state variables.

- It should be noted that the choice of state variables is not unique; other choices could have been made in (Ex 9.1.4).
9.1.3 Building a state-space model

Process models in the form of state-space models can be derived based on *balance equations* and appropriate *constitutive relations*. Examples of balance equations are material and energy balances, while the constitutive relations are expressions for transport laws, reaction kinetics, etc. The modelling principle is illustrated by the following example.

**Example 9.2. Modelling a chemical reactor.**

In a continuously stirred tank reactor (CSTR), or completely mixed chemical reactor, the following chemical reactions take place at constant temperature:

\[
2A \rightarrow B \quad r_1 = k_1 c_A^2 \quad k_1 = 1 \text{ (mol/ℓ)}^{-1} \text{h}^{-1}
\]

\[
B \rightarrow D \quad r_2 = k_2 c_B \quad k_2 = 2 \text{ h}^{-1}
\]

The reactor has a continuous inflow \( \dot{V}_f \) and a continuous outflow \( \dot{V}_e \) with the chemical components A, B and D dissolved in the liquid flows. The concentrations of the components in the reactor are \( c_A, c_B \) and \( c_D \), respectively. The concentrations are relatively low, which means that constant liquid density can be assumed in the flows and in the reactor.
9.1.3 Building a state-space model

A total material balance over the reactor, with densities cancelled out, yields

$$\frac{dV}{dt} = \dot{V}_f - \dot{V}_e$$  \hspace{1cm} (Ex 9.2.1)

Partial material balances for the chemical components yield

$$\frac{d(Vc_A)}{dt} = \dot{V}_f c_{Af} - \dot{V}_e c_A - V \cdot 2r_1$$

$$\frac{d(Vc_B)}{dt} = \dot{V}_f c_{Bf} - \dot{V}_e c_B + Vr_1 - Vr_2$$  \hspace{1cm} (Ex 9.2.2)

$$\frac{d(Vc_D)}{dt} = \dot{V}_f c_{Df} - \dot{V}_e c_D + Vr_2$$

Development of the derivatives in (Ex 9.2.2) according to the principle

$$\frac{d(Vc_X)}{dt} = V \frac{dc_X}{dt} + c_X \frac{dV}{dt}$$  \hspace{1cm} (Ex 9.2.3)

and elimination of \(dV/dt\) in (Ex 9.2.3) by (Ex 9.2.1) yield
9.1.3 Building a state-space model

\[
\begin{align*}
V \frac{dc_A}{dt} &= \dot{V}_f (c_{Af} - c_A) - 2V k_1 c_A^2 \\
V \frac{dc_B}{dt} &= \dot{V}_f (c_{Bf} - c_B) + V k_1 c_A^2 - V k_2 c_B \\
V \frac{dc_D}{dt} &= \dot{V}_f (c_{Df} - c_D) + V k_2 c_B
\end{align*}
\]  

(Ex 9.2.4)

With the definitions

- \( x_1 = c_A \), \( x_2 = c_B \), \( x_3 = c_D \)
- \( u_1 = c_{Af} \), \( u_2 = c_{Bf} \), \( u_3 = c_{Df} \), \( u_4 = \dot{V}_f / V \)

these equations are of the form

\[
\dot{x}_i = f_i(x_1, x_2, x_3, u_1, u_2, u_3, u_4)
\]  

(Ex 9.2.5)

or more compactly

\[
\dot{x}(t) = f(x, u)
\]  

(Ex 9.2.6)

which is in the general form of (9.4).

This is a system of nonlinear first order differential equations. However, linear differential equations are needed for efficient analysis and design methods based on linear algebra.
9.1.4 Linearization of a state-space model

As shown above, a state-space model derived from physical and chemical laws tends to be nonlinear. Here it will be shown how such a model can be linearized.

Consider the nonlinear state-space model

\[
\begin{align*}
\dot{x}(t) &= f(x(t), u(t)) \\
y(t) &= h(x(t), u(t))
\end{align*}
\]

where \( \dim(x) = n \), \( \dim(u) = m \), and \( \dim(y) = p \).

The equations can be linearized by a Taylor series expansion around a reference state \((\bar{x}, \bar{u})\) using only terms containing partial derivatives up to first order.

Consider the state equation \( \dot{x}(t) = f(x(t), u(t)) \). For the \( i^{th} \) state-variable relationship \( \dot{x}_i(t) = f_i(x(t), u(t)) \), the Taylor series expansion yields

\[
\dot{x}_i(t) \approx f_i(\bar{x}, \bar{u}) + \frac{\partial f_i(x,u)}{\partial x} \bigg|_{x=\bar{x}} (x(t) - \bar{x}) + \frac{\partial f_i(x,u)}{\partial u} \bigg|_{u=\bar{u}} (u(t) - \bar{u})
\]
9.1 State-space models

In (9.6), the **partial derivatives** with respect to the vectors \( x \) and \( u \) are **row vectors**

\[
\frac{\partial f_i(x,u)}{\partial x} = \begin{bmatrix} \frac{\partial f_i(x,u)}{\partial x_1} & \frac{\partial f_i(x,u)}{\partial x_2} & \cdots \end{bmatrix}, \quad \frac{\partial f_i(x,u)}{\partial u} = \begin{bmatrix} \frac{\partial f_i(x,u)}{\partial u_1} & \frac{\partial f_i(x,u)}{\partial u_2} & \cdots \end{bmatrix}
\] (9.7)

If \((\bar{x}, \bar{u})\) is chosen as a steady state, where \(\dot{x}(t) = 0, f_i(\bar{x}, \bar{u}) = 0\). By introducing the **deviation variables**

\[
\delta x(t) = x(t) - \bar{x}, \quad \delta u(t) = u(t) - \bar{u}, \quad \delta y(t) = y(t) - \bar{y}
\] (9.8)

and noting that \(\delta \dot{x}(t) = \dot{x}(t)\), (9.6) can be written as

\[
\delta \dot{x}_i(t) \approx \frac{\partial f_i(x,u)}{\partial x} \bigg|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \frac{\partial f_i(x,u)}{\partial u} \bigg|_{x=\bar{x}, u=\bar{u}} \delta u(t)
\] (9.9)

Consider now the relationship \(y(t) = h(x(t), u(t))\). At the steady state \((\bar{x}, \bar{u}), \bar{y} = h(\bar{x}, \bar{u})\). A similar linearization of \(y_j(t) = h_j(x(t), u(t))\) as above then yields

\[
\delta y_j(t) \approx \frac{\partial h_j(x,u)}{\partial x} \bigg|_{x=\bar{x}, u=\bar{u}} \delta x(t) + \frac{\partial h_j(x,u)}{\partial u} \bigg|_{x=\bar{x}, u=\bar{u}} \delta u(t)
\] (9.10)
9.1 State-space models

When all states and outputs are considered, (9.9) and (9.10) can be written compactly as

\[
\begin{align*}
\delta \dot{x}(t) &= A \delta x(t) + B \delta u(t) \\
\delta y(t) &= C \delta x(t) + D \delta u(t)
\end{align*}
\]

(9.11)

where

\[
A = \left. \frac{\partial f(x,u)}{\partial x} \right|_{x=x, u=u} = \begin{bmatrix}
\frac{\partial f_1(x, u)}{\partial x} \\
\frac{\partial f_2(x, u)}{\partial x} \\
\vdots \\
\frac{\partial f_n(x, u)}{\partial x}
\end{bmatrix} \bigg|_{x=x, u=u}, \\
B = \left. \frac{\partial f(x,u)}{\partial u} \right|_{x=x, u=u} = \begin{bmatrix}
\frac{\partial f_1(x, u)}{\partial u} \\
\frac{\partial f_2(x, u)}{\partial u} \\
\vdots \\
\frac{\partial f_n(x, u)}{\partial u}
\end{bmatrix} \bigg|_{x=x, u=u}
\]

\[
C = \left. \frac{\partial h(x,u)}{\partial x} \right|_{x=x, u=u} = \begin{bmatrix}
\frac{\partial h_1(x, u)}{\partial x} \\
\frac{\partial h_2(x, u)}{\partial x} \\
\vdots \\
\frac{\partial h_p(x, u)}{\partial x}
\end{bmatrix} \bigg|_{x=x, u=u}, \\
D = \left. \frac{\partial h(x,u)}{\partial u} \right|_{x=x, u=u} = \begin{bmatrix}
\frac{\partial h_1(x, u)}{\partial u} \\
\frac{\partial h_2(x, u)}{\partial u} \\
\vdots \\
\frac{\partial h_p(x, u)}{\partial u}
\end{bmatrix} \bigg|_{x=x, u=u}
\]

The matrices have the dimensions

- \( \dim(A) = n \times n \), \( \dim(B) = n \times m \), \( \dim(C) = p \times n \), \( \dim(D) = p \times m \)

Note that \( \frac{\partial f_i(x, u)}{\partial x} \), \( \frac{\partial f_i(x, u)}{\partial u} \), \( \frac{\partial h_j(x, u)}{\partial x} \) and \( \frac{\partial h_j(x, u)}{\partial u} \) are row vectors as defined in (9.7). For a strictly proper system, \( D = 0 \).
Exercise 9.1

Linearize the chemical reactor in Example 9.2 around the steady state defined by $\bar{c}_{Af} = 2 \text{ mol/}\ell$, $\bar{c}_{Bf} = \bar{c}_{Df} = 0 \text{ mol/}\ell$, $\bar{V}_f = 3 \text{ m}^3/\hbar$, and $\bar{V} = 1.5 \text{ m}^3$. Note that other steady-state values that might be needed can be solved from (Ex 9.2.4).

Write the linearized model in state-space form.
9.2 Linear state-space models

In this section, linear, continuous-time, time-invariant state-space models are considered. The model is

- linear, when the state-variable derivatives and the outputs depend linearly on the state variables and the inputs
- continuous-time, when the variables are defined for continuous time
- time-invariant, when the model parameters do not change with time

Such a state-space model has the form

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{align*}
\]  

(9.12)

where \( x, u \) and \( y \) are understood to be deviation variables. The variables and the matrices are defined by

- state vector \( x(t) \in \mathbb{R}^n \), input vector \( u(t) \in \mathbb{R}^m \), output vector \( y(t) \in \mathbb{R}^p \)
- system matrix \( A \in \mathbb{R}^{n \times n} \), input matrix \( B \in \mathbb{R}^{n \times m} \), output matrix \( C \in \mathbb{R}^{p \times n} \), direct term \( D \in \mathbb{R}^{p \times m} \)

A state-space model can (in the text) be referred to as \( (A, B, C, D) \).
9.2 Linear state-space models

9.2.1 From state-space model to transfer function

The Laplace transform of a state-space model:

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Bu(t) \implies sx(s) - x(0^-) = AX(s) + BU(s) \quad (9.13) \\
y(t) &= Cx(t) + Du(t) \implies Y(s) = CX(s) + DU(s) \quad (9.14)
\end{align*}
\]

Equation (9.13) yields

\[
(sI - A)X(s) = x(0^-) + BU(s)
\]

\[
X(s) = (sI - A)^{-1}(x(0^-) + BU(s))
\]

which substituted into (9.14) gives

\[
Y(s) = C(sI - A)^{-1}x(0^-) + C(sI - A)^{-1}BU(s) + DU(s)
\]

If the initial state \( x(0^-) = 0 \), this gives

\[
Y(s) = (C(sI - A)^{-1}B + D)U(s) = G(s)U(s) \quad (9.15)
\]

where

\[
G(s) = C(sI - A)^{-1}B + D \quad (9.16)
\]

is the transfer (function) matrix of the system, i.e., a matrix of transfer functions.
9.2.2 Controllability and observability

Are \((A, B, C, D)\) and \(G(s)\) always equivalent (in the absence of time delays)?

**Example 9.3. Cancellation of poles and zeros.**

Consider the state-space model

\[
\begin{align*}
\dot{x}_1 &= -x_1 \\
\dot{x}_2 &= -x_2 + 2u \\
y &= x_1 + u
\end{align*}
\]

\[
\Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u
\]

\[
y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [1] u
\]

This is a state-space model of second order \((n = 2)\) with one input and one output. Thus, a second-order transfer function is expected. Equation (9.16) yields

\[
G(s) = [1 \ 0] \left(sI - \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\right)^{-1} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 1 = [1 \ 0] \begin{bmatrix} s + 1 & 0 \\ 0 & s + 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 1
\]

\[
= [1 \ 0] \begin{bmatrix} (s + 1)^{-1} & 0 \\ 0 & (s + 1)^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 1 = [((s + 1)^{-1} & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \end{bmatrix} + 1 = 1
\]

Thus, \(G(s) = 1\) although the system is of second order!

The reason is that two poles \(s = -1\) are cancelled out against two zeros \(s = -1\).
9.2 Linear state-space models

9.2.2 Controllability and observability

**Controllability**

A state \( x^* \) is controllable if there is a control signal which in finite time can move the state \( x = 0 \) to \( x^* \). *A system is controllable if all states are controllable.*

The linear system \((A, B)\) is controllable if and only if the *controllability matrix*

\[
\Gamma_c \equiv \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \quad (9.17)
\]

has full rank, i.e., \( \text{rank}(\Gamma_c) = n \). The matrix \( \Gamma_c \) has full rank if it is possible to find \( n \) columns in \( \Gamma_c \) that form a matrix with a non-zero determinant.

**Observability**

A state \( x^* \neq 0 \) is unobservable if, when \( x(0) = x^* \) and \( u(t) = 0, t \geq 0 \), the output \( y(t) = 0, t \geq 0 \). *A system is observable if there is no unobservable state.*

The linear system \((A, C)\) is observable if and only if the *observability matrix*

\[
\Gamma_o \equiv \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (9.18)
\]

has full rank, i.e., \( \text{rank}(\Gamma_o) = n \). The matrix \( \Gamma_o \) has full rank if it is possible to find \( n \) rows in \( \Gamma_o \) that form a matrix with a non-zero determinant.
9.2 Linear state-space models

Kalman’s decomposition
A linear state-space system $\mathcal{S} = (A, B, C, D)$ can, in general, be subdivided into four subsystems: a subsystem

- $\mathcal{S}_{co}$, which is controllable and observable
- $\mathcal{S}_{cu}$, which is controllable, but unobservable
- $\mathcal{S}_{uo}$, which is uncontrollable, but observable
- $\mathcal{S}_{uu}$, which is uncontrollable and unobservable

Minimal realization
A state-space description which is both controllable and observable is a minimal realization of the system.

Equivalence between $G(s)$ and $(A, B, C, D)$
The transfer matrix $G(s)$ represents a complete description of the system $(A, B, C, D)$ if and only if the system is both controllable and observable (i.e., the system description is a minimal realization).

Exercise 9.2
Examine the controllability and observability of the system in the Example 9.3 (beginning of Section 9.2.2).
9.2.3 Linear state-vector transformations

Consider the state-space model

\[
\dot{x}(t) = Ax(t) + Bu(t) \\
y(t) = Cx(t) + Du(t)
\]  

(9.19)

A variable transformation

\[
z = Tx , \ x = T^{-1}z
\]  

(9.20)

yields

\[
\dot{z}(t) = T\dot{x}(t) = TAx(t) + TBu(t)
\]

and

\[
\dot{z}(t) = TAT^{-1}z(t) + TBu(t) \\
y(t) = CT^{-1}z(t) + Du(t)
\]  

(9.21)

The \( A, B \) and \( C \) matrices of the state-space description are changed by the variable transformation so that

\[
(A, B, C, D) \Rightarrow (TAT^{-1}, TB, CT^{-1}, D)
\]  

(9.22)
9.2 Linear state-space models

Is \( G(s) \) affected by a linear state-vector transformation?

The transfer function matrix of the untransformed system \( (A, B, C, D) \) is

\[
G(s) = C(sI - A)^{-1}B + D \tag{9.23}
\]

The transformed system \( (TAT^{-1}, TB, CT^{-1}, D) \) has the transfer function matrix

\[
G(s) = CT^{-1}(sI - TAT^{-1})^{-1}TB + D \tag{9.24}
\]

or

\[
G(s) = CT^{-1}[T(sI - A)T^{-1}]^{-1}TB + D \\
= CT^{-1}[T(sI - A)^{-1}T^{-1}]TB + D \\
= C(sI - A)^{-1}B + D \tag{9.25}
\]

Thus, \( G(s) \) is not changed by a linear state-vector transformation. This is, of course, quite obvious since the input and the output vectors are not changed by the transformation.
Are the eigenvalues of the system matrix affected a linear transformation?

The eigenvalues of the system matrix are important system properties. The eigenvalues \( \lambda_i, i = 1, \ldots, n \), are given by the solutions to the equation

\[
\det(\lambda I - A) = 0
\]  
(9.26)

For the transformed system, the eigenvalues are then given by

\[
\det(\lambda I - TAT^{-1}) = 0
\]  
(9.27)

The eigenvalues are the same, because

\[
\det(\lambda I - TAT^{-1}) = \det[T(\lambda I - A)T^{-1}]
= \det(T) \cdot \det(\lambda I - A) \cdot \det(T^{-1})
= \det(T) \cdot \det(T^{-1}) \cdot \det(\lambda I - A)
= \det(TT^{-1}) \cdot \det(\lambda I - A)
= \det(I) \cdot \det(\lambda I - A) = \det(\lambda I - A)
\]  
(9.28)

Since \( \det(\lambda I - TAT^{-1}) = \det(\lambda I - A) \), the solutions must be the same. Thus, the transformed system matrix has the same eigenvalues as the untransformed system matrix.
9. State-Space Methods

9.2 Linear state-space models

9.2.4 Canonical state-space forms for SISO systems

In this section, so-called canonical (i.e., standard) state-space forms for systems with a single input and a single output (SISO) are considered.

The derivation of the canonical forms is based on an $n^{th}$ order proper system with the transfer function

$$G(s) = \frac{b_1 s^{n-1} + b_2 s^{n-2} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_{n-1} s + a_n} + d \quad (9.29)$$

**Diagonal canonical form**

If the poles (i.e., the roots of the characteristic equation) of (9.29) are distinct (unequal) and real, (9.29) has the partial fraction expansion

$$G(s) = \frac{k_1}{s-\lambda_1} + \frac{k_2}{s-\lambda_2} + \cdots + \frac{k_{n-1}}{s-\lambda_{n-1}} + \frac{k_n}{s-\lambda_n} + d \quad (9.30)$$

where $\lambda_i$, $i = 1, 2, \ldots, n$, are the poles of (9.29) and $k_i$, $i = 1, 2, \ldots, n$, are constants that make (9.30) identical to (9.29). When the poles are known (from a factorization of the denominator of (9.29)), the constants can be calculated by

$$k_i = \lim_{s \to \lambda_i} (s - \lambda_i) G(s) \ , \ i = 1, 2, \ldots, n \quad (9.31)$$
9.2.4 Canonical state-space forms for SISO systems

Define

\[ X_i(s) = \frac{k_i}{s-\lambda_i} U(s), \quad i = 1,2, \ldots, n \]  

(9.32)

Because \( G(s) = Y(s)/U(s) \), substitution into (9.30) yields

\[ Y(s) = X_1(s) + X_2(s) + \cdots + X_{n-1}(s) + X_n(s) + d \cdot U(s) \]  

(9.33)

The inverse Laplace transform of (9.32) and (9.33) yields

\[ \dot{x}_i(t) = \lambda_i x_i(t) + k_i u(t), \quad i = 1,2, \ldots, n \]

\[ y(t) = x_1(t) + x_2(t) + \cdots + x_{n-1}(t) + x_n(t) + du(t) \]

or

\[ \dot{x}(t) = \Lambda x(t) + b u(t) \]

\[ y(t) = c^T x(t) + du(t) \]  

(9.34)

where

\[ \Lambda = \begin{bmatrix}
\lambda_1 & 0 & 0 & 0 & 0 \\
0 & \lambda_2 & 0 & 0 & 0 \\
0 & 0 & \ddots & 0 & 0 \\
0 & 0 & 0 & \lambda_{n-1} & 0 \\
0 & 0 & 0 & 0 & \lambda_n
\end{bmatrix}, \quad b = \begin{bmatrix}
k_1 \\
k_2 \\
\vdots \\
k_{n-1} \\
k_n
\end{bmatrix}, \quad c = \begin{bmatrix}
1 \\
1 \\
\vdots \\
1
\end{bmatrix} \]  

(9.35)
9.2.4 Canonical state-space forms for SISO systems

The diagonal canonical form can also be derived directly from another state-space form \((A, B, C, D)\) provided that the eigenvalues of \(A\) are distinct and real.

If \(\lambda_i\) is an eigenvalue and \(t_i\) is the corresponding left eigenvector of \(A\), the following equations apply:

\[
\begin{align*}
t_i^T (\lambda_i I - A) &= 0^T \iff \lambda_i t_i^T = t_i^T A, \quad i = 1, 2, \ldots, n \iff \Lambda T = TA \quad (9.36)
\end{align*}
\]

where

\[
T = \begin{bmatrix}
  t_1^T \\
t_2^T \\
  \vdots \\
t_{n-1}^T \\
t_n^T
\end{bmatrix}, \quad \Lambda = \begin{bmatrix}
  \lambda_1 & 0 & 0 & 0 & 0 \\
  0 & \lambda_2 & 0 & 0 & 0 \\
  0 & 0 & \ddots & 0 & 0 \\
  0 & 0 & 0 & \lambda_{n-1} & 0 \\
  0 & 0 & 0 & 0 & \lambda_n
\end{bmatrix} \quad (9.37)
\]

The matrix \(T\) is always invertible when the eigenvalues are distinct and real (but also in some other cases). The diagonal system matrix \(\Lambda\) is then obtained as

\[
\Lambda = TAT^{-1} \quad (9.38)
\]

which corresponds to the variable transformation

\[
z = Tx, \quad x = T^{-1}z \quad (9.39)
\]
9.2 Linear state-space models

**Controllable canonical form**

A system described by the transfer function (9.29) can be directly written in a state-space form called *controllable canonical form*.

In this form, the parameters of the state-space model

\[
\dot{x}(t) = Ax(t) + bu(t) \\
y(t) = c^T x(t) + du(t)
\]

are

\[
A = \begin{bmatrix}
-a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \\
1 & 0 & \cdots & 0 & 0 \\
0 & 1 & \ddots & 0 & 0 \\
\vdots & \ddots & \ddots & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}, \\
b = \begin{bmatrix}
1 \\
0 \\
\vdots \\
0 \\
b_1 \\
b_2 \\
\vdots \\
b_{n-1} \\
b_n
\end{bmatrix}, \\
c = \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_{n-1} \\
b_n
\end{bmatrix}
\]

Thus, \( A \) is a *sparse matrix* (i.e., a matrix with many zeros) with

- the coefficients of the denominator in (9.29) in the first row (with minus sign)
- a subdiagonal of 1’s directly below the main diagonal

The vector \( c \) contains the coefficients of the numerator of (9.29).
Observable canonical form

A system described by the transfer function (9.29) can be directly written in a state-space form called *observable canonical form*. In this form, the parameters of the state-space model (9.40) are

\[
A = \begin{bmatrix}
-a_1 & 1 & 0 & \cdots & 0 \\
-a_2 & 0 & \ddots & \ddots & 0 \\
\vdots & \vdots & \ddots & 1 & 0 \\
-a_{n-1} & 0 & \cdots & 0 & 1 \\
-a_n & 0 & \cdots & 0 & 0
\end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}
\]

(9.42)

Thus, \( A \) is a *sparse matrix* (i.e., a matrix with many zeros) with

- the coefficients of the denominator in (9.29) in the first column (with a minus sign)
- a subdiagonal of 1’s directly above the main diagonal

The vector \( b \) contains the coefficients of the numerator of (9.29).
9.2.5 Stability, poles, and zeros

Solution of the state-space equation

The general solution of the state-space equation is derived in order to show when the system is stable. The equation

\[ \dot{x}(t) = Ax(t) + Bu(t) \tag{9.43} \]

has the solution

\[ x(t) = e^{At}x(0) + e^{At} \int_0^t e^{-A\tau}Bu(\tau)d\tau \tag{9.44} \]

The solution can be verified by taking the time derivative of (9.44) and substitution of the obtained expression and (9.44) into (9.43). The exponential function \( e^{At} \) is defined via its MacLaurin series as

\[ e^{At} = I + \frac{1}{1!}At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots \tag{9.45} \]

It has previously been shown that

\[ \lambda_i t_i^T = t_i^T A \tag{9.46} \]

where \( \lambda_i \) is the \( i^{th} \) eigenvalue and \( t_i^T \) is the corresponding left eigenvector of \( A \).
9.2.5 Stability, poles, and zeros

Solution of the state-space equation

Multiplication of $e^{At}$ from the left by $t_i^T$ yields

$$t_i^T e^{At} = t_i^T \left( I + \frac{1}{1!} At + \frac{1}{2!} A^2 t^2 + \frac{1}{3!} A^3 t^3 + \cdots \right)$$

$$= t_i^T + \frac{1}{1!} t_i^T At + \frac{1}{2!} t_i^T A^2 t^2 + \frac{1}{3!} t_i^T A^3 t^3 + \cdots$$

$$= t_i^T + \frac{1}{1!} \lambda_i t_i^T t + \frac{1}{2!} \lambda_i t_i^T A t^2 + \frac{1}{3!} \lambda_i t_i^T A^2 t^3 + \cdots$$

$$= t_i^T + \frac{1}{1!} \lambda_i t_i^T t + \frac{1}{2!} \lambda_i^2 t_i^T t^2 + \frac{1}{3!} \lambda_i^2 t_i^T A t^3 + \cdots$$

$$= t_i^T + \frac{1}{1!} \lambda_i t_i^T t + \frac{1}{2!} \lambda_i^2 t_i^T t^2 + \frac{1}{3!} \lambda_i^3 t_i^T t^3 + \cdots$$

$$\vdots$$

$$= t_i^T \left( 1 + \frac{1}{1!} \lambda_i t + \frac{1}{2!} \lambda_i^2 t^2 + \frac{1}{3!} \lambda_i^3 t^3 + \cdots \right) = t_i^T e^{\lambda_i t} \quad (9.47)$$

where (9.46) has been used to successively replace $t_i^T A$ by $\lambda_i t_i^T$.

Equation (9.47) shows that $e^{\lambda_i t}$ is the $i^{th}$ eigenvalue and $t_i^T$ is the corresponding left eigenvector of $e^{At}$. 
9.2.5 Stability, poles, and zeros

Solution of the state-space equation

Multiplication of the solution \( x(t) \) from the left by \( t_i^T \) yields

\[
t_i^T x(t) = e^{\lambda_i t} t_i^T x(0) + e^{\lambda_i t} t_i^T \int_0^t e^{-A\tau} Bu(\tau)d\tau
\]

\[
= e^{\lambda_i t} t_i^T x(0) + e^{\lambda_i t} \int_0^t e^{\lambda_i \tau} t_i^T Bu(\tau)d\tau
\] (9.48)

where (9.47) has been used to replace \( t_i^T e^{A\tau} \) by \( e^{\lambda_i t} t_i^T \).

Using the variable transformation \( z_i = t_i^T x(t), i = 1, \ldots, n \), (9.48) yields

\[
z_i = e^{\lambda_i t} z_i(0) + e^{\lambda_i t} \int_0^t e^{\lambda_i \tau} t_i^T Bu(\tau)d\tau, \quad i = 1, \ldots, n
\] (9.49)

This can be written more compactly as

\[
z(t) = e^{At} z(0) + e^{At} \int_0^t e^{-A\tau} TBu(\tau)d\tau
\] (9.50)

where

\[
A = \begin{bmatrix}
\lambda_1 & 0 & 0 & 0 & 0 \\
0 & \lambda_2 & 0 & 0 & 0 \\
0 & 0 & \ddots & 0 & 0 \\
0 & 0 & 0 & \lambda_{n-1} & 0 \\
0 & 0 & 0 & 0 & \lambda_n
\end{bmatrix}, \quad T = \begin{bmatrix}
t_1^T \\
t_2^T \\
\vdots \\
t_{n-1}^T \\
t_n^T
\end{bmatrix}, \quad z(t) = Tx(t)
\] (9.51)
Stability

In (9.46) and the subsequent derivation of (9.49), no assumptions regarding $\lambda_i$ were made. It could, for example, be a complex eigenvalue.

From (9.49) it is clear that the solution $z_i(t)$ will diverge if $\text{Re}(\lambda_i) > 0$ when $z_i(0) \neq 0$ and the integral is non-zero. This means that the system is unstable.

From this it follows that the eigenvalues $\lambda_i, i = 1, \ldots, n$, which are obtained by solving the characteristic equation

$$\det(\lambda I - A) = 0$$

(9.52)
determine the stability of the system.

The stability criterion is as follows:

- A continuous-time system whose system matrix $A$ has the eigenvalues $\lambda_i, i = 1, \ldots, n$, is stable if and only if all the eigenvalues have negative real part, i.e., iff

$$\text{Re}(\lambda_i) < 0, i = 1, \ldots, n$$

(9.53)
9.2 Linear state-space models

Poles

The eigenvalues $\lambda_i, i = 1, \ldots, n,$ of the system matrix $A$ in a minimal state-space realization are also called system poles. This means that

- the equivalence between eigenvalues and poles requires a controllable and observable minimum-order state-space system
- eigenvalues of uncontrollable and unobservable system parts are not considered to be poles

If the system is known in the form of a transfer function (matrix), it is of practical interest to be able to calculate the system poles directly from the transfer function(s) without first constructing a minimal state-space realization.

**Poles of a scalar transfer function**

The transfer function of a SISO system is factorized as

$$G(s) = \frac{B(s)}{A(s)} = \frac{B(s)}{s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n} = \frac{B(s)}{(s-p_1)(s-p_2)\ldots(s-p_n)}$$

(9.54)

Here, $p_i, i = 1, \ldots, n,$ are the poles of the system. They are identical to the eigenvalues $\lambda_i, i = 1, \ldots, n,$ of a minimal state-space realization.

For real-valued poles,

$$p_i = \lambda_i = -1/T_i$$

(9.55)

where $T_i$ is a time constant of the system.
### Poles of a transfer function matrix

For a MIMO system, i.e., a system with multiple input signals and multiple output signals, it is quite complicated to determine the poles from $G(s)$. However, it is not easy to determine a minimal state-space realization, either.

The following definitions are needed to determine the poles of $G(s)$.

**Subdeterminant.** A subdeterminant of a matrix $A$ is the determinant of a square submatrix of $A$ obtained by removing one or more rows and/or columns of $A$. A maximum subdeterminant is the determinant of a (sub)matrix of maximum size.

**Example.** The matrix $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ has 6 subdeterminants: 1, 2, 3, 4, 5 and 6 as well as $\begin{vmatrix} 1 & 2 \\ 4 & 5 \end{vmatrix} = -3$, $\begin{vmatrix} 1 & 3 \\ 4 & 6 \end{vmatrix} = -6$ and $\begin{vmatrix} 2 & 3 \\ 5 & 6 \end{vmatrix} = -3$.

The latter three are all maximum subdeterminants.
9.2.5 Stability, poles, and zeros

**Pole polynomial.** The pole polynomial \( p(s) \) of a MIMO system with the transfer function matrix \( G(s) \) is the lowest (least) common denominator (LCD) of all sub-determinants of \( G(s) \), including the determinant of \( G(s) \) when \( G(s) \) is square.

**Example.** The transfer function matrix \( G(s) = \begin{bmatrix} \frac{2}{s+1} & \frac{3}{s+2} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix} \) has the subdeterminants \( \frac{2}{s+1}, \frac{3}{s+2} \) and \( \frac{1}{s+1} \) (twice) and also the determinant
\[
\det G(s) = \frac{2}{s+1} \cdot \frac{1}{s+1} - \frac{1}{s+1} \cdot \frac{3}{s+2} = \frac{-s+1}{(s+1)^2(s+2)}.
\]
The pole polynomial (LCD) is thus \( p(s) = (s + 1)^2(s + 2) \).

**Pole theorem.** The poles of \( G(s) \) are the zeros of the pole polynomial \( p(s) \).

**Example.** The transfer function matrix \( G(s) = \begin{bmatrix} \frac{2}{s+1} & \frac{3}{s+2} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix} \) with the pole polynomial \( p(s) = (s + 1)^2(s + 2) \) has the poles \( s = -1 \) (double pole) and \( s = -2 \). The number of poles is not easily seen directly from \( G(s) \).
9.2 Linear state-space models

Zeros
A zero of a SISO system is a value \( s = z \) such that \( G(z) = 0 \). This means that the zeros are the poles of \( 1/G(s) \). For a square MIMO system \( G(s) \), the zeros are the poles of \( G^{-1}(s) \).

A zero in the right-half complex plane (RHP) means that the system is non-minimum phase and thus more difficult to control than a similar minimum-phase system.

Zeros of a scalar transfer function
The transfer function of a SISO system is factorized as

\[
G(s) = \frac{b_{n-m}s^m + b_{n-m+1}s^{m-1} + \ldots + b_{n-1}s + b_n}{A(s)} = \frac{b_{n-m}(s-z_1)(s-z_2)\ldots(s-z_m)}{A(s)}
\]

(9.56)

Here, \( z_i, i = 1, ..., m \), are the zeros of the system.

Real-valued zeros are equal to the negative inverse of numerator time constants of a system.
9.2.5 Stability, poles, and zeros

**Zeros of a transfer function matrix**

In addition to the definitions for pole calculation, the following definition is needed for calculation of the zeros of $G(s)$.

**Zero polynomial.** The zero polynomial of $G(s)$ is the greatest common factor (or divisor, GCF) of the numerators of the maximum (sub)determinants of $G(s)$ when they are normalized to have the pole polynomial $p(s)$ as denominator.

**Zero theorem.** The zeros of $G(s)$ are the zeros of the zero polynomial.

**Observation.** For a square system $G(s)$, the maximum determinant is $\det G(s)$.

**Example.** The transfer function matrix $G(s) = \begin{bmatrix} \frac{2}{s+1} & \frac{3}{s+2} \\ \frac{1}{s+1} & \frac{1}{s+1} \end{bmatrix}$ has the maximum determinant

$$\det G(s) = \frac{2}{s+1} \cdot \frac{1}{s+1} - \frac{1}{s+1} \cdot \frac{3}{s+2} = \frac{-s+1}{(s+1)^2(s+2)}.$$  

Because the denominator is the pole polynomial, it is already normalized. Thus, the zero polynomial is $-s + 1 = 0$, which has the zero $s = 1$. Because $s > 0$, the system might be difficult to control.
9. State-Space Methods

9.3 Control design by state feedback

Because the state vector $x(t)$ contains all information about a system at time $t$, it is natural to (try to) use $x(t)$ for feedback control.

There are various control design methods based on so-called state feedback, e.g.,

- **pole placement**, where the controller is designed to give the closed-loop system specified poles;
- **Linear-quadratic control**, where the control law is derived by minimizing a quadratic performance criterion.

The output equation $y(t) = Cx(t) + Du(t)$ determines which (combination of) states are measured — it is rare that all states can be measured. However, if the system is observable, it is possible to

- **estimate the state** vector using only measurements $y(t)$ and the known state-space model.

For control of systems with stochastic disturbances, stochastic control theory can be used. The state vector is then typically estimated by a so-called Kalman filter. The Kalman filter can also be used for estimation of unknown system parameters.

In this course, stochastic control theory is **not** studied.
9.3.1 Pole placement

**Basic technique**

The control law for a continuous-time state feedback has the form

\[ u(t) = u_r(t) - Kx(t) \]  \hspace{1cm} (9.57)

where \( K \) is a matrix of controller gains and \( u_r(t) \) is a reference signal for the input. The reference signal can be used to define a setpoint \( x_r(t) \) for the state vector via \( u_r(t) = K_r x_r(t) \). In analysis and control design, it is often assumed that \( u_r(t) = 0 \).

Substitution of (9.57) into the state equation (9.43) yields

\[ \dot{x}(t) = (A - BK)x(t) + Bu_r(t) \]  \hspace{1cm} (9.58)

where \( A - BK \) is the system matrix of the controlled system.

Since the eigenvalues of the system matrix determine the dynamics of the system, the dynamics of the closed-loop system are affected in a transparent way by \( K \). In principle, it is possible to choose \( K \) to give the closed-loop system desired poles.
9.3.1 Pole placement

**Example 9.4. Stabilization by pole placement.**

Consider the system
\[
\dot{x}(t) = Ax(t) + Bu(t)
\]
with
\[
A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

This system has the poles (i.e., eigenvalues) \(-1\) and \(+1\). Because one pole is positive, the system is unstable.

We want to stabilize the system by state feedback, and if possible, give the closed-loop system two poles at \(-1\).

There are two states and one input. This means that the state feedback (9.57) must have the form
\[
u(t) = u_r(t) - Kx(t) = u_r(t) - [k_1 \quad k_2]x(t)
\]

Substitution into the state equation yields
\[
\dot{x}(t) = \left( \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \end{bmatrix} [k_1 \quad k_2] \right)x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_r(t)
\]
\[
= \begin{bmatrix} -k_1 & 1 - k_2 \\ 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u_r(t)
\]
The closed-loop poles are found by solving the characteristic equation

\[
\det(\lambda I - (A - BK)) = \det \left( \lambda I - \begin{bmatrix} -k_1 & 1 - k_2 \\ 1 & 0 \end{bmatrix} \right) = \det \begin{bmatrix} \lambda + k_1 & -1 + k_2 \\ -1 & \lambda \end{bmatrix}
\]

\[
= (\lambda + k_1)\lambda + (k_2 - 1) = \lambda^2 + k_1\lambda + k_2 - 1 = 0
\]

Because we want the poles to be \( \lambda_1 = \lambda_2 = -1 \), we want the characteristic equation

\[
(\lambda + 1)^2 = 0 \quad \Leftrightarrow \quad \lambda^2 + 2\lambda + 1 = 0
\]

which is obtained with \( k_1 = 2 \) and \( k_2 = 2 \).

**Restrictions on pole placement**

An interesting question is whether the poles \textit{always} can be placed as desired.

As the following example shows, the answer is \textit{no}!
9.3.1 Pole placement

**Example 9.5. A non-realizable pole placement.**

Consider the system
\[
\dot{x}(t) = Ax(t) + Bu(t)
\]
with
\[
A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\]

This system has the poles \(-1\) and \(+1\) and is thus unstable.

The state feedback has the same form as in Example 9.4. The characteristic equation becomes
\[
\det(\lambda I - (A - BK)) = \det \left( \lambda I - \begin{bmatrix} -1 - k_1 & -k_2 \\ 0 & 1 \end{bmatrix} \right)
\]
\[
= \det \begin{bmatrix} \lambda + 1 + k_1 & k_2 \\ 0 & \lambda - 1 \end{bmatrix} = (\lambda + 1 + k_1)(\lambda - 1) = 0
\]

This means that the closed-loop poles are
\[
\lambda_1 = -1 - k_1 \quad \text{and} \quad \lambda_2 = 1
\]

As can be seen, the unstable pole \(\lambda_2 = 1\) cannot be affected because \(k_2\) has no effect on the pole placement.

Thus, *it is impossible to stabilize this system by state feedback*!
The poles of the system (9.43)

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

can be arbitrarily placed (but complex poles have to be complex-conjugated pairs) with the state feedback (9.57)

\[
u(t) = u_r(t) - Kx(t)
\]

if and only if the system is controllable. The system is controllable if and only if the controllability matrix (9.17)

\[
\Gamma_c \equiv \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}
\]

has full rank, i.e., \( \text{rank} \Gamma_c = n \) when \( A \) is an \( n \times n \) matrix.

Note, however, that the requirement to place the poles arbitrarily is unnecessarily restrictive. In practice, the poles are placed in certain region of the left-half complex plane (LHP).

If there are uncontrollable, unstable states, stable closed-loop operation cannot be achieved. If the uncontrollable states are stable, pole placement can be used for the poles associated with the controllable states.
9.3 Control design by state feedback

### Choice of pole placement

As a “rule of thumb”,
- the poles should be placed in the dashed area in the complex LHP
- the area close to the origin should generally be avoided even in the dashed area

The dashed area corresponds to eigenvalues of the form

\[
\lambda_i = \mu_i \pm j\omega_i \\
\mu_i < 0, \quad 0 \leq \omega_i \leq -\mu_i
\]  

(9.59)

The placement of the poles affects the system behaviour as follows:
- The distance of the poles from the origin determine the dynamics of the system — a longer distance means faster dynamics (smaller time constant).
- Complex poles result in a step response with overshoot and/or oscillations.
- For a closed-loop system of second order, the relative damping \( \zeta \approx 0.7 \) yields a “nice” step response. This corresponds to \( \omega_i = -\mu_i = 0.5\sqrt{2}\omega_n \approx 0.7\omega_n \), where \( \omega_n \) is the undamped natural frequency of the system.
- For systems of “high” order, the dominant dynamics are determined by the poles closest to the origin.
- Poles far from the origin mean fast dynamics, but also large input changes.
Step-response specifications

Often it is desired that the closed-loop system approximately behaves as a slightly underdamped second-order system.

A second-order SISO system with the steady-state gain 1, which is desired of the transfer function for a closed-loop system, has the transfer function

\[ G(s) = \frac{\omega_n^2}{s^2 + 2\zeta \omega_n s + \omega_n^2} \]  

(9.60)

with parameters as defined in Chapter 5. This transfer function has the poles

\[ \lambda = -\zeta \omega_n \pm j\beta \omega_n, \quad \beta = \sqrt{1 - \zeta^2} \]  

(9.61)

If \( \zeta \) and \( \omega_n \) are specified, the desired poles can thus be calculated by (9.61).

In practice, \( \zeta \) and \( \omega_n \) are often not specified directly, but instead parameters related to a step response are specified, such as

- maximum relative overshoot, \( M = (y_{\text{max}}/y_{\infty}) - 1 \)
- rise time \( t_r \), which is the time it takes to pass \( y_{\infty} \) for the first time

The relative damping and the undamped natural frequency are then given by

\[ \zeta = \frac{-\ln M}{\sqrt{\pi^2 + \ln^2 M}}, \quad \omega_n = \frac{\pi - \arctan(\beta/\zeta)}{\beta t_r}, \quad \beta = \sqrt{1 - \zeta^2} \]  

(9.62)
9.3.1 Pole placement

**Example 9.6. Step-response specifications for an electrical servo motor.**

The dynamics of an electrical servo motor are described by the transfer function

\[ Y(s) = \frac{1}{s(s+1)} U(s) \]

where the input is an applied voltage and the output is the drive shaft position. The time unit is seconds. The system has the state-space representation

\[
\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)
\]

The motor is to be controlled by state feedback to give a closed-loop system with

- relative damping \( \zeta \approx 0.7 \), rise time \( t_r \approx 0.8 \) sek
- no steady-state control error

Eq. (9.62) gives \( \omega_n \approx 4.21 \) and (9.61) the desired poles \( \lambda \approx -3 \pm 3j \).

The control law has the form

\[ u(t) = u_r(t) - Kx(t) = k_r r(t) - [k_1 \quad k_2] x(t) \]

where \( r(t) \) is the setpoint for \( y(t) \).

Note that this control law is different from a simple feedback of \( y(t) \) according to \( u(t) = k(r(t) - y(t)) \).
Step-response specifications

Example 9.6

Substitution of the control law into the state equation yields

\[
\dot{x}(t) = \begin{bmatrix} 0 \\ -k_1 \\ -1 - k_2 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} k_r r(t)
\]

The transfer function for the close-loop system becomes

\[
G_c(s) = \frac{Y(s)}{R(s)} = \left[1 \quad 0\right] \left(sI - \begin{bmatrix} 0 \\ -k_1 \\ -1 - k_2 \end{bmatrix}\right)^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} k_r = \frac{k_r}{s^2 + (1+k_2)s + k_1}
\]

The specifications mean that the controller parameters should be selected as

\[
k_1 = \omega_n^2 \approx 18, \quad k_2 = 2\zeta\omega_n - 1 \approx 5 \quad \text{and} \quad k_r = k_1 = 18
\]

The full line (——) in the figure shows the output \(y(t)\) when the setpoint is changed as a step from \(r = 0\) to \(r = 1\).

The dashed line (– – –) shows the result if both poles are placed at \(\lambda = -\sqrt{18}\). This is obtained with \(k_2 = 2\sqrt{18} - 1 \approx 7.5\).
Some drawbacks with pole placement

The design method seems attractive, but in reality it is not very practical. There are also theoretical shortcomings.

- State feedback normally results in a **PD controller** because the states often represent outputs and their time derivatives. Thus, a **stationary control error** can be expected after some disturbance.
  
  In the example, a correct setpoint change was achieved, but some other disturbance would give a steady-state error.

- State feedback **does not affect the location of zeros**. Thus, performance limiting **RHP zeros cannot be removed**. However, this is a problem also with many other model-based control design methods.

- State feedback uses the full state vector, but **usually all states cannot be measured**. In practice, unmeasured states have to be estimated from measurements in combination with the process model.
9.3.2 Linear-quadratic control

Introduction

By pole placement, the poles of the controlled system can be specified, but the control actions (signal sizes) in terms of inputs, outputs and state variables cannot be directly affected.

This issue can be tackled by linear-quadratic (LQ) control. The controller is obtained as the solution to an optimization problem, where

- a quadratic objective function in terms of state variables and control variables (inputs) is minimized
- subject to a linear constraint defined by a linear state-space model.

The optimization with a quadratic objective function and a linear constraint has an (implicit) analytic solution.

The resulting controller is a state-feedback controller, but modifications are possible to handle the drawbacks with pole placement.
Problem formulation and solution

In the basic formulation, the linear constraint is given by the state equation

\[ \dot{x}(t) = Ax(t) + Bu(t) \]  \hspace{1cm} (9.63)

where the state variables are defined as deviations from the desired state. This means that \( x = 0 \) is the desired state.

Assume that at time \( t = 0 \), \( x(0) = x_0 \neq 0 \). It is desired to reach the state \( x = 0 \) in such a way that the cost function

\[ J = \int_0^\infty [x(t)^TQ_xx(t) + u(t)^TQ_uu(t)] \, dt \]  \hspace{1cm} (9.64)

is minimized. Here \( Q_x \) and \( Q_u \) are symmetric weight matrices such that

- \( Q_x \) is positive semidefinite, i.e., \( x(t)^TQ_xx(t) \geq 0 \) for any \( x(t) \neq 0 \)
- \( Q_u \) is positive definite, i.e., \( u(t)^TQ_uu(t) > 0 \) for any \( u(t) \neq 0 \)

Mathematically, the properties of \( Q_x \) and \( Q_u \) can be expressed as \( Q_x = Q_x^T \succeq 0 \) and \( Q_u = Q_u^T > 0 \).

Usually \( Q_x \) and \( Q_u \) are diagonal matrices that weight the squares \( x_i(t)^2, i = 1, ..., n \) and \( u_j(t)^2, j = 1, ..., m \).
9.3.2 Linear-quadratic control

The system (9.63) is *stabilizable* by state feedback if there exists a matrix $K$ such that all eigenvalues of the matrix $A - BK$ have negative real part. If the system is stabilizable, the solution to the optimization problem is

$$ u(t) = -Kx(t), \quad K = Q_u^{-1}B^TP $$

(9.65)

where $P$ is a symmetric positive definite matrix (i.e., $P = P^T > 0$), which is obtained as the unique solution to the *algebraic Riccati equation*

$$ A^TP + PA + Q_x - PBQ_u^{-1}B^TP = 0 $$

(9.66)

The relationship $u(t) = -Kx(t)$ implies that the optimal solution is that, in general, *all state variables should be fed back.*

Unfortunately, there is no general closed-form solution for $P$. In practice, $P$ has to be determined *numerically* for systems of higher order than 2. In MATLAB, the command `lqr` (or LQR) is available (type `>>help lqr` for a description; note that the symbols used in `lqr` are different from those used here).

The fact that the integration in (9.64) is from $t = 0$ to $t = \infty$ causes $P$, and thus also $K$, to be constant matrices.

If the integration were up to some finite time $t = t_f$, the matrices would be time dependent and the *control law would be time variant.*
Example 9.7. Linear-quadratic control of a double integrator.

An electric direct current (DC) motor can be modelled as a double integrator by the state-space model

\[ \dot{x}(t) = Ax(t) + Bu(t), \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]

It is desired to control the motor such that the cost function

\[ J = \int_0^\infty [x(t)^T Q_x x(t) + u(t)^T Q_u u(t)] \, dt, \quad Q_x = \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix}, \quad Q_u = 1 \]

is minimized. Given that \( u \) here is a scalar variable, the cost function becomes

\[ J = \int_0^\infty (4x_1^2 + u^2) \, dt \]

This means that the state variable \( x_2 \) (which correspond to the angular velocity of the motor shaft) is not weighted. Thus, any value for \( x_2 \) is accepted. However, \( x_1 \) (the angular position of the shaft) and \( u \) (the current used to control the motor) are weighted.
Problem formulation and solution

The optimal control law is given by (9.65) where $P$ is obtained by solving (9.66). From the matrix multiplications in (9.66) it follows that $P$ has the same size as $A$. Because $P$ is a symmetric matrix, it can in this case be expressed as

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$$

Substitution into $A^T P + PA + Q_x = PBQ_u^{-1} B^T P$ gives

Left side: $\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} + \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 4 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 4 & p_{11} \\ p_{11} & 2p_{12} \end{bmatrix}$

Right side: $\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot 1 \cdot \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} p_{12}^2 & p_{12}p_{22} \\ p_{12}p_{22} & p_{22}^2 \end{bmatrix}$

Equating the two right-hand matrices gives the three equations

$$p_{12}^2 = 4, \quad p_{11} = p_{12}p_{22}, \quad 2p_{12} = p_{22}^2$$

A necessary requirement for $P$ to be positive definite is that the diagonal elements are positive. The only solution is then

$$p_{11} = 4, \quad p_{12} = 2, \quad p_{22} = 2, \quad \text{i.e.,} \quad P = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}$$
Problem formulation and solution

The optimal feedback matrix is thus

\[ K = Q_u^{-1} B^T P = 1 \cdot [0 \ 1] \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix} = [2 \ 2] \]

and the control law is

\[ u(t) = -2x_1(t) - 2x_2(t), \text{ or } u(t) = 2(r - x_1(t)) - 2x_2(t) \]

where \( r \) is the setpoint for \( x_1 \) (note that there is no setpoint for \( x_2 \)).

Which are the poles of the controlled system? According to (9.58), \( A - BK \) is the system matrix of the controlled system. The eigenvalues of this matrix yields the poles. We have

\[ A - BK = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - 0 \begin{bmatrix} 2 & 2 \\ -2 & -2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix} \]

The eigenvalues of are the solution to

\[ \det \left( \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} \right) = \begin{vmatrix} \lambda - 1 \\ 2 & \lambda + 2 \end{vmatrix} = 0 \]

\[ \Leftrightarrow \lambda(\lambda + 2) + 2 = \lambda^2 + 2\lambda + 2 = 0 \]

\[ \Leftrightarrow \lambda = -1 \pm \sqrt{(-1)^2 - 2} = -1 \pm j \]

The system is stable and the poles are placed according to recommendations.
9.3 Control design by state feedback

Integral action

The control law (9.65) is optimal for moving a system from an initial state $x_0 \neq 0$ to a final state $x_\infty = 0$ if there are no unmodelled disturbances. For non-vanishing disturbances, integral action is usually required to achieve no steady-state control error.

In the following, the problem formulation is modified to result in a controller with integral action. Such a controller can handle both setpoint changes and unknown disturbances without steady-state control error.

In the derivation of the control law, a constant input disturbance $d$ is assumed. The state-space model can then be written

$$\dot{x}(t) = Ax(t) + Bu(t) + Md$$
$$y(t) = Cx(t) + Du(t) + Ed$$  

(9.67)

A new variable is defined by

$$q(t) = \int_0^t (y(t) - r)dt$$  

(9.68)

where $r$ is a constant setpoint for the output $y(t)$. This means that $q(t)$ is the (negative) integrated control error. The time derivative of $q(t)$ is

$$\dot{q}(t) = y(t) - r = Cx(t) + Du(t) + Ed - r$$  

(9.69)
9.3.2 Linear-quadratic control

When the first equation of (9.67) and (9.69) are differentiated (once more) with respect to time, the constant terms will disappear. This gives an extended state-space model

\[ \ddot{x}(t) = A\dot{x}(t) + B\dot{u}(t) \quad \text{or} \quad \ddot{q}(t) = C\dot{x}(t) + D\dot{u}(t) \]

(9.70)

With the definitions

\[ w(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix}, \quad v(t) = \dot{u}(t), \quad \widehat{A} = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix}, \quad \widehat{B} = \begin{bmatrix} B \\ D \end{bmatrix} \]

(9.71)

(9.70) can be written compactly as

\[ \dot{w}(t) = \widehat{A}w(t) + \widehat{B}v(t) \]

(9.72)

Minimization of the loss function

\[ J = \int_0^\infty [w(t)^TQ_w w(t) + v(t)^TQ_v v(t)] \, dt \]

(9.73)

now gives the control law

\[ v(t) = -Kw(t), \quad K = Q_v^{-1}\widehat{B}^T P \]

(9.74)

where \( P \) is obtained by solving the algebraic Riccati equation

\[ \widehat{A}^T P + P\widehat{A} + Q_w - P\widehat{B}Q_v^{-1}\widehat{B}^T P = 0 \]

(9.75)
9.3.2 Linear-quadratic control

The partitioning $K = [K_1 \quad K_2]$ and the variable definitions (9.71) yield

$$\dot{u}(t) = -K_1 \dot{x}(t) - K_2 \dot{q}(t) = -K_1 \dot{x}(t) + K_2 (r - y(t)) \quad (9.76)$$

Integration from time $\tau = 0$ to $\tau = t$ with initial values $x(0) = 0$ and $u(0) = 0$ gives

$$u(t) = -K_1 x(t) + K_2 \int_0^t (r(\tau) - y(\tau)) d\tau \quad (9.77)$$

where the setpoint now is allowed to be time varying. This is a multivariable control law with integral action that causes $y(t) \to r(t_s)$ if $r(t) = r(t_s) = \text{constant for } t > t_s$.

A requirement for this control law to stabilize the system $(A, B)$ is that the system is controllable (i.e., the controllability matrix $\Gamma_c$ has full rank) and that the matrix $[A \quad B \quad C \quad D]$ has full rank.

In the loss function (9.73), $Q_w$ usually has the form $Q_w = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$. This means that

- $Q_1$ is a weight for $\dot{x}(t)$, i.e., movements of $x(t)$
- $Q_2$ is a weight for the control error $r(t) - y(t)$, i.e., the integral action
- $Q_v$ is a weight for $\dot{u}(t)$, i.e., movements of $u(t)$
9.3.3 State estimation

**Introduction**

Assume that the matrices of the state-space model (9.63) are known. One way of estimating \( x(t) \) is to simulate the system

\[
\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t), \quad \hat{x}(0) = \hat{x}_0
\]  

(9.78)

in real time. Here, \( \hat{x}(t) \) is the estimate of \( x(t) \), \( u(t) \) is a known input (control signal), and \( \hat{x}_0 \) is an estimate of \( x(0) \) when the simulation is started.

A problem with this approach is that the true initial state \( x(0) = x_0 \) is unknown. There is no guarantee that \( \hat{x}(t) \) is close to \( x(t) \) as time increases, especially if \( \hat{x}_0 \neq x_0 \).

Another drawback is that measured outputs \( y(t) \) are not used in (9.78). This means that all available information is not used, which is unlikely to be optimal.
9.3 Control design by state feedback

**A state observer**

The estimation can be improved by utilizing measured outputs $y(t)$. If the estimate $\hat{x}(t)$ is available, the outputs can be estimated by

$$\hat{y}(t) = C\hat{x}(t) + Du(t) \quad (9.79)$$

The difference

$$y(t) - \hat{y}(t) = y(t) - C\hat{x}(t) - Du(t) \quad (9.80)$$

$$= C(x(t) - \hat{x}(t)) \quad (9.81)$$

can be taken as a measure of how well $\hat{x}(t)$ estimates $x(t)$. The estimate $\hat{x}(t)$ can be improved by adjusting the estimate by the difference $y(t) - \hat{y}(t)$ according to

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + H(y(t) - C\hat{x}(t) - Du(t)) \quad (9.82)$$

where $H$ is the observer gain matrix. The term $H(y(t) - C\hat{x}(t) - Du(t))$ can be interpreted as feedback of the estimation error similarly as the control error is fed back in a feedback controller. This kind of estimator is called a state observer.

The estimate $\hat{x}(t)$ is used instead of $x(t)$ in a state-feedback controller, i.e.,

$$u(t) = -K\hat{x}(t) \quad (9.83)$$

This means that the controller + state observer only uses the output $y(t)$, not the state $x(t)$.
9.3.3 State estimation

**Convergence properties**

The estimation error of $x(t)$ is denoted

$$\delta x(t) = x(t) - \hat{x}(t)$$  \hspace{1cm} (9.84)

Differentiation of (9.84) with respect to time gives

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

$$-A\hat{x}(t) - Bu(t) - H(y(t) - C\hat{x}(t) - Du(t))$$

$$= A(x(t) - \hat{x}(t)) - H(Cx(t) - C\hat{x}(t)) = (A - HC)\delta x(t)$$

Thus, the estimation error is governed by the differential equation

$$\delta \dot{x}(t) = (A - HC)\delta x(t)$$ \hspace{1cm} (9.85)

where the matrix $(A - HC)$ determines the convergence properties.

- The estimator is stable and the estimation error converges towards zero if, and only if, all eigenvalues of $(A - HC)$ have negative real part.
- The more negative the real part of the eigenvalues, the faster convergence.
- The eigenvalues of $(A - HC)$ can be placed arbitrarily by means of $H$ if the system $(A, C)$ is observable.

Is it reasonable to try to achieve very fast convergence?
Example 9.8. The dynamics of a state observer

State estimation of the electrical servo motor in Example 9.6 is considered. The state-space model of the motor is

\[
\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \quad y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)
\]

If only the output \( y(t) \) is measured, the states can be estimated by the observer

\[
\dot{\hat{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) + \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} (y(t) - \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{x}(t))
\]

The system matrix \( (A - HC) \) for the estimation error of the observer is

\[
A - HC = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ -1 \end{bmatrix}
\]

The characteristic equation of \( A - HC \) is thus

\[
\det \left( \lambda I - \begin{bmatrix} h_1 & 1 \\ h_2 & -1 \end{bmatrix} \right) = \begin{vmatrix} \lambda + h_1 & -1 \\ h_2 & \lambda + 1 \end{vmatrix} = \lambda^2 + (1 + h_1)\lambda + h_1 + h_2 = 0
\]

Desired eigenvalues \( \lambda_1 \) and \( \lambda_2 \) can be obtained by the selection of \( h_1 \) and \( h_2 \).
A state observer

The plots show how well the state $x_2$ is estimated for two different choices of observer eigenvalues when $x_2$ changes as shown in the diagrams. The initial state estimate $\hat{x}_2(0) = 0$ is used, but in reality the (unknown) initial state is $x_2(0) = 1$.

The upper diagram shows the estimate $\hat{x}_2$ as function of time when the eigenvalues are $\lambda = -4 \pm 4j$, which is obtained by $h_1 = 7$ and $h_2 = 25$.

In the lower diagram, the eigenvalues are $\lambda = -15 \pm 15j$, which is obtained by $h_1 = 29$ and $h_2 = 421$.

It can be seen that better state reconstruction is obtained in the lower diagram, where the observer eigenvalues are further to the left in the complex plane.
A state observer

How well does the estimation result hold in practice?

Assume that the measured output $y(t)$ contains high-frequency noise. If the frequency is 50 rad/time unit and the amplitude is 0.05 the measured output is

$$y(t) = x_1(t) + 0.05 \sin 50t$$

The figures show the estimation result with the same observers as above (upper diagram $\lambda = -4 \pm 4j$, lower diagram $\lambda = -15 \pm 15j$).

Here it is clear that the slower observer is better.

**Note.** In the construction of the observer it was not taken into account that $x_1$ is measured. Thus $x_1$ is also estimated (but not shown in the figures). An observer that only estimates unmeasured states is called a reduced (Luenberger) observer.