SYSTEM THEORY
STATE SPACE ANALYSIS
AND
CONTROL THEORY

Lecture Notes in Control Theory

PhD (Dr. ing.)
David Di Ruscio

Master study in Systems and Control Engineering
Faculty of Technology
Telemark University College

August 1996

August 29, 2017

Report:
Systems and Control Engineering
Faculty of Technology
Telemark University College
Kjølnes ring 56
N-3914 Porsgrunn

Preface

These lecture notes is meant to be used in the Control Theory part of the course SCE1106 which is to be held for the master study in Systems and Control Engineering. The contents is also basic theory for courses as System Identification and Advanced Control theory.

The following words should be noted

All this –

was for you, dear reader,
I wanted to write a book
that you would understand.

For what good is it to me
if you can't understand it ?

But you got to try hard —

This verse is from Kailath (1980).

Contents

Part I

State Space Analysis

Chapter 1

Basic System Theory

1.1 Models of dynamic systems

The aim of this section is not to discuss modeling principles of dynamic systems in detail. However we will in this introductory section mention that dynamic models may be developed in many ways. For instance so called first principles methods as mass balances, force balances, energy balances, i.e., conservation of law methods, leads to ether non-linear models of the type

$$\begin{aligned} \dot{x} &= f(x,u) && (1.1) \\ y &= g(x) && (1.2) \end{aligned}$$

or linear or linearized models of the type

$$\begin{aligned} \dot{x} &= Ax + Bu && (1.3) \\ y &= Dx && (1.4) \end{aligned}$$

Note also that a linearized approximation of the non-linear model usually exist. We will in the following give a simple example of a system which may be described by a linear continuous time state space model

Example 1.1 (Model of a damped spring system)
Assume given an object with mass, m, influenced by three forces. One force F_1 used to pull the mass, one force $F_2 = kx$ from the spring and one force $F_3 = \mu\dot{x} = \mu v$ that represents the friction or viscous damping.

We define x as the position of the object and $\dot{x} = v$ as the velocity of the object. Furthermore the force F_1 may be defined as a manipulable control input variable and we use u as a symbol for this control input, i.e., $u = F_1$.

from this we have the following force balance

$$ma = m\dot{v} = \sum_{i=1}^{3} F_i = F_1 - F_2 - F_3 = -kx - \mu v + u \qquad (1.5)$$

The model for the damped spring system consists of two continuous time ordinary differential equations. Those two ODEs may be written in standard state space form *as follows*

$$\overbrace{\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix}}^{\dot{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\mu}{m} \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x \\ v \end{bmatrix}}^{x} + \overbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}^{B} u \tag{1.6}$$

Modeling from first principles, e.g., as the in the damped spring example above, often leads to a standard linear continuous time state space model on the form

$$\dot{x} = Ax + Bu \tag{1.7}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^r$ is the control input vector, $A \in \mathbb{R}^{n\ timesn}$ is state matrix and $B \in \mathbb{R}^{n\ timesr}$ is the control input matrix.

1.2 Linear Time State Space Models

An important class of state space models is the time invariant linear and continuous time state space model of the form

$$\dot{x} = Ax + Bu, \qquad x(0) = x_0, \tag{1.8}$$

$$y = Dx, \tag{1.9}$$

where $u \in \mathbb{R}^r$ is the control vector, $x \in \mathbb{R}^n$ is the state vector, $y \in \mathbb{R}^m$ is the measurements vector and $x_0 = x(t_0) \in \mathbb{R}^n$ is the initial value of the state vector, which usually is assumed to be known. Time invariant means that the model matrices A, B and D are constant matrices, i.e. time invariant.

It can be shown that the exact solution of the state equation (1.8) at time $t_0 \leq t$ is given by

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{1.10}$$

As we see, the solution consists of two parts. The first part represents the autonomous response (homogenous solution) driven only by initial values different from zero. The second term represents the in homogenous solution driven by the control variable, $u(t)$.

In order to compute the first term we have to compute the matrix exponential $e^{A(t-t_0)}$. This matrix exponential is defined as the transition matrix, because it defines the transition of the state from the initial value, $x(t_0)$, to the final state $x(t)$ in an autonomous system $\dot{x} = Ax$ with known initial state $x(t_0)$. The transition matrix is defined as follows

$$\Phi(t) \overset{\text{def}}{=} e^{At}. \tag{1.11}$$

Using this definition of the transition matrix we see that the solution (1.10) can be written as follows

$$x(t) = \Phi(t-t_0)x(t_0) + \int_{t_0}^{t} \Phi(t-\tau)Bu(\tau)d\tau. \tag{1.12}$$

The second term in the solution (1.10) (or equivalent as in (1.12)) consists of a convolutional integral. This integral must usually be computed numerically, e.g. it is usually hard to obtain an analytically solution. However, an important special case is the case where the control $u(\tau)$ is constant over the integration interval $t_0 < \tau \leq t$.

$$x(t) = \Phi(t - t_0)x(t_0) + \Delta u(t_0), \tag{1.13}$$

where Δ is shown to be

$$\Delta = \int_{t_0}^{t} e^{A(t-\tau)} B d\tau = \int_{0}^{t-t_0} e^{A\tau} B d\tau \tag{1.14}$$

Note also that

$$\Delta = A^{-1}(e^{A(t-t_0)} - I)B, \tag{1.15}$$

when A is non singular. It is this solution which usually is used in order to compute the general solution to the state equation. Hence, the control input $u(t)$ is assumed to be constant over piece wise identical intervals $\Delta t = t - t_0$.

The constant interval Δt is in control theory and control systems defined as the sampling time in the digital controller. If we now are putting $t = t_0 + \Delta t$ in the solution (1.13) then we get

$$x(t_0 + \Delta t) = \Phi(\Delta t)x(t_0) + \Delta u(t_0), \tag{1.16}$$

where Δ is given by

$$\Delta = A^{-1}(e^{A\Delta t} - I)B. \tag{1.17}$$

The solution given by (1.16) and (1.17) is the starting point for making a discrete time state space model for the system. In digital control systems discrete time models are very important. Discrete time models are also very important for simulation purposes of a dynamic system.

Consider now the case where we let t_0 in (1.16) and (1.17) take the discrete time values

$$t_0 = k\Delta t \quad \forall \quad k = 0, 1, \ldots, \tag{1.18}$$

We then have a discrete time model of the form

$$x((k+1)\Delta t) = \Phi(\Delta t)x(k\Delta t) + \Delta u(k\Delta t), \tag{1.19}$$

It is however common to use the short hand notation

$$x_{k+1} = \Phi x_k + \Delta u_k. \tag{1.20}$$

We have here defined

$$x_k = x(k\Delta t) = x(t_0) \tag{1.21}$$

Note also that we usually are using symbols as A and B also for discrete time state space models, e.g., so that the model (1.20) is written as

$$x_{k+1} = Ax_k + Bu_k. \tag{1.22}$$

we will usually using the symbols A and B also for discrete time models. However, in cases where there can be conflicts symbols as Φ and Δ are used.

It is important to note that the steady state solution to a continuous time model $\dot{x} = Ax + Bu$ can be found by putting $\dot{x} = 0$. I.e., the steady state solution when time approach infinity ($t \to \infty$) is given by

$$x = -A^{-1}Bu. \tag{1.23}$$

Here the system matrix A is assumed to be non singular.

In a stable system, the transients and dynamic responses will die out as time approach infinity, and al variables will be constant as function of time. Therefore is also the derivative of the states with time equal to zero, i.e.

$$\dot{x} = \frac{dx}{dt} = 0 \tag{1.24}$$

Note also that the steady state solution of a continuous time model and the discrete time model should be the same. This is obvious

1.2.1 Proof of the solution of the state equation

It can be shown that the homogenous solution to the state equation $\dot{x} = Ax + Bu$ (with known initial state $x(t_0)$) when $u = 0$ is of the form

$$x(t) = e^{A(t-t_0)}z \tag{1.25}$$

because $dx/dt = Ae^{A(t-t_0)}z = Ax$.

The solution of the in homogenous differential equation can be found by assuming that the vector z is time variant. We then have from (1.25) that

$$\dot{x} = Ae^{A(t-t_0)}z + e^{A(t-t_0)}\dot{z} \tag{1.26}$$

We also have from the state equation that

$$\dot{x} = Ax + Bu = Ae^{A(t-t_0)}z + Bu \tag{1.27}$$

where we have used that x is given as in (1.25). Comparing (1.26) and (1.27) shows that

$$e^{A(t-t_0)}\dot{z} = Bu. \tag{1.28}$$

This gives

$$\dot{z} = e^{-A(t-t_0)}Bu \tag{1.29}$$

where we have used that $(e^A)^{-1} = e^{-A}$. This gives the following solution for the vector z, i.e.,

$$z(t) = z_0 + \int_{t_0}^{t} e^{-A(\tau - t_0)} B u d\tau. \tag{1.30}$$

We are putting (1.30) in (1.25). This gives

$$x(t) = e^{A(t-t_0)} z = e^{A(t-t_0)} z_0 + \int_{t_0}^{t} e^{A(t-\tau)} B u d\tau. \tag{1.31}$$

Putting $t = t_0$ shows that $x(t_0) = z_0$ and the final solution is found to be

$$x(t) = e^{A(t-t_0)} x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)} B u d\tau. \tag{1.32}$$

This section is meant to show how the solution to the continuous time state equation can be proved.

1.3 Linear transformation of state space models

Let x be the state vector in the state space realization (A, B, D) such that

$$\begin{aligned} \dot{x} &= Ax + Bu, \qquad x(0) = x_0 \qquad (1.33) \\ y &= Dx \qquad (1.34) \end{aligned}$$

An equivalent realization of the system defined by (1.33) and (1.34) can be found by choosing another basis for the state space (choosing another state). The state vector x can be transformed to a new coordinate system. This can be done by defining a non singular transformation matrix $T \in \mathbb{R}^{n \times n}$ and the following linear transformation,

$$x = Tz \quad \Leftrightarrow \quad z = T^{-1}x \tag{1.35}$$

An equivalent realization of the system (1.33) and (1.34) is then given by

$$\begin{aligned} \dot{z} &= T^{-1}ATz + T^{-1}Bu, \qquad z(0) = T^{-1}x_0 \qquad (1.36) \\ y &= DTz \qquad (1.37) \end{aligned}$$

These two state space realizations can be shown to be identical and represent the same system. The two systems has the same transfer function from the input, u to the output, y.

An infinite number of non singular transformation matrices T can be chosen. This leads to an infinite number of state space model realizations. Some of these realizations has special properties, e.g., state space models with special properties can be found by choosing T properly.

1.4 Eigenvalues and eigenvectors

Consider given a matrix $A \in \mathbb{R}^{n\times n}$. The characteristic polynomial of A is then defined as

$$\begin{aligned} p(A) &= \det(\lambda I - A) = \det(A - \lambda I) && (1.38) \\ &= \lambda^n + p_n\lambda^{n-1} + \cdots + p_2\lambda + p_1 && (1.39) \end{aligned}$$

where the n coefficients $p_1, p_2, \ldots, p_{n-1}, p_n$ are real values. These coefficients can be found by actually expressing the determinant. The *characteristic equation* is defined as

$$p(A) = \det(\lambda I - A) = 0. \tag{1.40}$$

The n roots of the character polynomial and equivalently, the n solutions to the characteristic equation, $\lambda_i \ \forall \ i = 1, \ldots, n$, is defined as the eigenvalues of the matrix A. The characteristic equation has always n solutions and the matrix $A \in \mathbb{R}^{n\times n}$ has always n eigenvalues. The eigenvalues can be real or complex. If the eigenvalues are complex, then they will consists of complex conjugate pair, i.e., if $\lambda_k = \alpha + j\beta$ is an eigenvalue, then $\lambda_{k+1} = \alpha - j\beta$ will also be an eigenvalue. The matrix A is said to have multiple eigenvalues if two or more of the eigenvalues are identical.

The spectrum of the eigenvalues of the matrix A is defined as all the eigenvalues, i.e. the collection $\sigma(A) := \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ of al the eigenvalues is the spectrum of A.

The spectral radius of the matrix A is defined from the eigenvalue with the largest absolute value, i.e., $\rho(A) = \max|\lambda_i| \ \forall \ i = 1, \ldots, n$.

The following theorem is useful in linear algebra and system theory

Teorem 1.4.1 (Cayley-Hamilton) *Given a matrix $A \in \mathbb{R}^{n\times n}$ and the characteristic polynomial*

$$p(A) = det(\lambda I - A) = \lambda^n + p_n\lambda^{n-1} + \cdots + p_2\lambda + p_1. \tag{1.41}$$

The Cayley-Hamilton theorem states that the matrix A satisfies its own characteristic polynomial, i.e., such that

$$A^n + p_nA^{n-1} + \cdots + p_2A + p_1I = 0 \tag{1.42}$$

where I is the identity matrix with the same dimension as A.

Proof The prof is stated only for the case in which the eigenvector matrix M of A is non-singular. An eigenvalue decomposition of A gives that $A = M\Lambda M^{-1}$ where Λ is a diagonal matrix with the eigenvalues $\lambda_i \ \forall \ i = 1, \ldots, n$ on the diagonal. Putting this into (1.42) gives

$$M(\Lambda^n + p_n\Lambda^{n-1} + \cdots + p_2\Lambda + p_1I)M^{-1} = 0. \tag{1.43}$$

This gives n equations, i.e.,

$$\lambda_i^n + p_n\lambda_i^{n-1} + \cdots + p_2\lambda_i + p_1 = 0 \ \forall \ i = 1, \ldots, n \tag{1.44}$$

which is of the same form as the characteristic equation (1.41). △

The Cayley-Hamilton theorem can be used for, e.g.:

- to find an expression for the matrix inverse A^{-1}, i.e.
$$A^{-1} = -\frac{1}{p_1}(A^{n-1} + p_n A^{n-2} + \cdots + p_2 I) \tag{1.45}$$
- to find an expression for the power A^n as a function of $A^{n-1}, \cdots, A$, i.e.
$$A^n = -(p_n A^{n-1} + \cdots + p_2 A + p_1 I) \tag{1.46}$$
- to find a way of computing the coefficients $p_1, \cdots, p_n$ of the characteristic polynomial by Krylovs method. This is presented in the next section.
- develop the controllability and observability matrices of an linear dynamical system, i.e. from the matrix pairs (A, B) and (A, D).

1.4.1 Krylovs method used to find the coefficients of the characteristic equation

We will in this section study a method which can be used to compute the coefficients, $p_1, \cdots, p_n$, in the characteristic polynomial of a $n \times n$ matrix A. This method is referred to as Krylovs method, Krylov (1931).

If we multiply Equation (1.42) from right with a vector $b \in \mathbb{R}^n$, then a linear system of equations can be defined as

$$\overbrace{\begin{bmatrix} b & Ab & \cdots & A^{n-1}b \end{bmatrix}}^{C_n} \overbrace{\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}}^{p} = -A^n b. \tag{1.47}$$

This equation, $C_n p = -A^n b$, can be solved with respect to the vector p of coefficients. We have that

$$p = -C_n^{-1} A^n b \tag{1.48}$$

if the vector b is chosen in such a way that the matrix pair (A, b) is controllable, i.e.. in this case in such a way that the controllability matrix C_n is invertible (non-singular).

An arbitrarily random vector b is here usually sufficient. Note that the solution p generally is independent of the choice of b as long as the matrix C_n is invertible.

Note also that (1.42) can directly be written as the linear system of equations

$$\mathcal{A} p = -\mathrm{vec}(A^n) \tag{1.49}$$

where

$$\mathcal{A} = \begin{bmatrix} \mathrm{vec}(I) & \mathrm{vec}(A) & \cdots & \mathrm{vec}(A^{n-1}) \end{bmatrix} \in \mathbb{R}^{n^2 \times n}. \tag{1.50}$$

The solution is given by

$$p = -(\mathcal{A}^T\mathcal{A})^{-1}\mathcal{A}^T \text{vec}(A^n) \tag{1.51}$$

An advantage of this method is that we do not have to chose the vector b. A disadvantage is that this last method is much more computing expense than the first method in which an arbitrarily vector b is chosen.

1.5 Similarity Transformations and eigenvectors

Assume given a non-singular matrix

$T \in \mathbb{R}^{n\times n}$ and a matrix $A \in \mathbb{R}^{n\times n}$. The matrix B defined by

$$B = T^{-1}AT \tag{1.52}$$

is then said to be similar to A. In particular, the eigenvalues of B is identical to the eigenvalues of A. The equation (1.52) is defined as a similarity transformation.

If we are putting the transformation matrix T equal to the eigenvector matrix, M, of the matrix A, then we have that

$$\Lambda = M^{-1}AM \tag{1.53}$$

where Λ is the eigenvalue matrix of the system (matrix A). The eigenvalue matrix Λ os a diagonal matrix with the eigenvalues on the diagonal. This can equivalently be written as

$$AM = M\Lambda \tag{1.54}$$

where

$$M = \left[\begin{array}{cccc} m_1 & m_2 & \cdots & m_n \end{array}\right] \tag{1.55}$$

The columns, m_1, $\cdots$, m_n in the eigenvector matrix M is the eigenvectors corresponding to the eigenvalues λ_1, $\cdots$, λ_n.

Λ is a diagonal matrix with the eigenvalues $\lambda_1, \cdots, \lambda_n$ on the diagonal. Hence, Equation (1.54) can then be written as n linear equations which can be used to compute the eigenvectors, i.e.,

$$\begin{aligned} Am_1 &= \lambda_1 m_1 \\ Am_2 &= \lambda_2 m_2 \\ &\vdots \\ Am_n &= \lambda_n m_n \end{aligned} \tag{1.56}$$

If the matrix A and the eigenvalues $\lambda_1, \cdots, \lambda_n$ are known, then, the eigenvectors and eigenvector matrix (1.55) can be found by solving the linear equations (1.56).

1.6 Time constant

Consider the case in which a 1st order differential equation

$$\dot{x} = ax, \tag{1.57}$$

with known initial value $x(t_0)$ is given. The time constant of the system is then defined as

$$T = -\frac{1}{a}. \tag{1.58}$$

The solution to Equation (1.57) can then be written as

$$x(t) = e^{a(t-t_0)}x(t_0) = e^{-\frac{1}{T}(t-t_0)}x(t_0). \tag{1.59}$$

We see that the solution $x(t)$ at time instant $t = t_0 + T$ is given by

$$x(t_0 + T) = e^{-\frac{1}{T}(t_0+T-t_0)}x(t_0) = e^{-1}x(t_0) \approx 0.37x(t_0). \tag{1.60}$$

I.e., the response have fallen 63% after $t = T$ Time Constant time units. See also Example 1.5 for illustration.

Example 1.2 (Time response of a 1st order system and the time constant T)

In connection with the Time Constant consider a transfer function model of a 1st order system $y(s) = h_p(s)u(s)$ where the transfer function is

$$h_p(s) = \frac{K}{Ts+1}. \tag{1.61}$$

The corresponding continuous state space model description is

$$\dot{x} = -\frac{1}{T}x + \frac{K}{T}u, \tag{1.62}$$

where K is the steady state system gain and T is the Time Constant.

The solution of the state eq. (1.62) is then given by the general solution in eq. (1.10), i.e.

$$x(t) = e^{-\frac{t-t_0}{T}}x(t_0) + \int_{t_0}^{t} e^{-\frac{t-\tau}{T}}\frac{K}{T}u(\tau)d\tau, \tag{1.63}$$

where t_0 is the initial time. For simplicity of illustration assume that a unit step $u = 1 \ \forall \ t_0 \leq t$ is feed into the system, initial time $t_0 = 0$, the system gain $K = 1$ and zero initial state $x(t_0) = 0$. The solution of eq. (1.63) is then

$$x(t) = \frac{1}{T}\int_{t_0}^{t} e^{-\frac{t-\tau}{T}}d\tau = \frac{1}{T}[Te^{-\frac{t-\tau}{T}}]_0^t = 1 - e^{-\frac{t}{T}} \tag{1.64}$$

We observe that the final state at time $t = \infty$ is $x(\infty) = 1$ and that the value of the state at time equal to the Time Constant, i.e. $t = T$ is

$$x(T) = 1 - e^{-1} \approx 0.63. \tag{1.65}$$

I.e. the value $x(T)$ of the state after $t = T$ time units is equal to (the common) 63% of the final value of the state $x(\infty) = 1$. see Figure 1.1 for illustration of the step response.

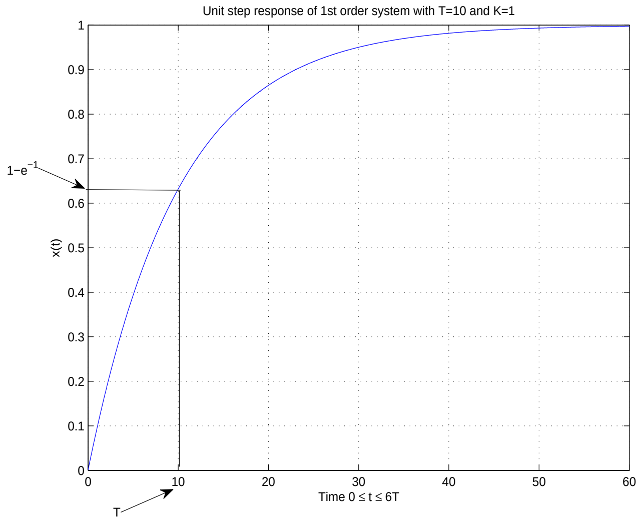

Figure 1.1: Time response of 1st order system $\dot{x} = ax + bu$ where $x(0) = 0$, $a = -\frac{1}{T}$, $b = \frac{K}{T}$, gain $K = 1$ and with Time Constant $T = 10$. Notice that after time equal to the time constant T then the response have reached $1 - e^{-1} \approx 0.63$ (63 %) of the final value.

Definition 1.1 (Time constants)
Given a linear dynamic system, $\dot{x} = Ax + Bu$, where $x \in \mathbb{R}^n$ is the state vector of the system. The system matrix A has n eigenvalues given by

$$\lambda_i = \lambda_i(A) \;\forall\; i = 1, \ldots, n. \tag{1.66}$$

If the eigenvalues are all real, distinct and have negative values (stable system), then the system will have th n time constants given by

$$T_i = -\frac{1}{\lambda_i} \;\; i = 1, \ldots, n. \tag{1.67}$$

Note also that the connection with the eigenvalues in a discrete time system

$$x_{k+1} = \phi x + \delta u, \tag{1.68}$$

and the continuous equivalent

$$\dot{x} = -\frac{1}{T}x + bu, \tag{1.69}$$

then is given by

$$\phi = e^{-\frac{1}{T}\Delta t} \tag{1.70}$$

which gives that

$$T = -\frac{\Delta t}{\ln \phi}. \tag{1.71}$$

Methods in system identification can be used to identify discrete time models from known input and output data of a system. Usually there are the parameters ϕ and δ which are estimated (computed). The relationship (1.71) is therefore very useful in order to find the time constant of the real time system.

1.7 The matrix exponent and the transition matrix

We have earlier in this section shown that that the transition matrix are involved in the exact solution of a linear time invariant dynamical system. Consider the autonomous system

$$\dot{x} = Ax \tag{1.72}$$

with known initial value $x_0 = x(t_0 = 0)$. Then the solution is given by

$$x(t) = \Phi(t)x(0) \tag{1.73}$$

where the transition matrix $\Phi(t)$ is given by

$$\Phi(t) = e^{At}. \tag{1.74}$$

As we see, the problem of computing the transition matrix $\Phi(t)$, is the same problem as computing the matrix exponent

$$F = e^{A}. \tag{1.75}$$

1.7.1 Computing the matrix exponent by diagonalisation

Let $f(A)$ be an analytical matrix function of A which also should contain the eigenvalue spectrum of A. A more general formulation of the similarity transformation given in (1.52) is then defined as

$$f(B) = T^{-1}f(A)T \tag{1.76}$$

Assume now that we want to compute the matrix exponent e^A. As we have shown in Equation (1.53 the matrix A can be decomposed as

$$A = M\Lambda M^{-1} \tag{1.77}$$

when the eigenvector matrix M is invertible. Using (1.77), (1.76) and $f(A) = e^A$ gives

$$e^A = Me^{\Lambda}M^{-1} \tag{1.78}$$

As we see, when the eigenvector matrix M and the eigenvalue matrix Λ of the matrix A are known, then the matrix exponential e^A can simply be computed from (1.78).

Equation (1.78) can be proved by starting with the autonomous system

$$\dot{x} = Ax \tag{1.79}$$

with known initial state vector $x(0)$. This system has the solution

$$x(t) = e^{At}x(0). \tag{1.80}$$

Transforming (1.79) by using $x = Mz$ gives

$$\dot{z} = \Lambda z \tag{1.81}$$

with initial state $z(0) = M^{-1}x(0)$. The canonical (transformed) system (1.81) have the solution

$$z(t) = e^{\Lambda t}z(0). \tag{1.82}$$

Using the transformation $x = Mz$ and putting this into (1.82) gives

$$x(t) = Me^{\Lambda t}z(0) = Me^{\Lambda t}M^{-1}x(0). \tag{1.83}$$

Comparing the two solutions (1.83) and (1.80) gives (1.78).

Note that in some circumstances there may be simpler to compute the transition matrix or matrix exponential $f(A)$ by solving the equation system

$$f(A)M = Mf(\Lambda) \tag{1.84}$$

because we in this case do not explicitly have to compute the matrix inverse M^{-1}. For some problems M is not invertible. This may be the case for systems which have multiple eigenvalues. We are referring to Parlet (1976) for a more detailed description of matrix functions and computing methods.

1.7.2 Parlets method for computing the matrix exponent

It can be shown, Parlet (1976), that the matrix exponent $F = e^A$ and the system matrix A commutes, i.e. the following is satisfied

$$FA = AF \tag{1.85}$$

If the matrix A has a special structure, e.g., upper or lower triangular, then Equation (1.7.2) can with advantage be used in order to compute the unknown elements in the transition matrix.

Note that the matrix exponential of an upper triangular matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix}, \tag{1.86}$$

is given by

$$F = e^A = \begin{bmatrix} e^{a_{11}} & f_{12} \\ 0 & e^{a_{22}} \end{bmatrix}. \tag{1.87}$$

The unknown coefficient f_{12} can then simply be found from equation ().

Example 1.3 (computing the transition matrix)
Given an autonomous system described by

$$\dot{x} = Ax, \tag{1.88}$$

with the initial state $x_0 = x(0)$. The system matrix A is given by

$$A = \begin{bmatrix} \lambda_1 & \alpha \\ 0 & \lambda_2 \end{bmatrix}. \tag{1.89}$$

We want to compute the transition matrix

$$F = e^{At} \tag{1.90}$$

by Parlets method. First we find immediately that

$$F = \begin{bmatrix} e^{\lambda_1 t} & f \\ 0 & e^{\lambda_2 t} \end{bmatrix}. \tag{1.91}$$

We now have to find the unknown constant f_{12} in the transition matrix. This can be done from the equation system

$$AF = FA. \tag{1.92}$$

This gives four equations but only one of them gives information of interest,

$$\lambda_1 f_{12} + \alpha e^{\lambda_2 t} = e^{\lambda_1 t}\alpha + f_{12}\lambda_2. \tag{1.93}$$

Solving with respect to f_{12} gives

$$f_{12} = \alpha \frac{e^{\lambda_1 t} - e^{\lambda_2 t}}{\lambda_1 - \lambda_2} \tag{1.94}$$

As we see, this method can simply be used for system matrices which have a triangular structure, and in which the eigenvalues are distinct and not identical to zero.

1.7.3 Matrix exponential by series expansion

It can be shown that the matrix exponential F^A can be expressed as an infinite Taylor series

$$e^A = I + A + \frac{1}{2}A^2 + \cdots \tag{1.95}$$

The transition matrix can be expressed in the same way, e.g.,

$$e^{At} = I + At + \frac{1}{2}A^2 t^2 + \cdots \tag{1.96}$$

This is in general not a good method for computing the transition matrix, because it will in general lead to numerical problems when computing powers of A like A^9, A^{10}, etc. especially when A contains small values. This is so due to the finite precision of the computer. Note that the machine precision of a 32 bit computer is $eps = 1/2^{52}$.

The series method is however very useful for computing the transition matrix of many simple systems. This will be illustrated in the following example.

Example 1.4 (computing transition matrix)
Given an autonomous system described by

$$\dot{x} = Ax, \tag{1.97}$$

where the initial state $x_0 = x(0)$ is given and the system matrix is given by

$$A = \begin{bmatrix} 0 & \alpha \\ 0 & 0 \end{bmatrix}. \tag{1.98}$$

The transition matrix for this system is simply found from the two first terms of the Taylor series (1.96) because A is so called nil-potent, i.e., we have that $A^2 = 0$, $A^3 = 0$ and so on. We have

$$\Phi(t) = I + At = \begin{bmatrix} 1 & \alpha t \\ 0 & 1 \end{bmatrix}. \tag{1.99}$$

1.8 Examples

Example 1.5 (autonomous response and time constant)
Given an autonomous system

$$\dot{x} = ax \tag{1.100}$$

where the initial state is $x_0 = x(t_0 = 0) = 1$ and the system parameter $a = -\frac{1}{T}$ where the time constant is $T = 5$. The solution of this differential equation is

$$x(t) = e^{-\frac{1}{T}t}x_0 = e^{-\frac{1}{5}t}. \tag{1.101}$$

Let us now plot the solution in the time interval $0 \leq t \leq 25$. Note that the state will have approximately reached the steady state value after $4T$ (four times the time constant). The solution is illustrated in Figure 1.2.

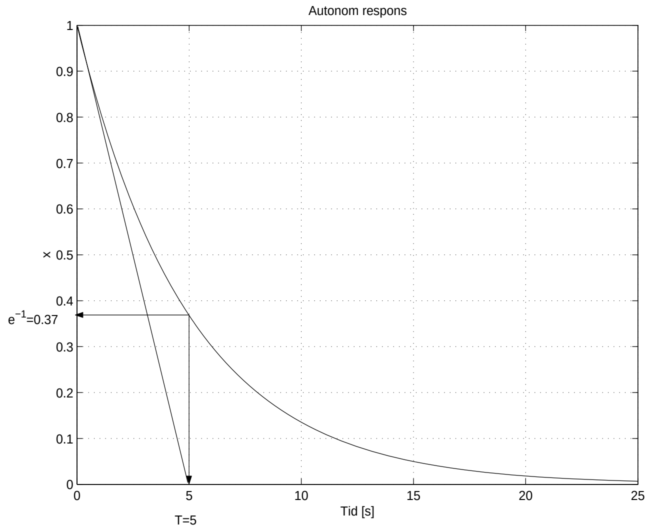

Figure 1.2: Time response of autonomous system $\dot{x} = ax$ where $x_0 = 0$ and $a = -\frac{1}{T}$ and with Time Constant $T = 5$.

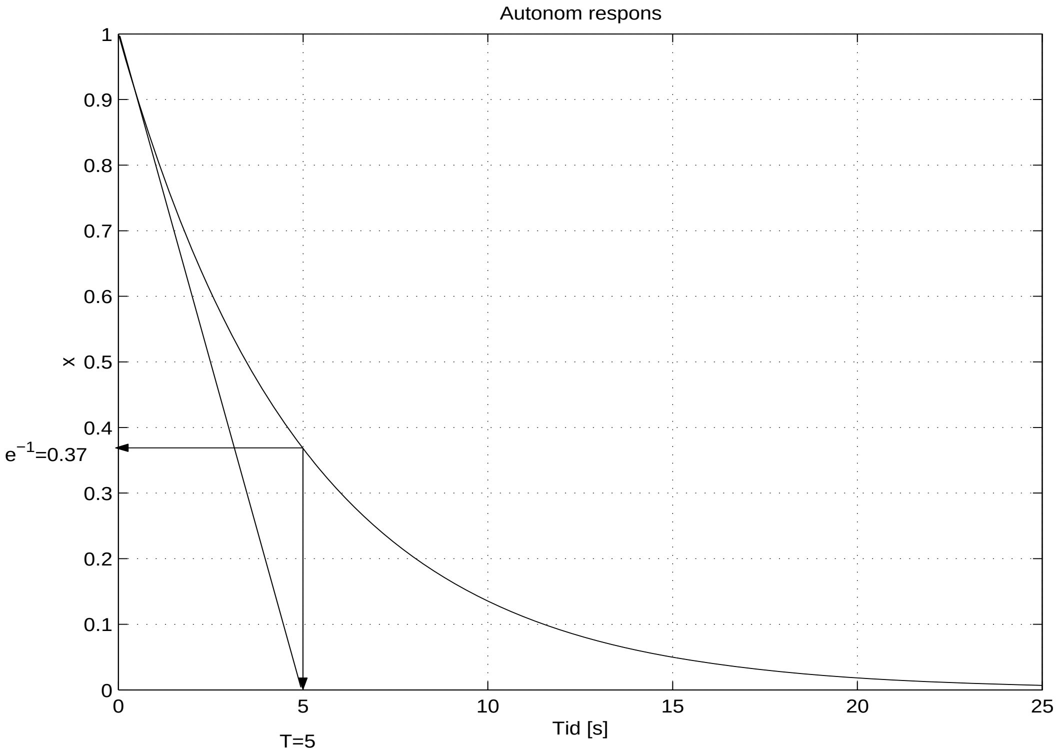

Figure 1.3: Time response of autonomous system $\dot{x} = ax$ where $x_0 = 0$ and $a = -\frac{1}{T}$ and with Time Constant $T = 5$.

Example 1.6 (computation of matrix exponent)
Given the system matrix

$$A = \begin{bmatrix} -3 & 1 \\ 2 & -2 \end{bmatrix}. \tag{1.102}$$

The eigenvalue matrix Λ *and the corresponding eigenvector matrix can be shown to be as follows*

$$M = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} -1 & 0 \\ 0 & -4 \end{bmatrix} \tag{1.103}$$

Find the matrix exponent $F = e^A$.

We have the relation $F = Me^{\Lambda}M^{-1}$, *which is equivalent with*

$$FM = Me^{\Lambda} \tag{1.104}$$

which gives

$$\begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} e^{-1} & 0 \\ 0 & e^{-4} \end{bmatrix} \tag{1.105}$$

From this we have the four equations

$$\begin{bmatrix} f_{11} + 2f_{12} & f_{11} - f_{12} \\ f_{21} + 2f_{22} & f_{21} - f_{22} \end{bmatrix} = \begin{bmatrix} e^{-1} & e^{-4} \\ 2e^{-1} & -e^{-4} \end{bmatrix} \tag{1.106}$$

Taking element $1,1$ *minus element* $1,2$ *on the left hand side of Equation (1.106) gives*

$$3f_{12} = e^{-1} - e^{-4} \tag{1.107}$$

Putting the expression for f_{12} *into element* $1,2$ *on the left hand side gives* f_{11}*, i.e.,*

$$f_{11} - f_{12} = e^{-4} \tag{1.108}$$

which gives

$$f_{11} = \frac{1}{3}(e^{-1} + 2e^{-4}) \tag{1.109}$$

Taking element $2,1$ *minus element* $2,2$ *on the left hand side of (1.106) gives*

$$3f_{22} = 2e^{-1} + e^{-4} \tag{1.110}$$

Putting the expression for f_{22} *into e.g., element* $2,2$ *gives* f_{21}*. This gives the final result*

$$F = e^{A} = \frac{1}{3}\begin{bmatrix} e^{-1} + 2e^{-4} & e^{-1} - e^{-4} \\ 2e^{-1} - 2e^{-4} & 2e^{-1} + e^{-4} \end{bmatrix} \tag{1.111}$$

Note that the transition matrix could have been computed similarly, i.e.,

$$\Phi(t) = e^{At} = Me^{\Lambda t}M^{-1} = \frac{1}{3}\begin{bmatrix} e^{-t} + 2e^{-4t} & e^{-t} - e^{-4t} \\ 2e^{-t} - 2e^{-4t} & 2e^{-t} + e^{-4t} \end{bmatrix} \tag{1.112}$$

Example 1.7 (computation of transition matrix for upper triangular system)

Consider given an autonomous system described by the matrix differential equation

$$\dot{x} = Ax, \tag{1.113}$$

where the initial state $x_0 = x(0)$ *is given and where the system matrix is given by*

$$A = \begin{bmatrix} \lambda_1 & \alpha \\ 0 & \lambda_2 \end{bmatrix}. \tag{1.114}$$

The transition matrix $\Phi(t) = e^{At}$ *will have the same upper triangular structure as* A *and the diagonal elements in* $\Phi(t)$ *is simply* $e^{-\lambda_1 t}$ *and* $e^{-\lambda_2}$*, i.e.,*

$$\Phi(t) = \begin{bmatrix} e^{\lambda_1 t} & f_{12} \\ 0 & e^{\lambda_2 t} \end{bmatrix}. \tag{1.115}$$

The unknown element f_{12} *can now simply be computed from Parlets method, i.e., we solve the equation*

$$\Phi A t = A t \Phi \tag{1.116}$$

or equivalent

$$\Phi A = A\Phi. \tag{1.117}$$

This gives the equation

$$e^{\lambda_1 t}\alpha + f_{12}\lambda_2 = \lambda_1 f_{12} + \alpha e^{\lambda_2 t}. \tag{1.118}$$

Solving for the remaining element f_{12} *gives*

$$f_{12} = \alpha \frac{e^{\lambda_1 t} - e^{\lambda_2 t}}{\lambda_1 - \lambda_2}. \tag{1.119}$$

Note that this method only can be used when the system has distinct eigenvalues, i.e. when $\lambda_1 \neq \lambda_2$*.*

However, it can be shown that in the limit when $\lambda_2 \to \lambda_1 = \lambda$ *that*

$$f_{12} = \alpha t e^{\lambda_1 t} \tag{1.120}$$

Example 1.8 (Set of higher order ODE to set of first order ODE)
Consider a system described be the following couple of coupled differential equations

$$\begin{aligned} \ddot{y}_1 + k_1\dot{y}_1 + k_2 y_1 &= u_1 + k_3 u_2 \\ \dot{y}_2 + k_4 y_2 + k_3\dot{y}_1 &= k_6 u_1 \end{aligned}$$

where u_1 *and* u_2 *is defined as the control inputs and* y_1 *and* y_2 *is defined as the measurements or outputs*

We now define the outputs and if necessary the derivatives of the outputs as states. Hence, define the states

$$x_1 = y_1, \quad x_2 = \dot{y}_1, \quad x_3 = y_2 \tag{1.121}$$

This gives the following set of 1st order differential equations for the states

$$\dot{x}_1 = x_2 \tag{1.122}$$

$$\dot{x}_2 = -k_2 x_1 - k_1 x_2 + u_1 + k_3 u_2 \tag{1.123}$$

$$\dot{x}_3 = -k_5 x_2 - k_4 x_3 + k_6 u_1 \tag{1.124}$$

and the following measurements (outputs) variables

$$y_1 = x_1 \tag{1.125}$$

$$y_2 = x_3 \tag{1.126}$$

The model is put on matrix (State Space) form as follows

$$\begin{aligned} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ -k_2 & -k_1 & 0 \\ 0 & -k_5 & k_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & k_3 \\ k_6 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \end{aligned}$$

and finally in matrix form as follows

$$\dot{x} = Ax + Bu \tag{1.127}$$

$$y = Dx \tag{1.128}$$

1.9 Transfer function and transfer matrix models

Laplace transforming $\dot{x}$ gives

$$L(\dot{x}(t)) = sx(s) - x(t=0). \tag{1.129}$$

Similarly, the Laplace transform of a time dependent variable x is defined as

$$L(x(t)) = x(s). \tag{1.130}$$

Using (1.129) and the definition (1.130) in the state space model

$$\begin{aligned} \dot{x} &= Ax + Bu, \qquad x(t=0) = 0, & (1.131)\\ y &= Dx + Eu, & (1.132) \end{aligned}$$

gives the Laplace transformed model equivalent

$$\begin{aligned} x(s) &= (sI - A)^{-1}Bu(s), & (1.133)\\ y(s) &= Dx(s) + Eu(s). & (1.134) \end{aligned}$$

We can now write (1.133) and (1.134) as a transfer matrix model

$$y(s) = H(s)u(s), \tag{1.135}$$

where $H(s)$ is the transfer matrix of the system

$$H(s) = D(sI - A)^{-1}B + E. \tag{1.136}$$

For single-input and single-output systems then $H(s)$ will be a scalar function of the Laplace variable s. In this case we usually are using a small letter, i.e., we are putting $h(s) = H(s)$. Note also that we have included a direct influence from the input u to the output y in the measurement (output) equation. This will be the case in some circumstances. However, the matrix or parameter E is usually zero in control systems, in particular $E = 0$ in standard feedback systems.

Note also that when the eigenvalue decomposition $A = M\Lambda M^{-1}$ exists then we have that the transfer matrix can be expressed and computed as follows

$$\begin{aligned} H(s) &= D(sI - A)^{-1}B + E & (1.137)\\ &= DM(sI - \Lambda)^{-1}M^{-1}B + E. & (1.138) \end{aligned}$$

Finally note the following important relationship. The properties of a time dependent state space model when $t \to \infty$, i.e. the steady state properties, can be analyzed in a transfer function Laplacian model by putting $s = 0$. The transient behavior when $t = 0$ is analyzed by letting $s \to \infty$.

For Single Input and Single Output (SISO) systems we often write the plant model as

$$h_p(s) = D(sI - A)^{-1}B + E = \frac{\rho(s)}{\pi(s)}, \tag{1.139}$$

where $\rho(s)$ is the zero polynomial and $\pi(s)$ is the pole polynomial

1.10 Poles and zeroes

For a Single Input Single Output (SISO) system we may write the transfer function model from the control input, u, to the output measurement, y,as follows

$$y(s) = h(s)u(s), \tag{1.140}$$

where the transfer function may be written as

$$h(s) = \frac{\rho(s)}{\pi(s)}, \tag{1.141}$$

where $\rho(s)$ is the zero polynomial and $\pi(s)$ is the pole polynomial.

The poles of the system is the roots of the denominator pole polynomial, i.e.,

$$\pi(s) = 0. \tag{1.142}$$

The zeroes of the system is given by the roots of the numerator zero polynomial, i.e.

$$\rho(s) = 0. \tag{1.143}$$

Remark that the n poles is the n roots of the pole polynomial $\pi(s) = 0$ and that these n poles is identical to the n eigenvalues of the A matrix and that the pole polynomial also may be deduced from the system matrix A of an equivalent observable and controllable state space model, i.e.,

$$\pi(s) = \det(sI - A) = 0. \tag{1.144}$$

For the poles to be equal to the eigenvalues of the A matrix the linear state space model $\dot{x} = Ax + Bu$ and $y = Dx$ have to be a minimal realization, i.e., the model is both controllable and observable.

1.11 Time Delay

A time delay (or dead time) in a system may in continuous time domain be described as follows. Suppose a variable $y(t)$ is equal to a variable $x(t)$ delayed $\tau \geq 0$ time units. Then we may write

$$y(t) = x(t - \tau) \tag{1.145}$$

where $y(t) = 0$ when $0 \leq t < \tau$ and $y(t) = x(t - \tau)$ when time $t \geq \tau$.

The Laplace plane model equivalent of the time delay is

$$y(s) = e^{-\tau s}x(s) \tag{1.146}$$

Notice that the exponential $e^{-\tau s}$ is an irrational function and that we can not do algebra with such functions, and that rational approximations to the exact delay have to be used if a delay should be used in algebraic calculations.

Numerous approximations exist and may be derived from the series approximation of the exponential.

1.12 Linearization

In many cases the starting point of a control problem or model analysis problem is a non-linear model of the form

$$\dot{x} = f(x,u), \tag{1.147}$$

$$y = g(x,u). \tag{1.148}$$

Here $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^r$ is the control input vector and $y \in \mathbb{R}^m$ is the output or measurements vector. The functions $f(\cdot,\cdot) \in \mathbb{R}^n$ and $g(\cdot,\cdot) \in \mathbb{R}^m$ may be non-linear smooth functions of x and u. Note also that the initial state is $x(t_0)$ which should be given ore known before the state space model can be simulated in time.

In this case it may be of interest to derive a linear model approximation to (1.147) and (1.148).

The two first (linear terms) of a Taylor series expansion of the right hand side of (1.147) around the points x_0 and u_0 gives

$$f(x,u) \approx f(x_0,u_0) + \left.\frac{\partial f}{\partial x^T}\right|_{x_0,u_0}(x-x_0) + \left.\frac{\partial f}{\partial u^T}\right|_{x_0,u_0}(u-u_0). \tag{1.149}$$

Define the deviation variables

$$\Delta x = x - x_0, \tag{1.150}$$

$$\Delta u = u - u_0. \tag{1.151}$$

Also define the matrices

$$A = \left.\frac{\partial f}{\partial x^T}\right|_{x_0,u_0} \in \mathbb{R}^{n\times n} \tag{1.152}$$

which also is named the Jacobian matrix. Similarly, define

$$B = \left.\frac{\partial f}{\partial u^T}\right|_{x_0,u_0} \in \mathbb{R}^{n\times r} \tag{1.153}$$

Putting (1.149), (1.150) and (1.151) into the state equation (1.147) gives the linearized state equation model

$$\dot{\Delta x} = A\Delta x + B\Delta u + v, \tag{1.154}$$

where

$$v = f(x_0,u_0) - \dot{x}_0. \tag{1.155}$$

Usually the points x_0 and u_0 is constant steady state values such that

$$\dot{x_0} = f(x_0,u_0) = 0. \tag{1.156}$$

Hence, a linearized state equation is given by

$$\dot{\Delta x} = A\Delta x + B\Delta u. \tag{1.157}$$

Similarly the output equation (1.148) can be linearized by approximating the right hand side by the first two terms of a Taylor series expansion, i.e.,

$$y \approx g(x_0, u_0) + \frac{\partial g}{\partial x^T}\bigg|_{x_0,u_0} (x - x_0) + \frac{\partial g}{\partial u^T}\bigg|_{x_0,u_0} (u - u_0). \tag{1.158}$$

Now defining

$$\begin{aligned} y_0 &= g(x_0, u_0) && (1.159)\\ \Delta y &= y - y_0 && (1.160)\\ D &= \frac{\partial g}{\partial x^T}\bigg|_{x_0,u_0} && (1.161)\\ E &= \frac{\partial g}{\partial u^T}\bigg|_{x_0,u_0} && (1.162) \end{aligned}$$

gives the linearized output equation

$$\Delta y = D\Delta x + E\Delta u. \tag{1.163}$$

Usually the deviation variables are defined as

$$\begin{aligned} x &:= x - x_0, && (1.164)\\ u &:= u - u_0. && (1.165) \end{aligned}$$

Hence, a linear or linearized state space model, given by (1.157) and (1.163), is usually written as follows.

$$\begin{aligned} \dot{x} &= Ax + Bu, && (1.166)\\ y &= Dx + Eu. && (1.167) \end{aligned}$$

One should therefore note that the variables in a linearized model may be deviation variables, but this is not always the case. One should also note that only linear models can be transformed to Laplace plane models. Note also that the initial state in the linearized model is given by $\Delta x(t_0) = x(t_0) - x_0$.

Example 1.9 (Linearization of a pendulum model)
An non-linear model for a pendulum can be written as the following second order differential equation, i.e.,

$$\ddot{\theta} + \frac{b}{mr^2}\dot{\theta} + \frac{g}{r}\sin(\theta) = 0, \tag{1.168}$$

where θ is the angular position (deviation of the pendulum from the vertical line, i.e. from the steady state position). $m = 8$ is the mass of the pendulum, $r = 5$ is the length of the pendulum arm, $b = 10$ is a friction coefficient in the base point and $g = 9.81 m/s^2$ is the acceleration of gravity constant.

The second order model can be written as a set of 1st order differential equations by defining the states

$$\begin{aligned} x_1 &= \theta \quad \textit{angular position} && (1.169)\\ x_2 &= \dot{\theta} \quad \textit{angular velocity} && (1.170) \end{aligned}$$

from this definitions we have that $\dot{x}_1 = \dot{\theta} = x_2$ *which gives the state space model*

$$\begin{aligned} \dot{x}_1 &= x_2, && (1.171) \\ \dot{x}_2 &= -\frac{g}{r}\sin(x_1) - \frac{b}{mr^2}x_2, && (1.172) \end{aligned}$$

which is equivalent to a non-linear model

$$\dot{x} = f(x) \tag{1.173}$$

with the vector function

$$f(x) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{g}{r}\sin(x_1) - \frac{b}{mr^2}x_2 \end{bmatrix} \tag{1.174}$$

Linearizing around the steady state solution $x_1 = 0$ *and* $x_2 = 0$ *gives*

$$\dot{x} = Ax, \tag{1.175}$$

where the Jacobian is given by

$$A = \frac{\partial f}{\partial x^T}\Big|_0 = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_0 = \begin{bmatrix} 0 & 1 \\ -\frac{g}{r}\cos(x_1) & -\frac{b}{mr^2} \end{bmatrix}_0 = \begin{bmatrix} 0 & 1 \\ -\frac{g}{r} & -\frac{b}{mr^2} \end{bmatrix} \tag{1.176}$$

Putting into the numerical values we obtain

$$A = \begin{bmatrix} 0 & 1 \\ -1.962 & -0.050 \end{bmatrix} \tag{1.177}$$

Note that the linearized model could have been obtained more directly by using that $\sin(x_1) \approx x_1$ *for small angles* x_1.

Example 1.10 (Simulation of a non-linear pendulum model)
The nonlinear state space pendulum model

$$\begin{aligned} \dot{x}_1 &= x_2, && (1.178) \\ \dot{x}_2 &= -\frac{g}{r}\sin(x_1) - \frac{b}{mr^2}x_2, && (1.179) \end{aligned}$$

with $g = 9.81$, $r = 5$, $m = 8$ *and* $b = 10$ *can be simply simulated in MATLAB by using an ODE solver, e.g.,*

```
>> sol=ode15s(@ fx_pendel, 0:0.1:50,[1;0]);
>> plot(sol.x,sol.y)
```

Here sol is an object where sol.x is the time axis and sol.y is the states. The ode15s function simulate the pendulum model over the time horizon $t = t_0 : h : t_f$, *i.e. from the initial time* $t_0 = 0$ *and to the final time* $t_f = 50$ *with step length (sampling interval)* $\Delta t = 0.1$. *Try it! The file fx_pendel is an m-file function given in the following*

```
function fx=fx_pendel(t,x)
% fx_pendel
% fx=fx_pendel(t,x)
% Modell av pendel.

m=8; g=9.81; b=10; r=5;

fx=zeros(2,1);
fx(1)=x(2);
fx(2)=-b*x(2)/(m*r^2)-g*sin(x(1))/r;
```

1.12.1 Calculating the Jacobian matrix numerically

In some circumstances the Jacobian matrix $A = \frac{df(x)}{dx^T}$, and matrices $\frac{dg(x)}{dx^T}$, may be hard and difficult to calculate analytically. In these cases it may be of grate interest to calculate the derivatives numerically.

Notice that we will in this section assume that the vector $f(x) \in \mathbb{R}^m$ and vector $x \in \mathbb{R}^n$. The Jacobian matrix is in this case defined as

$$A = \frac{df(x)}{dx^T} = \begin{bmatrix} \frac{df(x)}{dx_1} & \cdots & \frac{df(x)}{dx_i} & \cdots & \frac{df(x)}{dx_n} \end{bmatrix} = \begin{bmatrix} \frac{df_1(x)}{dx_1} & \cdots & \frac{df_1(x)}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_m(x)}{dx_1} & \cdots & \frac{df_m(x)}{dx_n} \end{bmatrix} \tag{1.180}$$

This may for instance be done by using an approximation to the derivative, e.g. as the simple approximation (or similar)

$$\frac{df(x_i)}{dx_i} \approx \frac{f(x_i + h) - f(x_i)}{h} \;\forall\; i = 1, 2, ..., n \tag{1.181}$$

for some small number h. Hence, for each elements in vector $f(x) \in \mathbb{R}^m$, say for generality each of the m elements in in vector $f(x)$, we loop through all n elements in x using this approximation. This means that we for each element $1 \leq j \leq m$ in $f(x)$ we calculate

$$a_{ji} = \frac{df_j(x)}{dx_i} \approx \frac{f_j(x_i + h) - f_j(x_i)}{h} \;\forall\; i = 1, 2, ..., n \tag{1.182}$$

However, this may in MATLAB be done effectively by calculating one column of the Jacobian matrix A at at time, hence

$$A(:, i) \approx \frac{f(x_i + h) - f(x_i)}{h} \;\forall\; i = 1, 2, ..., n \tag{1.183}$$

where $A(:, i)$ is MATLAB notation for the entire $i - th$ column of the A matrix. The cost of this procedure for numerically calculating the Jacobian matrix are $n+1$ function evaluations. This numerical procedure to calculate the Jacobian matrix is implemented in the following MATLAB **jacobi.m** function.

```
function A=jacobi(fx_fil,t,x)
%JACOBI Function to calculate the Jacobian matrix numerically
%       A=jacobi('fx_file',t,x)
%       PURPOSE
%       Function to calculate the Jacobian matrix A=df/dx of the nonlinear
%       function fx=f(t,x), i.e. the linearization of a possible non-linear
%       function fx=f(t,x) around values t and x.
%       ON INPUT
%       fx_file - m-file to define the function fx=f(t,x) with
%                 syntax: function fx=fx_file(t,x) where fx is,
%                 say  m dimensional
%       t       - time instant.
%       x       - column vector with same dimension as in the function
%           fx=f(t,x), say of dimension n
%       ON OUTPUT
%       A       - The jacobian matrix A=df/dx of dimension (m,n)
h=1e-5;
n=length(x);
xh=x;
fx=feval(fx_fil,t,x);
m=length(fx);
A=zeros(m,n);
for i=1:n
   xh(i)=x(i)+h;
   fxh=feval(fx_fil,t,xh);
   xh(i)=xh(i)-h;
   A(:,i)=(fxh-fx)/h;
end
% END JACOBI
```

In order to illustrate how MATLAB may be used to numerically calculating the Jacobian matrix we use the pendulum example in Example 1.10 which gives the same result as the analytic expression in Example 1.9 and Equation 1.177.

```
>> A=jacobi('fx_pendel',0,[0;0])

A =

         0    1.0000
   -1.9620   -0.0500

>>
```

1.13 Stability of linear systems

1.13.1 Stability of continuous time linear systems

Given a linear continuous time system

$$\begin{aligned} \dot{x} &= Ax + Bu, && (1.184) \\ y &= Dx. && (1.185) \end{aligned}$$

- If the eigenvalues of A is complex, then they are existing in complex conjugate pairs, i.e. $\lambda_i = \alpha \pm \beta j$ where $j = \sqrt{-1}$ is the imaginary number. Hence, one pair of complex conjugate eigenvalues results in the eigenvalues $\lambda_i = \alpha + \beta j$ and $\lambda_{i+1} = \alpha - \beta j$. This is a property of the eigenvalues of a real matrix A.
- The system is stable if the real part of the eigenvalues, $\lambda_i(A)\ \forall\ i = 1, 2, \ldots, n$, are negative. The eigenvalues of a stable system are located in the left part of the complex plane.
- If one ore some of the eigenvalues are located on the imaginary axis we say that the system is marginally stable.
- If one (or more) of the eigenvalues are zero we say that we have an integrator (or have several integrators) in the system.
- If one ore more of the eigenvalues are located in the right half plane, i.e. with positive real parts, then the system is unstable.

The transfer function/matrix model equivalent to the state space model is $y = h(s)u$ where the transfer function/matrix is given by

$$h(s) = D(sI - A)^{-1}B. \tag{1.186}$$

The pole polynomial of a transfer function model is given by $\det(sI - A)$ and its roots, the solution of the pole polynomial $\det(sI - A) = 0$ is the system poles.

The poles of a transfer function model and the eigenvalues of the A matrix coincide for controllable and observable systems, i.e. the poles are identical to the observable and controllable eigenvalues in the state space model equivalent. This also implies that possible unobservable and/or uncontrollable states/eigenvalues in a state space model can not be computed from the transfer function model.

1.13.2 Stability of discrete time linear systems

Given a linear discrete time system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, && (1.187) \\ y_k &= Dx_k. && (1.188) \end{aligned}$$

- The discrete time linear system described by Eq. (1.187) is stable if the eigenvalues of the system matrix A is located inside the unit circle in the complex plane. This is equivalent to check wether the eigenvalues has magnitude less than one, i.e., $|\lambda_i| < 1\ \forall\ i = 1, \ldots, n$.

- If one (or more) of the eigenvalues have magnitude equal to one, we have an integrator (or integrators) in the system (if the eigenvalue are real).

Example 1.11 *Given a linear continuous time system $\dot{x} = Ax + Bu$ and $y = Dx$ with model matrices*

$$A = \begin{bmatrix} -\alpha & -\beta \\ \beta & -\alpha \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{1.189}$$

The eigenvalues of the system matrices are simply given by a complex conjugate pair of eigenvalues, i.e., $\lambda_{1,2} = -\alpha \pm \beta i$. This system is then stable for all $\alpha > 0$. The eigenvalues may simply be found as the elements on the diagonal of the A matrix for this example, but in general as the roots of the characteristic equation $\det(\lambda I - A) = \lambda^2 + 2\alpha\lambda + \alpha^2 + \beta^2 = 0$.

The transfer function model is $y = h(s)u$ with

$$h(s) = D(sI - A)^{-1}B = \frac{s + \alpha}{s^2 + 2\alpha s + \alpha^2 + \beta^2} \tag{1.190}$$

1.14 State Controllability

The question of how we can (and if there is possible to) find a suitable control input $u(t)$ that will take the system from an initial state $x(t_0)$ to any desired final state $x(t_1)$ in a finite (often very small) time, is answered by the theory of state controllability.

Definition 1.2 (State Controllability)
A system described by a state space model $\dot{x} = Ax + Bu$ with initial state $x(t_0)$ given is controllable if there, for an arbitrarily finite time $t_1 > t_0$ exist a control function $u(t)$ defined over the time interval $t_0 \leq t \leq t_1$, such that the final state, $x(t_1)$, can be arbitrarily specified.

There exist a few algebraic definitions which can be used for the analysis of state controllability. Such a theorem is defined via the so called controllability matrix.

Teorem 1.14.1 (Controllability matrix)
A system described by a state space model $\dot{x} = Ax + Bu$ is state controllable if the controllability matrix

$$C_n = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{1.191}$$

has full rank, i.e.,

$$rank(C_n) = n. \tag{1.192}$$

Note that for single input systems, i.e., $r = 1$, then $C_n \in \mathbb{R}^{n \times n}$ which implies that C_n should be invertible and that $\det(C_n) \neq 0$ *in order for the system to be state controllable.*

Remark 1.1 (Diagonal form and controllability)
Consider a state space model $\dot{x} = Ax + Bu$ and $y = Dx + Eu$ and its diagonal canonical form

$$\begin{aligned} \dot{z} &= \Lambda z + M^{-1}Bu & (1.193)\\ y &= DMz + Eu & (1.194) \end{aligned}$$

where Λ is a diagonal matrix with the eigenvalues $\lambda_i \; \forall \; i = 1, \dots n$ of A on the diagonal and $M = \begin{bmatrix} m_1 & \cdots & m_n \end{bmatrix}$ is the corresponding eigenvector matrix. Note the relationship $Am_i = \lambda_i m_i$ between the ith eigenvalue, λ_i, and the ith eigenvector, m_i.

The system is controllable if no rows in the matrix $M^{-1}B$ is identically equal to zero.

One should also note that there also is a dual phenomena, observability. The system is observable if no columns in the matrix DM is identically equal to zero.

1.15 State Observability

Definition 1.3 (State Observability)
A system described by a state space model $\dot{x} = Ax + Bu$ and $y = Dx$ with initial state $x(t_0)$ is observable if there, from knowledge of known inputs, $u(t)$, and outputs, $y(t)$, over a time interval $t_0 \leq t \leq t_1$, is possible to compute the (initial) state vector, $x(t_0)$.

Teorem 1.15.1 (Observability matrix)
Define the *observability matrix*

$$O_i = \begin{bmatrix} D \\ DA \\ DA^2 \\ \vdots \\ DA^{i-1} \end{bmatrix} \in \mathbb{R}^{mi \times n}, \qquad (1.195)$$

The pair (D, A) is observable if and only if the *observability matrix* O_i for $i = n$ has rank n, i.e. $\text{rank}(O_n) = n$.

If $\text{rank}(D) = r_D \geq 1$ and $n - r_D + 1 > 0$, then we have that the pair (D, A) is observable if and only if the reduced observability matrix O_{n-r_D+1} have rank n. For single output systems we use $i = n$ and $O_n \in n \times n$.
△

Example 1.12 (Observability of continuous autonomous system) *Consider a single output autonomous system*

$$\begin{aligned} \dot{x} &= Ax, & (1.196)\\ y &= Dx & (1.197) \end{aligned}$$

from this we have that

$$\begin{aligned} y &= Dx && (1.198)\\ \dot{y} &= DAx && (1.199)\\ \ddot{y} &= DA^2x && (1.200)\\ &\vdots && (1.201)\\ \overset{(n-1)}{y} &= DA^{n-1}x && (1.202) \end{aligned}$$

where $\overset{(n-1)}{y}$ *denotes the* $n-1$*th derivative of* y*, i.e.,* $\overset{(n-1)}{y} = \frac{d^{n-1}y}{dt^{n-1}}$*. From these* n *equations we define the following matrix equation*

$$y_{0|n} = O_n x \tag{1.203}$$

where

$$y_{0|n} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ \vdots \\ \overset{(n-1)}{y} \end{bmatrix} \tag{1.204}$$

and where $O_n \in \mathbb{R}^{n\times n}$ *observability matrix as defined in the above definition, Equation (1.195). If the observability matrix,* O_n*, is non-singular then we can compute the state vector* $x(t)$ *as*

$$x = O_n^{-1} y_{0|n} \tag{1.205}$$

An $n \times n$ *matrix* O_n *is non-singular if rank*$(O_n) = n$*.*

Example 1.13 (Observability of discrete autonomous system) *Consider a single output autonomous system*

$$\begin{aligned} x_{k+1} &= Ax_k, && (1.206)\\ y_k &= Dx_k && (1.207) \end{aligned}$$

from this we have that

$$\begin{aligned} y_k &= Dx_k && (1.208)\\ y_{k+1} &= DAx_k && (1.209)\\ y_{k+2} &= DA^2x_k && (1.210)\\ &\vdots && (1.211)\\ y_{k+n-1} &= DA^{n-1}x && (1.212) \end{aligned}$$

From these n *equations we define the following matrix equation*

$$y_{k|n} = O_n x_k \tag{1.213}$$

where

$$y_{k|n} = \begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n-1} \end{bmatrix} \tag{1.214}$$

and where $O_n \in \mathbb{R}^{n\times n}$ *observability matrix as defined in the above definition, Equation (1.195). If the observability matrix,* O_n*, is non-singular then we can compute the state vector* x_k *as*

$$x_k = O_n^{-1} y_{k|n} \tag{1.215}$$

An $n \times n$ *matrix* O_n *is non-singular if rank*$(O_n) = n$.

Chapter 2

Canonical forms

2.1 Introduction

A state space model realizations (A, B, D) can be represented in an infinite number of realizations (coordinate systems). Let x be the state vector for the realization (A, B, D). Then, the transformed state $z = T^{-1}x$ will be the state of the realization $(T^{-1}AT,\ T^{-1}B,\ DT)$. Hence, there exists an infinite number of such non-singular transformation matrices T.

The number of parameters in the model (A, B, D) are $n^2 + nr + mn$. However, the number of free independent parameters are much less. It can be shown that the minimum number of free and independent parameters are $nm + nr$. This means that there may exist a transformation matrix, T, that transforms a realization (A, B, D) to a canonical form with special structure and with a minimum number of free and independent parameters. In such canonical form models there also is a number of fixed parameters which is equal to zero and one, in addition to the $nm + nr$ independent system parameters. Hence, canonical forms are realizations of a given realization (A, B, D) with a special structure and with a minimal number of free parameters and as many ones and zeros as possibile.

Canonical form state space realizations is important in, omong others, the following problems:

- **Analysis of the characteristic properties of the system and the model.** Computations of eigenvalues and thereby analysis of stability and time constants. The characteristic equation which often is the starting point for computations of eigenvalues are easily obtained from the canonical model realization. It can also be analysis of controllability and reachability, observability and constructibility.

- **Controller design.** A pole placement controller can be easily designed if the model is on a so called *controller canonical form.*

- *System identification.* If we want to fit a model to real system input and output data, then it may be more easy to find the best model parameters such that the model output is as similar to the real output data, if we have as few free parameters as possible. Canonical form models is a central step in classical system identification methods such as e.g. the prediction error method. It has been a tremendous research activity on canonical forms.

- **Choice of state variables.** Canonical form models have a close relationship with the choice of state variables. As an example, we can transform a given physical state variable, x, to a so called controllable canonical form by using, $z = C_n^{-1}x$ where C_n is the controllability matrix of the system. z is here representing the transformed state space.

We are in the following not considering diagonal and block diagonal forms, i.e., so called eigenvalue canonical form. The most common canonical forms in system theory are as follows:

1. **Controller canonical form.** (No: regulator kanonisk form)
2. **Controllability canonical form.** (No: styrbarhets kanonisk form)
3. **Observer canonical form.** (No: estimator kanonisk form)
4. **Observability canonical form.** (No: Observerbarhets kanonisk form)

2.2 Controller canonical form

2.2.1 From transfer function to controller canonical form

Only single input single output (SISO) systems are considered in this section. In order to illustrate the controller canonical form consider a transfer function model with $n = 3$ states, i.e.,

$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \tag{2.1}$$

$$y(s) = G(s)u(s) \tag{2.2}$$

The following 3rd order state space model is equivalent.

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{2.3}$$

$$y = \begin{bmatrix} b_0 & b_1 & b_2 \end{bmatrix} x \tag{2.4}$$

In order to show this we can write down a block diagram for the system based on the two model representations and show that the resulting block diagram is the same. Another method for proving the equivalence is to compute the transfer function model from the state space model.

The above state space model realization is not unique. For instance rows and columns in the model matrices can be interchanged. This can be easily done by a so called permutation matrix. Consider the following permutation matrix, P, defined by

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.5}$$

Note the following properties for permutation matrices, i.e., $P = P^{-1} = P^T$.

The resulting state space model obtained from the transformation $z = Px$ is given by

$$\dot{z} = \begin{bmatrix} -a_2 & -a_1 & -a_0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} z + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u \tag{2.6}$$

$$y = \begin{bmatrix} b_2 & b_1 & b_0 \end{bmatrix} z \tag{2.7}$$

It should be noted that it this last state space form which is referred to as *controller canonical form*.

Those two canonical forms, i.e., the formulation given by (2.54) and (2.4) and the formulation given by (2.12) and (2.7) are essentially identical state space canonical forms.

Example 2.1 (2nd order system)

Given a SISO process described by a 2nd order transfer function model

$$G(s) = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} \tag{2.8}$$

$$y(s) = G(s)u(s) \tag{2.9}$$

An equivalent 2nd order linear state space model is of the form

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.10}$$

$$y = \begin{bmatrix} b_0 & b_1 \end{bmatrix} x \tag{2.11}$$

and equivalently by using the transformation $z = Px$ with permutation matrix $P = P^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ we obtain the controller canonical form

$$\dot{z} = \begin{bmatrix} -a_1 & -a_0 \\ 1 & 0 \end{bmatrix} z + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{2.12}$$

$$y = \begin{bmatrix} b_1 & b_0 \end{bmatrix} z \tag{2.13}$$

The simplest is to prove this by taking the Laplace transformation of the state space model Eq. (2.11).

2.2.2 From state space form to controller canonical form

Single input multiple output (SIMO) systems

In the following a method for the transformation of a SIMO controllable realization given by (A, B, D) to so called controller canonical form is presented.

From the system matrix, A, we find the characteristic polynomial

$$\det(sI - A) = s^n + a_1 s^{n-1} + \cdots a_{n-1} s + a_n$$

from the coefficients $a_1, \cdots, a_{n-1}$ in the characteristic polynomial we obtain the following upper triangular matrix with ones on the main diagonal.

$$M = \begin{bmatrix} 1 & a_1 & \cdots & a_{n-1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & & \ddots & a_1 \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{2.14}$$

The matrix M has 1 on the main diagonal, a_1 on the diagonal above, and so on. The matrix, M is referred to as a Toeplitz matrix because of its structure. Matrices which elements are constant along its diagonals are called Toeplitz matrices.

Consider now the transformation matrix

$$T = C_n M \tag{2.15}$$

where C_n is the controllability matrix for the pair (A, B) given by

$$C_n = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix} \tag{2.16}$$

The resulting state space model obtained by the transformation

$$z = T^{-1} x \tag{2.17}$$

is on so called controller canonical form. The structure of the model will be as illustrated in (2.12) and (2.7).

The system output matrix DT in the regulator canonical form will in general be a full matrix, i.e., with parameters different from zeros and ones. It will be mn free parameters in the output matrix DT and n free parameters in the controller canonical form system matrix $T^{-1}AT$. The input matrix $T^{-1}B$ will contain only zeros and ones. The number n of free parameters in $T^{-1}AT$ is identical to the coefficients $a_1, \cdots, a_{n-1}, a_n$ in the characteristic polynomial. It is a number of $mn + n$ free and independent parameters in a controller canonical state space model realization.

2.3 Controllability canonical form

Single input systems

If the system is controllable, then a non singular trnaformation matrix can be obtained directly from the controllability matrix. Define the transformation matrix

$$T = C_n \tag{2.18}$$

where C_n is the controllability matrix for the pair (A, B).

The resulting transformed state space model by using $z = T^{-1}x$ will have the following structure

$$\begin{aligned} \dot{z} &= \overbrace{\begin{bmatrix} 0 & 0 & -a_0 \\ 1 & 0 & -a_1 \\ 0 & 1 & -a_2 \end{bmatrix}}^{A_{co}} z + \overbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}}^{B_{co}} u \\ y &= DTz \end{aligned} \tag{2.19}$$

where $A_{co} = T^{-1}AT$ and $B_{co} = T^{-1}B$. This is referred to as controllable canonical form. The system matrix A_{co} is said to be on controllable canonical form. Note that the coefficients of the characteristic polynomial $\det(\lambda I - A)$ can be obtained from the right column in A_{co}.

Multiple input systems

The method described above will often also work for systems with multiple inputs. Note that the controllability matrix, C_n, is an $(n \times nr)$ matrix. A transformation matrix T can often be taken directly as the n first columns in C_n. However, a problem which may occur is that the n first columns is linearly dependent and that the transformation matrix T thereby becomes singular. However, when the system is controllable then we can obtain n linearly independent columns from the controllability matrix $C_n \in \mathbb{R}^{\ltimes \times \ltimes \setminus}$, and take the transformation matrix T as those columns.

Remark 2.1 *It is worth noticing that for SISO systems, a controllable canonical form state space model can be obtained by first constructing the extended controllability matrix*

$$C_{n+1} = \begin{bmatrix} B & AB & \cdots & A^{n-1}B & A^nB \end{bmatrix} \tag{2.20}$$

and thereby in MATLAB notation

$$Aco = (C_{n+1}(:, 1:n))^{-1} C_{n+1}(:, 2:n+1) \tag{2.21}$$

Hence, the system matrix A_{co} is on controllable canonical form.

2.4 Observer canonical form

2.4.1 From transfer function to observer canonical form

Given the transfer function model as presented in the above Section, 2.2.1, i.e.,

$$\begin{aligned} G(s) &= \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0} \\ y(s) &= G(s)u(s) \end{aligned} \tag{2.22}$$

A state space model on observer canonical form can be developed from the transfer function model as presented in the following. Equation (2.22) can be written as follows

$$\frac{y(s)}{u(s)} = \frac{b_2 s^{-1} + b_1 s^{-2} + b_0 s^{-3}}{1 + a_2 s^{-1} + a_1 s^{-2} + a_0 s^{-3}}$$

by multiplying the denominator and nominator in Equation (2.22) with s^{-3}. This gives

$$y(s) = (b_2 u(s) - a_2 y(s))s^{-1} + (b_1 u(s) - a_1 y(s))s^{-2} + (b_0 u(s) - a_0 y(s))s^{-3} \tag{2.23}$$

Let us make the following definitions

$$\begin{aligned} s x_1(s) &\overset{\text{def}}{=} b_2 u(s) - a_2 y(s) + x_2(s) && (2.24)\\ s x_2(s) &\overset{\text{def}}{=} b_1 u(s) - a_1 y(s) + x_3(s) && (2.25)\\ s x_3(s) &\overset{\text{def}}{=} b_0 u(s) - a_0 y(s) && (2.26)\\ y(s) &\overset{\text{def}}{=} x_1(s) && (2.27) \end{aligned}$$

Putting the definitions, (2.24)-(2.26), into the expression given by Equation (2.23) gives that $y(s) = x_1(s)$, as is exactly the same as the definition, Equation (2.27). This means that the model given by the definitions, Equations (2.24)-(2.26), is an equivalent representation of the original transfer function model, Equation (2.22). Transforming the model given by Equations (2.24)-(2.27) to the state space (inverse Laplace transformation) gives

$$\begin{aligned} \dot{x}_1 &= -a_2 x_1 + x_2 + b_2 u\\ \dot{x}_2 &= -a_1 x_1 + x_3 + b_1 u\\ \dot{x}_3 &= -a_0 x_1 + b_0 u\\ y &= x_1 \end{aligned}$$

The last state space model can simply be written on so called *observer canonical form*, Hence we have

$$\begin{aligned} \dot{x} &= \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} u\\ y &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x && (2.28) \end{aligned}$$

Example 2.2 (Sinus)
Given a process described by a sinusoid movement. This process may be e.g. a disturbance, like waves and temperatures. Let

$$y(t) = \sin(\omega t) \tag{2.29}$$

A sinusoid disturbance is common in many processes. This example is therefore of importance when modeling disturbances. We want to write a state space description *of the sinusoid process. The sinusoid time domain description of the process can in the Laplace plane be written as follows, see e.g. (se Edgar et al (1989), s. 47)*

$$y(s) = \frac{\omega}{\omega^2 + s^2} u(s)$$

where $u(s) = 1$, is an unit impulse at time $t = t_0$, e.g. at time $t_0 = 0$. This Laplace formulation can be written as

$$s^2 y(s) = \underbrace{\underbrace{\omega u(s) - \omega^2 y(s)}_{sx_2}}_{s^2 x_1}$$

here, we define

$$\begin{array}{lllllll} s^2 y(s) & = & s^2 x_1 & \Rightarrow & y & = & x_1 \\ sx_2 & = & \omega u(s) - \omega^2 y(s) & \Rightarrow & \dot{x}_2 & = & \omega u(t) - \omega^2 y(t) \\ s^2 x_1 & = & sx_2 & \Rightarrow & \dot{x}_1 & = & x_2 \end{array}$$

This gives the state space model

$$\overbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}^{\dot{x}} = \overbrace{\begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}^{x} + \overbrace{\begin{bmatrix} 0 \\ \omega \end{bmatrix}}^{B} u(t) \tag{2.30}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{D} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.31}$$

where $u(t)$ is a unit impulse given by

$$u(t) = \begin{cases} 1 & \mathit{for} \quad t = t_0 \\ 0 & \mathit{for\ al} \quad t > t_0 \end{cases} \tag{2.32}$$

Example 2.3 (Shifted sinus)
Given a process described by a sinusoid wave

$$y(t) = \sin(\omega t + \phi). \tag{2.33}$$

The corresponding Laplace transformation is (se e.g. Edgar et al (1989), s. 47)

$$y(s) = \frac{s\sin(\phi) + \omega\cos(\phi)}{s^2 + \omega^2} u(s)$$

Using (2.11) gives an equivalent state space model

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\omega^2 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{2.34}$$

$$y = \begin{bmatrix} \omega\cos(\phi) & \sin(\phi) \end{bmatrix} x \tag{2.35}$$

where $u(t)$ is a unit impulse. In this example we have used the results in Example 2.1.

Example 2.4 (Observability canonical form: 2nd order system)
Given a 2nd order system described by the transfer function model

$$y = \frac{b_1 s + b_0}{s^2 + a_1 s + a_0} u. \tag{2.36}$$

Multiply eq. (2.36 with the de-numerator on the right hand side and write

$$s(s + a_1)y + a_0 y = b_1 s u + b_0 u \tag{2.37}$$

Collect all terms propertional with s on the left hand side and write

$$s \overbrace{(sy + a_1 y - b_1 u)}^{x_2} y + a_0 y = b_0 u \tag{2.38}$$

Simply define $y = x_1$ and

$$x_2 = sy + a_1 y - b_1 u \tag{2.39}$$

This gives from eq. (2.39)

$$\dot{x}_1 = -a_1 x_1 + x_2 + b_1 u \tag{2.40}$$

and from eq. (2.38) we obtain

$$\dot{x}_2 = -a_0 x_1 + b_0 u \tag{2.41}$$

Eqs. (2.40) and (2.41) gives the linear state space model equivalent

$$\overbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}}^{\dot{x}} = \overbrace{\begin{bmatrix} -a_1 & 1 \\ -a_0 & 0 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}^{x} + \overbrace{\begin{bmatrix} b_1 \\ b_0 \end{bmatrix}}^{B} u \tag{2.42}$$

$$y = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{D} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{2.43}$$

Hence, eq. (2.43) is a 2nd order ($n = 2$ states) state space model on observer canonical form.

2.5 Observability canonical form

The observability canonical form is commonly used in system identification, i.e., how to find the model parameters such that the model fits the real system data. The name have its origin because a transformation matrix, T, is constructed from n linearly independent rows in the observability matrix, O_n

Given a realization (A, B, D). The observability matrix for the pair (A, D) is defined by

$$O_L = \begin{bmatrix} D \\ DA \\ DA^2 \\ \vdots \\ DA^{L-1} \end{bmatrix} \tag{2.44}$$

O_L is denoted an extended observability matrix when $L > n$ and as an observability matrix when $L = n$. An algorithm for transforming an observable state space model (A, B, D) to observable canonical form is as follows.

Algorithm 2.5.1 (observability canonical form)
Step 1: *Construct a non singular transformation matrix, T, from n linearly dependent rows in the observability matrix O_L where $L \geq n$. We can chose*

$$T = O_L(1:n,:)$$

when this choice is non singular. This will always hold for SISO observable systems.
Step 2: *An observability canonical form state space model is then given by*

$$\begin{aligned} A_c &= (O_c^T O_c)^{-1} O_c^T O_L A T^{-1} && (2.45) \\ B_c &= (O_c^T O_c)^{-1} O_c^T O_L B && (2.46) \\ D_c &= D T^{-1} && (2.47) \end{aligned}$$

where

$$O_c = O_L T^{-1} \tag{2.48}$$

Note that $O_c = I$ for single output systems. This is not the case for multiple output systems.

This Algorithm 2.5.1 works for multiple output and multiple input (MIMO) systems. The critical step is step 1. For systems with multiple outputs it is not sure that the n first rows in O_L is linearly independent. In such cases one have to search through O_L in order to obtain n linearly dependent rows, and use this rows as a basis for T.

Example 2.5 (From transfer function model to OCF)
To illustrate the observability canonical form (OCF) we here use a 2nd order system described with the transfer function model

$$y = \frac{b_0}{s^2 + a_1 s + a_0} u \tag{2.49}$$

From (2.49) we express with definitions

$$s^2 y = \underbrace{\underbrace{-a_0 y - a_1 s y + b_0 u}_{s x_2}}_{s^2 x_1} \tag{2.50}$$

Hence, from (2.50) we have the following definitions

$$\begin{array}{lllllll} s^2 x_1 & = & s x_2 & \Rightarrow & \dot{x}_1 & = & x_2 \\ s x_2 & = & -a_0 y - a_1 s y + b_0 u & \Rightarrow & \dot{x}_2 & = & -a_0 x_1 - a_1 x_2 \\ s^2 y & = & s^2 x_1 & \Rightarrow & y & = & x_1 \end{array}$$

From this we find the following 2nd order state space model on observability canonical form

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} 0 \\ b_0 \end{bmatrix} u \tag{2.51}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x \tag{2.52}$$

where $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ *is the state vector.*

Example 2.6 (Transfer function model and OCF)
Consider a system described with the following transfer function model

$$y = \frac{b_0 + sb_1}{s^2 + a_1 s + a_0} u \tag{2.53}$$

The equivalent observability canonical form state space model is

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} x + \begin{bmatrix} b_1 \\ b_0 \end{bmatrix} u \tag{2.54}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x \tag{2.55}$$

The simplest way to prove this is to go from the state space form (2.55) to the transfer function model (2.53).

2.6 Duality between canonical forms

We are in the following going to discuss the principle of duality between canonical form and state space representations. The importance of this is that it normally is enough to learn one method of how to transform a given state space model to canonical form. The reason for this is because the duality principle can be used to, e.g., transform an observer canonical form to a controller canonical form. There is a duality between the observer and controller canonical forms.

2.6.1 Duality between controller and observer canonical forms

A closer study of the controller and observer canonical forms, and the controllability and observability canonical forms, shows that those are what we call dual canonical forms.

Let x_c be the state vector for the system in controller canonical form, (index c for *controller form*) with realization (A_c, B_c, D_c), i.e.,

$$\dot{x}_c = A_c x_c + B_c u \tag{2.56}$$

$$y = D_c x_c \tag{2.57}$$

The transformation matrix, T, witch transforms a realization (A, B, D) to controller canonical form is given by

$$T_c = C_n M \tag{2.58}$$

where M is an upper triangular Toeplitz matrix formed from the coefficients in the characteristic polynomial.

Similarly, let x_o be the state vector for a system on observer canonical form, (index o for *observer form*) with realization (A_o, B_o, D_o), i.e.,

$$\dot{x}_c = A_o x_o + B_o u \tag{2.59}$$

$$y = D_o x_o \tag{2.60}$$

The transformation matrix, T, that transforms a given realization (A, B, D) to so called observer canonical from is then given by

$$T_o = O_n^{-1}(M^T)^{-1} \tag{2.61}$$

where M is the Toplitz matrix.

Comparing the two canonical forms we find the following connections:

$$A_c = A_o^T \quad B_c = D_o^T \quad D_c = B_o^T \tag{2.62}$$

This phenomena is what we call *controller and observer duality*.

2.6.2 Duality between controllability and observability canonical forms

In the same way, there is a duality between controllability and observability canonical forms.

As we have shown,

$$T_{co} = C_n \tag{2.63}$$

where C_n is the controllability matrix of the system, is the transformation matrix which transform a given realization (A, B, D) to controllable canonical form, given by

$$\dot{x}_{co} = A_{co} x_{co} + B_{co} u \tag{2.64}$$

$$y = D_{co} x_{co} \tag{2.65}$$

where index *co* denotes *controllability form*. In the same way

$$T_{ob} = O_n^{-1} \tag{2.66}$$

is the transformation matrix $x = T x_{ob}$ which transforms a given realization (A, B, D) to so called observability canonical form, given by

$$\dot{x}_{co} = A_{ob} x_{ob} + B_{ob} u \tag{2.67}$$

$$y = D_{ob} x_{ob} \tag{2.68}$$

where index *ob* stand for *observability form*. Comparing the two canonical forms shows that:

$$A_{co} = A_{ob}^T \quad B_{co} = D_{ob}^T \quad D_{co} = B_{ob}^T \tag{2.69}$$

2.7 Examples

Example 2.7 (Synthesis of controller based on COF)
Given a state space model on COntroller canonical Form (COF)

$$\dot{x} = \overbrace{\begin{bmatrix} -a_1 & -a_0 \\ 1 & 0 \end{bmatrix}}^{A} x + \overbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}^{B} u \tag{2.70}$$

Problem: *We want ti find a controller of the form Vi ø*

$$u = -G(x^s - x) + u^s \tag{2.71}$$

$$G = \begin{bmatrix} g_1 & g_2 \end{bmatrix} \tag{2.72}$$

in such a way that the closed loop system has the eigenvalues (poles) given by s_1 and s_2.

Solution: *The closed loop system is described by*

$$\dot{x} = (A + BG)x + \begin{bmatrix} B - GB \end{bmatrix} \begin{bmatrix} u^s \\ x^s \end{bmatrix} \tag{2.73}$$

$$A + BG = \begin{bmatrix} -a - 1 + g_1 & -a_2 + g_2 \\ 1 & 0 \end{bmatrix} \tag{2.74}$$

Note that the closed loop system matrix, A+BG, also is on controller canonical form. An expression for the eigenvalues can be found from the characteristic equation, i.e.

$$|sI - (A + BG)| = \begin{vmatrix} s + a_1 - g_1 & a_2 - g_2 \\ -1 & s \end{vmatrix} = s^2 + (a_1 - g_1)s + a_2 - g_2 = 0 \tag{2.75}$$

If the eigenvalues of the closed loop system should be s_1 and s_2 then the closed loop characteristic equation must be være

$$(s - s_1)(s - s_2) = s^2 - (s_1 + s_2)s + s_1 s_2 = 0 \tag{2.76}$$

Comparing the coefficients in the two polynomials gives

$$a_2 - g_2 = s_1 s_2 \tag{2.77}$$

$$a_1 - g_1 = -(s_1 + s_2) \tag{2.78}$$

This gives the coefficients g_1 and g_2 in the controller as a function of the prescribed eigenvalues s_1 and s_2 and the system parameters a_1 and a_2. We have

$$g_2 = a_2 - s_1 s_2 \tag{2.79}$$

$$g_1 = a_1 + s_1 + s_2 \tag{2.80}$$

Example 2.8 (Four canonical forms)
Given a system

$$\dot{x} = Ax + Bu \tag{2.81}$$

$$y = Dx + Eu \tag{2.82}$$

where

$$\begin{array}{c} A = \begin{bmatrix} -4 & 3 & 2 \\ 2 & -6 & 1 \\ 1 & 2 & -8 \end{bmatrix} \quad B = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \\ D = \begin{bmatrix} 2 & -1 & 1 \end{bmatrix} \end{array} \tag{2.83}$$

We will in this example compute all the four canonical forms which are described in this section. Nota that the results would have been the same if the given state space form had been a discrete time system. A MATLAB script for the computations of the canonical forms is given in Example 2.9. Running the script within MATLAB gives the following results.

Controller canonical form (cof):

$$\begin{array}{c} A_{\text{krf}} = \begin{bmatrix} -18 & -94 & -113 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad B_{\text{krf}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ D_{\text{krf}} = \begin{bmatrix} 3 & 58 & 279 \end{bmatrix} \end{array} \tag{2.84}$$

Observer canonical form (obf):

$$\begin{array}{c} A_{\text{kef}} = \begin{bmatrix} -18 & 1 & 0 \\ -94 & 0 & 1 \\ -113 & 0 & 0 \end{bmatrix} \quad B_{\text{kef}} = \begin{bmatrix} 3 \\ 58 \\ 279 \end{bmatrix} \\ D_{\text{kef}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{array} \tag{2.85}$$

Controllable canonical form (cf):

$$\begin{array}{c} A_{\text{ksf}} = \begin{bmatrix} 0 & 0 & -113 \\ 1 & 0 & -94 \\ 0 & 1 & -18 \end{bmatrix} \quad B_{\text{ksf}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ D_{\text{ksf}} = \begin{bmatrix} 3 & 4 & -75 \end{bmatrix} \end{array} \tag{2.86}$$

Observability canonical form (of):

$$\begin{array}{c} A_{\text{kof}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -113 & -94 & -18 \end{bmatrix} \quad B_{\text{kof}} = \begin{bmatrix} 3 \\ 4 \\ -75 \end{bmatrix} \\ D_{\text{kof}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{array} \tag{2.87}$$

Note the following connections:

$$\begin{array}{lll} A_{\text{kof}} = A_{\text{ksf}}^T & B_{\text{kof}} = D_{\text{ksf}}^T & D_{\text{kof}} = B_{\text{ksf}}^T \\ A_{\text{kef}} = A_{\text{krf}}^T & B_{\text{kef}} = D_{\text{krf}}^T & D_{\text{kef}} = B_{\text{krf}}^T \end{array} \tag{2.88}$$

Example 2.9 (Matlab script for computation of canonical forms)

```
disp(' SYSTEM MODELL) MATRICES:')
A=[-4,3,2;2,-6,1;1,2,-8]
B=[1;2;3]
D=[2,-1,1]

disp('1. CONTROLLER CANONICAL FORM (krf):')
disp('...HIT e key to continue'), pause
Qs=ctrb(A,B);
alfa=poly(A); M=eye(3); M(1,2)=alfa(2); M(2,3)=alfa(2); M(1,3)=alfa(3);
Tkrf=Qs*M;
Akrf=inv(Tkrf)*A*Tkrf, Bkrf=inv(Tkrf)*B, Dkrf=D*Tkrf

disp('2. OBSERVER CANONICAL FORM (kef):')
disp('...hit a key to continue'), pause
Qo=obsv(A,D);
Tkef=inv(Qo)*inv(M');
Akef=inv(Tkef)*A*Tkef, Bkef=inv(Tkef)*B, Dkef=D*Tkef

disp('3. CONTROLLABILITY CANONICAL FORM( ksf):')
disp('...hit a key to continue'), pause
Tksf=Qs;
Aksf=inv(Tksf)*A*Tksf, Bksf=inv(Tksf)*B, Dksf=D*Tksf

disp('4. OBSERVABILITY CANONICAL FORM (kof):')
disp('...hit a key to continue'), pause
Tkof=inv(Qo);
Akof=inv(Tkof)*A*Tkof, Bkof=inv(Tkof)*B, Dkof=D*Tkof
```

2.8 Summary

It is worth noticing that the problem of computing canonical forms is sensitive for numerical errors. This is mostly a problem in MIMO systems. Even for controllable and observable systems, the transformation matrix, T, constructed from independent columns in the controllability matrix (ore rows in the observability matrix) may be close to singular.

Note also that the system must be controllable for the controller and controllable canonical forms to exists. Similarly, the system must be observable for the observer and observability canonical forms to exists.

The parameters in a canonical form may be very sensitive to perturbations. This can results in a poor choice for using the canonical form in a prediction error system identification method. In some circumstances there may be impossible to find a canonical form. This problem may occur for MIMO systems and is usually not a problem for SISO systems.

There exists other essentially unique state space model realizations with good numerical properties. The most common is *balanced realization*, *input normal realization* and *output normal realization*. Those realizations are computed through a singular value decomposition. This is not a topic here.

Part II

Analysis and design of control systems

Chapter 3

Simple PID tuning and model reduction

3.1 Feedback systems

Intuitively, feed-forward control gives perfect control in case of an exact model. However, models are never exact and the control system will have to involve feedback.

The main reasons for using feedback control are in case of unknown disturbances, model uncertainty and unstable plants. Feedback systems are studied in the following.

3.2 Standard feedback system

Consider a standard feedback system as illustrated in Figure 3.1.

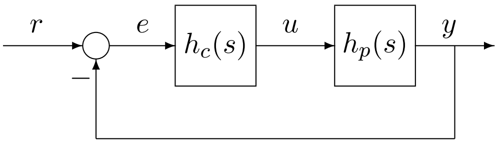

Figure 3.1: Standard feedback system. Plant described by a transfer function model $h_p(s)$ and controller transfer function $h_c(s)$.

In Figure 3.1 y is the measured plant output assumed for simplicity to be equal to the desired output. Often in the feedback path there is an implementation of the function $y_m = h_m(s)y$ where y_m is the measurement of the desired output y. Here we assume, as explained, that the measurement system is exact and that $h_m(s) = 1$ and that $y_m = y$. In Figure 3.1 r is the specified reference for the plant output. The error difference $e = r - y$ is the input signal to the controller represented by the transfer function $h_c(s)$. Here the true plant is represented with a transfer function model $h_p(s)$. In the following we will study this standard feedback system.

3.3 Standard feedback systems with disturbances

Usually, in addition to the input u the system is also influenced by an external disturbance v as illustrated in Figure 3.2

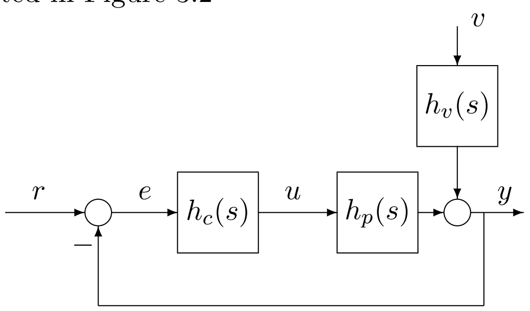

Figure 3.2: Standard feedback system with disturbance v at the output. Plant described by a transfer function model $h_p(s)$, disturbance model $h_v(s)$ and controller transfer function $h_c(s)$.

Hence, the plant may be influenced both from the control input u and an external disturbance v. The plant is illustrated with the dashed box in Figure 3.3.

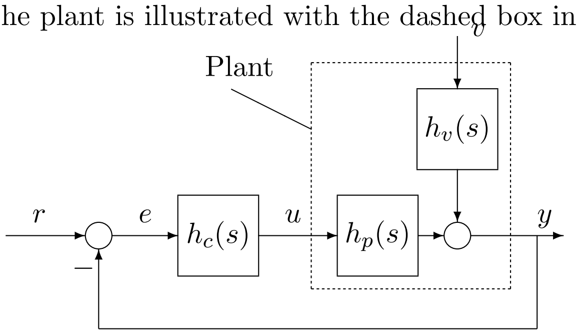

Figure 3.3: Standard feedback system with disturbance v at the output. Plant described by a transfer function model $h_p(s)$, disturbance model $h_v(s)$ and controller transfer function $h_c(s)$. Plant indicated with the dashed box.

A more general situation is as illustrated in Figure 3.4 where the measurement system is illustrated in the dashed box and where the actual measurement y_m is related to the desired output y as $y_m = h_m(s)y + w$ where w is measurement noise, and $h_m(s)$ a model for the measurement system. Usually, as described above we assume $h_m(s) = 1$. The measurement noise w is usually high frequent noise but may also represent drifts etc.

3.4 Simple PID tuning rules

We will in this section present a simple and direct method for PID controller tuning and design. The method is very simple and intuitive and leads to a practical and robust method which also give very good results. The method is based on an approximate model of 1st or 2nd order with time delay or inverse response. If the

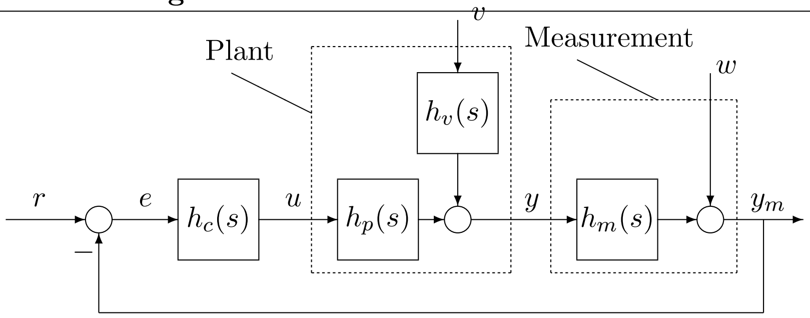

Figure 3.4: Standard feedback system with disturbance v at the output. Plant described by a transfer function model $h_p(s)$, a disturbance model $h_v(s)$ and controller transfer function $h_c(s)$. Plant indicated with the dashed box. The actual measurement y_m is related to the desired output y as $y_m = h_m(s)y + w$ where w is measurement noise.

process is approximated with a 2nd order model with time delay then a PID controller is the result, but if a 1st order model with time delay is used a PI controller is the result. Hence, the resulting controller is defined by the starting point which is the approximate model. Only stable models is considered,

Consider an approximate 2nd order model with time delay given by the following transfer function

$$h_p(s) = k\frac{e^{-\tau s}}{(1+T_1 s)(1+T_2 s)}. \tag{3.1}$$

or with inverse response as follows

$$h_p(s) = k\frac{1-\tau s}{(1+T_1 s)(1+T_2 s)}, \tag{3.2}$$

Note that (3.2) is an approximation of the exact time delay model (3.1) because,

$$e^{-\tau s} \approx 1 - \tau s. \tag{3.3}$$

The method can still be used if the second time constant, $T_2 = 0$. Then we have an approximate model of the form

$$h_p(s) = k\frac{e^{-\tau s}}{1+T_1 s} \approx k\frac{1-\tau s}{1+T_1 s}. \tag{3.4}$$

If the starting point is a more complicated higher order model or you have identified a higher order model from data by system identification methods such as, e.g. DSR, then a 1st or 2nd order model approximation can be constructed by model reduction techniques. System identification methods can possibly also be used to construct a lower order model directly.

In the above low order models the parameter τ represents the effective time delay or the inverse response time and the time constant $T_1 \geq T_2 \geq 0$ is the dominant time constant. Note that many higher order models with time delay can be approximated by a 1st or 2nd order model with time delay. A method for model reduction which is to be used in this section is the half-rule.

3.4.1 The half rule

As an introductory example of the half-rule consider the system (3.1) or (3.2) and assume that this system do not have dominant 2nd order dynamics, then this model can with advantage be approximated with a 1st order model with inverse response as given in the following model (3.5).

$$h_p(s) = k\frac{1-\tau s}{1+T_1 s}, \tag{3.5}$$

where we from the half rule find the parameters

$$\begin{aligned} T_1 &:= T_1 + \tfrac{1}{2}T_2, && (3.6)\\ \tau &:= \tau + \tfrac{1}{2}T_2. && (3.7) \end{aligned}$$

The half rule says that the neglected smallest time constant T_2 is distributed evenly over the remaining time constant $T1$ and the time delay or inverse response time τ.

The model (3.5) is a very good starting point for the tuning of a PI controller. If the system has dominant 2nd order dynamics in such a way that it is not wise to neglect T_2, then we use a 2nd order model of the form (3.2) and designing a PID controller.

Example 3.1 (Model reduction by the half rule)
Given a 4th order process $y = h_p(s)u$ where the process is described by the transfer function, h_p, given by

$$h_p(s) = k\frac{e^{-\tau s}}{(1+T_1 s)(1+T_2 s)(1+T_3 s)(1+T_4 s)} \tag{3.8}$$

where $T_1 \geq T_2 \geq T_3 \geq T_4 > 0$ is the time constants and τ is the time delay.

Using the half rule gives a 1st order model with time delay approximation

$$h_p(s) = k\frac{1-\tau s}{1+T_1 s}, \tag{3.9}$$

where

$$\begin{aligned} T_1 &:= T_1 + \frac{1}{2}T_2, && (3.10)\\ \tau &:= \tau + \frac{1}{2}T_2 + T_3 + T_4, && (3.11) \end{aligned}$$

or by a 2nd order model approximation

$$h_p(s) = k\frac{1-\tau s}{(1+T_1 s)(1+T_2 s)}, \tag{3.12}$$

where

$$\begin{aligned} T_1 &:= T_1, && (3.13)\\ T_2 &:= T_2 + \frac{1}{2}T_3, && (3.14)\\ \tau &:= \tau + \frac{1}{2}T_3 + T_4. && (3.15) \end{aligned}$$

Example 3.2 (Model reduction by the half rule)
Given a 4th order process $y = h_p(s)u$ where the process is described by the transfer function, h_p, given by

$$h_p(s) = 2\frac{1}{(1+6s)(1+4s)(1+2s)(1+1s)}. \tag{3.16}$$

This model may be approximated by a 1st order+time delay model

$$h_p^{1a}(s) = 2\frac{e^{-5s}}{1+8s}, \tag{3.17}$$

i.e., the time constant is $T = 6 + \frac{1}{2}4 = 8$ and the time delay is $\tau = \frac{1}{2}4 + 2 + 1 = 5$. Remark: a better 1st order approximation at high frequencies will be $h_p^{1a}(s) = 2\frac{e^{-5s}}{1+10s}$.

The model may be approximated by a 2nd order+time delay model

$$h_p^{2a}(s) = 2\frac{e^{-2s}}{(1+6s)(1+5s)}, \tag{3.18}$$

where the first time constant is $T_1 := T_1 = 6$, the second is $T_2 := T_2 + \frac{1}{2}T_3$ and the time delay $\tau = \frac{1}{2}T_3 + T_4 = 2$.

The step responses are illustrated in Figure 3.5. The step response is implemented in the MATLAB m-file **ex2_half.m**.

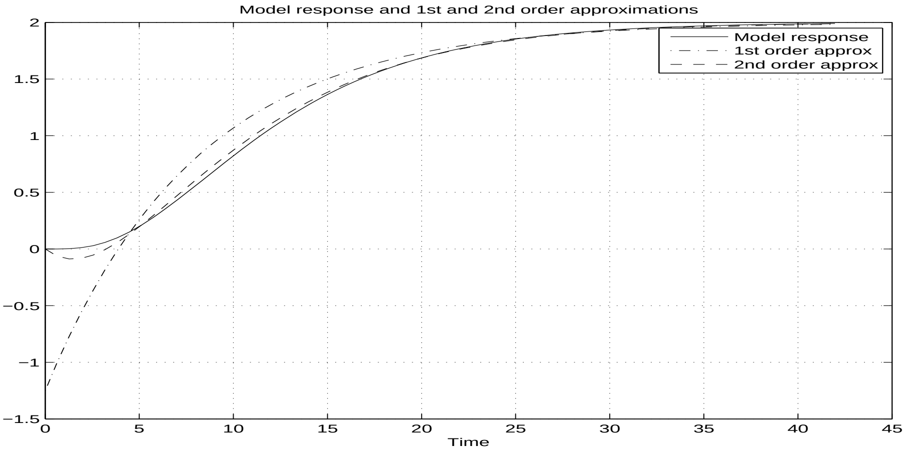

Figure 3.5: Step response of model $h_p(s)$ and the 1st and 2nd order model approximations in Example 3.2 The time delay is approximated with an inverse response term in the responses, i.e. $e^{-\tau s} \approx 1 - \tau s$.

3.4.2 Tuning of PI and PID controllers

Let us specify the response from the reference signal, r, to the measurement output, y, as follows

$$\frac{y}{r} = \frac{1 - \tau s}{1 + T_c s}. \tag{3.19}$$

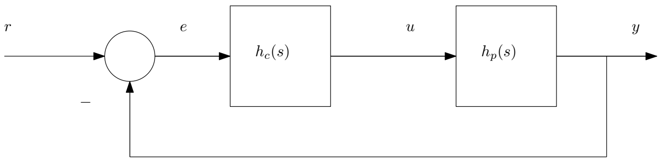

Figure 3.6: Standard feedback system with process $h_p(s)$ and controller $h_c(s)$.

here T_c is a user specified time constant for the set-point response. The choice for T_c will be discussed later but it is natural to chose it related to the time delay or inverse response time τ. It is nothing to do with a transport delay or inverse response phenomena in a process (when the process already is designed), hence the time delay ore inverse response must also be in the set point response $\frac{y}{r}$. This is the reason for the numerator polynomial $1 - \tau s$ in (3.19).

We also know from the block diagram in Figure 3.6 of the control system that

$$\frac{y}{r} = \frac{h_p h_c}{1 + h_p h_c}, \tag{3.20}$$

where $h_c(s)$ is the transfer function for the controller, which at this stage is unknown. Solving for the controller gives

$$h_c(s) = \frac{1}{h_p(s)} \frac{\frac{y}{r}}{1 - \frac{y}{r}} = \frac{1}{h_p(s)} \frac{1}{\frac{1}{\frac{y}{r}} - 1}. \tag{3.21}$$

Putting the transfer functions (3.19) and (3.2) (or (3.5)) into the expression (3.21) for the controller, $h_c(s)$. This gives the following controller

$$h_c(s) = \frac{1}{k} \frac{T_1}{T_c + \tau} \frac{1 + T_1 s}{T_1 s} (1 + T_2 s). \tag{3.22}$$

This is a so called cascade formulation of a PID controller.

$$h_c(s) = K_p \frac{1 + T_i s}{T_i s} (1 + T_d s), \tag{3.23}$$

where the parameters in the cascade PID controller are given by

$$\begin{aligned} K_p &= \frac{1}{k} \frac{T_1}{T_c + \tau}, && (3.24) \\ T_i &= T_1 && (3.25) \\ T_d &= T_2. && (3.26) \end{aligned}$$

Derivative action is usually only recommended for systems with dominant 2nd order dynamics, i.e. with a large time constant T_2, say large relative to the time delay and in case where $T_2 > \tau$ is a rule of thumb. Notice also in connection with this that in case of high frequent noise on the output, it make sense to skip derivative action in order not to amplify noise on the control action and to the system.

Note that if we instead have used (3.5)) in (3.21) then a PI controller

$$h_c(s) = K_p \frac{1 + T_i s}{T_i s}, \tag{3.27}$$

with parameters as given in (3.24) and (3.25) have been the result.

At this stage the only unknown parameters in the controller settings is the time constant T_c for the set point response. We can for many processes chose T_c equal (ore grater) then the time delay or the inverse response time τ. Hence, a simple choice is simply

$$T_c = \tau. \tag{3.28}$$

This gives the proportional constant

$$K_p \;=\; \frac{1}{2k}\frac{T_1}{\tau}, \tag{3.29}$$

3.4.3 Choice of T_c

Locking at the formula (3.27) for K_p shows us that it may be natural to force the term $\frac{T_1}{T_c+\tau}$ to be positive. Hence we must chose $T_c + \tau > 0$. This means that T_c can be chosen relatively within wide ranges $-\tau < T_c < \infty$. However, an optimal setting for T_c will be a trade of between:

1. Fast response in order to controlling disturbances. This is favored by a small value of T_c.
2. Stability and robustness and using a small amount of control u. This is favored by a small value of T_c.

A good trade of between these wishes is as presented in Skogestad (2002) to chose as in (3.28). However, the choice

$$T_c \geq \tau. \tag{3.30}$$

Then from (3.24) we have that

$$K_p \leq K_p^{\max} = \frac{1}{2k}\frac{T_1}{\tau} \tag{3.31}$$

3.4.4 Modification of the integral term

3.5 PID controller for oscillating process

Given a process described by

$$h_p(s) = k\frac{e^{-\tau s}}{\tau_0^2 s^2 + 2\tau_0\xi s + 1}, \tag{3.32}$$

where ζ is the relative damping, $\tau_0 = \frac{1}{\omega}$ determines the speed of response, and where ω is the natural frequency. Remark that when $\zeta = 1$ we have two real poles/time constants and a pole polynomial $\tau_0^2 s^2 + 2\tau_0\xi s + 1 = (\tau_0 s + 1)^2$.

Note that $\xi < 1$ gives oscillating process dynamics. The poles of the system is given by

$$s = \frac{-\xi \pm \sqrt{\xi^2 - 1}}{\tau_0}. \tag{3.33}$$

We see that the poles becomes complex when $\xi^2 - 1 < 0$ and that the dynamics may be oscillating in this case. It is possible with oscillating dynamics when ξ is negative and with $|\xi| < 1$ but in this cases we must have $\tau_0 < 0$ in order for the system to be stable.

The period time of the oscillations, P_u, is given by

$$P_u = \frac{2\pi}{\sqrt{1 - \zeta^2}} \tau_0. \tag{3.34}$$

Note that with significant oscillations we have $\zeta^2 << 1$ and $P_u = 2\pi\tau_0$.

Putting (3.32) and (3.19) into (3.21) gives the following controller

$$h_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{3.35}$$

where

$$K_p = \frac{2\tau_0\xi}{k(T_c + \tau)} \tag{3.36}$$

$$T_i = 2\tau_0\xi \tag{3.37}$$

$$T_d = \frac{\tau_0}{2\xi} \tag{3.38}$$

Hence, as we see (3.35) is an ideal PID controller.

Notice that the above PID controller is valid for under-damped processes where the relative damping is $0 < \xi < 1$. When $xi = 1$ we have two real poles and both a PI and a PID controller may be used. The case without damping, i.e. when $xi = 0$ needs to be analyzed separately as in the next subsection.

3.6 ID controller for systems with no damping

In this section we study pure oscillating systems, that is systems with no damping and where $\xi = 0$. This could be the case for a spring or a pendulum without friction. A linearized process model for such a process could be

$$h_p(s) = k \frac{e^{-\tau s}}{\tau_0^2 s^2 + 1}. \tag{3.39}$$

Putting (3.39) and (3.19) into (3.21) gives the following controller

$$h_c(s) = \frac{1}{k(T_c + \tau)s} + \frac{\tau_0^2}{k(T_c + \tau)} s, \tag{3.40}$$

which is a pure ID (Integral and Derivative) controller.

Notice that the PID controller Eq. (3.35) with parameters as in Eqs. (3.36)-(3.38) may be written as

$$h_c(s) = K_p + \frac{K_p}{T_i s} + K_p T_d s, \tag{3.41}$$

where the proportional gain is as in (3.36) and

$$\frac{K_p}{T_i} = \frac{1}{k(T_c + \tau)}, \tag{3.42}$$

$$K_p T_d = \frac{\tau_0^2}{k(T_c + \tau)} \tag{3.43}$$

Hence, this result is consistent with the above result for systems without damping and $\xi = 0$.

3.7 PI Control of first order process

In some simple cases the process may be modeled by a 1st order process

$$h_p(s) = k\frac{1}{1 + Ts}, \tag{3.44}$$

where k is the gain and T the time constant.

For such processes one may add a fictive time delay, τ, due to un-modeled effects and time delay due to sampling, and it make sense to add a time delay $\tau \geq \frac{\Delta t}{2}$ where Δt is the sampling interval.

However, we will in this section use the simple controller tuning method directly from the 1st order model (3.44).

Let us specify the closed loop response as

$$\frac{y}{r}(s) = \frac{1}{1 + T_c s}, \tag{3.45}$$

where T_c is a user specified time constant for the closed loop response from r to y.

Putting (3.44) and (3.45) into (3.21) gives the following controller

$$h_c(s) = \frac{T}{kT_c}\frac{1 + Ts}{Ts} = K_p \frac{1 + T_i s}{T_i s} \tag{3.46}$$

which is a PI controller with

$$K_p = \frac{T}{kT_c} \tag{3.47}$$

$$T_i = T \tag{3.48}$$

Hence, as we see (3.35) is an ideal PI controller.

The prescribed time constant, T_c, should with advantage be specified smaller than the time constant in the process. It make sense to specify T_c up to ten times faster, hence chose $\frac{T}{10} \leq T_c \leq T$. Notice that smaller T_c (faster set-point response) implies higher controller proportional gain K_p which implies higher control action in u and saturation may be a problem. Notice also that physical insight of the process also should be used when specifying time constants

3.8 Integrating process with time delay

Consider a system described by the transfer function

$$h_p(s) = k\frac{e^{-\tau s}}{s}, \tag{3.49}$$

which is an integrator process with time delay. Note that we may approximate Eq. (3.49) as an inverse response with $\tau \geq 0$. Using the method for a process $y = h_p(s)u$ leads to a P-controller with proportional gain

$$K_p = \frac{1}{k(T_c + \tau)} = \frac{1}{2k\tau}, \tag{3.50}$$

where the last equality is obtained by the simple rule of thumb, $T_c = \tau$.

Consider now a feedback system with a disturbance v occuring at the input side of the plant as illustrated in Figure 3.7.

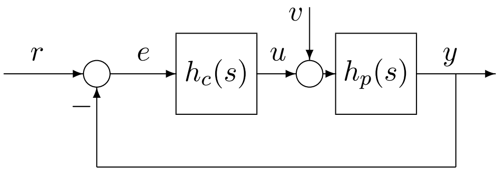

Figure 3.7: Standard feedback system with disturbance at the input. Plant described by a transfer function model $h_p(s)$ and controller transfer function $h_c(s)$.

A P-controller will give set-point error for disturbances at the input, i.e. for systems $y = h_p(s)(u + v)$ because the response from the disturbance to the output then is given by

$$y = \overbrace{\frac{h_c h_p}{1 + h_c h_p}}^{\frac{y}{r}(s)} r + \overbrace{\frac{h_p}{1 + h_c h_p}}^{\frac{y}{v}(s)} v. \tag{3.51}$$

Locking at the response from the disturbance, v, to the output, y, for a process $h_p = k\frac{e^{-\tau s}}{s}$ and a P-controller, i.e., $h_c = K_p$ gives,

$$\frac{y}{v}(s) = \frac{k\frac{e^{-\tau s}}{s}}{1 + Kpk\frac{e^{-\tau s}}{s}}. \tag{3.52}$$

In steady state we have $\frac{y}{v}(s=0) = \frac{1}{K_p}$ and that

$$y = r + \frac{1}{K_p} v. \tag{3.53}$$

This implies that we need a PI-controller for integrating+time delay processes in order to eliminate the offset from load disturbances v at the input, i.e., we need a controller in which $\frac{y}{v}(s=0) = 0$. Note that load disturbances at the output will be removed by using a P-controller, i.e., for systems $y = h_p(s)u + v$ and integrating plus time delay process as in Eq. (3.49).

Table 3.1: PI-controller settings for an integrating plus time delay process $h_p(s) = k\frac{e^{-\tau s}}{s}$ with gain, k, and time delay $\tau \geq 0$. Setting 1 is the Skogestad IMC (SIMC) setting. Settings 2 is suggested by Haugen and settings 3 are proposed in this note. The process model, h_p, used for the derivation of the settings and the corresponding relative damping factor ζ used are indicated in the table for completeness.

	K_p	T_i	$h_p(s)$	ζ
1	$\frac{1}{2k\tau}$	$8\ \tau$	$\frac{k}{s}$	1
2	$\frac{1}{2k\tau}$	$4\ \tau$	$\frac{k}{s}$	$\frac{\sqrt{2}}{2}$
3	$\frac{1}{2k\tau}$	$6\ \tau$	$k\frac{1-\tau s}{s}$	$\frac{\sqrt{3}}{2}$

3.8.1 The SIMC settings: neglecting the time delay

In practice, for the reason of eliminating load disturbances v at the input, i.e., for systems $y = h_p(s)(u+v)$ and in case of un-modeled effects we use a PI controller for integrating processes. The Skogestad PI settings are

$$K_p = \frac{1}{k(T_c + \tau)}, \quad T_i = 4(T_c + \tau) \tag{3.54}$$

and with the time constant for the setpoint response, $T_c = \tau$ we obtain the SIMC settings

$$K_p = \frac{1}{2k\tau}, \quad T_i = 8\tau. \tag{3.55}$$

This may be proved as follows. Consider a 1st order system with time delay, and with a large time constant, T, i.e. we may write the model as

$$h_p(s) = k' \frac{e^{-\tau s}}{1 + Ts} = \frac{k'}{T} \frac{e^{-\tau s}}{\frac{1}{T} + s} \approx k \frac{e^{-\tau s}}{s} \tag{3.56}$$

with $k = \frac{k'}{T}$. For systems with large time constants and neglecting the time delay we obtain the transfer function

$$h_p(s) \approx k \frac{1}{s} \tag{3.57}$$

which is used for the derivation of the SIMC PI-settings.

The pole polynomial for the disturbance and set-point response is obtained from

$$\begin{aligned} 1+h_c h_p &= 1+K_p\frac{1+T_i s}{T_i s}\frac{k}{s} = \frac{1}{s^2}(s^2+\frac{K_p k}{T_i}(1+T_i s)) \\ &= \frac{1}{s^2}\frac{K_p k}{T_i}(\frac{T_i}{K_p k}s^2+T_i s+1)) \end{aligned} \tag{3.58}$$

This gives a pole polynomial on standard second order form as

$$\pi(s) = \frac{T_i}{K_p k}s^2+T_i s+1 = \tau_0^2 s^2+2\zeta\tau_0 s+1 \tag{3.59}$$

By comparing the coefficients in the pole polynomial and the corresponding coefficients in the standard second order polynomial we may find relations between K_p and T_i. We have

$$\tau_0^2 = \frac{T_i}{K_p k},\ 2\zeta\tau_0 = T_i \tag{3.60}$$

This gives $(2\zeta\tau_0)^2 = T_i^2$ and

$$T_i = 4\zeta^2\frac{1}{K_p k} \tag{3.61}$$

Using the setting for the proportional gain, i.e.,

$$K_p = \frac{1}{k(T_c+\tau)} = \frac{T}{k'(T_c+\tau)}. \tag{3.62}$$

where k' is the gain and T the time constant in the 1st order process. Note that the slope is $k=\frac{k'}{T}$ in case of an integrating process. Hence we have

$$T_i = 4\zeta^2(T_c+\tau). \tag{3.63}$$

Putting $\zeta = 1$ gives real roots and a pole polynomial $\pi(s) = (1+\tau_0 s)(1+\tau_0 s) = \tau_0^2 s^2+2\tau_0 s+1$. Furthermore using the settings $K_p = \frac{1}{k(T_c+\tau)} = \frac{1}{2k\tau}$ gives the SIMC setting $T_i = 4(T_c+\tau) = 8\tau$ when $T_c = \tau$. Note also that this gives an approximate time constant $\tau_0 = \frac{1}{2\zeta}T_i = 4\tau$ for the responses.

Unfortunately, as pointed out by Haugen (2010) the response of eliminating disturbances in v is slow by this settings and the integral time constant T_i may be reduced by a factor of two, i.e. by allowing oscillations and requiring $\zeta = \frac{\sqrt{2}}{2} \approx 0.7$. This gives

$$T_i = 4\tau \tag{3.64}$$

This setting gives a Buttherworth pole polynomial $\pi(s) = \tau_0^2 s^2+\sqrt{2}\tau_0 s+1$ with $\tau_0 = 2\tau$.

3.8.2 Settings by approximating time delay as inverse response

Instead of neglecting the time delay as in the derivation of the SIMC PI settings we will in this section deduce alternative settings for the integral time constant T_i.

Let us study the disturbance response in case of a PI controller. We have

$$\frac{y}{v}(s) = \frac{h_p}{1+h_c h_p} = \frac{k\frac{e^{-\tau s}}{s}}{1+Kp\frac{1+T_i s}{T_i s}k\frac{e^{-\tau s}}{s}} = \frac{kse^{-\tau s}}{s^2+\frac{K_p k}{T_i}(1+T_i s)e^{-\tau s}} \tag{3.65}$$

Approximating the delay as an inverse response term we get

$$\frac{y}{v}(s) = \frac{ks(1-\tau s)}{s^2+\frac{K_p k}{T_i}(1+T_i s)(1-\tau s)} = \frac{T_i}{K_p}\frac{s(1-\tau s)}{\frac{T_i}{K_p k}s^2+(1+T_i s)(1-\tau s)} \tag{3.66}$$

The poles are given by the roots of the pole polynomial, i.e.,

$$\begin{aligned}\pi(s) &= \frac{T_i}{K_p k}s^2+(1+T_i s)(1-\tau s) = T_i(\frac{1}{K_p k}-\tau)s^2+(T_i-\tau)s+1 \\ &= \tau_0^2 s^2+2\tau_0\zeta s+1 \end{aligned} \tag{3.67}$$

Comparing the coefficients with the standard second order form polynomial we find

$$\tau_0^2 = \frac{T_i}{K_p k} - T_i\tau = T_i(\frac{1}{K_p k}-\tau) \tag{3.68}$$

and

$$2\tau_0\zeta = T_i - \tau \tag{3.69}$$

Hence, from $4\zeta^2\tau_0^2 = (T_i-\tau)^2$ we find the following 2nd order polynomial for the relationship between T_i and K_p as a function of the relative damping coefficient ζ, i.e.,

$$T_i^2 - (4\zeta^2(\frac{1}{K_p k}-\tau)+2\tau)T_i+\tau^2 = 0. \tag{3.70}$$

With the setting $K_p = \frac{1}{2k\tau}$ for the proportional gain we obtain $\tau_0^2 = T_i\tau$. Requiring $\zeta = 1$ gives

$$4T_i\tau = (T_i-\tau)^2 \tag{3.71}$$

and

$$T_i^2 - 6\tau T_i + \tau^2 = 0 \tag{3.72}$$

with the positive solution

$$T_i = \frac{6+\sqrt{32}}{2}\tau = (3+2\sqrt{2}) \approx 6\tau. \tag{3.73}$$

This gives very good set-point and disturbance responses.

Putting $\zeta = \frac{\sqrt{2}}{2}$ gives

$$T_i = (2+\sqrt{3}) \approx 4\tau, \tag{3.74}$$

which is approximately the same setting as in Eq. (3.64).

Example 3.3 (PI control of integrating plus delay process)
Given an integrator plus time delay system described by the transfer function

$$h_p(s) = k\frac{e^{-\tau s}}{s}, \tag{3.75}$$

with gain $k = 1$ and time delay $\tau = 1$. The results by using a PI controller with settings as in Table 3.1 are illustrated in Figure 3.8, which shows set-point and disturbance rejection responses after a unit step in the reference, $r = 1$ at time $t = 0$, and a unit step in the disturbance from $v = 0$ to $v = 1$ at time $t = 40$. The simulations is done by the MATLAB m-file function **ex3b_half.m**. *As we see the SIMC settings gives a relatively slow response from both the set-point and the disturbance. The Buttherworth setting (2) results in the fastest responses but have small oscillations. The settings derived in this note (3) results in nice, smooth response approximately as fast as the response of the Butterworth settings.*

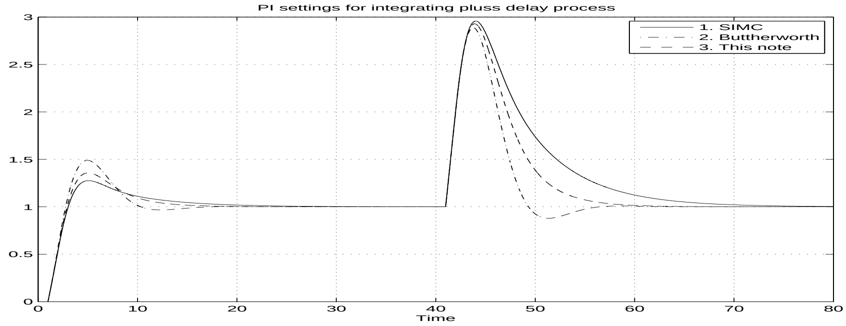

Figure 3.8: PI control of integrating pluss delay process, $h_p(s) = k\frac{e^{-\tau s}}{s}$. PI-controller $h_c(s) = K_p\frac{1+T_i s}{T_i s}$ with settings as in Table 3.1.

3.8.3 Settings by approximating time delay with Pade and Balchen approximation

Let us study the disturbance response in case of a PI controller. We have

$$\frac{y}{v}(s) = \frac{h_p}{1+h_c h_p} = \frac{k\frac{e^{-\tau s}}{s}}{1+Kp\frac{1+T_i s}{T_i s}k\frac{e^{-\tau s}}{s}} = \frac{kse^{-\tau s}}{s^2+\frac{K_p k}{T_i}(1+T_i s)e^{-\tau s}}. \tag{3.76}$$

Approximating the delay with an approximation

$$e^{-\tau s} \approx \frac{1 - \alpha s}{1 + \alpha s}, \tag{3.77}$$

where $\alpha = \frac{\tau}{2}$ gives the 1st order Pade approximation. An alternative is to use the Balchen approximation, i.e., $\alpha = \frac{2\tau}{\pi}$. This gives

$$\frac{y}{v}(s) = \frac{ks\frac{1-\alpha s}{1+\alpha s}}{s^2 + \frac{K_p k}{T_i}(1 + T_i s)\frac{1-\alpha s}{1+\alpha s}} = \frac{T_i}{K_p} \frac{s\frac{1-\alpha s}{1+\alpha s}}{\frac{T_i}{K_p k}s^2 + (1 + T_i s)\frac{1-\alpha s}{1+\alpha s}}. \tag{3.78}$$

which is equivalent with

$$\frac{y}{v}(s) = \frac{T_i}{K_p} \frac{s(1 - \alpha s)}{\frac{T_i}{K_p k}s^2(1 + \alpha s) + (1 + T_i s)(1 - \alpha s)}, \tag{3.79}$$

and

$$\frac{y}{v}(s) = \frac{T_i}{K_p} \frac{s(1 - \alpha s)}{\alpha\frac{T_i}{K_p k}s^3 + T_i(\frac{1}{K_p k} - \alpha)s^2 + (T_i - \alpha)s + 1}. \tag{3.80}$$

Hence, we have the pole polynomial

$$\pi(s) = \alpha\frac{T_i}{K_p k}s^3 + T_i(\frac{1}{K_p k} - \alpha)s^2 + (T_i - \alpha)s + 1. \tag{3.81}$$

We may now find a relationship between the controller parameters by specifying the polynomial coefficients. One choice is a Buttherworth configuration with $\zeta = \frac{\sqrt{2}}{2}$ in a prescribed 3rd order pole polynomial

$$\pi(s) = (1 + \tau_0 s)(\tau_0^2 s^2 + 2\zeta\tau_0 s + 1). \tag{3.82}$$

We will instead for the sake of increased robustness in the resulting feedback system chose $\zeta = 1$ and three multiple real time constants, i.e. a prescribed pole polynomial

$$\pi(s) = (1 + \tau_0 s)(\tau_0^2 s^2 + 2\tau_0 s + 1 = (1 + \tau_0 s)^3 = \tau_0^3 s^3 + 3\tau_0^2 + 3\tau_0 + 1. \tag{3.83}$$

Comparing the coefficients in polynomials (3.81) and (3.83) we find

$$\tau_0^3 = \alpha\frac{T_i}{K_p k}, \; 3\tau_0^2 = T_i(\frac{1}{K_p k} - \alpha), \; 3\tau_0 = T_i - \alpha. \tag{3.84}$$

Interestingly, from this, by using that $3\tau_0^3 = T_i(\frac{1}{K_p k} - \alpha)\tau_0$ we find the linear expression for T_i, i.e.,

$$3\alpha\frac{1}{K_p k} = (\frac{1}{K_p k} - \alpha)\frac{1}{3}(T_i - \alpha), \tag{3.85}$$

and the integral time constant

$$T_i = \frac{\alpha(\frac{10}{K_p k} + \alpha)}{\frac{1}{K_p k} - \alpha}. \tag{3.86}$$

Let us use the standard SIMC setting for the proportional gain, $K_p = \frac{1}{k(T_c+\tau)} = \frac{1}{2k\tau}$ for the simple choice $T_c = \tau$. We find

$$T_i = \frac{41}{6}\tau \approx 6.83\tau, \;\; \alpha = \frac{\tau}{2}. \tag{3.87}$$

This setting is approximately the same as we found by using the approximation $e^{-\tau s} \approx 1 - \tau s$. Furthermore using the Balchen approximation we find

$$T_i = \frac{20 + \frac{2}{\pi}}{\pi - 1}\tau \approx 9.64\tau, \;\; \alpha = \frac{2\tau}{\pi}. \tag{3.88}$$

This integral time setting gives better margins, i.e., a gain margin $GM \approx 3$, a phase margin $PM = 49.3$ and a maximal time delay error $d\tau_{\text{max}} \approx 1.68$. The corresponding SIMC PI settings with $T_i = 8\tau$ gives $GM \approx 2.96$, $PM = 46.86$ and $d\tau_{\text{max}} \approx 1.58$.

3.9 Re-tuning to avoid oscillations

Some feedback loops are wrong tuned so that the plant output is influenced by inherent oscillations. Suppose the PI controller parameters K_p^{old}, T_i^{old} is specified by the operator and that the feedback loop have inherent oscillations. We will in the following suggest how to re-tune the PI controller to avoid oscillations.

From eq. (3.61) we have that

$$\frac{4}{k} = \frac{K_p T_i}{\xi^2}. \tag{3.89}$$

Using eq. (3.89) for two different PI controller tunings K_p^{old}, T_i^{old} and K_p^{new}, T_i^{new} we obtain

$$K_p^{\text{new}} T_i^{\text{new}} = \frac{\xi_{\text{new}}^2}{\xi_{\text{old}}^2} K_p^{\text{old}} T_i^{\text{old}}. \tag{3.90}$$

It make sense to chose the relative damping in the re-tuned feedback loop as $\xi_{\text{new}} \geq 1$ to avoid oscillations. Hence, we have to ensure

$$K_p^{\text{new}} T_i^{\text{new}} \geq \frac{1}{\xi_{\text{old}}^2} K_p^{\text{old}} T_i^{\text{old}}. \tag{3.91}$$

3.10 Controller for special type systems

The method for standard controller design presented in this section may be used directly for simple processes.

3.10.1 Pure time delay process

Consider a pure time delay process

$$h_p(s) = k e^{-\tau s} \approx k(1 - \tau s). \tag{3.92}$$

The method leads in this case to a pure integral I controller

$$h_c(s) = \frac{1}{T_i s} \tag{3.93}$$

with $T_i = k(T_c + \tau)$.

The set-point response from r to y is specified as $\frac{y}{r} = \frac{1-\tau s}{1+T_c s}$ for the above control systems.

3.11 Examples

Example 3.4 (Effect of feedback control: SISO system)
Given a system described by the transfer function

$$h_p(s) = \frac{1-s}{1+s}. \tag{3.94}$$

This system have a zero for $s = 1$ and a pole $s = -1$. Since the pole is negative and located in the left half plane the system is stable. The zero is positive and located in the rigt half part of the complex plane. The system is therefore said to be a non-minimum phase system. Some limitations in the feedback system is therefore to be expected. Typically, there will be limitations in the set-point response from the reference, r, to the measurements output, y. Typically we can not use large values of the proportional constant, K_p.

Assume that we want to use a simple proportional feedback controller given by

$$u = K_p(r - y), \tag{3.95}$$

where r is the reference signal for y and K_p is the proportional constant. The closed loop system is therefore described by.

$$\frac{y}{r} = h_{cl}(s) = \frac{h_p(s)h_r(s)}{1-(-1)h_p(s)h_r(s)} = \frac{K_p(1-s)}{(1-K_p)s+1+K_p}, \tag{3.96}$$

where we have used negative feedback. As we see, the closed loop system also have a zero at $s = 1$. Hence, the open loop zero is not influenced by the feedback as expected. Zeroes are not influenced by feedback. However, the pole for the closed loop system is

$$s_{cl} = -\frac{1+K_p}{1-K_p}. \tag{3.97}$$

It is natural to demand that the closed loop system is stable, i.e., $s_{cl} < 0$. Hence we have that

$$-1 < K_p < 1. \tag{3.98}$$

This means that the speed of response is limited. In this example we got problems with the inverse response for large values of K_p, since the system is non-minimum *phase. The system have an inverse response since the gain at time zero ($t = 0$) is given by*

$$h_{cl}(s = \infty) = \frac{-K_p}{1 - K_p} = \!\rightarrow -\infty \quad \textit{når} \quad K_p \rightarrow 1 \tag{3.99}$$

In addition, the system have a positive steady state gain given by

$$h_{cl}(s = 0) = \frac{K_p}{1 + K_p} \tag{3.100}$$

This means that the inverse response goes to infinity as $K_p \rightarrow 1$. However, the response got faster since the pole of the closed loop system got more negative, i.e. $s_{cl} \rightarrow -\infty$ as $K_p \rightarrow 1$.

The problem is here that we can not get both fast response and a small inverse response. This is illustrated in Figure 3.9. We also se from Figure 3.10 that the amount of control signal u increases to infinity when $K_p \rightarrow 1$.

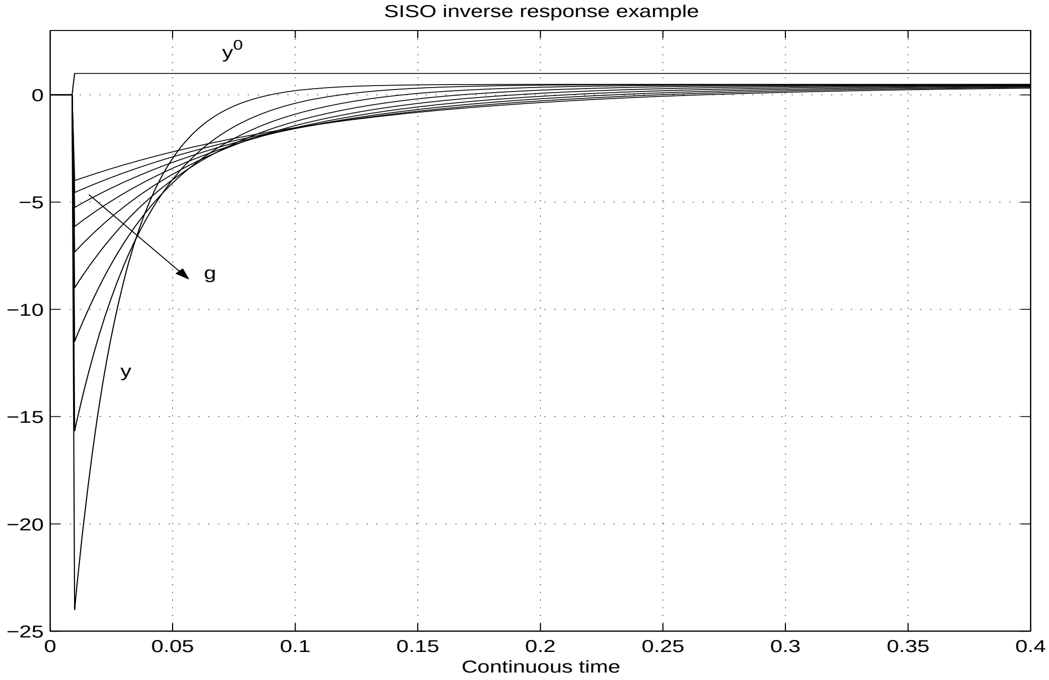

Figure 3.9: Step response simulation of a system with $h_p(s) = \frac{1-s}{1+s}$ and $u = K_p(r - y)$ for varying proportional coefficient $0.8 < K_p < 0.96$. Note that symbols $g = K_p$ and $r = y^0$ is used in the figure.

Example 3.5 (Inverse response in state space and transfer function models)

Given a system described by the state space model

$$\dot{x} = -\frac{1}{T}x + k\frac{T + \tau}{T^2}u, \tag{3.101}$$

$$y = x - \frac{k\tau}{T}u. \tag{3.102}$$

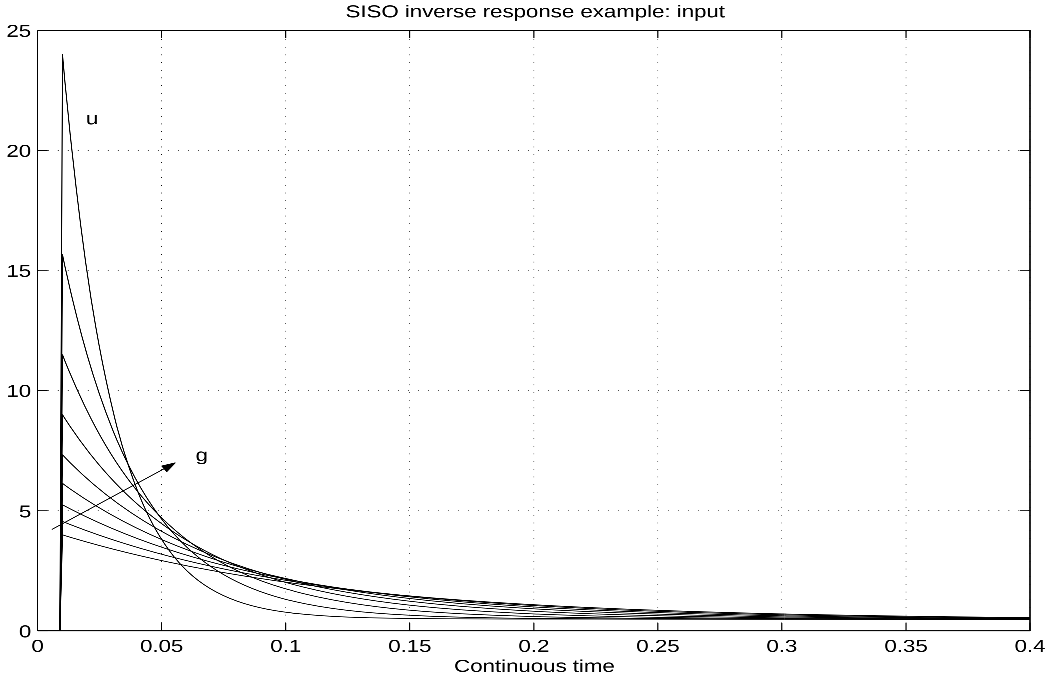

Figure 3.10: The input $u = K_p(r - y)$ after a step response simulation of a system with $h_p(s) = \frac{1-s}{1+s}$ for varying proportional coefficient $0.8 < K_p < 0.96$.Note that symbols $g = K_p$ and $r = y^0$ is used in the figure.

This model is equivalent with the following transfer function model

$$\frac{y}{u} = k\frac{1 - \tau s}{1 + Ts}. \tag{3.103}$$

This system have an inverse response because of the zero in the right half plane, i.e., the zero $s_0 = \frac{1}{\tau}$. Note also that the inverse response term $1 - \tau s$ in the numerator is an approximation to a pure time delay since $e^{-\tau s} \approx 1 - \tau s$. The transfer function model (3.103) is a good starting point for PI controller synthesis by, e.g. the Skogestad (2002) tuning method.

Example 3.6 (PI-control of a non-minimum phase SISO system)
Given a system described by the transfer function model $y = h_p(s)u$ where the transfer function is given by

$$h_p(s) = \frac{1 - 2s}{s^2 + 3s + 2} = \frac{1 - 2s}{(s + 1)(s + 2)}. \tag{3.104}$$

The system is to be controlled by a PI controller given by

$$h_c(s) = K_p\frac{1 + T_i s}{T_i s}. \tag{3.105}$$

A commonly used choice is to chose the integral time T_i equal to the dominant time constant in the system in order to cancel it and thereby simplify the loop transfer function , $h_0 = h_p h_c$. The system has two poles $s_1 = -1$ and $s_2 = -2$. This gives *the two time constants* $T_1 = -\frac{1}{s_1} = 1$ *and* $T_2 = -\frac{1}{s_2} = \frac{1}{2}$. *Hence, the loop transfer function is with* $T_i = T_1 = 1$ *given by*

$$h_0(s) = h_p h_c = \frac{1-2s}{(s+1)(s+2)} K_p \frac{1+T_i s}{T_i s} = \frac{K_p}{T_i} \frac{1-2s}{s(s+2)}, \tag{3.106}$$

where we have chosen $T_i = 1$. *We can now find expressions for* K_p *by demanding the closed loop system to be stable. The system from the reference, r, to the output, y, must be stable. The transfer function from the reference r to the output y in a feedback system (with negative feedback) is given by*

$$\frac{y}{r} = \frac{h_0}{1+h_0} = \frac{\frac{K_p}{T_i}\frac{1-2s}{s(s+2)}}{1+\frac{K_p}{T_i}\frac{1-2s}{s(s+2)}} = \frac{\frac{K_p}{T_i}(1-2s)}{s^2 + 2(1-\frac{K_p}{T_i})s + \frac{K_p}{T_i}} \tag{3.107}$$

It can be shown that the roots of a 2nd order polynomial , $s^2 + a_1 s + a_0 = 0$, *has roots in the left half plane (stable system) if the coefficients are positive, i.e., stable system if* $a_1 > 0$ *and* $a_0 > 0$. *This can be shown by study the polynomial,* $(s+\lambda_1)(s+\lambda_2) = s^2 + (\lambda_1 + \lambda_2)s + \lambda_1\lambda_2 = 0$ *which has roots* $s_1 = -\lambda_1$ *and* $s_2 = -\lambda_2$. *If the roots are in the left half plane, i.e.,* $s_1 < 0$ *and* $s_2 < 0$ *then we must have that* $\lambda_1 > 0$ *and* $\lambda_2 > 0$. *This implies that the coefficients must be positive, i.e.,* $a_0 = \lambda_1\lambda_2 > 0$ *and* $a_1 = \lambda_1 + \lambda_2 > 0$.

We obtain the following demands for K_p:

$$2(1 - \frac{K_p}{T_i}) > 0 \text{ and } \frac{K_p}{T_i} > 0. \tag{3.108}$$

This gives

$$0 < \frac{K_p}{T_i} < 1. \tag{3.109}$$

We have simulated the closed loop system for different values of K_p *after a positive and unit step change in the reference r. The results are presented in Figure 3.11. As we see, the response got more oscillations and overshot for increasing values of* $K_p < 1$. *At the same time the response got a larger inverse response starting at time* $t = 0$. *Inverse responses is typical when controlling non-minimum phase systems with zeroes in the right half plane.*

As we see it is difficult to at the same time obtain fast response, small overshot and small inverse response. the reason for this problems is the zero in the right half plane.

Some trial and errors gives the following reasonable choice

$$K_p = 0.42, \quad T_i = 1. \tag{3.110}$$

This PI controller settings gives a gain margin, $GM = 2.8$ $[dB]$, *and a phase margin* $PM = 71°$.

Example 3.7 (PI-control of non-minimum phase SISO system)
Consider a system described by the transfer function

$$h_p(s) = \frac{1-2s}{s^2+3s+2} = \frac{1-2s}{(s+1)(s+2)}. \tag{3.111}$$

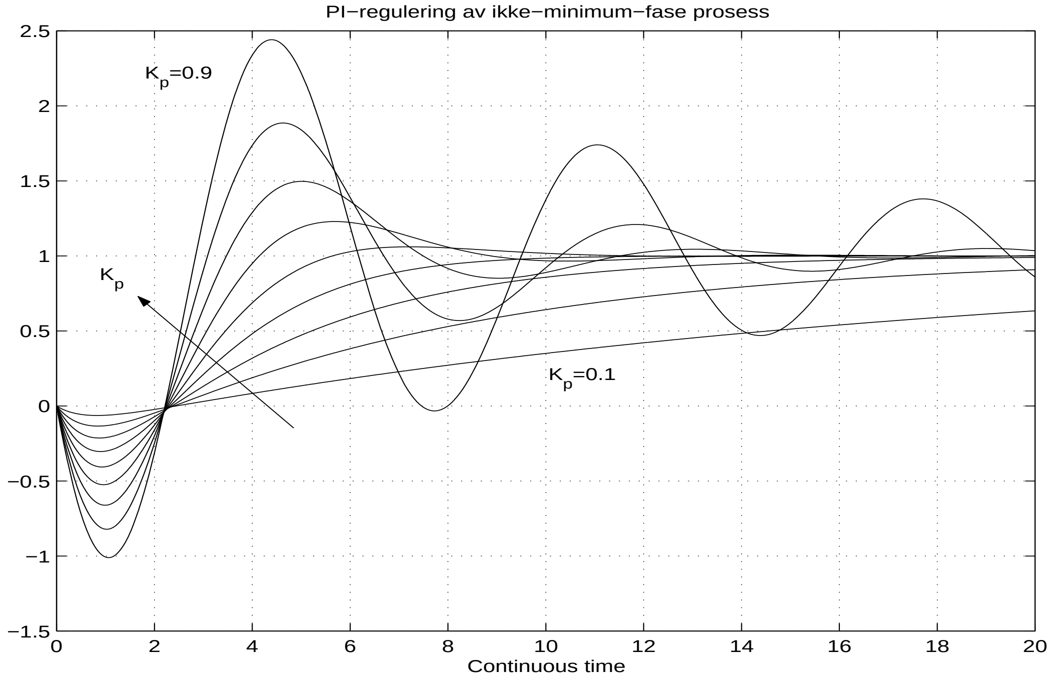

Figure 3.11: Unit step response (in the reference) simulation of a control system with process model, $h_p(s) = \frac{1-2s}{(s+1)(s+2)}$, and PI-controller $h_c(s) = K_p \frac{1+T_i s}{T_i s}$ with $T_i = 1$ and for varying proportional coefficients in the interval, $0.1 \leq K_p \leq 0.9$. The Figure is generated by the MATLAB script **siso_zero_ex.m**.

The frequency response of the system is given by

$$h_p(j\omega) = |h_p(j\omega)|e^{j\angle h_p(j\omega)}, \tag{3.112}$$

where the phase and magnitude are given by

$$\angle h_p(j\omega) = -(\arctan(2\omega) + \arctan(\omega) + \arctan(\frac{\omega}{2})), \tag{3.113}$$

$$|h_p(j\omega)| = \frac{\sqrt{1+4\omega^2}}{\sqrt{1+\omega^2}\sqrt{4+\omega^2}}. \tag{3.114}$$

The phase crossover frequency (critical frequency), ω_{180}, is then given by the frequency where the phase is -180°, i.e., $\angle h_p(j\omega_{180}) = -\pi$. The critical gain, K_{cu}, is then the gain such that $K_{cu}|h_p(j\omega_{180})| = 1$. The parameters K_{cu} and ω_{180} can for example be found by using the MATLAB function **margin**. *We obtain*

$$\omega_{180} = 1.8708, \tag{3.115}$$

$$K_{cu} = 1.5. \tag{3.116}$$

We may now simply find the parameters in a PI controller

$$h_c(s) = K_p \frac{1+T_i s}{T_i s}. \tag{3.117}$$

by using the Ziegler-Nichols method. Hence,

$$K_p = \frac{K_{cu}}{2.2} = 0.68, P_u = \frac{2\pi}{\omega_{180}} = 3.36, T_i = \frac{P_u}{1.2} = 2.79. \tag{3.118}$$

It can by simulations be shown that the response in, y, after a step in the reference, r, is relatively poor by this PI controller settings.

The closed loop behavior may be further investigated by the vlosed loop transfer function from r to y, i.e.,

$$\frac{y}{r} = \frac{h_0}{1+h_0} = \frac{\frac{K_p}{T_i}(1-2s)(1+T_i s)}{s^3 + (3-2K_p)s^2 + (K_p - 2\frac{K_p}{T_i} + 2)s + \frac{K_p}{T_i}}. \tag{3.119}$$

Example 3.8 (Inverse response and model reduction with the half rule)
Given a system described by the transfer function model $y = h_p(s)u$ where the transfer function is given by

$$h_p(s) = \frac{1-2s}{(s+1)(s+2)} = k\frac{1-\tau s}{(1+T_1 s)(1+T_2 s)}, \tag{3.120}$$

where

$$k = \frac{1}{2}, \ \tau = 2, \ T_1 = 1, \ T_2 = \frac{1}{2}. \tag{3.121}$$

A good 1st order reduced model for PI controller synthesis is

$$h_p(s) = k\frac{1-\tau s}{1+T_1 s}, \tag{3.122}$$

where $k = \frac{1}{2}$ and τ and T_1 is found by using the half rule, i.e.,

$$\tau \ := \ \tau + \frac{1}{2}T_2 = 2 + \frac{1}{4} = \frac{9}{4}, \tag{3.123}$$

$$T_1 \ := \ T_1 + \frac{1}{2}T_2 = 1 + \frac{1}{4} = \frac{5}{4}. \tag{3.124}$$

A good PI controller setting is then obtained from the Skogestad (2002) method by

$$T_i = T_1 = \frac{5}{4} \approx 1.25, \tag{3.125}$$

and

$$K_p = \frac{1}{2}\frac{T_1}{k\tau} = \frac{5}{9} \approx 0.56. \tag{3.126}$$

Simulation results of the set point responses for the various PI control tunings are shown in Figures (3.12) and (3.13).

Example 3.9 (PI control of 3rd order process)
Given a 3rd order process

$$h_p(s) = \frac{k}{(1+T_1 s)(1+T_2 s)(1+T_3 s)}, \tag{3.127}$$

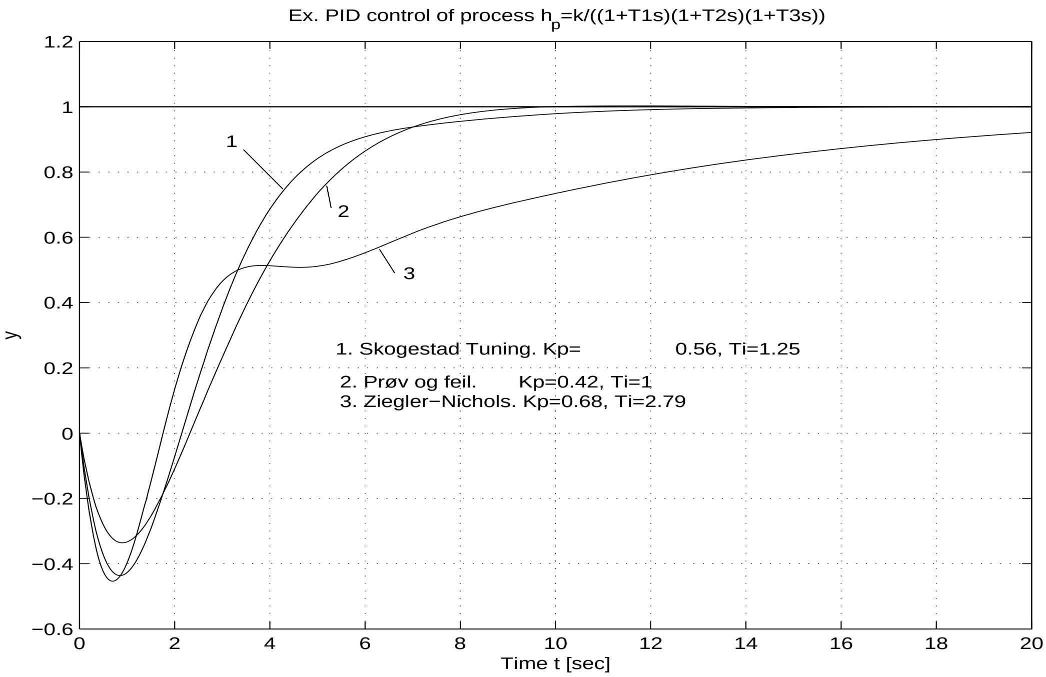

Figure 3.12: PI control of 2nd order process, $h_p(s) = 0.5\frac{1-2s}{(1+s)(1+0.5s)}$. PI-controller $h_c(s) = K_p\frac{1+T_is}{T_is}$ with parameters as in Example 3.8. The figure is generated with the MATLAB script **main_piex2.m**.

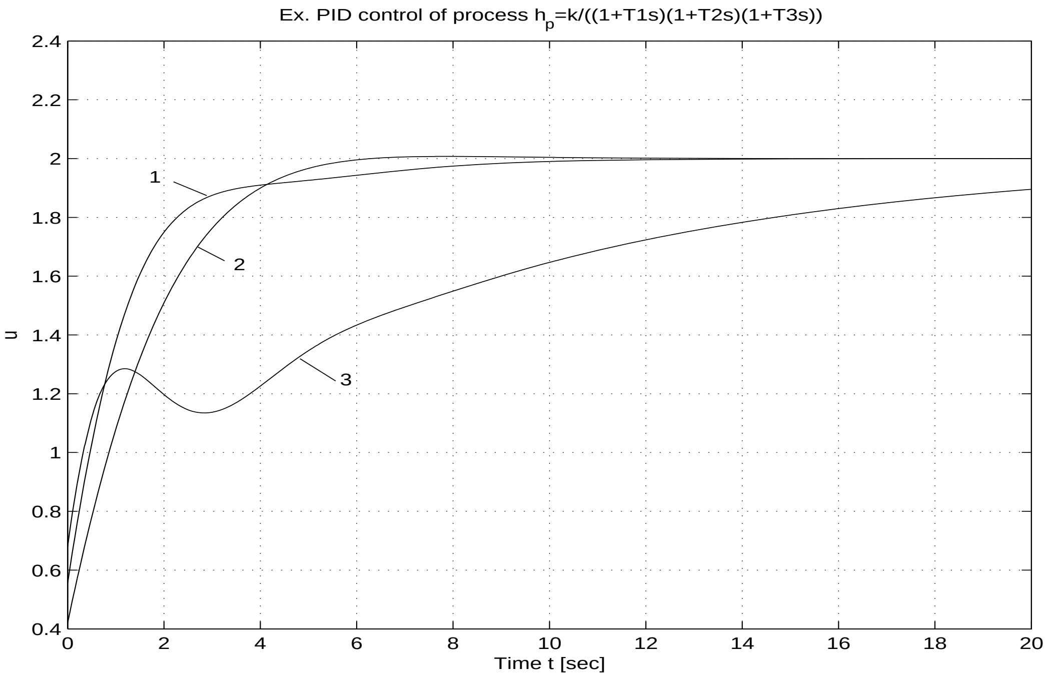

Figure 3.13: PI control of 2nd order process, $h_p(s) = 0.5\frac{1-2s}{(1+s)(1+0.5s)}$. PI-controller $h_c(s) = K_p\frac{1+T_is}{T_is}$ with parameters as in Example 3.8. The figure is generated with the MATLAB script **main_piex2.m**.

where $k = 0.5$, $T_1 = 8$, $T_2 = 5$ and $T_3 = 3$. For simulation purpose we use a state space equivalent, e.g. the following

$$\dot{x} = \begin{bmatrix} -\frac{1}{T_1} & \frac{1}{T_1} & 0 \\ 0 & -\frac{1}{T_2} & \frac{1}{T_2} \\ 0 & 0 & -\frac{1}{T_3} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{k}{T_3} \end{bmatrix} u \tag{3.128}$$

$$y = x_1 \tag{3.129}$$

Using the Skogestad method gives the following PI controller parameters

$$T_i = T = 10.5 \tag{3.130}$$

$$K_p = \frac{1}{2k} T \tau = \frac{10.5}{5.5} \approx 1.91 \tag{3.131}$$

where T and τ is the parameters in the reduced model, i.e. a first order model with inverse response

$$h_p(s) = k \frac{1 - \tau s}{1 + Ts} \tag{3.132}$$

The parameters in the reduced model is obtained from the half rule, i.e.

$$T = T_1 + \frac{1}{2} T_2 = 10.5 \tag{3.133}$$

$$\tau = \frac{1}{2} T_2 + T_3 = 5.5 \tag{3.134}$$

In order to compare the the Skogestad tuning with other PI control tuning rules we chose the PI control tunings in Table 9.3 in Balchen et al (2003). The Balchen tunings are

$$T_i = T_1 = 8 \tag{3.135}$$

$$K_p = \frac{1}{k} T_1 T_2 = 2 \frac{8}{5} = 3.2 \tag{3.136}$$

The tunings proposed in Balchen et al (2003) gives to much proportional action, both because of a small T_i and a large K_p (compared to the Skogestad tuning). The tunings proposed by Skogestad gives a more reasonable set-point response. Note that the proportional action is less because of both a larger T_i and a smaller K_p. Note that a set-point response without overshoot can be obtanied by reducing K_p, e.g. the settings $K_p = 1.3$ and $T_i = 10.5$ gives a rather nice response. The set-point responses are illustrated in Figures (3.14) and (3.15).

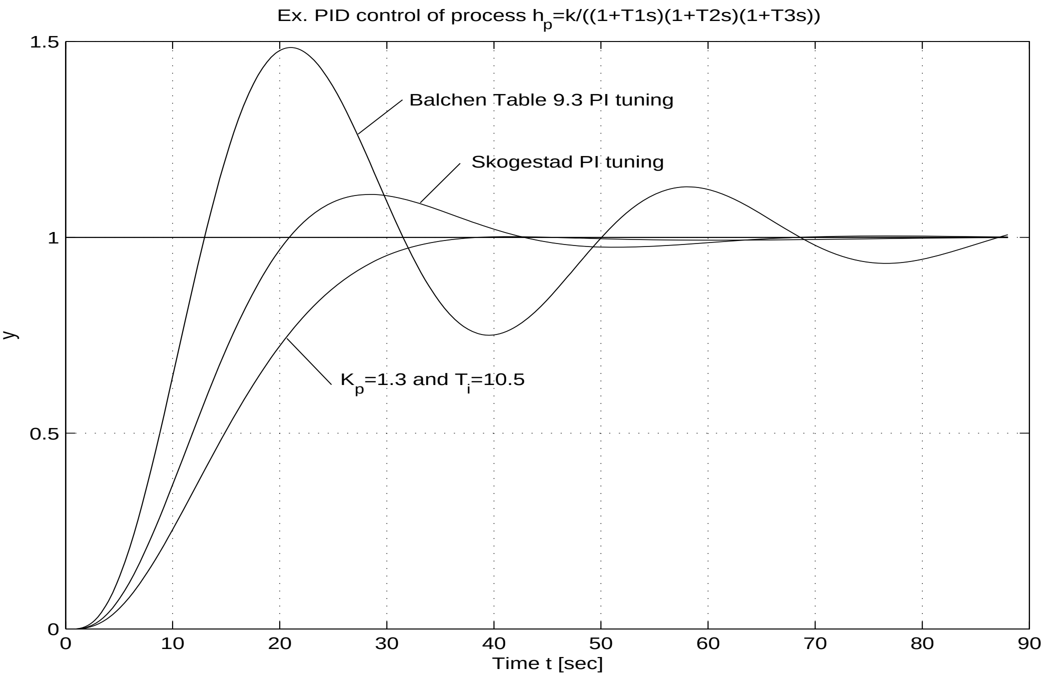

Figure 3.14: PI control of 3rd order process, $h_p(s) = \frac{k}{(1+8s)(1+5s)(1+3s)}$. PI-controller $h_c(s) = K_p\frac{1+T_is}{T_is}$ with parameters as in Example 3.9. The figure is generated with the MATLAB script **main_piex1.m**.

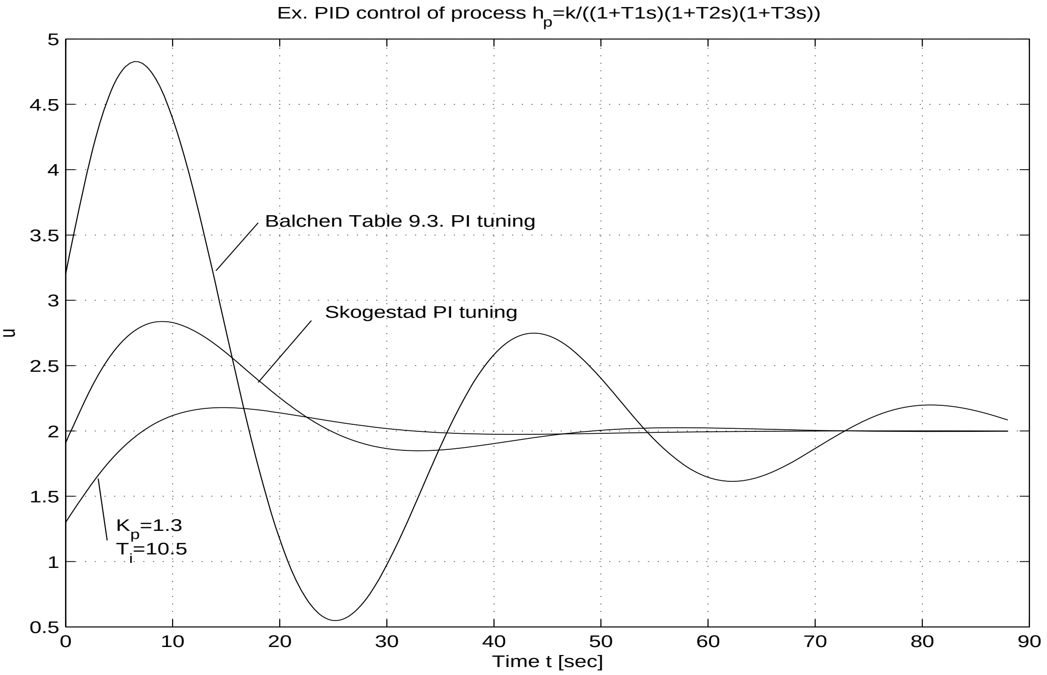

Figure 3.15: PI control of 3rd order process, $h_p(s) = \frac{k}{(1+8s)(1+5s)(1+3s)}$. PI-controller $h_c(s) = K_p\frac{1+T_is}{T_is}$ with parameters as in Example 3.9. The figure is generated with the MATLAB script **main_piex1.m**.

Chapter 4

The basic PID controller

4.1 The PI controller

4.1.1 Frequency domain description of the PI controller

The PI controller can in the Laplace plane be written as follows

$$u(s) = h_c(s)e(s), \tag{4.1}$$

where the controller input deviation, $e(s)$, is given by

$$e(s) = r(s) - y(s), \tag{4.2}$$

and the controller transfer function is given by

$$h_c(s) = K_p(1 + \frac{1}{T_i s}) = K_p \frac{1 + T_i s}{T_i s}. \tag{4.3}$$

4.1.2 Continuous Time domain description of the PI controller

In order to simulate and implementing the PI controller a state space formulation of the controller is needed. From (4.1) and (4.3) we have that

$$u(s) = \frac{K_p}{T_i}\frac{1}{s}e(s) + K_p e(s). \tag{4.4}$$

The PI controller have one internal state. And a state space model can be formulated in at least three ways. Define z as the controller state. Then we can chose the state in the following ways:

1. Choosing

$$z = \frac{K_p}{T_i s}e(s) \quad \Longrightarrow \quad u(s) = z(s) + K_p e(s), \tag{4.5}$$

gives the following continuous state space model formulation

$$\dot{z} = \frac{K_p}{T_i}e, \tag{4.6}$$

$$u = z + K_p e. \tag{4.7}$$

2. Choosing

$$z = \frac{K_p}{s} e(s) \quad \Longrightarrow \quad u(s) = \frac{1}{T_i} z(s) + K_p e(s), \tag{4.8}$$

gives the following continuous state space model formulation

$$\begin{aligned} \dot{z} &= K_p e, & (4.9) \\ u &= \frac{1}{T_i} z + K_p e. & (4.10) \end{aligned}$$

3. Choosing

$$z = \frac{1}{s} e(s) \quad \Longrightarrow \quad u(s) = \frac{K_p}{T_i} z + K_p e, \tag{4.11}$$

gives the following continuous state space model formulation

$$\begin{aligned} \dot{z} &= e, & (4.12) \\ u &= \frac{K_p}{T_i} z + K_p e. & (4.13) \end{aligned}$$

Hence, three different continuous state space formulations for the PI controller can be formulated. Alternative one given by (4.6) and (4.7) is the most common alternative since the controller state, z, has a more intuitive explanation because $z = u$ in steady state. In steady state, that is when $t \to \infty$ and the system is stable then we have that $\dot{z} = 0$ which gives $e = r - y = 0$ and hence $z = u$. The approach of assuming stability and putting the derivatives of the states equal to zero when $t \to \infty$ is a very simple way of steady state analysis of a system.

Using that the solution of a linear differential equation is as presented in Eq. (1.10) we may also formulate the PI controller by using the that the solution of the differential Eq. (4.6) is

$$z = z(t_0) + \frac{K_p}{T_i} \int_{t_0}^{t} e dt, \tag{4.14}$$

and then

$$u = K_p e + u(t_0) + \frac{K_p}{T_i} \int_{t_0}^{t} e dt, \tag{4.15}$$

since $z(t_0) = u(t_0)$.

4.1.3 Discrete Time domain description of the PI controller

In order to implement the PI controller in a digital computer we use an explicit Euler approximation of, $\dot{z}$, i.e.,

$$\dot{z} \approx \frac{z_{k+1} - z_k}{\Delta t} \tag{4.16}$$

where $\Delta t > 0$ is the sampling interval, in order to develop a discrete time state space model for the controller as follows

$$\begin{aligned} z_{k+1} &= z_k + \Delta t \frac{K_p}{T_i} e_k, & (4.17) \\ u_k &= z_k + K_p e_k. & (4.18) \end{aligned}$$

Example 4.1 (Inverse response and model reduction with the half rule)
Given a system modeled by the transfer function model $y = h_p(s)u$ where

$$h_p(s) = \frac{1-2s}{(s+1)(s+2)} \tag{4.19}$$

Simulate the system with a PI feedback system with settings $K_p = 0.56$ and $T_i = 1.25$. Write a MATLAB script for the solution.

```
% ex1_pi.m
% Purpose: Illustrate the implementation of a PI controller for
% controlling the process, h_p(s)=(1-2s)/(s+1)(s+2)
%
h_p=tf([0,-2,1],[1,3,2]);    % Process transfer function
sys=ss(h_p);                 % State space model
A=sys.a; B=sys.b; D=sys.c;

Ti=1.25; Kp=0.56;            % PI controller parameters
r=1;                         % The reference signal

Dt=0.1; t=0:Dt:20;           % Time horizon for simulation
N=length(t);

hc=Dt*Kp/Ti;             % Controller param. to save computing time
z=0; x=[0;0];            % Initial values for the states
for k=1:N
    y=D*x;               % The measurement output
    e=r-y;               % The controller input

    u=z+Kp*e;            % The controller output
    z=z+hc*e;            % Updating the controller state

    Y(k,1)=y; U(k,1)=u; % Storing variables

    x=x + Dt*(A*x+B*u); % Puting the control to the process
end

plot(t,Y), grid, xlabel('Time'), ylabel('y(t)')
title('Output response of PI controlled system')
```

Example 4.2 (PI control of a Chemical Reactor)
A chemical isothermal reactor (Van de Vusse) is studied in this example. The relationship from the feed flow rate u into the reactor to the concentration of the product y at the outlet of the reactor is modeled by the following non-linear state space model.

$$\dot{x_1} = -k_1x_1 - k_3x_1^2 + (v - x_1)u, \tag{4.20}$$

$$\dot{x_2} = k_1x_1 - k_2x_2 - x_2u, \tag{4.21}$$

$$y = x_2, \tag{4.22}$$

where the reaction rate coefficients are given by $k_1 = 50$, $k_2 = 100$, $k_3 = 10$. *The concentration of the by-product into the reactor,* v, *is treated as an unknown constant or slowly varying disturbance with nominal value* $v^s = 10$. *Choosing a steady state control* $u^s = 25$ *gives the steady states* $x_1^s = 2.5$ *and* $y^s = x_2^s = 1$.

A linearized model gives the transfer function model $y = h_p(s)u$ *where*

$$h_p(s) = K\frac{1-\tau s}{(1+Ts)^2} \tag{4.23}$$

with gain $K = \frac{2}{125}$, *inverse response time constant* $\tau = \frac{1}{250}$ *and a double time constant* $T = \frac{1}{125}$.

We want to use a PI controller so we are approximating the 2nd order model with a 1st order time delay model (using the half rule for model reduction)

$$h_p(s) = K\frac{1-\tau s}{1+Ts} \tag{4.24}$$

where the new model parameters are $T := T+\frac{1}{2}T = \frac{3}{250} \approx \frac{1}{83.3}$, *new inverse response time constant* $\tau := \tau + \frac{1}{2}T = \frac{1}{125}$ *and the gain* K *is unchanged.*

The SIMC PI controller parameters using the simple choice $T_c = \tau$ *gives* $K_p = \frac{1}{2K}\frac{T}{\tau} \approx 62.5$ *and* $T_i = T = \frac{1}{75}$.

The chemical reactor is controlled with a PI controller with these tuning parameters and implemented in the following MATLAB m-file. Se Figure 4.1 for a step response simulation.

```
% main_reacsim_pid.m
% Function to simulate PI control of a non-linear
% chemical reactor model using the explicit Euler method.
% Purpose: Supplement to Exercise 3

clear all
global u
% Nominal variables
u=25;
x=[2.5;1];

dt=0.0005;     % Sampling interval
tf=0.2;        % Final time instant
t=0:dt:tf;     % Time horizon
N=length(t);   % Number of discrete time instants

% PI controller parameters
Kp=62.5; Ti=1/125;
Kp=46.9; Ti=1/83.3; % SIMC tuning rules
Td=Ti/3;% Test

z=u;   % Initial controller state
r=1.0; % Step change in r from 1.0 to 1.02 at t=0
```

```
% Vectors to store the variables and allocate space
U=zeros(N,1); Y=zeros(N,1); X=zeros(N,2); R=zeros(N,1);

% The main simulation loop using explicit Euler method
y_old=x(2);
for k=1:N
    y=x(2);    % The measurement

    if k >= 41; r=1.02; end

    % The PI controller
    e=r-y;
    u=Kp*e+z - Kp*Td*(y-y_old)/dt; y_old=y;
    z=z+dt*Kp*e/Ti;

    U(k,1)=u;  % Save u
    Y(k,1)=y;  % Save output y
    X(k,:)=x’; % Save the states
    R(k,1)=r;  % The reference r

    fx=fx_chemreac(0,x); % Calculate the right hand side in dot(x)=f(x,u). u enters as gl
    x=x+dt*fx;           % Calculate x_(k+1) using explicit Euler
end

% Plotting results
figure(2)
subplot(311), plot(t,U), grid, title(’Control input u’)
subplot(312), plot(t,X(:,1)), grid, title(’State x_1’)
subplot(313), plot(t,Y,’-r’,t,R,’b’), grid, title(’Output y=x_2 and reference r’)
xlabel(’Continuous time 0 \leq t \leq t_f’)
% END main_reacsim_pid.m
```

4.2 The PID controller

4.2.1 Frequency domain description of the PID controller

The PID controller can be written as follows

$$u(s) = h_c(s)e(s), \tag{4.25}$$

where the controller input deviation, $e(s)$, is given by

$$e(s) = r(s) - y(s), \tag{4.26}$$

and the so called cascade (series) controller transfer function is given by

$$h_c(s) = K_p \frac{1 + T_i s}{T_i s}(1 + T_d s). \tag{4.27}$$

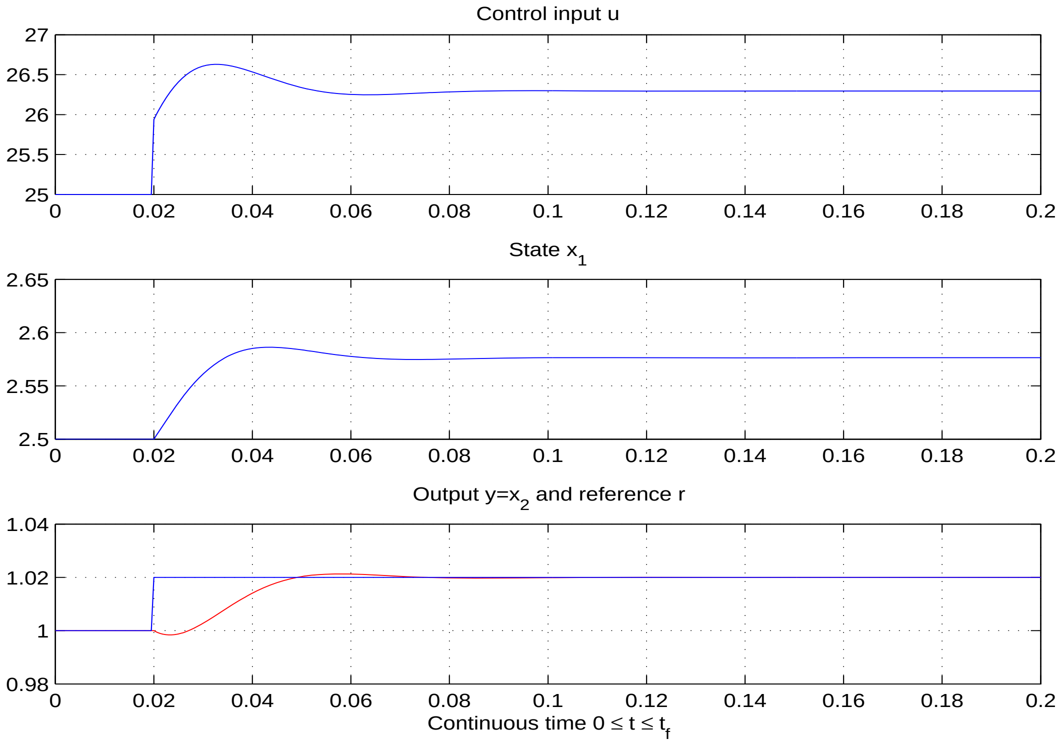

Figure 4.1: Simulated responses from the chemical reactor example in Example 4.2. The Figure is generated by the above m-file **main_reacsim_pid.m**

Note that this can be expressed as

$$h_c(s) = K_p(\frac{T_i + T_d}{T_i} + \frac{1}{T_i s} + T_d s) = K_p \frac{T_i + T_d}{T_i}(1 + \frac{1}{(T_i + T_d)s} + T_d \frac{T_i}{T_i + T_d} s)\text{(4.28)}$$

Note that the cascade formulation (4.27) is equivalent to the ideal form PID controller with transfer function

$$h_c(s) = K_p'(1 + \frac{1}{T_i' s} + T_d' s). \tag{4.29}$$

in which

$$K_p' = K_p \frac{T_i + T_d}{T_i} = K_p(1 + \frac{T_d}{T_i}) \tag{4.30}$$

$$T_i' = T_i + T_d = T_i(1 + \frac{T_d}{T_i}) \tag{4.31}$$

$$T_d' = T_d \frac{T_i}{T_i + T_d} = T_d \frac{1}{1 + \frac{T_d}{T_i}} \tag{4.32}$$

Often the derivative time, T_d, is much smaller than the integral time, T_i. Hence, in this case $K_p \approx K_p'$, $T_i \approx T_i'$ and $T_d \approx T_d'$.

4.2.2 Continuous Time domain description of the PID controller

The ideal form of the PID controller output can be written as follows

$$u(s) = K_p e(s) + \frac{K_p}{T_i s} e(s) + K_p T_d s e(s), \tag{4.33}$$

where $e(s)$ is the controller input, usually $e = r - y$. Choosing

$$z = \frac{K_p}{T_i s} e(s), \tag{4.34}$$

gives the time domain description of the PID controller

$$\dot{z} = \frac{K_p}{T_i} e \tag{4.35}$$

$$u = K_p e + z + K_p T_d \dot{e} \tag{4.36}$$

4.2.3 Discrete Time domain description of the PID controller

Similar as in the discrete implementation of the PI controller we now need an discrete approximation for, $\dot{e}$. At the present discrete time, k, only y_k and past outputs, y_{k-1}, y_{k-2}, $\cdots$ and so on is available. Hence, it is reasonable to use the following approximation

$$\dot{e} \approx \frac{e_k - e_{k-1}}{\Delta t} = \frac{r_k - y_k - (r_{k-1} - y_{k-1})}{\Delta t}, \tag{4.37}$$

where Δt is the sampling time and where we have used that $e = r - y$. In order not to get abrupt changes in the controller output, u, for step changes in the reference we usually are assuming that the reference signal is constant when implementing the PID controller, such that $r_k - r_{k-1} = 0$. Hence, we are using

$$\dot{e} \approx -\frac{y_k - y_{k-1}}{\Delta t}. \tag{4.38}$$

Hence, we have the following discrete implementation of the PID controller

$$z_{k+1} = z_k + \Delta t \frac{K_p}{T_i} e_k, \tag{4.39}$$

$$u_k = z_k + K_p e_k - K_p T_d \frac{y_k - y_{k-1}}{\Delta t} \tag{4.40}$$

For noisy systems in which the output can be modelled as $y = Dx + w$ where w is white noise (Normally gaussian distributed random numbers) a large derivative time constant, T_d, will increase the noise level on the control input, u. A rule of thumb is to chose a small T_d.

4.3 Anti windup and constraints

In practical implementations of the PI and PID controller there may be reasons for handling constraints on the control input, u. These constraints may be physical limitations of the controller organ. Hence, the practical and physical control input, $u(t)$, should lie within hard bound constraints as

$$u_{\text{min}} \leq u(t) \leq u_{\text{max}}. \tag{4.41}$$

When the controller output, $u(t)$, is within the bounds (4.41) then we are using the controller state equation

$$\begin{aligned} u &= K_p e + z + K_p T_d \dot{e} \qquad (4.42) \\ \dot{z} &= \frac{K_p}{T_i} e. \qquad (4.43) \end{aligned}$$

However, when the controller are saturated, i.e. when the controller output, u, are outside bounds, i.e. when

$$u < u_{\text{min}}, \tag{4.44}$$

then we are using

$$\begin{aligned} u &= u_{\text{min}} \qquad (4.45) \\ \dot{z} &= 0, \qquad (4.46) \end{aligned}$$

and when

$$u > u_{\text{max}}, \tag{4.47}$$

then we are using

$$\begin{aligned} u &= u_{\text{max}} \qquad (4.48) \\ \dot{z} &= 0, \qquad (4.49) \end{aligned}$$

in order not to integrate z when the controller are saturated. Note that the control input, u, is computed at all time instants by Equation (4.42) and checked if it is within bounds. This is the principle of Anti windup and it is implemented in the computer with pure logic.

Example 4.3 (PI controller implementation: anti windup and constraints)

Given the same system as in Example 4.2. In this example we have the additional constraints that

$$0 \leq u \leq 1.75 \tag{4.50}$$

A MATLAB script with anti windup and constraints implementation is presented in the following.

```
% ex1_pi_anti.m
% Purpose: Illustrate the implementation of a PI controller for
% controlling the process, h_p(s)=(1-2s)/(s+1)(s+2)
% -Implementation of Anti-Windup
%
```

```
h_p=tf([0,-2,1],[1,3,2]);   % Process transfer function
sys=ss(h_p);                % State space model
A=sys.a; B=sys.b; D=sys.c;

Ti=1.25; Kp=0.56;           % PI controller parameters
r=1;                        % The reference signal

Dt=0.1; t=0:Dt:20;          % Time horizon for simulation
N=length(t);

v=0.7; u_max=1.5; u_min=0;
z=0; x=[0;0];           % Initial values for the states
for k=1:N
    y=D*x;              % The measurement output
    e=r-y;              % The controller input

    u=z+Kp*e;           % The controller output
    Z(k,1)=z;

    if u>u_max          % Handeling constraints and Anti-windup
        u=u_max;
        z=z;
        %z=u-Kp*e;
        %z=z+Dt*Kp*e/Ti +Dt*(u_max-u)/Ti;
    elseif u<u_min;
        u=u_min;
        z=z;
    else
        z=z+Dt*Kp*e/Ti;     % Updating the controller state
    end

    Y(k,1)=y; U(k,1)=u; % Storing variables

    up=u+v;
    x=x + Dt*(A*x+B*up); % Puting the control to the process
end

U_max=u_max*ones(size(t'));
subplot(311), plot(t,Y), grid, xlabel('Time'), ylabel('y(t)')
title('Output response of PI controlled system')
subplot(312), plot(t,[U U_max]), grid, xlabel('Time'), ylabel('u(t)')
subplot(313), plot(t,Z), grid, xlabel('Time'), ylabel('z(t)')
```

4.4 Bumpless transfer

4.4.1 Bumpless transfer between manual and automatic mode

The problem or need for a bumpless transfer between manually process operations and closed loop feedback control is of practical importance. This topic will be discussed in this section. It is important to make sure that the manually setting for the control coincide with the controller output at the time of switching.

Suppose that the process is operating manually in open loop with a control input, u_{operator}, specified by the operator, i.e. the process input is

$$u = u_{\text{operator}} \tag{4.51}$$

and that at a specified time instant, $t = t_{\text{on}}$, we want to switch to closed loop feedback PID control.

$$\begin{aligned} u &= K_p e + z + K_p T_d \dot{e} && (4.52) \\ \dot{z} &= \frac{K_p}{T_i} e && (4.53) \end{aligned}$$

As we have seen, the PI and PID controllers need an initial value for the controller state, z. The change from manually to feedback control will be bumpless if the initial controller state is computed by

$$z = u_{\text{operator}} - K_p e - K_p T_d \dot{e} \tag{4.54}$$

when $t = t_{\text{on}}$ and switching to closed loop PID feedback control. At time instants $t_{\text{on}} < t$ we use the controller equations (4.58) and (4.59).

In order to switch bump-less from manually to automatic control we can use the following scheme.

if $t = t_{\text{on}}$

$$\begin{aligned} u &= u_{\text{operator}} && (4.55) \\ z &= u_{\text{operator}} - K_p e - K_p T_d \dot{e} && (4.56) \\ & && (4.57) \end{aligned}$$

else

$$\begin{aligned} u &= K_p e + z + K_p T_d \dot{e} && (4.58) \\ \dot{z} &= \frac{K_p}{T_i} e && (4.59) \end{aligned}$$

```
% ex1_pi_bump.m
% Purpose: Illustrate the implementation of a PI controller for
% controlling the process, h_p(s)=(1-2s)/(s+1)(s+2)
% -Implementation of Anti-Windup
%
```

```
h_p=tf([0,-2,1],[1,3,2]);   % Process transfer function
sys=ss(h_p);                % State space model
A=sys.a; B=sys.b; D=sys.c;

Ti=1.25; Kp=0.56;           % PI controller parameters
r=1;                        % The reference signal

Dt=0.1; t=0:Dt:20;          % Time horizon for simulation
N=length(t);
t_on=10;

v=0.5;                      % Constant disturbance
Hd=-D*inv(A)*B;             % Steady state gain
us=inv(Hd)*(r-Hd*v);        % Steady state control
xs=-inv(A)*B*(us+v);        % Steady state


u_max=1.75; u_min=0;
x=xs;                       % Initial values for the states
for k=1:N
    y=D*x;              % The measurement output
    e=r-y;              % The controller input    9

    if k <= t_on        % Manual control
        u=us;
    elseif k==t_on      % Swich to automatic for k>=t_on
        z=us-Kp*e;
        u=z+Kp*e;
    else                % Automatic control

    u=z+Kp*e;           % The controller output
    Z(k,1)=z;

    if u>u_max          % Handeling constraints and Anti-windup
        u=u_max;
        %z=z;
        z=u-Kp*e;
    elseif u<u_min;
        u=u_min;
        z=z;
    else
        z=z+Dt*Kp*e/Ti;     % Updating the controller state
    end
    end
    Y(k,1)=y; U(k,1)=u; % Storing variables

    up=u+v;
    x=x + Dt*(A*x+B*up); % Puting the control to the process
```

```
end

U_max=u_max*ones(size(t'));
subplot(311), plot(t,Y), grid, xlabel('Time'), ylabel('y(t)')
title('Output response of PI controlled system')
subplot(312), plot(t,[U U_max]), grid, xlabel('Time'), ylabel('u(t)')
subplot(313), plot(t,Z), grid, xlabel('Time'), ylabel('z(t)')
```

4.4.2 Bumpless transfer between PID parameter changes

Chapter 5

Time delay

5.1 Padé approximations to the exponential for e^{θ}

Approximations to the exponential e^{θ} are used in many circumstances. They are among others used in order to compute approximations to the transition matrix $e^{A\Delta t}$ and for discretizations of continuous time state space models. Another topic is to obtain rational approximations to the time delay $e^{-\tau s}$.

We will in this section present the theory behind what we call Padé approximations to il e^{θ}. The starting point is the following series expansion of the exponential

$$e^{\theta} = 1 + \theta + \frac{1}{2}\theta^2 + \frac{1}{6}\theta^3 + \text{hot} \tag{5.1}$$

where hot is short for “higher order terms”.

5.1.1 Developing a 1st order Padé approximation

Assume that we want to approximate e^{θ} with

$$e^{\theta} \approx \frac{1 + b_1\theta}{1 + a_1\theta} \tag{5.2}$$

where b_1 and a_1 are scalar coefficients. Equation (5.2) is referred to as a 1st order Padé approximation. Some times also the term $(1, 1)$ Padé approximation is used about this approximation. The reason for this is thet this approximation have a 1st order nominator and a 1st order denominator polynomial in θ.

We may now find the coefficients a_1 and b_1 such that the error between (5.1) and (5.2) is minimized, i.e. to minimize the error $\varepsilon = e^{\theta} - (1 + b_1\theta)/(1 + a_1\theta)$ with respect to a_1 and b_1. In order to find b_1 and a_1 we need two equations. Those equations may be found by putting the coefficients in the terms for θ and θ^2 in a series expansion for the error ε to specified values, e.g. eual to zero. The error term wil by this approach be of 3rd order, i.e. the error is proportional to θ^3.

The method we now are to present is based on that e^{θ} may be written as

$$e^{\theta} = \frac{1 + b_1\theta}{1 + a_1\theta} + c_3\theta^3 + \text{hot} \tag{5.3}$$

where hot stands for “higher order terms”. Equation (5.3) may be proved by series expansion of $(1+b_1\theta)/(1+a_1\theta)$ by using *long division* and combining this with a series expansion for e^{θ}. This gives the error

$$\varepsilon = e^{\theta} - \frac{1+b_1\theta}{1+a_1\theta} = c_1\theta + c_2\theta + c_3\theta^3 + \text{hot} \tag{5.4}$$

If we chose $c_1 = 0$ and $c_2 = 0$ in Equation (5.4) we deduce Equation (5.3). We assume that there eqists a_1 and b_1 which gives $c_1 = 0$ and $c_2 = 0$.

Putting (5.1) in the left hand side of Equation (5.3) and multiplying on both sides with $1 + a_1\theta$ gives,

$$(1+a_1\theta)(1+\theta+\frac{1}{2}\theta^2+\frac{1}{6}\theta^3+\text{hot}) = 1+b_1\theta+c_3\theta^3+a_1c_3\theta^4+\text{hot} \tag{5.5}$$

This last expression may be written as a polynomial, i.e. with terms of θ, θ^2, θ^3, and so on, Vi får

$$(1+a_1-b_1)\theta + (\frac{1}{2}+a_1)\theta^2 + (\frac{1}{6}+\frac{1}{2}a_1-c_3)\theta^3 + \text{hot} = 0 \tag{5.6}$$

where we have included the term $a_1c_3\theta^4$ in the hot term. We may now find three equations by putting the coefficients in front of θ, θ^2 and θ^3 equal to zero. This gives the three equations

$$1 + a_1 - b_1 = 0 \tag{5.7}$$

$$\frac{1}{2} + a_1 = 0 \tag{5.8}$$

$$\frac{1}{6} + \frac{1}{2}a_1 - c_3 = 0 \tag{5.9}$$

which gives

$$a_1 = -\tfrac{1}{2} \quad b_1 = \tfrac{1}{2} \quad c_3 = -\tfrac{1}{12} \tag{5.10}$$

We have now deduced the following approximation

$$e^{\theta} \approx \Phi_{1,1}(\theta) = \frac{1+\frac{1}{2}\theta}{1-\frac{1}{2}\theta} \tag{5.11}$$

The error in this approximation is given by $c_3\theta^3 = -\frac{1}{12}\theta^3$. This is usually a good approximation when $\theta < 1$.

Equation (5.11) is referred to as a $(1,1)$ (ore 1st order) Padé approximation to the exponential e^{θ} because both the nominator and the denominator to the approximation is of 1st order.

5.1.2 Alternative prof of the 1st order Padé approximation

we will in this section present an alternative development of the 1st order Padé approximation. Assume that e^{θ} are to be approximated with

$$e^{\theta} \approx \frac{1+b_1\theta}{1+a_1\theta} \tag{5.12}$$

where b_1 and a_1 are two scalar coefficients.

By using long-division ore from a formula collection book we find the series for $(1+x)^{-1}$ as

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + x^4 - \cdots \tag{5.13}$$

Note that this equation may be proved by using long-division. Using this formula we find

$$\frac{1+b_1\theta}{1+a_1\theta} = (1+b_1\theta)(1 - a_1\theta + a_1^2\theta^2 - a_1^3\theta^3 + \cdots) \tag{5.14}$$

This gives

$$\frac{1+b_1\theta}{1+a_1\theta} = 1 + (b_1 - a_1)\theta + a_1(a_1 - b_1)\theta^2 + (a_1^2(b_1 - a_1) + c_3)\theta^3 - c_3\theta^3 + \cdots \tag{5.15}$$

Note that Equation (5.15) may be proved directly by using *long-division*.

We now demand that the series for e^{θ} given by Equation (5.1) should be as equal to the series for equation (5.15) as possibile. We therefore are putting the coefficients in the terms θ and θ^2 equal to the coefficients in the corresponding coefficients in Equation (5.1). The coefficients in the θ^3 term gives an expression for the main error. Rhis gives

$$\begin{aligned} b_1 - a_1 &= 1 && (5.16)\\ a_1(a_1 - b_1) &= \frac{1}{2} && (5.17)\\ a_1^2(b_1 - a_1) + c_3 &= \frac{1}{6} && (5.18) \end{aligned}$$

Which gives $a_1 = -\frac{1}{2}$, $b_1 = 1 + a_1 = \frac{1}{2}$ and $c_3 = \frac{1}{6} - a_1^2 = -\frac{1}{12}$. Note that we have used Equation (5.15) in order to find the coefficient c_3 in the error term. The result is

$$e^{\theta} \approx \Phi_{1,1}(\theta) = \frac{1 + \frac{1}{2}\theta}{1 - \frac{1}{2}\theta} \tag{5.19}$$

which is equivalent with (5.11). The error in the approximation is given by

$$\varepsilon = e^{\theta} - \frac{1 + \frac{1}{2}\theta}{1 - \frac{1}{2}\theta} = c_3\theta^3 + \text{hot} \tag{5.20}$$

5.1.3 Developing a $(1,0)$ Padé approximation

We start the discussion by the following example of using long-division.

Example 5.1 (long-division)
We will shortly illustrate long-division of the term $1/(1+x)$. *We have*

$$\frac{1}{1+x} = \frac{1 + x - x}{1+x} \tag{5.21}$$

which is equivalent with

$$\frac{1}{1+x} = 1 - x\frac{1}{1+x} \tag{5.22}$$

By repetition use of Equation (5.22) we obtain

$$\frac{1}{1+x} = 1 - x(1 - x\frac{1}{1+x}) = 1 - x + x^2\frac{1}{1+x} \tag{5.23}$$

Putting again equation (5.22) on the right hand side gives

$$\frac{1}{1+x} = 1 - x + x^2(1 - x\frac{1}{1+x}) = 1 - x + x^2 - x^3\frac{1}{1+x} \tag{5.24}$$

Continuation of this process gives Equation (5.13).

From Example 5.1 we have that long-division of $1/(1+a\theta)$ gives

$$\frac{1}{1+a\theta} = 1 - a\theta + a^2\theta^2 - a^3\theta^3 + \text{hot} \tag{5.25}$$

We find the following expression for the error

$$\varepsilon = e^\theta - \frac{1}{1+a\theta} = (1+a)\theta + (\frac{1}{2} - a^2)\theta^2 + (\frac{1}{6} + a^3)\theta^3 + \text{hot} \tag{5.26}$$

The error is minimized by choosing $a = -1$. We therefore have the following approximation

$$e^\theta \approx \Phi_{1,0}(\theta) = \frac{1}{1-\theta} \tag{5.27}$$

where the main error is $(1/2 - a)\theta^2 = -1/2\theta^2$. $\Phi_{1,0}(\theta)$ is referred to as a $(1,0)$ Padé approximation, because the approximation have a 1st order denominator and a 0 order nominator polynomial in θ.

5.1.4 (s,t) Padé approximations

We will in this section introduce the notation $\Phi_{s,t}(\theta)$ as a short for an (s,t) Padé approximation to the exponential e^θ, i.e., a t-order nominator polynomial and a s-order denominator polynomial. Hence, in general we may express e^θ as

$$e^\theta = \Phi_{s,t}(\theta) + c_{s+t+1}\theta^{s+t+1} + \text{hot} \tag{5.28}$$

where

$$\Phi_{s,t}(\theta) = \frac{1 + b_1\theta + \cdots + b_t\theta^t}{1 + a_1\theta + \cdots + a_s\theta^s} \tag{5.29}$$

and $\Phi_{s,t}(\theta)$ is defined as a (s,t) Padé approximation to e^θ.

Similar to the developement of the $(1,0)$ and $(1,1)$ Padé approximations we may find the following (s,t) Padé approximations.

(s,t)	$\Phi_{s,t}(\theta)$	Main error
(0,1)	$1+\theta$	$\frac{1}{2}\theta^2$
(0,2)	$1+\theta+\frac{1}{2}\theta^2$	$\frac{1}{6}\theta^3$
(1,0)	$\frac{1}{1-\theta}$	$-\frac{1}{2}\theta^2$
(1,1)	$\frac{1+\frac{1}{2}\theta}{1-\frac{1}{2}\theta}$	$-\frac{1}{12}\theta^3$
(2,2)	$\frac{1+\frac{1}{2}\theta+\frac{1}{12}\theta^2}{1-\frac{1}{2}\theta+\frac{1}{12}\theta^2}$	$-\frac{1}{720}\theta^5$

Padé approximations are used in many circumstances. Of particular interest in control-theory is the problem of obtaining discrete state space models from continuous time state space models, and for simulation of dynamic systems.

Another important application of Padé approximations is to obtain rational approximations to the time delay in the frequency domain, $e^{-\tau s}$, which is a Laplacian description of a time delay. Padé approximations for $e^{-\tau s}$ will be presented in the next section.

Common methods for simulations of dynamic systems is the explicit Euler method, implicit Euler method and the trapezoid method. All those methods may be deduced from the Padé approximations. This will be illustrated in the following Example 5.2.

Example 5.2 (Discretization)
Consider a simple dynamic system

$$\dot{x} = -ax \tag{5.30}$$

An exact discrete time model is given by

$$x_{k+1} = \Phi x_k \tag{5.31}$$
$$\Phi = e^{-a\Delta t} \tag{5.32}$$

where Δt is the step length.
The explicit Euler method:
Using the explicit Euler method gives

$$\frac{x_{k+1}-x_k}{\Delta t} \approx -ax_k \tag{5.33}$$

This gives

$$x_{k+1} = \Phi_{0,1}x_k \tag{5.34}$$
$$\Phi_{0,1} = 1-a\Delta t \tag{5.35}$$

Note that the $(0,1)$ Padé approximation for $\Phi = e^{-a\Delta t}$ is identical to using the explicit Euler method for discretization.
Implicit Euler method:
Using the implicit Euler method for discretization gives

$$\frac{x_{k+1}-x_k}{\Delta t} \approx -ax_{k+1} \tag{5.36}$$

This gives

$$x_{k+1} = \Phi_{1,0}x_k \tag{5.37}$$
$$\Phi_{1,0} = (1+a\Delta t)^{-1} \tag{5.38}$$

Note that the approximation to the transition matrix we have found by using implicit Euler method is identical to a use (1, 0) Padé approximation to Φ.

Trapes metoden:

Vi benytter Trapes metoden for diskretisering og får

$$\frac{x_{k+1} - x_k}{\Delta t} \approx \frac{1}{2}(-ax_{k+1} - ax_k) \tag{5.39}$$

Dette gir

$$x_{k+1} = \Phi_{1,1} x_k \tag{5.40}$$

$$\tag{5.41}$$

$$\Phi_{1,1} = \frac{1 - \frac{\Delta t}{2} a}{1 + \frac{\Delta t}{2} a} \tag{5.42}$$

Denne approksimasjonen, dvs. Trapes metoden for diskretisering, er identisk med (1, 1) Padé approksimasjonen til Φ.

Fordelen med Trapes metoden og implisitt Eulers metode for diskretisering er at den resulterende diskrete modellen er numerisk stabil for alle $\Delta t > 0$, dvs. skrittlengden kan velges vilkårlig stor. Dette er ikke tilfelle med eksplisitt Eulers metode for diskretisering.

5.2 Padé approksimasjoner for $e^{-\tau s}$

Av spesiell interesse i reguleringsteknisk sammenheng er $\Phi_{1,1}(-\tau s)$ og $\Phi_{2,2}(-\tau s)$ approksimasjonene til transferfunksjonen $e^{-\tau s}$, som beskriver en transportforsinkelse i Laplace-planet.

$\Phi_{1,1}(-\tau s)$ approksimasjonen refereres også til som en 1.ordens Padé approksimasjon fordi approksimasjonen har 1. ordens teller og 1. ordens nevner polynom i s. En 1. ordens Padé approksimasjonen finner vi fra ligning (5.11) ved å sette $\theta = -\tau s$.

1. og 2. ordens Padé approksimasjoner er gitt ved

$$e^{-\tau s} \approx \frac{1 - \frac{1}{2}\tau s}{1 + \frac{1}{2}\tau s} \quad \text{(1. ordens Padé approksimasjon)} \tag{5.43}$$

$$e^{-\tau s} \approx \frac{1 - \frac{1}{2}\tau s + \frac{1}{12}\tau^2 s^2}{1 + \frac{1}{2}\tau s + \frac{1}{12}\tau^2 s^2} \quad \text{(2. ordens Padé approksimasjon)} \tag{5.44}$$

der s er Laplace-plan operator og τ er transportforsinkelsen.

Det kan vises at en n. ordens Padé approksimasjon for transferfunksjonen $e^{-\tau s}$ kan skrives slik.

$$e^{-\tau s} \approx \frac{1 - a_1 s + a_2 s^2 - a_3 s^3 + \cdots + (-1)^n a_n s^n}{1 - b_1 s + b_2 s^2 - b_3 s^3 + \cdots + b_n s^n} \tag{5.45}$$

5.3 Balchen approksimasjoner for $e^{-\tau s}$

I Balchen (1990) er det utledet approksimasjoner til transferfunksjonen $e^{-\tau s}$ som i mange tilfeller gir bedre resultater for bruk i forbindelse med stabilitetsanalyse sammenlignet med tilsvarende Padé approksimasjoner.

Noen Balchen approksimasjoner er som følger

$$e^{-\tau s} \approx \frac{1 - \frac{2}{\pi}\tau s}{1 + \frac{2}{\pi}\tau s} \qquad \text{(1. ordens Balchen approksimasjon)} \tag{5.46}$$

$$e^{-\tau s} \approx \frac{1 - \frac{3}{2\pi}\tau s + \frac{1}{\pi^2}\tau^2 s^2}{1 + \frac{3}{2\pi}\tau s + \frac{1}{\pi^2}\tau^2 s^2} \qquad \text{(2. ordens Balchen approksimasjon)} \tag{5.47}$$

$$e^{-\tau s} \approx \frac{1 - 0.504\tau s + 0.1013\tau^2 s^2 - 0.0108 s^3}{1 + 0.504\tau s + 0.1013\tau^2 s^2 + 0.0108 s^3} \qquad \text{(3. ordens Balchen approksimasjon)} \tag{5.48}$$

der s er Laplace-plan operator og τ er transportforsinkelsen.

Chapter 6

Feedback systems

6.1 Description of feedback systems

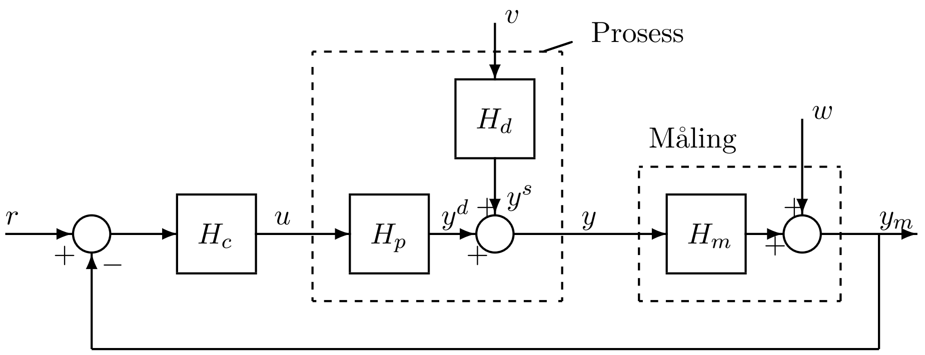

Figure 6.1: Standard feedback system. A model description of the control is included in H_p. Normally $H : m = I$ in this section. r, is the reference, u is the control, v is the disturbance, y is the process output, w is measurements noise and y_m is the process output measurements.

We are in this section going to investigate and describe the fundamental properties of a feedback system. The description is given for multiple input and output (MIMO) systems. We are normally using capital letters for transfer matrices in MIMO (multivariable) systems and lowercase letters for transfer functions in SISO (monovariable) systems.

Consider a system described by the linear continuous time system state space model

$$\dot{x} = Ax + Bu + Cv \tag{6.1}$$

$$y = Dx \tag{6.2}$$

where y is a vector of output variables, u is a vector of control variables and x is a vector of state variables. v is a vector of disturbance variables (disturbances).

However, note that v may contain variables not used as control variables. v may contain both measured and unmeasured variables.

The measurement, y_m, of the process output, y, is described by

$$y_m = y + w \tag{6.3}$$

where we for simplicity have used $H_m = I$. Se figure 6.1. w is a vector of measurements noise. This is usually a high frequence noise.

Note that in feedback systems as shown in Figure 6.1 we usually want to control y and not the measurement y_m.

From the state space model description of the process we find the following Laplace plane model description of the process

$$y = \overbrace{D(sI - A)^{-1}B}^{H_p} u + \overbrace{D(sI - A)^{-1}C}^{H_d} v \tag{6.4}$$

This model with $H_m = I$ is illustrated in Figure 6.1. Comparing the notation used in Figure 6.1 (i.e. $y = y^d + y^s$) we find that

$$y^d = H_p u \tag{6.5}$$

$$y^s = H^d v \tag{6.6}$$

$$y = y^d + y^s \tag{6.7}$$

As we see, the process output y can be divided into two parts, or influenced from two contributions. One contribution y^d from the manipulable control variables u and one contribution y^s driven from the disturbances v. y^d may be defined as the deterministic (known) contribution to y, i.e. the contribution driven from the known control variables u. y^s is denoted the stochastic (high frequency) contributions to y, i.e. contributions driven from unmeasured and measured stochastic disturbances v.

Figure 6.1 is also illustrated a controller of the form

$$u = H_c(r - y_m) = H_c(r - y - w) \tag{6.8}$$

Combining Equation (6.4) and (6.8) gives

$$\begin{aligned} y &= y^d + y^s \\ &= H_p H_c (r - y - w) + H_d v \\ &\Downarrow \\ (I + H_p H_c) y &= H_p H_c r + H_d v - H_p H_c w \end{aligned} \tag{6.9}$$

We find that the response of the closed loop system is given by

$$y = \overbrace{(I + H_p H_c)^{-1} H_p H_c}^{T} r + \overbrace{(I + H_p H_c)^{-1}}^{S} \overbrace{H_d v}^{y^s} - \overbrace{(I + H_p H_c)^{-1} H_p H_c}^{T} w \tag{6.10}$$

where we have marked the central closed loop system transfer matrices, T, and S.

The following definitions are commonly used in control theory

$$\begin{array}{lcll} H_0 & = & H_p H_c & \text{loop transfer matrix (function)} \\ S & = & (I + H_p H_c)^{-1} & \text{Sensitivity function} \\ & & & \text{(Norwegian: avviksforhold og sensitivitets-funksjon)(6.11)} \\ T & = & (I + H_p H_c)^{-1} H_p H_c & \text{complementary sensitivity function} \\ & & & \text{(Norwegian: følgeforhold)} \end{array}$$

We see that S is the closed loop transfer function (matrix) from $y^s = H_d v$ to the output y. y^s is the output from the disturbance model H_d and the contribution to the output y which is influenced from the disturbance v. y^s have a direct and additive influence on the process output y. It is natural to demand S to be small in order for the disturbance v to have a small influence on the process output.

T is the closed loop transfer function (matrix) from the reference r to the output y. For good tracking behavior it will be natural to demand $T \approx I$. Then we will have that $y \approx r$ when $S \approx 0$.

It is important to note the following relationship between S and T, i.e.,

$$S + T = I \tag{6.12}$$

This relationship, equation (6.12). can be proved as follows

$$\begin{array}{rcl} S + T & = & (I + H_0)^{-1} + (I + H_0)^{-1} H_0 \\ & \Downarrow & \\ S + T & = & (I + H_0)^{-1}(I + H_0) = I \qquad \text{QED.} \end{array}$$

where we have used the expressions for S and T as given in (6.11).

We will in the following discuss the importance of the relationship $S + T = I$. We have three signals involved in the control system, i.e., r, $y^s = H_d v$ and w, which influences upon the process output y. we have that

$$y = Tr + Sy^s - Tw \tag{6.13}$$

Ideally we want that $y = r$. This is the case if $T = I$, $S = 0$ and $w = 0$. This indicates that we have a problem if there are much measurements noise on y, i.e. when $w \neq 0$.

The relationship $S + T = I$ is important because it shows that the design problem of a feedback system is a trade off between on the one side, good tracking properties (fast response from r to y and good tracking when $T \approx I$) and on the other side, ability to reduce the influence of the disturbances w upon y which is obtained when $T \approx 0$.

From the identity $S + T = I$ we see that these demands is in conflict with each other.

The reason for this is that the measurements noise w have the same influence upon the output y as the reference r. This means that the transfer function from the measurements noise w to the process output y is given by $-T(s)$. We also have that the transfer function from the reference r to the process output y is given by $T(s)$.

Ideally we want S small in order to have a small influence upon the output y from the disturbance y^s. We also usually want T small in order for the influence from the noise w upon the output y to be small. We see from the identity (6.12) that this hardly may be fulfilled. Those demands is in conflict with each other because we have that $S + T = I$.

It is common practice to demand the feedback system to have good properties to reduce the influence from the disturbances $y^s = H_d v$ upon the output y which is the case when $S \approx 0$). We also want good tracking properties. This means that we usually want $S \approx 0$ og $T \approx I$. From the identity $S + T = I$ we see that this is possible, at least for some frequencies. However, there are at the same time difficoult to reduce the influence on y from measurements noise. In steady state, when $s = 0$, it is usually easy to obtain $S = 0$ and $T = I$. See Figure 6.2 for typically amplitude and phase behavior of H_0, S and T as a function of the frequency ω. Nothe that $s = j\omega$.

Remark 6.1 Let us have a look on the expression for the control u which influences the system. We have from Figure 6.1 that

$$u = H_c(r - y - w) \tag{6.14}$$

We also have that the output is described by

$$y = Tr + SH_d v - Tw \tag{6.15}$$

Putting this into the expression for the control gives

$$u = H_c Sr - H_c SH_d v - H_c Sw \tag{6.16}$$

where we have used that $I - T = S$ and $T - I = -S$

Remark 6.2 Note that if the system is monovariable (i.e. when y, r, v, w and u are scalar variables) then we have that

$$y = \overbrace{\frac{h_p h_c}{1 + h_p h_c}}^{T} r + \overbrace{\frac{1}{1 + h_p h_c}}^{S} h_d v - \frac{h_p h_c}{1 + h_p h_c} w \tag{6.17}$$

Note that all terms have the same nominator $1 + h_p h_c$. We define the *characteristic equation* of the system as

$$1 + h_p h_c = 1 + h_0 = 0 \tag{6.18}$$

The *characteristic equation* describes the stability properties of the system, here the stability of the closed loop system. The term $1 + h_0 = 0$ gives us a polynom in the Laplace operator s, the characteristic polynomial. If $h_0(s)$ contains transprt delay terms of the type $e^{-\tau s}$ then we can use approximations to the exponential. The roots of this characteristic polynomial is the poles of the closed loop system. The closed loop poles should lie in the left part of the complex plane in order for the closed loop system to be stable.

This indicates that stability is a property of the system and not influenced of the external signals r, v and w. The reason for this is that there only is the process model h_p and the controller h_c which influences the characteristic polynomial and thereby the stability.

Remark 6.3 Note the following alternative expressions for the complimentary sensitivity function given in (6.11). Equation (6.11) is rewritten for the sake of completeness.

$$\begin{aligned} T &= (I + H_pH_c)^{-1}H_pH_c & (6.19)\\ T &= H_pH_c(I + H_pH_c)^{-1} & (6.20)\\ T &= H_c(I + H_cH_p)^{-1}H_p & (6.21) \end{aligned}$$

Equation (6.20) can be proved as follows: from Figure 6.1 we have)with $v = 0$, $w = 0$ and $H_m = I$)

$$\begin{gathered} e = r - y = r - H_pH_ce \\ \Downarrow \\ e = (I + H_pH_c)^{-1}r \end{gathered} \tag{6.22}$$

Putting this into the expression $y = H_pH_ce$ gives

$$y = \overbrace{H_pH_c(I + H_pH_c)^{-1}}^{T}r \tag{6.23}$$

Equation (6.21) can be proved as follows: From Figure 6.1 we have that (with $v = 0$, $w = 0$ and $H_m = I$)

$$\begin{gathered} u = H_ce = H_c(r - y) = H_cr - H_cH_pu \\ \Downarrow \\ u = (I + H_cH_p)^{-1}H_cr \end{gathered} \tag{6.24}$$

Putting this into the expression $y = H_pu$ we obtain

$$y = \overbrace{H_c(I + H_cH_p)^{-1}H_p}^{T}r \tag{6.25}$$

Remark 6.4 As shown above, the output y of a linear system can be divided into two contributions. This can also be shown from the state space model given in (6.1) and (6.2).

$$\begin{gathered} \overbrace{\begin{aligned} \dot{x}^d &= Ax^d + Bu \\ y^d &= Dx^d \end{aligned}}^{\text{deterministisk part}} \qquad \overbrace{\begin{aligned} \dot{x}^s &= Ax^s + Cv \\ y^s &= Dx^s \end{aligned}}^{\text{stochastisk part}} \\ x = x^d + x^s \\ y = y^d + y^s \end{gathered} \tag{6.26}$$

This follows from the properties of linear systems.

Remark 6.5 In some circumstances symbols $N(s)$ for $S(s)$ is used and $M(s)$ for $T(s)$ is used.

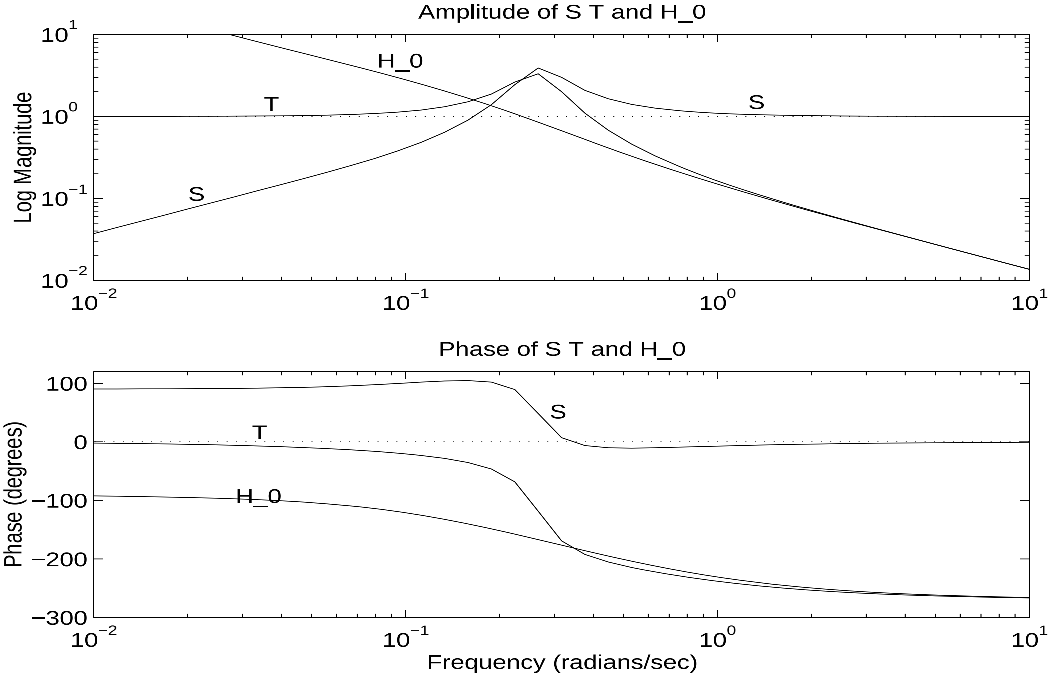

Figure 6.2: The figure shows typical amplitude and phase frequency plot for S, T og H_0. The frequency response is obtained by putting $s = j\omega$ in the transfer function. The process model used here is $h_p(s) = 3(1-2s)/(5s+1)(10s+1)$, with PI-controller $h_c(s) = K_p(1+T_is)/(T_is)$ with $T_i = 12.7$ and $K_p = 1.136$ (Ziegler-Nichols parameters). The feedback system has negative feedback.

6.2 Reasons for using feedback

There may be different reasons for using feedback control and some particular important reasons for using feedback is as listed in the following items:

1. Unknown disturbances.
2. Modeling errors. In case of a perfect model, feed-forward control is an intuitive option.
3. Unstable processes/systems.

Unknown (and not measured) disturbances are commonly in the process industry. Such process disturbances may be slowly varying trends and drifts, but more high frequency types of disturbances which are filtered through the process may also be present. Such process disturbances may be viewed as a stochastic disturbance y_s on the process output y (se Figure 6.1). This contribution is additive for linear systems.

Slowly varying process disturbances may e.g. be caused from different properties of raw products, changes in the process environments as temperature and pressure etc.

The influence of unknown process disturbances may be a great problem. Disturbances may lead to saturations in the manipulable control variables u. Even with feedback and in case of unknown slowly varying disturbances, it may happen that the best one can achieve is that the process output y are varying around the reference r. This may be satisfactory because some times it is the mean one earn money from, i.e. a satisfactory mean quality. However, great variations in quality should be avoided.

Usually we only have approximate process models h_p and these models are never perfect. With perfect models $y = h_p u$ we can use feed-forward control and notice that $y = r$ gives an ideal feed-forward controller $u = \frac{1}{h_p} r$. Such an ideal feed-forward controller is unrealistic due to modeling errors and the solution is to use feedback control (or a combination of feedback and feed-forward control).

Feedback is a grate tool to stabilize unstable processes and systems and some times such feedback solutions are remarkably simple and effective.

Chapter 7

Direct synthesis and design of standard controllers

7.1 On the PID controller formulations

An ideal PID controller may be written as follows

$$h_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{7.1}$$

where K_p is the proportional constant, T_i is the integral time and T_d is the derivative time.

Consider now a controller (in cascade form) of the form

$$h_c(s) = k\frac{(1 + T_1 s)(1 + T_2 s)}{T_3 s} \tag{7.2}$$

We will now show that the cascade form controller (7.2) may be written as an ideal PID controller. From Equation (7.2) we have

$$\begin{aligned} h_c(s) &= k\frac{1 + (T_1 + T_2)s + T_1 T_2 s^2}{T_3 s} \\ &= k(\frac{T_1 + T_2}{T_3} + \frac{1}{T_3 s} + \frac{T_1 T_2}{T_3} s) \\ &= k\frac{T_1 + T_2}{T_3}(1 + \frac{1}{(T_1 + T_2)s} + \frac{T_1 T_2}{T_1 + T_2} s) \end{aligned} \tag{7.3}$$

Comparing Equation (7.3) with the ideal form PID controller (7.1) shows that the two formulations are equivalent if

$$K_p = k\frac{T_1 + T_2}{T_3} \tag{7.4}$$

$$T_i = T_1 + T_2 \tag{7.5}$$

$$T_d = \frac{T_1 T_2}{T_1 + T_2} \tag{7.6}$$

This means that a cascade controller given by Equation (7.2) is identical to an ideal PID controller as in Equation (7.1) if the PID controller parameters K_p, T_i and T_d are chosen as in (7.4)-(7.6).

We will now show that under certain circumstances there are possible to find the controller parameters T_1, T_2 and T_3 in the cascade formulation as a function of the controller parameters K_p, T_i and T_d in the ideal PID controller.

From (7.4) and (7.5) we have directly that

$$T_3 = \frac{k}{K_p} T_i \tag{7.7}$$

From (7.4) we have

$$T_2 = T_i - T_1 \tag{7.8}$$

Putting this expression for T_2 into (7.6) gives

$$T_d = \frac{T_1(T_i - T_1)}{T_i} \tag{7.9}$$

This last expression may be written as a 2nd order equation for T_1. We have that

$$T_1^2 - T_i T_1 + T_i T_d = 0 \tag{7.10}$$

This equation have two solutions. It seams reasonable to chose the solution with the largest value in order also to avoid an negative value on T_1. Hence,

$$T_1 = \frac{T_i + \sqrt{T_i(T_i - 4T_d)}}{2} \tag{7.11}$$

In order to obtain a real solution we must have that $T_i - 4T_d \geq 0$.

Remark 7.1 (Ideal and cascade PID controllers) *Hence, the conclusion is then that an ideal PID controller given by Equation (7.1) may be written as a cascade PID controller as in Equation (7.2) if $T_i \geq 4T_d$. Hence, there are an exact relationship between the ideal PID (7.1) and the cacscade PID (7.2) when $T_i \geq 4T_d$.*

A useful approximation may be used when $T_i \gg T_d$. In this case we see from Equation (7.11) that $T_1 \approx T_i$. From (7.7) we find that $T_3 = T_i$ if $k = K_p$. From (7.6) and (7.5) we have that $T_d = \frac{T_1 T_2}{T_i} \approx T_2$ when $T_1 \approx T_i$. This means that we have the following useful remark regarding this approximation.

Remark 7.2 (Approximation between the ideal and cascade PID controllers) *The following approximation between the cascade and ideal PID controllers*

$$h_c(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \approx K_p \frac{(1 + T_i s)(1 + T_d s)}{T_i s} \tag{7.12}$$

may be used when

$$T_i \gg T_d \tag{7.13}$$

Example 7.1 (PID controller relationship)
Given an ideal PID controller as in Equation(7.1). Assume controller settings $T_i = 10$ *and* $T_d = 1$. K_p *is not specified. We want to find an alternatice cascade PID controller as in Equation (7.2).*

Using the formulas given in Equations (7.11) and (7.8) we find that

$$\begin{aligned} T_1 &= \frac{10 + \sqrt{10(10-4)}}{2} \approx 8.873 & (7.14)\\ T_2 &= 10 - 8.873 = 1.127 & (7.15)\end{aligned}$$

From equation (7.7) we find

$$T_3 = T_i = 10 \tag{7.16}$$

where we have put $k = K_p$. *We then have the following exact relationship between the ideal PID controller and the cascade PID controller.*

$$h_c(s) = K_p(1 + \frac{1}{10s} + s) = K_p\frac{(1 + 8.873s)(1 + 1.127s)}{10s} \tag{7.17}$$

7.2 Controlling a static (steady state) process

If we want the output, y, of a steady state process to be close to a specified reference signal, r, then we may simply use a feedback control strategy with an integral (I) controller in the feedback loop. Real world processes are not static but they may under certain circumstances approximated as a steady state (static) process. In practice, often a PI controller is used instead of an ideal Integral controller.

Some processes may be viewed and modeled as a steady state (static) process relative to the dynamics introduced by the feedback system. This may be the case for systems with small time constants, i.e. very fast systems, e.g. for systems with time constants smaller than the controller sample time.

An example is a Thermo Mechanical Pulping (TMP) process in the paper industry. The TMP process is used in order to produce paper pulp for making paper, from small wood chips. A TMP process is often controlled such that the consistency of the pulp obtaine a specified reference value, because this may give a specified quality of the paper. The for instance, u, be the dilution water flow rate to the refiner and let y be the measured consistency in the blow-line out of the TMP refiner. The consistency, y, is a measure of the dryness of the pulp paper mass and a reasonable value for the consistency is about 50%. The dynamics in the TMP refiner from u to the output y may be neglected compared compared to the sampling time in the control system. The dominant time constant in the process is in believed to be smaller than one second. On tye other hand the sampling time in the digital control system is about $\Delta T = 0.50$ $[s]$ to $\Delta T = 1.0$ $[s]$. Hence, the process may be viewed as a static system compared to the dynamics in the feedback system.

Another example is when designing local controllers for the control actuator itself. An example is a flow controller where u is the position of the valve and y is the mass flow out of the valve. In this case it may be reasonable to model y as a static function of u, i.e. $y = g(u)$ where $g(\cdot)$ is a function describing the behavior of the valve.

Example 7.2 (Integral (I) control of a static process)
We want in this example to investigate the behavior of a closed loop system of a static process controlled with an I-controller.

We have the controller and the process transfer functions

$$\begin{array}{lcll} h_c(s) & = & \frac{1}{T_i s} & \textit{I controller} \\ h_p & = & k & \textit{static process} \end{array}$$

The transfer function from the reference r to the output y is given by

$$\frac{y}{r}(s) = \frac{h_p h_c(s)}{1 + h_p h_c(s)} = \frac{k\frac{1}{T_i s}}{1 + k\frac{1}{T_i s}} = \frac{k}{T_i s + k} \tag{7.18}$$

This may be written as

$$\frac{y}{r}(s) = \frac{1}{1 + T_c s} \tag{7.19}$$

where T_c is given by

$$T_c = \frac{T_i}{k} \tag{7.20}$$

T_c is the time constant of the closed loop system after an change in the reference, and T_c may be chosen in order to obtain T_i. Note that T_c in general is different from the integral time T_i in the controller. In steady state we have that $\frac{y}{r}(s = 0) = 1$. The integrator in the controller ensures that we have zero steady state error, i.e. $y = r$ when $t \to \infty$.

We also see that we obtain ideal control when $T_c = 0$ and that this may be approximated with a small value on the integral time constant T_i.

Example 7.3 (PI control of a static process)
We want in this example investigate the closed loop properties of a static system controlled with a feedback PI controller. We have

$$\begin{array}{lcll} h_c(s) & = & K_p\frac{1+T_i s}{T_i s} & \textit{PI controller} \\ h_p & = & k & \textit{static process model} \end{array}$$

The transfer function from the reference, r, to the output,y, is given by

$$\frac{y}{r}(s) = \frac{h_p h_c(s)}{1 + h_p h_c(s)} = \frac{kK_p\frac{1+T_i s}{T_i s}}{1 + kK_p\frac{1+T_i s}{T_i s}} = \frac{kK_p(1 + T_i s)}{T_i s + kK_p(1 + T_i s)} \tag{7.21}$$

and

$$\frac{y}{r}(s) = \frac{kK_p(1 + T_i s)}{T_i(1 + kK_p)s + kK_p} = \frac{1 + T_i s}{1 + \frac{T_i(1+kK_p)}{kK_p}s} \tag{7.22}$$

This may be written as follows

$$\frac{y}{r}(s) = \frac{1+T_i s}{1+T_c s} \tag{7.23}$$

where T_c is given by

$$T_c = T_i\frac{1+kK_p}{kK_p} = T_i(1+\frac{1}{kK_p}) \tag{7.24}$$

T_c is the time constant of the closed loop system and note that T_c is different from T_i.

At low frequencies $s = 0$ we have in steady state that $\frac{y}{r}(s = 0) = 1$. As we see, we obtain ideal control when $\mid kK_p \mid\gg 1$, i.e., when $\mid \frac{1}{kK_p} \mid\approx 0$. Then we have that $T_c \approx T_i$ and $y/r \approx 1$. This may be achieved by using a large proportional constant K_p. We also note that the closed loop system is stable for $kK_p > -1$ fordi $T_i > 0$.

We may now specify the time constant T_c of the closed loop system and then obtain the PI controller settings parameters K_p and T_i from the above equations. This solution strategy is not unique. We may in addition for instance also specify T_i in order to define K_p. This indicates that there is not necessarily with two parameters in the controller. The reason for this is that this static process may be reasonably controlled by a simple I-controller with only one controller parameter T_i.

At the end you should also note the following state space model formulation of the closed loop system. From the theory of canonical forms (ore inverse Laplace transformations) we find that

$$\dot{x} = -\frac{1}{T_c}x + \frac{1}{T_c}r \tag{7.25}$$

$$y = \frac{T_c - T_i}{T_c}x + \frac{T_i}{T_c}r \tag{7.26}$$

As we see we have a direct proportional feed-through influence from the reference, r, to the output, y. Hence, it works like a feed-forward control action from the reference upon the output y.

7.3 Control of a non-minimum phase process

We are referring to Section 9.6 for a description of non-minimum phase systems. A common property of controlling non-minimum phase systems with standard controllers as P, PI ore PID controllers there exist an upper value for the proportional constant K_p in order to ensure stability of the closed loop system. We will in this section illustrate this by examples.

Example 7.4 (P control of a non-minimum phase system)
We want to analyse the closed loop system of a non-minimum phase system controlled by a P-controller. We have

$$\begin{aligned} h_c(s) &= K_p && \textit{P controller} \\ h_p(s) &= k\tfrac{1-T_1 s}{1+T_2 s} && \textit{non-minimum phase process} \end{aligned}$$

where $T_1 > 0$ is an inverse response time constant and $T_2 > 0$ is the time constant in the process. The transfer function from the reference r to the output y is given by

$$\frac{y}{r}(s) = \frac{h_p h_c(s)}{1 + h_p h_c(s)} = \frac{K_p k \frac{1-T_1 s}{1+T_2 s}}{1 + K_p k \frac{1-T_1 s}{1+T_2 s}} \tag{7.27}$$

This gives

$$\frac{y}{r}(s) = \frac{K_p k}{1 + K_p k} \frac{1 - T_1 s}{1 + T_c s} \tag{7.28}$$

where the time constant of the closed loop system is given by

$$T_c = \frac{T_2 - K_p k T_1}{1 + K_p k} \tag{7.29}$$

We see that the closed loop system have a pole $s = -1/T_c$. In order to ensure stability of the closed loop system we may demand the pole to be negative. The closed loop system is therefore stable if $T_c > 0$. If we assume that the denominator in the expression for T_c, i.e., $1 + K_p k > 0$ (which gives a lower limit $-1/k < K_p$) we obtain the following demand for ensuring stability.

$$T_c > 0 \quad \Rightarrow \quad -\tfrac{1}{k} < K_p < K_{cu} \tag{7.30}$$

where the upper critical value for K_p is given by

$$K_{cu} = \frac{T_2}{T_1} \frac{1}{k} \tag{7.31}$$

Hence, if T_1 is large then we must chose a small value on K_p, and if T_1 is small then K_p may be chosen larger. A limit solution is obtained when $T_1 \to 0$, hence

$$\lim_{T_1 \to 0} K_{cu} = \infty \tag{7.32}$$

Another limit conclusion is obtained by letting $T_1 \to \infty$, hence.,

$$\lim_{T_1 \to \infty} K_{cu} = 0 \tag{7.33}$$

The zero in the right half complex plane given by $s = 1/T_1$, gives constraints for the choice of K_p. We may conclude and state that the stability properties of the closed loop system is influenced strongly by the process zero in the right half plane. This is usually always the case when controlling non-minimum systems with standard PID feedback controllers.

Example 7.5 (PI control of a non-minimum phase system)
We want in this example investigate the behavior of a standard feedback system, with a non-minimum phase system controlled by a PI controller. We have

$$\begin{aligned} h_c(s) &= K_p \tfrac{1+T_i s}{T_i s} && \textit{PI controller} \\ h_p(s) &= k \tfrac{1-T_1 s}{1+T_2 s} && \textit{non-minimum phase process} \end{aligned}$$

where $T_1 > 0$ and $T_2 > 0$. The transfer function from the reference, r, to the output, y, is given by,

$$\frac{y}{r}(s) = \frac{h_p h_c(s)}{1 + h_p h_c(s)} = \frac{K_p \frac{1+T_i s}{T_i s} k \frac{1-T_1 s}{1+T_2 s}}{1 + K_p \frac{1+T_i s}{T_i s} k \frac{1-T_1 s}{1+T_2 s}} \tag{7.34}$$

We will now for simplicity chose $T_i = T_2$. This gives

$$\frac{y}{r}(s) = \frac{k K_p (1 - T_1 s)}{(T_2 - k K_p T_1) s + k K_p} = \frac{1 - T_1 s}{1 + T_c s} \tag{7.35}$$

where the time constant of the closed loop system is given by

$$T_c = \frac{T_2 - K_p k T_1}{K_p k} \tag{7.36}$$

Let us assume that $K_p k > 0$.

If $k > 0$ then this means that a lower limit for the proportional constant is $0 < K_p$. This only means that K_p should be positive.

We have the following important demand for the upper limit for K_p in order to ensure stability of the closed loop system.

$$T_c > 0 \Rightarrow 0 < K_p < K_{cu} \tag{7.37}$$

where the upper limit for K_p is given by

$$K_{cu} = \frac{T_2}{T_1} \frac{1}{k} \tag{7.38}$$

As we see, if T_1 is large then we must specify K_p small, and if T_1 is small then we may chose K_p large. A limiting case when $T_1 \to 0$ is given by

$$\lim_{T_1 \to 0} K_{cu} = \infty \tag{7.39}$$

Another limiting case when $T_1 \to \infty$ is given by

$$\lim_{T_1 \to \infty} K_{cu} = 0 \tag{7.40}$$

Hence, the zero in the right half plane, $s = 1/T_1$, gives limitations to the size of the proportional constant K_p. The stability properties of the closed loop controlled system is limited by the positive zero in the rhp. Note that another choice than putting $T_i = T_2$ may have been used in the above discussion.

7.4 Controlling of lead-lag systems

Example 7.6 (PI control of lead-lag process)
We want in this example to design a PI controller of a lead-lag type process. We have

$$\begin{aligned} h_c(s) &= K_p \tfrac{1+T_i s}{T_i s} && \textit{PI controller} \\ h_p(s) &= k \tfrac{1+T_1 s}{1+T_2 s} && \textit{lead-lag process} \end{aligned}$$

The transfer function from the reference r to the output, y, is given by the complimentary sensitivity function $T(s)$.

$$T(s) = \frac{h_p h_c(s)}{1 + h_p h_c(s)} = \frac{K_p \frac{1+T_i s}{T_i s} k \frac{1+T_1 s}{1+T_2 s}}{1 + K_p \frac{1+T_i s}{T_i s} k \frac{1+T_1 s}{1+T_2 s}} \tag{7.41}$$

This gives

$$\frac{y(s)}{r(s)} = T(s) = \frac{T_i T_1 s^2 + (T_1 + T_2)s + 1}{T_c^2 s^2 + 2T_c \xi s + 1} \tag{7.42}$$

where the parameters T_c and ξ are defined from.

$$T_c^2 = \frac{T_i(T_2 + K_p k T_1)}{K_p k}, \quad 2T_c \xi = \frac{T_i + K_p k(T_i + T_1)}{K_p k}. \tag{7.43}$$

If we now are specifying values for the time response parameters T_c and ξ then we will obtain two equations for finding the PI controller parameters K_p and T_i.

ξ is the relative damping of the closed loop system. If $\xi < 1$ then the system is under damped and the poles will be complex conjugate. If $\xi < 1$ then we will obtain oscillations in the response from r to y. $T_c = 1/\omega_n$ where ω_n is referred to as the undamped resonance frequency. We are finally referring to tables which shows the relationship between Laplace functions and time domain descriptions for an exact time domain equivalent to $T(s)$.

Chapter 8

Feed forward control

8.1 Introduction

Feed-forward control is maybe the most intuitive way of controlling a process. Consider a process to be controlled and that the process is given by the model

$$y = h_p(s)u \tag{8.1}$$

and that we want the output, y, to follow a specified reference, r. We will in this case obtain perfect control, i.e. $y = r$ by the feed-forward controller

$$u = \frac{1}{h_p(s)}r \tag{8.2}$$

if the model, $h_p(s)$, is an exact description of the controller. This feed-forward controller, which consists of the inverse of the process model, is usually not useful in practice due to modeling errors. Hence, we need additional feedback in order to obtain $y = r$ in steady state.

Feed-forward control may also be effective from measured ore estimated disturbances, v. Hence, the main variables to use in feed-forward control is the following:

- Feed-forward control from disturbances, v.
- Feed-forward control from references, r.

Usually a feed-forward controller involves the inversion of the process model $h_p(s)$. However, when the inverse of the process model does not exists, due to time delay or inverse responses in the process and the model, a model approximation which have an inverse may be used. For instance a model of lead-lag type is often used.

Feed-forward control from known references, r, is also denoted tracking control problems. Feed-forward control from references is also involved in a Model Predictive Control (MPC) solution, as well as in Linear Quadratic (LQ) optimal tracking problems.

8.2 Feedback with feed forward from the reference

let the system be described by the model

$$y = H_p u + H_d v \tag{8.3}$$

Assume that vi in addition to normal feedback from the output want a feed forward controller from the reference. A control input to the process can in this case be generated by

$$u = H_c(r - y) + u_f \tag{8.4}$$

where H_c is the controller transfer function (ore transfer matrix). The signal u_f is the feed forward which is a function of the reference signal. The feed forward control is specified to be

$$u_f = H_f^r r \tag{8.5}$$

where H_f^r is the transfer function/matrix from the reference to the feed forward signal.

Putting the control given by (8.4) into (8.3) gives

$$y = H_p H_c(r - y) + H_p u_f + H_d v \tag{8.6}$$

Ideally we want $y = r$. Putting this into equation (8.6) gives the following expression for the feed forward signal, u_f.

$$\begin{gathered} r = H_p u_f + H_d v \\ \Downarrow \\ u_f = H_p^{-1} r - H_p^{-1} H_d v \end{gathered} \tag{8.7}$$

where we have assumed that the transfer function model is invertible, i.e. we have assumed that H_p^{-1} exists. The ideal feed forward controller from the reference is therefore expressed by

$$H_f^r = H_p^{-1}. \tag{8.8}$$

where we have assumed that the process model is non-singular, i.e. H_p^{-1} exists.

As we see, we have also derived the ideal feed forward control from the disturbance, v, i.e., $H_f^v = H_p^{-1} H_d$. Another assumption for using it is that the disturbance is measured and known, ore estimated. We will in the following only analyze the feed forward control

$$u_f = H_p^{-1} r \tag{8.9}$$

Note that even if the process model $H_p(s)$ can be inverted theoretically, it is not sure that the inverse H_p^{-1} is a rational and implementable function. However, one can often use a realizable approximation to the inverse H_p^{-1}. This will be further studied lather.

The closed loop system with feed forward control is described by

$$y = (I + H_p H_c)^{-1}(H_p H_c + I)r + (I + H_p H_c)^{-1} H_d v \tag{8.10}$$

As we see, one may have perfect control with feed forward control from the reference because $y = r$ also without feedback control. This can be seen by putting $H_c = 0$ in (8.10).

If we have a perfect and non singular model model H_p and the inverse H_p^{-1} is rational and implementable, then we will have perfect control, i.e. $y = r$. (assuming also that the control is not saturated). This is only theoretically because we always have modeling errors in practice, often also the inverse H_p^{-1} is not rational. However, approximations may often give good and reasonable improvements over classical feed forward control.

For a SISO system we have a similar to Equation (8.10)

$$y = \frac{h_p h_c + 1}{1 + h_p h_c} r + \frac{h_d}{1 + h_p h_c} v \tag{8.11}$$

The stability of the feedback system with the feed-forward controller is given by the *characteristic equation*. The characteristic equation is found by putting the denominator in Equation (8.11) equal to zero. Hence, we have

$$1 + h_p h_c = 0 \tag{8.12}$$

For a system with feedback only we have the following transfer function from the reference, r, to the output, y, i.e.,

$$y = \frac{h_p h_c}{1 + h_p h_c} r + \frac{h_d}{1 + h_p h_c} v \tag{8.13}$$

As we see, a feedback system with feed-forward control, have the same characteristic equation as a feedback system without feed-forward control. This means that the feed-forward signal does not influence upon the stability properties of the system. This may also be shown to hold for systems with feed-forward from the disturbance, v.

Example 8.1
Given a stable process described by

$$h_p(s) = \frac{k}{1 + Ts} \tag{8.14}$$

The ideal feed-forward from the reference is then given by

$$h_f^r(s) = \frac{1}{k}(1 + Ts) \tag{8.15}$$

This solution is impractical due to the derivative term which will amplify high-frequent measurement of process noise. A more realistic and practical solution is to instead use the approximation

$$h_f^r(s) = \frac{1}{k}\frac{1 + Ts}{1 + T_f s} \tag{8.16}$$

where T_f may be locked upon as a filter time constant. This approximation to the ideal feed-forward control is known as a lead-lag feed-forward controller.

8.3 Feed-forward from the disturbance

8.3.1 Design based on a state space model

Consider a system described by a linear time invariant state space model

$$\dot{x} = Ax + Bu + Cv \tag{8.17}$$

$$y = Dx \tag{8.18}$$

We will in the following discussion assume that v is a measured process disturbance vector. We also assume that the control signal,u, is of the form

$$u = G(r - y) + u_f \tag{8.19}$$

$$u_f = G_f v \tag{8.20}$$

where G is a constant feedback matrix (P type controller) and that $u_f = G_f v$ is the feed-forward part. G_f is a constant feed-forward matrix.

Putting the control signal, u, into the state space model gives

$$\dot{x} = Ax + BG(r - y) + Bu_f + Cv \tag{8.21}$$

A simple method of designing the ideal feed-forward signal $u_f = G_f v$ is tho chose u_f such that the two last terms in (8.21) becomes equal to zero. Hence, we have

$$\begin{gathered} Bu_f + Cv = 0 \\ \Downarrow \\ u_f = -B^{-1}Cv \end{gathered} \tag{8.22}$$

where we have assumed that B is a non-singular matrix. The ideal feed-forward controller is then given by

$$u_f = G_f v, \quad G_f = -B^{-1}C. \tag{8.23}$$

This controller will cancel the influence of the disturbances upon the system.

Note also thet the feed-forward part of the controller does not influence upon the stability properties of the feedback system. This may be shown as follows Putting the feed-forward controller (8.23) into the state equation for the closed system(8.21) gives

$$\dot{x} = (A - BGD)x + BGr, \tag{8.24}$$

$$y = Dx. \tag{8.25}$$

As we see, it is only the feedback matrix, G, which influences upon the stability of the system.

Remark 8.1 The static feed-forward control (8.23) may off-course also be used if the feedback part of the controller is dynamic instead of static. A linearized controller with dynamic feedback part and feed-forward control may be described by the state space model

$$\dot{z} = A_u z + B_u(r - y), \tag{8.26}$$

$$u = D_u z + E_u(r - y) + u_f, \tag{8.27}$$

where A_u, B_u, D_u and E_u are controller matrices. z is the state vector in the controller. As we see, the control, u, is defined in terms of two input signals, the control deviation $e = r - y$ and the feed-forward signal $u_f = G_f v$. A PI controller may be written on this form.

8.4 Ratio control

So called ratio control is a special type of feed-forward control which is widely used in for instant the process industry. An example is for example a chemical reactor where it is important to feed the reactor with two substances, say substance A and substance B, in a specified ratio so that the feed to the reactor is blended with the correct ratio.

In ratio control the objective is for instance to hold the ratio of two variables close to or at a specific value. Consider as an example a process with two mass flow variables, q_A, and q_B, and that we want to hold the ratio

$$k = \frac{q_B}{q_A} \tag{8.28}$$

constant or close to constant also for varying q_A and q_B. Hence, the problem is to hold the ratio, k, constant instead of the two individual variables q_A and q_B. In industry, usually flow variables and flow controllers are involved when using ratio control. Hence we want $q_A = kq_B$ where k is the a constant ratio.

There are some common solutions to the problem of ratio control. We will discuss two solutions to ratio control:

Method 1. One standard solution to ratio control problem is the use one flow controller, for say flow q_B, and letting the set-point to this flow controller be taken as $r_B = kq_A$, i.e., as a factor k of a measurement of flow q_A. The entire ratio control system is then influenced by manipulating the valve for the q_A flow manually. This solution is illustrated in Figure 8.1.

Method 2. A safer solution Method 1 above is to modify the method by using two flow controllers where the entire ratio control system is influenced from one set-point, r_A, to a flow controller for flow q_A, and the set-point for the flow controller of flow q_B is taken similar as in Method 1 above, i.e. $r_B = kq_A$.

The reader should with advantage sketch block diagrams for the above methods for ratio control.

Example 8.2 (Feed-forward control with steady state models) *Consider a binary distillation column for the separation of two products A and B. The steady state mass balances are*

$$\begin{aligned} F &= B + D, & (8.29) \\ Fx_F &= Bx_B + Dx_D. & (8.30) \end{aligned}$$

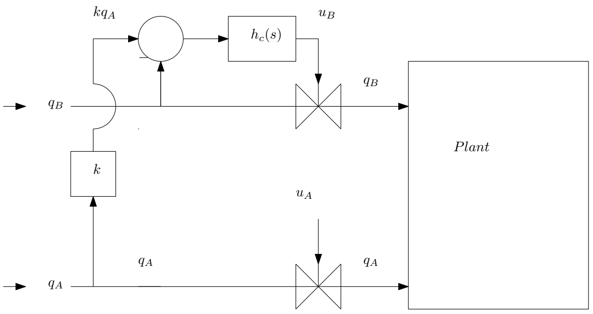

Figure 8.1: Simple ratio control structure.

where the feed flow rate F and the fraction of product A in the feed flow rate is x_F are considered as disturbances (ore load variables). The flow rate of distillate is D and the composition of product A in D is x_D, similarly, the flow rate of the bottom product is B and the composition of product A in B is x_B.

Consider the case in which all variables are measured. Then we may, e.g., solve for the distillate flow rate, D, as

$$D = F\frac{x_F - x_D}{x_D - x_B}. \tag{8.31}$$

Hence, Equation (8.31) may be used as a feed-forward controller from measured disturbances F and x_F. Note also that this feed-forward control strategy is nonlinear due to the product of F and x_F. Here the top and bottom compositions x_D and x_B, respectively, are measured output variables.

Chapter 9

Frequency domain analysis and controller synthesis

9.1 Complex numbers

The imaginary number j is defined as $j = \sqrt{-1}$ and $j^2 = -1$. Numbers of the type $c = a + jb$, where a and b are arbitrarily real numbers are called complex numbers. Moreover, a is called the real part, and b the imaginary part of the complex number c, respectively. This is defined formally as $a = \mathrm{Re}(c)$ and $b = \mathrm{Im}(c)$.

A complex number, c, can be written in so called polar form as follows

$$c = a + jb =\mid c \mid e^{j\phi}, \tag{9.1}$$

where the magnitude, $\mid c \mid$, (or length, modulus) to the complex number c is defined as

$$\mid c \mid = \sqrt{a^2 + b^2}, \tag{9.2}$$

and the phase angle ϕ is given by

$$\phi = \arctan(\frac{b}{a}). \tag{9.3}$$

The Euler formula is important and useful in connection with complex numbers, and is defined as follows

$$e^{j\phi} = \cos(\phi) + j\sin(\phi). \tag{9.4}$$

Note also that the ratio between two complex numbers, $c = a + jb$, and $f = d + je$, can be written on polar form as follows

$$\frac{c}{f} = \frac{a + jb}{d + je} = \frac{\sqrt{a^2 + b^2}e^{j\arctan(\frac{b}{a})}}{\sqrt{d^2 + e^2}e^{j\arctan(\frac{e}{d})}} = \frac{\sqrt{a^2 + b^2}}{\sqrt{d^2 + e^2}}e^{j(\arctan(\frac{b}{a}) - \arctan(\frac{e}{d}))}. \tag{9.5}$$

Similarly, the product of two complex numbers $c = a + jb$, and $f = d + je$, can be written on polar form as follows

$$\begin{aligned} cf = (a + jb)(d + je) &= \sqrt{a^2 + b^2}e^{j\arctan(\frac{b}{a})}\sqrt{d^2 + e^2}e^{j\arctan(\frac{e}{d})} \\ &= \sqrt{a^2 + b^2}\sqrt{d^2 + e^2}e^{j(\arctan(\frac{b}{a}) + \arctan(\frac{e}{d}))}. \end{aligned} \tag{9.6}$$

Also the following rules regarding calculus of exponentials are use-full. Given two scalars ϕ_1 and ϕ_2. Then

$$e^{\phi_1}e^{\phi_2} = e^{\phi_1+\phi_2}, \tag{9.7}$$

$$\frac{e^{\phi_1}}{e^{\phi_2}} = e^{\phi_1-\phi_2}. \tag{9.8}$$

9.2 Frequency response

We start this section by defining the frequency response of a system.

Definition 9.1 (Frequency response)
$h(j\omega)$ is defined as the frequency response of a system $h(s)$. By putting $s = j\omega$ in the transfer function $h(s)$ of a system, we obtain the frequency response $h(j\omega)$, where $0 \leq \omega \leq \infty$ $[\frac{rad}{s}]$ is the frequency.

The frequency response of a system describes how the output of a system behaves after a sinusoid signal with frequency ω $[rad/s]$ is feed into the input of the system. Assume that a sinusoid signal

$$u(t) = u_0 \sin(\omega t + \alpha) \tag{9.9}$$

is feed into the system. Here u_0 is the magnitude of the input signal, ω $[\frac{rad}{s}]$ is a specified frequency and α is a constant.

It can be shown that if this sinusoid input signal is exciting the system for long time, i.e. to all transients has died out and that there is steady state behavior ($t \to \infty$), then the output response of the system will be described by

$$y(t) = y_0 \sin(\omega t + \beta) \tag{9.10}$$

where y_0 is the magnitude and β is a constant describing the phase displacement from the input signal. It can be shown that the phase displacement is given by

$$\phi = \beta - \alpha \qquad [rad]. \tag{9.11}$$

It can be shown that the ratio y_0/u_0 between the magnitude of the output, y_0, and the magnitude of input signal, u_0, can be defined directly from the frequency response $h(\jmath\omega)$ of the system, i.e.,

$$\begin{array}{lcll} \frac{y_0}{u_0} & = & \mid h(j\omega) \mid & \text{(Magnitude. No: amplitudeforholdet)} \\ \phi & = & \angle h(j\omega) & \text{(Phase angle. No: fasevinkelen)} \end{array} \tag{9.12}$$

From the theory of complex numbers we can show that the frequency response $h(j\omega)$ can be written as

$$h(j\omega) =\mid h(j\omega) \mid e^{j\angle h(j\omega)} \tag{9.13}$$

which is the polar form of a complex number.

Note that, $\frac{y_0}{u_0} =\mid h(j\omega) \mid$, is the system gain at frequency ω, i.e. the system gain if the system is excited with a sinusoid signal with frequency ω). In particular, at low frequencies, i.e. when ($s = j\omega = 0$ or $\omega = 0$, we have that the steady state gain is given by $\frac{y_0}{u_0} =\mid h(j0) \mid$.

As we see, the magnitude $\mid h(j\omega) \mid$ and the phase angle (or phase shift) $\angle h(j\omega)$ depend on the frequency ω. This dependency, i.e. the magnitude $\mid h(j\omega)$ and the phase shift $\angle h(j\omega)$, may be plotted in so called Bode plots.

The most important results is summed up in the following Theorem.

Teorem 9.2.1 (Frequency response)
Given a linear system described by

$$y = h(s)u \tag{9.14}$$

where $h(s)$ is the transfer function of the system. If we feed a time varying sinusoid signal at the input of the system, i.e. an input given by

$$u(t) = u_0 \cdot \sin(\omega t + \alpha) \tag{9.15}$$

where u_0 is a constant input amplitude, ω $[rad/s]$ is a specified frequency of oscillation and α $[rad]$ is a constant phase displacement. The stationary output response is then given by

$$y(t) =\mid h(j\omega) \mid u_0 \cdot \sin(\omega t + \alpha + \phi) \tag{9.16}$$

where

$$\phi = \angle h(j\omega) \tag{9.17}$$

The magnitude $\mid h(j\omega) \mid$ and the phase shift (phase angle) $\angle h(j\omega)$ is defined from the frequency response of the system as follows

$$h(j\omega) =\mid h(j\omega) \mid e^{j\angle h(j\omega)} \tag{9.18}$$

Example 9.1 (Frequency response)
Given a process $y = h(s)u$ where

$$h(s) = \frac{-s+4}{s+8} = 0.5\frac{1-\frac{1}{4}s}{1+\frac{1}{8}s} \tag{9.19}$$

The frequency response of the system is defined as

$$h(j\omega) = 0.5\frac{1-j\frac{\omega}{4}}{1+j\frac{\omega}{8}} =\mid h(j\omega) \mid e^{j\angle h(j\omega)} \tag{9.20}$$

where

$$\mid h(j\omega) \mid= 0.5\frac{\sqrt{1+(\frac{\omega}{4})^2}}{\sqrt{1+(\frac{\omega}{8})^2}} \tag{9.21}$$

$$\angle h(j\omega) = \arctan(-\frac{\omega}{4}) - \arctan(\frac{\omega}{8}) \tag{9.22}$$

Assume that the system input, u, is excited with a signal

$$u(t) = \sin(\omega t) \tag{9.23}$$

From the frequency response theory we have that the stationary output response for the frequency $\omega = 0.3$ $[rad/s]$ is given by

$$\omega = 0.3 \quad \rightarrow \quad y(t) = 0.501 \sin(0.3t - 6.44^\circ \frac{\pi}{180^\circ}) \tag{9.24}$$

where the magnitude and the phase shift is given by

$$\mid h(j0.3) \mid = 0.501, \quad \angle h(j0.3) = -6.44^\circ. \tag{9.25}$$

Similarly, the stationary output response for a higher frequency $\omega = 10$ $[rad/s]$ is given by

$$\omega = 10 \quad \rightarrow \quad y(t) = 0.841 \sin(10t - 119.54^\circ \frac{\pi}{180^\circ}) \tag{9.26}$$

where the magnitude and the phase shift is given by

$$\mid h(j10) \mid = 0.841, \quad \angle h(j0.3) = -119.54^\circ. \tag{9.27}$$

This is illustrated in Figures 9.1 and 9.2.

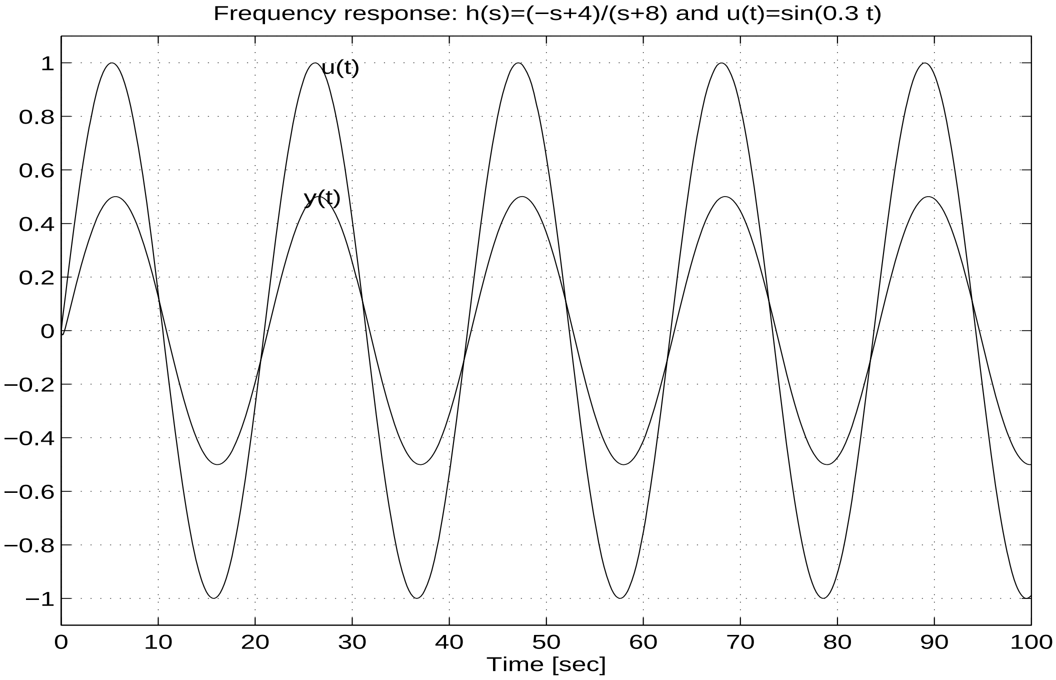

Figure 9.1: The Figure shows the output response $y(t)$ for a system $h(s) = (s - 4)/(s + 8)$ excited with a sinusoid input signal $u(t) = \sin(\omega t)$ where the frequency is $\omega = 0.3$ $[rad/s]$. Se Example 9.1 for details.

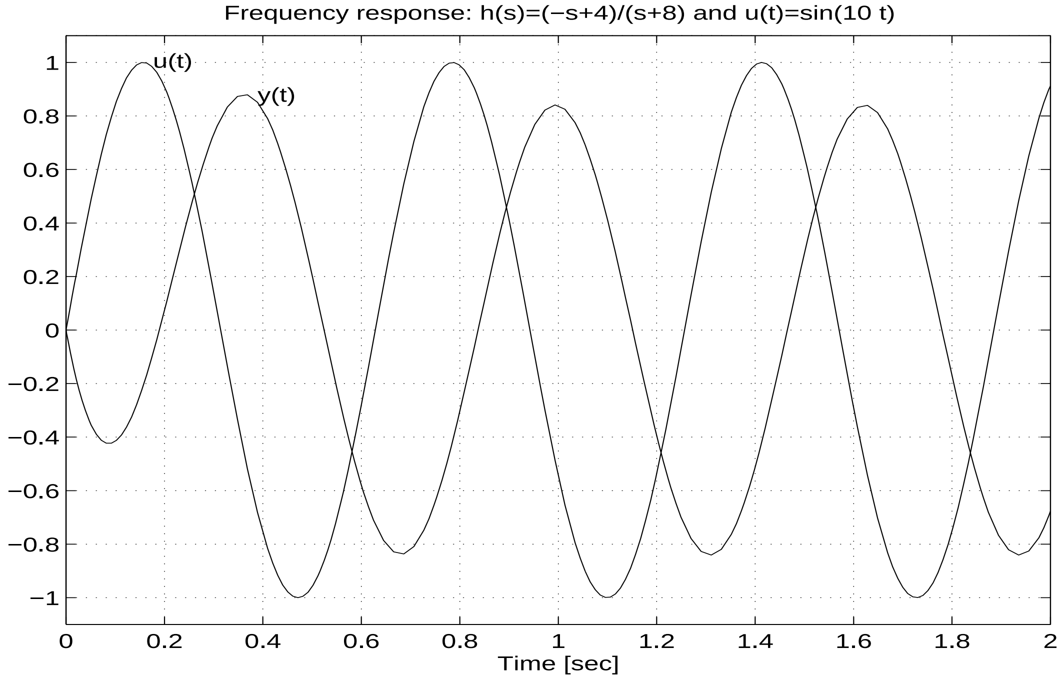

Figure 9.2: The Figure shows the output response $y(t)$ for a system $h(s) = (s - 4)/(s+8)$ excited with a sinusoid input signal $u(t) = \sin(\omega t)$ where the frequency is $\omega = 10$ $[rad/s]$. Se Example 9.1 for details.

9.3 Gain margin and Phase margin

The following definitions are important in the theory of frequency analysis and design of feedback control systems

- Gain margin, GM (No: forsterkningsmargin)
- Phase Margin, PM (No: fasemargin)
- Phase crossover frequency, ω_{180} (No: fase kryssfrekvens)
- Gain crossover frequency, ω_c (No: forsterknings kryssfrekvens)

The starting point for defining the gain margin and phase margin is the loop transfer function $H_0(s) = H_0(j\omega)$. We are only locking for the case when s takes the values along the imaginary axis, i.e. we are putting $s = j\omega$.

Gain margin

The Gain margin GM is defined as follows

$$GM \overset{\text{def}}{=} \frac{1}{\mid H_0(j\omega_{180}) \mid} \qquad \text{Gain Margin} \tag{9.28}$$

where ω_{180} is the phase crossover frequency, i.e. the frequency where the phase angle of the loop transfer function becomes $-180°$ the first time.

Phase crossover frequency

ω_{180} is defined as the *phase crossover frequency*. The definition of ω_{180} is as follows

$$\angle H_0(j\omega_{180}) = -180^\circ \tag{9.29}$$

The phase crossover frequency ω_{180} is formally computed as follows

$$\omega_{180} = \mathrm{sol}_\omega[\angle H_0(j\omega) + 180^\circ = 0] \tag{9.30}$$

where $\mathrm{sol}_\omega[f(\omega) = 0]$ means solving the function $f(\omega) = 0$ with respect to ω.

Phase Margin

The Phase Margin PM is defined as

$$PM \overset{\text{def}}{=} \angle H_0(j\omega_c) + 180^\circ \qquad \text{Phase Margin} \tag{9.31}$$

where ω_c is the gain crossover frequency, i.e. the frequency where the magnitude of the loop transfer function is equal to one, i.e. $\mid H_0(j\omega_c) \mid = 1$.

Gain crossover frequency

the gain crossover frequency, ω_c, is defined as follows

$$\mid H_0(j\omega_c) \mid = 1 \tag{9.32}$$

the gain crossover frequency, ω_c, is formally computed as follows

$$\omega_c = \mathrm{sol}_\omega[\mid H_0(j\omega) \mid -1 = 0] \tag{9.33}$$

where $\mathrm{sol}_\omega[f(\omega) = 0]$ means the solution of the function $f(\omega) = 0$ with respect to ω.

The gain Margin, GM, is a measure of how much the system gain can be increased at the phase crossover frequency, $\omega = \omega_{180}$, before the system becomes unstable. Such an increase of the system gain can be the result of an increasing proportional constant, K_p, in a PID controller or an uncertain process gain.

The Gain Margin, GM, is often presented in decibel $[DB]$ units. For a given absolute value of the Gain Margin we get it in decibel by the conversion

$$GM \ [DB] = 20 \log(GM) \tag{9.34}$$

For example, a Gain Margin of 2 corresponds to $GM = 6.02 \ [DB]$.

The Phase Margin, PM, is the extra negative phase angle which can be tolerated before the system becomes unstable. The Phase Margin, PM is the extra phase which can be added to $\angle H_0(j\omega_c)$ before the phase becomes -180° and the system becomes unstable.

Typical demands for a feedback control system is a Gain Margin $GM > 2$ (which is equivalent to $GM > 20\log(2) = 6.02 \ dB$) and a Phase Margin $PM > 30^\circ - 45^\circ$.

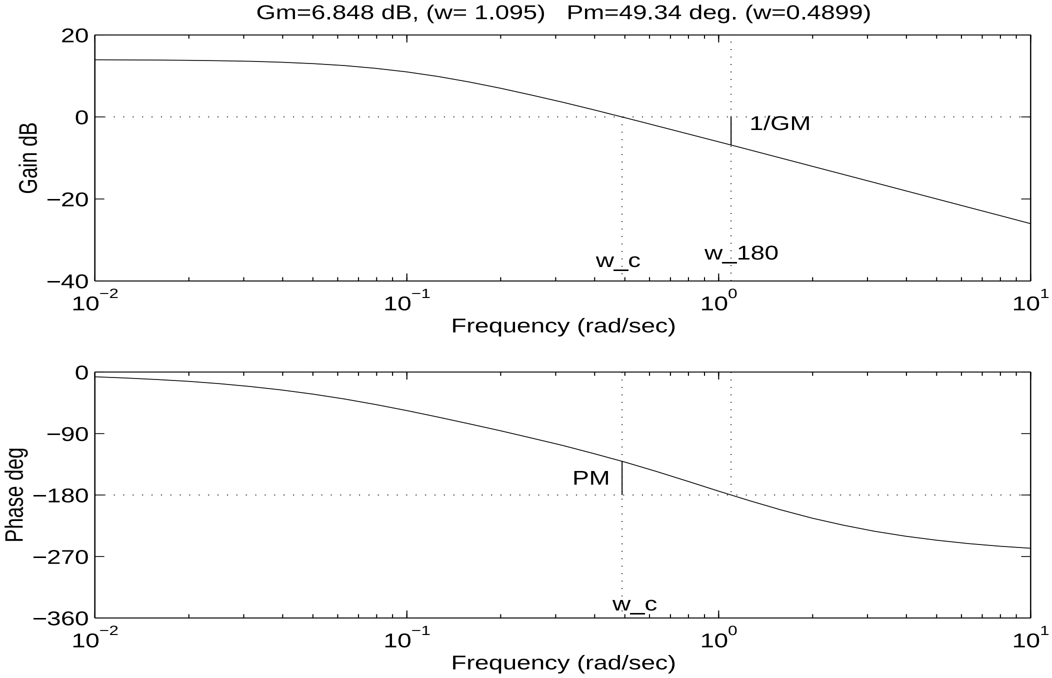

Figure 9.3: Bode plot for $H_0(s) = 0.5(1-s)/((1+0.1s)(1+s))$. The Gain margin GM and the Phase Margin PM is indicated. For this system we have $GM = 2.2$, $PM = 49.34°$, $\omega_c = 0.4899$ $[rad/s]$ and $\omega_{180} = 1.095$ $[rad/s]$. The figure shows the magnitude in $[dB]$ and $1/GM$ is illustrated. Transformation to $[dB]$ is done by $1/GM = -20 \cdot log(GM)$ $[dB]$. Here the MATLAB Control System Toolbox function *margin*is used.

Example 9.2 (loop transfer function)
Consider given a Proportional and Integral (PI) controller and a process represented/modeled by a 1st order process with time delay, i.e.

$$h_c = K_p \frac{1+T_i s}{T_i s} \tag{9.35}$$

$$h_p = \frac{k}{1+Ts} e^{-\tau s} \tag{9.36}$$

Let us for simplicity chose the integral time constant equal to the time constant of the process, $T_i = T$. Hence, we have the following loop transfer function ($h_0 = h_p h_c$),

$$h_0(s) = h_p h_c = k \frac{1}{s} e^{-\tau s} \tag{9.37}$$

$$k = \frac{K_p k}{T} \tag{9.38}$$

We will in the following show that it is, in particular, very simple to derive Gain Margin (GM) and Phase Margin (PM) for a process with a loop transfer function of this form.

Example 9.3 (Magnitude and Phase of integrator with time delay)
Given a Proportional (P) controller and a process represented by an integrator and a transport delay, i.e.,

$$h_c = K_p, \tag{9.39}$$

$$h_p = \frac{1}{s} e^{-\tau s}. \tag{9.40}$$

Note that it is usually sufficient to use a P controller in order to control integrating processes. However, a PI controller may also be used if there are modelling errors. This gives the following loop transfer function

$$h_0(s) = K_p \frac{1}{s} e^{-\tau s} \tag{9.41}$$

We will in the following derive the magnitude and the phase of the frequency response of the system. The frequency response of the process is defined by putting $s = j\omega$ in the transfer function $h_0(s)$, i.e.,

$$h_0(j\omega) = K_p \frac{1}{j\omega} e^{-\tau j\omega} \tag{9.42}$$

The magnitude and phase is given by

$$\mid h_0(j\omega) \mid = \frac{K_p}{\omega} \tag{9.43}$$

$$\angle h_0(j\omega) = -(\frac{\pi}{2} + \tau\omega) \tag{9.44}$$

This means that the frequency response, $h_0(j\omega)$, can be written in polar form as follows

$$h_0(j\omega) = \mid h_0(j\omega) \mid e^{j\angle h_0(j\omega)} = \frac{K_p}{\omega} e^{-j(\frac{\pi}{2}+\tau\omega)} \tag{9.45}$$

Example 9.4 (Gain and Phase margin)
Consider the following loop transfer function

$$h_0(s) = K_p \frac{1}{s} e^{-\tau s} \tag{9.46}$$

The magnitude and phase of the frequency response, $h_0(j\omega)$, was derived in Example 9.3.

The gain crossover frequency, ω_c, is given by

$$\begin{array}{c} \omega_c = sol_\omega[\mid H_0(j\omega) \mid -1 = 0] = sol_\omega[\frac{K_p}{\omega} - 1 = 0] \\ \Downarrow \\ \omega_c = K_p \end{array} \tag{9.47}$$

The phase crossover frequency is given by

$$\begin{array}{c} \omega_{180} = sol_\omega[\angle H_0(j\omega) + 180 = 0] = sol_\omega[-(\frac{\pi}{2} + \tau\omega) + 180^\circ = 0] \\ \Downarrow \\ \omega_{180} = \frac{\pi}{2\tau} \end{array} \tag{9.48}$$

Hence, the magnitude and phase of the frequency response is given by

$$\mid h_0(j\omega_{180}) \mid \; = \; \frac{K_p}{\omega_{180}} = \frac{2\tau K_p}{\pi} \tag{9.49}$$

$$\angle h_0(j\omega_c) \; = \; -(\frac{\pi}{2} + \tau\omega_c) = -(\frac{\pi}{2} + \tau K_p) \tag{9.50}$$

Hence, we have the following Gain Margin (GM) and Phase Margin (PM).

$$GM \; = \; \frac{\pi}{2\tau K_p} \tag{9.51}$$

$$PM \; = \; \frac{\pi}{2} - \tau K_p \tag{9.52}$$

Note that if we specify the closed loop feedback system to have a Gain Margin, $GM = 2$, then we should specify a proportional constant

$$K_p = \frac{\pi}{4\tau} \tag{9.53}$$

This controller gives a Phase Margin

$$PM = \frac{\pi}{4} = 45^\circ \tag{9.54}$$

Remark 9.1 Assume that there are an uncertainty, $d\tau$, in the transport delay, τ, which is used in the process model. Let $\tau_p = \tau + d\tau$ be the true transport delay in the process. Hence, τ is the modelled transport delay and $d\tau$ the uncertainty (or the error in the transport delay). The Phase Margin (PM) can be used to identify how much uncertainty, $d\tau$, which can be tolerated before the closed loop feedback system becomes unstable.

If PM is the Phase Margin computed with respect to the model with transport delay τ, and PM_p is the Phase Margin for the real process with transport delay τ_p, then we have that

$$PM_p = PM - d\tau\omega_c \tag{9.55}$$

The system becomes unstable if we have an uncertainty $d\tau$ which results in a zero Phase Margin, i.e., $PM_p = 0$. Hence, we have the following upper limit for the uncertainty in the transport delay

$$d\tau_{\max} = \frac{PM}{\omega_c} \tag{9.56}$$

This means that the closed loop feedback system can tolerate an uncertainty in the transport delay, $d\tau$, given by

$$d\tau < \frac{PM}{\omega_c} = d\tau_{\max} \tag{9.57}$$

where PM and ω_c is computed with respect to the loop transfer function, $h_0(s) = h_p(s)h_c(s)$. Note also that if the unit of gain crossover frequency, ω_c is given in $[rad/s]$ then we must have that the unit of the Phase Margin PM is given in $[rad]$. The unit in the uncertainty, $d\tau$, is then consistent and given in $[s]$. Remember also that 1 $[rad] = 180°/\pi \approx 57.3°$.

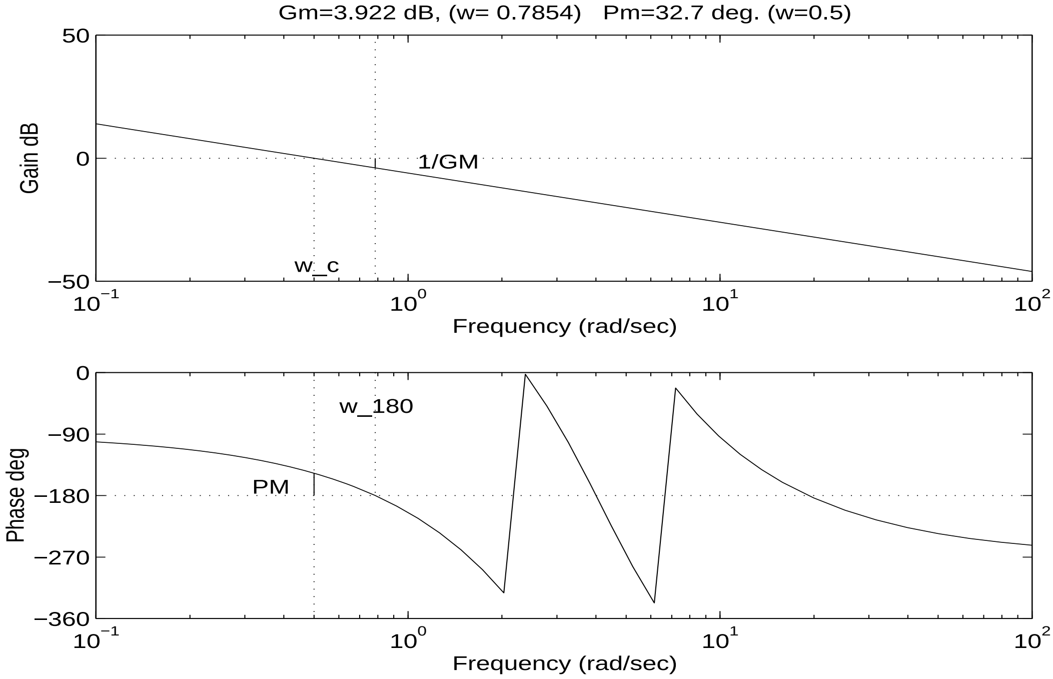

Figure 9.4: Bode plot for $H_0(s) = 0.5e^{-2s}/s$. The Gain Margin (GM) and the Phase Margin (PM) are indicated in the Figure. The closed loop feedback system has an $GM = \pi/2$, $PM = \pi/2 - 1 = 32.7°$, $\omega_c = 0.5$ $[rad/s]$ and $\omega_{180} = \pi/4$ $[rad/s]$. Nothe that the figure presents the magnitude in $[dB]$ but that we have illustrated $1/GM$. Consistent transformation to dB is that $1/GM = -20 \cdot log(GM)$ $[dB]$. We have used the MATLAB Control System Toolbox function *margin*. Using Equation (9.57) we find that this system are tolerating an uncertainty $d\tau = 0.726[s]$in the transport delay, τ, before the system becomes unstable.

Example 9.5 (Design og PI regulator)
We will in this example suggest settings for the proportional constant, K_p, and the integral time, T_i, in a PI controller. The process is the same as in Example 9.2, but with specific umerical values.

$$h_c = K_p \frac{1+T_i s}{T_i s} \tag{9.58}$$

$$h_p = \frac{2}{1+4s} e^{-s} \tag{9.59}$$

We demand that the gain Margin should be $GM = 2$. We also want the Phase Margin to be $PM > 30°$. We note that there is a transport delay $\tau = 1$ in the process.

For simplicity, let us chose the integral time constant equal to the time constant in the process, i.e.,

$$T_i = T = 4 \tag{9.60}$$

This may often be a reasonable setting for the integral time constant. Hence, we have the following simple loop transfer function

$$h_0(s) = k \frac{1}{s} e^{-s} \tag{9.61}$$

$$k = \frac{1}{2} K_p \tag{9.62}$$

We are now using the results found for GM and PM as found in Example 9.4. 9.4.

$$GM = \frac{\pi}{2\tau k} = \frac{\pi}{K_p} \tag{9.63}$$

$$PM = \frac{\pi}{2} - \tau k = \frac{\pi}{2} - \frac{K_p}{2} \tag{9.64}$$

the demands for the gain Margin $GM = 2$ and the Phase Margin $PM > 30°$ gives

$$GM = \frac{\pi}{K_p} = 2, \tag{9.65}$$

$$PM = \frac{\pi}{2} - \frac{K_p}{2} > 30°. \tag{9.66}$$

From the demand $GM = 2$ we find that

$$K_p = \frac{\pi}{2}. \tag{9.67}$$

Moreover, we find that the system has a Phase Margin

$$PM = \frac{\pi}{4} = 45° \tag{9.68}$$

As we see, that we can not demand a Gain Margin GM and at the same time demand a given Phase Margin PM. If the Phase Margin is not acceptable we can chose another value for T_i and recompute the calculations.

See Figures 9.6 and 9.7 for time response. simulations of the closed loop feedback system. We have also compared the PI controller settings found in this example with the settings found from Ziegler-Nichols method for PI controller synthesis.

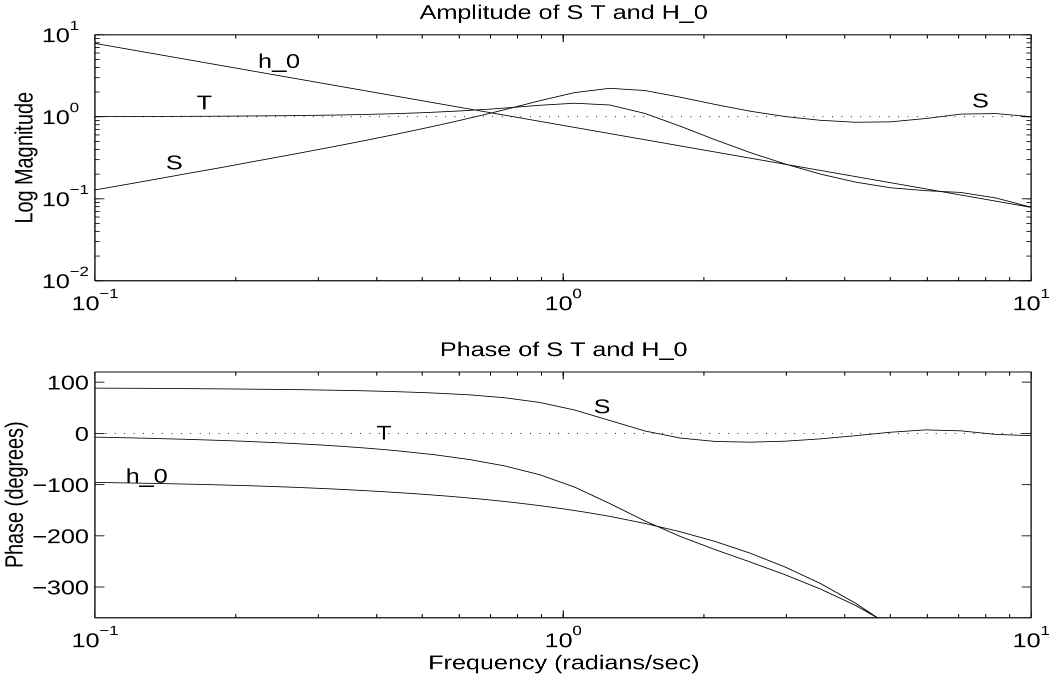

Figure 9.5: The figure shows magnitude and phase shift plot of S, T and H_0 for data as given in Example 9.5. Process $h_p(s) = 2e^{-s}/(4s+1)$, PI-controller $H_c(s) = K_p(1+T_is)/(T_is)$ with $T_i = 4$ and $K_p = \pi/2$. This system have $GM = 2$ and $PM = 45°$. The closed loop system have negative feedback.

9.4 Bodes stability criterion

The transfer function of a feedback controlled system is given by

$$y = \frac{h_0(s)}{1+h_0(s)}r + \frac{1}{1+h_0(s)}y^s \tag{9.69}$$

where negative feedback is assumed and r is the reference signal and y^s the disturbance. We have earlier shown that the stability of a feedback system is given by the *characteristic equation* which is defined from

$$1 + h_0(s) = 0 \tag{9.70}$$

The characteristic equation is obtained by putting the denominator of the transfer function equal to zero.

As we see, if the denominator of the transfer function is identical equal to zero, then the output y will be undefined (infinity). This indicates that a value of the loop transfer function, $h_0(s) = -1$, is a critical point. This is the basic for the classical Bode stability criterion and also the similar Nyquist stability criterion.

The frequency response of the loop transfer function, $h_0(j\omega)$, is obtained by putting $s = j\omega$ in the transfer function $h_0(s)$. We have

$$h_0(j\omega) =\mid h_0 \mid e^{j\angle h_0} \tag{9.71}$$

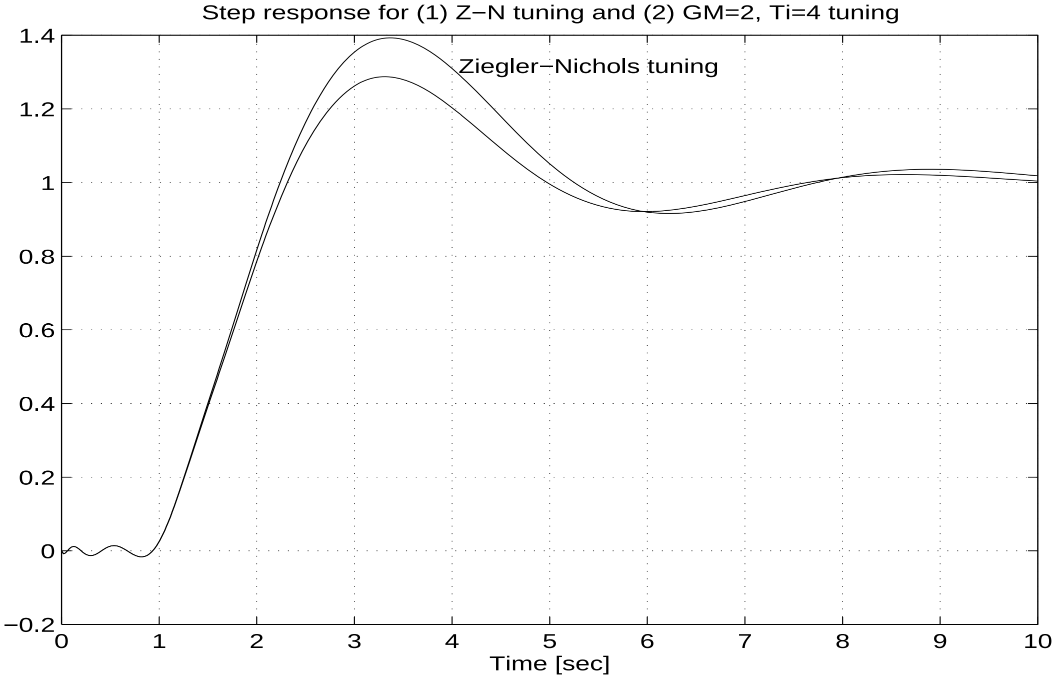

Figure 9.6: The figure shows the closed loop time response in the output y after a positive unit step response at time $t = 0$ in the reference. The process and data is as in Example 9.5, i.e. the process is $h_p(s) = 2e^{-s}/(4s+1)$ and PI controller $h_c(s) = K_p(1+T_is)/(T_is)$. Negative feedback is used. Two PI controller parameter settings are shown. The first is a Ziegler-Nichols settings. The critical gain and frequency is found to be $K_{cu} = 3.467$ and $\omega_{cu} = 1.715$. The ultimate period is then $P_u = 2\pi/\omega_{cu}$. This gives the Z-N PI controller parameters $K_p = K_{cu}/2.2 = 1.576$ and $T_i = P_u/1.2 = 3.052$. The closed loop system with Z-N settings have the margins $GM = 1.91$ and $PM = 38.6$. The other PI settings are simply found by putting $T_i = T = 4$and cancelling a pole in the loop transfer function, and thereby computing K_p so that $GM = 2$. This gives $T_i = 4$ and $K_p = \pi/2$. The closed loop system with this settings have a phase margin $PM = 45°$. As we see, the Z-N setting gives more overshot on the output compared to the second PI controller setting. The second setting is therefore to be chosen. Se Figure 9.7 for the response in the control input, u.

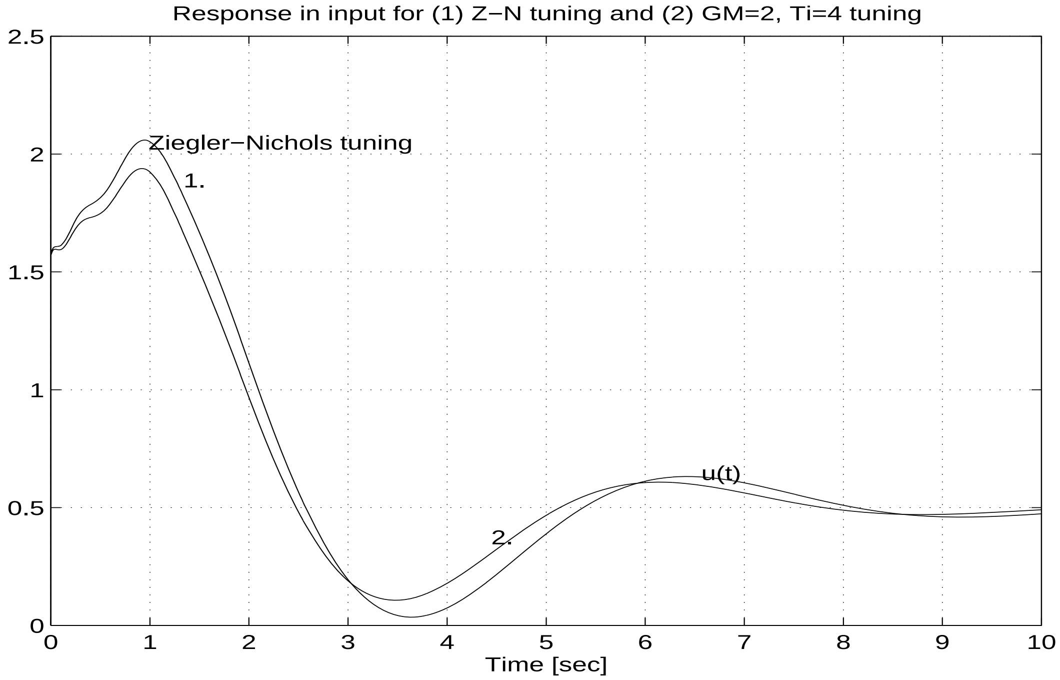

Figure 9.7: The figure shows the closed loop time response in the input u after a positive unit step response at time $t = 0$ in the reference. The process and data is as in Example 9.5, i.e. the process is $h_p(s) = 2e^{-s}/(4s+1)$ and PI controller $h_c(s) = K_p(1+T_is)/(T_is)$. Negative feedback is used. Two PI controller parameter settings are shown. The first is a Ziegler-Nichols settings. The critical gain and frequency is found to be $K_{cu} = 3.467$ and $\omega_{cu} = 1.715$. The ultimate period is then $P_u = 2\pi/\omega_{cu}$. This gives the Z-N PI controller parameters $K_p = K_{cu}/2.2 = 1.576$ and $T_i = P_u/1.2 = 3.052$. The closed loop system with Z-N settings have the margins $GM = 1.91$ and $PM = 38.6$. The other PI settings are simply found by putting $T_i = T = 4$and cancelling a pole in the loop transfer function, and thereby computing K_p so that $GM = 2$. This gives $T_i = 4$ and $K_p = \pi/2$. The closed loop system with this settings have a phase margin $PM = 45°$. As we see, the Z-N setting gives more overshot on the input and more control energy compared to the second PI controller setting. The second setting is therefore to be chosen. Se Figure 9.6 for the response in the control input, y.

Assume now that the frequency response of the loop transfer function in a point (i.e. for a given critical frequency $\omega = \omega_{cu}$) has magnitude equal to 1 and phase shift equal to $-180°$. Hence $\mid h_0 \mid= 1$ and $\angle h_0 = -\pi$). This means that

$$h_0(j\omega) = 1 \cdot e^{-j\pi} \tag{9.72}$$

Putting this into the denominator in the transfer function $T(j\omega) = h_0(j\omega)/(1 + h_0(j\omega)$ we find that

$$h_0(j\omega) + 1 = 1 \cdot e^{-j\pi} + 1 = 1 \cdot cos(-\pi) + 1j \cdot sin(-\pi) + 1 = -1 + 0j + 1 = 0 \tag{9.73}$$

This means that the output response of the closed loop system $y = Tr + Sy^s$ becomes undefined, and the system becomes unstable if the phase of the loop transfer function becomes $-180°$ and the magnitude of the loop transfer function becomes 1. This means that the closed loop feedback system becomes unstable if $|h_0(j\omega_{180})| = 1$, which also is equivalent that the system becomes unstable for a Gain Margin, GM $=$ 1. On this basis we formulate the Bodes stability criterion as follows

Definition 9.2 (Bode stability criterion)
A closed loop feedback system is unstable if the frequency response, $h_0(j\omega)$, of the loop transfer function, $h_0(s)$, have a magnitude greater than 1, i.e. $\mid h_0(j\omega) \mid > 1$ at the critical frequency.

A closed loop system is stable if the frequency response, $h_0(j\omega)$, of the loop transfer function, $h_0(s)$, have a magnitude less than 1, i.e. $\mid h_0(j\omega) \mid < 1$ at the critical frequency. This is formulated as follows

$$\text{stable closed loop system} \quad \Leftrightarrow \quad \mid h_0(j\omega_{180} \mid < 1 \tag{9.74}$$

where the critical frequency ω_{180} is the phase crossover frequency where the phase shift of the frequency response of the loop transfer function $h_0(j\omega)$ is $-180°$ i.e. $\angle h_0(j\omega_{180}) = -180°$.

Remark 9.2 *The Bode stability criterion* have two assumptions. The first one is that the Bode stability criterion only is valid for open loop stable systems, that means that the process $h_p(s)$ is stable. The second assumption is that Bodes stability criterion only is valid for systems where the phase shift $\angle h_0(j\omega)$ passes $-180°$ once.

If $\angle h_0(j\omega)$ passes $-180°$ several times or the open loop system is unstable, then Nyquists stability criterion may be used.

Remark 9.3 A great advantage with Bodes stability criterion is that it can be used for stability analysis of systems with transport delays. Bodes stability criterion can handle transport delay terms of the type $e^{-\tau s}$ in the transfer function. We does not need Padé approximations to $e^{-\tau s}$ which is the case when we are to compute the roots of the characteristic equation, ore the eigenvalues of the corresponding A matrix (for continuous systems).

Example 9.6 (Stability)
Given a system described by the transfer function

$$h(s) = 2k\frac{e^{-s}}{1 + 4s} \tag{9.75}$$

The frequency response is then given by putting $s = j\omega$, i.e.,

$$h(j\omega) = 2k\frac{e^{-j\omega}}{1 + j4\omega} \tag{9.76}$$

This can be written on polar form, expressed in terms of the magnitude and the phase shift

$$h(j\omega) = 2k\frac{e^{-j\omega}}{\sqrt{1+(4\omega)^2}e^{j\arctan(4\omega)}} = \frac{2k}{\sqrt{1+(4\omega)^2}}e^{-j(\omega+\arctan(4\omega)} \tag{9.77}$$

Hence, the magnitude and phase characteristics of the system is given by

$$\angle h(j\omega) = \phi = -(\omega + \arctan(4\omega)), \quad \mid h(j\omega) \mid = \frac{2k}{\sqrt{1+(4\omega)^2}}. \tag{9.78}$$

As we see, the proportional constant, k, does nor exists in the phase characteristic. This is typically also the case for systems controlled by P, PI or PID controllers.

We will first compute the phase crossover frequency, ω_{180}. Hence, we have to solve

$$\phi = -(\omega + \arctan(4\omega)) = -\pi \tag{9.79}$$

with respect to ω. This can for instance be done by the Newton-Raphsons method for the solution of non-linear equations $f(\omega) = 0$. here a trial and error method is used.

$$\begin{array}{lll} \omega = 1.5 & \Rightarrow & \phi = -2.91\frac{180}{\pi} = -166^\circ \\ \omega = 1.7 & \Rightarrow & \phi = -3.12\frac{180}{\pi} = -179^\circ \\ \omega = 1.715 & \Rightarrow & \phi = -3.141\frac{180}{\pi} = -179.96^\circ \\ \omega = 1.72 & \Rightarrow & \phi = -3.146\frac{180}{\pi} = -180.2^\circ \end{array} \tag{9.80}$$

This means that the phase crossover frequency, ω_{180}, is given by

$$\omega_{180} \approx 1.715. \tag{9.81}$$

The system is at the stability limit when the magnitude, $\mid h(j\omega) \mid)$, is equal to one at the frequency $\omega = \omega_{180}$. The critical and largest gain, k, this system can tolerate before being unstable is therefore given from

$$\mid h(j\omega) \mid) = \frac{2k}{\sqrt{1+(4\omega)^2}} = 1. \tag{9.82}$$

Using $\omega = \omega_{180}$ gives

$$K_{cu} = \frac{\sqrt{1+16\omega_{180}^2}}{2} = 3.466 \tag{9.83}$$

Hence, the system is stable if

$$-K_{cu} < k < K_{cu} \tag{9.84}$$

Note that the result only says something about the magnitude of the gain k.

Remark 9.4 The Bode stability criterion is often used to compute the critical gain K_{cu} at the critical (phase crossover) frequency ω_{180}.

The critical gain K_{cu} and the phase crossover frequency ω_{180} is the basic parameters for using the Ziegler-Nichols method for the PID controller settings. In particular, the Ziegler-Nichols methog gives the following PI controller parameters in terms of K_{cu} and ω_{180}

$$K_p = \frac{K_{cu}}{2.2}, \quad P_u = \frac{2\pi}{\omega_{180}}, \quad T_i = \frac{P_u}{1.2} = \frac{2\pi}{1.2}\frac{1}{\omega_{180}} \approx 5.25\frac{1}{\omega_{180}} \tag{9.85}$$

where P_u is the oscillating period, ore ultimate period. Hence, P_u is the periode time of an oscillation in the output $y(t)$. Note that the symbol ω_{cu} sometimes is used instead of ω_{180}, when the magnitude is 1 at the same time.

Remark 9.5 Note also that when the gain of the system is identical to K_{cu} then the gain crossover frequency, ω_c, will be equal to the phase crossover frequency ω_{180}. Hence, the frequency in which $\omega_c = \omega_{180}$ is referred to as the critical frequency.

9.5 Ziegler-Nichols method for PID-controller tuning

The Ziegler-Nichols (ZN) method for PID controller tuning can be used both as an experimental and as an analytical model based method for PID controller tuning. We will here start by presenting the model based ZN method.

Definition 9.3 (Critical gain, K_{cu})

Assume that the PID controller is substituted with a proportional P controller. Note that a PID controller is reduced to a P controller by letting $T_i = \infty$ and $T_d = 0$. The frequency response of the loop transfer function is then given by

$$h_0(j\omega) = K_p h_p(j\omega) \tag{9.86}$$

The critical gain, K_{cu}, is the largest gain which results in

$$|h_0(j\omega_{180})| = 1, \tag{9.87}$$

which gives

$$K_{cu} = \frac{1}{|h_p(j\omega_{180})|}. \tag{9.88}$$

This means that if the system, $h_p(s)$, is controlled with a P controller with $K_p = K_{cu}$ then the closed loop system will have a gain margin $GM = 1$ and the system is at the stability limit.

The Ziegler-Nichols method for PID controller tuning is then a function of the parameters K_{cu} and ω_{180}. Note also that the ultimate period is given by

$$P_u = \frac{2\pi}{\omega_{180}} \tag{9.89}$$

Table 9.1: ZN-method.

Controller	K_p	T_i	T_d
P	$\frac{K_{cu}}{2}$	∞	0
PI	$\frac{K_{cu}}{2.2}$	$\frac{2\pi}{1.2}\frac{1}{\omega_{180}}$	0
PID	$\frac{3K_{cu}}{5}$	$\frac{P_u}{2}$	$0.12P_u$

9.6 Minimum-phase and non-minimum-phase systems

Minimum-phase system

It can be shown that among all systems with frequency response, $h(j\omega)$, which have the same magnitude $\mid h(j\omega) \mid$ there exist one system with the smallest negative phase shift $\angle h(j\omega)$. This system is called a minimum phase system.

A necessary and sufficient condition for a system to be a minimum-phase system is that all zeroes and poles of the transfer function, $h(s)$, lies in the left half plane, i.e. to the left of the imaginary axis in the complex plane.

Non-minimum-phase system

Zeroes in the right half plane (RHP) and transport delay terms gives more phase shift to the transfer function $h_1(s)$ compared to a transfer function $h_2(s)$ without zeroes in the RHP and without transport delay terms. At the same time the transfer function $h_1(s)$ may have the same magnitude as $h_2(s)$. If the transfer functions $h_1(s)$ and $h_2(s)$ have the same magnitude for all frequencies ω where $s = j\omega$, then the system represented by $h_1(s)$ will be a non-minimum-phase system.

An example is the system (or process) $h_1(s) = e^{-\tau s}$ which have magnitude $\mid h_1(j\omega) \mid= 1$ and phase shift $\angle h_1(j\omega) = -\tau\omega$. This system have the same magnitude as the system $h_2(s) = 1$ which has zero phase shift, i.e. $\angle h_1(j\omega) = 0$. Hence, $h_1(s)$ is a non-minimum phase system.

Another example is the system $h(s) = \frac{a-s}{a+s}$ which have a zero $s = 1$ in the right half of the complex plane. The magnitude is $\mid h(j\omega) \mid= 1$ and the phase shift is $\angle h(j\omega) = -2\arctan(a\omega)$. This system is a non-minimum phase system because the system $h(s) = 1$ have the same magnitude but less phase shift, i.e. $\angle h_1(j\omega) = 0$.

A non-minimum-phase system is often recognized because the output, $y(t)$, from such a system have an negative inverse response (or time delay) after a positive unit response is feed into the input, $u(t)$. See Figure 9.8 for an illustration of the output response of a non-minimum-phase system. It should be remarked that it usually is "harder" to control a non-minimum-phase system compared to a minimum-phase system.

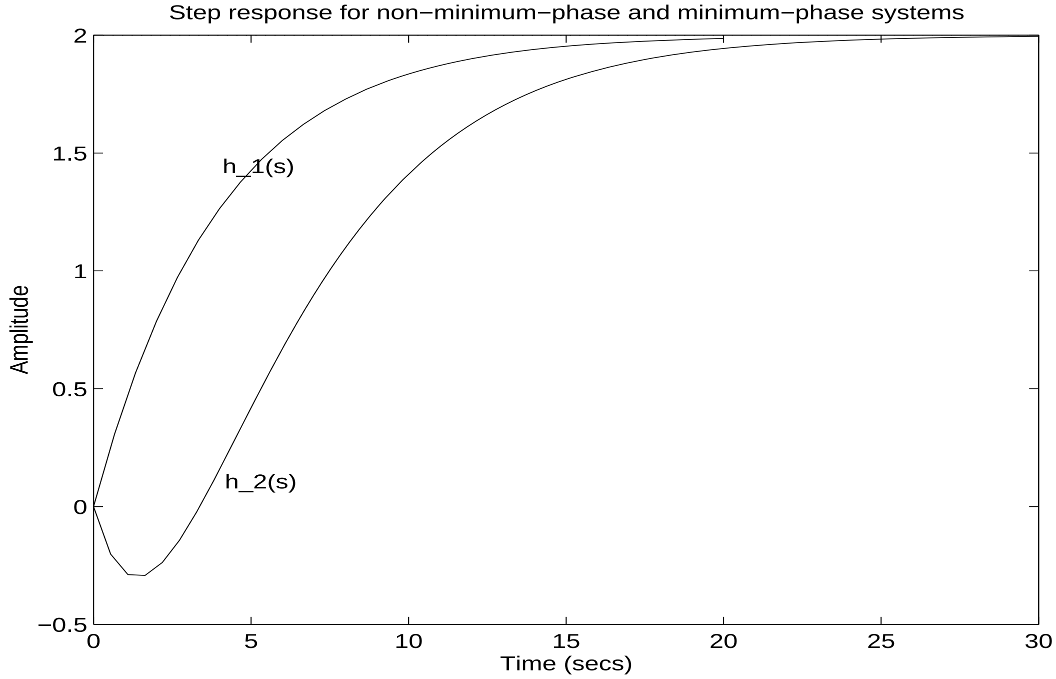

Figure 9.8: Unit step response plot of a minimum-phase system $h_1(s) = 2/(4s + 1)$ and a non-minimum-phase system $h_2(s) = 2e^{-5s}/(4s + 1)$. The response of the non-minimum-phase system is recognized from the negative inverse response. The MATLAB Control Systems toolbox function *step* is used.

9.7 The bandwidth of a feedback control system

The bandwidth of a feedback control system is defined as the frequency region in which the control system is effective. Let the bandwidth be described by the frequency region $[\omega_1, \omega_2]$.

Assume that the system is excited by a disturbance with frequency ω. This means that the control system can control and take care of the disturbances with frequencies $\omega_1 < \omega < \omega_2$.

A control system is normally effective if the system is excited by low frequent noise. Hence, we say that the control system is effective for low frequencies. This is in particular the case for control systems with integral actions. The lower frequency in the bandwidth is therefore put to $\omega_1 = 0$.

A definition of the bandwidth is therefore, $\omega_B = [0, \omega_2]$ or simply $\omega_B = \omega_2$.

A common definition of the upper frequency, ω_2, in the bandwidth is the gain crossover frequency ω_c. The gain crossover frequency, ω_c, is the frequency in which the magnitude of h_0 passes 1 the first time. It can be shown that ω_c only is an approximation to ω_B if the bandwidth is defined as the frequency region in which the control system is effective in order to control disturbances.

The influence of the disturbance, y^s, on the process output, y, is described by $y = Sy^s$ where $S = 1/(1 + h_0)$ is the sensitivity function. If the control system is to take care of a disturbance y^s with frequency ω then we must have that $\mid S(\omega) \mid< 1$ at this frequency, otherwise the disturbance influence on the output would have been amplified. From Figure 6.2 we see that $\mid S \mid< 1$ for frequencies somewhat less than ω_c. For the system as illustrated in Figure 6.2 and most common control systems we have that the exact bandwidth is somewhat less than ω_c. Hence we have that the bandwidth is

$$\omega_B < \omega_c \tag{9.90}$$

This means also that the control system is not effective if the system is excited by disturbances with a frequency greater than the bandwidth ω_B. An upper limit for the bandwidth of a control system is therefore ω_c.

Chapter 10

Discrete time systems

10.1 Introduction

we have already discussed discrete time versions of continuous time models and in particular discrete time versions of the standard PID controller as discussed in Chapter 4. However, we will in this section discuss and present the deviation form of the PID controller. Often incremental form or velocity form of the PID controller is used instead of deviation form. The point is that only the deviation Δu_k is computed at each new sample at time, k, and then use $u_k = u_{k-1} + \Delta u_k$ as the actual control signal. This version of the PID controller have some advantages and is usually the version implemented in practice and commercial PID controllers.

10.2 θ method for discretization

Consider a system described by a possibly non linear model

$$\dot{x} = f(x). \tag{10.1}$$

Consider the following method of obtaining a discrete version of (10.1)

$$\frac{x_{k+1} - x_k}{\Delta t} = \theta f(x_k) + (1 - \theta) f(x_{k+1}). \tag{10.2}$$

This is dentoted a so called θ method for discretization. One should here note that we may chose θ to obtain different famous discrete integration methods:

- $\theta = 1$ gives the explicit Euler method.
- $\theta = 0$ gives the implicit Euler method.
- $\theta = \frac{1}{2}$ gives the trapezoid method.

One should here note that the explicit Euler method may be numerically unstable and that there is an upper limit for the step length (sampling time) parameter, Δt. The explicit Euler method is a good choice when the step length parameter is chosen "small" and is accurate enough for most control systems analysis and design.

On the other hand the implicit methods are numerically stable for all choices of step length parameter, Δt. However, the solutions may be inaccurate for large step length parameters Δt.

10.3 Deviation form of the PI controller

As an introduction to the deviation form of a PID controller in the next section we here discuss a deviation formulation of the PI controller. We also for the sake of simplicity only use the explicit Euler method for discretization.

A conventional PI-controller can be written as

$$u = K_p \frac{1+T_i s}{T_i s}(r-y) = K_p(r-y) + \frac{K_p}{T_i}\frac{1}{s}(r-y). \tag{10.3}$$

Defining the PI-controller state, z, as

$$z = \frac{1}{s}(r-y). \tag{10.4}$$

Hence, the PI controller can in continuous time be written as

$$\begin{aligned} \dot{z} &= r-y, & (10.5)\\ u &= K_p(r-y) + \frac{K_p}{T_i}z. & (10.6) \end{aligned}$$

We will in the following assume that the reference is constant, $r_k = r$. However, note that in the final controller we use varying reference signal. A discrete formulation of the PI controller is then

$$\begin{aligned} z_{k+1} - z_k &= \Delta t(r-y_k), & (10.7)\\ u_k &= K_p(r-y_k) + \frac{K_p}{T_i}z_k, & (10.8) \end{aligned}$$

where Δt is the sampling interval. A deviation formulation of the PI controller is then found as follows

$$\begin{aligned} u_k - u_{k-1} &= K_p(r-y_k) + \frac{K_p}{T_i}z_k - (K_p(r-y_{k-1}) + \frac{K_p}{T_i}z_{k-1}) \\ &= -K_p(y_k - y_{k-1}) + \frac{K_p}{T_i}(z_k - z_{k-1}). & (10.9) \end{aligned}$$

From (10.7) we have that $z_k - z_{k-1} = \Delta t(r - y_{k-1})$. Substituting this into (10.9) gives

$$u_k = u_{k-1} + G_1(y_k - y_{k-1}) + G_2(y_{k-1} - r). \tag{10.10}$$

where

$$G_1 = -K_p,\ G_2 = -\frac{K_p}{T_i}\Delta t. \tag{10.11}$$

Furthermore, using that $y = Dx + w$ gives

$$u_k = u_{k-1} + G_1 D\Delta x_k + G_2(y_{k-1} - r). \tag{10.12}$$

The above discussion shows that the PI controller is exactly of the same structure as a state feedback controller (LQ optimal controller). The controller takes feedback from the deviation state vector $\Delta x_k = x_k - x_{k-1}$ while the PI-controller only uses feedback from the output deviation $\Delta y_k = D\Delta x_k$. But notice that $y_k = \Delta y_k + y_{k-1}$.

10.4 Deviation form of the PID controller

The PID controller may in discrete time be formulated on so called deviation form. Also incremental form or velocity form is used on this version of the discrete PID controller. We have that the PID controller may be expressed as

$$u_k = u_{k-1} + \Delta u_k, \tag{10.13}$$

where Δu_k is the computed control deviation (increment) which usually is computed as

$$\Delta u_k = g_0 e_k + g_1 e_{k-1} + g_2 (y_k - 2y_{k-1} + y_{k-2}), \tag{10.14}$$

where g_0, g_1 and g_2 are related to the PID controller parameters K_p, T_i and T_d. One should noe that different methods for discretization gives different formulas for g_0, g_1 and g_2.

Note also that the computed control deviation in (10.14) is based on known present and past variables and that $e_k = r_k - y_k$ and $e_{k-1} = r_{k-1} - y_{k-1}$.

10.4.1 The continuous time PID controller

We here review the PID controller in the time domain.

The PID controller may be written

$$u = K_p e + z + K_p T_d s e \tag{10.15}$$

where

$$z = \frac{K_p}{T_d s} e, \quad \Rightarrow sz = \frac{K_p}{T_i} e. \tag{10.16}$$

Inverse Laplace-transformation gives

$$\begin{aligned} u &= K_p e + z + K_p T_d \dot{e}, & (10.17)\\ \dot{z} &= \frac{K_p}{T_i} e. & (10.18) \end{aligned}$$

It make sense to put $e(t_0) = r - y = 0$ and $\dot{e}(t_0) = 0$ at startup. We then have that the initial value for the controller state may be chosen as

$$z(t_0) = u_0, \tag{10.19}$$

where u_0 is a nominal control or working point for the process when turning on the controller system in automatic mode . The nominal control may be found by analyzing the steady state behavior of the process. We have

$$y_0 = k u_0 = r, \tag{10.20}$$

where k is the gain. this gives the following initial value.

$$z(t_0) = \frac{r(t_0)}{k}. \tag{10.21}$$

10.4.2 Deviation form of the PID controller using the explicit Euler method

We have two possibilities for implementing the derivation $\dot{e}$. The first possibility is to use

$$\dot{e} \approx \frac{e_k - e_{k-1}}{\Delta t}. \tag{10.22}$$

the second possibility and the most common choice is to not take the derivative of the reference signal, i.e. using

$$\dot{e} = -\dot{y} \approx -\frac{y_k - y_{k-1}}{\Delta t}. \tag{10.23}$$

using the last choice and the explicit euler method for discretization of the controller state space model (10.18) gives

$$u_k = K_p e_k + z_k - \frac{K_p T_d}{\Delta t}(y_k - y_{k-1}), \tag{10.24}$$

$$z_{k+1} = z_k + \Delta t \frac{K_p}{T_i} e_k. \tag{10.25}$$

with initial value $z_0 = u_0 = \frac{r_0}{k}$.

This discrete state space model for the PID controller may be used directly. However, a formulation on deviation form may be derived as follows (compute the deviation $\Delta u_k = u_k - u_{k-1}$). This gives

$$\begin{aligned} u_k - u_{k-1} &= K_p e_k + z_k - \frac{K_p T_d}{\Delta t}(y_k - y_{k-1}) \\ &- (K_p e_{k-1} + z_{k-1} - \frac{K_p T_d}{\Delta t}(y_{k-1} - y_{k-2})) \end{aligned} \tag{10.26}$$

which may be written as

$$u_k - u_{k-1} = K_p e_k + z_k - z_{k-1} - K_p e_{k-1} - \frac{K_p T_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \tag{10.27}$$

Using Eq. (10.25) we have that

$$z_k - z_{k-1} = \Delta t \frac{K_p}{T_i} e_{k-1}. \tag{10.28}$$

This gives

$$u_k = u_{k-1} + K_p e_k - K_p(1 - \frac{\Delta t}{T_i}) e_{k-1} - \frac{K_p T_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \tag{10.29}$$

Hence, this is of the form

$$u_k = u_{k-1} + g_0 e_k + g_1 e_{k-1} + g_2(y_k - 2y_{k-1} + y_{k-2}) \tag{10.30}$$

where

$$g_0 = K_p, \; g_1 = -K_p(1 - \frac{\Delta t}{T_i}), \; g_2 = -\frac{K_p T_d}{\Delta t}. \tag{10.31}$$

The PID controller written on the form in Eq. (10.30) with parameters as in Eq. (10.31) is denoted the velocity form, deviation form or incremental form, and often used in practical implementations.

10.4.3 Deviation form of the PID controller using the trapezoid method

Using the trapezoid method for integrating the controller state space model (10.18) gives

$$\frac{z_{k+1} - z_k}{\Delta t} = \frac{1}{2}\frac{K_p}{T_i} e_k + \frac{1}{2}\frac{K_p}{T_i} e_{k+1} \tag{10.32}$$

As we see, it is not possible to formulate an implementable dicrete state space model for the PID controller of the same form as when the Explicit Euler method was used, as in Equations (10.24) and (10.25). The reason for this is that we do not know $e_{k+1} = r_{k+1} - y_{k+1}$ which in this last case is needed in order to compute and update the controller state z_{k+1}.

However, we may use the trapezoid method in order to formulate the PID controller on deviation (incremental) form. Using that

$$z_k - z_{k-1} = \frac{\Delta t}{2}\frac{K_p}{T_i}(e_{k-1} + e_k) \tag{10.33}$$

and putting this into (10.27) gives

$$\begin{aligned} u_k &= u_{k-1} + K_p(1 + \frac{\Delta t}{2T_i})e_k - K_p(1 - \frac{\Delta t}{2T_i})e_{k-1} \\ &- \frac{K_p T_d}{\Delta t}(y_k - 2y_{k-1} + y_{k-2}) \end{aligned} \tag{10.34}$$

This gives the controller formulation

$$u_k = u_{k-1} + g_0 e_k + g_1 e_{k-1} + g_2(y_k - 2y_{k-1} + y_{k-2}) \tag{10.35}$$

where

$$g_0 = K_p(1 + \frac{\Delta t}{2T_i}), \;\; g_1 = -K_p(1 - \frac{\Delta t}{2T_i}), \;\; g_2 = -\frac{K_p T_d}{\Delta t}. \tag{10.36}$$

10.5 Discussion

We have in this chapter discussed the deviation form of the discrete time PID controller. This form of the PID controller may with advantage be used when implementing the PID controller in practice. This form of the controller also have some advantages, i.e., the controller is insensitive to constant offset values on the control action due to the fact that only the deviation $\Delta_k = u_k - u_{k-1}$ matters. Anti-Windup is also not a topic in the deviation form of the controller because the controller integrator state is eliminated from the controller.

Chapter 11

Time delay in discrete systems

11.1 Modeling of time delay

We will in this section discuss systems with possibly time delay. Assume that the system without time delay is given by a proper state space model as follows

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k, && (11.1) \\ y_k^- &= Dx_k + Eu_k, && (11.2) \end{aligned}$$

and that the output of the system, y_k, is identical to, y_k^-, but delayed a delay τ samples. The time delay may then be exact expressed as

$$y_{k+\tau} = y_k^-. \tag{11.3}$$

Discrete time systems with time delay may be modeled by including a number of fictive dummy states for describing the time delay. Some alternative methods are described in the following.

11.1.1 Transport delay and controllability canonical form

Formulation 1: State space model for time delay

We will include a positive integer number τ fictive dummy state vectors of dimension m in order for describing the time delay, i.e.,

$$\left.\begin{aligned} x_{k+1}^1 &= Dx_k + Eu_k \\ x_{k+1}^2 &= x_k^1 \\ &\vdots \\ x_{k+1}^\tau &= x_k^{\tau-1} \end{aligned}\right\} \tag{11.4}$$

The output of the process is then given by

$$y_k = x_k^\tau \tag{11.5}$$

We se by comparing the defined equations (11.4) and (11.5) is an identical description as the original state space model given by (11.1), (11.2 and (11.3). Note that we in

(11.4) have defined a number τm fictive dummy state variables for describing the time delay.

Augmenting the model (11.1) and (11.2) with the state space model for the delay gives a complete model for the system with delay.

$$\overbrace{\begin{bmatrix} x \\ x^1 \\ x^2 \\ \vdots \\ x^\tau \end{bmatrix}_{k+1}}^{\tilde{x}_{k+1}} = \overbrace{\begin{bmatrix} A & 0 & 0 & \cdots & 0 & 0 \\ D & 0 & 0 & \cdots & 0 & 0 \\ 0 & I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 \end{bmatrix}}^{\tilde{A}} \overbrace{\begin{bmatrix} x \\ x^1 \\ x^2 \\ \vdots \\ x^\tau \end{bmatrix}_k}^{\tilde{x}_k} + \overbrace{\begin{bmatrix} B \\ E \\ 0 \\ \vdots \\ 0 \end{bmatrix}}^{\tilde{B}} u_k \tag{11.6}$$

$$y_k = \overbrace{\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & I \end{bmatrix}}^{\tilde{D}} \overbrace{\begin{bmatrix} x \\ x^1 \\ x^2 \\ \vdots \\ x^{\tau-1} \\ x^\tau \end{bmatrix}_k}^{\tilde{x}_k} \tag{11.7}$$

hence we have

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}u_k \tag{11.8}$$
$$y_k = \tilde{D}\tilde{x}_k \tag{11.9}$$

where the state vector $\tilde{x}_k \in \mathbb{R}^{n+\tau m}$ contains n states for the process (11.1) without delay and a number τm states for describing the time delay (11.3).

With the basis in this state space model, Equations (11.8) and (11.9), we may use all our theory for analyse and design of linear dynamic control systems.

Formulation 2: State space model for time delay

The formulation of the time delay in Equations (11.6) and (11.7) is not very compacter. We will in this section present a different more compact formulation. In some circumstances the model from y_k^- to y_k will be of interest in itself. We start by isolating this model. Consider the following state space model where $y_k- \in \mathbb{R}^m$ s delayed an integer number τ time instants.

$$\overbrace{\begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ \vdots \\ x^\tau \end{bmatrix}_{k+1}}^{x^\tau_{k+1}} = \overbrace{\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ I & 0 & 0 & \cdots & 0 & 0 \\ 0 & I & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & 0 \end{bmatrix}}^{A^\tau} \overbrace{\begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ \vdots \\ x^\tau \end{bmatrix}_k}^{x^\tau_k} + \overbrace{\begin{bmatrix} I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}^{B^\tau} y_k^- \tag{11.10}$$

$$y_k = \overbrace{\left[\begin{array}{cccccc} 0 & 0 & 0 & \cdots & 0 & I \end{array}\right]}^{D^\tau} \overbrace{\left[\begin{array}{l} x \\ x^1 \\ x^2 \\ \vdots \\ x^{\tau-1} \\ x^\tau \end{array}\right]_k}^{x_k^\tau} \tag{11.11}$$

which may be written as

$$x_{k+1}^\tau = A^\tau x_k^\tau + B^\tau y_k^- \tag{11.12}$$

$$y_k = D^\tau x_k^\tau \tag{11.13}$$

where $x_k^\tau \in \mathbb{R}^{\tau m}$. the initial state for the delay state is put to $x_0^\tau = 0$. Note here that the state space model (11.10) and (11.11) is on so called controllability canonical form.

Combining (11.12) and (11.13) with the state space model equations (11.1) and (11.2), gives an compact model for the entire system, i.e., the system without delay from u_k to y_k^-, and for the delay from y_k^- to the output y_k.

$$\overbrace{\left[\begin{array}{c} x \\ x^\tau \end{array}\right]_{k+1}}^{\tilde{x}_k} = \overbrace{\left[\begin{array}{cc} A & 0 \\ B^\tau D & A^\tau \end{array}\right]}^{\tilde{A}} \overbrace{\left[\begin{array}{c} x \\ x^\tau \end{array}\right]_k}^{\tilde{x}_k} + \overbrace{\left[\begin{array}{c} B \\ B^\tau E \end{array}\right]}^{\tilde{B}} u_k \tag{11.14}$$

$$y_k = \overbrace{\left[\begin{array}{cc} 0 & D^\tau \end{array}\right]}^{\tilde{D}} \overbrace{\left[\begin{array}{c} x \\ x^\tau \end{array}\right]_k}^{\tilde{x}_k} \tag{11.15}$$

Note that the state space model given by Equations (11.14) and (11.15), is identical with the state space model in (11.6) and (11.7).

11.1.2 Time delay and observability canonical form

A simple method for modeling the time delay may be obtained by directly taking Equation (11.3) as the starting point. Combining $y_{k+\tau} = y_k^-$ with a number $\tau - 1$ fictive dummy states, $y_{k+1} = y_{k+1}, \cdots, y_{k+\tau-1} = y_{k+\tau-1}$ we may write down the following state space model

$$\overbrace{\left[\begin{array}{l} y_{k+1} \\ y_{k+2} \\ y_{k+3} \\ \vdots \\ y_{k+\tau} \end{array}\right]}^{x_{k+1}^\tau} = \overbrace{\left[\begin{array}{cccccc} 0 & I & 0 & \cdots & 0 & 0 \\ 0 & 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & I \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{array}\right]}^{A^\tau} \overbrace{\left[\begin{array}{l} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+\tau-1} \end{array}\right]}^{x_k^\tau} + \overbrace{\left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ I \end{array}\right]}^{B^\tau} y_k^- \tag{11.16}$$

$$y_k = \overbrace{\begin{bmatrix} I & 0 & 0 & \cdots & 0 \end{bmatrix}}^{D^\tau} \overbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+\tau-1} \end{bmatrix}}^{x_k^\tau} \tag{11.17}$$

where $x_k^\tau \in \mathbb{R}^{\tau m}$.

The initial state for the time delay is put to $x_0^\tau = 0$. Note that the state space model (11.16) and (11.17) is on observability canonical form.

11.2 Implementation of time delay

The state space model for the delay contains a huge number of zeroes and ones when the time delay is large, ie when the delay state space model dimension $m\tau$ is large.

In the continuous time we have that a delay is described exact by $y_{k+\tau} = y_k^-$. It can be shown that instead of simulating the state space model for the delay we can obtain the same by using a matrix (array or shift register) of size $n_\tau \times m$ where we use $n_\tau = \tau$ as an integer number of delay samples.

The above state space model for the delay contains $n_\tau = \tau$ state equations which may be expressed as

$$\begin{array}{lcl} x_k^1 & = & y_{k-1}^- \\ x_k^2 & = & x_{k-1}^1 \\ & \vdots & \\ x_k^{\tau-1} & = & x_{k-1}^{\tau-2} \\ y_k & = & x_{k-1}^{\tau-1} \end{array} \tag{11.18}$$

where we have used $y_k = x_k^\tau$. This may be implemented efficiently by using a matrix (or vector x. The following algorithm (or variants of it) may be used:

Algorithm 11.2.1 (Implementing time delay of a signal)
Given a vector $y_k^- \in \mathbb{R}^m$. A time delay of the elements in the vector y_k^- of n_τ time instants (samples) may simply be implemented by using a matrix x of size $n_\tau \times m$.

At each sample, k, (each call of the algorithm) do the following:

1. *Put y_k^- in the first row (at the top) of the matrix x.*
2. *Interchange each row (elements) in matrix one position down in the matrix.*
3. *The delayed output y_k is taken from the bottom element (last row) in the matrix x.*

```
$y_k = x(\tau, 1:m)^T$
for $i = \tau : -1 : 2$
    $x(i, 1:m) = x(i-1, 1:m)$
end
$x(1, 1:m) = (y_k^-)^T$
```

Note that this algorithm should be evaluated at each time instant k.
△

11.3 Examples

Example 11.1 *Delay $\tau = 2$ samplews*
Given a system where the undelayed output y_k^- is delayed $\tau = 2$ samples. The model for the undelayed part from input u_k to output y_k^- is given by

$$x_{k+1} = Ax_k + Bu_k, \tag{11.19}$$

$$y_k = Dx_k. \tag{11.20}$$

For the delay we define $\tau = 2$ dummy states

$$x_{k+1}^1 = y_k^-, \tag{11.21}$$

$$x_{k+1}^2 = x_k^1, \tag{11.22}$$

$$y_k = x_k^2. \tag{11.23}$$

Chapter 12

Adjustment of PID control parameters [1]

Abstract

Methods for the adjustment of PID control parameters, such that the loop transfer function, the sensitivity transfer function or the complementary sensitivity transfer function is tuned to a prescribed amplitude and phase, are presented.

Keywords: PID control parameters; automatic tuning.

12.1 Introduction

Many good methods for on-line identification of process parameters needed for tuning PID controllers are available.

A method for the adaption of PID controller parameters in closed loop systems has been presented in Balchen et al (1989). The method is based on an algorithm where the phase angle $\angle N(j\omega)$ is prescribed, and the corresponding frequency and amplitude of the sensitivity transfer function $N(j\omega)$ are estimated. The proportional gain is adjusted by an integral controller with setpoint equal to a prescribed amplitude of N. The integral and derivative times are adjusted proportionally to the inverse of the frequency ω.

A different scheme has been presented in Schei (1991), where the amplitude of the complementary sensitivity transfer function $M(j\omega)$ and the frequency of oscillation are estimated from a describing function approach. The phase angle is fixed to $\angle M(j\omega) = -90°$. The controller parameters are updated at each oscillation period in such a way that the amplitude of the loop transfer function $h_0(j\omega)$ attains a specified value. The integral and derivative times are adjusted in the same way as in Balchen et al (1989).

[1] This chapter is based on the paper
Di Ruscio (1992). Adjustment of PID control parameters. *Modeling, Identification and Control*, Vol. 13, No. 4, pp. 189-197.

However, the methods for the adjustment of the proportional gain in these works may be improved.

This note is concerned with the problem of adjusting the PID controller parameters in such a way that some process performance specifications are reached. Such specifications may be prescribed values on the amplitude and phase of the loop, sensitivity or the complementary sensitivity transfer function.

The paper is organized as follows: Section 2 presents the main results. Two examples are presented in Section 3 and some concluding remarks follow in Section 4.

12.2 Main Results

The main results in this section are stated in Subsection 2.1, 2.2 and 2.3. Subsection 2.1 deals with the loop transfer function $h_0(j\omega)$, Subsection 2.2 considers the sensitivity transfer function $N(j\omega)$ and Subsection 2.3 is considered with the complementary sensitivity transfer function $M(j\omega)$ for the adjustment of PID controller parameters. All methods are based on the idea that the integral time T_i, and the derivative time T_d are functions of the frequency, ie. $T_i = T_i(\omega)$ and $T_d = T_d(\omega)$. T_i and T_d are usually set proportional to ω^{-1}. The problem is then reduced to the adjustment of the proportional gain K_p.

12.2.1 Method 1

The following iteration formula for the adjustment of K_p such that the amplitude of the loop transfer function h_0 is tuned to a prescribed value $|h_0^s|$ is proposed.

$$K_p^{n+1} = K_p^n \frac{|h_0^s|}{|h_0^n|} = K_p^n + K_p^n \frac{|h_0^s| - |h_0^n|}{|h_0^n|} \tag{12.1}$$

where n is the index of iteration. The formula, Equation (12.1), has a quadratic convergence rate near the solution. This can be seen by using the Newton method for solving

$$\bar{f}(K_p) = |h_0^s| - |h_0| \tag{12.2}$$

to zero, which results in the following iteration scheme,

$$K_p^{n+1} = K_p^n - (\frac{\partial \bar{f}}{\partial K_p})_n^{-1} \bar{f}_n \tag{12.3}$$

where the gradient of $\bar{f}$ with respect to K_p is given by

$$\frac{\partial \bar{f}}{\partial K_p} = -\frac{\partial |h_0|}{\partial K_p} = -\frac{1}{K_p}|h_0| \tag{12.4}$$

Substituting Equation (12.4) into Equation (12.3) gives the iteration formula, Equation (12.1).

The adjustment formula, Equation (12.1), can be shown to be the same as the procedure for adjusting K_p described in an unclear and complicated way in Schei (1991).

12.2.2 Method 2

A method for the adjustment of K_p such that the amplitude of the sensitivity transfer function N is tuned to a prescribed value $|N_0|$ can be derived as follows.

An iteration formula for tuning

$$f(K_p) = |N_0| - |N| \tag{12.5}$$

to zero, is

$$K_p^{n+1} = K_p^n + (\frac{\partial |N|}{\partial K_p})_n^{-1} f_n \tag{12.6}$$

The gradient of the amplitude of N with respect to K_p can be shown to be

$$\frac{\partial |N|}{\partial K_p} = -\frac{1}{K_p}|N|(1 - \Re e(N)) \tag{12.7}$$

See the Appendix for a proof. Substituting Equation (12.7) into Equation (12.6) gives the iteration formula

$$K_p^{n+1} = K_p^n - K_p^n \frac{|N_0| - |N^n|}{|N^n|(1 - \Re e(N^n))} \tag{12.8}$$

A problem with this formula is that the gradient (12.7) may be zero in the frequency range considered. This difficulty is avoided by the following approximation.

$$K_p^{n+1} = K_p^n \frac{|N_0|}{|N^n|} = K_p^n + K_p^n \frac{|N_0| - |N^n|}{|N^n|} \tag{12.9}$$

This iteration scheme is found to be reasonable during simulations, and will hereafter be referred to as Method 2. Equation (12.9) is found from Equation (12.8) with $\Re e(N) = 2$. Other approximations, with different convergence properties, may be proposed by fixing $\Re e(N)$ to a constant near the singular point.

12.2.3 Method 3

A method for the adjustment of K_p such that the amplitude of the complementary sensitivity transfer function, M, is tuned to a prescribed value $|M_0|$ can be derived from the same procedure as Method 2. We have the iteration formula

$$K_p^{n+1} = K_p^n + K_p^n \frac{|M_0| - |M^n|}{|M^n|(1 - \Re e(M^n))} \tag{12.10}$$

where we have used that the gradient of the amplitude of M with respect to K_p can be shown to be

$$\frac{\partial |M|}{\partial K_p} = \frac{1}{K_p}|M|(1 - \Re e(M)) \tag{12.11}$$

and that the iteration formula, Equation (12.10), is a Newton method for tuning

$$\tilde{f}(K_p) = |M_0| - |M| \tag{12.12}$$

to zero. See the Appendix for a proof of Equation (12.11).

If the phase angle $\angle M(j\omega)$ is prescribed, $|M|$ and ω estimated, then the iteration formula, Equation (12.10), for adjusting K_p may be justified. Note that $\Re e(M) \approx 1$ only for $\omega \approx 0$ which means that the iteration formula, Equation (12.10), will have no singularities in the main frequency range.

Remark 1
All proposed methods for the adjustment of K_p may be modified to

$$K_p^{n+1} = K_p^n - \lambda_n (\frac{\partial \hat{f}}{\partial K_p})_n^{-1} \hat{f}_n \tag{12.13}$$

where $\hat{f}$ is the performance function to be tuned to zero. The rate of convergence of the proposed methods may be improved by choosing a suitable parameter λ_n.

Remark 2
Assume a PID controller with $T_i = T_i(\omega)$ and $T_d = T_d(\omega)$ and that $\angle N(\omega, K_p)$ and $|N(\omega, K_p)|$ are estimated or determined from a model. ω and K_p may then be determined directly by tuning the following set of equations to zero

$$f_1(\omega, K_p) = \angle N_0 - \angle N(\omega, K_p) \tag{12.14}$$
$$f_2(\omega, K_p) = |N_0| - |N(\omega, K_p)| \tag{12.15}$$

where $\angle N_0$ and $|N_0|$ are prescribed phase and amplitude of the sensitivity transfer function, respectively. A Newton method of the same form as Equation (12.13) may be used, ie.

$$\underline{\theta}_{n+1} = \underline{\theta}_n - \Lambda_n (\frac{\partial \underline{f}}{\partial \underline{\theta}})_n^{-1} \underline{f}_n \tag{12.16}$$

where $\underline{\theta} = \begin{bmatrix} \omega & K_p \end{bmatrix}^T$ and $\underline{f} = \begin{bmatrix} f_1 & f_2 \end{bmatrix}^T$. The matrix Λ_n is introduced to improve the convergence. The Jacobian matrix is given by

$$\frac{\partial \underline{f}}{\partial \underline{\theta}} = \begin{bmatrix} \frac{\partial f_1}{\partial \omega} & \frac{\partial f_1}{\partial K_p} \\ \frac{\partial f_2}{\partial \omega} & \frac{\partial f_2}{\partial K_p} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \angle N}{\partial \omega} & \frac{\partial \angle N}{\partial K_p} \\ \frac{\partial |N|}{\partial \omega} & \frac{\partial |N|}{\partial K_p} \end{bmatrix} \tag{12.17}$$

This iteration scheme is based on the total differential of $\angle N$ and $|N|$, ie.

$$d\angle N = \frac{\partial \angle N}{\partial \omega} d\omega + \frac{\partial \angle N}{\partial K_p} dK_p \tag{12.18}$$
$$d|N| = \frac{\partial |N|}{\partial \omega} d\omega + \frac{\partial |N|}{\partial K_p} dK_p \tag{12.19}$$

The partial derivatives of $\angle N$ and $|N|$ with respect to K_p may be determined analytically. $\frac{\partial |N|}{\partial K_p}$ is given by Equation (12.7). It can further be shown that

$$\frac{\partial \angle N}{\partial K_p} = \frac{1}{K_p} |N| \sin(\angle N) \tag{12.20}$$

A proof of Equation (12.20) is given in the Appendix. The partial derivatives of $\angle N$ and $|N|$ with respect to ω may be adjusted numerically during the iterations, or determined analytically if the process model is known.

12.3 Examples

12.3.1 Example 1

A PI controller is tuned for the process model

$$h_u = \frac{2}{(1+s)(1+4s)} \tag{12.21}$$

Assume that $|N|$ and ω are estimated, the phase angle $\angle N(j\omega)$ prescribed, and that the following parameters are specified.

$$|N_0| = 1.5, \quad \angle N_0 = 30^\circ, \quad T_i = \frac{4}{\omega} \tag{12.22}$$

Tuning results for Methods 2 and 3 are shown in Figure (12.1). The final results are $K_p = 1.94$, $T_i = 3.4$ and $\omega = 1.18$. For this example Method 1 did not converge.

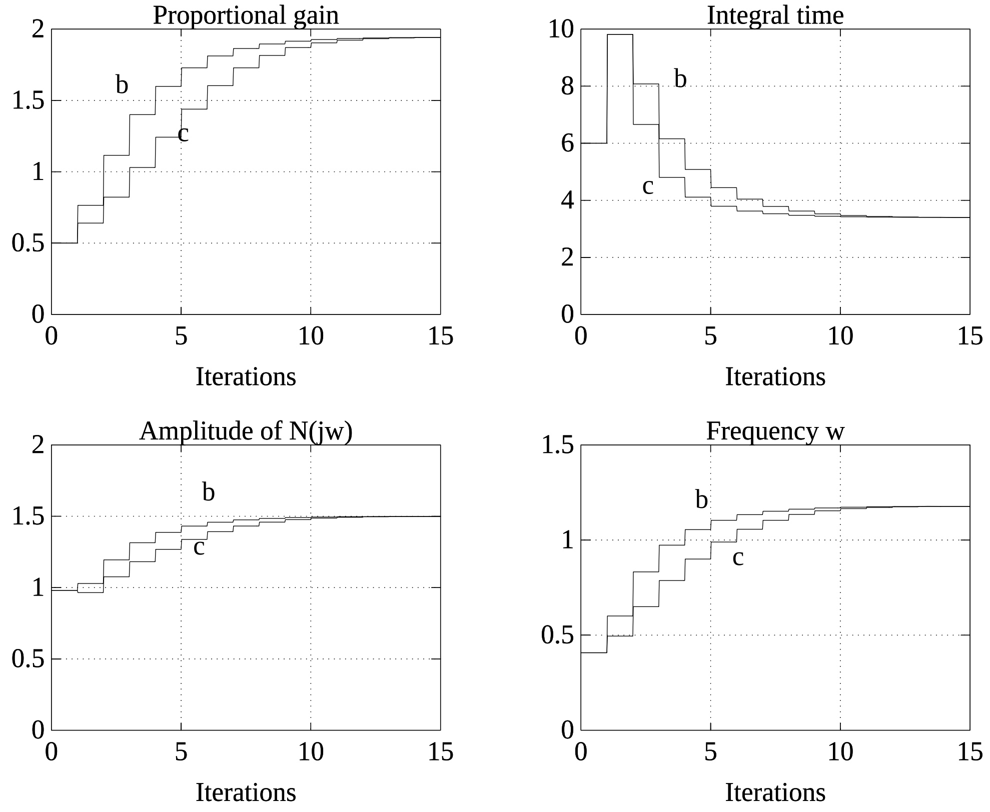

Figure 12.1: K_p, T_i, $|N|$ and w during tuning of the PI controller, for b, Method 2 and c, Method 3.

For this example, Figure (12.1) shows that Methods 2 and 3 have approximately equal rates of convergence. Now, assume that $|M|$ and ω are estimated, the above specifications are then identical to

$$|M_0| = 0.81, \quad \angle M_0 = -111.7^\circ, \quad T_i = \frac{4}{\omega} \tag{12.23}$$

Simulation results based on Method 3 for estimation of $|M|$ and Method 2 for estimation of $|N|$ are shown in Figure (12.2).

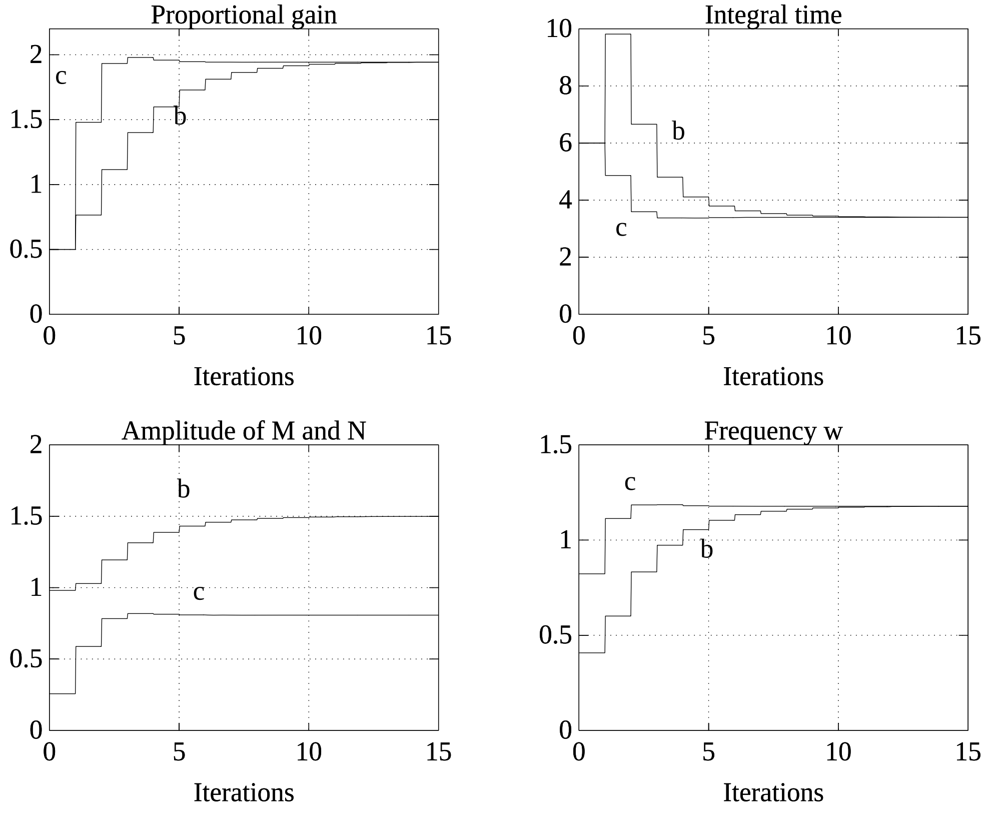

Figure 12.2: K_p, T_i, $|N|$, $|M|$ and w during tuning of the PI controller. b - Method 2 and c - Method 3.

Figure (12.2) shows that Method 3 based on estimation of $|M|$ has considerable faster rate of convergence compared to Method 2, based on estimation of $|N|$. However, the convergence may be improved by the use of the modified scheme, Equation (12.13), with a suitable parameter λ_n.

12.3.2 Example 2

A PI controller is tuned for the process model

$$h_u = \frac{2}{(1+s)(1+4s)}\ e^{-s} \tag{12.24}$$

Assume that $|N|$ and ω are estimated, the phase angle $\angle N(j\omega)$ prescribed, and that the following parameters are specified.

$$|N_0| = 2, \quad \angle N_0 = 0^\circ, \quad T_i = \frac{5}{\omega} \tag{12.25}$$

Tuning results for Methods 1, 2 and 3 are shown in Figure (12.3). All methods converged for this example. The final results are $K_p = 1.24$, $T_i = 5.52$ and $\omega = 0.91$.

Note that the rate of convergence is considerably faster than for Example 1. This example indicate faster convergence for processes with a time delay, because the frequency, ω, is nearly constant during the adaption of the controller parameters. Compare with Example 1.

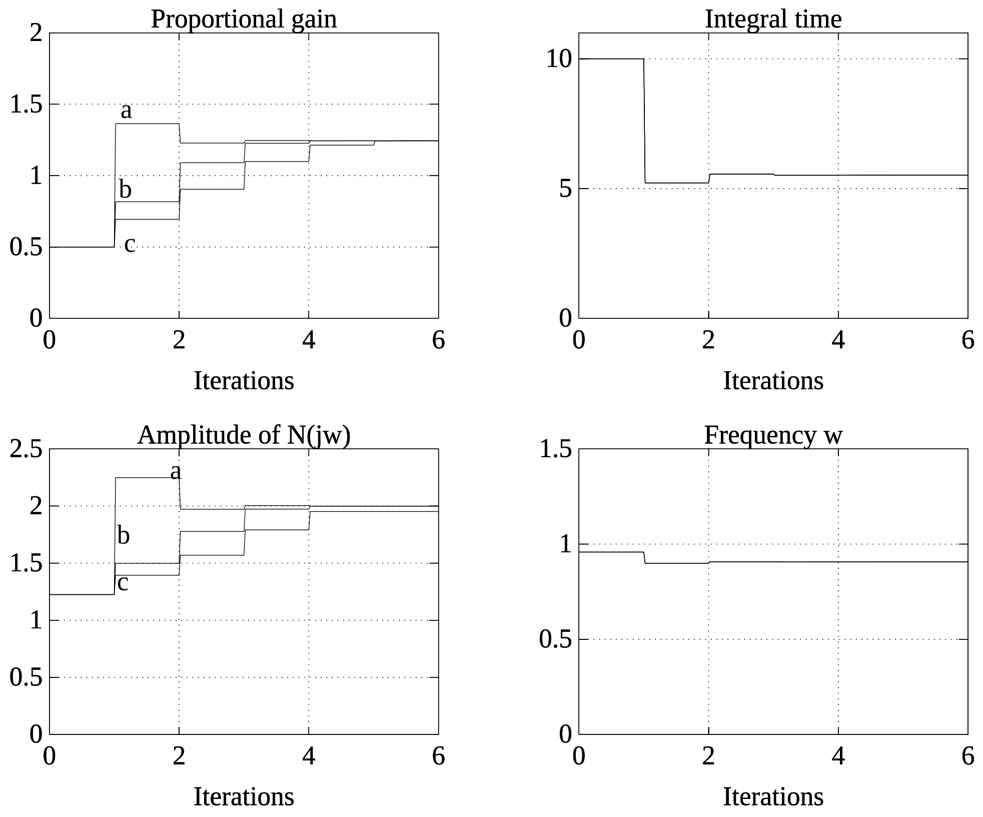

Figure 12.3: K_p, T_i, $|N|$ and w during tuning of the PI controller. a - Method 1, b - Method 2 and c - Method 3.

12.4 Concluding Remarks

Three methods based on gradient informations for the adjustment of the proportional gain in an PID controller are presented. The integral time constant, and the derivative time constant, are set proportional to the inverse of the frequency.

A method for the determination of PID parameters presented in an unclear and complicated way in Schei (1991) is shown to be a simple Newton method for the adjustment of K_p such that the amplitude of the loop transfer function $h_0(j\omega)$ is tuned to a specified value.

Simple analytical expressions for the gradient of the amplitude of the sensitivity transfer function $N(j\omega)$ and the complementary sensitivity transfer function $M(j\omega)$ with respect to the proportional gain K_p are presented. These expressions are the basis for two Newton methods for the adjustment of the proportional gain.

It is shown that the proportional gain should be adjusted from gradient informations of $|M|$ rather than gradient informations of $|N|$. This statement is justified from the fact that the gradient of $|N|$ with respect to K_p may be zero in the main frequency range, which means that the Newton method will be singular. However, an approximate iteration formula based on $|N|$, which may be used, is presented.

12.5 References

Åstrøm K. J., T. Hägglund (1984). Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins. *Automatica*, vol. 20, no. 5, pp. 645 - 651.

Balchen J. G., K. Telnes and D. Di Ruscio (1989). Frequency response adaptive control of a refrigeration cycle. *Modeling, Identification and Control*, vol. 10, no. 1, pp. 3 - 11.

Schei T. S. (1991). A New Method for Automatic Tuning of PID Control Parameters. *Proceedings 1th ECC - European Control Conference*, Grenoble, July 2-5, 1991, pp. 1522 - 1527.

12.6 Appendix

12.6.1 A proof of Equation (12.7)

The sensitivity transfer function is given by

$$N(j\omega) = \frac{1}{1 + h_0(j\omega)} \tag{12.26}$$

Define $h = |h_0(j\omega)|$, $\phi = \angle h_0(j\omega)$ and $h_0(j\omega) = he^{j\phi}$ to simplify the notation. We have

$$N(j\omega) = \frac{1}{1 + he^{j\phi}} = \frac{1 + he^{-j\phi}}{(1 + he^{j\phi})(1 + he^{-j\phi})} = \frac{1 + he^{-j\phi}}{1 + h(e^{j\phi} + e^{-j\phi}) + h^2} \tag{12.27}$$

which may be written

$$N(j\omega) = \frac{1 + hcos(-\phi) + jhsin(-\phi)}{1 + h^2 + 2hcos(\phi)} = \Re e(N) + j\Im m(N) \tag{12.28}$$

The amplitude of $N(j\omega)$ is given by

$$|N| = \frac{[(1 + hcos(-\phi))^2 + (hsin(-\phi))^2]^{\frac{1}{2}}}{1 + h^2 + 2hcos(\phi)} \tag{12.29}$$

or

$$|N| = \frac{1}{\sqrt{1 + h^2 + 2hcos(\phi)}} \tag{12.30}$$

The gradient of $|N(j\omega)|$ with respect to K_p is

$$\frac{\partial|N|}{\partial K_p}=-\frac{1}{2}\frac{1}{(1+h^2+2hcos(\phi))^{\frac{3}{2}}}[(2h+2cos(\phi))\frac{\partial h}{\partial K_p}-2hsin(\phi)\frac{\partial\phi}{\partial K_p}] \quad (12.31)$$

For a PID controller we have

$$\frac{\partial h}{\partial K_p}=\frac{1}{K_p}h \quad (12.32)$$

and

$$\frac{\partial\phi}{\partial K_p}=0 \quad (12.33)$$

when ω is constant. Substituting Equations (12.32), (12.33) and (12.30) into Equation (12.31) yields

$$\frac{\partial|N|}{\partial K_p}=-\frac{1}{K_p}|N|(1-\frac{1+hcos(\phi)}{1+h^2+2hcos(\phi)}) \quad (12.34)$$

which is seen to be equal to

$$\frac{\partial|N|}{\partial K_p}=-\frac{1}{K_p}|N|(1-|N|cos(\angle N)) \quad (12.35)$$

by the use of Equation (12.28), and Equation (12.7) is proved.

12.6.2 A proof of Equation (12.11)

The complementary sensitivity transfer function is given by

$$M(j\omega)=\frac{h_0(j\omega)}{1+h_0(j\omega)}=\frac{h^2+hcos(\phi)+jhsin(\phi)}{1+h^2+2hcos(\phi)}=\Re e(M)+j\Im m(M) \quad (12.36)$$

The amplitude of $M(j\omega)$ is

$$|M|=\frac{h}{\sqrt{1+h^2+2hcos(\phi)}} \quad (12.37)$$

The gradient of $|M(j\omega)|$ with respect to K_p is

$$\frac{\partial|M|}{\partial K_p}=\frac{\frac{\partial h}{\partial K_p}(1+h^2+2hcos(\phi))-h[(h+cos(\phi))\frac{\partial h}{\partial K_p}-hsin(\phi)\frac{\partial\phi}{\partial K_p}]}{(1+h^2+2hcos(\phi))^{\frac{3}{2}}} \quad (12.38)$$

which, by the use of Equations (12.32) and (12.33), may be shown to be

$$\frac{\partial|M|}{\partial K_p}=\frac{1}{K_p}|M|(1-\frac{h^2+hcos(\phi)}{1+h^2+2hcos(\phi)}) \quad (12.39)$$

or

$$\frac{\partial|M|}{\partial K_p}=\frac{1}{K_p}|M|(1-|M|cos(\angle M)) \quad (12.40)$$

and Equation (12.11) is proved.

Note that, $\Re e(N)+\Re e(M)=1$, which may be substituted into Equations (12.7) and (12.11) for the derivation of alternative expressions.

12.6.3 A proof of Equation (12.20)

From Equation (12.28) we have

$$\tan(\angle N) = \frac{h\sin(-\phi)}{1 + h\cos(-\phi)} \tag{12.41}$$

The partial derivative of the right hand side of Equation (12.41) with respect to K_p is given by

$$\frac{\partial \tan(\angle N)}{\partial K_p} = \frac{1}{K_p}|N|\frac{\tan(\angle N)}{\cos(\angle N)} \tag{12.42}$$

The chain rule on the left hand side of Equation (12.41) gives

$$\frac{\partial \tan(\angle N)}{\partial K_p} = \frac{1}{\cos^2(\angle N)}\frac{\partial \angle N}{\partial K_p} \tag{12.43}$$

Combining Equations (12.42) and (12.43) gives

$$\frac{\partial \angle N}{\partial K_p} = \frac{1}{K_p}|N|\sin(\angle N) \tag{12.44}$$

and Equation (12.20) is proved.

Part III

Model based control

Chapter 13

Modified Smith Predictor

13.1 Introduction

The Smith predictor is frequently used for controlling processes with long-dead times. Unfortunately, the Smith predictor cannot compensate for load disturbances upon an integrator processes with a long dead time.

When working with a course in process control the author constructed a modified Smith predictor for controlling an integrator process with long dead time. We will in this chapter discuss the Smith predictor and the necessary modifications in order to control an integrator processes with a long dead time.

13.2 The Smith Predictor

Assume that a process model transfer function $h_m(s)$ is available for the true process $h_p(s)$. The process model can be factorized into the product of a rational part (which can be inverted) $h_m^-(s)$ and a non-rational part $h_m^+(s)$ (non-invertible part, e.g. transport delay). We have

$$h_m(s) = h_m^-(s) h_m^+(s) \tag{13.1}$$

The Smith predictor is illustrated in Figure (13.1).

13.2.1 Transfer Function

Consider the Smith predictor in Figure (13.1). The transfer function model from the two inputs, i.e. the reference and disturbance signals, to the process output is

$$y(s) = h_r(s) r(s) + h_d(s) d(s), \tag{13.2}$$

where

$$h_d(s) = \frac{(1 + h_c(h_m^- - h_m))h_v}{1 + h_m^- h_c + (h_p - h_m)h_c}, \tag{13.3}$$

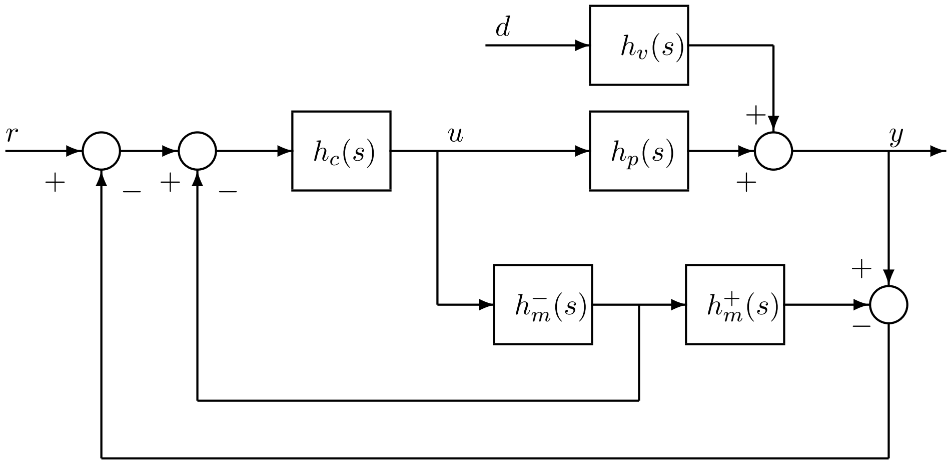

Figure 13.1: The Smith predictor for controlling processes with long dead times.

and

$$h_r(s) = \frac{h_c h_p}{1 + h_m^- h_c + (h_p - h_m)h_c}. \tag{13.4}$$

13.2.2 Process with integrator and time delay

Assume that the disturbance (load) transfer function $h_v(s)$ is equal to the process transfer function $h_p(s) = h_m(s)$. I.e. consider a process:

$$h_p(s) = h_m(s) = h_v(s) = \frac{1}{s}e^{-\tau s}. \tag{13.5}$$

The Smith predictor is constructed by putting:

$$h_m^-(s) = \frac{1}{s}, \tag{13.6}$$

$$h_m^+(s) = e^{-\tau s}. \tag{13.7}$$

The transfer function from the disturbance d to the process output is:

$$y(s) = \frac{(1 + h_c(h_m^- - h_m))h_v}{1 + h_m^- h_c} d(s) \tag{13.8}$$

Inserting gives

$$y(s) = \frac{(1 + h_c(\frac{1}{s} - \frac{1}{s}e^{-\tau s}))\frac{1}{s}e^{-\tau s}}{1 + \frac{1}{s}h_c} d(s) \tag{13.9}$$

series expansion of $e^{-\tau s}$ gives

$$y(s) = \frac{(1 + h_c(\frac{1}{s} - \frac{1}{s}(1 - \tau s + \frac{1}{2}\tau^2 s^2 + \cdots)))\frac{1}{s}e^{-\tau s}}{1 + \frac{1}{s}h_c} d(s) \tag{13.10}$$

which can be written as

$$y(s) = \frac{(1 + h_c(\tau - \frac{1}{2}\tau^2 s + \cdots))e^{-\tau s}}{s + h_c} d(s) \tag{13.11}$$

Assume a constant disturbance (load) step change with amplitude Δd at time zero. We have:

$$d(s) = \frac{1}{s}\Delta d \tag{13.12}$$

Using the final value theorem we find that:

$$\begin{aligned}\lim_{t\to\infty} y(t) &= \lim_{s\to 0} sy(s) = \lim_{s\to 0} \frac{(1 + h_c(\tau - \frac{1}{2}\tau^2 s + \cdots))e^{-\tau s}}{s + h_c}\Delta d && (13.13)\\ &= \lim_{s\to 0}(\frac{1}{h_c(s)} + \tau)\Delta d && (13.14)\end{aligned}$$

For a PI controller we have that

$$\lim_{s\to 0} \frac{1}{h_c(s)} = 0 \tag{13.15}$$

Using this we find that

$$\lim_{t\to\infty} y(t) = \tau\Delta d \tag{13.16}$$

From this we conclude that the off-set error is different from zero and that the error is large when the time delay is large. Hence, the Smith predictor cannot compensate for constant load changes when the process is a pure integrator with dead time.

13.3 Modified Smith predictor

The Smith predictor cannot reject load disturbances for processes with integration (process with integrator and time delay).

Consider a process with a pure integration with time delay, i.e.

$$h_p(s) = \frac{k_p}{s} e^{-\tau s}. \tag{13.17}$$

The process model is specified as:

$$h_m(s) = h_m^-(s) e^{-\tau s}. \tag{13.18}$$

Instead of putting $h_m^-(s) = \frac{k_p}{s}$ the rational part of the process model is specified to be

$$h_m^-(s) = \frac{k}{s + a}. \tag{13.19}$$

It can now be shown that

$$\lim_{t\to\infty} y(t) = 0 \tag{13.20}$$

for a constant load step change $v(s) = \Delta v/s$.

However, there is one drawback. The term $e^{-\tau s}$ will occur in the denominator term of the transfer functions $h_r(s)$ and $h_d(s)$ (the characteristic equation) when $h_m(s) \neq h_p(s)$.

13.4 Modified Smith Predictor by Åström

The modified Smith predictor by Åström *et al* is illustrated in Figure 13.2.

Consider in the following that the process gain is $k_p = 1$, see Figure 13.2. The set-point response is given by

$$H_r(s) = \frac{Y(s)}{R(s)} = \frac{k}{s+k}e^{-\tau s} \tag{13.21}$$

Note that the set-point response is independent of the choice of $M(s)$. The transfer function $M(s)$ will be discussed later. The set-point response is simply a delayed first order response. The controller gain k is chosen in order to shape the time constant $\frac{1}{k}$ of the 1.st order response. Note also that

$$\lim_{s\to 0} H_r(s) = 1 \tag{13.22}$$

Hence, we have zero steady state offset, i.e. $y = r$ in steady state.

The load response is given by

$$H_d(s) = \frac{Y(s)}{D(s)} = \frac{\frac{1}{s}e^{-\tau s}}{1 + M(s)\frac{1}{s}e^{-\tau s}} \tag{13.23}$$

The problem is now to shape the transfer function $M(s)$ so that $H_D(s)$ rejects (steady state) load disturbances, i.e.,

$$\lim_{s\to 0} H_d(s) = 0 \tag{13.24}$$

It is also of interest to chose $M(s)$ so that to remove the non-rational term $e^{-\tau s}$ from the denominator term of $H_D(s)$, in order to increase the bandwidth of the load response.

The following transfer function with three adjustable parameters, k_1, k_2 and k_3 is proposed:

$$M(s) = \frac{k_4 + \frac{1}{s}k_3}{1 + k_1 + \frac{1}{s}k_2 + \frac{1}{s^2}k_3 - (\frac{1}{s}k_4 + \frac{1}{s^2}k_3) - (\frac{1}{s}k_4 + \frac{1}{s^2}k_3)e^{-\tau s}} \tag{13.25}$$

where $k_4 = k_2 + k_r\tau$.

With this choice, the load (disturbance) response is given by

$$H_d(s) = \frac{e^{-\tau s}(s^2(1+k_1) + k_2 s + k_3 - (k_4 s + k_3)e^{-\tau s})}{s(s^2(1+k_1) + sk_2 + k_3)} \tag{13.26}$$

The denominator is a third order polynomial in s.

The modified Smith predictor in Figure 13.2 has some remarkable properties. It follows that the load response is decoupled from the set-point response. The signal $\hat{d}$ can be interpreted as an estimate of the (constant) input disturbance d. We have

$$\lim_{t\to\infty} \hat{d}(t) = d. \tag{13.27}$$

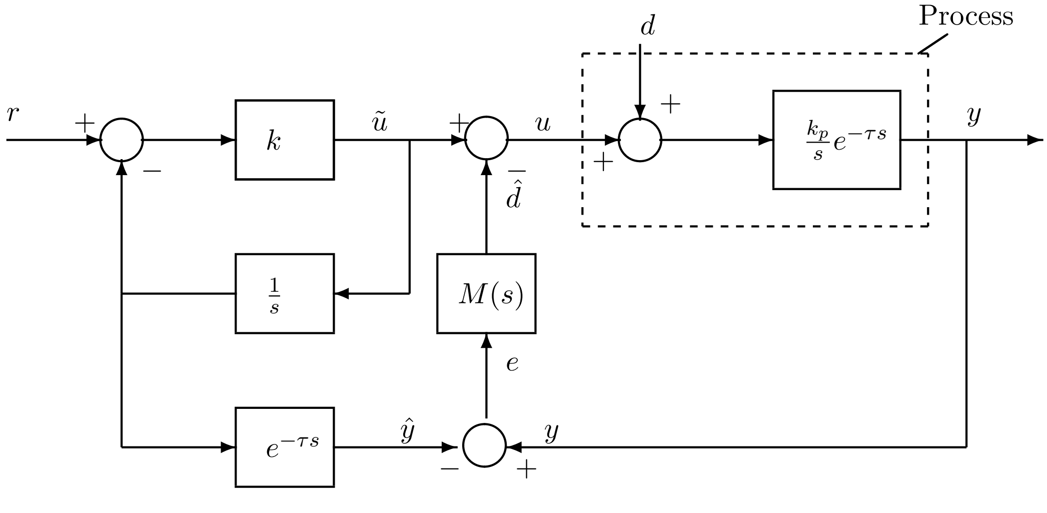

Figure 13.2: The Modified Smith predictor for controlling processes with an integrator and long dead time proposed by Åström *et al.* (1994).

13.4.1 A Model Based Point of View

The modified Smith predictor, Figure 13.2, have a feedback from the load (disturbance) estimate. It is of interest to compare this strategy with a model based control strategy. I.e. by using a state estimator. The states could be y and d where the load d is modeled by a pure integrator.

From Figure 13.2 we have that the estimate of y is taken as

$$\hat{y} = \frac{k}{s+k}e^{-\tau s}r \tag{13.28}$$

In the time domain we have

$$\dot{\hat{y}} = -k\hat{y} + kr \tag{13.29}$$

where we for simplicity has neglected the time delay.

Using a state estimator (Kalman filter) we would have used the following estimator for the output y.

$$\dot{\hat{y}} = u + \hat{d} + K_1(y - \hat{y}) \tag{13.30}$$

which can be written as

$$\dot{\hat{y}} = -K_1\hat{y} + u + \hat{d} + K_1 y \tag{13.31}$$

where K_1 is the Kalman filter gain. We want to compare the two estimators, Equations (13.29) (13.31). When are they similar ?

From Figure 13.2 we have that

$$\begin{aligned} \tilde{u} &= u + \hat{d} && (13.32)\\ \tilde{U}(s) &= \frac{sk}{s+k}R(s) && (13.33) \end{aligned}$$

Hence in steady state we have (the first point)

$$\lim_{t\to\infty} \tilde{u}(t) = \lim_{t\to\infty} (u(t) + \hat{d}(t)) = 0 \tag{13.34}$$

and from (13.22 (the second point)

$$\lim_{t\to\infty} y(t) = r \tag{13.35}$$

By replacing $u + \hat{d}$ and y in the state estimator (Kalman filter) Equation (13.31) with the steady state values (i.e., $u + \hat{d} = 0$ and $y = r$), we have

$$\dot{\hat{y}} = -K_1\hat{y} + K_1 r \tag{13.36}$$

Intuitively, the estimator for y used in the modified Smith predictor is similar to a state estimator (Kalman filter) updated from the steady state values of the inputs to the estimator.

Part IV

Control of multivariable systems

Chapter 14

Multivariable control

14.1 Interaction and pairing of variables

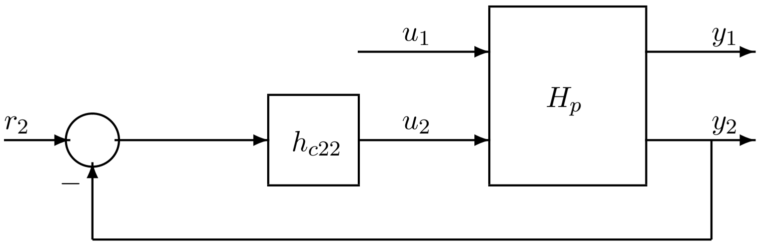

Figure 14.1: Multivariable control with one loop open.

We will in this section study the control of systems with two control input variables, u_1 and u_2 and two output measurements variables y_1 and y_2. Such a system is referred to as a (2×2) MIMO control system, where MIMO stands for Multiple Input and Multiple Output. Often only MIMO systems are used to denote such a control system.

The process described in Figure 14.1 is described by the transfer matrix model

$$y = H_p u \tag{14.1}$$

where H_p is a 2×2 transfer matrix. y and u are 2×1 vectors. We have

$$\overbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}^{y} = \overbrace{\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}}^{H_p} \overbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}^{u} \tag{14.2}$$

which is equivalent with

$$\begin{aligned} y_1 &= h_{11}u_1 + h_{12}u_2 && (14.3) \\ y_2 &= h_{21}u_1 + h_{22}u_2 && (14.4) \end{aligned}$$

When studying the transfer function from u_1 to y_1 we see that there is a cross coupling from u_2 represented by the transfer function h_{12}. If $h_{12} = 0$ then we see that there is no cross coupling and the relationship between u_1 and y_1 is decoupled from u_2. In the same way we see that there is a cross coupling from u_1 to y_2, represented with the transfer function h_{21}. I.e., u_1 is influencing the transfer function from u_2 to y_2.

For a 2 × 2 multivariable control system we have four possible pairings of inputs and outputs variables. Those possible pairings are defined as follows:

$$\begin{array}{lll} u_1 & \rightarrow & y_1 \quad (\text{using } u_1 \text{ for controlling } y_1) \\ u_2 & \rightarrow & y_1 \quad (\text{using } u_2 \text{ for controlling } y_1) \\ u_1 & \rightarrow & y_2 \quad (\text{using } u_1 \text{ for controlling } y_2) \\ u_2 & \rightarrow & y_2 \quad (\text{using } u_2 \text{ for controlling } y_2) \end{array} \tag{14.5}$$

For a 2 × 2 MIMO control system we have four different single input and single output control system strategies, which are defined as follows

$$\begin{array}{ll} u_1 = h_{c11}(r_1 - y_1) & (\text{loop } u_1 - y_1 \text{ closed}) \\ u_2 = h_{c22}(r_2 - y_2) & (\text{loop } u_2 - y_2 \text{ closed}) \\ u_1 = h_{c12}(r_2 - y_2) & (\text{loop } u_1 - y_2 \text{ closed}) \\ u_2 = h_{c21}(r_1 - y_1) & (\text{loop } u_2 - y_1 \text{ closed}) \end{array} \tag{14.6}$$

We will in this section study two problems of interests in connection with control of MIMO systems. These problems are defined below.

Problem 14.1 (Interaction)
Assume that we want to design a controller for the loop $u_1 - y_1$ in Figure 14.1. We have in this case two questions of interests. How will the interaction from the cross coupling in the process influence the transfer function from u_1 to y_1? How will the interaction from the controller h_{c22} for the closed loop system $u_2 - y_2$ influence the transfer function from u_1 to y_1.

Problem 14.2 (Pairing of input and output variables)
Assume that we want to use single loop controllers in order to control a MIMO process. We have one central question of interest. Which control input is to be used in order to control a particular output variable? We want to find out if it make sense to use u_1 or u_2 in order to control y_1, and if it make sense to use u_1 or u_2 in order to control y_2.

The solution to the pairing problem is therefore to chose the two best control strategies.

14.1.1 Closed loop $u_2 \rightarrow y_2$

We are using the system illustrated in Figure 14.1 as the starting point.

If u_2 is described by a feedback from y_2, i.e., $u_2 = h_{c22}(r_2 - y_2)$, then we have that the resulting transfer function from u_1 to y_1 will be given by

$$y_1 = h_{11}(1 - \frac{h_{12}h_{21}}{h_{11}h_{22} + h_{11}/h_{c22}})u_1 + \frac{h_{12}h_{c22}}{1 + h_{11}h_{c22}}r_2 \tag{14.7}$$

See Section 14.1.2 for proof. This can be written as

$$y_1 = h_{11}(1 - \Delta_{11})u_1 + \frac{h_{12}h_{c22}}{1 + h_{22}h_{c22}}r_2 \tag{14.8}$$

where Δ_{11} is given by

$$\Delta_{11}(s) = \frac{h_{12}h_{21}}{h_{11}h_{22}}T_{22}(s) \tag{14.9}$$

where $T_{22}(s)$ is given by

$$T_{22}(s) = \frac{h_{22}h_{c22}}{1 + h_{22}h_{c22}} \tag{14.10}$$

$T_{22}(s)$ is the complementary sensitivity function for the closed loop $u_2 - y_2$ if there was no cross coupling, i.e. if $h_{21} = 0$. See Equation (14.18) below for a proof.

The transfer function Δ_{11} represent the interaction (from cross couplings in the system and the controller h_{c22}) on the transfer function from u_1 to y_1. Δ_{11} describes the interaction from the closed loop and the interaction from the closed loop in the process. As we see, the resulting transfer function from u_1 to y_1 is dependent on the controller h_{c22} in the closed loop $u_2 - y_2$ as well as the cross couplings h_{12} and h_{21} in the process.

If Δ_{11} is small then the interaction is small. We also see that there is no cross coupling if $h_{12} = 0$ and/or $h_{21} = 0$, hence in this case $\Delta_{11} = 0$.

We have shown that Δ_{11} which describes the interaction is dependent on the controller h_{c22}.

Assume now that we want to close the loop $u_1 - y_1$ with a controller $u_1 = h_{c11}(r_1 - y_1)$. The term Δ_{11} shows that the two controllers not should be controlled independent of each other when the term $h_{12}h_{21}/(h_{11}h_{22})$, which represents cross couplings in the process, is large.

Consider now the case in which the controller h_{c22} have integral action, i.e. the case when h_{c22} e.g. is a PI or a PID controller. We can in this case show that Δ_{11} is independent of the controller h_{c22} at low frequencies. We have that $T_{22}(s = 0) = 1$. This means that

$$\Delta_{11}(s = 0) = \frac{h_{12}h_{21}}{h_{11}h_{22}}(s = 0) \tag{14.11}$$

Often it is sufficient to study $\Delta_{11}(s = 0)$. The magnitude $\Delta_{11}(s = 0)$ describes if it make sense to use u_1 in order to control y_1. It is clear that $u_1 - y_1$ is a good pairing if $\Delta_{11}(s = 0)$ is small, i.e. when $\Delta_{11}(s = 0) \approx 0$.

If $T_{22}(s) \approx 1$ then we have the following approximation

$$\Delta_{11}(s) \approx \frac{h_{12}h_{21}}{h_{11}h_{22}} \tag{14.12}$$

An advantage with this approximation is that it is independent of the controller h_{c22}. If the model is known, then we can plot $\Delta(s)$ as a function of the frequency ω where $s = j\omega$, e.g. in a Bode diagram. Hence, $u_1 - y_1$ will be a good pairing strategy if the magnitude of $\Delta_{11}(s)$ is small.

As we see from Equation (14.8) the reference r_2 for the closed loop $u_2 - y_2$ influences upon the output y_1 via the transfer function $h_{12}h_{c22}/(1 + h_{22}h_{c22})$. This is a phenomena which often is omitted when studying single loop control of MIMO systems. The contribution from r_2 is influencing like a disturbance on the process output y_1. This contribution may be important if h_{12} is large.

14.1.2 Proof of Equation (14.7)

We will in this section derive the results described in Section 14.1.1. The starting point is Figure 14.1 with loop $u_1 - y_1$ open and loop $u_2 - y_2$ closed. The process is described by

$$\begin{aligned} y_1 &= h_{11}u_1 + h_{12}u_2, && (14.13)\\ y_2 &= h_{21}u_1 + h_{22}u_2, && (14.14) \end{aligned}$$

with the closed loop $u_2 - y_2$ described by the feedback

$$u_2 = h_{c22}(r_2 - y_2), \tag{14.15}$$

where h_{c22} is the controller transfer function.

If we are putting Equation (14.15) into Equations (14.13) and (14.14) then we get the following description of the total system with loop $u_2 - y_2$ closed and loop $u_1 - y_1$ open,

$$\begin{aligned} y_1 &= h_{11}u_1 + h_{12}h_{c22}(r_2 - y_2), && (14.16)\\ y_2 &= h_{21}u_1 + h_{22}h_{c22}(r_2 - y_2). && (14.17) \end{aligned}$$

We know get the transfer function for the closed loop from Equation (14.17), i.e.,

$$y_2 = \frac{h_{21}}{1 + h_{22}h_{c22}}u_1 + \overbrace{\frac{h_{22}h_{c22}}{1 + h_{22}h_{c22}}}^{T_{22}} r_2. \tag{14.18}$$

The resulting transfer function from u_1 to y_1 is obtained by putting Equation (14.18) into (14.16). This gives

$$y_1 = h_{11}u_1 + h_{12}h_{c22}(r_2 - \frac{h_{21}}{1 + h_{22}h_{c22}}u_1 - \frac{h_{22}h_{c22}}{1 + h_{22}h_{c22}}r_2), \tag{14.19}$$

which gives

$$y_1 = (h_{11} - \frac{h_{12}h_{21}h_{c22}}{1 + h_{22}h_{c22}})u_1 + \frac{h_{12}h_{c22}}{1 + h_{22}h_{c22}}r_2. \tag{14.20}$$

Equation (14.20) can be described as

$$y_1 = h_{11}(1 - \frac{h_{12}h_{21}}{h_{11}h_{22}}T_{22}(s))u_1 + \frac{h_{12}h_{c22}}{1 + h_{22}h_{c22}}r_2, \tag{14.21}$$

where $T_{22}(s)$ is the complementary sensitivity function for the closed loop $u_2 - y_2$, i.e.,

$$T_{22}(s) = \frac{h_{22}h_{c22}}{1 + h_{22}h_{c22}}. \tag{14.22}$$

14.1.3 Rules for pairing variables

The system in Figure 14.1 is to be controlled by two single loop controllers. The four possible control strategies is as shown in 14.6. In the same way as the development in Section 14.1.2 and the result in Section 14.1.1 we can develop the following expressions, Δ_{11} is represented for the sake of completeness.

$$\Delta_{11}(s) = \frac{h_{12}h_{21}}{h_{11}h_{22}} \overbrace{\frac{h_{22}h_{c22}}{1 + h_{22}h_{c22}}}^{T_{22}}, \qquad \Delta_{11}(s) \approx \frac{h_{12}h_{21}}{h_{11}h_{22}}. \tag{14.23}$$

$$\Delta_{22}(s) = \frac{h_{21}h_{12}}{h_{22}h_{11}} \overbrace{\frac{h_{11}h_{c11}}{1 + h_{11}h_{c11}}}^{T_{11}}, \qquad \Delta_{22}(s) \approx \frac{h_{12}h_{21}}{h_{11}h_{22}}. \tag{14.24}$$

$$\Delta_{21}(s) = \frac{h_{22}h_{11}}{h_{21}h_{12}} \overbrace{\frac{h_{12}h_{c21}}{1 + h_{12}h_{c21}}}^{T_{21}}, \qquad \Delta_{21}(s) \approx \frac{h_{22}h_{11}}{h_{21}h_{12}}. \tag{14.25}$$

$$\Delta_{12}(s) = \frac{h_{11}h_{22}}{h_{12}h_{21}} \overbrace{\frac{h_{21}h_{c12}}{1 + h_{21}h_{c12}}}^{T_{12}}, \qquad \Delta_{12}(s) \approx \frac{h_{11}h_{22}}{h_{12}h_{21}}. \tag{14.26}$$

We have here assumed that the controllers $h_{c,i,j}$ have integral action. This means that $T_{i,j}(s = 0) = 1$ and that $T_{i,j}(s) \approx 1$ at low frequencies. This is the background for the above approximations. Note that $\Delta_{11}(s = 0) = \Delta_{22}(s = 0)$ and that $\Delta_{21}(s = 0) = \Delta_{12}(s = 0)$.

We form the following matrix of interactions

$$\Delta(s) = \begin{bmatrix} \Delta_{11}(s) & \Delta_{12}(s) \\ \Delta_{21}(s) & \Delta_{22}(s) \end{bmatrix} \tag{14.27}$$

Foe $s = 0$ we have the following steady state interaction matrix, $\Delta(s = 0)$, i.e.,

$$\Delta(s = 0) = \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix} = \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{12} & \Delta_{11} \end{bmatrix} \tag{14.28}$$

This last expression is often valid approximately also in the dynamic case, i.e. if $1/h_{c,i,j} \ll 1$), and at low frequencies when $s \approx 0$. From the elements in the interaction matrix Δ we can find the pairings which make sense

It make sense to write down the following table

$$\begin{array}{c|cc} & u_1 & u_2 \\ \hline y_1 & \Delta_{11} & \Delta_{12} \\ y_2 & \Delta_{21} & \Delta_{22} \end{array} \tag{14.29}$$

We are now assuming that the transfer function $T_{i,j}(s) \approx 1$ and we therefore are taking the approximative expressions for $\Delta_{ij}(s)$ as the starting point. We have the following rules for pairing of variables in single loop control of MIMO systems, Balchen (1988).

Regel 14.1 (Pairing of variables based on the Δ interaction matrix)

1. If $\mid \Delta_{i,j} \mid \approx 0$ then there is small interaction. It is therefore favourable to control y_i with u_j.
2. If $0 \leq \mid \Delta_{i,j} \mid \leq 0.8$ then there is some interaction. Then there is acceptable to control y_i with u_j.
3. If $0.8 \leq \mid \Delta_{i,j} \mid \leq 1$ then there is a strong interaction. $u_j - y_i$ is therefore an undesired pairing.
4. If $\mid \Delta_{i,j} \mid \geq 1$ then the pairing $u_j - y_i$ should not be used.

Remark 14.1 The above rules for pairing of input and output variables is generally valid when the elements $\Delta_{ij}(s)$ varies with s, i.e. as a function of the frequency ω where $s = j\omega$. A good approximation is that the transfer functions $T_{i,j}(s = j\omega) \approx 1$ for frequencies less than the bandwidth of the system. For a 2×2 system it therefore is sufficient to study the magnitude for

$$\Delta_{11}(s) = \Delta_{22}(s) = \frac{h_{12}h_{21}}{h_{11}h_{22}} \tag{14.30}$$

and

$$\Delta_{21}(s) = \Delta_{12}(s) = \frac{h_{11}h_{22}}{h_{12}h_{21}} \tag{14.31}$$

Remark 14.2 One should always first study Δ under stationary behavior, i.e. first put $s = 0$. Such an analysis is sufficient for most systems. Loosely spoken, this is the case for systems which is easy to control. For a 2×2 system it normally holds to study

$$\Delta_{11}(0) = \Delta_{22}(0) = \frac{h_{12}h_{21}}{h_{11}h_{22}}(s = 0) \tag{14.32}$$

and

$$\Delta_{21}(0) = \Delta_{12}(0) = \frac{h_{11}h_{22}}{h_{12}h_{21}}(s = 0) \tag{14.33}$$

Remark 14.3 The above rules for defining the pairings of input and output variables is valid also for $m \times m$ MIMO systems, i.e. for systems with m inputs and m outputs. We are referring to Balchen (1988) for a description.

Example 14.1 (Control of a distillation column)
A linearized steady state model for a distillation column is given as follows

$$\overbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}^{y} = \overbrace{\begin{bmatrix} 0.9033 & -0.9137 \\ 0.9366 & -0.9262 \end{bmatrix}}^{H_p} \overbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}^{u} \tag{14.34}$$

where

$$\begin{array}{lcl} y_1 & - & \textit{composition of bottom product } [\cdot] \\ y_2 & - & \textit{composition of top product } [\cdot] \\ u_1 & - & \textit{mass flow of reflux to the top of the column } [kg/s] \\ u_2 & - & \textit{mass flow of steam from the reboiler } [kg/s] \end{array} \tag{14.35}$$

From the elements $h_{i,j}$ in the steady state transfer matrix $H_p(s = 0)$ we have that

$$\Delta_{11}(0) = \Delta_{22}(0) = \frac{h_{12}h_{21}}{h_{11}h_{22}}(s = 0) \approx 1.02 \tag{14.36}$$

and

$$\Delta_{21}(0) = \Delta_{12}(0) = \frac{h_{11}h_{22}}{h_{12}h_{21}}(s = 0) \approx 0.98 \tag{14.37}$$

We are now forming the following table from the interaction matrix Δ, i.e.,

$$\begin{array}{c|cc} & u_1 & u_2 \\ \hline y_1 & 1.02 & 0.98 \\ y_2 & 0.98 & 1.02 \end{array} \tag{14.38}$$

This means that the following pairings should be chosen

$$\begin{array}{lcll} u_1 & \rightarrow & y_2 & \textit{(using } u_1 \textit{ for controlling } y_2\textit{)} \\ u_2 & \rightarrow & y_1 & \textit{(using } u_2 \textit{ for controlling } y_1\textit{)} \end{array} \tag{14.39}$$

remark that the interaction indices $\Delta_{12} = \Delta_{21} = 0.98$, which gives us the pairings $u_1 \rightarrow y_2$ and $u_2 \rightarrow y_1$, is very close to one. This is typically for distillation columns. A distillation column is an example of a process with large interactions and strong cross couplings between the input and output variables.

14.2 Relative Gain Array (RGA)

the starting point is a linear system described by the transfer matrix model

$$y = H_p u \tag{14.40}$$

where y is a $m \times 1$ vector, u is a $r \times 1$ vector and H_p is a $m \times r$ transfer matrix. We will in the following assume that that there is an equal number of input and output variables, i.e. we assume that $m = r$. This gives a square transfer matrix H_p.

The RGA matrix Λ is defined simply as

$$\Lambda = H_p \times (H_p^{-1})^T \tag{14.41}$$

where × denotes element by element multiplication. We have assumed that H_p is non singular. For a 2 × 2 we have that

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} \\ \lambda_{21} & \lambda_{22} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 1-\lambda_{11} \\ 1-\lambda_{11} & \lambda_{11} \end{bmatrix} \quad \text{where} \quad \lambda_{11} = \frac{1}{1-\frac{h_{12}h_{21}}{h_{11}h_{22}}} \tag{14.42}$$

Regel 14.2 (Pairing of variables based on RGA)
The starting point is the elements λ_{ij} in the RGA matrix Λ.

1. Chose the pairing $u_j \to y_i$ for which the corresponding RGA element λ_{ij} is positive and so close to 1 in magnitude as possible.
2. The pairing $u_j \to y_i$ must be avoided if the RGA element is negative, i.e. when $\lambda_{ij} < 0$.

The RGA matrix have a number of properties. First we note that Λ is a symmetric matrix. It can also be shown that the elements in each row or column in Λ sums to 1, i.e.

$$\sum_{j=1}^{m} \lambda_{ij} = 1 \text{ for } i = 1, \cdots, m. \tag{14.43}$$

The reason for that one should not use pairings which coincides with negative RGA elements is that such a pairing strategy is *structural unstable*. There may exists stable control structures which corresponds to negative RGA elements. However, the total system may be unstable if one of the control loops are broken ore put in manual. Such a control system is defined to be *structural unstable*.

It is of importance to denote this difference of a *structural unstable* control system and a *unstable* control system, in connection with pairing of variables by using RGA analysis ore Balchen index analysis, for single loop control of MIMO systems.

Example 14.2 (Controlling a distillation column)
We are taking Example 14.1 as the starting point. The following steady state model for the distillation column are given, i.e.,

$$\overbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}}^{y} = \overbrace{\begin{bmatrix} 0.9033 & -0.9137 \\ 0.9366 & -0.9262 \end{bmatrix}}^{H_p} \overbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}^{u} \tag{14.44}$$

The RGA matrix is then given by

$$\Lambda = H_p \times (H_p^{-1})^T = \begin{bmatrix} -43.72 & 44.72 \\ 44.72 & -43.72 \end{bmatrix} \tag{14.45}$$

The correct pairing strategy is therefore $u_1 \to y_2$ and $u_2 \to y_1$. The corresponding RGA elements is positive for those pairings, i.e. $\lambda_{12} = \lambda_{21} = 44.72$.

14.2.1 Relationship between RGA and Δ

Consider a 2×2 system. In this case we have the following relationship between the elements in the RGA matrix and the elements in the Δ matrix.

$$\lambda_{11} = \frac{1}{1-\Delta_{11}}, \;\; \Delta_{11} = \frac{h_{12}h_{21}}{h_{11}h_{22}} \tag{14.46}$$

$$\lambda_{12} = \frac{1}{1-\Delta_{12}}, \;\; \Delta_{12} = \frac{h_{11}h_{22}}{h_{12}h_{21}} \tag{14.47}$$

$$\lambda_{21} = \frac{1}{1-\Delta_{21}}, \;\; \Delta_{21} = \frac{h_{22}h_{11}}{h_{21}h_{12}} \tag{14.48}$$

$$\lambda_{22} = \frac{1}{1-\Delta_{22}}, \;\; \Delta_{22} = \frac{h_{12}h_{21}}{h_{11}h_{22}} \tag{14.49}$$

Note that $\lambda_{11} = \lambda_{22}$, $\lambda_{12} = \lambda_{21}$, $\Delta_{11} = \Delta_{22}$ and $\Delta_{12} = \Delta_{21}$. The pairing $u_j \rightarrow y_i$ should be avoided if $\lambda_{ij} < 0$. The similar should be avoided when $\Delta_{ij} > 1$.

Chapter 15

Multiple inputs and outputs and control structures

15.1 Split Range Control

We will in this section discuss the case in which we have multiple inputs (manipulated variables, control inputs) and only one output (one measurement) to be controlled. In this case we have a redundant number of input manipulable variables.

A common situation is a standard feedback system of a SISO system with one input u_1 and one output y and where the input (and/or the output) becomes saturated. This means e.g. that the input is bounded as $u_1^{\text{min}} \leq u_1 \leq u_1^{\text{max}}$. In this situation it may be useful and/or necessary to add an extra input u_2 (or more inputs) in order to overcome the saturation and in order to achieve the goal $y = r$ where r is a desired specified reference for the output y. A strategy could then be to use u_1 for control when $y \in [y_{\mathbf{min}}, y_1]$ and a second manipulable input u_2 when $y \in [y_1, y_{\mathbf{max}}]$

In the *split-range control* strategy there are multiple controllers which acts in parallel rather than in series as in the *cascade controller* strategy.

If a model of the MIMO system is available, an optimal controller strategy, with e.g. integral action, is believed to be very useful in this situation.

References

Balchen, J. G. (1984). *Reguleringsteknikk*, Bind 1, Tapir.

Balchen, J. G. and K. I. Mumme (1988). *Process Control, Structures and Applications.* Van Nostrand Reinhold Company Inc., New York.

Balchen, J. G. (1990). Rational transfer function approximations to transport delay. *Modeling, Identification and Control*, Vol. 11, No. 3, pp. 127-140.

Di Ruscio, D. (1992). Adjustment of PID control parameters. *Modeling, Identification and Control*, Vol. 13, No. 4.

Kailath, T. (1980). *Linear Systems*, Prentice-Hall, Inc.

Mataušek, M. R and A. D. Micić (1996). A modified Smith predictor for controlling a process with an integrator and long dead time. *IEEE Transactions on Automatic Control*, Vol. 41, No. 8.

Parlet, B. N. (1976). A Recurrence among the Elements of Functions of Triangular Matrices. *Linear Algebra and its Applications*, Vol. 14, pp. 117-121.

Petkov P. H., N. D. Christov and M. M. Konstantinov (1991). *Computational Methods for Linear Control Systems.* Prentice Hall International (UK) Ltd.

Skogestad, S. and I. Postlethwaite (1996). *Multivariable Feedback Control*, Wiley.

Åström, K. J., Hang, C. C. and B. C. Lim (1994). A new Smith predictor for controlling a process with an integrator and long dead time. *IEEE Transactions on Automatic Control*, Vol. 39, No. 2.

Appendix A

Laplace Transformations

In practice we need only a few basic rules about Laplace transformations and the relationship to differential equations. Some of these will be described in this appendix.

A.1 Laplace transformation of the exponential

Consider the 1st order autonomous scalar differential equation

$$\dot{x} = -ax, \quad x(t=0) = x_0 \tag{A.1}$$

where we assume that $a \geq 0$ is a positive scalar number. The solution to Eq. (A.1) is

$$x(t) = e^{-at}x_0, \quad x(t=0) = x_0 \tag{A.2}$$

From the theory of Laplace transformations we have that

$$L(\dot{x}) = sx - x(t=0) = sx - x_0 \tag{A.3}$$

Then the Laplace transformation of Eq. (A.1) yields

$$x(s) = \frac{1}{s+a}x_0. \tag{A.4}$$

Using the property of linearity of Laplace transformations

$$L(ax(t) + by(t)) = aL(x) + bL(y) = ax(s) + by(s), \tag{A.5}$$

we find from the laplace transformation of the solution Eq. (A.2) and Eq. (A.4) that the Laplace transformation of the exponential is given by

$$L(e^{-at}) = \frac{1}{s+a}. \tag{A.6}$$

A.2 Laplace transformation of triogonometric functions

Consider the Laplace transformation of the exponential of a complex number. Using Eq. (A.6) we obtain

$$L(e^{j\omega t}) = \frac{1}{s - j\omega} = \frac{s + j\omega}{s^2 + \omega^2} = \frac{s}{s^2 + \omega^2} + j\frac{\omega}{s^2 + \omega^2}. \tag{A.7}$$

Using and comparing with the Laplace transformation of the Euler formula

$$e^{j\omega t} = \sin(\omega t) + j\cos(\omega t), \tag{A.8}$$

we find that

$$L(\sin(\omega t)) = \frac{s}{s^2 + \omega^2}, \tag{A.9}$$

$$L(\cos(\omega t)) = \frac{\omega}{s^2 + \omega^2}. \tag{A.10}$$

Using the identity

$$\sin(\omega t + \phi) = \sin(\omega t)\cos(\phi) + \cos(\omega t)\sin(\phi), \tag{A.11}$$

gives

$$L(\sin(\omega t + \phi)) = \frac{\cos(\phi)s + \omega\sin(\phi)}{s^2 + \omega^2}. \tag{A.12}$$