System Identification

Lecture 4: Parameter Estimation Methods 1

Sahar Moghimi
We have a set of models (based on the parameter vector)

\[ M^* = \{ M(\theta) | \theta \in D_M \} \]

...and a set of measurements
- Prediction error minimization
- State space
- Probabilistic approach
Prediction error identification methods

- **Approach:** Choose parameters to make the PE as small as possible

\[ \varepsilon(t, \theta^*) = y(t) - \hat{y}(t|\theta^*) \]

- In order to define a scalar measure:

\[ \varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta), \quad 1 \leq t \leq N \quad (7.10) \]

Then use the following norm:

\[ V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} \ell(\varepsilon_F(t, \theta)) \quad (7.11) \]

where \( \ell(\cdot) \) is a scalar-valued (typically linear filter) function. The function \( V_N(\theta, Z^N) \) is, for given time-varying model, a measure of the model parameter \( \theta \). It is a natural norm for the estimate \( \hat{\theta}_N \) is then defined by minimizing the norm:

\[ \hat{\theta}_N = \hat{\theta}_N(Z^N) \]

**Linear filter:** If set properly can reduce the effect of noise.
Least squares method

- Special case of prediction error methods:

\[
\varepsilon(t, \theta) = y(t) - \varphi^T(t)\theta
\]

\[
V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} \frac{1}{2} [y(t) - \varphi^T(t)\theta]^2
\]  \hspace{1cm} (7.33)

\[
\hat{\theta}_{LS}^N = \arg \min_{\theta} V_N(\theta, Z^N) = \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^{N} \varphi(t)y(t)
\]  \hspace{1cm} (7.34)
On the consistency of LSE

\begin{equation}
    y(t) = \varphi^T(t)\theta_0 + v_0(t)
\end{equation}

\begin{equation}
    \lim_{N \to \infty} \hat{\theta}_N^\text{LS} - \theta_0 = \lim_{N \to \infty} R^{-1}(N) \frac{1}{N} \sum_{t=1}^{N} \varphi(t)v_0(t) = (R^*)^{-1}f^*.
\end{equation}

\begin{equation}
    R^* = E\varphi(t)\varphi^T(t), \quad f^* = E\varphi(t)v_0(t)
\end{equation}

provided \( v_0 \) and \( \varphi \) are quasi-stationary, so that Theorem 2.3 can be applied. For the LSE to be consistent, that is, for \( \hat{\theta}_N^\text{LS} \) to converge to \( \theta_0 \), we thus have to require:

i. \( R^* \) is non-singular. This will be secured by the input properties, as in (1.17)–(1.18), and discussed in much more detail in Chapter 13.

ii. \( f^* = 0 \). This will be the case if either:

(a) \( \{v_0(t)\} \) is a sequence of independent random variables with zero mean values (white noise). Then \( v_0(t) \) will not depend on what happened up to time \( t - 1 \) and hence \( E\varphi(t)v_0(t) = 0 \).

(b) The input sequence \( \{u(t)\} \) is independent of the zero mean sequence \( \{v_0(t)\} \) and \( n_d = 0 \) in (7.32). Then \( \varphi(t) \) contains only \( u \)-terms and hence \( E\varphi(t)v_0(t) = 0 \).
Weighted LES: to value different measurements in a different fashion:

\[ V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^{N} \alpha_t \left[ y(t) - \varphi^T(t)\theta \right]^2 \]  \hspace{1cm} (7.39)

or

\[ V_N(\theta, Z^N) = \sum_{t=1}^{N} \beta(N, t) \left[ y(t) - \varphi^T(t)\theta \right]^2 \]  \hspace{1cm} (7.40)

The expression for the resulting estimate is quite analogous to (7.34):

\[ \hat{\theta}_N^{LS} = \left[ \sum_{t=1}^{N} \beta(N, t)\varphi(t)\varphi^T(t) \right]^{-1} \sum_{t=1}^{N} \beta(N, t)\varphi(t)y(t) \]  \hspace{1cm} (7.41)
Dealing with the issue of best regressors

- **Weighted Principle Component Regression (PCR)**
  - Consider

\[
Y = XA + E
\]

where \( Y = \begin{bmatrix} y(t) \\ y(t+1) \\ \vdots \\ y(t+N) \end{bmatrix} \), \( X = \begin{bmatrix} x(t) & x(t-1) & \ldots & x(t-p) \\ x(t+1) & x(t) & \ldots & x(t-p+1) \\ \vdots & \vdots & \ddots & \vdots \\ x(t+N) & x(t-1+N) & \ldots & x(t-p+N) \end{bmatrix} \), \( A = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} \)

- The number of columns (p+1) in X should be sufficiently large to cover the whole duration of the impulse response function which can approximate the system relatively accurately.

- If p+1 is greater than the number of nonzero samples in the true impulse response, then the true value of the corresponding extraneous elements in A should be zero.
We consider decomposing $X$ into its principal components and solving the linear regression problem in the domain of PCs using Singular Value Decomposition (SVD).

$$X = U \cdot D \cdot V^T$$

- Treat as black box: code widely available
- In MATLAB: $[U, D, V] = \text{svd}(X, 0)$
Dealing with the issue of best regressors

- The $d_i$ are called the singular values of $X$.
- If $D$ is singular, some of the $d_i$ will be 0.
- In general $\text{rank}(X) = $ number of nonzero $d_i$.

- The column vectors of $U$ (principal components) are the eigenvectors of the matrix $XX^T$.
- The column vectors of $V$ are the eigenvectors of the matrix $X^TX$, and
- $D$ is a diagonal matrix with its diagonal elements (singular values) being the square root of the eigenvalues of $X^TX$ (or $XX^T$).
Dealing with the issue of best regressors

- Why is SVD so useful?
- Application #1: inverses
  \[ A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = V W^{-1} U^T \]
  - Using fact that inverse = transpose for orthogonal matrices
  - Since \( W \) is diagonal, \( W^{-1} \) also diagonal with reciprocals of entries of \( W \)
Dealing with the issue of best regressors

- $A^{-1} = (V^T)^{-1} W^{-1} U^{-1} = V W^{-1} U^T$
- This fails when some $w_i$ are 0
  - It’s supposed to fail – singular matrix
- Pseudoinverse: if $w_i = 0$, set $1/w_i$ to 0 (!)
  - “Closest” matrix to inverse
  - Defined for all (even non-square, singular, etc.) matrices
  - Equal to $(A^T A)^{-1} A^T$ if $A^T A$ invertible
Dealing with the issue of best regressors

- Solving $Ax=b$ by least squares
- $x=\text{pseudoinverse}(A)$ times $b$
- Compute pseudoinverse using SVD
  - Lets you see if data is singular
  - Even if not singular, ratio of max to min singular values (condition number) tells you how stable the solution will be
  - Set $1/w_i$ to 0 if $w_i$ is small (even if not exactly 0)
Dealing with the issue of best regressors

\[ Y = UDV^T A + E = UB + E \]

where \( B \triangleq DV^T A \).

- If some of the diagonal elements of \( D \) are small, it means that the corresponding PCs have small variances and consequently less information.
- We rank the PCs in a descending order according to their singular values in evaluating their contributions to the output.
- We form a series of matrices \( U \) each containing a subset of PCs by adding one PC at a time as a column vector.
- The matrices \( U \) differ by the number of columns involved and their associated regression equations \( (Y = UB + E) \) represent the candidate models from which the "best" model should be selected.
- Note that these candidate models are data-specific.
- For model selection, we employ the widely used model order selection criteria.
Dealing with the issue of best regressors

- With \( q \) PCs and the corresponding \( U, V \) and \( D \) matrices denoted as \( \hat{U} (N \times q) \), \( \hat{V} (p \times q) \) and \( \hat{D} (q \times q) \)

The least squares solution of \( B \) is:

\[
\hat{B} = (\hat{U}^T \hat{U})^{-1} \hat{U}^T Y = \hat{U}^T Y
\]

Therefore:

\[
\hat{A} = \hat{V} \hat{D}^{-1} \hat{B}
\]
Estimating State space models: Subspace method

- Estimating the state space model using LSE

\[ x(t + 1) = Ax(t) + Bu(t) \]
\[ y(t) = Cx(t) + Du(t) + v(t) \]

- If we do not have an insight into the particular structure, for different state vectors, we can have an indefinite number of solutions
Estimating State space models: Subspace method

- If we knew the sequence of state vectors:

\[ Y(t) = \begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix}, \quad \Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \]

\[ \Phi(t) = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \quad E(t) = \begin{bmatrix} w(t) \\ v(t) \end{bmatrix} \]

\[ Y(t) = \Theta \Phi(t) + E(t) \quad (7.56) \]

- Therefore the state and output can be estimated using LS.
Estimating State space models: Subspace method

- If we define

\[
\hat{Y}_r(t) = \begin{bmatrix}
\hat{y}(t|t-1) \\
\vdots \\
\hat{y}(t+r-1|t-1)
\end{bmatrix} \quad (7.59a)
\]

\[
\hat{Y} = \begin{bmatrix}
\hat{Y}_r(1) & \ldots & \hat{Y}_r(N)
\end{bmatrix} \quad (7.59b)
\]

Then the following is true as \( N \to \infty \) (see Lemmas 4A.1 and 4A.2 and their proofs):

1. The system (7.57) has an \( n \)th order minimal state space description if and only if the rank \( \hat{Y} \) is equal to \( n \) for all \( r \geq n \).

2. The state vector of any minimal realization in innovations form can be chosen as linear combinations of \( \hat{Y}_r \) that form a row basis for \( \hat{Y} \), i.e.,

\[
x(t) = L\hat{Y}_r(t) \quad (7.60)
\]

where the \( n \times pr \) matrix \( L \) is such that \( L\hat{Y} \) spans \( \hat{Y} \). \( (p \) is the dimension of the output vector \( y(t) \).)
Estimating State space models: Subspace method

- K step ahead prediction based on a finite number of measurements:

\[
\hat{y}(t + k - 1|t - 1) = \alpha_1 y(t - 1) + \ldots + \alpha_{s_1} y(t - s_1) \\
+ \beta_1 u(t - 1) + \ldots + \beta_{s_2} u(t - s_2)
\]  

(7.61)

- Algorithm steps:

1. Choose \( s_1, s_2, r \) and \( \ell \) and form \( \hat{Y}_r(t) \) in (7.65) and \( Y \) as in (7.59).
2. Estimate the rank \( n \) of \( Y \) and determine \( L \) in (7.60) so that \( x(t) \) corresponds to a well-conditioned basis for it.
3. Estimate $A$, $B$, $C$, $D$ and the noise covariance matrices by applying the LS method to the linear regression (7.56).

$$Y(t) = \Theta \Phi(t) + E(t)$$

(7.56)

\[
Y(t) = \begin{bmatrix} x(t + 1) \\ y(t) \end{bmatrix}, \quad \Theta = \begin{bmatrix} A & B \\ C & D \end{bmatrix}
\]

\[
\Phi(t) = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \quad E(t) = \begin{bmatrix} w(t) \\ v(t) \end{bmatrix}
\]
Instrumental variable method

Consider the ARX model,

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \varepsilon(t)$$

or, equivalently,

$$y(t) = \varphi^T(t)\theta + \varepsilon(t)$$

where $\varepsilon(t)$ is the equation error $(y(t) - y_m(t))$, and

$$\varphi(t) = [-y(t-1) \ldots -y(t-n_a) \ u(t-1) \ldots u(t-n_b)]^T$$

$$\theta = [a_1 \ldots a_{n_a} \ b_1 \ldots b_{n_b}]^T$$
The least squares estimate

$$\hat{\theta} = \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi(t)y(t) \right]$$

has the estimation error (when $N \to \infty$)

$$\hat{\theta} - \theta_0 = E \left[ \varphi(t)\varphi^T(t) \right]^{-1} E \left[ \varphi(t)\varepsilon(t) \right]$$

Consequently, for $\hat{\theta} - \theta_0 = 0$ to hold, we must have

$$E \left[ \varphi(t)\varepsilon(t) \right] = 0,$$

which is satisfied if, and essentially only if, $\varepsilon(t)$ is white noise. Hence, the least squares estimate is not consistent for correlated noise sources!
Introduce a vector $\mathbf{z}(t) \in \mathbb{R}^{n_{\theta}}$ with entries uncorrelated with $\varepsilon(t)$. Then (for large values of $N$)

$$0 = \frac{1}{N} \sum_{t=1}^{N} \mathbf{z}(t)\varepsilon(t) = \frac{1}{N} \sum_{t=1}^{N} \mathbf{z}(t) \left[ y(t) - \varphi^{T}(t)\theta \right]$$

which yields (if the inverse exists)

$$\hat{\theta} = \left[ \frac{1}{N} \sum_{t=1}^{N} \mathbf{z}(t)\varphi^{T}(t) \right]^{-1} \left[ \frac{1}{N} \sum_{t=1}^{N} \mathbf{z}(t)y(t) \right]$$

The elements of $\mathbf{z}(t)$ are usually called the instruments. Note that if $\mathbf{z}(t) = \varphi(t)$, the IV estimate reduces to the LS estimate.
Obviously, the choice of instruments is very important. They have to be chosen

(i) such that $z(t)$ is uncorrelated with $\varepsilon(t)$ ($Ez(t)\varepsilon(t) = 0$), and

(ii) such that the matrix

$$\frac{1}{N} \sum_{t=1}^{N} z(t)\varphi^T(t) \rightarrow Ez(t)\varphi^T(t)$$

has full rank. In other words it is essential that $z(t)$ and $\varphi(t)$ are correlated.
In practice these demands are fulfilled by choosing the instruments to consist of delayed and/or filtered inputs. The instruments are commonly chosen such that

\[
\mathbf{z}(t) = \begin{bmatrix}
-\eta(t-1) & \ldots & -\eta(t-n_a) & u(t-1) & \ldots & u(t-n_b)
\end{bmatrix}^T
\]

where the signal \( \eta(t) \) is obtained by filtering the input as

\[
C(q^{-1})\eta(t) = D(q^{-1})u(t).
\]

In the special case when \( C(q^{-1}) = 1 \) and \( D(q^{-1}) = -q^{-n_b} \),

\[
\mathbf{z}(t) = \begin{bmatrix}
u(t-1) & \ldots & u(t-n_a-n_b)
\end{bmatrix}^T
\]

**Rem:** Notice that \( u(t) \) and the noise \( \varepsilon(t) \) are assumed to be independent.
Recall that the basic IV estimate is derived from

$$\min_\theta \left\| \sum_{t=1}^{N} z(t)\varepsilon(t) \right\|^2$$

More flexibility is obtained if the instrument vector $z(t)$ is augmented to dimension $n_z$ ($n_z \geq n_\theta$), and if we allow for a weighting and a prefiltering of the residuals by some stable filter $F(q^{-1})$, i.e.,

$$\min_\theta \left\| \sum_{t=1}^{N} z(t)F(q^{-1})\varepsilon(t) \right\|^2_Q$$

where $\|x\|_Q^2 = x^TQx$ and $Q$ is a positive definite weighting matrix.
Inserting
\[ \varepsilon(t) = y(t) - \varphi^T(t)\theta \]
yields the so-called extended IV method

\[ \hat{\theta} = \arg \min_\theta \left\| \left[ \sum_{t=1}^{N} z(t)F(q^{-1})\varphi^T(t) \right] \theta - \left[ \sum_{t=1}^{N} z(t)F(q^{-1})y(t) \right] \right\|_Q^2 \]

When \( F(q^{-1}) \equiv 1 \) and \( Q = I \), the basic IV method is obtained.
Introduce

\[ R_N = \frac{1}{N} \sum_{t=1}^{N} z(t)F(q^{-1})\varphi^T(t) \]
\[ r_N = \frac{1}{N} \sum_{t=1}^{N} z(t)F(q^{-1})y(t) \]
Then

\[ \hat{\theta} = \arg \min_{\theta} \| R_N \theta - r_N \|_Q^2 \]

\[ = \arg \min_{\theta} (R_N \theta - r_N)^T Q (R_N \theta - r_N) \]

\[ = \left[ R_N^T Q R_N \right]^{-1} R_N^T Q r_N \]

Note that due to numerical instability the algorithm should not be implemented in this manner.

**Rem:** Notice that $R_N$ is in general not a square matrix.
Example

The true system is given by

\[(1 - 1.5q^{-1} + 0.7q^{-2})y(t) = (1.0q^{-1} + 0.5q^{-2})u(t) + (1 - 1.0q^{-1} + 0.2q^{-2})e(t)\]

- **ARMAX model**
- \(u(t)\) is from an ARMA process, independent of \(e(t)\)
- \(e(t)\) is white noise with zero mean and variance 1
- \(N = 250\) (number of data points)

**estimation**

- use ARX model and assume \(n_a = 2, n_b = 2\)
- compare the LS method with IVM
```matlab
%% Generate the data
close all; clear all;
N = 250; Ts = 1;
a = [1 -1.5 0.7]; b = [0 1.5]; c = [1 -1 0.2];
Au = [1 -0.1 -0.12]; Bu = [0 1 0.2]; Mu = idpoly(Au,Bu,Ts);
u = sim(Mu,randn(2*N,1)); % u is ARMA process
noise_var = 1; e = randn(2*N,1);
M = idpoly(a,b,c,1,1,noise_var,Ts);
y = sim(M,[u e]);
uv = u(N+1:end); ev = e(N+1:end); yv = y(N+1:end);
u = u(1:N); e = e(1:N); y = y(1:N);
DATe = iddata(y,u,Ts); DATv = iddata(yv,uv,Ts);

%% Identification
na = 2; nb = 2; nc = 2;
theta_iv = iv4(DATe,[na nb 1]); % ARX using iv4
theta_ls = arx(DATe,[na nb 1]); % ARX using LS

%% Compare the measured output and the model output
[yhat2,fit2] = compare(DATv,theta_iv);
yhat4,fit4 = compare(DATv,theta_ls);
figure; t = 1:N;
plot(t,yhat2(1).y(t),'--',t,yhat4(1).y(t),'-',t,yv(t));
legend('model (iv)';'model (LS)';'measured');
title('Comparison on validation data set','FontSize',16);
```

Exercise:

- Implement the LSE and IV methods on your data.
- Analyze the estimation error and discuss your findings