Model Predictive Control

Lecture 04: Modified 4-tank system and disturbance models

John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02619 Model Predictive Control

Mathematical Modeling with Differential Equations

Conservation Principle

Physical models are based on conservation principles.

1. Conservation of mass
2. Conservation of energy
3. Conservation of momentum (force)

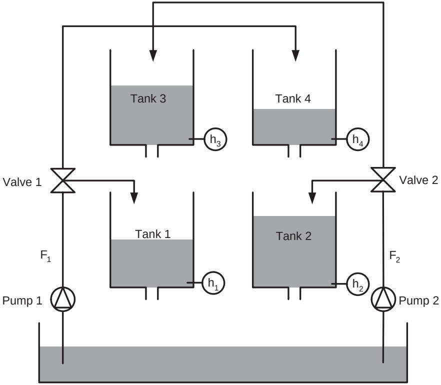
The general derivation of the system equations have the form

$$\text{Accumulated} = \text{Influx} - \text{Outflux} + \overbrace{\text{Produced} - \text{Consumed}}^{=\text{Generated}}$$

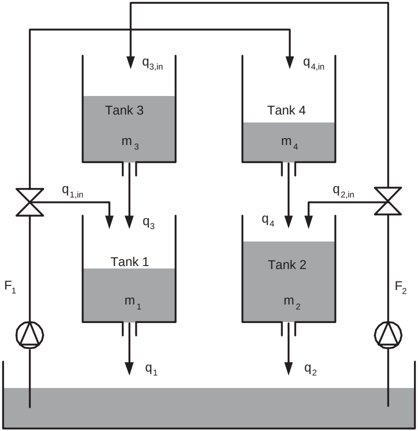
For non-reactive systems the generation term is absent

$$\text{Accumulated} = \text{Influx} - \text{Outflux}$$

4-Tank System - Motivating Example



Example - Tank 1



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

with

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

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

Example - Tank 1

1. Conservation of mass

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

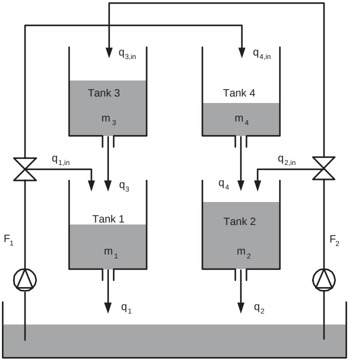
2. Divide by Δt

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

3. Let $\Delta t \to 0$

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

4-Tank System - Model



Mass balances

$$\frac{dm_1}{dt}(t) = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \qquad m_1(t_0) = m_{1,0}$$
$$\frac{dm_2}{dt}(t) = \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) \qquad m_2(t_0) = m_{2,0}$$
$$\frac{dm_3}{dt}(t) = \rho q_{3,in}(t) - \rho q_3(t) \qquad m_3(t_0) = m_{3,0}$$
$$\frac{dm_4}{dt}(t) = \rho q_{4,in}(t) - \rho q_4(t) \qquad m_4(t_0) = m_{4,0}$$

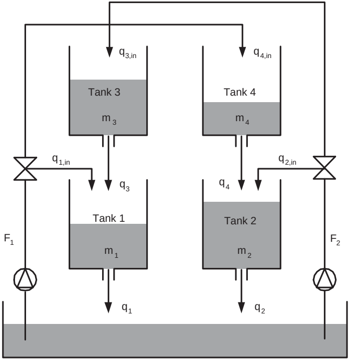
Inflows

$$q_{1,in}(t) = \gamma_1 F_1(t) \qquad q_{2,in}(t) = \gamma_2 F_2(t)$$
$$q_{3,in}(t) = (1-\gamma_2) F_2(t) \quad q_{4,in}(t) = (1-\gamma_1) F_1(t)$$

Outflows

$$q_i(t) = a_i \sqrt{2 g h_i(t)} \quad h_i(t) = \frac{m_i(t)}{\rho A_i} \quad i \in \{1,2,3,4\}$$

4-Tank System - Model



System of ordinary differential equations

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

with the vectors defined as

$$x = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \qquad u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

$$p = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & A_1 & A_2 & A_3 & A_4 & \gamma_1 & \gamma_2 & g & \rho \end{bmatrix}^T$$

This is a non-stiff ODE system
as all processes take place on the same time-scale

Generic Input-Output Model

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

$$y(t) = g(x(t), p) \qquad \text{Sensor function}$$

$$z(t) = h(x(t), p) \qquad \text{Output function}$$

Simulation in Matlab

The model

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

may be implemented in Matlab as

```
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.

...
```

and called using

```
[T,X] = ode45(@ProcessModel,[t0 tf],x0,odeOptions,u,p)
```

Model for the 4-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);       % Inflow from valve 1 to tank 1 [cm3/s]
qin(2,1) = gamma(2)*F(2);       % Inflow from valve 2 to tank 2 [cm3/s]
qin(3,1) = (1-gamma(2))*F(2);   % Inflow from valve 2 to tank 3 [cm3/s]
qin(4,1) = (1-gamma(1))*F(1);   % Inflow from 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
xdot = zeros(4,1);
xdot(1,1) = rho*(qin(1,1)+qout(3,1)-qout(1,1));   % Mass balance Tank 1
xdot(2,1) = rho*(qin(2,1)+qout(4,1)-qout(2,1));   % Mass balance Tank 2
xdot(3,1) = rho*(qin(3,1)-qout(3,1));             % Mass balance Tank 3
xdot(4,1) = rho*(qin(4,1)-qout(4,1));             % Mass balance Tank 4
```

Sensor function

Sensors measuring the level (height) of all tanks

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} \frac{m_1}{\rho A_1} \\ \frac{m_2}{\rho A_2} \\ \frac{m_3}{\rho A_3} \\ \frac{m_4}{\rho A_4} \end{bmatrix} = g(x,p) \tag{1}$$

Matlab implementation

```
function y = FourTankSystemSensor(x,p)
% FOURTANKSYSTEMSENSOR Level for each tank in the four tank system
%
% Syntax: y = FourTankSystemSensor(x,p)

% Extract states and parameters
m = x;
A = p(5:8,1);
rho = p(12,1);

% Compute level in each tank
rhoA = rho*A;
y = m./rhoA;
```

Output function

In this case the output is the level (height) in tank 1 and tank 2

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} \frac{m_1}{\rho A_1} \\ \frac{m_2}{\rho A_2} \end{bmatrix} = h(x, p) \tag{2}$$

Matlab implementation

```
function z = FourTankSystemOutput(x,p)
% FOURTANKSYSTEMOUTPUT Level for the lower tanks in the four tank system
%
% Syntax: z = FourTankSystemOutput(x,p)

% Extract states and parameters
m = x(1:2,1);
A = p(5:6,1);
rho = p(12,1);

% Compute level in each tank
rhoA = rho*A;
z = m./rhoA;
```

Define Simulation Parameters

```
% ------------------------------------------------------------
% 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];
% ------------------------------------------------------------
```

Simulation Scenario and Simulation

```
% --------------------------------------------------------------
% 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]
% --------------------------------------------------------------
```

Simulate the system

```
% --------------------------------------------------------------
% Compute the solution / Simulate
% --------------------------------------------------------------
% Solve the system of differential equations
[T,X] = ode45(@FourTankSystem,[t0 tf],x0,[],u,p);
```

Computation of additional variables

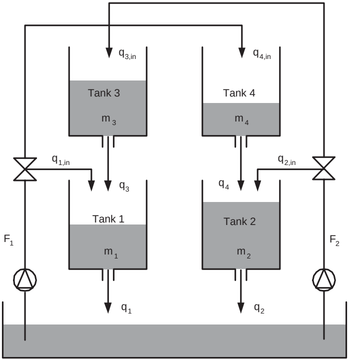
Compute additional variables for plotting

```
% ---------------------------------------------------------------
% 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
% ---------------------------------------------------------------
```

4-Tank System - Model



Mass balances

$$
\begin{aligned}
\frac{dm_1}{dt}(t) &= \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) & m_1(t_0) &= m_{1,0} \\
\frac{dm_2}{dt}(t) &= \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) & m_2(t_0) &= m_{2,0} \\
\frac{dm_3}{dt}(t) &= \rho q_{3,in}(t) - \rho q_3(t) & m_3(t_0) &= m_{3,0} \\
\frac{dm_4}{dt}(t) &= \rho q_{4,in}(t) - \rho q_4(t) & m_4(t_0) &= m_{4,0}
\end{aligned}
$$

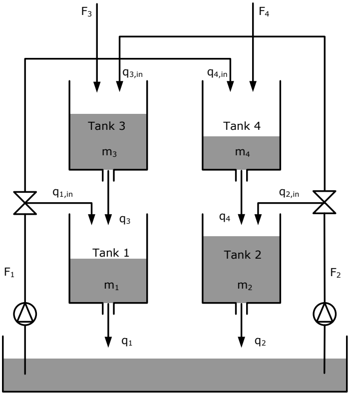
Inflows

$$
\begin{aligned}
q_{1,in}(t) &= \gamma_1 F_1(t) & q_{2,in}(t) &= \gamma_2 F_2(t) \\
q_{3,in}(t) &= (1-\gamma_2) F_2(t) & q_{4,in}(t) &= (1-\gamma_1) F_1(t)
\end{aligned}
$$

Outflows

$$
q_i(t) = a_i \sqrt{2 g h_i(t)} \quad h_i(t) = \frac{m_i(t)}{\rho A_i} \quad i \in \{1,2,3,4\}
$$

Modified 4-Tank System - Model



Mass balances

$$\frac{dm_1}{dt}(t) = \rho q_{1,in}(t) + \rho q_3(t) - \rho q_1(t) \qquad m_1(t_0) = m_{1,0}$$

$$\frac{dm_2}{dt}(t) = \rho q_{2,in}(t) + \rho q_4(t) - \rho q_2(t) \qquad m_2(t_0) = m_{2,0}$$

$$\frac{dm_3}{dt}(t) = \rho q_{3,in}(t) - \rho q_3(t) + \rho F_3(t) \qquad m_3(t_0) = m_{3,0}$$

$$\frac{dm_4}{dt}(t) = \rho q_{4,in}(t) - \rho q_4(t) + \rho F_4(t) \qquad m_4(t_0) = m_{4,0}$$

Inflows

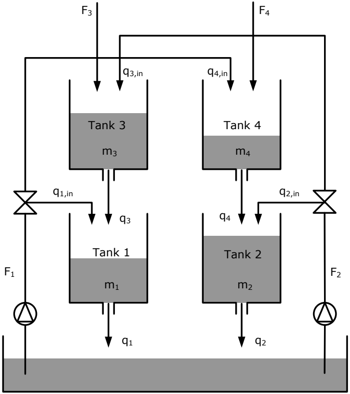
$$q_{1,in}(t) = \gamma_1 F_1(t) \qquad q_{2,in}(t) = \gamma_2 F_2(t)$$

$$q_{3,in}(t) = (1-\gamma_2) F_2(t) \qquad q_{4,in}(t) = (1-\gamma_1) F_1(t)$$

Outflows

$$q_i(t) = a_i \sqrt{2 g h_i(t)} \qquad h_i(t) = \frac{m_i(t)}{\rho A_i} \qquad i \in \{1,2,3,4\}$$

Modified 4-Tank System - Model



System of ordinary differential equations

$$\dot{x}(t) = f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0$$

with the vectors defined as

$$x = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix} \quad u = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \quad d = \begin{bmatrix} F_3 \\ F_4 \end{bmatrix}$$

$$p = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & A_1 & A_2 & A_3 & A_4 & \gamma_1 & \gamma_2 & g & \rho \end{bmatrix}^T$$

This is a non-stiff ODE system
as all processes take place on the same time-scale

Model for the Modified 4-Tank System

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

% Unpack states, MVs, and parameters
m     = x;                       % Mass of liquid in each tank [g]
F     = [u; d];                  % Flow rates [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);        % Inflow from valve 1 to tank 1 [cm3/s]
qin(2,1) = gamma(2)*F(2);        % Inflow from valve 2 to tank 2 [cm3/s]
qin(3,1) = (1-gamma(2))*F(2);    % Inflow from valve 2 to tank 3 [cm3/s]
qin(4,1) = (1-gamma(1))*F(1);    % Inflow from 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
xdot = zeros(4,1);
xdot(1,1) = rho*(qin(1,1)+qout(3,1)-qout(1,1));   % Mass balance Tank 1
xdot(2,1) = rho*(qin(2,1)+qout(4,1)-qout(2,1));   % Mass balance Tank 2
xdot(3,1) = rho*(qin(3,1)-qout(3,1)+F(3));        % Mass balance Tank 3
xdot(4,1) = rho*(qin(4,1)-qout(4,1)+F(4));        % Mass balance Tank 4
```

Linearization and Taylor Expansion

Generic Input-Output Model and Linearization

- Model

$$\frac{dx(t)}{dt} = f(x(t), u(t), p) \qquad x(t_0) = x_0 \qquad \text{Process model}$$
$$y(t) = g(x(t), p) \qquad \text{Sensor function}$$
$$z(t) = h(x(t), p) \qquad \text{Output function}$$

- Taylor expansion around $(\bar{x}, \bar{u}, \bar{y}, \bar{z})$ and p

$$\frac{d}{dt}\delta x(t) = A\delta x(t) + B\delta u(t) + \bar{f} \quad A = \frac{\partial f}{\partial x}(\bar{x}, \bar{u}, p) \quad B = \frac{\partial f}{\partial u}(\bar{x}, \bar{u}, p) \quad \bar{f} = f(\bar{x}, \bar{u}, p)$$
$$\delta y(t) = C\delta x(t) + \bar{g} - \bar{y} \qquad C = \frac{\partial g}{\partial x}(\bar{x}, p) \quad \bar{g} = g(\bar{x}, p)$$
$$\delta z(t) = C_z\delta x(t) + \bar{h} - \bar{z} \qquad C_z = \frac{\partial h}{\partial x}(\bar{x}, p) \quad \bar{h} = h(\bar{x}, p)$$

Deviation variables: $\delta x = x - \bar{x}$, $\delta u = u - \bar{u}$, $\delta y = y - \bar{y}$, $\delta z = z - \bar{z}$,

Generic Input-Output Model and Steady State (SS)

- Model

$$\begin{aligned} \frac{dx(t)}{dt} &= f(x(t), u(t), p) \qquad x(t_0) = x_0 && \text{Process model} \\ y(t) &= g(x(t), p) && \text{Sensor function} \\ z(t) &= h(x(t), p) && \text{Output function} \end{aligned}$$

- Steady state (x_s, u_s, y_s, z_s) with p:

$$\begin{aligned} 0 &= f(x_s, u_s, p) \\ y_s &= g(x_s, p) \\ z_s &= h(x_s, p) \end{aligned}$$

Generic Input-Output Model and Linearization at SS

- Model

$$
\begin{aligned}
\frac{dx(t)}{dt} &= f(x(t), u(t), p) \qquad x(t_0) = x_0 && \text{Process model} \\
y(t) &= g(x(t), p) && \text{Sensor function} \\
z(t) &= h(x(t), p) && \text{Output function}
\end{aligned}
$$

- Taylor expansion around steady state (x_s, u_s, y_s, z_s) and p

$$
\begin{aligned}
\frac{d}{dt}\delta x(t) &= A\delta x(t) + B\delta u(t) && A = \frac{\partial f}{\partial x}(x_s, u_s, p) \quad B = \frac{\partial f}{\partial u}(x_s, u_s, p) \\
\delta y(t) &= C\delta x(t) && C = \frac{\partial g}{\partial x}(x_s, p) \\
\delta z(t) &= C_z\delta x(t) && C_z = \frac{\partial h}{\partial x}(x_s, p)
\end{aligned}
$$

Deviation variables: $\delta x = x - x_s$, $\delta u = u - u_s$, $\delta y = y - y_s$, $\delta z = z - z_s$,

Exact Discretization of Linear Systems with ZOH inputs

- ZOH input

$$\delta u(t) = \delta u_k \qquad t_k \le t < t_{k+1} = t_k + T_s$$

- Continuous-time system

$$\begin{aligned}
\frac{d}{dt}\delta x(t) &= A_c \delta x(t) + B_c \delta u(t) \\
\delta y(t) &= C \delta x(t) \\
\delta z(t) &= C_z \delta x(t)
\end{aligned}$$

- Discrete-time system

$$\begin{aligned}
\delta x(t_{k+1}) &= A \delta x(t_k) + B \delta u(t_k) \\
\delta y(t_k) &= C \delta x(t_k) \\
\delta z(t_k) &= C_z \delta x(t_k)
\end{aligned}$$

- Matrices for the discrete-time dynamics (difference equation)

$$A = \exp(A_c T_s) \qquad B = \int_0^{T_s} \exp(A_c t) B_c dt$$

- Matrix exponential function

$$\begin{bmatrix} A & B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} T_s \right)$$

Generic Input-Output Model and Linearization

- Model

$$\frac{dx(t)}{dt} = f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0 \qquad \text{Process model}$$
$$y(t) = g(x(t), p) \qquad \text{Sensor function}$$
$$z(t) = h(x(t), p) \qquad \text{Output function}$$

- Taylor expansion around $(\bar{x}, \bar{u}, \bar{d}, \bar{y}, \bar{z})$ and p

$$\frac{d}{dt}\delta x(t) = A\delta x(t) + B\delta u(t) + E\delta d(t) + \bar{f}$$
$$\delta y(t) = C\delta x(t) + \bar{g}$$
$$\delta z(t) = C_z\delta x(t) + \bar{h}$$

Matrices:

$$A = \frac{\partial f}{\partial x}(\bar{x}, \bar{u}, \bar{d}, p) \quad B = \frac{\partial f}{\partial u}(\bar{x}, \bar{u}, \bar{d}, p) \quad E = \frac{\partial f}{\partial d}(\bar{x}, \bar{u}, \bar{d}, p) \quad \bar{f} = f(\bar{x}, \bar{u}, \bar{d}, p)$$
$$C = \frac{\partial g}{\partial x}(\bar{x}, p) \quad \bar{g} = g(\bar{x}, p)$$
$$C_z = \frac{\partial h}{\partial x}(\bar{x}, p) \quad \bar{h} = h(\bar{x}, p)$$

Deviation variables: $\delta x = x - \bar{x}$, $\delta u = u - \bar{u}$, $\delta d = d - \bar{d}$, $\delta y = y - \bar{y}$, $\delta z = z - \bar{z}$,

Generic Input-Output Model and Steady State (SS)

- Model

$$
\begin{aligned}
\frac{dx(t)}{dt} &= f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0 && \text{Process model} \\
y(t) &= g(x(t), p) && \text{Sensor function} \\
z(t) &= h(x(t), p) && \text{Output function}
\end{aligned}
$$

- Steady state $(x_s, u_s, d_s, y_s, z_s)$ with p:

$$
\begin{aligned}
0 &= f(x_s, u_s, d_s, p) \\
y_s &= g(x_s, p) \\
z_s &= h(x_s, p)
\end{aligned}
$$

Generic Input-Output Model and Linearization at SS

- Model

$$\begin{aligned}
\frac{dx(t)}{dt} &= f(x(t), u(t), d(t), p) \qquad x(t_0) = x_0 && \text{Process model} \\
y(t) &= g(x(t), p) && \text{Sensor function} \\
z(t) &= h(x(t), p) && \text{Output function}
\end{aligned}$$

- Taylor expansion around steady state $(x_s, u_s, d_s, y_s, z_s)$ and p

$$\begin{aligned}
\frac{d}{dt}\delta x(t) &= A\delta x(t) + B\delta u(t) + E\delta d(t) \\
\delta y(t) &= C\delta x(t) \\
\delta z(t) &= C_z\delta x(t)
\end{aligned}$$

Matrices:

$$\begin{aligned}
A &= \frac{\partial f}{\partial x}(x_s, u_s, d_s, p) \quad B = \frac{\partial f}{\partial u}(x_s, u_s, d_s, p) \quad E = \frac{\partial f}{\partial d}(x_s, u_s, d_s, p) \\
C &= \frac{\partial g}{\partial x}(x_s, p) \\
C_z &= \frac{\partial h}{\partial x}(x_s, p)
\end{aligned}$$

Dev. var: $\delta x = x - x_s$, $\delta u = u - u_s$, $\delta d = d - d_s$, $\delta y = y - y_s$, $\delta z = z - z_s$

Exact Discretization of Linear Systems with ZOH inputs

- ZOH input

$$\delta u(t) = \delta u_k \quad \delta d(t) = \delta d_k \qquad t_k \leq t < t_{k+1} = t_k + T_s$$

- Continuous-time system

$$\begin{aligned} \frac{d}{dt}\delta x(t) &= A_c \delta x(t) + B_c \delta u(t) + E_c \delta d(t) \\ \delta y(t) &= C \delta x(t) \\ \delta z(t) &= C_z \delta x(t) \end{aligned}$$

- Discrete-time system

$$\begin{aligned} \delta x(t_{k+1}) &= A \delta x(t_k) + B \delta u(t_k) + E \delta d(t_k) \\ \delta y(t_k) &= C \delta x(t_k) \\ \delta z(t_k) &= C_z \delta x(t_k) \end{aligned}$$

- Matrices for the discrete-time dynamics (difference equation)

$$A = \exp(A_c T_s) \qquad B = \int_0^{T_s} \exp\left(A_c t\right) B_c dt \qquad E = \int_0^{T_s} \exp\left(A_c t\right) E_c dt$$

- Matrix exponential function

$$\begin{bmatrix} A & B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} T_s \right) \qquad \begin{bmatrix} A & E \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & E_c \\ 0 & 0 \end{bmatrix} T_s \right)$$

Exact Discretization of Linear Systems with ZOH inputs

- Matrices for the discrete-time dynamics (difference equation)

$$A = \exp(A_c T_s) \qquad B = \int_0^{T_s} \exp\left(A_c t\right) B_c dt \qquad E = \int_0^{T_s} \exp\left(A_c t\right) E_c dt$$

- Matrix exponential function - computational procedure I

$$\begin{bmatrix} A & B \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c \\ 0 & 0 \end{bmatrix} T_s\right) \qquad \begin{bmatrix} A & E \\ 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & E_c \\ 0 & 0 \end{bmatrix} T_s\right)$$

- Matrix exponential function - computational procedure II

$$\begin{bmatrix} A & B & E \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} = \exp\left(\begin{bmatrix} A_c & B_c & E_c \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} T_s\right)$$

Disturbance Modeling

Stochastic Processes

- Wiener Process = Brownian motion (integrated white noise)

$$d\boldsymbol{x}(t) = \sigma d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- State dependent diffusion (Poisson)

$$d\boldsymbol{x}(t) = \sigma\sqrt{\boldsymbol{x}(t)}d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- State dependent diffusion

$$d\boldsymbol{x}(t) = \sigma\boldsymbol{x}(t)d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- Langevin model (Ornstein-Uhlenbeck process)

$$d\boldsymbol{x}(t) = -a\boldsymbol{x}(t)dt + \sigma d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- Geometric Brownian motion (exponential Brownian motion)

$$d\boldsymbol{x}(t) = -a\boldsymbol{x}(t)dt + \sigma\boldsymbol{x}(t)d\boldsymbol{\omega}(t) \qquad d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)$$

- Vasicek model

$$d\boldsymbol{x}(t) = a(\bar{x} - \boldsymbol{x}(t))dt + \sigma d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- Cox-Ingersoll-Ross (CIR) model

$$d\boldsymbol{x}(t) = a(\bar{x} - \boldsymbol{x}(t))dt + \sigma\sqrt{\boldsymbol{x}(t)}d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- Return to mean, state dependent diffusion

$$d\boldsymbol{x}(t) = a(\bar{x} - \boldsymbol{x}(t))dt + \sigma\boldsymbol{x}(t)d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

Stochastic Processes

- Time invariant diffusion term

$$
\begin{aligned}
d\boldsymbol{x}(t) &= \sigma d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a dt + \sigma d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a \boldsymbol{x}(t) dt + \sigma d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x} - \boldsymbol{x}(t)) dt + \sigma d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x}(t) - \boldsymbol{x}(t)) dt + \sigma d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)
\end{aligned}
$$

- Square-root diffusion term (Poisson process)

$$
\begin{aligned}
d\boldsymbol{x}(t) &= \sigma \sqrt{\boldsymbol{x}(t)}\, d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a dt + \sigma \sqrt{\boldsymbol{x}(t)}\, d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a \boldsymbol{x}(t) dt + \sigma \sqrt{\boldsymbol{x}(t)}\, d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x} - \boldsymbol{x}(t)) dt + \sigma \sqrt{\boldsymbol{x}(t)}\, d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x}(t) - \boldsymbol{x}(t)) dt + \sigma \sqrt{\boldsymbol{x}(t)}\, d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)
\end{aligned}
$$

- State-dependent diffusion term

$$
\begin{aligned}
d\boldsymbol{x}(t) &= \sigma \boldsymbol{x}(t) d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a dt + \sigma \boldsymbol{x}(t) d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a \boldsymbol{x}(t) dt + \sigma \boldsymbol{x}(t) d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x} - \boldsymbol{x}(t)) dt + \sigma \boldsymbol{x}(t) d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\
d\boldsymbol{x}(t) &= a(\bar{x}(t) - \boldsymbol{x}(t)) dt + \sigma \boldsymbol{x}(t) d\boldsymbol{\omega}(t) && d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)
\end{aligned}
$$

Stochastic Processes used in Finance and their Names

- Merton (1973)
$$dx(t) = a\,dt + \sigma d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Vasicek (1977)
$$dx(t) = (a + bx(t))\,dt + \sigma d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Cox-Ingersoll-Ross (1985) SR
$$dx(t) = (a + bx(t))\,dt + \sigma\sqrt{x(t)}\,d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Dothan (1978)
$$dx(t) = \sigma x(t) d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Geometric Brownian Motion (GBM) - Black and Scholes (1973)
$$dx(t) = bx(t)dt + \sigma x(t) d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Brennan-Schwartz (1977)
$$dx(t) = (a + bx(t))\,dt + \sigma x(t) d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Cox-Ingersoll-Ross (1980) VR
$$dx(t) = \sigma x(t)^{3/2} d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$
- Cox (1975) & Cox and Ross (1976)
$$dx(t) = bx(t)dt + \sigma\,[x(t)]^{\gamma}\,d\omega(t) \qquad d\omega(t) \sim N_{iid}(0, dt)$$

Simulation of Stochastic Processes

- Stochastic Differential Equation (SDE)

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{w}(t) \qquad d\boldsymbol{w}(t) \sim N_{iid}(0, dt)$$

- Euler-Maryuama solution

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + f(\boldsymbol{x}_k)\Delta t + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k \qquad \Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t)$$

Realization of Linear Stochastic Processes

$$Y(s) = H(s)dW(s) \qquad H(s) = \frac{\beta_1 s^3 + \beta_2 s^2 + \beta_3 s + \beta_4}{\alpha_0 s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4} \quad \alpha_0 \neq 0$$

1. Determine state dimension: $n = 4$
2. Convert to standard transfer function $H(s) = \frac{b_1 s^3 + b_2 s^2 + b_3 s + b_4}{s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4}$

$$a_i = \frac{\alpha_i}{\alpha_0} \quad i = 1, 2, 3, 4 \qquad b_i = \frac{\beta_i}{\alpha_0} \quad i = 1, 2, 3, 4$$

3. Continuous-time realization

$$\begin{aligned} d\boldsymbol{x}(t) &= A_c \boldsymbol{x}(t)dt + B_c d\boldsymbol{\omega}(t) \qquad d\boldsymbol{\omega}(t) \sim N_{iid}(0, Idt) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) \end{aligned}$$

with

$$A_c = \left[\begin{array}{c|ccc} -a_1 & 1 & 0 & 0 \\ -a_2 & 0 & 1 & 0 \\ -a_3 & 0 & 0 & 1 \\ \hline -a_4 & 0 & 0 & 0 \end{array}\right] \qquad B_c = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

Linear Stochastic Processes

- Continuous time transfer function

$$Y(s) = H(s)dW(s) \quad H(s) = \frac{\beta_1 s^{n-1} + \ldots + \beta_{n-1}s + \beta_n}{\alpha_0 s^n + \alpha_1 s^{n-1} + \ldots + \alpha_{n-1}s + \alpha_n}$$

- Continuous time state space

$$\begin{aligned} d\boldsymbol{x}(t) &= A_c \boldsymbol{x}(t)dt + B_c d\boldsymbol{\omega}(t) \qquad d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt) \\ \boldsymbol{y}(t) &= C\boldsymbol{x}(t) \end{aligned}$$

- Discrete time state space

$$\begin{aligned} \boldsymbol{x}_{k+1} &= A\boldsymbol{x}_k + \boldsymbol{w}_k \qquad \boldsymbol{w}_k \sim N_{iid}(0, Q) \\ \boldsymbol{y}_k &= C\boldsymbol{x}_k \end{aligned}$$

with

$$A = \exp\left(A_c \Delta t\right) \quad Q = \int_0^{\Delta t} e^{A_c t} B_c B_c' e^{A_c' t} dt$$

that can be computed as

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \exp\left(\begin{bmatrix} -A_c & B_c B_c' \\ 0 & A_c' \end{bmatrix} \Delta t \right) \quad A = \Phi_{22}' \qquad Q = \Phi_{22}' \Phi_{12}$$

Linear Stochastic Processes Examples

- Real poles - combination of first order filters

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

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

$$H(s) = \frac{\sigma}{(\tau_1 s + 1)(\tau_2 s + 1)} \qquad H(s) = \frac{\sigma}{(\tau_1 s + 1)(\tau_2 s + 1)}\frac{1}{s}$$

$$H(s) = \frac{\sigma(\beta s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)} \qquad H(s) = \frac{\sigma(\beta s + 1)}{(\tau_1 s + 1)(\tau_2 s + 1)}\frac{1}{s}$$

- Second order system

$$H(s) = \frac{\sigma}{\tau^2 s^2 + 1} \qquad H(s) = \frac{\sigma}{\tau^2 s^2 + 1}\frac{1}{s}$$

$$H(s) = \frac{\sigma}{\tau^2 s^2 + 2\zeta\tau s + 1} \qquad H(s) = \frac{\sigma}{\tau^2 s^2 + 2\zeta\tau s + 1}\frac{1}{s}$$

$$H(s) = \frac{\sigma\,(\beta s + 1)}{\tau^2 s^2 + 2\zeta\tau s + 1} \qquad H(s) = \frac{\sigma\,(\beta s + 1)}{\tau^2 s^2 + 2\zeta\tau s + 1}\frac{1}{s}$$

- General pole-zero specification

$$H(s) = \frac{\sigma}{s - p_1} \qquad H(s) = \frac{\sigma}{s - p_1}\frac{1}{s}$$

$$H(s) = \frac{\sigma}{(s - p_1)(s - p_2)} \qquad H(s) = \frac{\sigma}{(s - p_1)(s - p_2)}\frac{1}{s}$$

$$H(s) = \frac{\sigma(s - z_1)}{(s - p_1)(s - p_2)} \qquad H(s) = \frac{\sigma(s - z_1)}{(s - p_1)(s - p_2)}\frac{1}{s}$$

Stochastic Processes and Potential Functions

- Potential function: $V(x)$
- Gradient of potential function: $\nabla V(x)$
- Determinstic

$$\dot{x}(t) = -\nabla V(x(t))$$

$$x_{k+1} = x_k - \nabla V(x_k)\Delta t$$

- Stochastic

$$d\boldsymbol{x}(t) = -\nabla V(\boldsymbol{x}(t))dt + g(\boldsymbol{x}(t))d\boldsymbol{\omega}(t) \quad d\boldsymbol{\omega}(t) \sim N_{iid}(0, dt)$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \nabla V(\boldsymbol{x}_k)\Delta t + g(\boldsymbol{x}_k)\Delta \boldsymbol{w}_k \quad \Delta \boldsymbol{w}_k \sim N_{iid}(0, \Delta t)$$

- Example potential functions

$$\begin{aligned}
V(x) &= \frac{1}{2}x^2 & \nabla V(x) &= x & x &\in \mathbb{R} \\
V(x) &= \frac{1}{2}x^2 - \tau \log x & \nabla V(x) &= x - \frac{\tau}{x} & x &> 0 \\
V(x) &= \frac{1}{2}x^2 - \tau \log(x+1)(1-x) & \nabla V(x) &= x - \frac{\tau}{x+1} + \frac{\tau}{1-x} & -1 < x &< 1 \\
V(x) &= \exp\left(\frac{1}{2}x^2\right) & \nabla V(x) &= xV(x) = x\exp\left(\frac{1}{2}x^2\right) & x &\in \mathbb{R} \\
V(x) &= -\exp\left(-\frac{1}{2}x^2\right) & \nabla V(x) &= -xV(x) = x\exp\left(-\frac{1}{2}x^2\right) & x &\in \mathbb{R}
\end{aligned}$$

Stochastic Processes and Potential Functions - $g(x) = \sigma$

- Model

$$d\boldsymbol{x}(t) = -\nabla V(\boldsymbol{x}(t))dt + \sigma d\boldsymbol{\omega}(t) \qquad d\boldsymbol{\omega}(t) \sim N_{iid}(0, Idt)$$

- Examples

$$\begin{aligned}
V(x) &= \frac{1}{2}x^2 & d\boldsymbol{x}(t) &= -\boldsymbol{x}(t)dt + \sigma d\boldsymbol{\omega}(t) \\
V(x) &= \frac{1}{2}x^2 - \tau \log x & d\boldsymbol{x}(t) &= -\left(\boldsymbol{x}(t) - \frac{\tau}{\boldsymbol{x}(t)}\right)dt + \sigma d\boldsymbol{\omega}(t) \\
V(x) &= \frac{1}{2}x^2 - \tau \log(x+1)(1-x) & d\boldsymbol{x}(t) &= -\left(\boldsymbol{x}(t) - \frac{\tau}{\boldsymbol{x}(t)+1} + \frac{\tau}{1-\boldsymbol{x}(t)}\right)dt + \sigma d\boldsymbol{\omega}(t) \\
V(x) &= e^{\frac{1}{2}x^2} & d\boldsymbol{x}(t) &= -\boldsymbol{x}(t)e^{\frac{1}{2}[\boldsymbol{x}(t)]^2}dt + \sigma d\boldsymbol{\omega}(t) \\
V(x) &= -e^{-\frac{1}{2}x^2} & d\boldsymbol{x}(t) &= -\boldsymbol{x}(t)e^{-\frac{1}{2}[\boldsymbol{x}(t)]^2}dt + \sigma d\boldsymbol{\omega}(t)
\end{aligned}$$

System Models and Disturbance Models

- System model

$$\begin{aligned} d\boldsymbol{x}_s(t) &= f_s(\boldsymbol{x}_s(t), u(t), \boldsymbol{d}(t))dt + \sigma(\boldsymbol{x}_s(t), u(t), \boldsymbol{d}(t))d\boldsymbol{\omega}_s(t) & d\boldsymbol{\omega}_s(t) &\sim N_{iid}(0, Idt) \\ \boldsymbol{z}_s(t) &= g_s(\boldsymbol{x}_s(t)) \\ \boldsymbol{y}_s(t_k) &= h_s(\boldsymbol{x}_s(t_k)) + \boldsymbol{v}_s(t_k) & \boldsymbol{v}_s(t_k) &\sim N_{iid}(0, R_{v_s}) \end{aligned}$$

- Disturbance: $\boldsymbol{d}(t) = \boldsymbol{z}_d(t)$
- Disturbance model - nonlinear stochastic process

$$\begin{aligned} d\boldsymbol{x}_d(t) &= f_d(\boldsymbol{x}_d(t))dt + \sigma_d(\boldsymbol{x}_d(t))d\boldsymbol{\omega}_d(t) & d\boldsymbol{\omega}_d(t) &\sim N_{iid}(0, Idt) \\ \boldsymbol{z}_d(t) &= g_d(\boldsymbol{x}_d(t)) \\ \boldsymbol{y}_d(t_k) &= h_d(\boldsymbol{x}_d(t_k)) + \boldsymbol{v}_d(t_k) & \boldsymbol{v}_d(t_k) &\sim N_{iid}(0, R_{v_d}) \end{aligned}$$

- Combined model

$$\begin{aligned} d\boldsymbol{x}(t) &= f(\boldsymbol{x}(t), u(t))dt + \sigma(\boldsymbol{x}(t), u(t))d\boldsymbol{\omega}(t) & d\boldsymbol{\omega}(t) &\sim N_{iid}(0, Idt) \\ \boldsymbol{z}(t) &= g(\boldsymbol{x}(t)) \\ \boldsymbol{y}(t_k) &= h(\boldsymbol{x}(t_k)) + \boldsymbol{v}(t_k) & \boldsymbol{v}(t_k) &\sim N_{iid}(0, R_v) \end{aligned}$$

$$x = \begin{bmatrix} x_s \\ x_d \end{bmatrix} \quad z = \begin{bmatrix} z_s \\ z_d \end{bmatrix} \quad y = \begin{bmatrix} y_s \\ y_d \end{bmatrix} \quad \omega = \begin{bmatrix} \omega_s \\ \omega_d \end{bmatrix} \quad f = \begin{bmatrix} f_s \\ f_d \end{bmatrix} \quad g = \begin{bmatrix} g_s \\ g_d \end{bmatrix} \quad h = \begin{bmatrix} h_s \\ h_d \end{bmatrix}$$

$$\sigma = \begin{bmatrix} \sigma_s & 0 \\ 0 & \sigma_d \end{bmatrix} \quad R_v = \begin{bmatrix} R_{v_s} & 0 \\ 0 & R_{v_d} \end{bmatrix}$$

Example
Modified Four Tank System

Disturbance Model - $F_3(t)$

- Diffusion independent of states

$$
\begin{aligned}
dF_3(t) &= \sigma_{F_3} d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3 - F_3(t))dt + \sigma_{F_3} d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3(t) - F_3(t))dt + \sigma_{F_3} d\omega_{F_3}(t)
\end{aligned}
$$

- Diffusion dependent on square root of states (Poisson)

$$
\begin{aligned}
dF_3(t) &= \sigma_{F_3} \sqrt{F_3(t)} d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3 - F_3(t))dt + \sigma_{F_3} \sqrt{F_3(t)} d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3(t) - F_3(t))dt + \sigma_{F_3} \sqrt{F_3(t)} d\omega_{F_3}(t)
\end{aligned}
$$

- Diffusion dependent on states

$$
\begin{aligned}
dF_3(t) &= \sigma_{F_3} F_3(t) d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3 - F_3(t))dt + \sigma_{F_3} F_3(t) d\omega_{F_3}(t) \\
dF_3(t) &= a_{F_3}(\bar{F}_3(t) - F_3(t))dt + \sigma_{F_3} F_3(t) d\omega_{F_3}(t)
\end{aligned}
$$

- Limits on F_3: $F_{3,\min} \leq F_3(t) \leq F_{3,max}$. Potential function

$$
\begin{aligned}
V(F_3(t)) = \frac{1}{2} \left(\frac{F_3(t) - \bar{F}_3(t)}{F_{3,\max} - F_{3,min}} \right)^2 \\
- \tau \log \left(\frac{F_3 - F_{3,\min}}{F_{3,\max} - F_{3,min}} \right) \log \left(\frac{F_{3,\max} - F_3}{F_{3,\max} - F_{3,min}} \right)
\end{aligned}
$$