Exercises in System Identification

David Di Ruscio
University of South-Eastern Norway
N-3914 Porsgrunn, Norway
E-mail: David.Di.Ruscio@hit.no

February 21, 2020

Contents

1 Exercise (realization, identification of autonomous systems)

Assume that s series of output data from an autonomous dynamic system is known and given by

$$y_0 = 1, \quad y_1 = 0.9, \quad y_2 = 0.81, \quad y_3 = 0.729, \quad y_4 = 0.6561, \tag{1}$$

An autonomous dynamic system is only driven by initial states, x_0, different from zero. We are assuming that the system can be described by a linearized autonomous state space model of unknown order, i.e.,

$$\begin{aligned} x_{k+1} &= Ax_k, & (2)\\ y_k &= Dx_k, & (3) \end{aligned}$$

where the initial state vector is unknown and different from zero, i.e. $x_0 \neq 0$, the system matrices A and D are also unknown.
We have the following matrix equations

$$\begin{aligned} Y_{k|L} &= O_L X_{k|1}, & (4)\\ Y_{k+1|L} &= O_L A X_{k|1} & (5) \end{aligned}$$

where we for simplicity of notation often write $X_{k|1} = X_k$. The Hankel matrix equations X_k, $Y_{k|L}$ and $Y_{k+1|L}$ are illustrated in the tasks below.
We will in this exercise compute (or estimate) the the system matrices A and D and the initial state vector, x_0, from known output data observations of the system.

a) Use the data in Eq. (1) and the matrix Eqs. (4)- (5) with $k = 0$ and $L = 2$. Show that

$$Y_{0|2} = O_2 X_0, \tag{6}$$

where

$$Y_{0|2} = \begin{bmatrix} y_0 & y_1 & y_2 \\ y_1 & y_2 & y_3 \end{bmatrix}, O_2 = \begin{bmatrix} D \\ DA \end{bmatrix}, X_0 = \begin{bmatrix} x_0 & x_1 & x_2 \end{bmatrix}.$$

b) Find a Singular Value decomposition (SVD) of the data matrix $Y_{0|2}$, i.e, find a decomposition such that

$$Y_{0|2} = USV^T.$$

c) Find an expression for the (extended) observability matrix O_2 and the state matrix X_0 from the results in step b) above.

How can you find the system matrix D and the initial state vector x_0 from the matrices O_2 and X_0?

d) Show that

$$Y_{1|2} = O_2 A X_0, \tag{7}$$

where

$$Y_{1|2} = \begin{bmatrix} y_1 & y_2 & y_3 \\ y_2 & y_3 & y_4 \end{bmatrix}, \tag{8}$$

and O_2 and X_0 is as defined above.

e) Find an expression for the transition matrix A from the relationship in step d) above and the matrices O_2 and X_0 as found in step c).

f) You should now have found a complete model for the system, i.e. the model matrices A, D and the initial state x_0 should at this stage be known. Simulate the model and compare the model outputs with the observed known outputs which was the starting point for the computations.

g) To the end we will show some relations between the data Hankel matrices $Y_{0|2}$ and $Y_{1|2}$.

Take equations (6) and (7) as the starting point and show that

$$Y_{1|2} = \overbrace{O_2 A (O_2^T O_2)^{-1} O_2^T}^{\tilde{A}} Y_{0|2}, \tag{9}$$

and

$$Y_{1|2} = Y_{0|2} X_0^T (X_0 X_0^T)^{-1} A X_0. \tag{10}$$

Tips: a solution proposal is implemented as a MATLAB script, **losn_oppg1.m**.

Solution proposal: Exercise 1

```
% losn_oppg1.m
% Loesningsforslag til oppgave 1

% genererer y0, y1, y2, y3, y4
As=0.9; Ds=1; xs0=1;
[y,x]=dsrsim(As,0,Ds,0,zeros(5,1),xs0);
% elternativ: skriv inn dataene gitt i oppgaveteksten.

%%%%%% Loesning
m=1;                                % antall y variable
disp('Hankel matrisene')
Y02=[y(1) y(2) y(3)                 % Hankelmatrise for beregning av n, O_2 og X0
    y(2) y(3) y(4)]

Y12=[y(2) y(3) y(4)                 % Shiftet Hankelmatrise for beregning av A.
     y(3) y(4) y(5)]

disp('b) Beregner SVD av Y_02')
[U,S,V]=svd(Y02);
sv=diag(S)'                         % Singulaerverdiene
n=1;                                % default value for dread.
n=dread('Tast inn antall sing. verdier forskellig fra null:',n) % Samme som n=rank(Y0)

U1=U(:,1:n); S1=S(1:n,1:n); V1=V(:,1:n);

disp('c) beregner O_2 og X_0')
O2=U1
X0=S1*V1'

disp('e) beregner x0 D og A')
x0=X0(:,1)
D=O2(1:m,:)
A=pinv(O2'*O2)*O2'*Y12*X0'*pinv(X0*X0')

disp('f) simulerer den identifiserte modellen')
yd=dsrsim(A,0,D,0,zeros(5,1),x0)
```

2 Exercise (state matrix equation for SID)

We will in this exercise derive a matrix equation which is of central importance in Subspace system IDentification (SID) methods. Assume that the system is described by a linear discrete time model (the kalman filter on innovation form)

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + K\varepsilon_k, && (11)\\ y_k &= Dx_k + Eu_k + \varepsilon_k. && (12) \end{aligned}$$

Note that this Kalman filter is equivalent to the following alternative formulation

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Ce_k, && (13)\\ y_k &= Dx_k + Eu_k + Fe_k, && (14) \end{aligned}$$

where $K = CF^{-1}$, $\varepsilon_k = Fe_k$, $\mathrm{E}(e_k e_k^T) = I_m$, i.e., such that at $\mathrm{E}(\varepsilon_k \varepsilon_k^T) = FF^T$. We will in this exercise assume that a series of input and output data observations, u_k and y_k $\forall$ $k = 1, 2, \ldots, N$, are known.

a) Show that (13) and (14) (or (11) og (12)) can be written as an ESSM model (se also Equation (1.17) in the Lecture notes),

$$y_{k|L} = O_L x_k + H_L^d u_{k|L} + H_L^s e_{k|L}, \tag{15}$$

where

$$y_{k|L} = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+L-1} \end{bmatrix}, \quad u_{k|L} = \begin{bmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+L-1} \end{bmatrix}, \quad e_{k|L} = \begin{bmatrix} e_k \\ e_{k+1} \\ \vdots \\ e_{k+L-1} \end{bmatrix}. \tag{16}$$

You can with advantage chose a prescribed parameter L, e.g. $L = 3$. Also write down the structure on the matrices O_L, H_L^d and H_L^s,

b) How can you from equation (15) compute the state vector x_k? Assume that y_k, u_k and e_k, $k \geq 0$ as well as the model are known.

c) Show that equation (15) can be written as a data matrix equation

$$Y_{k|L} = O_L X_k + H_L^d U_{k|L} + H_L^s E_{k|L}, \tag{17}$$

where

$$X_k = \begin{bmatrix} x_k & x_{k+1} & \cdots & x_{k+K-1} \end{bmatrix} \in \mathbb{R}^{n \times K}. \tag{18}$$

3 Exercise (realisation, FIR- and state space models)

Given a system described by the following finite number of impulse responses

$$h_1 = 0.5, \quad h_2 = 0.45, \quad h_3 = 0.405, \quad h_4 = 0.3645, \quad h_5 = 0.32805. \tag{19}$$

Note that these impulse responses may be parameters in a so called Finite Impulse response (FIR) model of the form

$$y_t = h_5 u_{t-5} + h_4 u_{t-4} + h_3 u_{t-3} + h_2 u_{t-2} + h_1 u_{t-1} + E u_t. \tag{20}$$

Note also that these impulse responses may be estimated by exciting the system by a unit impulse response at time $t = 0$, i.e. by using the following input experiment on the system

$$U = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{21}$$

The outputs, y_t, will in this case be identical to the impulse response parameters in the system, i.e., we have the following

$$\begin{array}{lllllll} t=0 & \Rightarrow & y_0 & = & Eu_0 & = & E, \\ t=1 & \Rightarrow & y_1 & = & h_1 u_0 & = & h_1, \\ t=2 & \Rightarrow & y_2 & = & h_2 u_0 & = & h_2, \\ t=3 & \Rightarrow & y_3 & = & h_3 u_0 & = & h_3, \\ t=4 & \Rightarrow & y_4 & = & h_4 u_0 & = & h_4, \\ t=5 & \Rightarrow & y_5 & = & h_5 u_0 & = & h_5. \end{array} \tag{22}$$

We also assume that $u_t = 0$ for $t < 0$. Note also that if we are describing the system by a FIR model on incremental form, i.e. by expressing $y_t - y_{t-1}$ and using (20) then we get a so called FIR step response model of the form

$$y_t = y_{t-1} + h_5 \Delta u_{t-5} + h_4 \Delta u_{t-4} + h_3 \Delta u_{t-3} + h_2 \Delta u_{t-2} + h_1 \Delta u_{t-1} + E \Delta u_t. \tag{23}$$

Here $\Delta u_{t-1} = u_{t-1} - u_{t-1}$ and so on. We can in this case simply identify the impulse responses be exiting the system input with a unit step response, i.e. by using the following experiment

$$U = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{bmatrix}^T = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T. \tag{24}$$

We will in this exercise compute a state space model and the matrices (A, B, D) for the system based on the known impulse responses.

a) Write down the Hankel matrix $\mathbf{H}_{1|L}$ by using $L = 2$ and $J = 3$. Tips: se Chapter 2.2.5 in the Lecture notes.

b) Find the system order, n.

c) Find the extended observability matrix, O_2, and the extended controllability matrix, C_3, for the system. Use the SVD of the hankel matrix. Define which realization you are choosing

d) Find the system matrices B and D.

e) Write down the Hankel matrix $\mathbf{H}_{2|L}$ and find the system transition matrix A.

f) Check your results, i.e., check wether the computed model (A, B, D) is reasonable, i.e., check wether $h_1 = \hat{D}\hat{B}$, $h_2 = \hat{D}\hat{A}\hat{B}$, etc, where $\hat{D}$, $\hat{A}$ and $\hat{B}$ are the computed model matrices/parameters.

Solution proposal: Exercise 3

```
% losn_oppg3.m
% LOESNING TIL OPPGAVE 3
% Modifisert 19/2-2000, DDIR

h1=0.5;
h2=0.45;
h3=0.405;
h4=0.3645;
h5= 0.32805;

[m,r]=size(h1);             % Finner antall inganger r og antall utganger m.

disp('a) Hankelmatrisen H12 =')
H12=[h1 h2 h3               % a) Hankelmatrise for beregning av n, O2, C3, D og B.
     h2 h3 h4]

disp('Forslag til systemets orden: n=rank(H12)')
n=rank(H12)                 %  b) Systemets orden dim(x)=n.

disp('Beregner SVD')
[U,S,V]=svd(H12);           %  c) Finn O2 og C3.

disp('Singulaerverdiene til hankelmatrisen H12 er:')
s=diag(S)'

n=dread('b) Spesifiser systemet orden n = ',n);
if (n >= 3) ~= (n <= 1)
   disp(' Spesifiser systemets orden  1 <= n <= 2');
end

U1=U(:,1:n);
S1=S(1:n,1:n);
V1=V(:,1:n);

disp('c) Beregnet O2 og C3 er:')
O2 = U1                      %  Velger internt balansert realisering.
C3 = S1*V1'

disp('d) og e) Beregner A B og D')
D=O2(1:m,1:n)                %  d) Finn systemmatrisene D og B.
B=C3(:,1:r)

H22=[h2 h3 h4;
     h3 h4 h5];              %  e) Finn systemmatrise A
A=pinv(O2'*O2)*O2'*H22*C3'*pinv(C3*C3')
```

```
%%% SJEKKER MODELLEN MOT IMPULSRESPONSENE %%%%%%
disp('Trykk en tast og fortsett ...')
pause

disp('f) Sjekker modellen ved aa regne ut impulsresponser')
h1_e=D*B                  % skal vaere lik oppgitt h1
h2_e=D*A*B                % skal vaere lik oppgitt h2
h3_e=D*A^2*B              % skal vaere lik oppgitt h3
h4_e=D*A^3*B              % skal vaere lik oppgitt h4
```

4 Exercise (ESSM, matrix equation for SID)

In connection with the subspace system identification method DSR a so called Extended State Space Model (ESSM) is used. One advantage with the ESSM is that the unknown state vector x_k is not present in the equation. A part of the identification problem is therefore simpified (the part for computing the B and E matrices). We will in this exercise show how we can find an ESSM model from a state space model and a polynomial model as the starting point.
Consider a system described by a state space model with the following matrices and parameters

$$\begin{array}{ll} A = \begin{bmatrix} 0 & 1 \\ 0 & a_1 \end{bmatrix}, & B = \begin{bmatrix} b_0 \\ b_1 + a_1 b_0 \end{bmatrix}, \\ D = \begin{bmatrix} 1 & 0 \end{bmatrix}, & E = 0, \end{array} \tag{25}$$

where the parameters are $a_1 = 0.8$, $b_0 = 0.4$ and $b_1 = 0.6$.

a) Chose an identification horizon $L = 3$ and show that we can construct the following ESSM model (se Chapter 1 in the Lecture notes)

$$\overbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \end{bmatrix}}^{y_{k+1|3}} = \overbrace{\begin{bmatrix} 0 & 0.6098 & 0.4878 \\ 0 & 0.4878 & 0.3902 \\ 0 & 0.3902 & 0.3122 \end{bmatrix}}^{\tilde{A}_3} \overbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \end{bmatrix}}^{y_{k|3}} + \overbrace{\begin{bmatrix} -0.2927 & -0.1951 & 0 \\ 0.3659 & 0.2439 & 0 \\ 0.2927 & 0.7951 & 0.4 \end{bmatrix}}^{\tilde{B}_3} \overbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \end{bmatrix}}^{u_{k|3}}$$

This exercise can be solved by deriving the expressions for $\tilde{A}_3$, $\tilde{B}_3$ and thereafter put into numerical values.

b) Find the eigenvalues of the system matrix A ?

c) Find the eigenvalues of the system matrix $\tilde{A}_L$ in the extended state space model ?

d) Show that a state space model with the matrices as in (25) can be written as a polynomial model given by

$$y_k = a_1 y_{k-1} + b_0 u_{k-1} + b_1 u_{k-2}. \tag{26}$$

e) Show that (26) can be written as a ESSM given by

$$\overbrace{\begin{bmatrix} y_{k+1} \\ y_{k+2} \\ y_{k+3} \end{bmatrix}}^{y_{k+1|3}} = \overbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & a_1 \end{bmatrix}}^{\tilde{A}_3} \overbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ y_{k+2} \end{bmatrix}}^{y_{k|3}} + \overbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & b_1 & b_0 \end{bmatrix}}^{\tilde{B}_3} \overbrace{\begin{bmatrix} u_k \\ u_{k+1} \\ u_{k+2} \end{bmatrix}}^{u_{k|3}}$$

f) What is the eigenvalues of $\tilde{A}_3$ in exercise e) above?

5 Exercise (impulse responses, FIR and state space)

Given a system described by the linear state space model

$$x_{k+1} = Ax_k + Bu_k, \tag{27}$$
$$y_k = Dx_k + Eu_k. \tag{28}$$

a) Show that we can write

$$y_{k+1} = DAx_k + DBu_k + Eu_{k+1}, \tag{29}$$
$$y_{k+2} = DA^2x_k + DABu_k + DBu_{k+1} + Eu_{k+2}, \tag{30}$$
$$y_{k+3} = DA^3x_k + DA^2Bu_k + DABu_{k+1} + DBu_{k+2} + Eu_{k+3}, \tag{31}$$

b) How can y_{k+M} generally be expressed? Tips: show first that we can write

$$x_{k+M} = A^M x_k + C_M^d u_{k|M}, \tag{32}$$

and thereafter from $y_{k+M} = Dx_{k+M} + Eu_{k+M}$ we have

$$y_{k+M} = DA^M x_k + DC_M^d u_{k|M} + Eu_{k+M}. \tag{33}$$

c) Assume that A is stable and that $A^t \approx 0$ for "large" $t > 0$. Show that we in this case can describe y_t by a Finite Impulse Response (FIR) model

$$y_t = DC_M^d u_{t-M|M} + Eu_t. \tag{34}$$

Remark: there exists system matrices where $A^t = 0$ even for small t, e.g. for $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ we have that $A^2 = 0$. This matrix is called nilpotent and it rises up e.g. when modelling transport delay systems. We will also always have that $\lim_{t\to\infty} A^t = 0$ when A is stable and all eigenvalues are located inside the unit circle in the complex plane.

d) Compare the model with the FIR model in Exercise 3 and find the relationship between the impulse responses and the state space model matrices. Tips: find that $h_1 = DB$, $h_2 = DAB$, and so on.

6 Exercise (Identification of refiner data using, DSR)

We will in this task use observed and recorded refiner process data from the former Union Co. pulp and paper mill in Skien, Norway. We will concentrate in finding the model and relationship between three input (manipulable control variables) and two output variables which are measured and recorded from the Thermo Mechanical Pulping (TMP) refiner. A TMP refiner is pressurized in a casing. The pressure in the casing is approximately 4 [Bar].
The input time series variables are stored on the data file, **utmp.dat**, and the output data time series on the file **ytmp.dat**, i.e.,

Data set 1: utmp.dat and **ytmp.dat**. (35)

The D-SR Toolbox for MATLAB function **DSR** is to be used for dynamic data alalysis and modelling of the refiner. Download the toolbox and remember to define the path to the file location.

a) Plot the variables which are stored in the data files. Mark the figures with the symbols y_1, u_1 and so on. How many samples consists the data files of? Some MATLAB functions which is useful are **subplot**, **size** and **load**, etc.

b) Perform an analysis of the system order by using the **dsr.m** function. How many states will you use ? It may be useful to do the steps c) and d) below in order to find the best choice of system order.

c) Use the 1500 first samples in the data files for constructing the state space model for the refiner. How many states are you using?

d) Validate the model by simulate it over all the N samples in the data files. Compute the prediction error. The prediction error is a measure of the difference between the real outputs y_k and the predicted simulated outputs $\bar{y}_k^s$. You can compute the simulated outputs $\bar{y}_k^s$ by using the **dsrsim.m** function. The predicted simulated error is computed as

$$V_N^s = \frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y}_k^s)^2 \tag{36}$$

e) Validate the model by comparing the optimal prediction with the real outputs. The optimal prediction is obtained by simulating the Kalman filer. The complete Kalman filter is computed by the DSR method. Use the **dsropt.m** function in order to compute the optimal prediction. Plot the optimal predictions, $\bar{y}_k$, in the same figure as the real outputs, y_k.

Note that a solution proposal for this task is partially implemented in the MATLAB script, **tmp_demo1.m**.

7 Exercise (Identification of refiner data using, DSR)

We will in this task use observed and recorded refiner process data from the former Union Co. pulp and paper mill in Skien, Norway. We will concentrate in finding the model and relationship between three input (manipulable control variables) and two output variables which are measured and recorded from the Thermo Mechanical Pulping (TMP) refiner. The input variables is in this task setpoints to local controllers in the process. The output variables are the refiner power and the consistency in the blow line pipe from the refiner casing.
The input and output data time series are stored on the files

Data set 2: utmp2.dat and **ytmp2.dat**. (37)

The subspace system identification algorithm **DSR** is to be used for data analysis and modelling of the refiner data.

a) Plot the variables which are stored in the data files. Mark the figures with the symbols y_1, u_1 and so on. How many samples N consists the data files of? Some MATLAB functions which is useful are **subplot**, **size** and **load**, etc.

b) Those time series have trends, i.e. they vary around some non zero stationary points. It is often common to eliminate those trends by centering the time series. Centering means to subtract the mean of the time series from the actual time series. However, for dynamic systems it may often be better to remove trends by subtracting the mean of, say, the first $25 - 30$ first samples of the time series.

   Remove such a trend from the time series and plot the variables.

c) Perform an analysis of the system order by using the **dsr.m** function. How many states will you use ? It may be useful to do the steps c) and d) below in order to find the best choice of system order.

d) Use the 1500 first samples in the data files for constructing the state space model for the refiner. How many states are you using?

e) Validate the model by simulate it over all the N samples in the data files. Compute the prediction error. The prediction error is a measure of the difference between the real outputs y_k and the predicted simulated outputs $\bar{y}_k^s$. You can compute the simulated outputs $\bar{y}_k^s$ by using the **dsrsim.m** function. The predicted simulated error is computed as

$$V_N^s = \frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y}_k^s)^2 \tag{38}$$

f) Validate the model by comparing the optimal prediction with the real outputs. The optimal prediction is obtained by simulating the Kalman filer. The complete Kalman filter is computed by the DSR method. Use the

dsropt.m function in order to compute the optimal prediction. Plot the optimal predictions, $\bar{y}_k$, in the same figure as the real outputs, y_k. The prediction erreor for one output may be computed as follows

$$V_N = \frac{1}{N}\sum_{k=1}^{N}(y_k - \bar{y}_k)^2 \tag{39}$$

where $\bar{y}_k$ is the optimal prediction from the Kalman filter.

g) Try to tune a PI controller. Use the water set point, u_2, as the manipulable control variable and the consistency, y_2, as the output variable to be controlled.

Note that a solution proposal is partially implemented in the MATLAB script, **tmp_demo2.m**.

8 Exercise (regression, PCA and PCR)

a) The Principal Component Analysis (PCA) and Principal Comonent Regression (PCR) methods are simple and robust implemented through a Singular Value decomosition (SVD) of the input, X, data matrix. Write a MATLAB function which implements both PCA and PCR in the same script. The first line in the script file may lock as follows

$$[B, U1, S1, V1, T, P] = \text{mypcr}(Y, X, a)$$

where $1 \leq a \leq r$ is the number of principal components specified by the user. Remember to write comment sentences in the script as a documentation of the script and variables etc.

b) When writing software code for an numerical algorithm it make sense to use a set of test data with known solution in order to test the implementation. use the following test data with solution.

$$X = \begin{bmatrix} 1 & 1.1 & 0 \\ 1 & 1.1 & 0 \\ 2 & 0.8 & 1 \\ 2 & 0.8 & 1 \\ 1 & 0.9 & 0.1 \\ 1 & 0.9 & 0.1 \end{bmatrix}, \quad Y = \begin{bmatrix} 0.86 & 0.81 \\ 0.86 & 0.81 \\ 1.78 & 1.04 \\ 1.78 & 1.04 \\ 0.83 & 0.71 \\ 0.83 & 0.71 \end{bmatrix} \tag{40}$$

The data matrices Y and X are stored on the ascii files **xtest.dat** and **ytest.dat**. Note that we have $r = 3$ variables in the input data matrix, X, and $m = 2$ variables in the output data matrix, Y.

1. With $a = 3$ components (independent X variables) we got the solution

$$B_3 = \begin{bmatrix} 0.2 & 0.04 \\ 0.6 & 0.7 \\ 0.9 & 0.4 \end{bmatrix}. \tag{41}$$

   Note that this solution is identical to the Ordinary Least squares (OLS) solution, because $a = r$.

2. With $a = 2$ components (independent X variables) we got the solution

$$B_2 = \begin{bmatrix} 0.6321 & 0.3633 \\ 0.1970 & 0.3985 \\ 0.3564 & -0.0067 \end{bmatrix}. \tag{42}$$

3. With $a = 1$ components (independent X variables) we got the solution

$$B_1 = \begin{bmatrix} 0.5834 & 0.4063 \\ 0.3622 & 0.2523 \\ 0.1952 & 0.1360 \end{bmatrix}. \tag{43}$$

Check if you got the same solution by running your own PCA and PCR implementation. Note that a solution proposal is implemented in the MATLAB script function **mypcr.m**.

9 Exercise (DSR, PCR and system identification of dynamic systems)

In this exercise input and output time series data are given from a process. Those data are stored on the files **yov5.dat** (output data) and **uov5.dat** (input data).

a)

Use the MATLAB script *mypcr.m* which was written in Exercise 8 in order to identify a steady state model for the process.

b) Use the same data in order to identify a dynamic state space model for the process by using the **DSR** method. Comment your choice of system order. Compute the steady state gain for the process by using the identified state space model.

c) We have in step a) tried to identify the steady state gain in a dynamic system by using a least squares techniqueas PCR. Compare the steady state gain which was found from the identified dynamic model in step b) with the result in step a). Comment the results.

d) Assume that the system outputs are to be controlled by single input and single output (SISO) PID controllers. Use RGA analysis in order to find rules for pairing of input and output variables.

1. based on the steady state gain found from the identified dynamic model in step b).
2. based on the steady state gain identified by using PCR as in step a).

e) Compare the models by locking at the variance matrices of the simulated error. Tips: the variance matrix of the simulated error may be computed as follows

$$V = \frac{1}{N-1}\sum_{k=1}^{N}(y_k - \hat{y}_k^d)(y_k - \hat{y}_k^d)^T = (Y - Y_d)^T(Y - Y_d)/(N-1). \quad (44)$$

A solution proposal for this exercise is presented in the MATLAB script, losn_oppg8.m

Solution proposal Exercise 9

```
% losn_oppg8.m
%
load yov5.dat
load uov5.dat
y=yov5; u=uov5;
[N,m]=size(y);

disp('a) Beregner forsterkning vha PCR')
hd_pcr=mypcr(y,u);
hd_pcr=hd_pcr'

disp('b) Beregn modell vha DSR')
[a,b,d,e,c,f,x0]=dsr(y,u,2);
n=length(a);

disp(' Beregner forsterkning vha for DSR modell')
hd_dsr=d*inv(eye(n)-a)*b+e

disp('d) RGA analyse')
disp('RGA for DSR modell')
rga_dsr=hd_dsr.*(inv(hd_dsr))'

disp('RGA for PCR modell')
rga_pcr=hd_pcr.*(inv(hd_pcr))'

yd_dsr=dsrsim(a,b,d,e,u,x0);
yd_pcr=hd_pcr*u';
yd_pcr=yd_pcr';

disp('e) Variansmatrisen til simulert feil')
pe_dsr=y-yd_dsr;
pe_pcr=y-yd_pcr;
V_dsr=pe_dsr'*pe_dsr/(N-1)
V_pcr=pe_pcr'*pe_pcr/(N-1)
```

10 Exercise (Partial Least Squares (PLS), univariate data)

The PLS algorithm can easily and simply be implemented trough a controllability matrix for the matrix pair (X^TY, X^TX), i.e.,

$$K_a = \begin{bmatrix} X^TY & X^TXX^TY & (X^TX)^2X^TY & \cdots & (X^TX)^{a-1}X^TY \end{bmatrix} \quad (45)$$

The PLS solution for B in a linear model $Y = XB + E$ where Y and X are known data matrices are given by

$$B_{\mathrm{PLS}} = K_a(K_a^TX^TXK_a)^{-1}K_a^TX^TY \quad (46)$$

Write a MATLAB function in order to implement the PLS algorithm. The first line in the script may lock as follows

$$B_{\mathrm{PLS}} = \mathrm{mypls}(Y, X, a)$$

where $1 \leq a \leq r$ is the number of PLS components. Remember to write comments in the file as a documentation of variables etc used in the script.
Test the algorithm and compare the results obtained in Exercise 8 and 9.
Remark: An PLS implementation by the direct use of Equations (45) and (46) is not numerically stable. Se Chapter 17 and pages 298-303 for a numerically stable implementation of the PLS algorithm.

11 Exercise (SID of deterministic systems)

The background for this exercise is as follows. Given a system where an experiment is performed on the system inputs. The known input and output data

$$\left.\begin{matrix} u_k \\ y_k \end{matrix}\right\} \ \forall \quad k = 1, \ldots, N \ \textbf{(Known data vectors)}$$

of the system are stored and saved in the data matrices U and Y, the input experiments in U and the outputs in Y, in the commin way such that $U \in \mathbb{R}^{N\times r}$ and $Y \in \mathbb{R}^{N\times m}$ are known data matrices.
We want to identify a dynamic state space linearized model from the known data matrices Y and U. are known data matrices.
We are assuming that the system can be represented by a linear discrete time state space model of the form

$$x_{k+1} = Ax_k + Bu_k, \tag{47}$$
$$y_k = Dx_k + Eu_k, \tag{48}$$

where k is discrete time.
Two set of input and output data, from two different systems are generated and given as follows.

System 1

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1.0000 \\ -1.5000 \\ -0.9500 \\ 1.5450 \\ -1.0095 \\ 1.4914 \\ 0.9423 \end{bmatrix} \tag{49}$$

System 2

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 1 \\ -1 \\ 1 \\ -1 \\ -1 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1.1000 \\ -3.1000 \\ -0.8000 \\ 3.6500 \\ -0.7850 \\ 3.0875 \\ 0.5607 \end{bmatrix} \tag{50}$$

The above data matrices are stored on the files **u.dat**, **y1.dat** and **y2.dat**.

Exercise questions

The exercise is best answered by both using pen and paper and the MATLAB program on the computer for computations.

a) Write down the expressions for the matrices which is in the following matrix equations

$$Y_{k+1|L} = \tilde{A}_L Y_{k|L} + \tilde{B}_L U_{k|L+g} \tag{51}$$

and

$$Y_{k|L} = O_L X_k + H_L^d U_{k|L+g-1} \tag{52}$$

where we have specified $k = 1$, $L = 2$ and $g = 1$.

In particular, define the matrices $U_{1|3}$, $Y_{1|2}$ and $Y_{2|2}$ which are needed in the computations.

b) What is the meaning of the parameters L and g?

How many columns, K, is it in the data matrices $U_{1|3}$, $Y_{1|2}$ and $Y_{2|2}$ when all the known observations data are used?

c) Write down the expression and do the computations in MATLAB, for a projection matrix $U_{1|3}^{\perp}$ with the following property.

$$U_{1|3} U_{1|3}^{\perp} = 0_{K \times K} \tag{53}$$

Is there a demand for the number of columns, K, in the data matrix $U_{1|3}$ in order for such a projection matrix to exists?

d) Multiply all terms from right in the matrix Equation (51) with the projection matrix defined in Step c). The result will be a matrix equation given by

$$Z_{2|2} = \tilde{A}_L Z_{1|2} \tag{54}$$

$$Z_{1|2} = O_2 \tilde{X}_1 \tag{55}$$

Write down the formulas for the data matrices $Z_{2|2}$, $Z_{1|2}$ and $\tilde{X}_1$. Perform the computations within MATLAB.

e) Perform a Singular Value decomposition (SVD) of the data matrix $Z_{1|2}$. Find the system order, n, and the observability matrix, O_2. Use an output normal realization.

f) Find a formula for the computation of the system transition matrix, A, expressed in terms of $Z_{2|2}$ and the SVD from Step e). Also find the system matrix D.

g) Find a formula for the computation of $\tilde{A}_2$.

h) Find a formula for the computation of the matrix $\tilde{B}_2$.

i) Show how we can find the system matrices B and E.

j) Assume that $g = 0$. This means that $E = 0$ and that the system is proper. Write down the expressions for the matrices in the ESSM model in Step a) also for this case.

Tips: a solution proposal for this exercise is implemented in the MATLAB script file **losn_oppg10.m**.

Solution proposal Exercise 11

```
% losn_oppg10.m
% Loesningsforslag til oppgave 10.

%%%%%%%%%%%%%%%%%%%%% THE ACTUAL SYSTEM MATRICES %%%%%%%%%%%%%%%%%%%%%%%%%%%
isys=1;
isys=dread('Spesifiser system 1 eller 2 !',isys);

if isys == 1                            % 1st order system.
 A=0.9; B=0.5;                          % Model for y1.dat
 D=1;   E=-1;
elseif isys == 2                        % 2nd order system.
 A=[1.5 1;-0.7 0]; B=[2;-1.3];          % Model for y2.dat
 D=[1 0];          E=-1.1;
end

%%%%%%%%%%%%%%%%%%%%% EXPERIMENT DESIGN %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
u=[-1;1;1;-1;1;-1;-1];                  % Experiment design, N=7.
y=dsrsim(A,B,D,E,u);                    % Generate data.
m=1; r=1;                               % SISO system.

% Merk: last inn datamatrisene u.dat, y1.dat og y2.dat i stedet for
% simuleringen over.

%%%%%%%%%%%%%%%%%%%%% USER SPECIFIED PARAMETERS %%%%%%%%%%%%%%%%%%%%%%%%%%%
L=2;                                    % The # of block rows.
g=1;                                    % The model structure parameter.

%%%%%%%%%%%%%%%%%%%%% COMPUTE STATE SPACE MODEL MATRICES %%%%%%%%%%%%%%%%%%%
if g==1                                 % Ordering the input and output data.
 U1 = [u(1) u(2) u(3) u(4) u(5)         % The # of rows is L+g=L+1=3.
       u(2) u(3) u(4) u(5) u(6)
       u(3) u(4) u(5) u(6) u(7)];
elseif g==0
 U1 = [u(1) u(2) u(3) u(4) u(5)         % The # of roes is L+g=L=2.
       u(2) u(3) u(4) u(5) u(6)];
end

Y1  = [y(1) y(2) y(3) y(4) y(5)
       y(2) y(3) y(4) y(5) y(6)];

Y2=   [y(2) y(3) y(4) y(5) y(6)
       y(3) y(4) y(5) y(6) y(7)];

[rowsU1,K]=size(U1);
```

```
disp('a) Hankelmatrisene er:')
U1, Y1, Y2

disp('c) En projeksjonsmatrise slik at U1*Up=0 er')
Up = eye(K) - U1'*inv(U1*U1')*U1       % Define projection matrix
                                       % so that U1*Up = 0.

Z2=Y2*Up;                              % Compute matrices in Z2 = At*Z1.
Z1=Y1*Up;

[U,S,V] = svd(Z1);                     % Find system order etc.
disp('Singulaerverdiene til Z1 er')
sv=diag(S(1:L*m,1:L*m))'

n=0;                                   % Try to find order automatically.
for i=1:L*m; if sv(i)/sv(1) > 1.0e-7; n=n+1; end; end
n=dread('Spesifiser systemets orden ?',n);

disp('e) Observerbarhetsmatrisen er')
O=U(:,1:n), S1=S(1:n,1:n); V1=V(:,1:n); % Find observability matrix etc.

disp('f) Systemmatrisene A og D er')
d=O(1:m,:)                              % Find D and A.
a=O'*Z2*V1*inv(S1)

disp('g) Utvidet systemmatrise')
At=O*a*O'                               % Find Atilde and Btilde
disp('f) Utvidet systemmatrise')
Bt=(Y2-At*Y1)*U1'*pinv(U1*U1')

if g==1                                 % Find E and B.
 col3=Bt(:,3); col2=Bt(:,2); col1=Bt(:,1);

 h2=col3;                               % Note h1=[0;E]
 h1=col2+At*h2;                         %      h2=[E;DB]
 ob=col1+At*h1;                         %      ob=[DB;DAB]

 e=h2(2);
else
 col2=Bt(:,2); col1=Bt(:,1)

 h1=col2;                               %      h2=[E;DB]
 ob=col1+At*h1;                         %      ob=[DB;DAB]

 e=0;
end
```

```
disp('i) E og B matrisen er')
e
b=0'*ob
```

12 Exercise (SID of deterministic systems, the shift-invariance technique)

We will in this exercise use the same numerical values and observed data as in Exercise 11. We will use the so called "shift invariance technique" in order to estimate the system order, n, the extended observability matrix O_{L+1}, and the system matrices A and D. This is an alternative method to that in Exercise 11.

a) Write down expressions for the matrices in the matrix equation

$$Y_{k|L+1} = O_{L+1}X_k + H^d_{L+1}U_{k|L+g} \tag{56}$$

where the parameters $k = 1$, $L = 2$ $g = 1$ are specified.

In particular, define the data Hankel matrices $U_{1|3}$ and $Y_{1|3}$ within MATLAB, which are needed for the computations.

b) Multiply all terms from right in the matrix Equation (51) with the projection matrix $U^{\perp}_{L+1}$. You will then obtain a matrix equation of the form

$$Z_{k|L+1} = O_{L+1}\tilde{X}_k \tag{57}$$

Write down formulas for the data matrices $Z_{k|L+1}$ and $\tilde{X}_k$. Perform the computations within MATLAB.

c) Perform a Singular Value decomposition (SVD) of the data matrix Z_{L+1} and find the system order, n, the extended observability matrix O_{L+1}, and the system matrices A and D. You should obtain the same results as in Exercise 11. The system matrices B and E can be found in the same way as in Exercise 11.

13 Exercise (SID of deterministic and stochastic systems)

This exercise is an extension of Exercise 11. The point with the exercise is to make a MATLAB function which can be used for the identification of state space models from data from combined deterministic and stochastic systems. The MATLAB function which was written in Exercise 11 can in general only be used for deterministic noise free systems. This exercise is an extension in such a way that the resulting method works for systems with both process and measurements noise.
We assume that the system may be described by the following State Space Model (SSM) model

$$x_{k+1} = Ax_k + Bu_k + Ce_k \tag{58}$$
$$y_k = Dx_k + Eu_k + Fe_k \tag{59}$$

where the integer $k \geq 0$ is discrete-time, $x \in \mathbb{R}^n$ is the state vector with initial value x_0, $y \in \mathbb{R}^m$ is the system output, $u \in \mathbb{R}^r$ is the system input, $e \in \mathbb{R}^m$ is an unknown innovations process of white noise, assumed to be covariance stationary, with zero mean and covariance matrix $\mathrm{E}(e_k e_k^T) = I$. The constant matrices in the SSM are of appropriate dimensions. A is the *state transition* matrix, B is the *external input* matrix, D is the *output* matrix and E is the direct *control input to output* (feed-through) matrix. C and F is related to the *Kalman gain* matrix as $K = CF^{-1}$.
We assume the following input and output data to be known

$$\left.\begin{matrix} u_k \\ y_k \end{matrix}\right\} \ \forall \quad k = 0, \ldots, N-1 \ \textbf{(Known data vectors)}$$

Exercise questions

The exercise is best answered by both using pen and paper and the MATLAB program on the computer for computations.

a) Given the extended state space model (ESSM)

$$Y_{J+1|L} = \tilde{A}_L Y_{J|L} + \tilde{B}_L U_{J|L+g} + \tilde{C}_L E_{J|L+1}, \tag{60}$$

and the state matrix equation

$$Y_{J|L} = O_L X_J + H_L^d U_{J|L+g-1} + H_L^s E_{J|L}. \tag{61}$$

The data matrices which are to be used to remove noise from the future data, i.e., to remove the term $\tilde{C}_L E_{J|L+1}$ from Equations (60) and (61), are given by the past data matrices

$$Y_{0|J} \quad \text{and} \quad U_{0|J}.$$

Specify $J = 2$, $L = 2$ and $g = 1$. Write down expressions for the matrices in the above equations. In particular, define the data matrices $Y_{0|J}$, $U_{0|J}$, $U_{J|L+g}$, $Y_{J|L}$ and $Y_{J+1|L}$ in MATLAB. These data matrices are needed in the computations.

b) What are the meaning of the parameters J, L and g?

c) Derive the following equations

$$Z_{J+1|L} = \tilde{A}_L Z_{J|L} \tag{62}$$

$$Z^d_{J+1|L} = \tilde{A}_L Z^d_{J|L} + \tilde{B}_L U_{J|L+g} \tag{63}$$

i.e., find formulas for the projected data matrices $Z_{J+1|L}$, $Z_{J|L}$, $Z^d_{J+1|L}$ and $Z^d_{J|L}$, and do the computations in MATLAB

d) Perform an Singular Value Decomposition (SVD) of the projected data matrix $Z_{J|L}$, and find the system order, n, and the extended observability matrix, O_L. Use the output normal realization choice.

e) Write down a formula for the determination of the system transition matrix A in terms of $Z_{J+1|L}$ and the SVD in Step e) above. Also find the system matrix D.

f) Find a formula for computing the matrix, $\tilde{A}_L$.

g) Find a formula for computing the matrix, $\tilde{B}_L$.

h) Show how to find the system matrices, B and E.

i) Find an expression for the initial state vector, x_0. Tips: when deriving the extended state space model such an expression is used.

j) Use the time series data from Exercise 6, i.e., the data **ytmp.dat** and **utmp.dat** from Union Co., and find a state space model from the data. Do a validation of the model.

Remarks: It is suggested that during the work with this exercise you should have written a MATLAB script function with a similar syntacs as follows:

$$[A, B, D, E, x0] = \text{my\_sub}(Y, U, L, g, J);$$

Remarks: In connection with system identification of combined deterministic and stochastic systems is is of central importance with a correct and numerically stable and robust implementation of the identification algorithm. The implementations, DSR.M and DSR_E.M in the D-SR Toolbox for MATLAB are numerically stable and robust implementations of the DSR method. The methods can be used in order to identify the complete Kalman filter model including initial states for both open and closed loop systems. Those implementations have shown to work superior compared to other implementations and algorithm.
The syntax is:

$$[A, B, D, E, C, F, x0] = \text{dsr}(Y, U, L, g, J); \tag{64}$$

$$[A, B, D, E, K, F, x0] = \text{dsr\_e}(Y, U, L, g, J, n); \tag{65}$$

14 Exercise (Prediction error methods and other SID methods)

The data series on the files **yov8.dat** and **uov8.dat** are to be used in this exercise. The data is from a SISO system influenced by stochastic disturbances. The time series from Exercise **??**, i.e., **uov5.dat** and **yov5.dat** are also to be used.
One point with this exercise is to illustrate the use of the D-SR Toolbox and the System Identification Toolbox (ident) fir MATLAB. We will also compare different model structures and methods for system identification.
The exercise is best solved by both using pen and paper as well as the MATLAB computer software. Write your own MATLAB m-script file for the different subtasks.

Exercise part 1

Use the time series data (**yov8.dat**,**uov8.dat**) and (**uov5.dat**,**yov5.dat**).

a) Use the DSR algorithm in order to decide the dynamic order of the system

b) Use the DSR algorithm, i.e. the D-SR Toolbox function *dsr.m*, in order to identify a dynamic state space model for the process. Compute the steady state gain, zeroes, simulated error criterion and the prediction error criterion for the model.

c) Use the IDENT function *n4sid.m* in order to identify a state space model for the process. Compute the steady state gain, zeroes, simulated error criterion and the prediction error criterion for the model.

Exercise part 2

a) Use PCA and PCR in order to identify a steady state model for the process. Compute the steady state gain and the prediction error criterion for the model.

b) Assume that the system can be described by a pure ARX model. use the ident Toolbox function *arx.m* in order to identify a model for the system. Use the function *th2ss.m* in order to transform the model to a state space model. Compute the steady state gain, zeroes, and the prediction error criterion for the model.

c) Assume that the system can be described by an ARMAX model. Use the ident Toolbox function *armax.m* in order to identify a model for the process. Use the function *th2ss.m* in order to transform the model to a state space model. Compute the steady state gain, zeroes, and the prediction error criterion for the model.

d) Use the ident Toolbox function *pem.m* in order to identify a model for the process. Use the function *th2ss.m* in order to transform the model to a state space model. Compute the steady state gain, zeroes, and the prediction error criterion for the model.

The results from the different system identification methods can with advantage be presented in a table. Answer the following questions. Is the system/process minimum-phase or non-minimum-phase ? Which model is in your eyes the best model? Is there great differences between the different models?
Tips: a solution proposal for this exercise is implemented in the m-file **losn_oppg12.m**.

Solution proposal to exercise 14

```
% losn_oppg12.m
% L O E S N I N G S  F O R S L A G  T I L  OPPGAVE 12 %%%%%%%%%%%%%
%
load uov8.dat
load yov8.dat

u=uov8; y=yov8;
clear uov8 yov8
[N,r]=size(u);

%a)%%%%%%%%%%%%%%%%%%%%%%%% D S R %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for L=1:5
 [a,b,d,e,c,f,x0]=dsr(y,u,L,1,L,1,1);
 a_dsr(L,1)=a; b_dsr(L,1)=d*b; d_dsr(L,1)=1; e_dsr(L,1)=e;
 c_dsr(L,1)=d*c*inv(f); x0_dsr(L,1)=x0;
 hd_dsr(L,1)=ddcgain(a,b,d,e);
end
ym_dsr=dlsim(a_dsr(1),b_dsr(1),d_dsr(1),e_dsr(1),u,x0_dsr(1));

%b)%%%%%%%%%%%%%%%%%%%%%%%% P C R %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hd=y'*u*pinv(u'*u);                                  % Least Squares
ym_pcr = hd*u;

%c)%%%%%%%%%%%%%%%%%%%%%%%% A R X %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
th_arx=arx([y u],[1,2,0]);
[a_arx,b_arx,d_arx,e_arx,c_arx]=th2ss(th_arx);
ym_arx=dlsim(a_arx,b_arx,d_arx,e_arx,u);
hd_arx=ddcgain(a_arx,b_arx,d_arx,e_arx);

%d)%%%%%%%%%%%%%%%%%%%%%%%% A R M A X %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
th_armax=armax([y u],[1,2,1,0]);
[a_armax,b_armax,d_armax,e_armax,c_armax]=th2ss(th_armax);
ym_armax=dlsim(a_armax,b_armax,d_armax,e_armax,u);
hd_armax=ddcgain(a_armax,b_armax,d_armax,e_armax);

%e)%%%%%%%%%%%%%%%%%%%%%%%% P E M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ms=canform(1,1,[1,1,1]);                             % Modellstructur
th0=ms2th(ms);                                       % Transform til theta form.
th_pem=pem([y u],th0);                               % Id. parametre.
[a_pem,b_pem,d_pem,e_pem,c_pem,x0_pem]=th2ss(th_pem);
ym_pem=dlsim(a_pem,b_pem,d_pem,e_pem,u,x0_pem);
hd_pem=ddcgain(a_pem,b_pem,d_pem,e_pem);
% or as follows, for siso systems.
% po=[na nb nc nd nf nk]=[1 2 1 0 0 0]; th=pem([y u],po);
```

```
%%%%%%%%%%%%%%%%%%%%%%%% O E %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
th_oe=oe([y u],[2,1,0]);
[a_oe,b_oe,d_oe,e_oe,c_oe,x0_oe]=th2ss(th_oe);
hd_oe=ddcgain(a_oe,b_oe,d_oe,e_oe);


% KOMMENTARER
% -DSR OG PEM er generelle metoder, dvs. hele Kalman-filteret estimeres.
% -ARMAX identifiserer ogsaa et fullstendig Kalman-filter, men denne
%  virker bare for SISO systemer. For et SISO system kan ARMAX med fordel
%  benyttes i stedet for PEM.
% -ARX og OE er kan ikke estimere hele Kalman-filteret.
% -Parametrene i en ARX modell kan identifiseres direkte vha minste
%  kvadraters metode.
```

15 Exercise (Using PEM, ARMAX, OE, ARX and DSR)

The time series data on the files, **yov9.dat** and **uov9.dat**, are to be used in this exercise. The time series is from a SISO system excited by stochastic disturbances and generated by a state space model of the form

$$x_{k+1} = \begin{bmatrix} 1.5 & 1 \\ -0.7 & 0 \end{bmatrix} x_k + \begin{bmatrix} 2 \\ -1.3 \end{bmatrix} u_k + \begin{bmatrix} 0.5 \\ 0.3 \end{bmatrix} e_k \tag{66}$$

$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} x_k - u_k + e_k \tag{67}$$

where the initial state vector is given by

$$x_{k=1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{68}$$

The input experiment performed on the system is given by a sum of two sinusoid systems, i.e.,

$$u_k = \sin((k-1)/5) + \sin((k-1)/10) \quad \forall \quad k = 1, \cdots, 1000$$

The input time series can be generated in MATLAB as follows

```
>> t=1:1000;
>> t=t';
>> U=sin((t-1)/5)+sin((t-1)/10);
```

The noise process e_k is white with zero mean and generated in MATLAB by the commands given by

```
>> randn('seed',0)
>> e=randn(1000,1);
```

One goal with the exercise is to give experience in using the prediction error methods for system identification, PEM, ARMAX, OE, and ARX, within the IDENT Toolbox. The subspace method DSR is also to be used. The exercise is best solved by both using pen and paper and the MATLAB computer program. The reader is recommended to write their own m-file script for the solutions.

Exercise 15 part 1

We will first analyze the underlying system by first compute some characteristic parameters of the system. Compute the following:

- Compute the eigenvalues of the system given by (66).
- Compute the steady state gain h^d from u_k to y_k.
- Compute the steady state gain h^s from e_k to y_k.
- Simulate the above system by e.g. using DSRSIM.m, DLSIM.M or using a for loop. Compare the outputs with the data on the given files **yov9.dat** and **uov9.dat**.

Exercise 15 part 2

a) Use the DSR method in order to identify a state space dynamic model for the process. Compute the steady state gain h^d from u_k to y_k, steady state gain h^s from e_k to y_k and the prediction error criterion for the model.

b) Use PCA and PCR in order to identify a steady state model for the process. Compute the steady state gain and the prediction error criterion for the model. Comment upon the number of principal components used and the connection with the ordinary least squares (OLS) method.

c) Assume that the system can be described by a pure ARX model. Specify the polynomials $A(q)$ and $B(q)$ the order na and nb of the polynomials. Use the IDENT Toolbox function *arx.m* in order to identify a model for the system. Use the function *th2ss.m* to transform the model to a state space model. Compute the steady state gain and the prediction error criterion for the model.

d) Assume that the system can be described by an ARMAX model. Specify the polynomials $A(q)$, $B(q)$ and $C(q)$, and its orders na, nb and nc. Use the IDENT Toolbox function *armax.m* in order to identify a model for the system. Use the function *th2ss.m* to transform the model to a state space model. Compute the steady state gain and the prediction error criterion for the model.

e) Assume that the system can be described by an Output Error (OE) model. Specify the order and the polynomials which describes the OE model. Use the IDENT Toolbox function *oe.m* in order to identify a model for the system. Use the function *th2ss.m* to transform the model to a state space model. Compute the steady state gain and the prediction error criterion for the model.

f) Use the IDENT Toolbox function *pem.m* in order to estimate a model for the system. Specify the orders of the polynomials which describes the model. Compute the steady state gains h^d and h^s and the prediction error criterion.

Exercise 15 part 3

Compare the different models obtained in part 1 with respect to eigenvalues, steady state gain, prediction error criterion. the results can with advantage be presented in a table.
Which methods is correct to use on the system given by (66) and (67).
A solution proposal to Exercise 15 is given in the MATLAB m-file script, **losn_oppg13.m**.

Solution proposal for Exercise 15

```
% losn_oppg13.m
% Solution to exercise 9, System Identification
% Written by David Di Ruscio, 17/11-97

%%% GIVEN A STATE SPACE MODEL DESCRIBED BY %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  x_k+1 = A x_k + B u_k + C v_k
%  y_k   = D x_k + E u_k + F v_k
% where
a1=-1.5; a0=0.7; b1=2; b0=-1.3;
A=[-a1,1;-a0,0]; B=[b1;b0]; C=[0.5;0.3]; D=[1,0]; E=-1; F=1;

%%%%%%%%%% GENERATE INPUT AND OUTPUT DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N=1000; u=utype(N,6); randn('seed',0); e=randn(N,1);
y=dlsim(A,[B C],D,[E F],[u e]);

% transfer function from u_k to y_k
% num=[E b1+a1*E b0+a0*E]; den=[1,a1,a0];
% th=arx([y u],[2,3,0]);
% [a,b,d,e,c]=th2ss(th);

[N,r]=size(u);

%%%%% IDENTIFY MODELS WITH DSR, PCR/LS, ARX, ARMAX, PEM AND OE %%%%%%%%%%%%%%%%
% Note: only DSR, ARMAX and PEM are general enough to identify the above model

%a)%%%%%%%%%%%%%%%%%%%%%%%% D S R %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
L=3; [a_dsr,b_dsr,d_dsr,e_dsr,c_dsr,f_dsr,x0_dsr]=dsr(y,u,L,1,L,1,2);
hd_dsr=ddcgain(a_dsr,b_dsr,d_dsr,e_dsr);
hs_dsr=ddcgain(a_dsr,c_dsr,d_dsr,f_dsr);
ym_dsr=dlsim(a_dsr,b_dsr,d_dsr,e_dsr,u,x0_dsr);

%b)%%%%%%%%%%%%%%%%%%%%%%%% P C R %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% assume model on the form y_k = Hd*u_k which gives normal eq. y'=Hd*u'
hd_pcr=y'*u*pinv(u'*u);             % Least Squares=PCR with two components
ym_pcr = (hd_pcr*u')';

%c)%%%%%%%%%%%%%%%%%%%%%%%% A R X %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
na=2; nb=3; nk=0;
th_arx=arx([y u],[na,nb,nk]);
[a_arx,b_arx,d_arx,e_arx,c_arx]=th2ss(th_arx);
ym_arx=dlsim(a_arx,b_arx,d_arx,e_arx,u);
hd_arx=ddcgain(a_arx,b_arx,d_arx,e_arx);

%d)%%%%%%%%%%%%%%%%%%%%%%%% A R M A X %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
na=2; nb=3; nc=2; nk=0; th_armax=armax([y u],[na,nb,nc,nk]);
```

```
[a_armax,b_armax,d_armax,e_armax,c_armax]=th2ss(th_armax);
ym_armax=dlsim(a_armax,b_armax,d_armax,e_armax,u);
hd_armax=ddcgain(a_armax,b_armax,d_armax,e_armax);
hs_armax=ddcgain(a_armax,c_armax,d_armax,1);

%e)%%%%%%%%%%%%%%%%%%%%%%% P E M %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
im_pem=2;
if im_pem==1
 n=2; m=1; ms=canform(n,m,[1,1,1]);                % Modellstructur
 th0=ms2th(ms);                                    % Transform til theta form.
 th_pem=pem([y u],th0);                            % Id. parametre.
 [a_pem,b_pem,d_pem,e_pem,c_pem,x0_pem]=th2ss(th_pem);
else
% or as follows, for siso systems.
% po=[na nb nc nd nf nk]=[2 3 2 0 0 0]; th=pem([y u],po);
 po=[2 3 2 0 0 0];
 th_pem=pem([y u],po);
end
[a_pem,b_pem,d_pem,e_pem,c_pem,x0_pem]=th2ss(th_pem);
ym_pem=dlsim(a_pem,b_pem,d_pem,e_pem,u,x0_pem);
hd_pem=ddcgain(a_pem,b_pem,d_pem,e_pem);
hs_pem=ddcgain(a_pem,c_pem,d_pem,1);

%%%%%%%%%%%%%%%%%%%%%%%%%% O E %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nb=3; na=2; nk=0; th_oe=oe([y u],[nb,na,nk]);
[a_oe,b_oe,d_oe,e_oe,c_oe,x0_oe]=th2ss(th_oe);
hd_oe=ddcgain(a_oe,b_oe,d_oe,e_oe);


% KOMMENTARER
% -DSR OG PEM er generelle metoder, dvs. hele Kalman-filteret estimeres.
% -ARMAX identifiserer ogsaa et fullstendig Kalman-filter, men denne
%  virker bare for SISO systemer. For et SISO system kan ARMAX med fordel
%  benyttes i stedet for PEM.
% -ARX og OE er kan ikke estimere hele Kalman-filteret.
% -Parametrene i en ARX modell kan identifiseres direkte vha minste
%  kvadraters metode.
```

16 Exercise (PEM, DSR. MIMO system.)

The time series on the files **yov10.dat** and **uov10.dat** are to be used in this exercise. The data is from a MIMO system described by the state space model

$$\overbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1}}^{x_{k+1}} = \overbrace{\begin{bmatrix} -1.5 & 1 & 0.1 \\ -0.7 & 0 & 0.1 \\ 0 & 0 & 0.85 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k}^{x_k} + \overbrace{\begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}}^{B} \overbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}_k}^{u_k} + \overbrace{\begin{bmatrix} 0 & 0.1 \\ 0.1 & 0 \\ 0 & 0.2 \end{bmatrix}}^{C} \overbrace{\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k}^{v_k} \quad (69)$$

$$\overbrace{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}_k}^{y_k} = \overbrace{\begin{bmatrix} 3 & 0 & -0.6 \\ 0 & 1 & 1 \end{bmatrix}}^{D} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \overbrace{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k}^{w_k} \quad (70)$$

where the initial state is given by

$$x_{k=1} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T. \quad (71)$$

A Pseudo Random Binary Signal (PRBS) experiment is performed in each of the two input variables of the system. The experiment is of length $N = 2000$ discrete time instants (samples). The experiment is generated within MATLAB as follows

$$u1 = \text{idinput}(N, 'prbs', [0, 1/M1]);$$
$$u2 = \text{idinput}(N, 'prbs', [0, 1/M2]);$$

where $M1 = 10$ and $M2 = 20$. The parameter $M1 = 10$ means that the experiment in input channel (variable) one is constant over intervals of length $M1 = 10$ samples. This gives an input data matrix

$$U = \begin{bmatrix} u1 & u2 \end{bmatrix},$$

where $U \in \mathbb{R}^{N \times 2}$.
The process noise v_k and the measurements noise w_k are both withe with zero mean and generated in MATLAB as follows

$$\text{randn}('seed', 0);$$
$$v = \text{randn}(N, 1);$$
$$w = \text{randn}(N, 1);$$

The simulated output y_k^d of the system is defined as the output of the deterministic part of the above state space model, i.e.,

$$x_{k+1}^d = Ax_k^d + Bu_k \quad (72)$$
$$y_k^d = Dx_k^d + Eu_k \quad (73)$$

It is in system identification, model identification and validation normal to investigate the error

$$e_k^d = y_k - y_k^d \ \forall \ k = 1, \ldots, N \quad (74)$$

This is called the *simulated error*.
A good model (A, B, D, E) is a model which results in a small simulated error e_k^d. The size of the error can be measured by some matrix norm of the covariance matrix of the error

$$\Lambda^d = \frac{1}{N-1}\sum_{k=1}^{N} e_k^d (e_k^d)^T = \frac{1}{N-1}(E^d)^T E^d \tag{75}$$

where $E^d = Y - Y^d$ is an $N \times m$ matrix of the error at the N time instants.
Note that the simulated error e_k^d in general is different from the prediction error $e_k = y_k - \bar{y}_k$ where $\bar{y}_k$ is the optimal prediction. The prediction error of the model is computed by first simulating the optimal predictor (the Kalman filter), i.e.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + K\overbrace{(y_k - Dx_k - Eu_k)}^{e_k} & (76)\\ \bar{y}_k &= Dx_k + Eu_k & (77) \end{aligned}$$

which gives the prediction error as

$$e_k = y_k - \bar{y}_k \ \forall \ k = 1, \ldots, N \tag{78}$$

The size of this error is measured as the size of the covariance matrix

$$\Lambda = \frac{1}{N-1}\sum_{k=1}^{N} e_k e_k^T = \frac{1}{N-1}E^T E \tag{79}$$

where $E = Y - \bar{Y}$ is an $N \times m$ matrix of the prediction error at the N discrete time instants.
Note that (79) is a true expected estimate of the exact covariance matrix $\Lambda_0 = \mathrm{E}(e_k e_k^T)$. A common way of measuring the size of the prediction error for systems with multiple outputs is to use the trace operator, i.e.,

$$V_N = \mathrm{trace}(\Lambda). \tag{80}$$

Similarly, the size of the simulated error covariance matrix can be measured as follows

$$V_N^d = \mathrm{trace}(\Lambda^d). \tag{81}$$

One of the goals of this exercise is to give experience in using the IDENT Toolbox function *pem* and the subspace system identification function *dsr* for the identification of MIMO systems.

Exercise 16 part 1

We will first analyze the underlying system by first compute some characteristic parameters of the system.

a) Compute the steady state gain h^d from u_k to y_k and the steady state gain h^s from e_k to y_k. Compute the eigenvalues of the transition matrix A ind investigate if the system is stable.

b) Simulate the given system by using *dsrsim.m* or *dlsim.m*. Compare the data with the data on the files **yov10.dat** and **uov10.dat**.

Exercise 16 part 2

a) Use the DSR algorithm in order to identify a state space model for the process. use the time series on the files **yov10.dat** and **uov10.dat**. Your own simulated data can also be used. Compute steady state gain h^d from u_k to y_k, and the prediction error criterion V_N for the model.

b) Use the IDENT Toolbox function *pem.m* in order to identify a state space model for the system. Specify the parameters in the calling/input arguments to *pem.m*. Compute steady state gain h^d from u_k to y_k, and the prediction error criterion V_N for the model.

Exercise 16 part 3

a) Compare the computation time used by the two methods. See the MATLAB function flops.m. The flops.m function works only for earlier versions of MATLAB. Try the *tic* and *toc* commands within MATLAB instead. Which of the methods, DSR ore PEM, are the fastest.

b) Compare the obtained models found by the different identification methods with respect to eigenvalues, steady state gain and value of the prediction error criterion.

Solution proposal Exercise 16

A solution proposal is given as a MATLAB m-file script. See the file *losn_oppg14.m*, given below.
The input experiment performed on the process is shown in Figure 1.

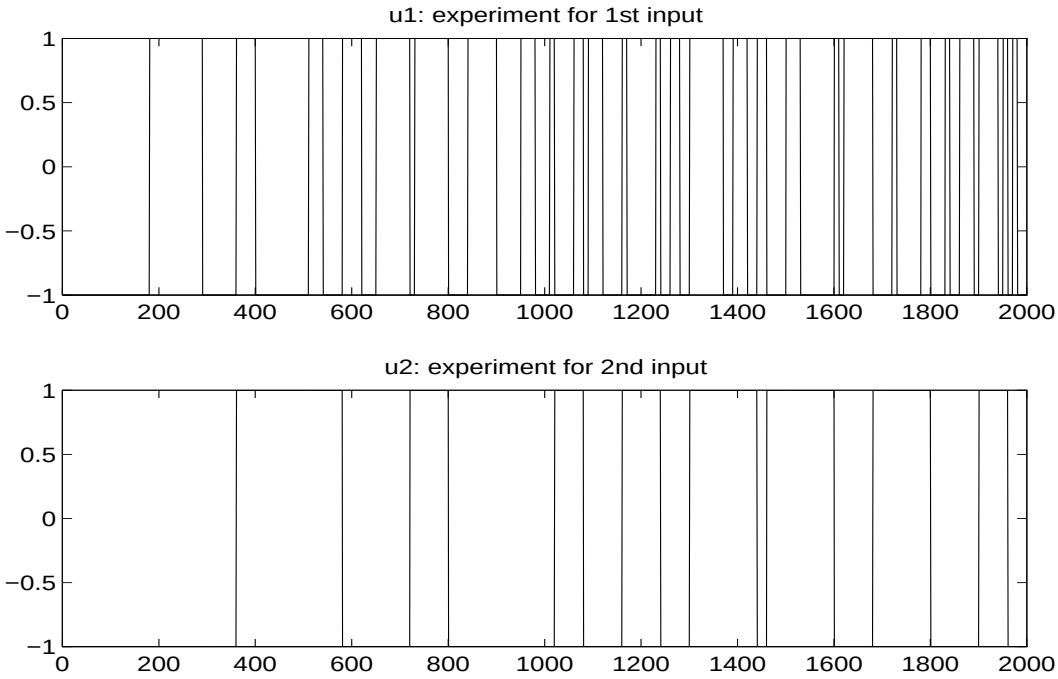

Figure 1: PRBS input to experiment design.

The actual output y_k and the simulated output y_k^d from the identified DSR model are shown in Figure 2.
The actual output y_k and the simulated output y_k^d from the identified PEM model are shown in Figure 3.

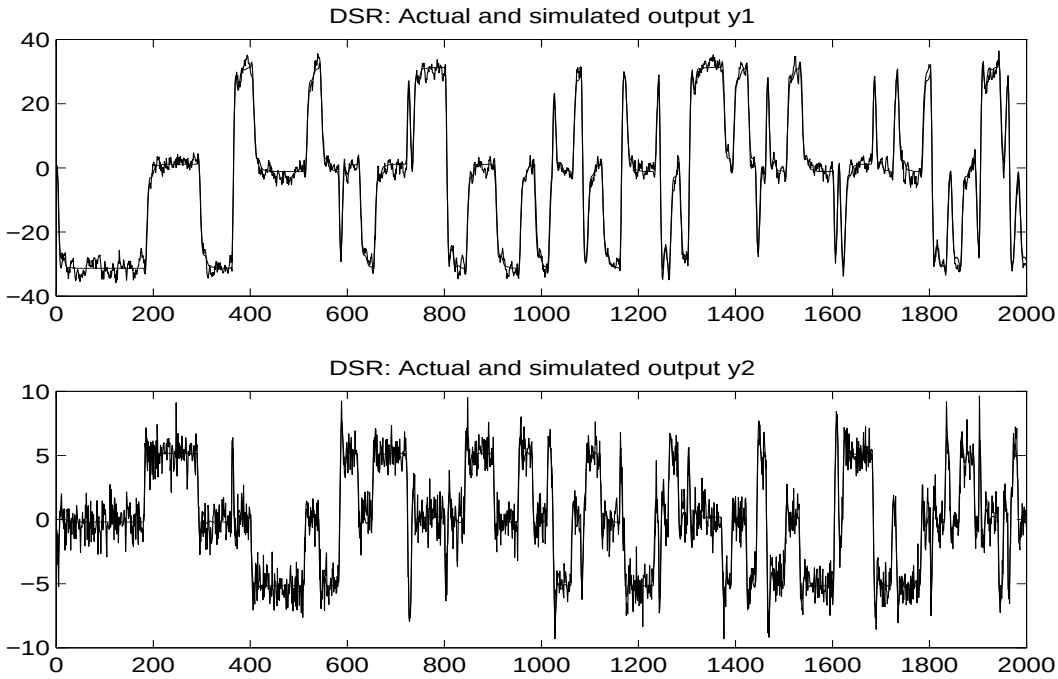

Figure 2: The figure illustrates results from the simulated DSR model as well as the actual output.

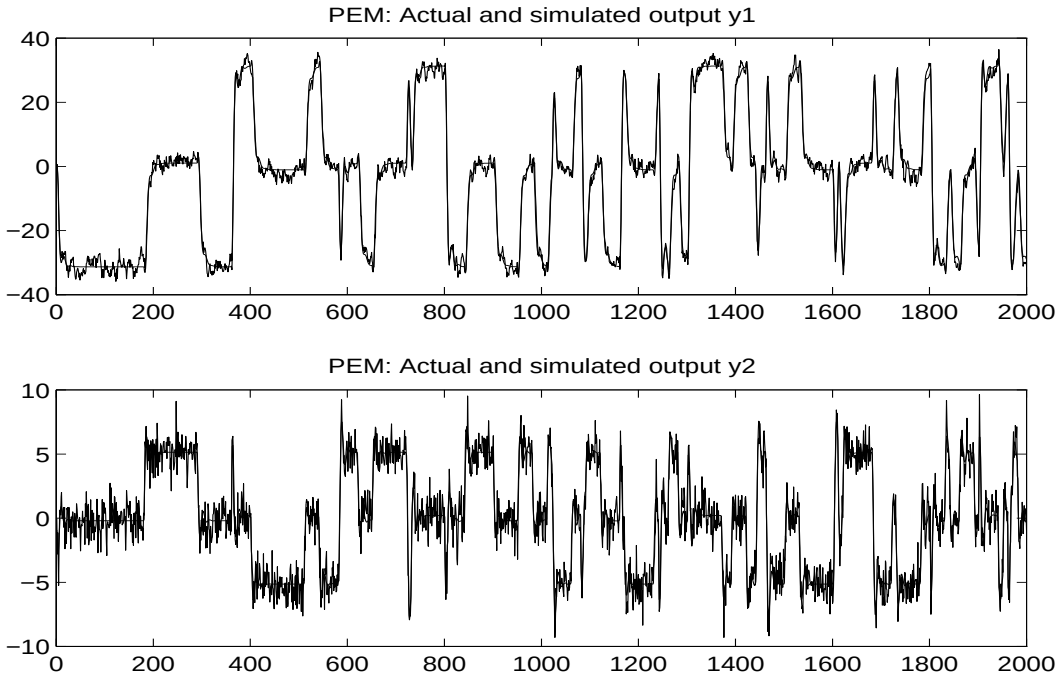

Figure 3: Actual output and results from the simulated PEM model.

Solution proposal, Exercise 16

```
%losn_oppg14.m
%
% Solution to excercise 14, System identification
% Abstarct: This is an excercise in using the DSR and PEM algorithms for
% identifying a MIMO system.

%%%%%%%%%%%%%%%%%%%%%%%% INPUT EXPERIMENT DESIGN %%%%%%%%%%%%%%%%%%%%%%%%%
disp('Make input experiment')
N=2000;
itool = 2;
if itool == 1
% M1=10; u1=idinput(N,'prbs',[0,1/M1]);      % need new IDENT Toolbox
% M2=20; u2=idinput(N,'prbs',[0,1/M2]);
else
 u1=prbs1(N,50,100);
 u2=prbs1(N,25,75);
end
u=[u1 u2];

figure(1)
subplot(211), plot(u1), title('u1: experiment for 1st input')
subplot(212), plot(u2), title('u2: experiment for 2nd input')
% print -deps input_ov10

%%%%%%%%%%%%%%%%%%%%%%%% Genereate data by simulating the model %%%%%%%%%%%%%

randn('seed',0)
[y,u,v,w,A,B,D,E,C,F] = modsim(1,1,u,N);                % yov10=y, uov10=u
yd=dlsim(A,B,D,E,u);                                    % Simulated output.

%%%%%%%%%%%%%%%%%%%%%%%% IDENTIFY MODEL WITH DSR TOOLBOX %%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(2)
flops(0);                                               % reset flop counter
[a_dsr,b_dsr,d_dsr,e_dsr,c_dsr,f_dsr,x0_dsr]=dsr(y,u,2); % DSR Toolbox
k_dsr=c_dsr*inv(f_dsr);                                 % Kalman gain matrix.
ym_dsr=dlsim(a_dsr,b_dsr,d_dsr,e_dsr,u,x0_dsr);         % simulated output
nf_dsr=flops;                                           % # of flops used by DSR
hd_dsr=ddcgain(a_dsr,b_dsr,d_dsr,e_dsr);
e_dsr=y-ym_dsr; L_dsr=e_dsr'*e_dsr; L_dsr=L_dsr/(N-1);   % Covariance matrix of model

figure(3)
subplot(211), plot([y(:,1) ym_dsr(:,1)]), title('DSR: Actual and simulated output y1')
subplot(212), plot([y(:,2) ym_dsr(:,2)]), title('DSR: Actual and simulated output y2')
% print -deps ym_dsr
```

```
%%%%%%%%%%%%%%%%%%%%%%%%% IDENTIFY MODEL WITH PEM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('Identify model with PEM....(wait)')

flops(0);
th0=canstart([y u],[2 1],2,[1,1,1]);                       % estimate initial model
th=pem([y u],th0);                                          % identify model with PEM
[a_pem,b_pem,d_pem,e_pem,k_pem,x0_pem]=th2ss(th);
ym_pem=dlsim(a_pem,b_pem,d_pem,e_pem,u,x0_pem);
nf_pem=flops;                                               % # of flops used by PEM.
hd_pem=ddcgain(a_pem,b_pem,d_pem,e_pem);                    % Deterministic gain
e_pem=y-ym_pem; L_pem=e_pem'*e_pem; L_pem=L_pem/(N-1);   % Covariance matrix of model

figure(4)
subplot(211), plot([y(:,1) ym_pem(:,1)]), title('PEM: Actual and simulated output y1')
subplot(212), plot([y(:,2) ym_pem(:,2)]), title('PEM: Actual and simulated output y2')
% print -deps ym_pem

disp('Number of flops used by DSR')
nf_dsr
disp('Number of flops used by PEM')
nf_pem

disp('Ratio flops_dsr and flops_pem=')
nf_pem/nf_dsr

disp('CONCLUSIONS')
disp('PEM are using approximately 25 times more flops as DSR.')
disp('This is in agrement with the statement that DSR is faster than PEM.')
```

17 Exercises (validation by using functions in the IDENT Tolbox for MATLAB)

a) Identify a model by using the **dsr.m** function. Take one of the given data set as the starting point.

b) Use the **dsr_prd.m** and **dsrpred.m** functions in order to compute an M-step ahead prediction of the output y_t. Chose for example $M = 5$.

c) Take the file **dsrpred.m** as the starting point and modify this file so that the theta format file is an output argument from the m-file. Call the modified file for **dsrpred2.m**. The theta format variable, th, is the model structure format used in the IDENT Toolbox.

d) Compute residuals, confidence intervals by the function **resid.m**.

18 Exercise (higher order ARX model followed by model reduction)

We will in this exercise study a "subspace" alike method for subspace identification which is based on the following. First estimate a higher order ARX model and then follow with a model reduction step. The method can be illustrated in the following steps, which also give guidelines for how to implement such a method for system identification in MATLAB:

1. Identify a higher order ARX model. Use with advantage the **arx.m** function in the IDENT Toolbox for MATLAB. For example as follows:

   ```
   th=arx([Y U],[L L 1]);
   ```

   where L is a "large" positive integer parameter representing the order of the higher order ARX model, e.g. chosen in the interval $L = 5$ to $L = 20$.

2. Transform the higher order ARX model to a higher state space model. Use with advantage the **th2ss.m** function in the IDENT Toolbox for MATLAB, e.g. as follows:

   ```
   [A B D E K]=th2ss(th);
   ```

3. Based on the above higher order state we can compute estimates of the impulse response matrices of the system, i.e.

   $$H_i = DA^{i-1}[B \;\; K] \;\; \forall \;\; i = 1, \ldots, 2L \tag{82}$$

4. A reduced nth order state space model can now be constructed by using Hankel matrix realization theory. Write down or define the Hankel matrices $H_{1|L}$ and $H_{2|L}$ and analyze the order of the system and find a state space model via Singular Value Analysis (SVD) on the following equations.

   $$H_{1|L} = O_L C_L, \tag{83}$$
   $$H_{2|L} = O_L A C_L. \tag{84}$$

   Hence, the system order n can be found by inspection of the number of large singular values of the Hankel matrix $H_{1|L}$. The extended observability matrix O_L and the extended controllability matrix C_L can be found from the SVD of $H_{1|L}$. The D, B and K model matrices are found from O_L and C_L. Finally, the transition matrix A is found from $H_{2|L}$ when O_L and C_L are known.

This algorithm may be useful for system identification of closed loop systems. A matlab m-file script to simulate a 2nd order SISO system controlled by a PI controller is implemented in the file **main_clop_dat.m**. The system is perturbed by a binary signal in the reference. Those data (Y, U) may be used in this exercise.
Compare the results obtained by this higher order ARX model with model reduction and the results obtained by using the **dsr_e.m** function.