Identification of MIMO linear models: introduction to subspace methods

Marco Lovera
Dipartimento di Scienze e Tecnologie Aerospaziali
Politecnico di Milano
marco.lovera@polimi.it

State space identification from impulse response data

Ho-Kalman realisation theory

Consider the finite dimensional,
linear time-invariant (LTI) state space model:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned}$$

Realisation: the problem of computing [A,B,C,D] or an equivalent realisation for the system, from the impulse response (Markov parameters) of the system:

$$\begin{cases} h(0) = D \\ h(t) = CA^{t-1}B, \quad t > 0 \end{cases}$$

Ho-Kalman realisation theory (cont.d)

A few definitions:

- Extended observability matrix:

$$\Gamma_i = \begin{bmatrix} C^T & (CA)^T & (CA^2)^T & \ldots & (CA^{i-1})^T \end{bmatrix}^T$$

- Extended controllability matrix:

$$\Delta_i = \begin{bmatrix} B & AB & A^2B & \ldots & A^{i-1}B \end{bmatrix}$$

Ho-Kalman realisation theory (cont.d)

Hankel matrix

$$\mathbf{u}_-(t) = [\ \mathbf{u}^T(t-1) \quad \mathbf{u}^T(t-2) \quad \dots \quad \mathbf{u}^T(t-j)\ ]^T$$

$$\mathbf{y}_+(t) = [\ \mathbf{y}^T(t) \quad \mathbf{y}^T(t+1) \quad \dots \quad \mathbf{y}^T(t+i-1)\ ]^T$$

$$H_{i,j} = \begin{bmatrix} h(1) & h(2) & h(3) & \dots & h(j) \\ h(2) & h(3) & h(4) & \dots & h(j+1) \\ h(3) & h(4) & h(5) & \dots & h(j+2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h(i) & h(i+1) & h(i+2) & \dots & h(j+i-1) \end{bmatrix}$$

$$y_+(t) = H_{ij} u_-(t)$$

Ho-Kalman realisation theory (cont.d)

Properties of the Hankel matrix:

- $H_{i,j}$, i,j ¸ n, has rank n iff h(t) admits an n_{th} order [A,B,C,D] realisation;

- $H_{i,j}$ can be equivalently written as

$$H_{ij} = \Gamma_i \Delta_j$$

Ho-Kalman realisation theory (cont.d)

The realisation can be constructed as follows:

- Let D=h(0);
- Construct the Hankel matrix $H_{i,j}$ from h(1), h(2), …;
- Factor the Hankel matrix to get Γ_i and Δ_j;
- Let C=first l rows of Γ_i;
- Let B=first m columns of Δ_j;
- Compute A exploiting *shift invariance*, i.e., solving

$$\Gamma_{\uparrow} A = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-2} \end{bmatrix} A = \begin{bmatrix} CA \\ CA^2 \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} = \Gamma_{\downarrow}$$

Kung's algorithm (1978)

What if *noisy* measurements of h(t) are available?

$$\tilde{h}(t) = h(t) + w(t)$$

Idea:

- Construct the noisy Hankel matrix \hat $H_{i,j}$
- Factor the matrix using the SVD:

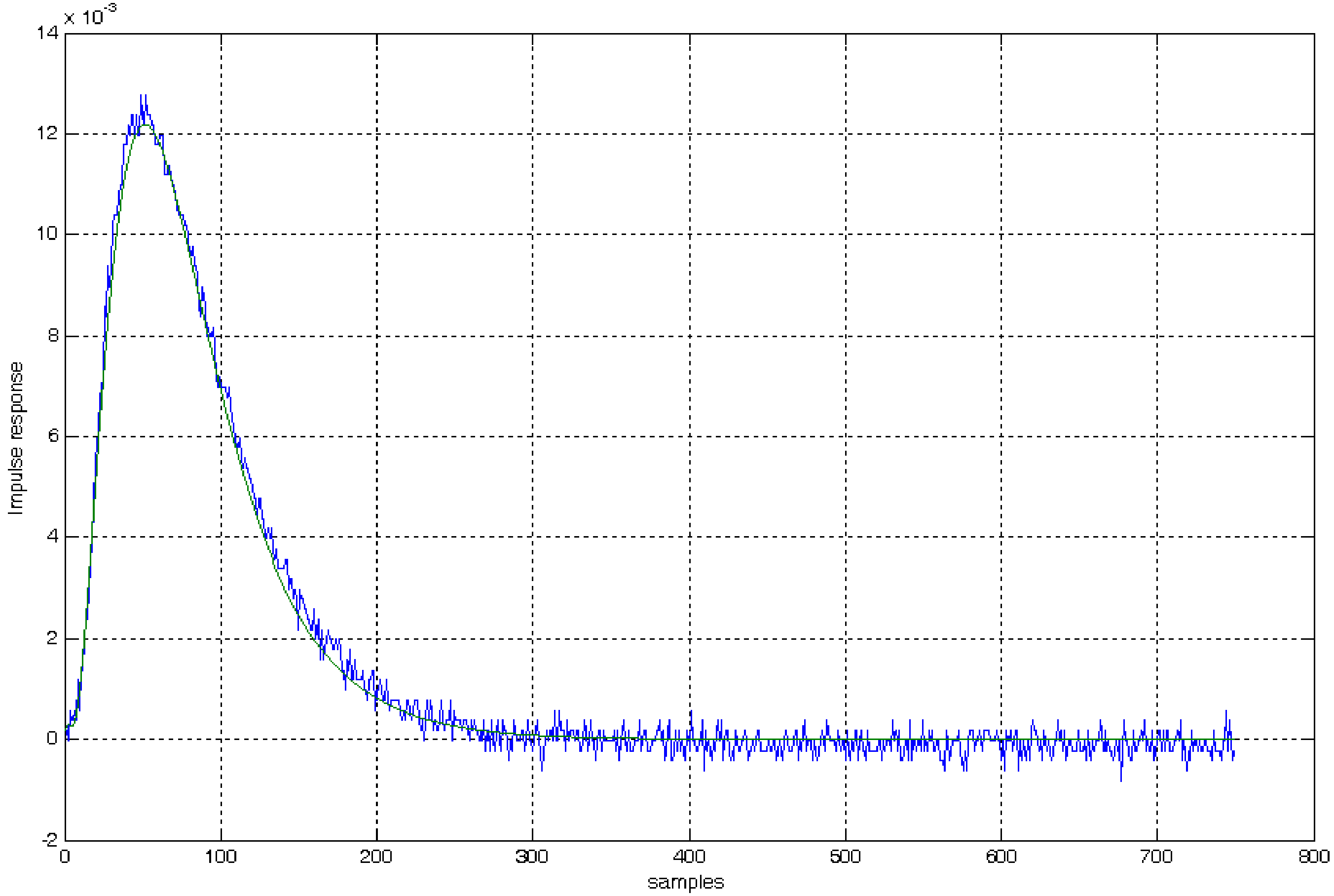
$$\tilde{H}_{ij} = \begin{bmatrix} \mathbf{U}_s & \mathbf{U}_0 \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_s & 0 \\ O & \mathbf{\Sigma}_0 \end{bmatrix} \begin{bmatrix} \mathbf{V}_s^T \\ \mathbf{V}_0^T \end{bmatrix}$$

- Estimate $H_{i,j}$ as the best rank n approximation:

$$\hat{H}_{ij} = \left(\mathbf{U}_s \mathbf{\Sigma}_s^{1/2}\right)\left(\mathbf{\Sigma}_s^{1/2} \mathbf{V}_s^T\right) \quad \rightarrow \quad \hat{\Gamma}_i = \mathbf{U}_s \mathbf{\Sigma}_s^{1/2}, \quad \widehat{\Delta}_j = \mathbf{\Sigma}_s^{1/2} \mathbf{V}_s^T$$

Experimental example

Model for a Peltier cell (n=4, i=20)



Subspace Model Identification: deterministic case

The data equation

Note that we can write the following equation (i > n)

$$\begin{bmatrix} y(t) \\ y(t+1) \\ y(t+2) \\ \vdots \\ y(t+i-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \dots \\ CA^{i-1} \end{bmatrix} x(t) + \begin{bmatrix} D & 0 & 0 & \dots & 0 \\ CB & D & 0 & \dots & 0 \\ CAB & CB & D & \dots & 0 \\ \vdots & & & & \\ CA^{i-1}B & \dots & & CB & D \end{bmatrix} \begin{bmatrix} u(t) \\ u(t+1) \\ u(t+2) \\ \vdots \\ u(t+i-1) \end{bmatrix}$$

which describes the system over a window of finite length.

The data equation (cont.d)

Repeating for various initial times we get the *data equation*

$$Y_{t,i,j} = \Gamma_i X_{t,j} + H_i U_{t,i,j}$$

where $Y_{t,i,j}$, $U_{t,i,j}$ are Hankel matrices:

$$Y_{t,i,j} = \begin{bmatrix} y(t) & \cdots & y(t+j-1) \\ y(t+1) & \cdots & y(t+j) \\ \vdots & \ddots & \vdots \\ y(t+i-1) & \cdots & y(t+i+j-2) \end{bmatrix}$$

and $X_{t,j}$ is defined as

$$X_{t,j} = \begin{bmatrix} x(t) & x(t+1) & \cdots & x(t+j-1) \end{bmatrix}$$

Orthogonal projection algorithm

The MOESP algorithm (Verhaegen and Dewilde 1991):

1. Construct projection $\Pi^{?}$ such that $U_{t,i,j}\ \Pi^{?} = 0$

2. Project data equation using $\Pi^{?}$ to recover column space of Γ_i

$$Y_{t,i,j}\Pi^{\perp} = \Gamma X_{t,j}\Pi^{\perp}$$

3. Construct a basis for the column space of Γ_l and estimate A and C.

4. Solve LS problem for estimation of B and D.

Computing the projection $\Pi^?$

We look for $\Pi^?$ such that $U_{t,i,j}\ \Pi^? = 0$.

The solution is given by

$$\Pi^{\perp} = I - U_{t,i,j}^{T}\left(U_{t,i,j}U_{t,i,j}^{T}\right)^{-1}U_{t,i,j}$$

since in fact

$$U_{t,i,j}\Pi^{\perp} = U_{t,i,j} - U_{t,i,j}U_{t,i,j}^{T}\left(U_{t,i,j}U_{t,i,j}^{T}\right)^{-1}U_{t,i,j} = 0$$

Note that constructing $\Pi^?$ requires $\left(U_{t,i,j}U_{t,i,j}^{T}\right)$ to be nonsingular.

Implementation of the projection

The projection $\Pi^?$ can be computed and implemented via the RQ factorisation:

$$\begin{bmatrix} U_{t,i,j} \\ Y_{t,i,j} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & \boxed{R_{22}} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = RQ, \quad QQ^T = \begin{bmatrix} Q_1 Q_1^T & Q_1 Q_2^T \\ Q_2 Q_1^T & Q_2 Q_2^T \end{bmatrix} = I$$

which can be written as

$$\begin{aligned} U_{t,i,j} &= R_{11} Q_1 \\ Y_{t,i,j} &= \Gamma_i X_{t,j} + H_i U_{t,i,j} = R_{21} Q_1 + R_{22} Q_2 \end{aligned}$$

and therefore

$$R_{22} = \Gamma_i X_{t,j} Q_2^T$$

Elimination of $H_iU_{t,i,j}$

Therefore, considering the equation

$$Y_{t,i,j} = \Gamma_i X_{t,j} + H_i U_{t,i,j} = R_{21}Q_1 + R_{22}Q_2$$

and right-multiplying by Q_2^T one gets

$$R_{22} = \Gamma_i X_{t,j} Q_2^T$$

so R_{22}, of dimension (il £ il) and computed from data only, contains information on Γ_i.

Under what conditions range(R_{22})=range(Γ_i)?

A rank condition

Theorem 1: if u(t) is such that

$$\text{rank}\left(\begin{bmatrix} X_{t,j} \\ U_{t,i,j} \end{bmatrix}\right) = n + im$$

then

$$\text{range}(R_{22}) = \text{range}(\Gamma_i).$$

Problem: this is not yet an identifiability condition, since it depends on the state.

However, it implies the following.

An identifiability condition

Theorem 2 (Jansson 1997):
if the input u is persistently exciting of order n+i, then

$$\lim_{N\to\infty} \frac{1}{N} \begin{bmatrix} X_{t,j} \\ U_{t,i,j} \end{bmatrix} \begin{bmatrix} X_{t,j}^T & U_{t,i,j}^T \end{bmatrix} > 0$$

(i.e., the rank condition of Theorem 1 holds).

Determination of the column space of Γ_i

Rank reduction of estimated column space of Γ_i performed via singular value decomposition of R_{22}. Under p.e. assumptions, rank(R_{22}=n), so

$$
\begin{aligned}
R_{22} &= \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \Sigma V^T = \\
&= \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} V^T \quad \Rightarrow \quad \hat{\Gamma}_i = U_n
\end{aligned}
$$

The inspection of the singular values provides information about model order.

Estimation of A and C

Let C=first l rows of computed Γ_i;

Compute A exploiting *shift invariance*, i.e., solving the system of linear equations

$$\Gamma_{\uparrow} A = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-2} \end{bmatrix} A = \begin{bmatrix} CA \\ CA^2 \\ CA \\ \vdots \\ CA^{i-1} \end{bmatrix} = \Gamma_{\downarrow}$$

A simple example (Van Der Veen et al. 1993)

Consider the LTI system ($|\alpha|<1$)

$$\begin{array}{rcl} x(t+1) & = & \alpha x(t) + \alpha u(t) \\ y(t) & = & x(t) + u(t) \end{array}$$

and apply the input sequence ($x(1)=0$)

$$u = [\ 1 \quad 2 \quad 1 \quad 1\ ]^T$$

that gives the corresponding output sequence

$$y = [\ 1 \quad 2+\alpha \quad 1+2\alpha+\alpha^2 \quad 1+\alpha+2\alpha^2+\alpha^3\ ]^T$$

A simple example (cont.d)

Choosing i=2 and j=3 we can construct the compound matrix

$$\begin{bmatrix} U_{t,i,j} \\ Y_{t,i,j} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 2+\alpha & 1+2\alpha+\alpha^2 \\ 2+\alpha & 1+2\alpha+\alpha^2 & 1+\alpha+2\alpha^2+\alpha^3 \end{bmatrix} = RQ$$

and computing the RQ factorisation we get

$$R = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 1 & 2+\alpha & \boxed{5\alpha+3\alpha^2} \\ 2+\alpha & 1+2\alpha+\alpha^2 & \boxed{5\alpha^2+3\alpha^3} \end{bmatrix}$$

A simple example (cont.d)

We can now factor R_{22} as :

$$R_{22} = \begin{bmatrix} 5\alpha + 3\alpha^2 \\ 5\alpha^2 + 3\alpha^3 \end{bmatrix} = \begin{bmatrix} 1 \\ \alpha \end{bmatrix} \begin{bmatrix} 5\alpha + 3\alpha^2 \end{bmatrix}$$

So

$$\hat{\Gamma}_2 = \begin{bmatrix} 1 \\ \alpha \end{bmatrix}$$

and finally

$$\hat{C} = 1, \quad \hat{A} = \alpha$$

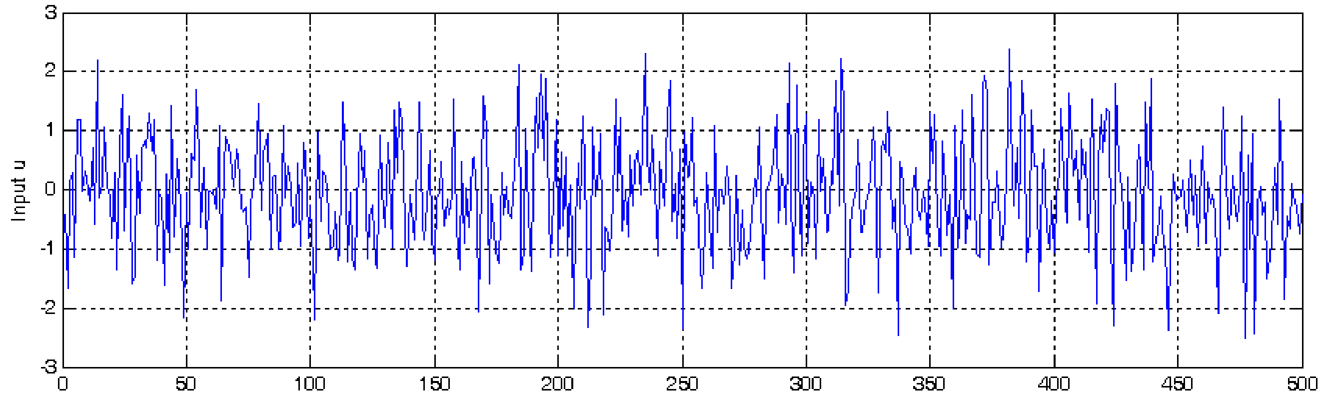
A numerical example

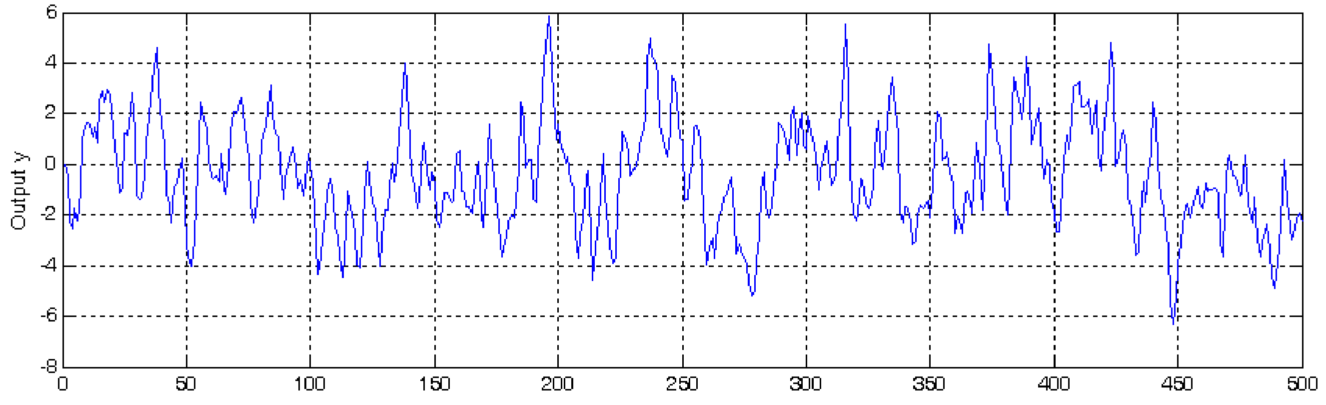
Consider the order 2 system

$$\begin{aligned} x_1(t+1) &= 0.3x_1(t) + x_2(t) + u(t) \\ x_2(t+1) &= 0.7x_2(t) + u(t) \\ y(t) &= x_1(t) \end{aligned}$$

and measure the response to a 500 samples realisation of white gaussian noise.

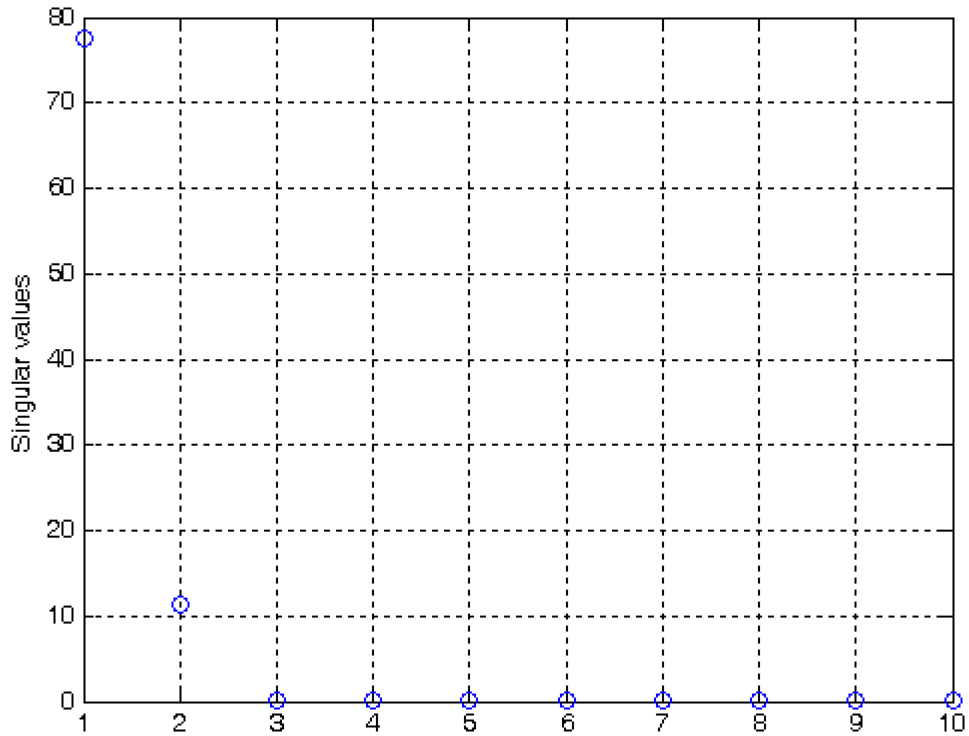
I/O data


Input u
3
2
1
0
-1
-2
-3
0
50
100
150
200
250
300
350
400
450
500


Output y
6
4
2
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
450
500

Construction of R_{22} and SVD

U_{tij} and Y_{tij} are constructed with i=10 and j=490, so R_{22} is 10 £ 10. Its singular values are given by



Estimated A and C

Numerical results of the estimation procedure:

$$A = \begin{bmatrix} 0.8033 & 0.5950 \\ -0.0874 & 0.1967 \end{bmatrix}, \quad C = \begin{bmatrix} -0.5897 & 0.7799 \end{bmatrix}$$

Note that

- The computed A and C are in a different state space basis from the original system;
- They are equivalent to the original A and C;
- Question: what determines the basis of the estimated matrices?

MATLAB code for the estimation of A and C

```
function [A,C]=omoesp(u,y,i,j,n);

sy=size(y);su=size(u);
datalen=min([max(sy) max(su)]);
m=min(su); l=min(sy);

H=[];
for ii=1:i
     H=[H u(ii:ii+j-1,:)];
end

for ii=1:i
     H=[H y(ii:ii+j-1,:)];
end

R=triu(qr(H))';

R22=R(m*i+1:(m+l)*i,m*i+1:(m+l)*i);

[U,S,Vt]=svd(R22);

Un=U(:,1:n);

C=Un(1:l,:);

A=Un(1:l*(i-1),:)\Un(l+1:l*i,:);
```

Estimation of B and D

The output of the identified model is given by:

$$\hat{y}(t) = Du(t) + \sum_{r=0}^{t-1} \mathbf{C}\mathbf{A}^{t-r-1} Bu(r)$$

we aim at writing the above as a linear regression in the elements of B and D:

$$\hat{y}(t) = \phi_D^T(t) vec(\mathbf{D}) + \phi_B^T(t) vec(\mathbf{B})$$

where for X 2 R$^{(\text{m £ n})}$

$$vec(X) = \begin{bmatrix} x_{11} & \dots & x_{m1} & x_{12} & \dots & x_{m2} & x_{1n} & \dots & x_{mn} \end{bmatrix}^T$$

For this, we need to introduce Kronecker products.

The Kronecker product

Let A 2 $R^{(m\ £\ n)}$ and B 2 $R^{(r\ £\ s)}$, then the (mr £ ns) matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & & & \vdots \\ a_{m1}B & a_{m2}B & \dots a_{mn}B & \end{bmatrix}$$

is called the Kronecker product of A and B.

vec operation and Kronecker product

There is a connection between Kronecker products and the vec operation.

Let A 2 R$^{(m £ n)}$, B 2 R$^{(n £ o)}$, C 2 R$^{(o £ p)}$, then

$$vec(ABC) = \left(C^T \otimes A\right) vec(B)$$

Estimation of B and D (cont.d)

Using Kronecker products the output of the identified model

$$\hat{y}(t) = Du(t) + \sum_{r=0}^{t-1} \mathbf{C}\mathbf{A}^{t-r-1} Bu(r)$$

can be written as

$$\hat{y}(t) = [u(t)^T \otimes \mathbf{I}_l] vec(\mathbf{D}) + (\sum_{r=0}^{t-1} u(r)^T \otimes \mathbf{C}\mathbf{A}^{t-r-1}) vec(\mathbf{B})$$

so that B and D can be obtained from:

$$B, D = \arg\min_{B,D} \sum_{k=0}^{s} \left[y(t) - [u(t)^T \otimes \mathbf{I}_l] vec(\mathbf{D}) - (\sum_{r=0}^{t-1} u(r)^T \otimes \mathbf{C}\mathbf{A}^{t-r-1}) vec(\mathbf{B}) \right]^2$$

which is clearly a least squares problem in B and D.

Subspace Model Identification: output error case

SMI: output error case

Consider the finite dimensional,
linear time-invariant (LTI) state space model:

$$x(t+1) = Ax(t) + Bu(t)$$

with the measurement equation

$$y(t) = Cx(t) + Du(t) + v(t)$$

where v is a zero-mean, white measurement noise, uncorrelated with u.
We want to analyse the effect of v on the identification algorithm we studied in the deterministic case.

The data equation with measurement error

When adding measurement noise the data equation becomes

$$Y_{t,i,j} = \Gamma_i X_{t,j} + H_i U_{t,i,j} + V_{t,i,j}$$

where $V_{t,i,j}$ is defined as

$$V_{t,i,j} = \begin{bmatrix} v(t) & \cdots & v(t+j-1) \\ v(t+1) & \cdots & v(t+j) \\ \vdots & \ddots & \vdots \\ v(t+i-1) & \cdots & v(t+i+j-2) \end{bmatrix}$$

Effect of measurement noise

As in the deterministic case, we:

- Construct projection $\Pi^?$ such that $U_{t,i,j}$ $\Pi^?$ =0
- Project data equation using $\Pi^?$ to recover column space of Γ_i

$$Y_{t,i,j}\Pi^{\perp} = \Gamma X_{t,j}\Pi^{\perp} + V_{t,i,j}\Pi^{\perp}$$

- Using the RQ factorisation we obtain now

$$R_{22} = \Gamma_i X_{t,j} Q_2^T + V_{t,i,j} Q_2^T$$

Asymptotic properties of R_{22}

Can we use R_{22} to estimate the observability subspace?

Theorem 3:

if v ' WN(0, σ^2I) and u is p.e. of order n+i, then

$$\lim_{N\to\infty} \frac{1}{N} R_{22} R_{22}^T = \Gamma_i M \Gamma_i^T + \sigma^2 I_{il}, \quad M = \lim_{N\to\infty} \frac{1}{N} X_{t,j} \Pi^{\perp} X_{t,j}^T$$

and

$$\lim_{N\to\infty} \frac{1}{N} R_{22} R_{22}^T = \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n + \sigma^2 I_n & 0 \\ 0 & \sigma^2 I_{il-n} \end{bmatrix} U^T$$

A numerical example

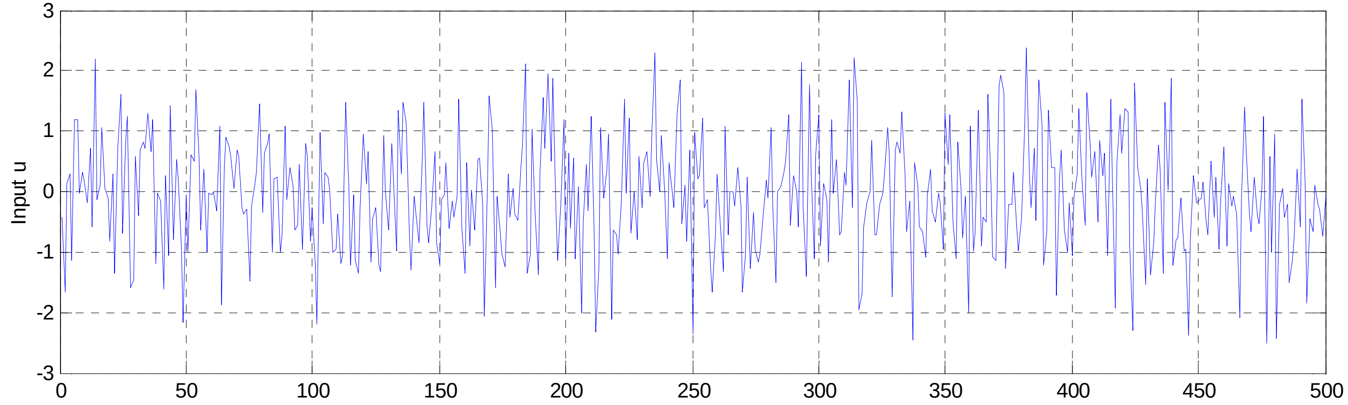
Consider again the order n=2 system

$$
\begin{aligned}
x_1(t+1) &= 0.3x_1(t) + x_2(t) + u(t) \\
x_2(t+1) &= 0.7x_2(t) + u(t) \\
y(t) &= x_1(t) + v(t)
\end{aligned}
$$

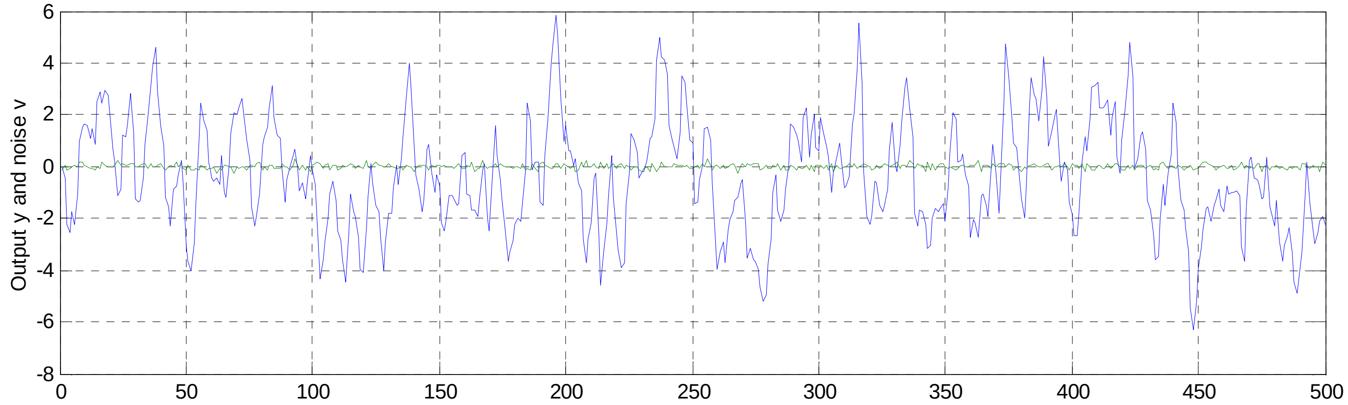
and measure the response to a 500 samples realisation of white gaussian noise, subject to v ' (0,0.01).

We repeat the identification 1000 times, with different realisations of the noise v to assess the *average* effect of measurement noise.

I/O data



Input u
3
2
1
0
-1
-2
-3
0
50
100
150
200
250
300
350
400
450
500

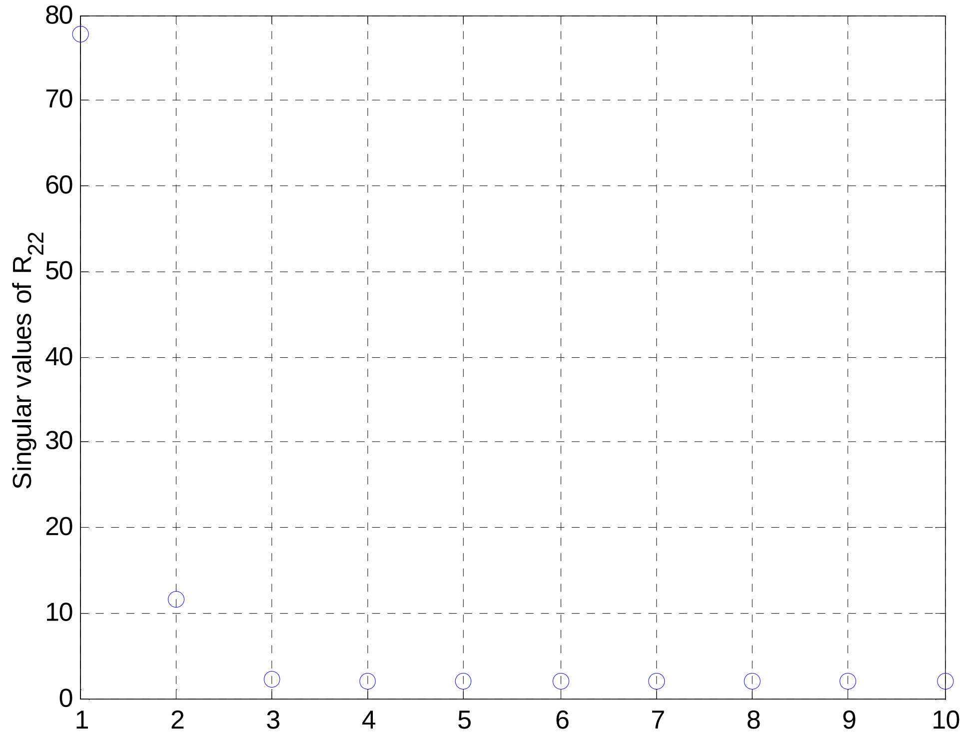


Output y and noise v
6
4
2
0
-2
-4
-6
-8
0
50
100
150
200
250
300
350
400
450
500

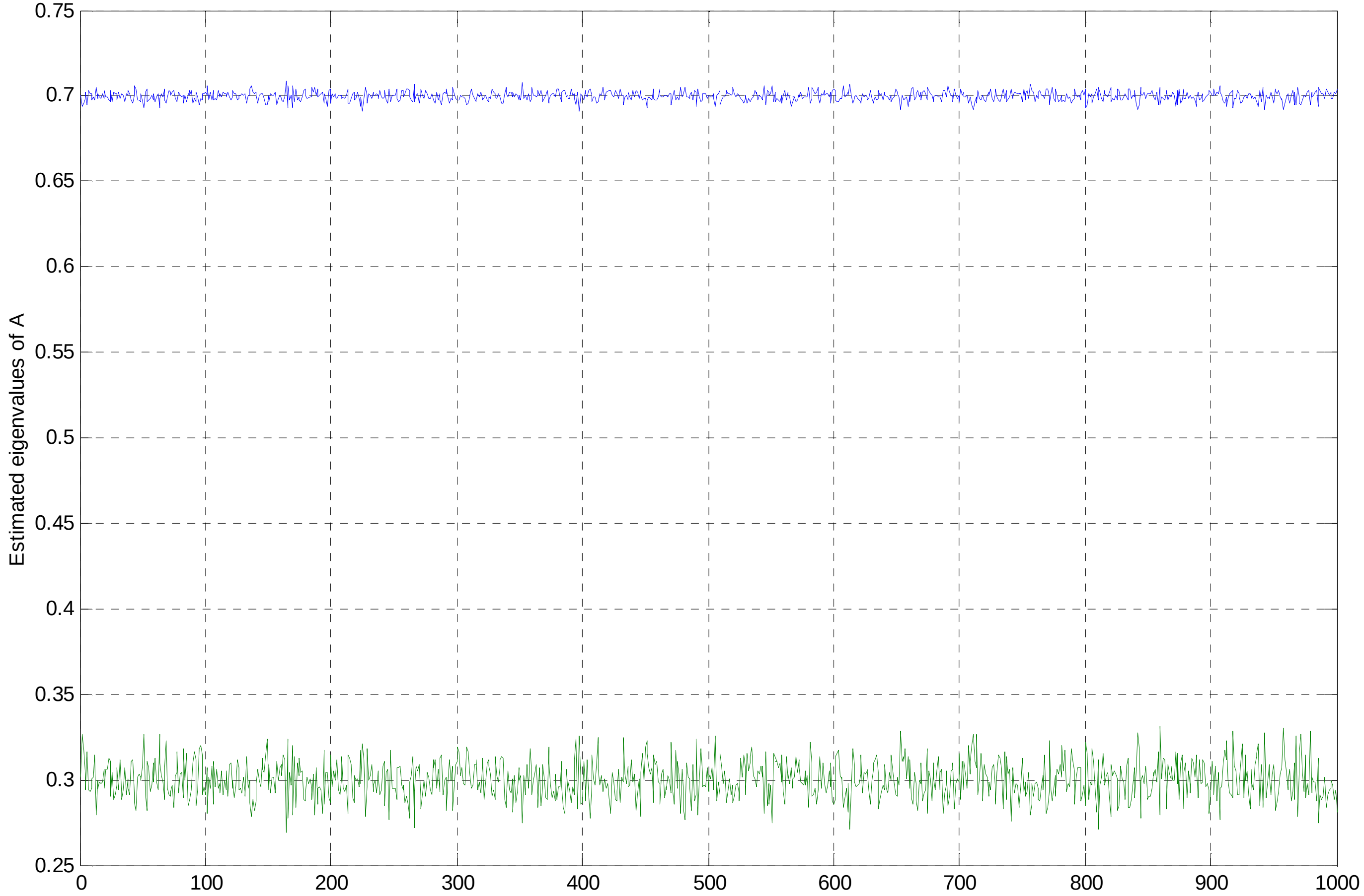
Construction of R_{22} and SVD

R_{22} is constructed with i=10 and j=490.

Its singular values are given by



Estimated eigenvalues of A



Subspace Model Identification: the general case

SMI: the general case

Consider the finite dimensional,
linear time-invariant (LTI) state space model:

$$x(t+1) \;=\; Ax(t) + Bu(t) + w(t)$$

with the measurement equation

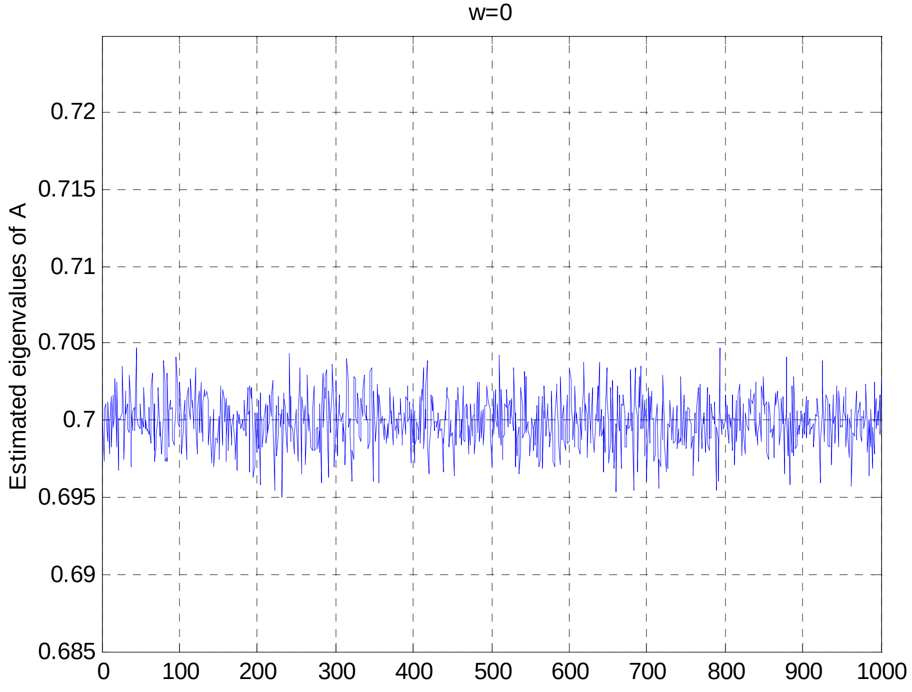
$$y(t) \;=\; Cx(t) + Du(t) + v(t)$$

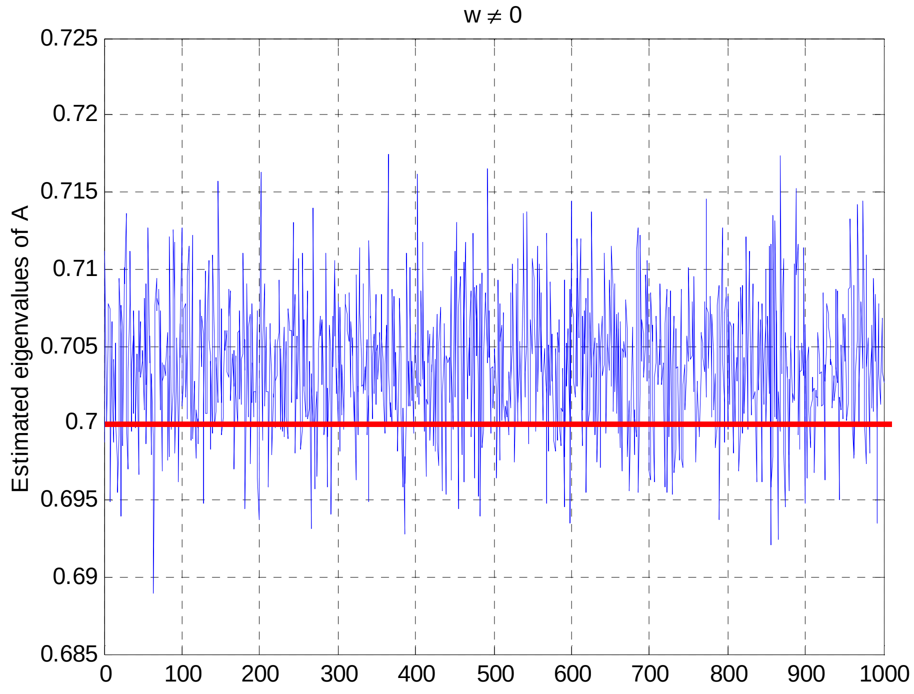
with w and v zero-mean white noises, uncorrelated with u.

Does the orthogonal projection algorithm still work?

Example

Consider the n=1 system A=0.7; B=1; C=1; D=0; and compare the performance of the SMI algorithm with and without process noise w:





What happened?

When process noise is present, the data equation becomes

$$Y_{t,i,j} = \Gamma_i X_{t,j} + H_i U_{t,i,j} + \boxed{E_i W_{t,i,j} + V_{t,i,j}}$$

and therefore the residual is not white anymore and the results we have seen so far do not hold.

The problem can be solved by introducing Instrumental Variables.

Instrumental variable (IV) algorithms

Assume that a matrix Z (Instrumental Variable) can be found such that

$$rank\left(\lim_{N\to\infty}\frac{1}{N}(X_{t,j}\Pi^{\perp})Z^T\right) = n \qquad \lim_{N\to\infty}\frac{1}{N}(E_i W_{t,i,j}+V_{t,i,j})Z^T = 0$$

Then the column space of Γ_i can be estimated from

$$Y_{t,i,j}\Pi^{\perp}Z^T = \Gamma X_{t,j}\Pi^{\perp}Z^T+(E_i W_{t,i,j}+V_{t,i,j})\Pi^{\perp}Z^T$$

Implementation issues

The term $Y_{t,i,j}\Pi^{?}Z^{T}$ can be computed from the RQ factorisation

$$\begin{bmatrix} U_{t,i,j} \\ Z \\ Y_{t,i,j} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix}$$

And it holds that

$$Y_{t,i,j}\Pi^{\perp}Z^{T} = R_{32}R_{22}^{T}$$

and therefore

$$\text{range}\left(\Gamma_i\right) = \text{range}\left(\lim_{N\to\infty}\frac{1}{N}R_{32}R_{22}^{T}\right)$$

How to choose the IVs

Possible choice of IVs (MOESP-PO, Verhaegen 1994):

- Consider the available I/O data set and split it in two parts (past and future), the second shifted ahead of i samples with respect to the first;
- Write two separate data equations for past and future data:

$$Y_1 = \Gamma_i X_1 + H_i U_1 + E_i W_1 + V_1$$
$$Y_2 = \Gamma_i X_2 + H_i U_2 + E_i W_2 + V_2$$

- Use past data as IVs in the future data equation;

Implementation issues

- Using Past Inputs and Outputs as IVs one can compute the RQ factorisation

$$\begin{bmatrix} U_2 \\ Z \\ Y_2 \end{bmatrix} = \begin{bmatrix} U_2 \\ \begin{bmatrix} U_1 \\ Y_1 \end{bmatrix} \\ Y_2 \end{bmatrix} = \begin{bmatrix} R_{11} & 0 & 0 & 0 \\ R_{21} & R_{22} & 0 & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ R_{41} & \boxed{R_{42} \quad R_{43}} & & R_{44} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{bmatrix}$$

$$\text{range}\left(\lim_{N\to\infty} \frac{1}{\sqrt{N}} \begin{bmatrix} R_{42} & R_{43} \end{bmatrix} \right) = \text{range}\left(\Gamma_i\right)$$

- Rank reduction of estimated column space of Γ_i performed via a singular value decomposition.

Persistency of excitation conditions

- An input u which is p.e. of order n+2i will “almost always” lead to a consistent estimate of A and C.

- The theory for the IV algorithm is not complete yet…

MATLAB code for the estimation of A and C

```
function [A,C]=moesppo(u,y,i,j,n);

sy=size(y);su=size(u);
datalen=min([max(sy) max(su)]);
m=min(su); l=min(sy);
Up=[];Uf=[];Yp=[];Yf=[];

for ii=1:i
    Up=[Up u(ii:ii+j1,:)];
    Yp=[Yp y(ii:ii+j1,:)];
end

for ii=i+1:2*i
    Uf=[Uf u(ii:ii+j1,:)];
    Yf=[Yf y(ii:ii+j1,:)];
end
```

```
R=triu(qr([Uf Up Yp Yf]))';

R4243=R((2*m+l)*i+1:2*(m+l)*i,m*i+
    1:(2*m+l)*i);

[U,S,Vt]=svd(R4243);

Un=U(:,1:n);

C=Un(1:l,:);

A=Un(1:l*(i1),:)``Un(l+1:l*i,:);
```

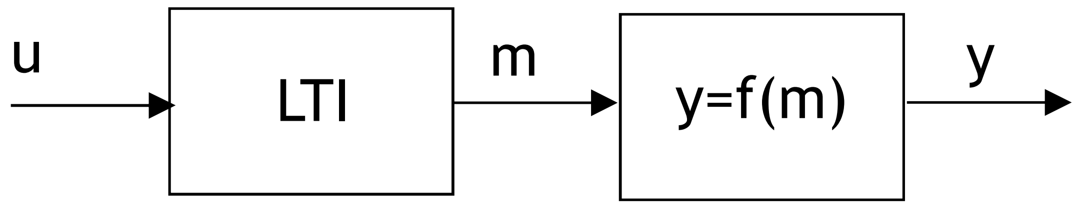
(Some) extensions of SMI algorithms

- Recursive versions of all the presented algorithms;

- Identification of linear models in *continuous time*:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) + w(t)\\ y(t) &= Cx(t) + Du(t) + v(t)\end{aligned}$$

- Identification of classes of nonlinear models, including, e.g., Wiener models:



Other (important) topics

- Choice of parameter i:
  - The choice of i affects the variance of the estimates;
  - No general guidelines except for condition i >> n;

- Asymptotic variance of the estimated [A,B,C,D] matrices:
  - Analytical expressions for the variance of the estimates exist;
  - Expressions too complicated to be of practical use!
  - The estimates are asymptotically Gaussian;
  - No results available for efficiency;

SMI vs Prediction Error Methods

Advantages:

- SMI algorithms work equally well for SISO and MIMO problems;
- They are very reliable from the numerical point of view;

Disadvantages:

- SMI algorithms are not “optimal” in any sense;
- Very difficult to use them for structured problems;

Available software tools

- Functions for SMI in the System Identification Toolbox for Matlab;
- Dedicated SMI Toolbox, again based on Matlab;
- Fast code in C and Fortran available in the Slicot library.