Numerical methods in identification, control and signal processing: system identification, array signal processing

Marco Lovera
Dipartimento di Elettronica e Informazione
Politecnico di Milano
lovera@elet.polimi.it

Outline

- Least Squares and Total Least Squares identification;
- Identification from impulse response data;
- Subspace Model Identification methods;
- Array signal processing;
- Recursive state space identification algorithms;

Dipartimento di Elettronica e Informazione

POLITECNICO DI MILANO

Least Squares and Total Least Squares

I/O SysId problems leading to LS

All I/O model classes used in SysId which are linear in the parameters lead to LS estimation problems.
Some examples:

- AR and ARX linear models;
- Bilinear models;
- Polynomial NARX models;
- LPV models in I/O form;

General form of the linear regression

Whenever a linear-in-the-parameters model is considered, it can be written as

$$y(t) = \phi^T(t)\theta + \epsilon(t), \quad t = 1, \dots, N$$

and the problem of estimating the parameter vector θ on the basis of N samples of I/O data can be formulated as

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} (y(i) - \phi^T(i)\theta)^2 = \frac{1}{N} \sum_{i=1}^{N} \epsilon(i)^2$$

General form of the linear regression

The problem can be reformulated as the one of solving the set of equations:

$$\begin{aligned} y(1) &= \phi^T(1)\theta \\ y(2) &= \phi^T(2)\theta \\ \vdots &= \vdots \\ y(N) &= \phi^T(N)\theta \end{aligned}$$

or, equivalently:

$$\begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \vdots \\ \phi^T(N) \end{bmatrix} \theta$$

The LS problem in Ax=b form

Problem: find x $\in \mathbb{R}^n$ such that Ax=b where

- A $\in \mathbb{R}^{m \times n}$, m>n, is the data matrix;
- b $\in \mathbb{R}^m$ is the observation vector.

Since the system is overdetermined (m>n) the condition b $\in$ range(A), which guarantees existence of an exact solution, is likely to be violated.
We will therefore look for x such that

$$\min_{x \in \mathcal{R}^n} \|Ax - b\|_2$$

The LS problem in Ax=b form (cont.d)

Two important properties of the LS problem:

- The LS cost function

$$\phi(x) = \frac{1}{2}\|Ax - b\|_2^2$$

  is differentiable, so we can associate the minimisers to the gradient equation $\nabla\phi(x)=0$;

- The 2-norm is invariant under unitary multiplication, so we can look for unitary Q such that

$$\min_{x\in\mathcal{R}^n} \|Q^T Ax - Q^T b\|_2$$

  is easier to solve.

The LS residual

The quantity

$$r_{LS} = Ax_{LS} - b$$

is termed the minimum residual and its norm is denoted by

$$\rho_{LS} = \|Ax_{LS} - b\|_2$$

The size of r_{LS} measures our ability to predict b using the columns of A.

The normal equations

If A has *full column rank*, then the solution x_{LS} of the least squares problem is unique and it solves the linear system

$$A^T A x_{LS} = A^T b, \quad (A^T A > 0)$$

usually called the *normal equations*.

Note that

$$\phi(x) = \frac{1}{2}\|Ax-b\|_2^2 = \frac{1}{2}(Ax-b)^T(Ax-b), \quad \rightarrow \nabla\phi(x) = A^T(Ax-b)$$

so solving the normal equations is equivalent to imposing $\nabla\phi(x)=0$.

The normal equations (cont.d)

Implication of the normal equations: the residual is given by

$$r_{LS} = b - Ax_{LS} = b - A(A^TA)^{-1}A^Tb = b - P_Ab$$

where P_A is the orthogonal projector on range(A). In particular, $r_{LS} \perp$ range(A).

Therefore the LS problem can be also stated as: find the minimal perturbation r_{LS} to b such that (b-r_{LS}) can be predicted by the columns of A!

The full rank condition

In terms of the linear regression problem

$$y(t) = \phi^T(t)\theta + \epsilon(t) \qquad \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} (y(i) - \phi^T(i)\theta)^2$$

the normal equations can be written as

$$\sum_{i=1}^{N} \phi(i)\phi^T(i)\theta = \sum_{i=1}^{N} \phi(i)y(i)$$

so the full rank condition corresponds to

$$\sum_{i=1}^{N} \phi(i)\phi^T(i) > 0$$

i.e., to a persistency of excitation condition within the identification framework.

Solving the LS problems

- Direct solution of the normal equations;
- QR factorisation approach;
- SVD approach;

Normal equations approach

One can solve the normal equations using the following algorithm:

- Compute $C=A^TA$ and $d=A^Tb$;
- Compute the Cholesky factorisation $C=GG^T$;
- Solve $Gy=d$;
- Solve $G^Tx_{LS}=y$.

Accuracy: $$\frac{\|\widehat{x}_{LS}-x_{LS}\|}{\|x_{LS}\|} \simeq O(\epsilon_{machine})k_2(A)^2$$

QR factorisation approach

Assume Q has been computed such that

$$Q^T A = R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$$

and let

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix}$$

then

$$\|Ax - b\|_2^2 = \|Q^T Ax - Q^T b\|_2^2 = \|R_1 x - c\|_2^2 + \|d\|_2^2$$

QR factorisation approach (cont.d)

Clearly if rank(A)=rank(R_1)=n then x_{LS} is given by the solution of

$$R_1 x_{LS} = c$$

and the residual of the LS problem is given by

$$\rho_{LS} = \|d\|_2$$

SVD approach

The solution of the full rank LS problem is best expressed in terms of the SVD of A:

Theorem

Suppose $U^TAV=\Sigma$ is the SVD of A, where

$U=[u_1,u_2,\cdots,u_m]$ and $V=[v_1,v_2,\cdots,v_n]$.

Then:

$$x_{LS} = \sum_{i=1}^{n} \frac{u_i^T b}{\sigma_i} v_i$$

$$\rho_{LS}^2 = \sum_{i=n+1}^{m} (u_i^T b)^2$$

SVD approach in the rank deficient case

The solution of the rank LS problem can be written in terms of the SVD of A also in the rank deficient case:

Theorem

Suppose $U^T AV = \Sigma$ is the SVD of A, rank(A)=r<n, where $U=[u_1,u_2,\cdots,u_m]$ and $V=[v_1,v_2,\cdots,v_n]$.

Then:

$$x_{LS} = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i$$

is the smallest 2-norm minimiser, and

$$\rho_{LS}^2 = \sum_{i=r+1}^{m} (u_i^T b)^2$$

SVD approach in the rank deficient case

Proof.

$$\|Ax-b\|_2^2 = \|(U^TAV)(V^Tx) - U^Tb\|_2^2 = \|\Sigma\alpha - U^Tb\| =$$

$$= \sum_{i=1}^{r} (\sigma_i\alpha_i - u_i^Tb)^2 + \sum_{r+1}^{m} (u_i^Tb)^2$$

where $\alpha=V^Tx$.

We obtain a minimiser by choosing

$$\alpha_i = \frac{u_i^Tb}{\sigma_i}, \quad i = 1, \ldots, r$$

and the minimum norm minimiser by setting $\alpha_i=0$ for $n \leq i \leq r+1$.

Total Least Squares

In classical LS problems matrix A is assumed to be known exactly.
What if A comes from *noisy* measurements, just as b?

A simple example: the scalar model $\alpha x = \beta$ in which x must be estimated from measurements

$a_i = a_i^o + \Delta a_i$ and $b_i = b_i^o + \Delta b_i$, $i = 1, \ldots, m$ of α and β .

Total Least Squares (cont.d)

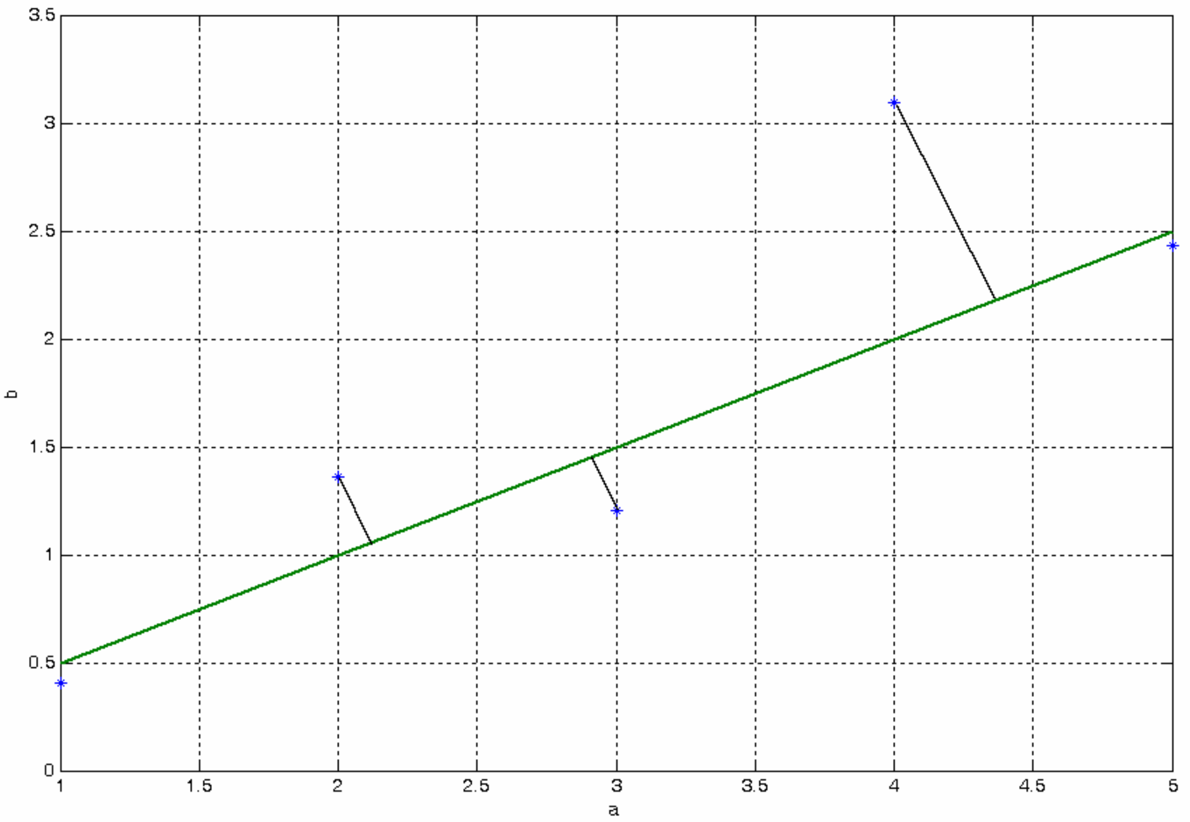
LS case: $\Delta a_i=0$, so the best estimate is given by

$$x_{LS} = \frac{\sum_{i=1}^{m} a_i b_i}{\sum_{i=1}^{m} a_i^2}$$

TLS case: $\Delta a_i \neq 0$ and $\Delta b_i \neq 0$, independent and identically distributed; then the best estimate is given by the *orthogonal regression*.

Total Least Squares (cont.d)

Example: illustration of the scalar model $\alpha x = \beta$



Total Least Squares: applications

- Errors In Variables (EIV) problems;
- Orthogonal Least Squares fitting;
- Modelling situations which require *symmetric* treatment of variables;
- Parameter estimation vs prediction problems;

Total Least Squares: formulation

Given the overdetermined linear system Ax=b, with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^{m}$,

$$\min_{\left[\hat{A} \ \ \hat{b}\right] \in \mathcal{R}^{m \times (n+1)}} \| \left[A \ \ b\right] - \left[\hat{A} \ \ \hat{b}\right] \|_F$$

subject to the constraint

$$\hat{b} \in \text{range}(\hat{A})$$

Once a minimiser is found, any x:

$$\hat{A}x = \hat{b}$$

is called a TLS solution.

Total Least Squares and the SVD

Let

$$V^T \begin{bmatrix} A & b \end{bmatrix} U = \Sigma$$

the SVD of [A b] and assume that $\sigma_{n+1} \neq 0$. Then, the set of equations

$$\begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} \simeq 0$$

is incompatible, and we must look for a rank n approximation of [A b] in order to solve it.

Total Least Squares and the SVD (cont.d)

This is done using the Eckart-Young-Mirsky Theorem, giving the optimal rank n approximation as

$$\begin{bmatrix}\hat{A} & \hat{b}\end{bmatrix} = U\hat{\Sigma}V^T, \qquad \hat{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n, 0)$$

such that

$$\begin{bmatrix}A & b\end{bmatrix} - \begin{bmatrix}\hat{A} & \hat{b}\end{bmatrix} = \sigma_{n+1} u_{n+1} v_{n+1}^T$$

$$\sigma_{n+1} = \min_{rank(\begin{bmatrix}\hat{A} & \hat{b}\end{bmatrix})} \| \begin{bmatrix}A & b\end{bmatrix} - \begin{bmatrix}\hat{A} & \hat{b}\end{bmatrix} \|_F$$

Total Least Squares and the SVD (cont.d)

The system

$$\begin{bmatrix} \hat{A} & \hat{b} \end{bmatrix} \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

and its solution is given by v_{n+1}, suitably scaled to have the last element equal to -1

$$\begin{bmatrix} \hat{x} \\ -1 \end{bmatrix} = \frac{-1}{v_{n+1,n+1}} v_{n+1}$$

provided that $v_{n+1,n+1} \neq 0$.

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

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

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

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

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

Ho-Kalman realisation theory (cont.d)

Properties of the Hankel matrix:

- $H_{i,j}$, i,j $\geq$ 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:

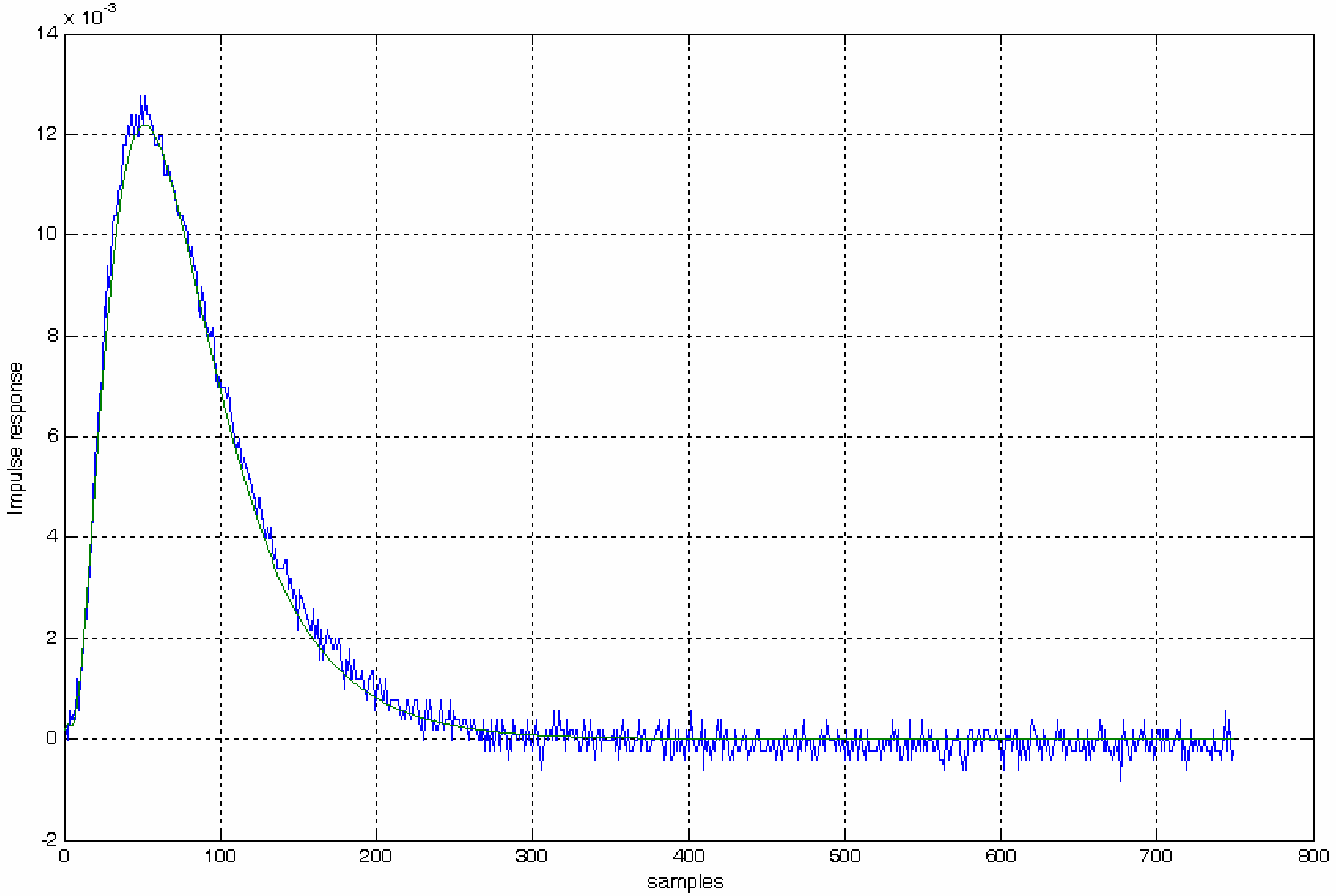
$$\tilde{H}_{ij} = \left[\begin{array}{cc} \mathbf{U}_s & \mathbf{U}_0 \end{array} \right] \left[\begin{array}{cc} \mathbf{\Sigma}_s & 0 \\ O & \mathbf{\Sigma}_0 \end{array} \right] \left[\begin{array}{c} \mathbf{V}_s^T \\ \mathbf{V}_0^T \end{array} \right]$$

- 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)

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

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 $\Pi^\perp$ such that $U_{t,i,j}\ \Pi^\perp$ =0

2. Project data equation using $\Pi^\perp$ 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 $\Pi^{\perp}$

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

The solution is given by

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

since in fact

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

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

Implementation of the projection

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

$$\begin{bmatrix} U_{t,i,j} \\ Y_{t,i,j} \end{bmatrix} = \begin{bmatrix} R_{11} & 0 \\ R_{21} & \boxed{R_{22}} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = RQ, \quad QQ^T = \begin{bmatrix} Q_1Q_1^T & Q_1Q_2^T \\ Q_2Q_1^T & Q_2Q_2^T \end{bmatrix} = I$$

which can be written as

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

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 (|α|<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

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

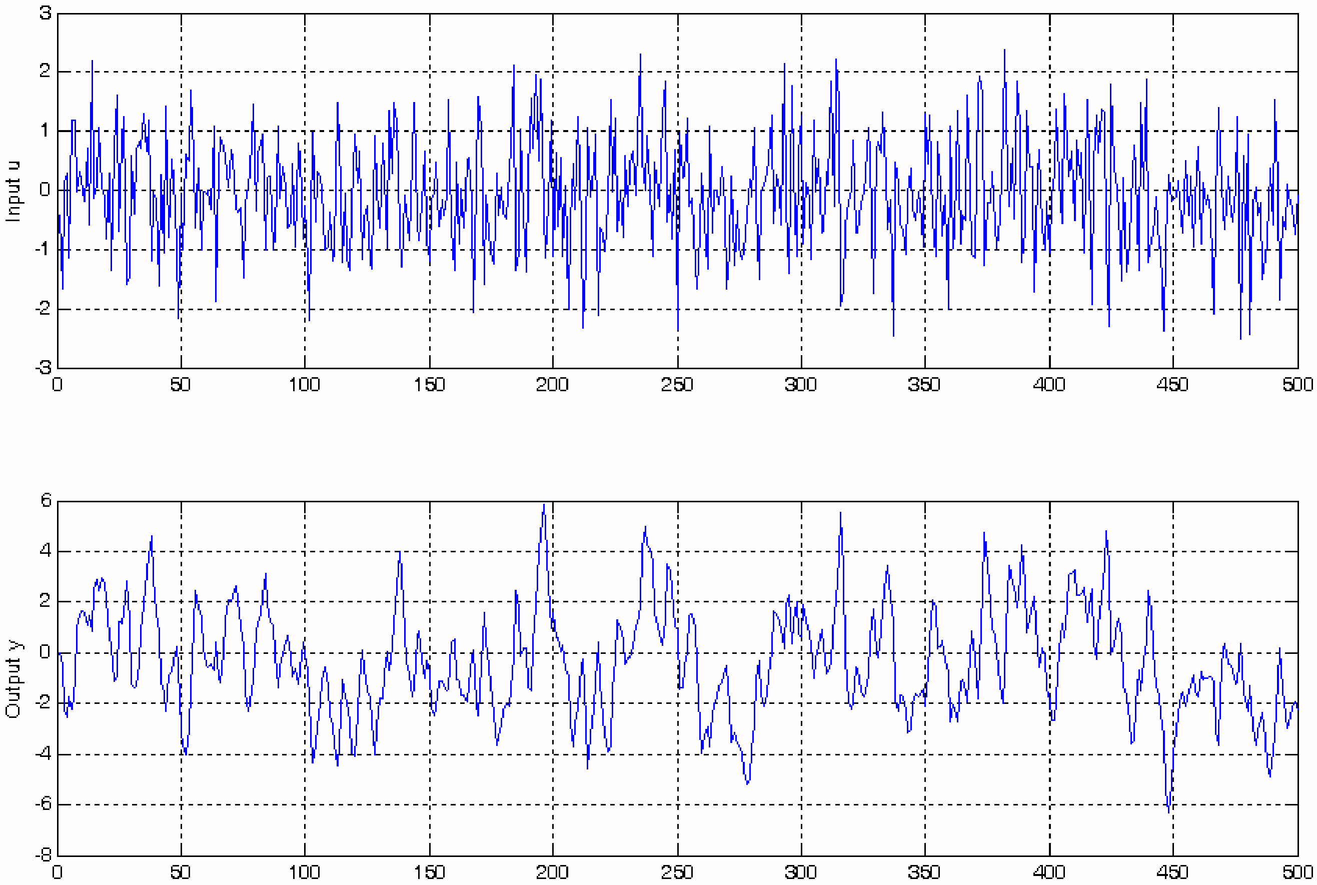
A numerical example

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.

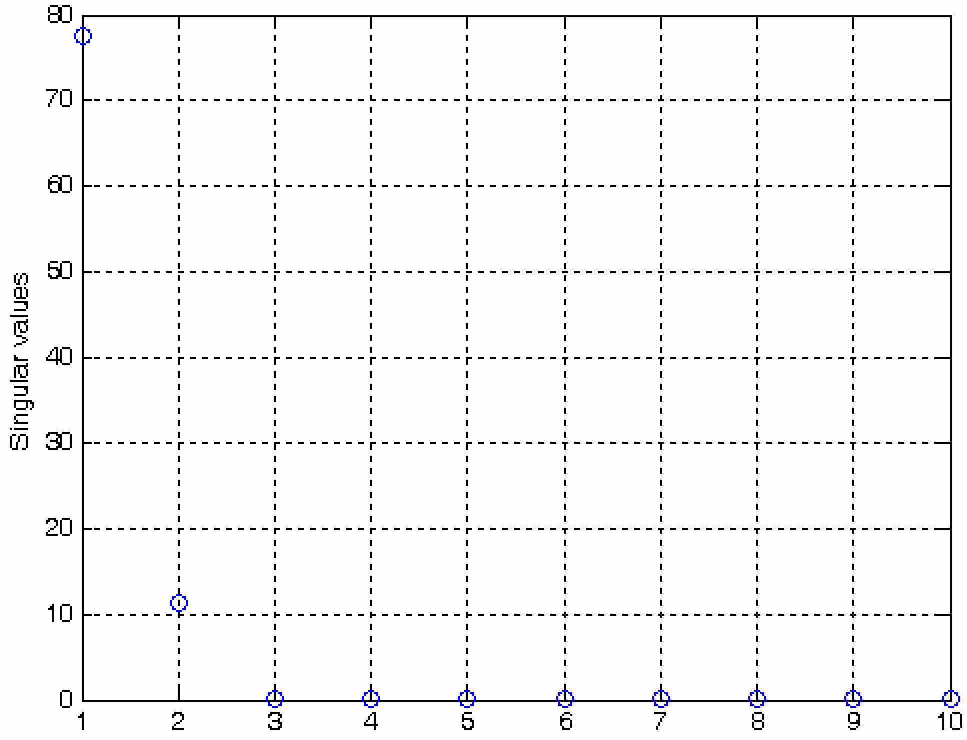
I/O data



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 $X \in \mathbb{R}^{(m \times n)}$

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

For this, we need to introduce Kronecker products.

The Kronecker product

Let A $\in \mathbb{R}^{(m \times n)}$ and B $\in \mathbb{R}^{(r \times s)}$, then the (mr $\times$ 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 $\in \mathbb{R}^{(m \times n)}$, B $\in \mathbb{R}^{(n \times o)}$, C $\in \mathbb{R}^{(o \times p)}$, 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 $\Pi^{\perp}$ such that $U_{t,i,j}\ \Pi^{\perp}$ =0

- Project data equation using $\Pi^{\perp}$ 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 $\simeq$ WN(0, σ^2I) and u is p.e. of order n+i, then

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

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

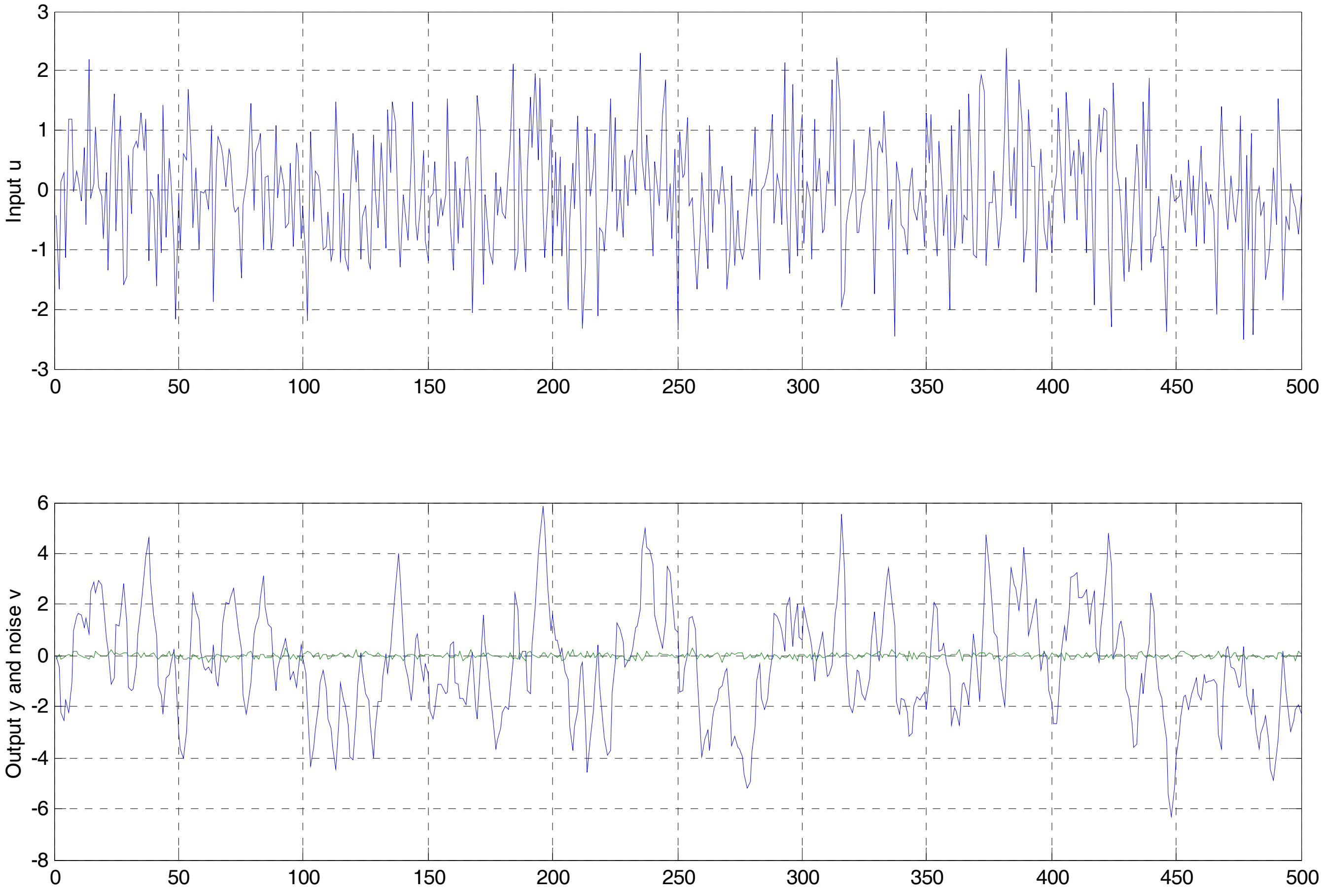
Consider again the order n=2 system

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

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

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

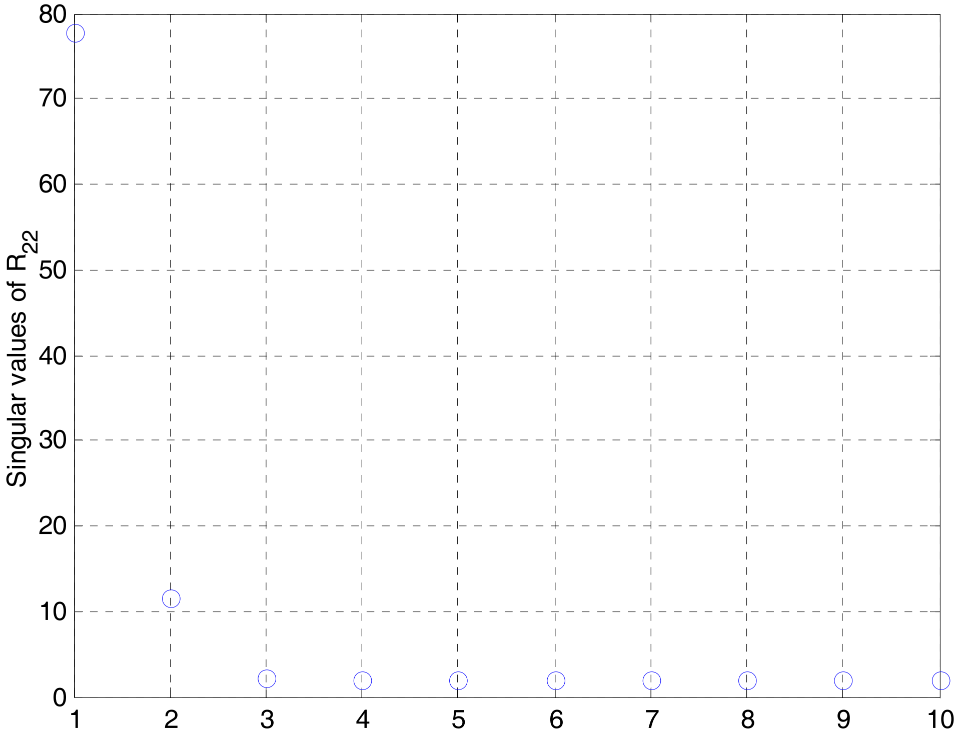
I/O data


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

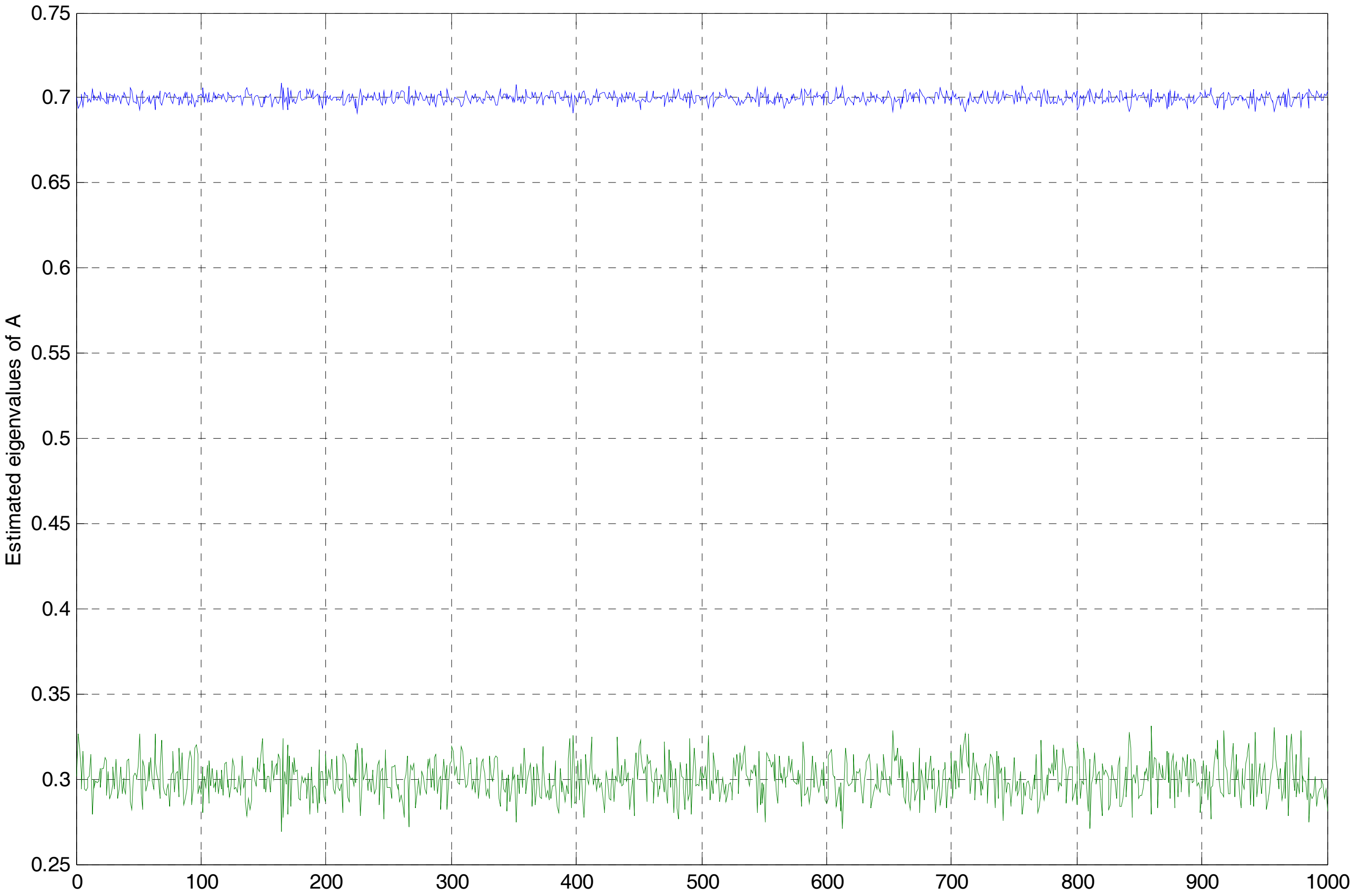
Construction of R_{22} and SVD

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

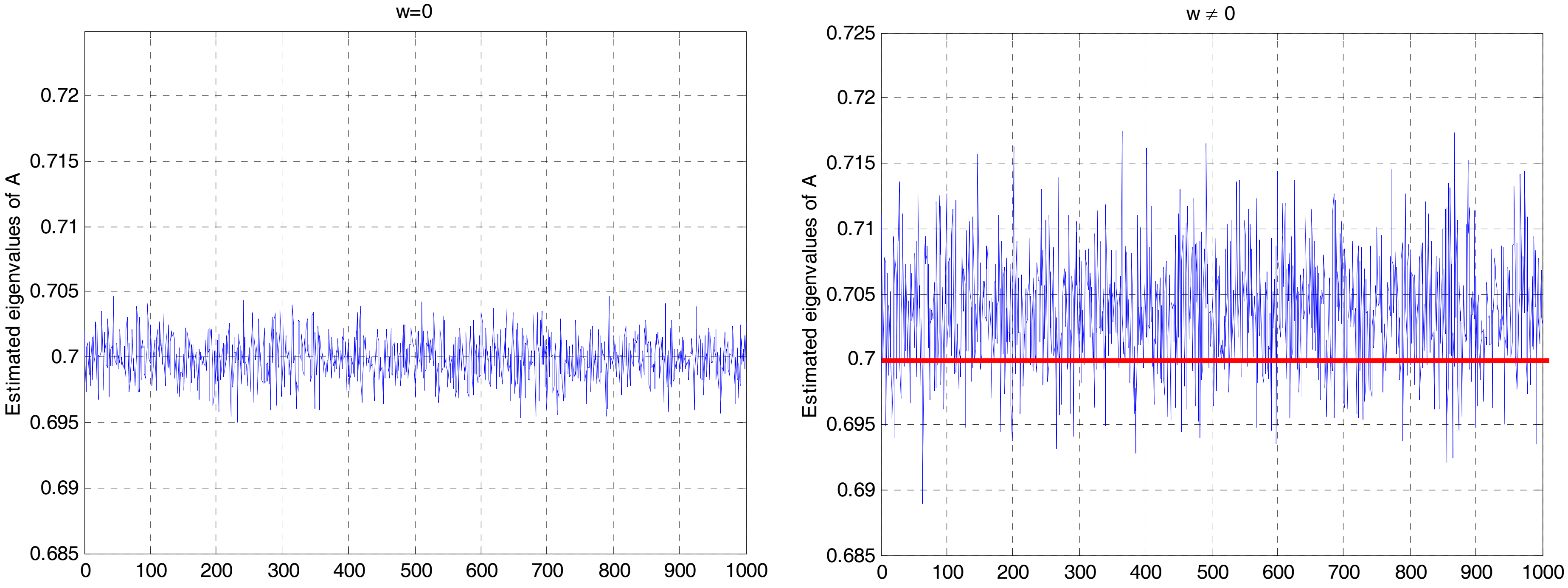
$$y(t) \;=\; Cx(t) + Du(t) + v(t)$$

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

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

Then the column space of Γ_i can be estimated from

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

Implementation issues

The term $Y_{t,i,j}\Pi^{\perp}Z^{T}$ 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

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

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+j-1,:)];
    Yp=[Yp y(ii:ii+j-1,:)];
end

for ii=i+1:2*i
    Uf=[Uf u(ii:ii+j-1,:)];
    Yf=[Yf y(ii:ii+j-1,:)];
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*(i-1),:)` `Un(l+1:l*i,:);
```

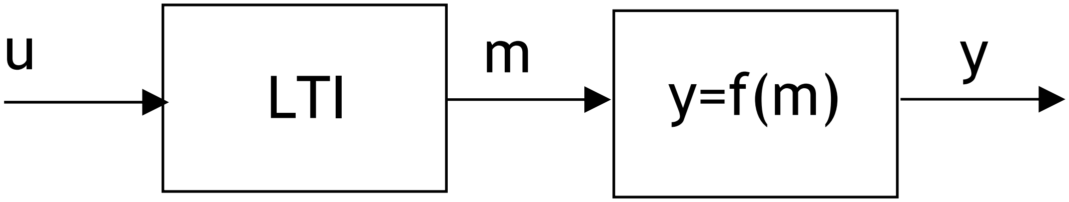
(Some) extensions of SMI algorithms

- 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.

Array signal processing

Array signal processing

Problem:

extracting information from signals received simultaneously by an array of sensors.

Typical applications:

- radar data analysis;
- underwater acoustics;
- geophysics;

Information of interest: the so-called Direction Of Arrival (DOA) of waves reaching the array.

Signal model for DOA problems

Consider an array of m sensors, on which n incident waves arrive.

At a given frequency, a snapshot of the received signals can be written as

$$x = As + n$$

where

- $x \in \mathbb{C}^m$ is the received signal;
- $s \in \mathbb{C}^n$ contains the amplitudes of the sources;
- $A \in \mathbb{C}^{(m,n)}$
- $n \in \mathbb{C}^m$ is a noise vector.

Signal model for DOA problems (cont.d)

A can be written as

$$A = \begin{bmatrix} a(\theta_1) & a(\theta_2) & \dots & a(\theta_n) \end{bmatrix}$$

where

- θ_i is the bearing of source i with respect to the array normal;
- $a(\theta_i)$ is the steering vector of source i;

Common assumptions:

- s and n are stationary random variables;
- n are zero mean and uncorrelated with s;

Signal model for DOA problems (cont.d)

Under the cited assumptions one can write

$$R = E[xx^H] = ASA^H + R_n$$

where $S=E[s\ s^H]$ and $R_n=E[n\ n^H]$.

In a practical situation only an estimate of R will be available:

$$\hat{R} = \frac{1}{K}\sum_{k=1}^{K} x(k)x(k)^H$$

Typical approach to the DOA problem

Most techniques proceeds in two steps:

- Approximate R with a low rank matrix ASA^H from which the steering matrix can be recovered;
- Use the computed A to determine the actual directions of arrival.

We will focus on the first point.

NOTE that since A changes with time, *tracking* of the low rank approximation of A is the actual problem.

Basic idea

In the noise free case

$$R = ASA^H$$

therefore, taking the SVD of R

$$R = ASA^H = \begin{bmatrix} U_n & U_n^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_n & V_n^{\perp} \end{bmatrix}^H = U_n \Sigma_n V_n^H$$

so U_n is an orthonormal basis for span(A).

In the presence of noise

$$R = ASA^H + R_n = \begin{bmatrix} \hat{U}_n & \hat{U}_n^{\perp} \end{bmatrix} \begin{bmatrix} \hat{\Sigma}_n & 0 \\ 0 & \hat{\Sigma}_n^{\perp} \end{bmatrix} \begin{bmatrix} \hat{V}_n & \hat{V}_n^{\perp} \end{bmatrix}^H$$

and in this case U_n is the best rank n approximation of span(A).

Recursive update of the estimate

- At each time step the estimate of R is updated as

$$\hat{R}(t) = \beta(t)\hat{R}(t-1) + \alpha(t)x(t)x(t)^H$$

- Recomputing the SVD would be uselessly burdensome $\rightarrow$ algorithms for recursive update;

- What do we need to update?
  - Only $\mathrm{U_n}$, if n is constant and known;
  - U and the singular values if the number of sources is unknown and/or time-varying.

Yang’s tracking method

Yang’s criterion: given a random vector x it is possible to show that the criterion, W $\in \mathbb{C}^{m,n}$

$$V(W) = E\|x - WW^T x\|^2$$

has a unique global minimum for unitary *W* given by the dominant eigenvectors of R=E[x x^H].

The problem of estimating the dominant left singular vector is equivalent to an unconstrained optimisation problem!

Gradient tracking algorithm

Note that Yang's criterion can be written as

$$V(W) = E\|x - WW^T x\|^2 = \text{trace}(R) - 2\text{trace}(W^H RW) + \\ + \text{trace}(W^H RWW^H W)$$

and the gradient with respect to W is therefore

$$\nabla V(W) = [-2R + RWW^H + WW^H R]W$$

so an update rule for the subpace estimate can be given by

$$W(t) = W(t-1) - \mu[-2\hat{R}(t) + \hat{R}(t)W(t-1)W(t-1)^H + \\ + W(t-1)W(t-1)^H \hat{R}(t)]W(t-1)$$

Gradient tracking algorithm (cont.d)

where:

- μ is the gain of the gradient iteration;
- The estimate of R can be defined as previously, or simply as

$$\widehat{R}(t) = x(t)x(t)^H$$

as in the Least Mean Squares algorithm for adaptive filtering.

Further simplification: let $W(t-1)^H W(t-1) \simeq I$.

Projection approximation subspace tracking

Recall the form of Yang's criterion

$$V(W) = E\|x - WW^H x\|^2$$

and approximate it with

$$V(W(t)) = \sum_{i=1}^{t} \beta^{t-i}\|x(i) - W(t)W^H(t)x(i)\|^2$$

where $0 < \beta \leq 1$ is a forgetting factor.

Introduce now the *projection approximation*:

$$W^H(t)x(i) \simeq W^H(t-1)x(i) = y(i)$$

Projection approximation subspace tracking

(cont.d)

Using the approximation, the criterion becomes

$$V'(W(t)) = \sum_{i=1}^{t} \beta^{t-i} \|x(i) - W(t)y(i)\|^2$$

which is now quadratic in W(t) and can be therefore optimised using the conventional Recursive Least Squares (RLS) algorithm:

$$W(t) = R_{xy}(t) R_{yy}^{-1}(t)$$

where

$$R_{xy}(t) = \beta R_{xy}(t-1) + x(t)y^H(t)$$

$$R_y(t) = \beta R_y(t-1) + y(t)y^H(t)$$

Recursive state space identification algorithms

I/O vs state space recursive methods

Recursive subspace identification algorithms rely on the following ideas:

- The underlying algebraic tools (QR and SVD) used in computing span(Γ_i) can be made recursive;
- The estimates of A and C can be therefore updated simply;
- With A and C known, the estimation of B and D is a linear problem → RLS algorithm.

Recursive subspace identification (1)

First step: update of the RQ factorisation.

In the orthogonal projection case:

$$\begin{bmatrix} U(t+1) \\ Y(t+1) \end{bmatrix} = \begin{bmatrix} R_{11}(t) & 0 & z_u \\ R_{21}(t) & R_{22}(t) & z_y \end{bmatrix} \begin{bmatrix} Q_1(t) & 0 \\ Q_2(t) & 0 \\ 0 & 1 \end{bmatrix}$$

z_u zeroed by sequence of Givens rotations:

$$\begin{bmatrix} R_{11}(t+1) & 0 \quad\quad 0 \\ R_{21}(t+1) & \boxed{R_{22}(t) \quad z} \end{bmatrix} = \begin{bmatrix} R_{11}(t+1) & 0 \\ R_{21}(t+1) & \boxed{R_{22}(t+1)} \end{bmatrix}$$

$$\Rightarrow \quad R_{22}(t+1)R_{22}(t+1)^T = R_{22}(t)R_{22}(t)^T + zz^T$$

Recursive subspace identification (2)

Second step: use z to update the subspace estimate.

Most used algorithms for SVD update based on optimisation of suitable criteria:

- Yang's criterion (Yang 1995):
  - Unbiased update in output error case (w=f=0);
- Gustafsson's criterion (Gustafsson 1997):
  - Unbiased update also with $w \neq 0$ and white f using instrumental variables.

Recursive subspace identification (3)

Yang's criterion: given a random vector z it is possible to show that the criterion

$$V(W) = E\|z - WW^T z\|^2$$

has a unique global minimum for *W* given by the dominant eigenvectors of R_z=E[z z^T].

$\Rightarrow$ PAST algorithm for recursive update of observability subspace

Recursive subspace identification (4)

Gustafsson’s criterion: given a random vector x it is possible to show that the criterion

$$V_{IV}(W(t)) = \left\| R_{x\xi}(t) - W(t)W^T(t)R_{x\xi}(t) \right\|_F^2$$

has a unique global minimum for *W* given by the dominant eigenvectors of $R_{x\xi}=E[x\xi^T]$.

⇒ IV-PAST algorithm for recursive update of observability subspace.

Recursive subspace identification (5)

Updating the estimates of B and D:

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

$$= \begin{bmatrix} \phi_1(t)^T & \phi_2(t)^T \end{bmatrix} \begin{bmatrix} vec(D) \\ vec(B) \end{bmatrix}$$

ϕ_1(t) depends only on u(t); as for ϕ_2, note that

$$\phi_2(t+1)^T = \sum_{r=0}^{t} u(r)^T \otimes \mathbf{C}\mathbf{A}^{t-r}) =$$

$$= \phi_2(t)^T (I_m \otimes A) + (u(t) \otimes C)$$

so the regressors (and the estimates) can be updated recursively.

Simulation examples

Consider the linear system

$$x(t+1) = \begin{bmatrix} \lambda_1(t) & 0 \\ 0 & \lambda_2(t) \end{bmatrix} x(t) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 2 \end{bmatrix} x(t) + 0.05u(t) + v(t)$$

where

$$\lambda_1(t) = \begin{cases} 0.7 & t < 1000 \\ 0.7 + 0.1(e^{-(t-1000)/(4000-1000)} - 1)/(e^{-1} - 1) \end{cases}$$

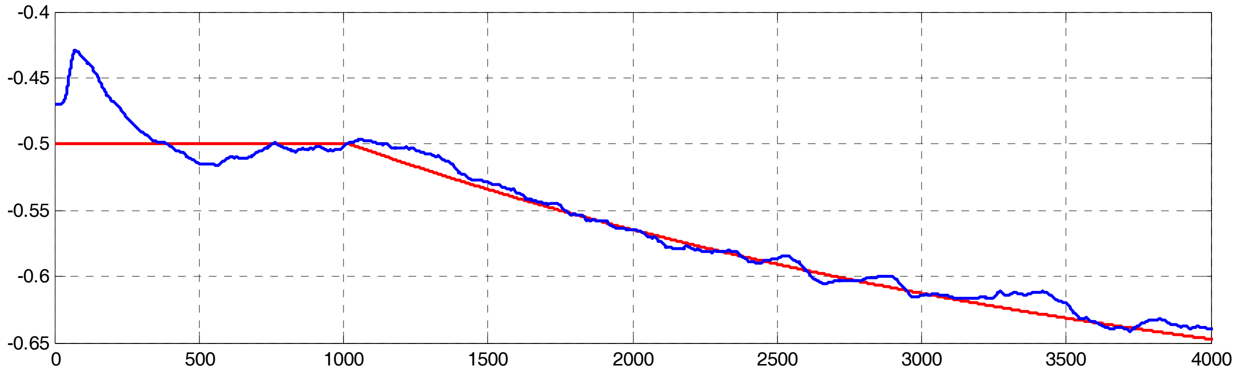
$$\lambda_2(t) = \begin{cases} -0.5 & t < 1000 \\ -0.5 - 0.15(e^{-(t-1000)/(4000-1000)} - 1)/(e^{-1} - 1) \end{cases}$$

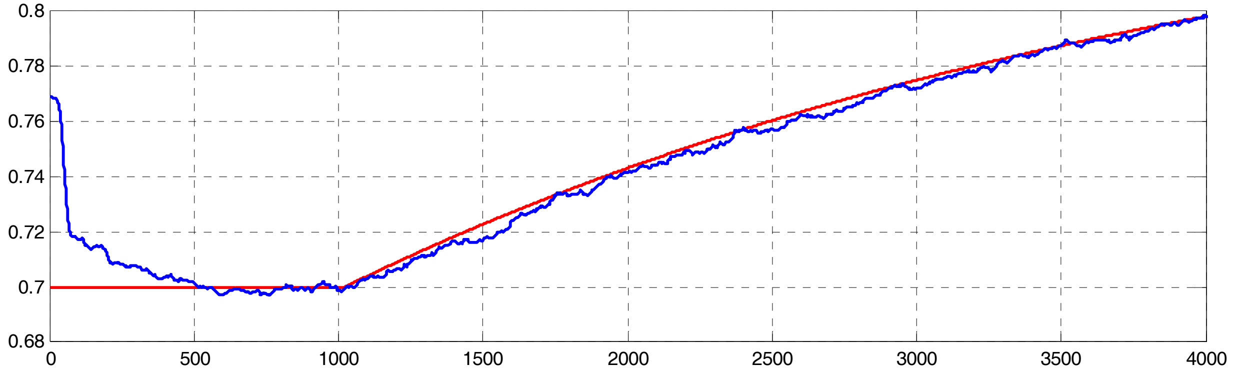
u(t) is a PRBS sequence limited between -1 and 1;

v(t) is a low pass filtered WGN with σ=0.1;

Simulation examples (2)

Estimated eigenvalues:





Simulation examples (3)

Consider now the system

$$x(t+1) = \begin{bmatrix} \lambda_1(t) & 0 \\ 0 & -0.5 \end{bmatrix} x(t) + \begin{bmatrix} 2 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 2 \end{bmatrix} x(t) + 0.05u(t) + v(t)$$

where

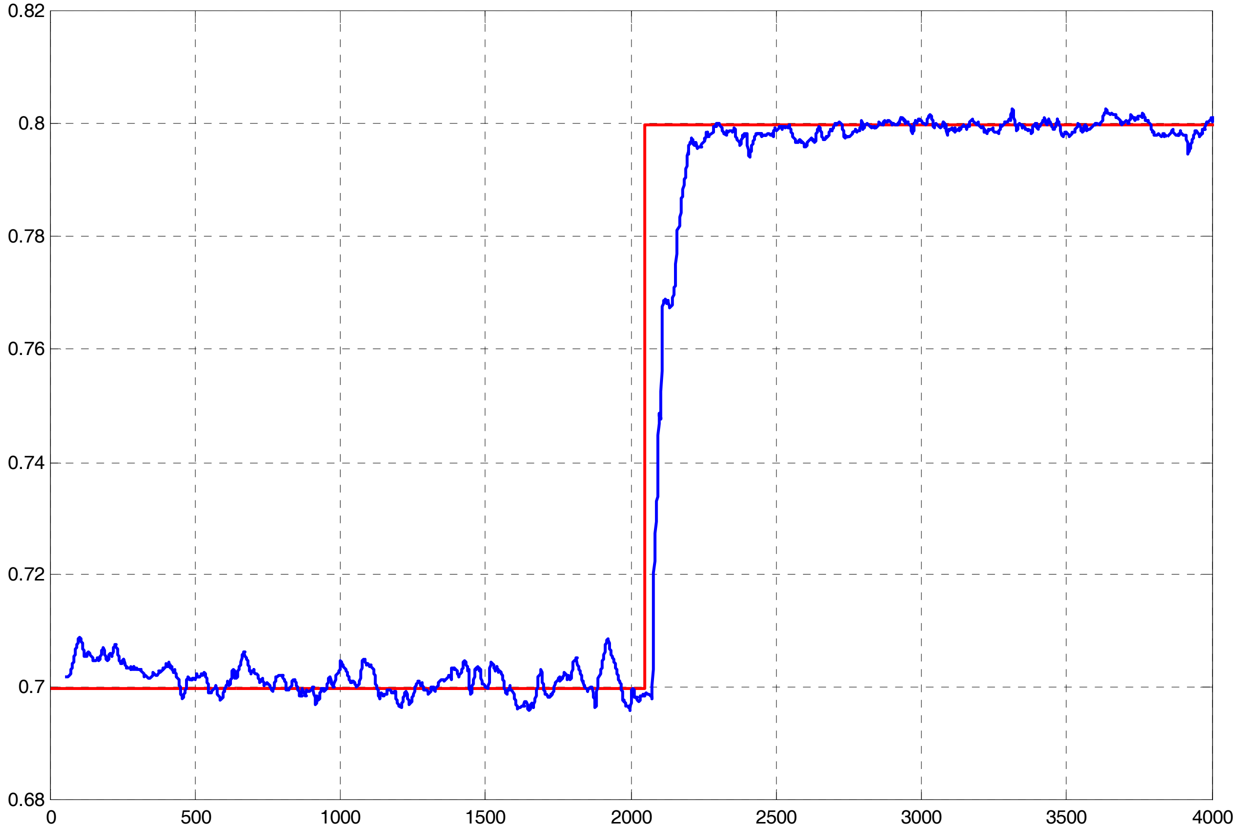
$$\lambda_1(t) = \begin{cases} 0.7 & t \leq 2000 \\ 0.8 & t > 2000 \end{cases}$$

u(t) is a PRBS sequence limited between -1 and 1;

v(t) is a low pass filtered WGN with σ=0.1;

Simulation examples (4)

Estimated λ_1 eigenvalue:



Conclusions

Considered problems:

- I/O system identification (LS problems);
- State space identification (realisation and subspace methods);
- Array signal processing;
- Recursive system identification;

All successfully solved combining QR and SVD factorisations!