John Bagterp Jørgensen

Constrained Predictive Control

A Computational Approach

September 2, 2015

Springer
Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Contents

Part I

Modeling and Simulation

1

The Quadruple Tank Process

In this chapter, we present the quadruple tank process and develop a mathematical model that describes its dynamics. We illustrate the modeling process and discuss deterministic as well stochastic simulation using Matlab.

This chapter enables you to model simple processes using differential equations, simulate these models using Matlab, and develop linearized deterministic and stochastic models.

1.1 The Quadruple Tank Process

The quadruple tank process is illustrated schematically in Figure 1.1. It consist of four interconnected tanks. The water flow can be distributed in different ways depending on the valve positions.

Flow rate F_1 and flow rate F_2 can be manipulated in the configuration illustrated in Figure 1.1 . These variables are therefore called the manipulated variables, u, or MVs. The process is equipped with sensors measuring the level h_i in each tank $i \in \{1, 2, 3, 4\}$. The measured variables, y, are the combination of actual values of the heights and sensor noise. We do assume that we want to control the levels in tank 1 and tank 2, i.e. we want to control h_1 and h_2. In this case h_1 and h_2 are called the controlled variables or the CVs. The CVs are also denoted z and is typical a subset of the measured variables, y.

Let u denote the manipulated variables (MVs). Let y denote the measured variables and let z denote the controlled variables (CVs). Then in the case of perfect measurements without noise

$$u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \qquad y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} \qquad z = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

The generic input-output relations for this process is illustrated in Figure 1.2. It consists of manipulated inputs, u, measured outputs, y, and controlled

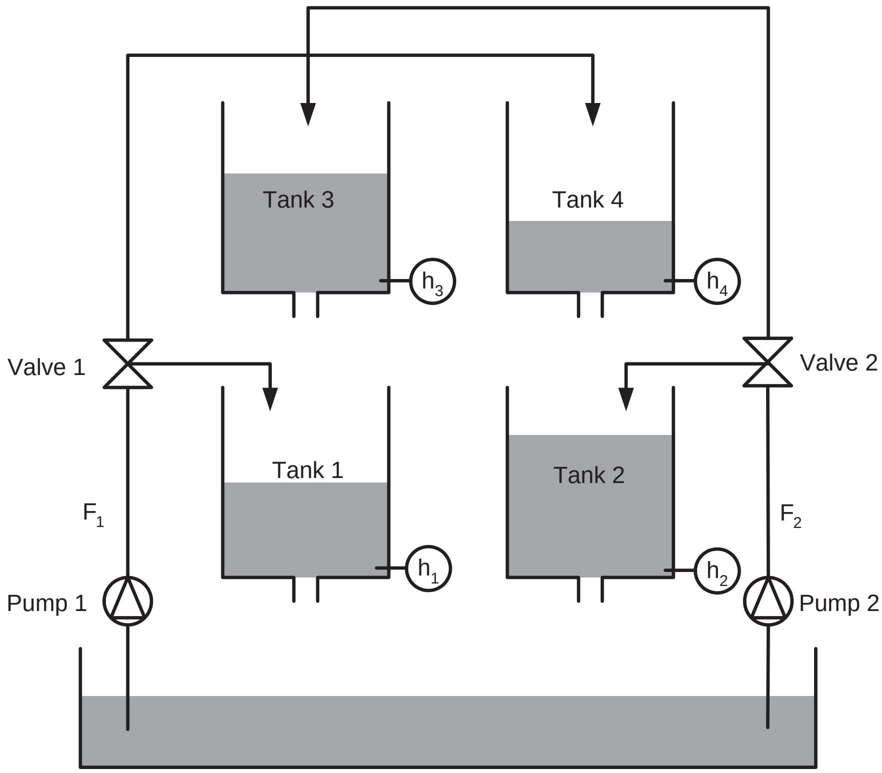

Fig. 1.1. Schematic diagram of the quadruple tank process.

variables, z. Let the states of the process be denoted x. As processes evolve in continuous time, the process is modeled using differential equations. The four tank system and most of the systems we will consider, can be modeled using systems of ordinary differential equations. Hence for the deterministic formulation without measurement noise and process noise, the block diagram in Figure 1.2 is a convenient illustration of the system of equations

$$\frac{dx(t)}{dt} = f(x(t), u(t)) \quad x(t_0) = x_0 \tag{1.1a}$$

$$y(t) = g(x(t)) \tag{1.1b}$$

$$z(t) = h(x(t)) \tag{1.1c}$$

$t \in \mathbb{R}$ is time, $x \in \mathbb{R}^{n_x}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the vector of manipulated variables, $y \in \mathbb{R}^{n_y}$ is the measured (observed) values, and $z \in \mathbb{R}^{n_z}$ is the outputs. $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \mapsto \mathbb{R}^{n_x}$ is the model of the system, $g : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ is the sensor function, and $h : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_z}$ is the output function. $\frac{dx(t)}{dt}$ is sometimes denoted $\dot{x}(t)$, i.e. $\frac{dx(t)}{dt} = \dot{x}(t)$. Most process systems are *autonomous*,

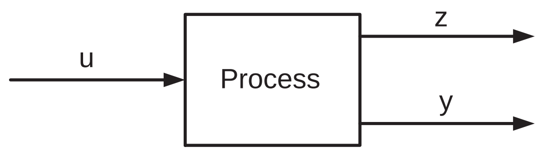

Fig. 1.2. Generic input-output model.

i.e. $f(x(t), u(t))$ in does not depend directly on time, t. *Non-autonomous* systems depends directly on time, i.e. they have a process model of the form $f(t, x(t), u(t))$.

1.1.1 Modeling a Single Tank

The four tank system may be modeled using mass balances of water in each tank. The modeling will be based on the diagram and the symbols in Figure 1.3.

The mass balance is a fundamental principle saying that mass is conserved. The mass balance of non-reactive systems, i.e. systems in which there is no chemical reaction, the mass balance takes the form

$$\text{Accumulated} = \text{In - Out}$$

Consider the time interval, $[t, t + \Delta t]$, with Δt sufficiently small such that the flow rates to and from the tank can be considered constant in this small interval. This is an approximation that we will subsequently vanish when we let $\Delta t \to 0$.

Let $m_1(t)$ [g] be the mass of water in tank 1, ρ [g/cm^3] be the density of water, $q_{1,in}$ [cm^3/s] the volumetric flow rate into tank 1 from valve 1, q_3 [cm^3/s] the volumetric flow rate of water from tank 3 to tank 1, and q_1 [cm^3/s] the volumetric flow rate of water out of tank 1 (see Figure 1.3). Consequently

$$\begin{aligned} \text{Accumulated} &= m_1(t + \Delta t) - m_1(t) \\ \text{In} &= \rho q_{1,in}(t)\Delta t + \rho q_3(t)\Delta t \\ \text{Out} &= \rho q_1(t)\Delta t \end{aligned}$$

and the mass balance becomes

$$\underbrace{m_1(t + \Delta t) - m_1(t)}_{\text{Accumulated}} = \underbrace{\rho q_{1,in}(t)\Delta t + \rho q_3(t)\Delta t}_{\text{In}} - \underbrace{\rho q_1(t)\Delta t}_{\text{Out}} \tag{1.2}$$

which by division by Δt yields

$$\frac{m_1(t + \Delta t) - m_1(t)}{\Delta t} = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \tag{1.3}$$

Let $\Delta t \to 0$ such that (1.3) becomes

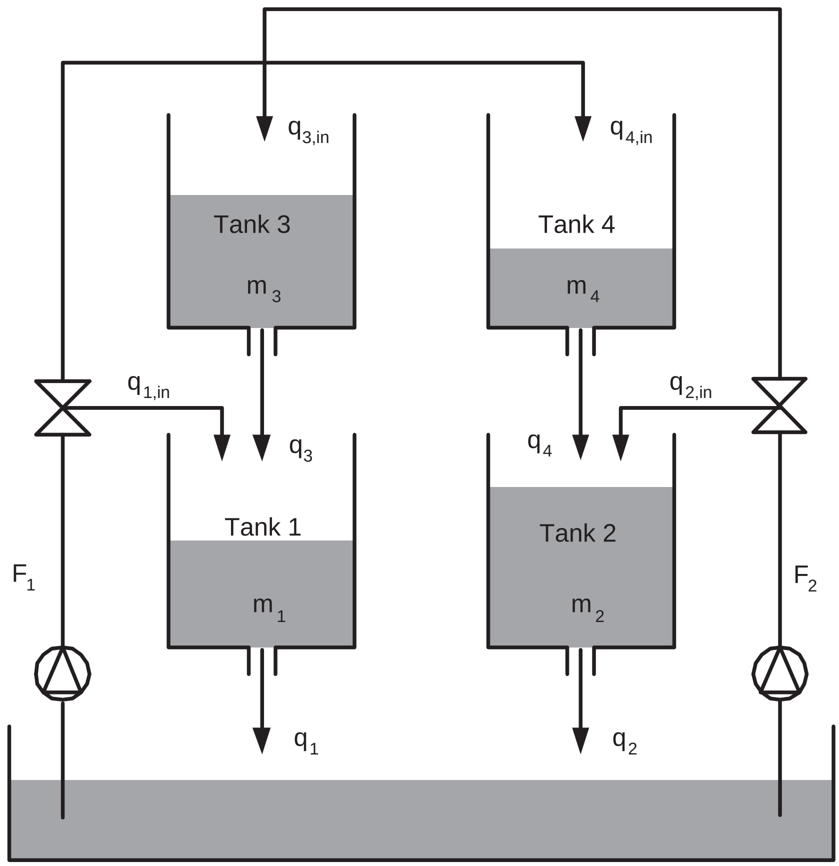

Fig. 1.3. Schematic diagram of the quadruple tank process.

$$\frac{dm_1(t)}{dt} = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \tag{1.4}$$

This expression is obtained by remembering that the differential operator is defined as

$$\frac{dm_1(t)}{dt} \triangleq \lim_{\Delta t \to 0} \frac{m_1(t + \Delta t) - m_1(t)}{\Delta t} \tag{1.5}$$

1.1.2 Mass Balances for the Four Tanks

Using this principle for mass balances, the mass balance for each of the four tanks may be formulated as the differential equations

$$\frac{dm_1(t)}{dt} = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \tag{1.6a}$$
$$\frac{dm_2(t)}{dt} = \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) \tag{1.6b}$$
$$\frac{dm_3(t)}{dt} = \rho q_{3,in}(t) - \rho q_3(t) \tag{1.6c}$$
$$\frac{dm_4(t)}{dt} = \rho q_{4,in}(t) - \rho q_4(t) \tag{1.6d}$$

The initial values, i.e. the mass of water in each of the four tanks at time t_0 are

$$m_1(t_0) = m_{1,0} \tag{1.7a}$$
$$m_2(t_0) = m_{2,0} \tag{1.7b}$$
$$m_3(t_0) = m_{3,0} \tag{1.7c}$$
$$m_4(t_0) = m_{4,0} \tag{1.7d}$$

1.1.3 Inflows

The flow rates from the valves into each of the four tanks are obtained using static mass balances around each of the two valves. Define γ_1 as a constant specifying the fraction of water from pump 1 that goes into tank 1

$$\gamma_1 = \frac{\rho q_{1,in}(t)}{\rho F_1(t)} = \frac{q_{1,in}(t)}{F_1(t)} \tag{1.8}$$

Since mass flowing into valve 1 equals the mass flowing out from valve 1, we have

$$\rho F_1(t) = \rho q_{1,in}(t) + \rho q_{3,in}(t) \tag{1.9}$$

which implies

$$q_{4,in}(t) = F_1(t) - q_{1,in}(t) = F_1(t) - \gamma_1 q_{1,in}(t) = (1 - \gamma_1)F_1(t) \tag{1.10}$$

The parameter γ_1 and the flow rate $F_1(t)$ in pump1 determines the flow rates $q_{1,in}(t)$ and $q_{4,in}(t)$ as

$$q_{1,in}(t) = \gamma_1 F_1(t) \tag{1.11a}$$
$$q_{4,in}(t) = (1 - \gamma_1)F_1(t) \tag{1.11b}$$

Similarly, define the constant parameter γ_2 as the fraction of water from pump 2 that flows into tank 2

$$\gamma_2 = \frac{\rho q_{2,in}(t)}{\rho F_2(t)} = \frac{q_{2,in}(t)}{F_2(t)} \tag{1.12}$$

This determines the flow rates into tank 2 and tank 3 from valve 2 as

$$q_{2,in}(t) = \gamma_2 F_2(t) \tag{1.13a}$$
$$q_{3,in}(t) = (1-\gamma_2) F_2(t) \tag{1.13b}$$

Consequently, the flow rates into each of the tanks from the valves are described by the equations

$$q_{1,in}(t) = \gamma_1 F_1(t) \tag{1.14a}$$
$$q_{2,in}(t) = \gamma_2 F_2(t) \tag{1.14b}$$
$$q_{3,in}(t) = (1-\gamma_2) F_2(t) \tag{1.14c}$$
$$q_{4,in}(t) = (1-\gamma_1) F_1(t) \tag{1.14d}$$

1.1.4 Outflows

The water flows out of each tank through small pipes inserted into the bottom of each tank. This flow is driven by gravity and determined using Bernoulli's Principle. Bernoulli's Principle states that mechanical energy of a fluid along a stream line is conserved, i.e. the sum of potential energy, kinetic energy, and work remains constant along the stream line

$$\rho g h + \frac{1}{2}\rho v^2 + p = \text{constant} \tag{1.15}$$

ρ is the density of the fluid, h is the liquid height above a baseline, v is the velocity, and p is the pressure.

Consider the outflow from tank 1. Let h_1 be the liquid height above a baseline in tank 1. The baseline is the outlet of the pipe. This implies that the pressure at the liquid surface in the top of the tank and the pressure at the pipe outlet are both equal to atmospheric pressure and hence identical. It is important to note, that h_1 is not the height about the bottom of the tank but level above the pipe outlet. Let v_1 be the linear velocity [m/s] of water flowing in the outlet pipe from tank 1. The velocity, v_{1a}, at the top of the tank is much smaller than v_{1a}, i.e. $v_{1a} \ll v_1$ as $a_1 \ll A_1$ in which a_1 is the cross sectional area of the pipe and A_1 is the cross sectional area of the tank. Then Bernoulli's equation applied to the liquid level in the top and the pipe outlet gives

$$\rho g h_1 + \frac{1}{2}\rho v_{1a}^2 + p = \rho g 0 + \frac{1}{2}\rho v_1^2 + p \tag{1.16}$$

such that

$$v_1 = \sqrt{v_{1a}^2 + 2 g h_1} \approx \sqrt{2 g h_1} \tag{1.17}$$

The volumetric flow rate in the outlet pipe from tank 1 is

$$q_1 = a_1 v_1 = a_1 \sqrt{2 g h_1} \tag{1.18}$$

Applying Bernoulli's Principle to each tank gives the following volumetric outflow rates

$$q_1 = a_1 \sqrt{2gh_1} \tag{1.19a}$$
$$q_2 = a_2 \sqrt{2gh_2} \tag{1.19b}$$
$$q_3 = a_3 \sqrt{2gh_3} \tag{1.19c}$$
$$q_4 = a_4 \sqrt{2gh_4} \tag{1.19d}$$

Assume that the cross sectional area, A_i $i \in \{1, 2, 3, 4\}$, for each tank is constant and independent of height, e.g. they could be cylindrical. Let V_i be the volume of the water in tank $i \in \{1, 2, 3, 4\}$. This implies

$$V_i = A_i h_i \qquad i \in \{1, 2, 3, 4\} \tag{1.20}$$

and consequently that mass m_i in tank i is related to height h_i of the liquid level in tank i by

$$m_i = \rho V_i = \rho A_i h_i \qquad i \in \{1, 2, 3, 4\} \tag{1.21}$$

Accordingly, we may compute the liquid height for each tank by the relations

$$h_1 = \frac{m_1}{\rho A_1} \tag{1.22a}$$
$$h_2 = \frac{m_2}{\rho A_2} \tag{1.22b}$$
$$h_3 = \frac{m_3}{\rho A_3} \tag{1.22c}$$
$$h_4 = \frac{m_4}{\rho A_4} \tag{1.22d}$$

1.1.5 The Complete Model of the Four Tank System

The complete model to simulate the four tank systems consist of the equations for the flow from the valves to each tank (1.23), the relations for the liquid heights (1.24), the relations for the outlet flow rates (1.25), the differential equations (1.26) and their initial conditions (1.27).

Flow rates from the valves

$$q_{1,in}(t) = \gamma_1 F_1(t) \tag{1.23a}$$
$$q_{2,in}(t) = \gamma_2 F_2(t) \tag{1.23b}$$
$$q_{3,in}(t) = (1 - \gamma_2) F_2(t) \tag{1.23c}$$
$$q_{4,in}(t) = (1 - \gamma_1) F_1(t) \tag{1.23d}$$

Liquid heights

$$h_1 = \frac{m_1}{\rho A_1} \tag{1.24a}$$
$$h_2 = \frac{m_2}{\rho A_2} \tag{1.24b}$$
$$h_3 = \frac{m_3}{\rho A_3} \tag{1.24c}$$
$$h_4 = \frac{m_4}{\rho A_4} \tag{1.24d}$$

Flow rates out of each tank

$$q_1 = a_1\sqrt{2gh_1} \tag{1.25a}$$
$$q_2 = a_2\sqrt{2gh_2} \tag{1.25b}$$
$$q_3 = a_3\sqrt{2gh_3} \tag{1.25c}$$
$$q_4 = a_4\sqrt{2gh_4} \tag{1.25d}$$

Mass balances

$$\frac{dm_1(t)}{dt} = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \tag{1.26a}$$
$$\frac{dm_2(t)}{dt} = \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) \tag{1.26b}$$
$$\frac{dm_3(t)}{dt} = \rho q_{3,in}(t) - \rho q_3(t) \tag{1.26c}$$
$$\frac{dm_4(t)}{dt} = \rho q_{4,in}(t) - \rho q_4(t) \tag{1.26d}$$

Initial conditions for the state variables.

$$m_1(t_0) = m_{1,0} \tag{1.27a}$$
$$m_2(t_0) = m_{2,0} \tag{1.27b}$$
$$m_3(t_0) = m_{3,0} \tag{1.27c}$$
$$m_4(t_0) = m_{4,0} \tag{1.27d}$$

1.2 Simulation using Matlab

Let

$$x = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \quad u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad y = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} \quad z = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \tag{1.28}$$

The complete model (1.23)-(1.26) of the four tank system defines the vector function f. (1.24) defines the sensor function g and (1.24a)-(1.24b) defines the output function h. The parameter vector, p, in the model is defined as

$$p = \left[a_1\, a_2\, a_3\, a_4\, A_1\, A_2\, A_3\, A_4\, \gamma_1\, \gamma_2\, g\, \rho\right]' \tag{1.29}$$

Using this notation, the system of differential equations governing the evolution of the system may be represented as

$$\frac{dx(t)}{dt} = f(t, x(t), u(t), p) \qquad x(t_0) = x_0 \quad t \in [t_0,\, t_f] \tag{1.30}$$

For simulation in Matlab, this system is defined by the m-file `ProcessModel.m` with the interface

```
function xdot = ProcessModel(t,x,u,p)
% Process Model   dx/dt = f(t,x,u,p)
%
% Computes the RHS of dx/dt = f(t,x,u,p)
% It stores the result in xdot.

...
```

Given t, x, u, and p, `ProcessModel` computes $f(t, x, u, p)$ and returns the result. The solution to this system may be obtained using one of Matlabs ODE-solvers, e.g. `ode15s`. Using `ode15s`, the solution is obtained by the command

```
[T,X] = ode15s(@ProcessModel,[t0 tf],x0,odeOptions,u,p)
```

in which `odeOptions = []` specifies default options for the ODE-solver.

Algorithm 1 Matlab implementation of the model for the four tank system.

```
function xdot = FourTankSystem(t,x,u,p)
% FOURTANKSYSTEM Model dx/dt = f(t,x,u,p) for 4-tank system
%
% This function implements a differential equation model
% for the 4-tank system.
%
% Syntax: xdot = FourTankSystem(t,x,u,p)

% Unpack states, MVs, and parameters
m     = x;               % Mass of liquid in each tank [g]
F     = u;               % Flow rates in pumps [cm3/s]
a     = p(1:4,1);        % Pipe cross sectional areas [cm2]
A     = p(5:8,1);        % Tank cross sectional areas [cm2]
gamma = p(9:10,1);       % Valve positions [-]
g     = p(11,1);         % Acceleration of gravity [cm/s2]
rho   = p(12,1);         % Density of water [g/cm3]

% Inflows
qin = zeros(4,1);
qin(1,1) = gamma(1)*F(1);      % Valve 1 to tank 1 [cm3/s]
qin(2,1) = gamma(2)*F(2);      % Valve 2 to tank 2 [cm3/s]
qin(3,1) = (1-gamma(2))*F(2); % Valve 2 to tank 3 [cm3/s]
qin(4,1) = (1-gamma(1))*F(1); % Valve 1 to tank 4 [cm3/s]

% Outflows
h = m./(rho*A);             % Liquid level in each tank [cm]
qout = a.*sqrt(2*g*h);      % Outflow from each tank [cm3/s]

% Differential equations, mass balances
xdot = zeros(4,1);
xdot(1,1) = rho*(qin(1,1)+qout(3,1)-qout(1,1)); % Tank 1
xdot(2,1) = rho*(qin(2,1)+qout(4,1)-qout(2,1)); % Tank 2
xdot(3,1) = rho*(qin(3,1)-qout(3,1));           % Tank 3
xdot(4,1) = rho*(qin(4,1)-qout(4,1));           % Tank 4
```

1.2.1 Simulation of the Four-Tank System

A Matlab implementation of the model for the Four Tank System is provided in Algorithm 1. This Matlab implementation of the four tank system model is such that it is easy to read but not necessarily computational efficient. For a computational efficient algorithm, we would not unpack variables but use the input arguments directly in the expressions for algebraic equations (inflows and outflows) and in the differential equations (mass balances).

In the following we describe a Matlab script file that may be used to simulate the Four Tank System. First define the parameters for the system

```
% ----------------------------------------------------------
% Parameters
% ----------------------------------------------------------
  a1 = 1.2272     %[cm2] Area of outlet pipe 1
  a2 = 1.2272     %[cm2] Area of outlet pipe 2
  a3 = 1.2272     %[cm2] Area of outlet pipe 3
  a4 = 1.2272     %[cm2] Area of outlet pipe 4

  A1 = 380.1327   %[cm2] Cross sectional area of tank 1
  A2 = 380.1327   %[cm2] Cross sectional area of tank 2
  A3 = 380.1327   %[cm2] Cross sectional area of tank 3
  A4 = 380.1327   %[cm2] Cross sectional area of tank 4


  gamma1 = 0.45;  % Flow distribution constant. Valve 1
  gamma2 = 0.40;  % Flow distribution constant. Valve 2

  g = 981;        %[cm/s2] The acceleration of gravity
  rho = 1.00;     %[g/cm3] Density of water

  p = [a1;a2;a3;a4; A1;A2;A3;A4; gamma1;gamma2; g; rho];
% ---------------------------------------------------------
```

Next define the simulation scenario, i.e. the time interval for the solution, the initial conditions, x_0, and the profiles of the input variables, u. In this case we consider starting with empty tanks and flow rates $F_1 = F_2 = 300\ \text{cm}^3/\text{s}$.

```
% ---------------------------------------------------------
% Simulation scenario
% ---------------------------------------------------------
  t0 = 0.0;         % [s] Initial time
  tf = 20*60;       % [s] Final time

  m10 = 0.0;        % [g] Liquid mass in tank 1 at time t0
  m20 = 0.0;        % [g] Liquid mass in tank 2 at time t0
  m30 = 0.0;        % [g] Liquid mass in tank 3 at time t0
  m40 = 0.0;        % [g] Liquid mass in tank 4 at time t0

  F1 = 300;         % [cm3/s] Flow rate from pump 1
  F2 = 300;         % [cm3/s] Flow rate from pump 2

  x0 = [m10; m20; m30; m40];
  u = [F1; F2]
% ---------------------------------------------------------
```

The solution, $x(t)$, as well as the outputs, $y(t)$, and the outlet flow rates $q_i(t)$ $i \in \{1, 2, 3, 4\}$ are computed by the code

```
% ---------------------------------------------------------
% Compute the solution / Simulate
% ---------------------------------------------------------
% Solve the system of differential equations
[T,X] = ode15s(@FourTankSystem,[t0 tf],x0,[],u,p);

% help variables
[nT,nX] = size(X);
a = p(1:4,1)';
A = p(5:8,1)';

% Compute the measured variables
H = zeros(nT,nX);
for i=1:nT
    H(i,:) = X(i,:)./(rho*A);
end

% Compute the flows out of each tank
Qout = zeros(nT,nX);
for i=1:nT
    Qout(i,:) = a.*sqrt(2*g*H(i,:));
end
% ---------------------------------------------------------
```

Finally, the results are plotted. Figures 1.4-1.6 contain the plots showing the results of the simulation. The system reach a steady state in which the masses in each of the four tanks remain constant and does not change, $\dot{x}(t) = 0$, implying that the total flow rates into each tank equals the flow rate out of each tank.

1.2.2 Discrete-Time Simulation

Computer controlled systems are sampled at discrete times. Let the sample time be T_s. Then the system is sampled at a fixed interval. This means that the discrete time instants at which the system is sampled are

$$t_k = t_0 + kT_s \qquad k = 0, 1, 2, \dots \tag{1.31}$$

At these times, the sensor data is converted from analog to digital signals and available for the computer controlling the system. The sensor signals at these discrete times may be denoted

$$y_k = g(x_k) \qquad k = 0, 1, 2, \dots \tag{1.32}$$

in which $y_k = y(t_k)$ and $x_k = x(t_k)$.

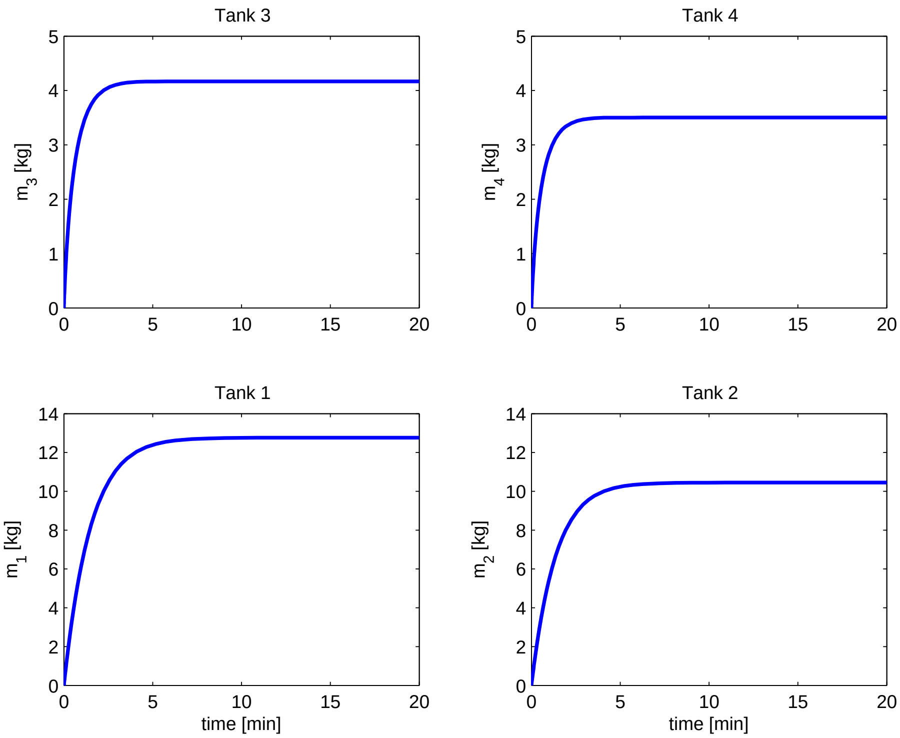

Fig. 1.4. Mass of liquid in each of the four tanks.

In each interval, $t_k \leq t < t_{k+1}$, between two sample times the manipulated input, $u(t)$ is kept constant at the value u_k, i.e.

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} \tag{1.33}$$

This is called zero-order-hold, since $u(t)$ is implemented as by piecewise zero-order functions.

In the interval $t_k \leq t < t_{k+1}$, the process evolution is governed by the system of differential equations

$$\frac{dx(t)}{dt} = f(x(t), u_k) \qquad x(t_k) = x_k, \quad t_k \leq t < t_{k+1} \tag{1.34}$$

with x_k being the state values at the beginning of this interval, u_k being the constant inputs during the interval, and $x_{k+1} = x(t_{k+1})$ being the solution at the end of the interval. This solution may also be defined by

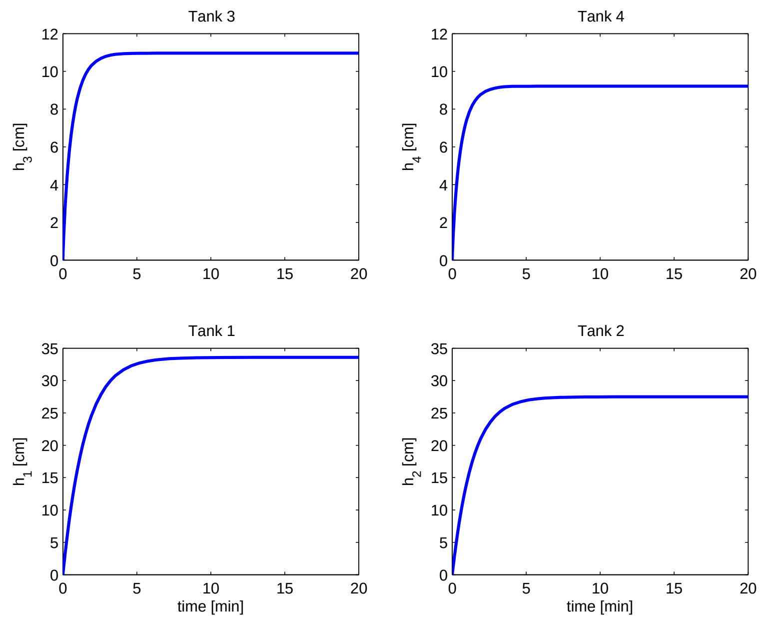

Fig. 1.5. Liquid levels in each of the four tanks.

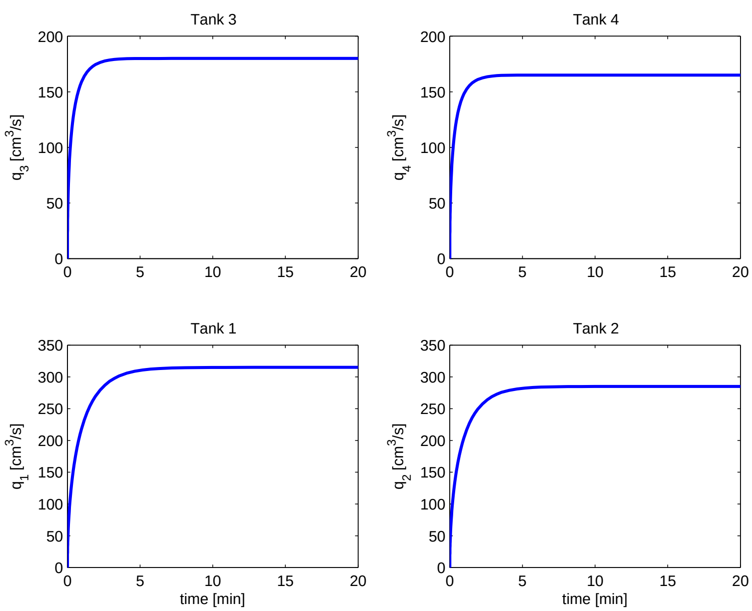

Fig. 1.6. Flow rates out of each of the four tanks.

$$
\begin{aligned}
x(t_{k+1}) &= x(t_k) + \int_{t_k}^{t_{k+1}} f(x(t), u_k)dt \\
&= x_k + \int_{t_k}^{t_{k+1}} f(x(t), u_k)dt \\
&= F(x_k, u_k)
\end{aligned}
\tag{1.35}
$$

or

$$
x_{k+1} = x(t_{k+1}) = F(x_k, u_k) \tag{1.36}
$$

in which $F(x_k, u_k)$ is the operator defining a system of *difference* equations. $F(x_k, u_k)$ is defined as the function

$$
F(x_k, u_k) = x_k + \int_{t_k}^{t_{k+1}} f(x(t), u_k)dt \tag{1.37}
$$

In practice $F(x_k, u_k)$ is computed by solution of (1.34).

Consequently, assuming that x_0 and $\{u_k\}_0^N$ are known we may simulate the system at discrete times by solution of the difference and algebraic equations

$$
x_{k+1} = F(x_k, u_k) \tag{1.38a}
$$

$$
y_k = g(x_k) \tag{1.38b}
$$

$$
z_k = h(x_k) \tag{1.38c}
$$

with $x_k = x(t_k)$, $y_k = y(t_k)$, $z_k = z(t_k)$, and $F(x_k, u_k)$ being a discrete-time map transferring the system from its current state to the next state.

Matlab Implementation

In Matlab, the discrete-time operator F may be implemented by the code

```
[T,X] = ode15s(@f,Tspan,x,odeOptions,u);
xnext = X(end,:)';
```

The system of differential equations (1.34) is solved by `ode15s` and the solution returned is $x_{k+1} = F(x_k, u_k)$. Hence, (1.38) may be solved using Matlab code like

Matlab Simulation

The flow rate sequence depicted in Figure 1.7 is realized using a sample time of 1 minute (60 s). To simulate the evolution of the system with these piecewise constant inputs, we use the code in Algorithm 3. `FourTankSystemSensor` is an implementation of the sensor function (1.24), and `FourTankSystemOutput` is an implementation of the output function (1.24a)-(1.24b).

Algorithm 2 Discrete-time simulation of continuous-discrete systems.

```
for i=0:N
  k=i+1;
  y(:,k) = g(x(:,k));
  z(:,k) = h(x(:,k));
  [Tk,Xk] = ode15s(@f,[t(k) t(k+1)],x(:,k),...
                      odeOptions,u(:,k));
  x(:,k+1) = Xk(end,:)';
  T = [T; Tk];
  X = [X; Xk];
end
```

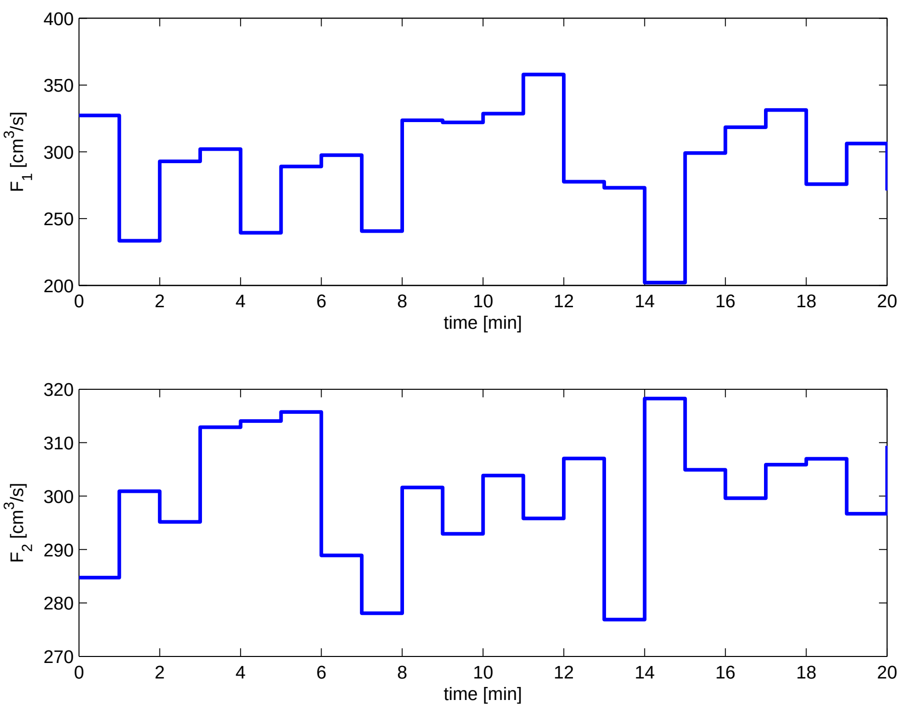

Fig. 1.7. Flow rates.

The result of this simulation is illustrated in Figure 1.8. It shows the liquid masses in each tank as function of time.

Figure 1.9 contains a plot of the inputs and outputs. Input-output plots are often used for visual inspection of the performance of a system, while plots of states and fluxes can be used for detailed analysis of system behavior.

Algorithm 3 Discrete-time simulation of the four tank system.

```
X = zeros(0,nx);
T = zeros(0,1);

x(:,1) = x0;     % Initial condition
for k = 1:N-1
    % Sensor function and output function
    y(:,k) = FourTankSystemSensor(x(:,k),p);
    z(:,k) = FourTankSystemOutput(x(:,k),p);

    % Simulate from time t[k] to time t[k+1]
    [Tk,Xk]=ode15s(@FourTankSystem,[t(k) t(k+1)],x(:,k),...
                            [],u(:,k),p);
    x(:,k+1) = Xk(end,:)';

    % Store the simulated results (for plotting)
    T = [T; Tk];
    X = [X; Xk];
end
k = N;
    % Sensor function and output function
    y(:,k) = FourTankSystemSensor(x(:,k),p);
    z(:,k) = FourTankSystemOutput(x(:,k),p);
```

It is important to remember that the model is *not* the process, but a model of the process. The process does not necessarily evolve as predicted by the simulation model. For evaluation and comparison purposes one often plots the actual measured process values and the simulated process measurements. The simulated process measurements are the four heights for the four tank system. These simulated measurements are illustrated in Figure 1.10.

1.2.3 Stochastic Simulation

Consider the stochastic discrete-time system

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, u_k, \boldsymbol{w}_k) \quad \boldsymbol{x}_0 \sim N(\bar{x}_0, P_0) \quad \boldsymbol{w}_k \sim N_{iid}(0, Q) \tag{1.39a}$$

$$\boldsymbol{y}_k = g(\boldsymbol{x}_k) + \boldsymbol{v}_k \qquad \boldsymbol{v}_k \sim N_{iid}(0, R) \tag{1.39b}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) \tag{1.39c}$$

Bold symbols indicate stochastic variables. $\boldsymbol{x}_k$ is the state vector, u_k is the manipulated input vector, $\boldsymbol{w}_k$ is process noise, $\boldsymbol{y}_k$ is the measurement vector, $\boldsymbol{v}_k$ is additive measurement noise, and $\boldsymbol{z}_k$ is the outputs.

The initial state may not be known exactly. Therefore, initial condition, $\boldsymbol{x}_0$, is a random variable with mean value $\bar{x}_0$ and covariance P_0. In most

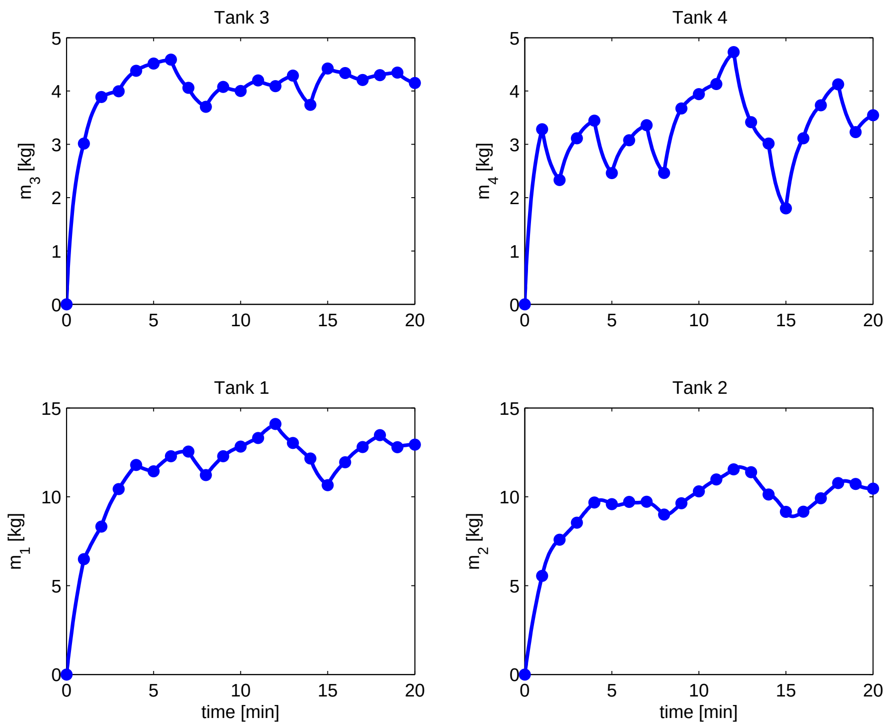

Fig. 1.8. The evolution of masses in the four tanks for the flow rate scenario.

applications we will consider $\boldsymbol{x}_0$ a normally (Gaussian) distributed variable. It can of course be the realization of any other distribution. However, in the absence of further knowledge about the distribution, it is often assumed to be normally distributed.

The process noise, $\{\boldsymbol{w}_k\}$, is assumed to be independent and identically normally distributed, i.e. $\boldsymbol{w}_k \sim N(0, Q)$. The process noise may be that the input streams to the system are random. E.g. for the four tank system one may assume that the flow rates, F_1 and F_2, have a stochastic component. The process noise may also be used to represent model uncertainty. E.g. the position of the valves, γ_1 and γ_2, could be modeled as stochastic variables. Finally, the process noise, $\{\boldsymbol{w}_k\}$, may be state uncertainty, i.e. the process noise, $\boldsymbol{w}_k$, is added to the states.

The measurement noise, $\boldsymbol{v}_k$, may be used to represent that sensors on physical process does not measure perfectly but is corrupted by noise. In most applications the measurement noise is additive. Usually normal distributions are used to represent sensor noise, $\boldsymbol{v}_k \sim N_{iid}(0, R)$.

To simulate the discrete-time stochastic system (1.39) we need to make a realization of the stochastic variables

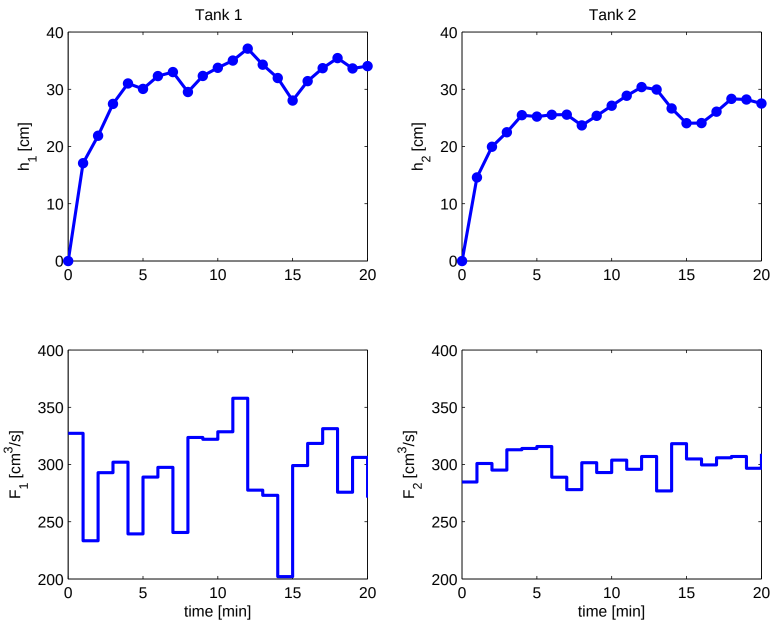

Fig. 1.9. Input-output plot for the flow rate scenario.

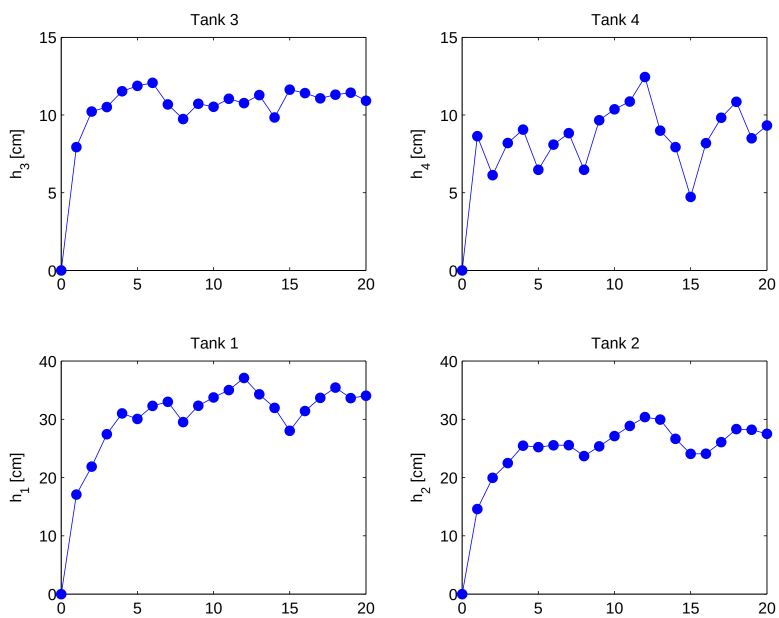

Fig. 1.10. Input-output plot for the flow rate scenario.

Algorithm 4 Generation of $\boldsymbol{w}_k \sim N(0, Q)$

```
Q = [2 1; 1 10];
L = chol(Q)';

MySeed = 100;            % You can use any number to make
randn('state', MySeed); % the random sequence reproducible
w = L*randn(2,1000);     % Generation of the random sequence
```

$$\boldsymbol{x}_0 \sim N(\bar{x}_0, P_0) \tag{1.40a}$$
$$\boldsymbol{w}_k \sim N_{iid}(0, Q) \tag{1.40b}$$
$$\boldsymbol{v}_k \sim N_{iid}(0, R) \tag{1.40c}$$

These normally distributed stochastic variables may be realized from a stochastic variable, $\boldsymbol{e}_k$, with standard normal distribution

$$\boldsymbol{e}_k \sim N_{iid}(0, I) \tag{1.41}$$

In Matlab, the standard normal distribution may be realized using `randn`.

Consider, $\boldsymbol{w}_k \sim N(0, Q)$, and let

$$Q = LL' \tag{1.42}$$

in which L is a lower triangular matrix. If the covariance matrix Q is positive definite, L may be computed by a Cholesky factorization of Q. Then $\boldsymbol{e}_k$ and $\boldsymbol{w}_k$ are related by

$$\boldsymbol{w}_k = L\boldsymbol{e}_k \tag{1.43}$$

$\boldsymbol{w}_k$ is a linear function of $\boldsymbol{e}_k$, which is normally distributed. Therefore, $\boldsymbol{w}_k$ is also normally distributed. The mean and variance of $\boldsymbol{w}_k$ are

$$E\{\boldsymbol{w}_k\} = E\{L\boldsymbol{e}_k\} = LE\{\boldsymbol{e}_k\} = 0 \tag{1.44a}$$
$$V\{\boldsymbol{w}_k\} = \langle \boldsymbol{w}_k, \boldsymbol{w}_k \rangle = \langle L\boldsymbol{e}_k, L\boldsymbol{e}_k \rangle = L\underbrace{\langle \boldsymbol{e}_k, \boldsymbol{e}_k \rangle}_{=I}L' = LL' = Q \tag{1.44b}$$

Consequently, $\boldsymbol{w}_k$ has the required distribution

$$\boldsymbol{w}_k \sim N_{iid}(0, Q) \tag{1.45}$$

The Matlab code in Algorithm 4 illustrates generation of a stochastic sequence $\{\boldsymbol{w}_k\}$ that stems from a normal distribution with zero mean and covariance Q. The generated sequence $\{w_k\}$ and its distribution is plotted in Figure 1.11.

In the case when Q is positive semi-definite, the Cholesky factorization cannot be used. Instead we apply the LDL-factorization

$$Q = LDL' = LD^{1/2}(D^{1/2})'L' = \bar{L}\bar{L}', \quad \bar{L} = LD^{1/2} \tag{1.46}$$

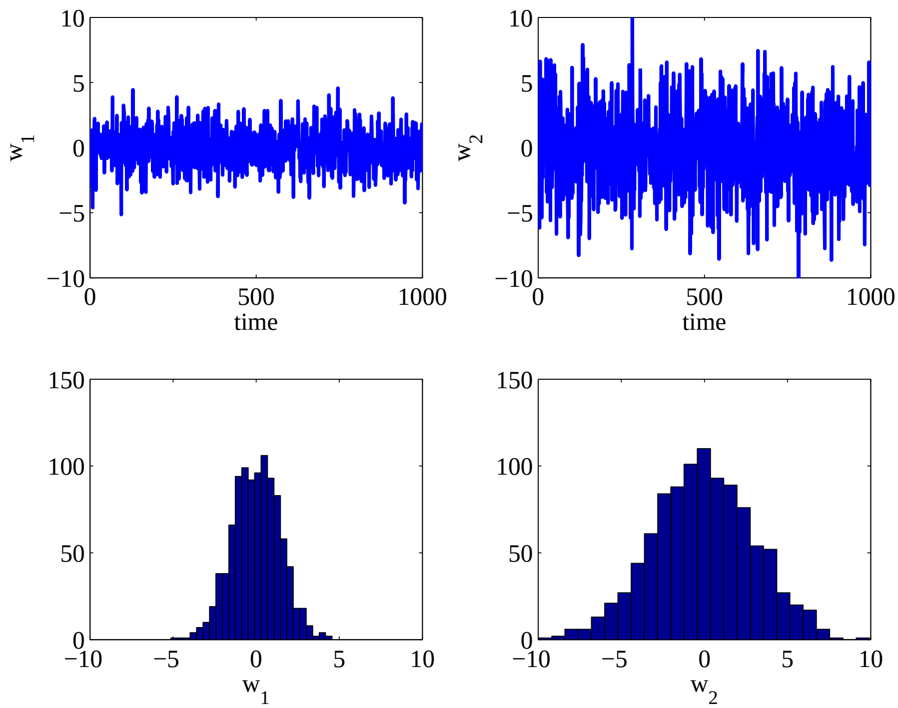

Fig. 1.11. Stochastic Process Noise.

in which L is a lower triangular matrix with unit diagonal and D is a diagonal matrix. $\bar{L}$ is a lower triangular matrix and the required stochastic variables, $\boldsymbol{w}_k \sim N_{iid}(0, Q)$, may be computed by

$$\boldsymbol{w}_k = \bar{L}\boldsymbol{e}_k \qquad \boldsymbol{e}_k \sim N_{iid}(0, I) \tag{1.47}$$

Matlab Simulation

We consider the four tank system with uncertain flow rates, F_1 and F_2, as well as measurement noise on the sensors for h_1, h_2, h_3, and h_4.

Let the input flow rates, F_1 and F_2, be the sum of a deterministic component and a stochastic component

$$\begin{bmatrix} \boldsymbol{F}_1 \\ \boldsymbol{F}_2 \end{bmatrix} = \begin{bmatrix} F_{1s} \\ F_{2s} \end{bmatrix} + \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} \qquad \begin{bmatrix} F_{1s} \\ F_{2s} \end{bmatrix} = \begin{bmatrix} 300 \\ 300 \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \end{bmatrix} \sim N_{iid}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 30^2 & 0 \\ 0 & 10^2 \end{bmatrix} \right)$$

Let all sensors be independent and measure the tank level in each tank, respectively. We assume that all sensors have measurement noise with the same variance, i.e.

Algorithm 5 Definition of simulation scenario for stochastic simulation

```
t0 =    0.0;         % [s] Initial time
tf = 20*60;          % [s] Final time
Ts = 10;             % [s] Sample Time
t = [t0:Ts:tf]';     % [s] Sample instants
N = length(t);

m10 = 0;           % [g] Liquid mass in tank 1 at time t0
m20 = 0;           % [g] Liquid mass in tank 2 at time t0
m30 = 0;           % [g] Liquid mass in tank 3 at time t0
m40 = 0;           % [g] Liquid mass in tank 4 at time t0

F1 = 300;            % [cm3/s] Flow rate from pump 1
F2 = 300;            % [cm3/s] Flow rate from pump 2

x0 = [m10; m20; m30; m40];
u = [repmat(F1,1,N); repmat(F2,1,N)];

% Process Noise
Q = [20^2 0;0 40^2];
Lq = chol(Q,'lower');
w = Lq*randn(2,N);

% Measurement Noise
R = eye(4);
Lr = chol(R,'lower');
v = Lr*randn(4,N);
```

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \boldsymbol{y}_3 \\ \boldsymbol{y}_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} + \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \end{bmatrix} \qquad \boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \boldsymbol{v}_3 \\ \boldsymbol{v}_4 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1^2 & 0 & 0 & 0 \\ 0 & 1^2 & 0 & 0 \\ 0 & 0 & 1^2 & 0 \\ 0 & 0 & 0 & 1^2 \end{bmatrix} \right)$$

The outputs, $\boldsymbol{z}$, are the levels in tank 1 and tank 2:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

The definition of the scenario for this stochastic simulation is done using the Matlab code in Algorithm 5. The Matlab code in Algorithm 6 is used to simulate the system. Figures 1.12 and 1.13 contain the results of this stochastic simulation. It should be noticed that the process noise, $\boldsymbol{w}_k$, is assumed to be constant in each simulation interval.

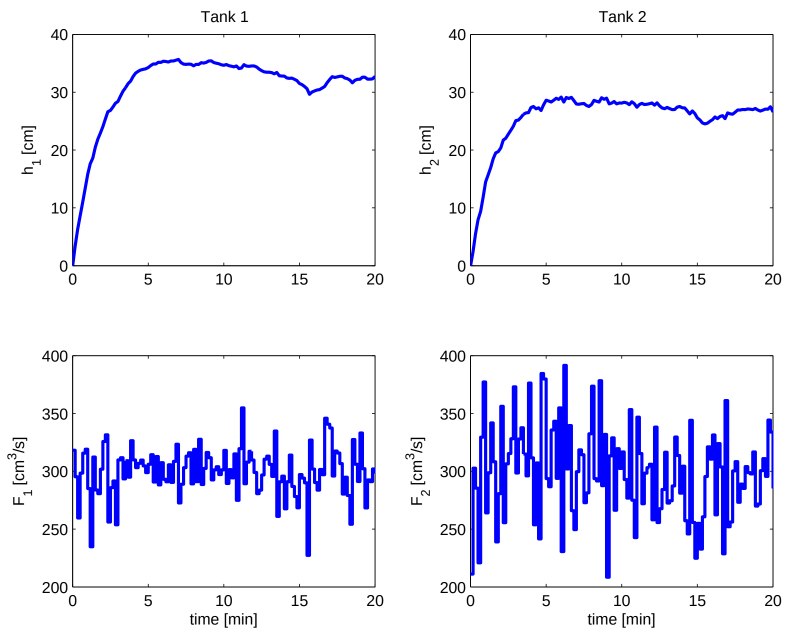

Fig. 1.12. Stochastic process simulation. Top: Outputs. Bottom: Inputs including the stochastic component.

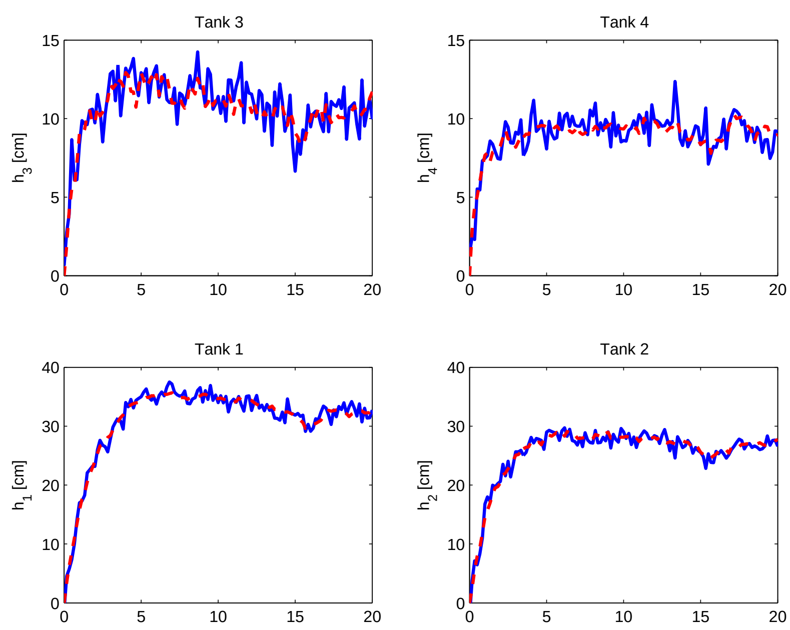

Fig. 1.13. Stochastic process simulation. Blue: Measured levels with measurement noise. Red: Levels without measurement noise.

Algorithm 6 Stochastic Simulation

```
nx = 4; nu = 2; ny = 4; nz = 2;
x = zeros(nx,N);
y = zeros(ny,N);
z = zeros(nz,N);

X = zeros(0,nx);
T = zeros(0,1);

x(:,1) = x0; % Initial condition
for k = 1:N-1
    % Sensor function and output function
    y(:,k) = FourTankSystemSensor(x(:,k),p)+v(:,k);
    z(:,k) = FourTankSystemOutput(x(:,k),p);

    % Simulate from time t[k] to time t[k+1]
    [Tk,Xk]=ode15s(@FourTankSystem,[t(k) t(k+1)],x(:,k),...
                        [],u(:,k)+w(:,k),p);
    x(:,k+1) = Xk(end,:)';

    % Store the simulated results (for plotting)
    T = [T; Tk];
    X = [X; Xk];
end
k = N;
    % Sensor function and output function
    y(:,k) = FourTankSystemSensor(x(:,k),p)+v(:,k);
    z(:,k) = FourTankSystemOutput(x(:,k),p);
```

1.3 Linearization

The model of tank system may be represented as

$$\dot{x}(t) = f(x(t), u(t)) \qquad x(t_0) = x_0 \tag{1.48a}$$

and

$$y(t) = g(x(t)) \tag{1.48b}$$
$$z(t) = h(x(t)) \tag{1.48c}$$

Let $u(t) = u_s$ be a specific input the process. The steady-state, x_s, is the states at which the system remains steady, i.e. $\frac{dx_s}{dt} = 0$. Consequently, given u_s the steady state x_s may be computed as the solution to

$$0 = f(x_s, u_s) \tag{1.49}$$

In Matlab `fsolve` can be used to determine the steady state. Having the steady state, it is trivial to compute the corresponding outputs, $y_s = g(x_s)$ and $z_s = h(x_s)$.

Consider a first-order approximation to the system (1.48a) around the steady-state pair (x_s, u_s):

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ &\approx \underbrace{f(x_s, u_s)}_{=0} + \frac{\partial f}{\partial x}(x_s, u_s)(x(t) - x_s) + \frac{\partial f}{\partial u}(x_s, u_s)(u(t) - u_s) \\ &= A[x(t) - x_s] + B[u(t) - u_s]\end{aligned} \tag{1.50}$$

The matrices A and B are defined as

$$A = \frac{\partial f}{\partial x}(x_s, u_s) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} (x_s, u_s) \tag{1.51a}$$

$$B = \frac{\partial f}{\partial u}(x_s, u_s) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix} (x_s, u_s) \tag{1.51b}$$

Using the original variables, the linear (affine) system (1.50) may be represented as

$$\dot{x}(t) = Ax(t) + Bu(t) + b \tag{1.52}$$

in which

$$b = f(x_s, u_s) - Ax_s - Bu_s = -(Ax_s + Bu_s) \tag{1.53}$$

Introduce the deviation variables defined as

$$X(t) = x(t) - x_s \tag{1.54a}$$
$$U(t) = u(t) - u_s \tag{1.54b}$$

and note that

$$\dot{X}(t) = \frac{dX(t)}{dt} = \frac{d(x(t) - x_s)}{dt} = \frac{dx(t)}{dt} - \frac{dx_s}{dt} = \frac{dx(t)}{dt} = \dot{x}(t)$$

such that (1.50) may be represented as the linear system

$$\dot{X}(t) = AX(t) + BU(t) \tag{1.55a}$$

with the initial condition

$$X(t_0) = X_0 = x_0 - x_s \tag{1.55b}$$

The 1st order Taylor expansion of the sensor function

$$y(t) = g(x(t)) \tag{1.56}$$

is

$$\begin{aligned} y(t) &\approx g(x_s) + \frac{\partial g}{\partial x}(x_s)\,(x(t) - x_s) \\ &= y_s + C(x(t) - x_s) \end{aligned} \tag{1.57}$$

Define the matrix C as

$$C = \frac{\partial g}{\partial x}(x_s) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_{n_x}} \\ \vdots & & \vdots \\ \frac{\partial g_{n_y}}{\partial x_1} & \cdots & \frac{\partial g_{n_y}}{\partial x_{n_x}} \end{bmatrix} (x_s) \tag{1.58}$$

and the deviation variable, $Y(t)$, as

$$Y(t) = y(t) - y_s \tag{1.59}$$

Then a first order Taylor approximation of the sensor function may be expressed as

$$Y(t) = CX(t) \tag{1.60}$$

Similarly, the output equation

$$z(t) = h(x(t)) \tag{1.61}$$

has the first order Taylor approximation

$$Z(t) = C_z X(t) \tag{1.62}$$

with

$$Z(t) = z(t) - z_s \qquad z_s = h(x_s) \tag{1.63}$$

$$C_z = \frac{\partial h}{\partial x}(x_s) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_{n_x}} \\ \vdots & & \vdots \\ \frac{\partial h_{n_z}}{\partial x_1} & \cdots & \frac{\partial h_{n_z}}{\partial x_{n_x}} \end{bmatrix} (x_s) \tag{1.64}$$

Linearization

The system matrices (A, B, C, C_z) defining the linear system

$$\dot{X}(t) = AX(t) + BU(t) \qquad X(t_0) = X_0 \tag{1.65a}$$

$$Y(t) = CX(t) \tag{1.65b}$$

$$Z(t) = C_z X(t) \tag{1.65c}$$

approximating the nonlinear system

$$\dot{x}(t) = f(x(t), u(t)) \qquad x(t_0) = x_0 \tag{1.66a}$$

$$y(t) = g(x(t)) \tag{1.66b}$$

$$z(t) = h(x(t)) \tag{1.66c}$$

around the steady state (x_s, u_s, y_s, z_s) defined by

$$f(x_s, u_s) = 0 \tag{1.67a}$$

$$y_s = g(x_s) \tag{1.67b}$$

$$z_s = h(x_s) \tag{1.67c}$$

are computed as

$$A = \frac{\partial f}{\partial x}(x_s, u_s) \tag{1.68a}$$

$$B = \frac{\partial f}{\partial u}(x_s, u_s) \tag{1.68b}$$

$$C = \frac{\partial g}{\partial x}(x_s) \tag{1.68c}$$

$$C_z = \frac{\partial h}{\partial x}(x_s) \tag{1.68d}$$

The deviation variables $X(t)$, $U(t)$, $Y(t)$, and $Z(t)$ are defined by

$$X(t) = x(t) - x_s \tag{1.69a}$$

$$U(t) = u(t) - u_s \tag{1.69b}$$

$$Y(t) = y(t) - y_s \tag{1.69c}$$

$$Z(t) = z(t) - z_s \tag{1.69d}$$

Example 1 (Linearization of the Four-Tank System Model).
In this example we derive a linear model for the four tank system model (1.23)-(1.26).

Consider the model (1.23)-(1.26) with the differential (1.26) equations expressed as

$$\frac{dm_1(t)}{dt} = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) = f_1(\{m_i\}_{i=1}^4, \{F_i\}_{i=1}^2) \tag{1.70a}$$

$$\frac{dm_2(t)}{dt} = \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) = f_2(\{m_i\}_{i=1}^4, \{F_i\}_{i=1}^2) \tag{1.70b}$$

$$\frac{dm_3(t)}{dt} = \rho q_{3,in}(t) - \rho q_3(t) = f_3(\{m_i\}_{i=1}^4, \{F_i\}_{i=1}^2) \tag{1.70c}$$

$$\frac{dm_4(t)}{dt} = \rho q_{4,in}(t) - \rho q_4(t) = f_4(\{m_i\}_{i=1}^4, \{F_i\}_{i=1}^2) \tag{1.70d}$$

The derivatives of (1.23) with respect to the states, $\{m_1, m_2, m_3, m_4\}$ are

$$\frac{\partial q_{i,in}}{\partial m_j} = 0 \qquad i \in \{1,2,3,4\}, \; j \in \{1,2,3,4\} \tag{1.71}$$

and the derivatives of (1.23) with respect to the MVs, $\{F_1, F_2\}$, are

$$\frac{\partial q_{1,in}}{\partial F_1} = \gamma_1 \qquad \frac{\partial q_{1,in}}{\partial F_2} = 0 \tag{1.72a}$$

$$\frac{\partial q_{2,in}}{\partial F_1} = 0 \qquad \frac{\partial q_{2,in}}{\partial F_2} = \gamma_2 \tag{1.72b}$$

$$\frac{\partial q_{3,in}}{\partial F_1} = 0 \qquad \frac{\partial q_{2,in}}{\partial F_2} = 1 - \gamma_2 \tag{1.72c}$$

$$\frac{\partial q_{4,in}}{\partial F_1} = 1 - \gamma_1 \qquad \frac{\partial q_{2,in}}{\partial F_2} = 0 \tag{1.72d}$$

The derivatives of (1.24) with respect to the states and the MVs are

$$\frac{\partial h_i}{\partial m_i} = \frac{1}{\rho A_i} \qquad i \in \{1,2,3,4\} \tag{1.73a}$$

$$\frac{\partial h_i}{\partial F_j} = 0 \qquad i \in \{1,2,3,4\}, \; j \in \{1,2\} \tag{1.73b}$$

The derivatives of the flow rates out of the tanks (1.25) with respect to the states and the MVs are

$$\frac{\partial q_i}{\partial m_i} = \frac{a_i g}{\sqrt{2gh_i}} \frac{\partial h_i}{\partial m_i} = \frac{a_i g}{A_i \rho \sqrt{2gh_i}} \qquad i \in \{1,2,3,4\} \tag{1.74a}$$

$$\frac{\partial q_i}{\partial F_j} = 0 \qquad i \in \{1,2,3,4\}, \; j \in \{1,2\} \tag{1.74b}$$

Define the time constants, T_i, as

$$T_i = \frac{A_i\sqrt{2gh_i}}{a_i g} = \frac{A_i}{a_i}\sqrt{\frac{2h_i}{g}} \qquad i = \{1,2,3,4\} \tag{1.75}$$

The derivatives of the mass balance, f_1, for tank 1 with respect to the states are

$$\frac{\partial f_1}{\partial m_1} = -\rho\frac{\partial q_1}{\partial m_1} = -\frac{a_1 g}{A_1\sqrt{2gh_1}} = -\frac{1}{T_1} \tag{1.76a}$$
$$\frac{\partial f_1}{\partial m_2} = 0 \tag{1.76b}$$
$$\frac{\partial f_1}{\partial m_3} = \rho\frac{\partial q_3}{\partial m_3} = \frac{a_3 g}{A_3\sqrt{2gh_3}} = \frac{1}{T_3} \tag{1.76c}$$
$$\frac{\partial f_1}{\partial m_4} = 0 \tag{1.76d}$$

The derivatives of the mass balance, f_1, for tank 1 with respect to the MVs are

$$\frac{\partial f_1}{\partial F_1} = \rho\frac{\partial q_{1,in}}{\partial F_1} = \rho\gamma_1 \tag{1.77a}$$
$$\frac{\partial f_1}{\partial F_2} = 0 \tag{1.77b}$$

The derivatives of the mass balance, f_2, for tank 2 with respect to the states are

$$\frac{\partial f_2}{\partial m_1} = 0 \tag{1.78a}$$
$$\frac{\partial f_2}{\partial m_2} = -\rho\frac{\partial q_2}{\partial m_2} = -\frac{a_2 g}{A_2\sqrt{2gh_2}} = -\frac{1}{T_2} \tag{1.78b}$$
$$\frac{\partial f_2}{\partial m_3} = 0 \tag{1.78c}$$
$$\frac{\partial f_2}{\partial m_4} = \rho\frac{\partial q_4}{\partial m_4} = \frac{a_4 g}{A_4\sqrt{2gh_4}} = \frac{1}{T_4} \tag{1.78d}$$

The derivatives of the mass balance, f_2, for tank 2 with respect to the MVs are

$$\frac{\partial f_2}{\partial F_1} = 0 \tag{1.79a}$$
$$\frac{\partial f_2}{\partial F_2} = \rho\frac{\partial q_{2,in}}{\partial F_2} = \rho\gamma_2 \tag{1.79b}$$

The derivatives of the mass balance, f_3, for tank 3 with respect to the states are

$$\frac{\partial f_3}{\partial m_1} = 0 \tag{1.80a}$$

$$\frac{\partial f_3}{\partial m_2} = 0 \tag{1.80b}$$

$$\frac{\partial f_3}{\partial m_3} = -\rho \frac{\partial q_3}{\partial m_3} = -\frac{a_3 g}{A_3\sqrt{2gh_3}} = -\frac{1}{T_3} \tag{1.80c}$$

$$\frac{\partial f_3}{\partial m_4} = 0 \tag{1.80d}$$

The derivatives of the mass balance, f_3, for tank 3 with respect to the MVs are

$$\frac{\partial f_3}{\partial F_1} = 0 \tag{1.81a}$$

$$\frac{\partial f_3}{\partial F_2} = \rho \frac{\partial q_{3,in}}{\partial F_2} = \rho(1-\gamma_2) \tag{1.81b}$$

The derivatives of the mass balance, f_4, for tank 4 with respect to the states are

$$\frac{\partial f_4}{\partial m_1} = 0 \tag{1.82a}$$

$$\frac{\partial f_4}{\partial m_2} = 0 \tag{1.82b}$$

$$\frac{\partial f_4}{\partial m_3} = 0 \tag{1.82c}$$

$$\frac{\partial f_4}{\partial m_4} = -\rho \frac{\partial q_4}{\partial m_4} = -\frac{a_4 g}{A_4\sqrt{2gh_4}} = -\frac{1}{T_4} \tag{1.82d}$$

The derivatives of the mass balance, f_4, for tank 4 with respect to the MVs are

$$\frac{\partial f_4}{\partial F_1} = \rho \frac{\partial q_{4,in}}{\partial F_1} = \rho(1-\gamma_1) \tag{1.83a}$$

$$\frac{\partial f_4}{\partial F_2} = 0 \tag{1.83b}$$

Consequently, the linear system in deviation variables, $X(t) = x(t) - x_s$ and $U(t) = u(t) - u_s$, is

$$\dot{X}(t) = AX(t) + BU(t) \quad X(t_0) = X_0 \tag{1.84}$$

with the matrices A and B computed as

$$A = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{1}{T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{1}{T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \qquad B = \begin{bmatrix} \rho\gamma_1 & 0 \\ 0 & \rho\gamma_2 \\ 0 & \rho(1-\gamma_2) \\ \rho(1-\gamma_1) & 0 \end{bmatrix} \tag{1.85}$$

The sensors measure the liquid level height in each tank, i.e. $y_i = h_i$ for $i \in \{1,2,3,4\}$. This implies that the measured values are related to the states by (1.24). Hence, the measured values is vector function of the type $y(t) = g(x(t))$. The derivatives are

$$\frac{\partial y_i}{\partial m_j} = \frac{\delta_{ij}}{A_i \rho} \quad i \in \{1,2,3,4\}\,,\, j \in \{1,2,3,4\} \tag{1.86}$$

with Kronecker's delta-function defined as

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{1.87}$$

Consequently, the measured values, $Y(t) = y(t) - y_s$, can be expressed as the linear function

$$Y(t) = CX(t) \tag{1.88}$$

with

$$C = \begin{bmatrix} \frac{1}{\rho A_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho A_2} & 0 & 0 \\ 0 & 0 & \frac{1}{\rho A_3} & 0 \\ 0 & 0 & 0 & \frac{1}{\rho A_4} \end{bmatrix} \tag{1.89}$$

The outputs are the liquid level in tank 1 and tank 2, i.e. h_1 and h_2. Therefore, the outputs as deviation variables, $Z(t) = z(t) - z_s$, may be expressed as

$$Z(t) = C_z X(t) \tag{1.90}$$

in which

$$C_z = \begin{bmatrix} \frac{1}{\rho A_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho A_2} & 0 & 0 \end{bmatrix} \tag{1.91}$$

In summary, linearization of the four tank system model produces the linear system

$$\dot{X}(t) = AX(t) + BU(t) \qquad X(t_0) = X_0 \tag{1.92a}$$

$$Y(t) = CX(t) \tag{1.92b}$$

$$Z(t) = C_z X(t) \tag{1.92c}$$

with the matrices

$$A = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{1}{T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{1}{T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix} \qquad B = \begin{bmatrix} \rho\gamma_1 & 0 \\ 0 & \rho\gamma_2 \\ 0 & \rho(1-\gamma_2) \\ \rho(1-\gamma_1) & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} \frac{1}{\rho A_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho A_2} & 0 & 0 \\ 0 & 0 & \frac{1}{\rho A_3} & 0 \\ 0 & 0 & 0 & \frac{1}{\rho A_4} \end{bmatrix} \qquad C_z = \begin{bmatrix} \frac{1}{\rho A_1} & 0 & 0 & 0 \\ 0 & \frac{1}{\rho A_2} & 0 & 0 \end{bmatrix}$$

The time constants, T_i, defined by (1.75).

Comparison of Linear and Nonlinear Response

The first step in deriving a linear model consist of determining the steady state, x_s, at which to perform the linearization. Define a wrapper function having an interface compatible with the interface required by `fsolve`

```
function xdot = FourTankSystemWrap(x,u,p)
xdot = FourTankSystem(0,x,u,p);
```

Then the steady state x_s satisfying

$$0 = f(x_s, u_s) \tag{1.93}$$

may be determined using Matlab code like

```
    xs0 = 5000*ones(4,1);   % Initial guess of steady state
    us  = [300; 300];       % Steady-state inputs
    xs = fsolve(@FourTankSystemWrap,xs0,[],us,p)
```

Consequently, the steady state pair $\{u_s, x_s, y_s = g(x_s), z_s = h(x_s)\}$ is

$$u_s = \begin{bmatrix} F_{1s} \\ F_{2s} \end{bmatrix} = \begin{bmatrix} 300 \text{ cm}^3/\text{s} \\ 300 \text{ cm}^3/\text{s} \end{bmatrix} \qquad z_s = \begin{bmatrix} h_{1s} \\ h_{2s} \end{bmatrix} = \begin{bmatrix} 33.6 \text{ cm} \\ 27.5 \text{ cm} \end{bmatrix}$$

$$x_s = \begin{bmatrix} m_{1s} \\ m_{2s} \\ m_{3s} \\ m_{4s} \end{bmatrix} = \begin{bmatrix} 12765 \text{ g} \\ 10449 \text{ g} \\ 4168 \text{ g} \\ 3502 \text{ g} \end{bmatrix} \qquad y_s = \begin{bmatrix} h_{1s} \\ h_{2s} \\ h_{3s} \\ h_{4s} \end{bmatrix} = \begin{bmatrix} 33.6 \text{ cm} \\ 27.5 \text{ cm} \\ 11.0 \text{ cm} \\ 9.2 \text{ cm} \end{bmatrix}$$

The corresponding time constants (1.75) are

$$T_1 = \frac{A_1}{a_1}\sqrt{\frac{2h_{1s}}{g}} = 81.0 \text{ s}$$

$$T_2 = \frac{A_2}{a_2}\sqrt{\frac{2h_{2s}}{g}} = 73.3 \text{ s}$$

$$T_3 = \frac{A_3}{a_3}\sqrt{\frac{2h_{3s}}{g}} = 46.3 \text{ s}$$

$$T_4 = \frac{A_4}{a_4}\sqrt{\frac{2h_{4s}}{g}} = 42.5 \text{ s}$$

Linearization of the four tank system may be conducted using the Matlab code in Algorithm 7. The computed matrices (A, B, C, C_z) in the linear model (1.92) are listed in Figure 1.14.

The accuracy of the linear model may be assessed by comparison of the step responses for the linear and the nonlinear model. [TODO: insert figure illustring this]

Algorithm 7 Linearization of the Four Tank System.

```
% --------------------------------------------------------
% Parameters
% --------------------------------------------------------
ap = [a1; a2; a3; a4];   % [cm2] Pipe cross sectional areas
At = [A1; A2; A3; A4];   % [cm2] Tank cross sectional areas
gam = [gamma1; gamma2]; % [-] Valve constants
g = 981;                 % [cm/s2] Acceleration of gravity
rho = 1.00;              % [g/cm3] Density of water

p = [ap; At; gamma; g; rho];

% --------------------------------------------------------
% Steady State
% --------------------------------------------------------
us  = [300; 300]                  % [cm3/s] Flow rates
xs0 = [5000; 5000; 5000; 5000]   % [g] Initial guess on xs

xs = fsolve(@FourTankSystemWrap,xs0,[],us,p)
ys = FourTankSystemSensor(xs,p)
zs = FourTankSystemOutput(xs,p)

% --------------------------------------------------------
% Linearization
% --------------------------------------------------------
hs = ys;
T = (At./ap).*sqrt(2*hs/g)

A = [   -1/T(1) 0 1/T(3) 0; ...
        0 -1/T(2) 0 1/T(4); ...
        0 0 -1/T(3) 0; ...
        0 0 0 -1/T(4)    ]

B = [   rho*gam(1) 0; ...
        0 rho*gam(2); ...
        0 rho*(1-gam(2)); ...
        rho*(1-gam(1)) 0]

C = diag(1./(rho*At))

Cz = C(1:2,:)
```

```
A =
   -0.012338            0     0.021592            0
           0    -0.013637            0     0.023555
           0            0    -0.021592            0
           0            0            0    -0.023555

B =
        0.45            0
           0          0.4
           0          0.6
        0.55            0

C =
   0.0026307            0            0            0
           0    0.0026307            0            0
           0            0    0.0026307            0
           0            0            0    0.0026307

Cz =
   0.0026307            0            0            0
           0    0.0026307            0            0
```

Fig. 1.14. Linear model (A, B, C, C_z) for the Four Tank System.

1.3.1 Finite Difference Numerical Linearization

It is apparent that analytical linearization is cumbersome and error prone. Alternatively, the derivatives can be computed numerically. Consider the function $f(x, u)$ in (1.48a). Its first order Taylor approximation is

$$f(x, u) \approx f(x_s, u_s) + A(x - x_s) + B(u - u_s) \tag{1.94}$$

with

$$A = \frac{\partial f}{\partial x}(x_s, u_s) \tag{1.95a}$$

$$B = \frac{\partial f}{\partial u}(x_s, u_s) \tag{1.95b}$$

Using a forward finite difference approximation, the elements in A and B may be computed by

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_s, u_s) \approx \frac{f_i(x_s + \delta e_j, u_s) - f_i(x_s, u_s)}{\delta} \tag{1.96a}$$

$$B_{ij} = \frac{\partial f_i}{\partial u_j}(x_s, u_s) \approx \frac{f_i(x_s, u_s + \delta e_j) - f_i(x_s, u_s)}{\delta} \tag{1.96b}$$

Algorithm 8 Finite Difference Numerical Linearization.

```
function [A,B]=FiniteDifference(fun,xs,us,varargin)

sqrteps = sqrt(eps);

nx = length(xs);
nu = length(us);

A = zeros(nx,nx);
B = zeros(nx,nu);

t = 0;
fs = feval(fun,t,xs,us,varargin{:});

for i=1:nx
    delta = sqrteps*(1.0+abs(xs(i)));
    x = xs;
    x(i) = x(i) + delta;
    delta = x(i)-xs(i);
    f = feval(fun,t,x,us,varargin{:});
    A(:,i) = (f-fs)/delta;
end

for i=1:nu
    delta = 1e-9*(1.0+abs(us(i)));
    u = us;
    u(i) = u(i) + delta;
    delta = u(i)-us(i);
    f = feval(fun,t,xs,u,varargin{:});
    B(:,i) = (f-fs)/delta;
end
```

for a sufficiently small δ. The Matlab code for this numerical computation of A and B is provided in Algorithm 8. The same principle may be applied for computation of C and C_z

$$C_{ij} = \frac{\partial g_i}{\partial x_j}(x_s) \approx \frac{g_i(x_s + \delta e_j) - g_i(x_s)}{\delta} \tag{1.97a}$$

$$(C_z)_{ij} = \frac{\partial h_i}{\partial x_j}(x_s) \approx \frac{h_i(x_s + \delta e_j) - h_i(x_s)}{\delta} \tag{1.97b}$$

Consider the first order Taylor approximation

$$f(x) \approx f(x_s) + A(x - x_s) \tag{1.98}$$

with the Jacobian, A, defined as

$$A = \frac{\partial f}{\partial x}(x_s) \tag{1.99}$$

Numerically, A may be computed using a forward finite difference approximation as illustrated above. However, it can equally well be computed using a backward finite difference approximation or a central finite difference approximation. The forward, backward, and central finite difference approximations are

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_s) = \frac{f_i(x_s + \delta e_j) - f_i(x_s)}{\delta} + O(\delta) \tag{1.100a}$$

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_s) = \frac{f_i(x_s) - f_i(x_s - \delta e_j)}{\delta} + O(\delta) \tag{1.100b}$$

$$A_{ij} = \frac{\partial f_i}{\partial x_j}(x_s) = \frac{f_i(x_s + \delta e_i) - f_i(x_s - \delta e_j)}{2\delta} + O(\delta^2) \tag{1.100c}$$

1.3.2 Discretization

Consider the case in which the input, $U(t)$, is piecewise constant. This is for instance the case when the input is determined by a computer and implemented on the system using zero-order-hold electronics. Let T_s be the sampling time, such that the sampling instants, t_k, are

$$t_k = t_0 + kT_s \qquad k = 0, 1, 2, \ldots$$

The inputs may then be characterized by

$$U(t) = U_k \qquad t_k \leq t < t_{k+1} \quad k = 0, 1, 2, \ldots \tag{1.101}$$

Using the results in Appendix 2 and the piecewise constant input, the system of linear *differential* equations

$$\dot{X}(t) = AX(t) + BU(t) \qquad X(t_k) = X_k \quad t_k \leq t < t_{k+1} \tag{1.102}$$

has the solution

$$\begin{aligned} X_{k+1} = X(t_{k+1}) &= e^{A(t_{k+1}-t_k)}X_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}BU(s)ds \\ &= e^{AT_s}X_k + \int_0^{T_s} e^{A\tau}d\tau BU_k \end{aligned} \tag{1.103}$$

that may be converted to a system of linear *difference* equations

$$X_{k+1} = \bar{A}X_k + \bar{B}U_k \tag{1.104}$$

with $X_k = X(t_k)$ and

Algorithm 9 ZOH Discretization of Linear Systems.

```
function [Abar,Bbar]=c2dzoh(A,B,Ts)

[nx,nu]=size(B);
M = [A B; zeros(nu,nx) zeros(nu,nu)]*Ts;
Phi = expm(M);
Abar = Phi(1:nx,1:nx);
Bbar = Phi(1:nx,nx+1:nx+nu);
```

$$\bar{A} = \exp(AT_s) \tag{1.105a}$$

$$\bar{B} = \int_0^{T_s} \exp(A\tau)\, d\tau B \tag{1.105b}$$

In practice $\bar{A}$ and $\bar{B}$ are computed using the matrix exponential (`expm`) and the expression

$$\begin{bmatrix} \bar{A} & \bar{B} \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T_s \right) \tag{1.106}$$

Consequently, the continuous-time linear system

$$\dot{X}(t) = AX(t) + BU(t) \qquad X(t_0) = X_0 \tag{1.107a}$$

$$Y(t) = CX(t) \tag{1.107b}$$

$$Z(t) = C_z X(t) \tag{1.107c}$$

with the zero-order-hold input

$$U(t) = U_k \qquad t_k \le t < t_{k+1} \quad t_k = t_0 + kT_s \quad k = 0, 1, 2, \ldots \tag{1.108}$$

can be represented as the discrete-time linear system

$$X_{k+1} = \bar{A}X_k + \bar{B}U_k \tag{1.109a}$$

$$Y_k = CX_k \tag{1.109b}$$

$$Z_k = C_z X_k \tag{1.109c}$$

with $(\bar{A}, \bar{B})$ computed by (1.106).

1.4 Linear Stochastic System

We have already experienced that simulation of stochastic *difference* equations is straightforward. Difference equations are associated with discrete times. However, physical systems evolve in continuous time. In this section, we provide some introductory remarks to continuous-time stochastic systems and provides formulas for conversion of linear stochastic differential equations to equivalent linear stochastic difference equations.

Analysis and simulation of continuous-time stochastic systems presents a somewhat larger challenge than analysis and simulation of continuous-time deterministic systems. This is due to the fact that the theory and algorithms for solution of systems of stochastic differential equations (SDEs) is less developed than the theory and algorithms for solution of systems of ordinary differential equations (ODEs).

For nonlinear continuous-time systems, we would like to extend the continuous-time system of first order differential equations

$$\frac{dx(t)}{dt} = f(x(t), u(t)) \qquad x(t_0) = x_0$$

to the stochastic case using a construction like

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t), u(t)) + \sigma(\boldsymbol{x}(t), u(t))\boldsymbol{w}(t) \quad \text{Not well-defined.}$$

with $\{\boldsymbol{w}(t), t \in T\}$ being some stochastic process. However, from a mathematical point of view it turns out that this construction is not well defined. Instead, we extend the approximation

$$x(t + \delta t) - x(t) = f(x(t), u(t))\delta t + o(\delta t) \tag{1.110}$$

of a deterministic differential equation to the stochastic case

$$\boldsymbol{x}(t+\delta t)-\boldsymbol{x}(t) = f(\boldsymbol{x}(t), u(t))\delta t+\sigma(\boldsymbol{x}(t), u(t))[\boldsymbol{w}(t+\delta t)-\boldsymbol{w}(t)]+o(\delta t) \tag{1.111}$$

using increments $\Delta \boldsymbol{w}(t) = \boldsymbol{w}(t+\delta t)-\boldsymbol{w}(t)$ that are independent and normally distributed

$$\Delta \boldsymbol{w}(t) = [\boldsymbol{w}(t + \delta t) - \boldsymbol{w}(t)] \sim N_{iid}(0, I\delta t) = \sqrt{\delta t} N_{iid}(0, I) \tag{1.112}$$

(1.111) is a stochastic difference equation and can be easily simulated for a finite value of δ. In the limit $\delta t \to 0$, (1.111) becomes

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t), u(t))dt + \sigma(\boldsymbol{x}(t), u(t))d\boldsymbol{w}(t) \tag{1.113}$$

with $\{\boldsymbol{w}(t)\}$ being a standard Wiener Process (Brownian motion) defined axiomatically by the conditions

1. $\boldsymbol{w}(t)$ is normally distributed.
2. $\boldsymbol{w}(t)$ is independent of $\boldsymbol{w}(s)$ for all $s \neq t$.
3. $E\{\boldsymbol{w}(t)\} = 0$
4. $E\{d\boldsymbol{w}(t)d\boldsymbol{w}(t)'\} = I dt$

(1.113) is called a stochastic differential equation, or more correct an Ito stochastic differential equation as the limit is taken for the forward difference i.e. $\lim_{\delta t \to 0} \boldsymbol{x}(t + \delta t) - \boldsymbol{x}(t)$. The integral equation corresponding to (1.113) is

$$\boldsymbol{x}(t) = \boldsymbol{x}(t_0) + \int_{t_0}^{t} f(\boldsymbol{x}(s), u(s))ds + \int_{t_0}^{t} \sigma(\boldsymbol{x}(s), u(s))d\boldsymbol{w}(s) \tag{1.114}$$

with the stochastic integrals $\int_{t_0}^{t} f(\boldsymbol{x}(s), u(s))ds$ and $\int_{t_0}^{t} \sigma(\boldsymbol{x}(s), u(s))d\boldsymbol{w}(s)$ being Ito integrals. It should be noted, that Ito integrals require special rules for integration. These rules are different from the rules used for deterministic integrals.

The definition of stochastic differential equations and Wiener processes has led to the following definition of continuous white noise

$$\boldsymbol{e}(t) \sim N(0, I\delta(t)) \tag{1.115}$$

where $\delta(t)$ is Dirac's delta function defined as

$$\delta(t) = \begin{cases} \infty & t = 0 \\ 0 & t \neq 0 \end{cases} \qquad \int_{-\infty}^{\infty} \delta(t)dt = 1 \tag{1.116}$$

such that a Winer process $\{\boldsymbol{w}(t)\}$ can be interpreted as integrated white noise

$$\boldsymbol{w}(t) = \int_{0}^{t} \boldsymbol{e}(s)ds \tag{1.117}$$

In reality the definition of continuous white noise, only represents a new notation that enables us to define

$$\frac{d\boldsymbol{w}(t)}{dt} = \boldsymbol{e}(t) \tag{1.118}$$

such that we may express the stochastic differential equation (1.113) as

$$\frac{d\boldsymbol{x}(t)}{dt} = f(\boldsymbol{x}(t), u(t)) + \sigma(\boldsymbol{x}(t), u(t))\boldsymbol{e}(t) \tag{1.119}$$

It is important to remember that this is merely a notation. This notation is used to define the stochastic differential equation (1.113) which is the limit process of (1.111).

A main focus in the rest of this section is to approximate discrete-time stochastic systems as well as continuous-time stochastic systems with linear stochastic difference equations

$$\boldsymbol{x}_{k+1} = \bar{A}\boldsymbol{x}_k + \bar{B}u_k + \bar{G}\boldsymbol{w}_k \qquad \boldsymbol{w}_k \sim N_{iid}(0, \bar{Q}) \tag{1.120}$$

For continuous-time stochastic linear systems

$$d\boldsymbol{x}(t) = (A\boldsymbol{x}(t) + Bu(t))dt + Gd\boldsymbol{w}(t) \tag{1.121}$$

there exist matrices $(\bar{A}, \bar{B}, \bar{G}, \bar{Q})$ such that (1.120) and (1.121) are equivalent.

1.4.1 Discrete Time

Consider the nonlinear discrete-time system

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, u_k, \boldsymbol{w}_k) \quad \boldsymbol{x}_0 \sim N(\bar{x}_0, P_0) \quad \boldsymbol{w}_k \sim N(0, Q) \tag{1.122a}$$

$$\boldsymbol{y}_k = g(\boldsymbol{x}_k) + \boldsymbol{v}_k \qquad \boldsymbol{v}_k \sim N(0, R) \tag{1.122b}$$

$$\boldsymbol{z}_k = h(\boldsymbol{x}_k) \tag{1.122c}$$

Given u_s, the steady states of this system may be computed by solution of the system

$$x_s = F(x_s, u_s, 0) \quad \Leftrightarrow \quad r(x_s) = x_s - F(x_s, u_s, 0) = 0 \tag{1.123a}$$

and the corresponding values of the measured variables and the outputs are

$$y_s = g(x_s) \tag{1.123b}$$

$$z_s = h(x_s) \tag{1.123c}$$

Define the deviation variables as

$$\boldsymbol{X}(t) = \boldsymbol{x}(t) - x_s \tag{1.124a}$$

$$U(t) = u(t) - u_s \tag{1.124b}$$

$$\boldsymbol{Y}(t) = \boldsymbol{y}(t) - y_s \tag{1.124c}$$

$$\boldsymbol{Z}(t) = \boldsymbol{z}(t) - z_s \tag{1.124d}$$

Then a linear approximation of the stochastic nonlinear discrete-time system (1.122) is

$$\boldsymbol{X}_{k+1} = A\boldsymbol{X}_k + BU_k + G\boldsymbol{w}_k \quad \boldsymbol{X}_0 \sim N(\bar{X}_0, P_0),\ \boldsymbol{w}_k \sim N(0, Q) \tag{1.125a}$$

$$\boldsymbol{Y}_k = C\boldsymbol{X}_k + \boldsymbol{v}_k \qquad \boldsymbol{v}_k \sim N(0, R) \tag{1.125b}$$

$$\boldsymbol{Z}_k = C_z\boldsymbol{X}_k \tag{1.125c}$$

with $\bar{X}_0 = \bar{x}_0 - x_s$. The system matrices (A, B, G, C, C_z) are computed by

$$A = \frac{\partial F}{\partial x}(x_s, u_s, 0) \tag{1.126a}$$

$$B = \frac{\partial F}{\partial u}(x_s, u_s, 0) \tag{1.126b}$$

$$G = \frac{\partial F}{\partial w}(x_s, u_s, 0) \tag{1.126c}$$

$$C = \frac{\partial g}{\partial x}(x_s) \tag{1.126d}$$

$$C_z = \frac{\partial h}{\partial x}(x_s) \tag{1.126e}$$

Even for the linearized system in deviation variables, it is often convenient to use the following notation

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + Bu_k + G\boldsymbol{w}_k & \boldsymbol{x}_0 &\sim N(\bar{x}_0, P_0),\ \boldsymbol{w}_k \sim N(0, Q) && (1.127a)\\
\boldsymbol{y}_k &= C\boldsymbol{x}_k + \boldsymbol{v}_k & \boldsymbol{v}_k &\sim N(0, R) && (1.127b)\\
\boldsymbol{z}_k &= C_z\boldsymbol{x}_k & & && (1.127c)
\end{aligned}$$

We will abuse definition of variables a little and use this notation for the linearized system. It should be clear from the context whether the variables are deviation variables or physical variables.

1.4.2 Continuous-Discrete Time

We consider systems that evolve in continuous-time but are observed at discrete-time and controlled by piecewise constant inputs. Physical systems controlled by digital computers belong to this class of systems. The physical process evolved in continuous time. The digital computer control system measures process variables at discrete time, computes the inputs according to some algorithm, and implements this input using zero-order-hold.

Consider the continuous-time linear stochastic system described by the linear stochastic differential equation

$$d\boldsymbol{x}(t) = A\boldsymbol{x}(t)dt + Gd\boldsymbol{w}(t) \tag{1.128}$$

Let $\boldsymbol{x}(t_0) \sim N(\bar{x}_0, P_0)$ and let $\{\boldsymbol{w}(t)\}$ be a standard Wiener process. The solution to this system is

$$\boldsymbol{x}(t) = e^{A(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s) \tag{1.129}$$

and has the distribution

$$\boldsymbol{x}(t) \sim N(\bar{x}(t), P(t)) \tag{1.130}$$

with the mean value vector

$$\begin{aligned}
\bar{x}(t) &= E\{\boldsymbol{x}(t)\}\\
&= E\left\{e^{A(t-t_0)}\boldsymbol{x}(t_0) + \int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right\}\\
&= e^{A(t-t_0)}E\{\boldsymbol{x}(t_0)\} + E\left\{\int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right\}\\
&= e^{A(t-t_0)}\bar{x}(t_0)
\end{aligned} \tag{1.131}$$

and the covariance matrix

$$
\begin{aligned}
P(t) &= E\left\{(\boldsymbol{x}(t)-\bar{x}(t))(\boldsymbol{x}(t)-\bar{x}(t))'\right\} \\
&= e^{A(t-t_0)}E\left\{(\boldsymbol{x}(t_0)-\bar{x}(t_0))(\boldsymbol{x}(t_0)-\bar{x}(t_0))'\right\}e^{A'(t-t_0)} \\
&\qquad + e^{A(t-t_0)}E\left\{(\boldsymbol{x}(t_0)-\bar{x}(t_0))\left(\int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right)'\right\} \\
&\qquad + E\left\{\left(\int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right)(\boldsymbol{x}(t_0)-\bar{x}(t_0))'\right\}e^{A'(t-t_0)} \\
&\qquad + E\left\{\left(\int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right)\left(\int_{t_0}^{t} e^{A(t-s)}Gd\boldsymbol{w}(s)\right)'\right\} \\
&= e^{A(t-t_0)}P(t_0)e^{A'(t-t_0)} + \int_{t_0}^{t} e^{A(t-s)}GG'e^{A'(t-s)}ds
\end{aligned}
\tag{1.132}
$$

Consider the evolution of the system between two sampling instants, t_k and $t_{k+1} = t_k + T_s$. Let $\boldsymbol{x}(t_k) = \boldsymbol{x}_k$, $\bar{x}(t_k) = \bar{x}_k$, and $P(t_k) = P_k$. The solution can be expressed as

$$
\boldsymbol{x}(t_{k+1}) = e^{A(t_{k+1}-t_k)}\boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s) \tag{1.133}
$$

or

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= e^{AT_s}\boldsymbol{x}_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s) \\
&= e^{AT_s}\boldsymbol{x}_k + \boldsymbol{w}_k
\end{aligned}
\tag{1.134}
$$

with

$$
\boldsymbol{w}_k = \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s) \tag{1.135}
$$

Note that $\boldsymbol{w}_k \sim N_{iid}(0, \bar{Q})$ with

$$
\begin{aligned}
\bar{Q} &= E\left\{\boldsymbol{w}_k\boldsymbol{w}_k'\right\} \\
&= E\left\{\left(\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s)\right)\left(\int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s)\right)'\right\} \\
&= \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}GG'e^{A'(t_{k+1}-s)}ds \\
&= \int_0^{T_s} e^{A\tau}GG'e^{A'\tau}d\tau
\end{aligned}
\tag{1.136}
$$

Consequently, the evolution of the mean and the covariance from one sampling instant to the next may be expressed as

$$
\bar{x}_{k+1} = e^{AT_s}\bar{x}_k = \bar{A}\bar{x}_k \tag{1.137a}
$$

$$
P_{k+1} = e^{AT_s}P_ke^{A'T_s} + \int_0^{T_s} e^{A\tau}GG'e^{A'\tau}d\tau = \bar{A}P_k\bar{A}' + \bar{Q} \tag{1.137b}
$$

Algorithm 10 Computation of $(\bar{A}, \bar{Q})$ in (1.138) using (1.140).

```
function [Abar,Qbar] = c2dsw(A,G,Ts)

nx = size(A,1);
nx1 = nx+1;
nx2 = nx+nx;
M = [-A' G*G'; zeros(nx,nx) A]*Ts;
Phi = expm(M);
Abar = Phi(nx1:nx2,nx1:nx2)';
Qbar = Abar*Phi(1:nx,nx1:nx2);
```

with

$$\bar{A} = e^{AT_s} \tag{1.138a}$$

$$\bar{Q} = \int_0^{T_s} e^{A\tau} GG' e^{A'\tau} d\tau \tag{1.138b}$$

In conclusion the evolution of the linear stochastic differential equation (1.128) between sampling instants may be computed by the linear stochastic difference equation

$$\boldsymbol{x}_{k+1} = \bar{A}\boldsymbol{x}_k + \boldsymbol{w}_k \qquad \boldsymbol{x}_0 \sim N(\bar{x}_0, P_0),\ \ \boldsymbol{w}_k \sim N_{iid}(0, \bar{Q}) \tag{1.139}$$

Using the results in Appendix 2, the matrices $\bar{A}$ and $\bar{Q}$ may be computed by

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \exp\left(\begin{bmatrix} -A & GG' \\ 0 & A' \end{bmatrix} T_s \right) \tag{1.140a}$$

$$\bar{A} = \Phi_{22}' \tag{1.140b}$$

$$\bar{Q} = \Phi_{22}' \Phi_{12} = \bar{A}\Phi_{12} \tag{1.140c}$$

in which exp denotes the matrix exponential function. Algorithm 10 specifies a Matlab implementation of ...

Consider the continuous-time linear stochastic differential equation

$$d\boldsymbol{x}(t) = (A\boldsymbol{x}(t) + Bu(t))\, dt + G d\boldsymbol{w}(t) \tag{1.141a}$$

in which $\boldsymbol{x}(t_0) \sim N(\bar{x}_0, P_0)$, $\{\boldsymbol{w}(t)\}$ is a standard Wiener process, and the exogenous input, $u(t)$, is piecewise constant

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} \quad t_{k+1} = t_k + T_s \tag{1.141b}$$

The solution to this system is

Algorithm 11 $(\bar{A}, \bar{B}, \bar{Q})$ in (1.143) such that it is equal to (1.141a).

```
function [Abar,Bbar,Qbar] = c2dzohsw(A,B,G,Ts)

[Abar, Bbar] = c2dzoh(A,B,Ts);
[~,    Qbar] = c2dsw(A,G,Ts);
```

$$\begin{aligned}\boldsymbol{x}(t_{k+1}) = e^{A(t_{k+1}-t_k)}\boldsymbol{x}(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Bu(s)ds \\ + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Gd\boldsymbol{w}(s)\end{aligned} \tag{1.142}$$

or

$$\boldsymbol{x}_{k+1} = \bar{A}\boldsymbol{x}_k + \bar{B}u_k + \boldsymbol{w}_k \qquad \boldsymbol{x}_0 \sim N(\bar{x}_0, P_0),\, \boldsymbol{w}_k \sim N_{iid}(0, \bar{Q}) \tag{1.143}$$

with

$$\bar{A} = e^{AT_s} \tag{1.144a}$$

$$\bar{B} = \int_0^{T_s} e^{A\tau} d\tau B \tag{1.144b}$$

$$\bar{Q} = \int_0^{T_s} e^{A\tau} GG' e^{A'\tau} d\tau \tag{1.144c}$$

$\bar{A}$, $\bar{B}$, and $\bar{Q}$ may be computed by the following procedure

$$\begin{bmatrix} \bar{A} & \bar{B} \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T_s \right) \tag{1.145a}$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \exp\left(\begin{bmatrix} -A & GG' \\ 0 & A' \end{bmatrix} T_s \right) \tag{1.145b}$$

$$\bar{Q} = \Phi_{22}'\Phi_{12} \tag{1.145c}$$

Algorithm 11 specifies a Matlab implementation function for computation of $(\bar{A}, \bar{B}, \bar{Q})$ in the stochastic difference equation (1.143) such that it is equivalent to the stochastic differential equation (1.141a) when the deterministic inputs, $u(t)$, are zero-order-hold parameterized and the stochastic process, $\boldsymbol{w}(t)$, is a standard Wiener process.

These results lead us to the main result for continuous-discrete linear stochastic systems. A continuous-discrete system is a system in which a continuous-time physical process is sampled at discrete times, $t_k = t_0 + kT_s$, and manipulated at discrete times. Physical process controlled by computers is typically continuous-discrete systems. The process evolves in continuous-time, while the control algorithm in the computer is executed periodically, i.e. at discrete times with a fixed period.

Consider the continuous-discrete linear stochastic system

$$d\boldsymbol{x}(t) = (A\boldsymbol{x}(t) + Bu(t))\,dt + Gd\boldsymbol{w}(t) \tag{1.146a}$$
$$\boldsymbol{y}(t_k) = C\boldsymbol{x}(t_k) + \boldsymbol{v}(t_k) \tag{1.146b}$$
$$\boldsymbol{z}(t_k) = C_z\boldsymbol{x}(t_k) \tag{1.146c}$$

with $\boldsymbol{x}_0$, $\{\boldsymbol{w}(t)\}$, and $\boldsymbol{v}(t_k)$ being stochastic variables and processes with the distributions

$$\boldsymbol{x}_0 \sim N(\bar{x}_0, P_0) \tag{1.146d}$$
$$\{\boldsymbol{w}(t)\} \quad \text{standard Wiener process} \tag{1.146e}$$
$$\boldsymbol{v}(t_k) = \boldsymbol{v}_k \sim N_{iid}(0, R) \tag{1.146f}$$

and the controlled input being constant in each sampling period

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} = t_k + T_s \tag{1.146g}$$

Let $\boldsymbol{x}_k = \boldsymbol{x}(t_k)$, $\boldsymbol{y}_k = \boldsymbol{y}(t_k)$, $\boldsymbol{z}_k = \boldsymbol{z}(t_k)$. Then the following system of stochastic linear difference equations is equivalent to (1.146) at the discrete times $\{t_k, k \in \mathbb{N}_0\}$

$$\boldsymbol{x}_{k+1} = \bar{A}\boldsymbol{x}_k + \bar{B}u_k + \boldsymbol{w}_k \quad \boldsymbol{x}_0 \sim N(\bar{x}_0, P_0),\ \boldsymbol{w}_k \sim N_{iid}(0, \bar{Q}) \tag{1.147a}$$
$$\boldsymbol{y}_k = C\boldsymbol{x}_k + \boldsymbol{v}_k \qquad \boldsymbol{v}_k \sim N_{iid}(0, R) \tag{1.147b}$$
$$\boldsymbol{z}_k = C_z\boldsymbol{x}_k \tag{1.147c}$$

with $(\bar{A}, \bar{B}, \bar{Q})$ determined by (1.144).

1.5 Conclusion

In this chapter, we have covered modeling and simulation relevant to model predictive control using the quadruple tank process as case study.

1. Describe the physical system. What is the manipulated variables (MVs), the measured disturbance variables (DVs), the controlled variables (CVs), the measurements, the process noise, and the measurement noise.
2. Model the system using mass balances and physical laws. The resulting model is a system of first order ordinary differential equations.
3. Simulate the system using an ODE-solver. In Matlab `ode15s` can be used for stiff systems as well as non-stiff systems. However, `ode45` is faster and more appropriate for non-stiff systems.
4. Discrete-time simulation.
5. Stochastic simulation.
6. Steady state. In Matlab `fsolve` may be used to determine the steady state.
7. Linearization around a steady state.
8. Discretization of the linear deterministic and stochastic systems.

1.6 Notes

Johansson (2000) introduced the four tank system for control studies. The four tank system used in this paper corresponds to the experimental plant available at the Chemical Engineering Department at the Technical University of Denmark.

Åström (1970) provides an introduction stochastic models and control. He explains why continuous-time stochastic processes cannot be obtained as straightforward extension of deterministic processes with a stochastic part.

1.7 Exercises

Part II

Appendices

2

Continuous-to-Discrete Time Conversion

The topic of this chapter is conversion of continuous-time linear time invariant state space models to discrete-time linear time invariant state space models. The matrix exponential function plays a central role in this conversion. First, the matrix exponential function is introduced. Secondly, we demonstrate the application of the matrix exponential function for conversion of continuous-time linear time invariant models with zero-order-hold inputs to discrete-time linear time invariant state space models. Thirdly, we use the matrix exponential function in algorithm for computation of an discrete-time LQ problem that is equivalent to a continuous-time LQ problem with zero-order-hold inputs.

2.1 The Matrix Exponential Function

The exponential function plays a central role in solution of linear differential equations. The exponential function is defined such that it is the solution of a certain linear differential equation. In this, section we generalize the scalar exponential function to the matrix case.

2.1.1 Scalar Case

Let $a \in \mathbb{R}$. Analytically, the exponential function is defined as

$$\exp(a) = e^a = \sum_{k=0}^{\infty} \frac{a^k}{k!} = 1 + a + \frac{1}{2}a^2 + \frac{1}{6}a^3 + \dots \tag{2.1}$$

Then note that

$$x(t) = e^{at} = \sum_{k=0}^{\infty} \frac{(at)^k}{k!}$$

implies

$$x(0) = e^{a0} = \sum_{k=0}^{\infty} \frac{(a \cdot 0)^k}{k!} = 1 + \sum_{k=1}^{\infty} \frac{(a \cdot 0)^k}{k!} = 1$$

and

$$\begin{aligned}\dot{x}(t) = \frac{dx(t)}{dt} &= \frac{d}{dt}\left(e^{at}\right) = \frac{d}{dt}\left(\sum_{k=0}^{\infty} \frac{(at)^k}{k!}\right) = \sum_{k=0}^{\infty} \frac{a^k}{k!}\frac{dt^k}{dt} = \sum_{k=1}^{\infty} \frac{a^k}{k!} k t^{k-1} \\ &= a \sum_{k=1}^{\infty} \frac{a^{k-1}t^{k-1}}{(k-1)!} = a \sum_{k=0}^{\infty} \frac{(at)^k}{k!} = a e^{at} = a x(t)\end{aligned}$$

Therefore, by definition we may consider the exponential function as the solution

$$x(t) = e^{at} = \exp(at) \tag{2.2}$$

to the linear initial value problem

$$\dot{x}(t) = ax \quad x(0) = 1 \qquad x(\cdot) : \mathbb{R} \mapsto \mathbb{R} \tag{2.3}$$

It is this connection that makes the exponential function so important. Without proof we state the following properties of the exponential function

$$e^{at} > 0 \tag{2.4a}$$

$$e^{a(t+s)} = e^{at} e^{as} \tag{2.4b}$$

$$e^{-at} = \frac{1}{e^{at}} \tag{2.4c}$$

$$(e^{at})^s = e^{a(ts)} \tag{2.4d}$$

$$\frac{de^{at}}{dt} = a e^{at} \tag{2.4e}$$

2.1.2 Generalization to the Matrix Case

Let $A \in \mathbb{R}^{n \times n}$. The matrix exponential function is defined as

$$e^A = \exp(A) = \sum_{k=0}^{\infty} \frac{A^k}{k!} = I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + \dots \tag{2.5}$$

Note that $e^A \in \mathbb{R}^{n \times n}$. Let $X : \mathbb{R} \mapsto \mathbb{R}^{n \times n}$ be defined as $X(t) = \exp(At)$. The definition of the matrix exponential function implies that

$$X(t) = e^{At} = \exp(At) = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} = I + \sum_{k=1}^{\infty} \frac{(At)^k}{k!} \tag{2.6a}$$

$$X(0) = e^{A0} = \exp(A0) = I + \sum_{k=1}^{\infty} \frac{(A \cdot 0)^k}{k!} = I \tag{2.6b}$$

$$\begin{aligned} \dot{X}(t) &= \frac{dX(t)}{dt} = \frac{d}{dt}\left(e^{At}\right) = \frac{d}{dt}\left(I + \sum_{k=1}^{\infty} \frac{(At)^k}{k!}\right) = \sum_{k=1}^{\infty} \frac{A^k k t^{k-1}}{k!} \\ &= A \sum_{k=1}^{\infty} \frac{(At)^{k-1}}{(k-1)!} = A \sum_{k=0}^{\infty} \frac{(At)^k}{k!} = Ae^{At} = AX(t) \end{aligned} \tag{2.6c}$$

Let $X \in \mathbb{R}^{n \times n}$ and $A \in \mathbb{R}^{n \times n}$. Consequently, the matrix differential equation

$$\dot{X}(t) = \frac{dX(t)}{dt} = AX(t) \qquad X(0) = I \tag{2.7}$$

has the solution

$$X(t) = \exp(At) = \sum_{k=0}^{\infty} \frac{(At)^k}{k!} \tag{2.8}$$

It is this relation of the matrix exponential function to the solution of (2.7) that makes the matrix exponential function important.

We list the following properties of the matrix exponential function

$$e^{At} > 0 \quad \text{(positive definite)} \tag{2.9a}$$

$$e^{A(t+s)} = e^{At} e^{As} \tag{2.9b}$$

$$\left(e^{At}\right)^{-1} = e^{-At} \tag{2.9c}$$

$$\left(e^{At}\right)^{s} = e^{A(ts)} \tag{2.9d}$$

$$\frac{de^{At}}{dt} = Ae^{At} \tag{2.9e}$$

$$e^{A't} = \left(e^{At}\right)' \tag{2.9f}$$

2.1.3 Integrals Involving the Matrix Exponential Function

Consider the matrix differential equation

$$\dot{X}(t) = AX(t) \quad X(0) = I \tag{2.10}$$

with $X \in \mathbb{R}^{(n+m)\times(n+m)}$ and $A \in \mathbb{R}^{(n+m)\times(n+m)}$. Let A have the structure

$$A = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \qquad A_{11} \in \mathbb{R}^{n\times n}, A_{12} \in \mathbb{R}^{n\times m}, A_{22} \in \mathbb{R}^{m\times m} \tag{2.11}$$

Then A^k is also block upper triangular and the solution $X(t)$ has the structure

$$
\begin{aligned}
X(t) &= \exp(At) \\
&= \sum_{k=0}^{\infty} \frac{(At)^k}{k!} = \sum_{k=0}^{\infty} \frac{\left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} t\right)^k}{k!} = \begin{bmatrix} X_{11}(t) & X_{12}(t) \\ 0 & X_{22}(t) \end{bmatrix}
\end{aligned} \tag{2.12}
$$

Therefore

$$
\begin{aligned}
\frac{d}{dt}\begin{bmatrix} X_{11}(t) & X_{12}(t) \\ 0 & X_{22}(t) \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}\begin{bmatrix} X_{11}(t) & X_{12}(t) \\ 0 & X_{22}(t) \end{bmatrix} \\
&= \begin{bmatrix} A_{11}X_{11}(t) & A_{11}X_{11}(t) + A_{12}X_{12}(t) \\ 0 & A_{22}X_{22}(t) \end{bmatrix}
\end{aligned} \tag{2.13a}
$$

with the initial condition

$$
X(0) = \begin{bmatrix} X_{11}(0) & X_{12}(0) \\ 0 & X_{22}(0) \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I \tag{2.13b}
$$

This system of matrix differential equation (2.13) may be expressed as

$$
\dot{X}_{11}(t) = A_{11}X_{11}(t) \qquad X_{11}(0) = I \tag{2.14a}
$$

$$
\dot{X}_{22}(t) = A_{22}X_{22}(t) \qquad X_{22}(0) = I \tag{2.14b}
$$

$$
\dot{X}_{12}(t) = A_{11}X_{12}(t) + A_{12}X_{22}(t) \qquad X_{12}(0) = 0 \tag{2.14c}
$$

The solution of (2.14a) and (2.14b) are, respectively:

$$
X_{11}(t) = \exp(A_{11}t) = e^{A_{11}t} \tag{2.15a}
$$

$$
X_{22}(t) = \exp(A_{22}t) = e^{A_{22}t} \tag{2.15b}
$$

The solution of (2.14c) is a non-homogeneous system of linear differential equations and has the solution

$$
\begin{aligned}
X_{12}(t) &= e^{A_{11}t}X_{12}(0) + \int_0^t e^{A_{11}(t-s)}A_{12}X_{22}(s)ds \\
&= \int_0^t e^{A_{11}(t-s)}A_{12}e^{A_{22}s}ds
\end{aligned} \tag{2.15c}
$$

Consequently, we may compute the expressions (2.15) by

$$
\begin{bmatrix} e^{A_{11}t} & \int_0^t e^{A_{11}(t-s)}A_{12}e^{A_{22}s}ds \\ 0 & e^{A_{22}t} \end{bmatrix} = \begin{bmatrix} X_{11}(t) & X_{12}(t) \\ 0 & X_{22}(t) \end{bmatrix} = \exp\left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{12} \end{bmatrix} t\right) \tag{2.16}
$$

This formula is useful in two cases related to conversion of a continuous-time linear system to a discrete-time system when the inputs are zero-order-hold.

Case I

Let $A_{11} = A$, $A_{12} = B$ and $A_{22} = 0$. Then

$$e^{A_{11}t} = e^{At} \tag{2.17a}$$

$$e^{A_{22}t} = e^{0t} = I \tag{2.17b}$$

$$\int_0^t e^{A_{11}(t-s)} A_{12} e^{A_{22}s} ds = \int_0^t e^{A(t-s)} B e^{0s} ds = \int_0^t e^{A\tau} d\tau B \tag{2.17c}$$

Consequently

$$\begin{bmatrix} e^{At} & \int_0^t e^{A\tau} d\tau B \\ 0 & I \end{bmatrix} = \begin{bmatrix} \bar{A}(t) & \bar{B}(t) \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} t \right) \tag{2.18}$$

Case II

Let $A_{11} = -A'$, $A_{12} = Q$ and $A_{22} = A$. Then

$$e^{A_{11}t} = e^{-A't} \tag{2.19a}$$

$$e^{A_{22}t} = e^{At} \tag{2.19b}$$

$$\begin{aligned} \int_0^t e^{A_{11}(t-s)} A_{12} e^{A_{22}s} ds &= \int_0^t e^{-A'(t-s)} Q e^{As} ds \\ &= e^{-A't} \int_0^t e^{A'\tau} Q e^{A\tau} d\tau \end{aligned} \tag{2.19c}$$

and

$$\begin{bmatrix} e^{-A't} & e^{-A't} \int_0^t e^{A'\tau} Q e^{A\tau} d\tau \\ 0 & e^{At} \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \exp\left(\begin{bmatrix} -A' & Q \\ 0 & A \end{bmatrix} t \right) \tag{2.20}$$

This implies

$$e^{At} = \bar{A}(t) = \Phi_{22} \tag{2.21a}$$

$$\int_0^t e^{A'\tau} Q e^{A\tau} d\tau = \int_0^t \bar{A}(\tau)' Q \bar{A}(\tau) d\tau = \Phi_{22}' \Phi_{12} \tag{2.21b}$$

2.2 Continuous and Discrete Time LTI State Space Models

In this section we consider the conversion of the system of linear time invariant *differential* equations

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(t_0) = x_0 \tag{2.22}$$

to the equivalent system of linear time invariant *difference* equations

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \tag{2.23}$$

This corresponds to conversion from a continuous-time representation to an equivalent discrete-time representation. Let T_s be the sampling time and $t_k = t_0 + kT_s$ for $k \geq 0$. Then $x_k = x(t_k) = x(t_0 + kT_s)$. We do assume that the input $u(t)$ is parameterized as a zero-order-hold (ZOH)

$$u(t) = u_k \quad t_k \leq t < t_{k+1} \quad k = 0, 1, 2, \ldots \tag{2.24}$$

The discrete time matrices $(\bar{A}, \bar{B})$ are functions of the sampling time T_s. To derive these relations consider the solution to (2.22)

$$x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-s)}Bu(s)ds \tag{2.25}$$

which leads to

$$\begin{aligned} x_{k+1} = x(t_{k+1}) &= e^{A(t_{k+1}-t_k)}x_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}Bu(s)ds \\ &= e^{AT_s}x_k + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s)}dsBu_k \\ &= e^{AT_s}x_k + \int_{0}^{T_s} e^{A\tau}d\tau Bu_k \\ &= \bar{A}x_k + \bar{B}u_k \end{aligned} \tag{2.26}$$

with

$$\bar{A} = \bar{A}(T_s) = \exp(AT_s) \tag{2.27a}$$

$$\bar{B} = \bar{B}(T_s) = \int_0^{T_s} \exp(A\tau)d\tau B \tag{2.27b}$$

The matrices $\bar{A}$ and $\bar{B}$ may be computed using the matrix exponential function for the expression

$$\begin{bmatrix} \bar{A}(T_s) & \bar{B}(T_s) \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} T_s \right) \tag{2.28}$$

2.3 ZOH Continuous-Time LQ to Discrete-Time LQ

In this section we consider conversion of a continuous-time linear-quadratic control problem with a zero-order-hold input to an equivalent discrete-time linear-quadratic control problem.

The continuous-time LQ problem has the cost function

$$\phi = \int_{t_0}^{t_N} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \tag{2.29a}$$

subject to the linear differential equation

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x(t_0) = x_0 \quad t_0 \leq t \leq t_N \tag{2.29b}$$

and zero-order-hold discretization of the input

$$u(t) = u_k \qquad t_k \leq t < t_{k+1} \quad k = 0, 1, \dots, N-1 \tag{2.29c}$$

The sampling period is assumed constant, i.e. T_s, such that $t_k = t_0 + kT_s$ for $k = 0, 1, \dots, N$ and $x_k = x(t_k)$.

The linear differential equation (2.29b) and the zero-order-hold parametrization (2.29c) yield

$$\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} \bar{A}(t-t_k) & \bar{B}(t-t_k) \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \quad t_k \leq t < t_{k+1} \quad k = 0, 1, \dots, N-1 \tag{2.30}$$

in which

$$\begin{bmatrix} \bar{A}(t-t_k) & \bar{B}(t-t_k) \\ 0 & I \end{bmatrix} = \begin{bmatrix} e^{A(t-t_k)} & \int_{t_k}^{t} e^{A(t-s)} ds B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} (t - t_k) \right) \tag{2.31}$$

for $t_k \leq t < t_{k+1}$. Define $\tau = t - t_k$ and note $0 \leq \tau < t_{k+1} - t_k = T_s$. Then

$$\begin{aligned} \begin{bmatrix} \bar{A}(t-t_k) & \bar{B}(t-t_k) \\ 0 & I \end{bmatrix} &= \begin{bmatrix} \bar{A}(\tau) & \bar{B}(\tau) \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} e^{A\tau} & \int_0^{\tau} e^{As} ds B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) \end{aligned} \tag{2.32}$$

Using the relations (2.30) and (2.32), the cost function (2.29a) may be expressed as

$$\begin{aligned} \phi &= \int_{t_0}^{t_N} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \\ &= \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \\ &= \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} \bar{A}(t-t_k) & \bar{B}(t-t_k) \\ 0 & I \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} \bar{A}(t-t_k) & \bar{B}(t-t_k) \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} dt \\ &= \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \left(\int_0^{T_s} \begin{bmatrix} \bar{A}(\tau) & \bar{B}(\tau) \\ 0 & I \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} \bar{A}(\tau) & \bar{B}(\tau) \\ 0 & I \end{bmatrix} d\tau \right) \begin{bmatrix} x_k \\ u_k \end{bmatrix} \\ &= \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \end{aligned} \tag{2.33}$$

in which

$$\begin{aligned}\begin{bmatrix}\bar{Q} & \bar{M}\\ \bar{M}' & \bar{R}\end{bmatrix} &= \int_0^{T_s} \begin{bmatrix}\bar{A}(\tau) & \bar{B}(\tau)\\ 0 & I\end{bmatrix}' \begin{bmatrix}Q & M\\ M' & R\end{bmatrix} \begin{bmatrix}\bar{A}(\tau) & \bar{B}(\tau)\\ 0 & I\end{bmatrix} d\tau\\ &= \int_0^{T_s} \exp\left(\begin{bmatrix}A & B\\ 0 & 0\end{bmatrix}' \tau\right) \begin{bmatrix}Q & M\\ M' & R\end{bmatrix} \exp\left(\begin{bmatrix}A & B\\ 0 & 0\end{bmatrix} \tau\right) d\tau\end{aligned} \tag{2.34}$$

Consequently, the continuous-time LQ problem with ZOH parameterized inputs may be expressed as the following equivalent discrete-time LQ problem. The discrete-time LQ problem is

$$\phi = \sum_{k=0}^{N-1} \begin{bmatrix}x_k\\ u_k\end{bmatrix}' \begin{bmatrix}\bar{Q} & \bar{M}\\ \bar{M}' & \bar{R}\end{bmatrix} \begin{bmatrix}x_k\\ u_k\end{bmatrix} \tag{2.35a}$$

subject to

$$x_{k+1} = \bar{A}x_k + \bar{B}u_k \quad k = 0, 1, \ldots, N-1 \tag{2.35b}$$

with parameter matrices defined as

$$\bar{A} = \exp(AT_s) \tag{2.36a}$$

$$\bar{B} = \int_0^{T_s} e^{A\tau} d\tau B \tag{2.36b}$$

$$\begin{bmatrix}\bar{Q} & \bar{M}\\ \bar{M}' & \bar{R}\end{bmatrix} = \int_0^{T_s} \exp\left(\begin{bmatrix}A & B\\ 0 & 0\end{bmatrix}' \tau\right) \begin{bmatrix}Q & M\\ M' & R\end{bmatrix} \exp\left(\begin{bmatrix}A & B\\ 0 & 0\end{bmatrix} \tau\right) d\tau \tag{2.36c}$$

The matrix parameters (2.36) are computed using the procedure described by (2.20) and (2.21), i.e.

$$\begin{bmatrix}\Phi_{11} & \Phi_{12}\\ 0 & \Phi_{22}\end{bmatrix} = \exp\left(\left[\begin{array}{cc|cc}-A' & 0 & Q & M\\ -B' & 0 & M' & R\\ \hline 0 & 0 & A & B\\ 0 & 0 & 0 & 0\end{array}\right] T_s\right) \tag{2.37a}$$

$$\begin{bmatrix}\bar{A} & \bar{B}\\ 0 & I\end{bmatrix} = \Phi_{22} \tag{2.37b}$$

$$\begin{bmatrix}\bar{Q} & \bar{M}\\ \bar{M}' & \bar{R}\end{bmatrix} = \Phi_{22}'\Phi_{12} \tag{2.37c}$$

Note that in general, $\bar{M} \neq 0$ even though $M = 0$.

2.4 Output Regulation

In this section we use previous results to convert the continuous-time output regulation problem with zero-order-hold parametrization of the inputs and the references to an equivalent discrete-time regulation problem.

The continuous-time output regulation problem has the cost function

$$\phi = \frac{1}{2} \int_{t_0}^{t_N} (y(t) - r(t))' Q_y (y(t) - r(t)) dt \tag{2.38a}$$

in which $Q_y \in \mathbb{R}^{n \times n}$ is assumed to be symmetric and positive semi-definite. $y(t) \in \mathbb{R}^p$ is the outputs and $r(t) \in \mathbb{R}^p$ is the reference signal (set point) at time t. The dynamic evolution of the system is governed by the continuous-time linear time invariant state space model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.38b}$$
$$y(t) = Cx(t) + Du(t) \tag{2.38c}$$

in which the $x(t) \in \mathbb{R}^n$ is the state vector and $u(t) \in \mathbb{R}^m$ is the input vector. The inputs and the references are assumed to be constant in each sample period, i.e.

$$u(t) = u_k \qquad t_k \le t < t_{k+1} \quad k = 0, 1, \ldots, N-1 \tag{2.38d}$$
$$r(t) = r_k \qquad t_k \le t < t_{k+1} \quad k = 0, 1, \ldots, N-1 \tag{2.38e}$$

The system has a constant sample time, T_s. This implies $t_k = t_0 + kT_s$ for $k = 0, 1, \ldots, N$.

The cost function (2.38a) may be expressed as

$$\begin{aligned}
\phi &= \frac{1}{2} \int_{t_0}^{t_N} (y(t) - r(t))' Q_y (y(t) - r(t)) dt \\
&= \underbrace{\frac{1}{2} \int_{t_0}^{t_N} y(t)' Q_y y(t) dt}_{=\phi_2} + \underbrace{\int_{t_0}^{t_N} [-Q_y r(t)]' y(t) dt}_{=\phi_1} + \underbrace{\frac{1}{2} \int_{t_0}^{t_N} r(t)' Q_y r(t) dt}_{=\phi_0}
\end{aligned} \tag{2.39}$$

The output, $y(t)$, is

$$y(t) = Cx(t) + Du(t) = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \tag{2.40}$$

Therefore

$$\begin{aligned}
\phi_2 &= \frac{1}{2} \int_{t_0}^{t_N} y(t)' Q_y y(t) dt \\
&= \frac{1}{2} \int_{t_0}^{t_N} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} C & D \end{bmatrix}' Q_y \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \\
&= \frac{1}{2} \int_{t_0}^{t_N} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt
\end{aligned} \tag{2.41}$$

in which

$$\begin{bmatrix} Q & M \\ M' & R \end{bmatrix} = \begin{bmatrix} C & D \end{bmatrix}' Q_y \begin{bmatrix} C & D \end{bmatrix} = \begin{bmatrix} C'Q_yC & C'Q_yD \\ D'Q_yC & D'Q_yD \end{bmatrix} \tag{2.42}$$

Then the state and input may be expressed as

$$\begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} x(t_k+\tau) \\ u(t_k+\tau) \end{bmatrix} = \begin{bmatrix} \bar{A}(\tau) & \bar{B}(\tau) \\ 0 & I \end{bmatrix} = \begin{bmatrix} e^{A\tau} & \int_0^\tau e^{As} ds B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) \tag{2.43}$$

for $t = t_k + \tau$, $0 \leq \tau < T_s$, and $k = 0, 1, \ldots, N-1$. Consequently using the derivation in Section 2.3, ϕ_2 may be expressed as

$$\begin{aligned} \phi_2 &= \frac{1}{2}\int_{t_0}^{t_N} y(t)'Q_y y(t) dt = \frac{1}{2}\int_{t_0}^{t_N} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}' \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \\ &= \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \end{aligned} \tag{2.44}$$

with

$$\begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} = \int_0^{T_s} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}' \tau \right) \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) d\tau \tag{2.45}$$

The term ϕ_1 in the cost function, ϕ, may be expressed as

$$\begin{aligned} \phi_1 &= \int_{t_0}^{t_N} \left[-Q_y r(t)\right]' y(t) dt = \int_{t_0}^{t_N} \left[-Q_y r(t)\right]' \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \\ &= \sum_{k=0}^{N-1} \left\{ \left(- \begin{bmatrix} C & D \end{bmatrix}' Q_y r_k \right)' \int_{t_k}^{t_{k+1}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \right\} \end{aligned} \tag{2.46}$$

in which

$$\begin{aligned} \int_{t_k}^{t_{k+1}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt &= \int_0^{T_s} \begin{bmatrix} x(t_k+\tau) \\ u(t_k+\tau) \end{bmatrix} d\tau = \int_0^{T_s} \begin{bmatrix} \bar{A}(\tau) & \bar{B}(\tau) \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} d\tau \\ &= \int_0^{T_s} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) d\tau \begin{bmatrix} x_k \\ u_k \end{bmatrix} = \varPhi_1 \begin{bmatrix} x_k \\ u_k \end{bmatrix} \end{aligned} \tag{2.47}$$

and

$$\varPhi_1 = \int_0^{T_s} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) d\tau \tag{2.48}$$

ϕ_1 is therefore

$$\begin{aligned} \phi_1 &= \sum_{k=0}^{N-1} \left\{ \left(- \begin{bmatrix} C & D \end{bmatrix}' Q_y r_k \right)' \int_{t_k}^{t_{k+1}} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \right\} \\ &= \sum_{k=0}^{N-1} \left\{ \left(- \begin{bmatrix} C & D \end{bmatrix}' Q_y r_k \right)' \varPhi_1 \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\} \\ &= \sum_{k=0}^{N-1} \begin{bmatrix} \bar{q}_k \\ \bar{s}_k \end{bmatrix}' \begin{bmatrix} x_k \\ u_k \end{bmatrix} \end{aligned} \tag{2.49}$$

with

$$\begin{bmatrix} \bar{q}_k \\ \bar{s}_k \end{bmatrix} = \left[-\left(\begin{bmatrix} C & D \end{bmatrix} \Phi_1 \right)' Q_y \right] r_k = \Gamma r_k \quad \Gamma = -\left(\begin{bmatrix} C & D \end{bmatrix} \Phi_1 \right)' Q_y \tag{2.50}$$

Consequently, the cost function in the discrete-time LQ problem equivalent to (2.38a) is

$$\phi = \sum_{k=0}^{N-1} \left\{ \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}' \begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} \bar{q}_k \\ \bar{s}_k \end{bmatrix}' \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right\} + \phi_0 \tag{2.51a}$$

and the corresponding state evolution is governed by the difference equation

$$x_{k+1} = \bar{A} x_k + \bar{B} u_k \qquad k = 0, 1, \ldots, N-1 \tag{2.51b}$$

The matrix parameters in this discrete-time LQ problem are related to the continuous-time LQ problem parameters by

$$\begin{bmatrix} Q & M \\ M' & R \end{bmatrix} = \begin{bmatrix} C & D \end{bmatrix}' Q_y \begin{bmatrix} C & D \end{bmatrix} \tag{2.52a}$$

$$\bar{A} = \exp(A T_s) \tag{2.52b}$$

$$\bar{B} = \int_0^{T_s} \exp(A\tau) d\tau B \tag{2.52c}$$

$$\begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} = \int_0^{T_s} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix}' \tau \right) \begin{bmatrix} Q & M \\ M' & R \end{bmatrix} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) d\tau \tag{2.52d}$$

$$\Phi_1 = \int_0^{T_s} \exp\left(\begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \tau \right) d\tau \tag{2.52e}$$

$$\Gamma = -\left(\begin{bmatrix} C & D \end{bmatrix} \Phi_1 \right)' Q_y \tag{2.52f}$$

$$\begin{bmatrix} \bar{q}_k \\ \bar{s}_k \end{bmatrix} = \Gamma r_k \qquad k = 0, 1, \ldots, N-1 \tag{2.52g}$$

and the inconsequential zero order term in the cost function is

$$\phi_0 = \frac{1}{2} \int_{t_0}^{t_N} r(t)' Q_y r(t) dt = \frac{T_s}{2} \sum_{k=0}^{N-1} r_k' Q_y r_k \tag{2.52h}$$

The expressions in (2.52) may be computed using the following procedure

$$\begin{bmatrix} Q & M \\ M' & R \end{bmatrix} = \begin{bmatrix} C & D \end{bmatrix}' Q_y \begin{bmatrix} C & D \end{bmatrix} \tag{2.53a}$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \exp\left(\left[\begin{array}{cc|cc} -A' & 0 & Q & M \\ -B' & 0 & M' & R \\ \hline 0 & 0 & A & B \\ 0 & 0 & 0 & 0 \end{array} \right] T_s \right) \tag{2.53b}$$

$$\begin{bmatrix} \bar{A} & \bar{B} \\ 0 & I \end{bmatrix} = \Phi_{22} \tag{2.53c}$$

$$\begin{bmatrix} \bar{Q} & \bar{M} \\ \bar{M}' & \bar{R} \end{bmatrix} = \Phi_{22}' \Phi_{12} \tag{2.53d}$$

$$\begin{bmatrix} * & \Phi_1 \\ 0 & I \end{bmatrix} = \exp\left(\left[\begin{array}{cc|cc} A & B & I & 0 \\ 0 & 0 & 0 & I \\ \hline 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right] T_s \right) \tag{2.53e}$$

$$\Gamma = -\left(\begin{bmatrix} C & D \end{bmatrix} \Phi_1 \right)' Q_y \tag{2.53f}$$

$$\begin{bmatrix} \bar{q}_k \\ \bar{s}_k \end{bmatrix} = \Gamma r_k \tag{2.53g}$$

2.5 Notes

Moler and Van Loan (1978, 2003) provide a comprehensive discussion of computation of the exponential of a matrix. Datta (2004) discuss computation of the exponential of a matrix and its relation to control theory.

References

Karl J. Åström. *Introduction to Stochastic Control Theory.* Academic Press, 1970.

Biswa Nath Datta. *Numerical Methods for Linear Control Systems. Design and Analysis.* Elsevier, Amsterdam, 2004.

Karl Henrik Johansson. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on Control Systems Technology*, 8:456–465, 2000.

Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review*, 20:801–836, 1978.

Cleve Moler and Charles Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45:3–49, 2003.