Master study
Systems and Control Engineering
Department of Technology
Telemark University College
DDiR, March 23, 2007

Topic: System identification and optimal estimation

Exercise 7, State estimation and extended Kalman filter

Exercise text

We will in this exercise study state estimation and system identification of a chemical reactor. In particular, the following topics are studied:

- Extended Kalman filter (i.e., state estimation in non-linear systems).
- Augmented kalman filter (i.e., combined state and parameter estimation).
- Subspace system identification of process operated in open loop.
- Subspace system identification of process operated in closed loop).

The process and the model we are to work with are earlier used in the Control theory course. The reaction kinetics for the reactor are given by

$$A \overset{k_1}{\to} B \quad \overset{k_2}{\to} C, \tag{1}$$

$$2A \overset{k_3}{\to} D. \tag{2}$$

Note that we have a 2nd order reaction from product A to product D, and that the other reactions are of 1st order. The goal is to study and control the fraction of product B in the reactor. The product C and D are undesired bi products. The control to the reactor is the feed flow rate u i $[\frac{1}{\text{timer}}]$. The fraction of product A in the feed flow rate, u, is θ. The fractions of products A and B in the reactor are x_1 and x_2 respectively. As we see the reaction kinetics for products C and D does not influence upon products A and B. The process model from the input u to the output $y = x_2$ is given by the non-linear model,

$$\begin{aligned} \dot{x_1} &= -k_1 x_1 - k_3 x_1^2 + (\theta - x_1)u, & (3) \\ \dot{x_2} &= k_1 x_1 - k_2 x_2 - x_2 u, & (4) \\ y &= x_2, & (5) \end{aligned}$$

where the reaction coefficients are given by $k_1 = 50$, $k_2 = 100$, $k_3 = 10$. The following steady state values for the states, the control, the disturbance and the parameter are given: $x_1^s = 2.5$, $x_2^s = 1$, $u^s = 25$ and $\theta^s = 10$, respectively.
We will in the first part of the exercise assume that θ is constant and known, i.e., $\theta = \theta^s$. Lather, we will assume that θ is an unknown parameter which we want to estimate.

a) **Non-linear discrete time model**

Make a discrete time state space model by using the Explicit Euler method on the continuous time state space model, i.e., the model should be of the following form

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, \theta_k), && (6) \\ y_k &= g_k(x_k). && (7) \end{aligned}$$

b) **Linearized discrete time model**

With respect to the non-linear discrete time model in step a), find a linearized discrete time model of the form

$$\begin{aligned} \delta x_{k+1} &= A\delta x_k + B\delta u_k + C\delta\theta_k, && (8) \\ \delta y_k &= D\delta x_k. && (9) \end{aligned}$$

An alternative method of obtaining the discrete time linearized model is as follows. First, find a linearized continuous time model of the form

$$\begin{aligned} \delta\dot{x} &= A_c\delta x + B_c\delta u + C_c\delta\theta, && (10) \\ \delta y &= D_c\delta x. && (11) \end{aligned}$$

from the continuous non-linear model. Find a discrete time model of this model by using explicit Euler and compare this model by the discrete time linearized model found above. You should obtain the same lineaized discrete time model!

c) **Extended Kalman filter**

We assume that the real process can be modelled by

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, \theta_k) + v_k, && (12) \\ y_k &= g_k(x_k, u_k) + w_k. && (13) \end{aligned}$$

where v_k is process noise (which e.g. represents unmodelled effects), w_k is measurements noise. The following covariance matrices are assumed

$$V = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad W = 1.0 \cdot 10^{-8} \qquad (14)$$

Noise processes with such covariance can be generated in MATLAB as follows

```
v=0.1*randn(N,2)
w=0.0001*randn(N,1)
```

- Write down an algorithm for an extended Kalman filter. Be care of the order of the equations.
- Find the stationary Kalman filter gain matrix, K. Use the **dlqe.m** function in the MATLAB Control Toolbox.
- Simulate the system with a stationary discrete time extended Kalman filter.
- Simulate the system with a time variant discrete time extended Kalman filter.

d) **Augmented and extended Kalman filter**

We will in this subtask estimate the unknown disturbance, θ, in addition to the system state vector, x_k. We also assume that θ is slowly varying in such a way that it can be modelled by a so called "random walk", i.e.

$$\theta_{k+1} = \theta_k + d\theta_k, \tag{15}$$

where $d\theta_k$ is white noise with zero mean. You are in this subtask to write down an augmented and extended Kalman filter algorithm.

- Write down an augmented model for the discrete time process model and the disturbance (parameter) model.
- Design a kalman filter for the estimation of the state vector, x_k, and the disturbance, θ_k.

e) **System identification**

We want to identify a linearized discrete time state space model by using the **dsr.m** function. The data are collected in open loop and where the input experiment is generated by the pseudo random binary signal

```
U = u^s+prbs1(N,30,150);
```

Simulate the system without noise, i.e., with $w_k = 0$ and $v_k = 0$, and the above input and store the outputs in a vector Y. A state space model can then be identified by

```
[A,B,D,E,C,F,x0]=dsr(Y,U,L,g);
```

It will be natural to chose the parameters $g = 0$ and $L \geq 3$.

- **Alternative 1: using the data directly**
  The system have two states. There also is a mean value in y which is different from zero. This non zero mean give rise to one additional state such that the total model will have $n = 3$ states. Chose $L = 3$ and and identify a 3rd order model.

- **Alternative 2: using centered data**
  use the **dsr.m** function in order to identify a 2nd order model by using the centered data, i.e.,

```
dY = Y-1
dU = U-25
[A,B,D,E,C,F,x0]=dsr(dY,dU,L,g);
```

f) **System identification in closed loop**
Assume now that the system is controlled by a discrete PI controller given by

$$u_k = K_p(r_k - y_k) \tag{16}$$

$$z_{k+1} = z_k + \Delta t \frac{K_p}{T_i}(r_k - y_k) \tag{17}$$

where $K_p = 50$ and $T_i = \frac{1}{75}$. Simulate the controlled system by using an initial value for the controller state as, $z_1 = u^s = 25$ and the following reference signal

```
dR=prbs1(N,30,150);
R=ones(N,1)+0.1*dR;
```

Based on the observed process data Y and U from the closed loop experiment, then identify a state space model by using the **dsr.m** function. You can use the same alternatives as in Step e) above.

Reactor with Kalman filter

```
% File: losn_reak_kf.m
% Solution proposal for the Kalman filter part of Exercise 7
% Topic/Course: System identification and optimal estimation.

clear all

% Process parameters and steady state values.
x1s=2.5; x2s=1; ths=10; us=25;
k1=50; k2=100; k3=10;

% Time horizon, sampling interval etc. used in the simulations
dt=0.001; t0=0; t1=0.2;
t=t0:dt:t1; N=length(t);

ifilt=0;
ifilt=dread('Stationary (0) or time variant Kalman filter (1) ?(0-1)',ifilt);

% Initializing arrays used in the simulations
X=zeros(N,2); Y=zeros(N,1); U=ones(N,1)*us; f=zeros(2,1);
Xh=X;

% Linearized continuous time model
Dc=[0,1];
a11=-k1-2*k3*x1s-us; a12=0;
a21=k1;              a22=-k2-us;
Ac=[a11,a12;a21,a22];
Bc=[ths-x1s;-x2s];
Cc=[us;0];

% Linearised discrete time model
A=eye(2)+dt*Ac;
B=dt*Bc;
C=dt*Cc;
D=Dc;

% Discrete process and measurements noise.
v=0.1*randn(N,2);
w=0.0001*randn(N,1);

% Stationary (steady state) Kalman filter gain matrix
%W=1; V=diag([10,10]);
W=1e-8; V=diag([0.01 0.01]);
[K,Xb,Xh]=dlqe(A,eye(2),D,V,W);

% Simulation
```

```
x=[x1s;x2s];
xb=x+0.1;
for k=1:N
   % get the measurements from the process.
   y=D*x; %+w(k);

   Y(k,:)=y';
   X(k,:)=x';

   if k<20; u=U(k); else; u=25; end   % Make a step in the control at k=21

   % Kalman filter
   yb=D*xb;
   xh=xb+K*(y-yb);          % Aposteriori state estimate at time k.

   if ifilt==1
      K=Xb*D'*inv(D*Xb*D'+W);
   end

   %xb=A*xh+B*u+C*ths;       % Updating the apriori state estimate.
   f(1)=-k1*xh(1)-k3*xh(1)^2+(ths-xh(1))*u;
   f(2)=k1*xh(1)-k2*xh(2)-xh(2)*u;
   xb=xh+dt*f;
   if ifilt==1
      Xh=(eye(2)-K*D)*Xb*(eye(2)-K*D)'+K*W*K';
      Xb=A*Xh*A'+V;
   end

   Xhat(k,:)=xh';
   Xbar(k,:)=xb';


   % Process, updating the process state vector.
   f(1)=-k1*x(1)-k3*x(1)^2+(ths-x(1))*u +v(k,1);
   f(2)=k1*x(1)-k2*x(2)-x(2)*u +v(k,2);
   x=x+dt*f;
end
% Plot some results
plot(t,X(:,2),'r-',t,Xhat(:,2),'b--',t,Xbar(:,2),'k-.');
```

System identification of reactor: open loop experiment

```
% main_reak_ol.m
% Aapen sloeyfe-simulering og systemidentifikasjon av reaktor
% Skisse til loesningsforslag for oeving 7, punkt e).

clear all

% Prosessparametre.
x1s=2.5;x2s=1; ths=10; us=25;
k1=50; k2=100; k3=10;
d=[0,1];


% Tidshorisont og samplingsintervall
dt=0.001;
t=0:dt:1;
N=length(t);

% Generere forsoek i u_k.
dU=prbs1(N,30,150);
U=us+dU;

% Prosess og maalestoey.
w=randn(N,1)*0.0000;
v=randn(N,2)*0.0;

x=[x1s;x2s];
for k=1:N
   % "Henter" maalingen fra prosessen
   y=d*x+w(k);

   % Lagrer/logger data
   X(k,:)=x';
   Y(k,:)=y';

   % Integrerer prosessmodellen
   u=U(k);
   fx1=-k1*x(1)-k3*x(1)^2+(ths-x(1))*u+v(k,1);
   fx2=k1*x(1)-k2*x(2)-x(2)*u +v(k,2);
   x=x+dt*[fx1;fx2];

end

% Systemidentifikasjon, benytter dataene direkte.
```

```
L=3; J=4;
[a,b,d,e,c,f,x0]=dsr(Y,U,L,0,J,1,3);
ym=dlsim(a,b,d,e,U,x0);

% Plotter resultatene
figure(1)
subplot(211), plot(t,U), title('Forsoek i paadraget: u_k'), grid
subplot(212), plot(t,[Y ym]), title('Utgang og estimert utgang'), grid
```

System identification of reactor: closed loop experiment

```
% main_reak_cl.m
% Lukket slyfe simulering av reaktor med PI-regulatror.
% Skisse til loesning av oving 7, punkt f).

clear all

% Prosessparametre.
x1s=2.5;x2s=1; ths=10; us=25;
k1=50; k2=100; k3=10;
d=[0,1];

% Tidshorisont og samplingsintervall
dt=0.001;                % Samplingsintervall
t=0:dt:1; t=t';
N=length(t);

% Prosess og maalestoey.
w=randn(N,1)*0.0001;
v=randn(N,2)*0.1;

% Referansesignal
dR=prbs1(N,30,150);
R=ones(N,1)+0.1*dR;

% regulatorparametre
Kp=50; Ti=1/75;

z=25;        % Initialverdi paa regulatortilstanden
x=[x1s;x2s]; % Initialverdi for tilstandsvektoren

for k=1:N
   % "Henter" maalingen fra prosessen
   y=d*x+w(k);

   % PI-regulator.
   e=R(k)-y;
   u=Kp*e+z;
   z=z+dt*Kp*e/Ti;

   % Lagrer variable
   X(k,:)=x';
   Y(k,:)=y';
   U(k,:)=u';
```

```
   % Integrerer prosessmodellen
   fx1=-k1*x(1)-k3*x(1)^2+(ths-x(1))*u+v(k,1);
   fx2=k1*x(1)-k2*x(2)-x(2)*u +v(k,2);
   x=x+dt*[fx1;fx2];

end

% Identifikasjon vha DSR
L=3; J=4;
[a,b,d,e,c,f,x0]=dsr(Y,U,L,0,J,1,3);
ym=dlsim(a,b,d,e,U,x0);

figure(1)
subplot(411), plot(t,R),
title('Referansesignal: r_k'), grid
subplot(412), plot(t,U)
title('Paadragssignal: u_k'), grid
subplot(413), plot(t,[Y ym])
title('Utgang og estimert utgang'), grid
```