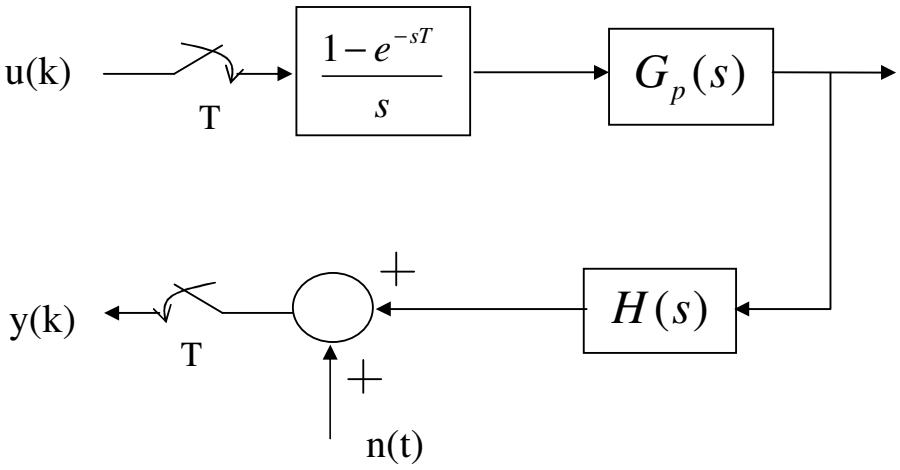
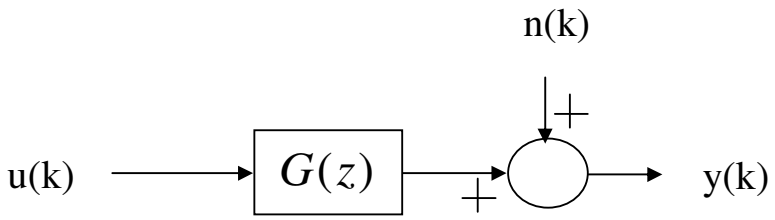
Chapter 8. System Identification

8.1 Introduction

Identify G(z) from experimental results:



Could be represented as:



Could identify G(z):

- A convolution model:

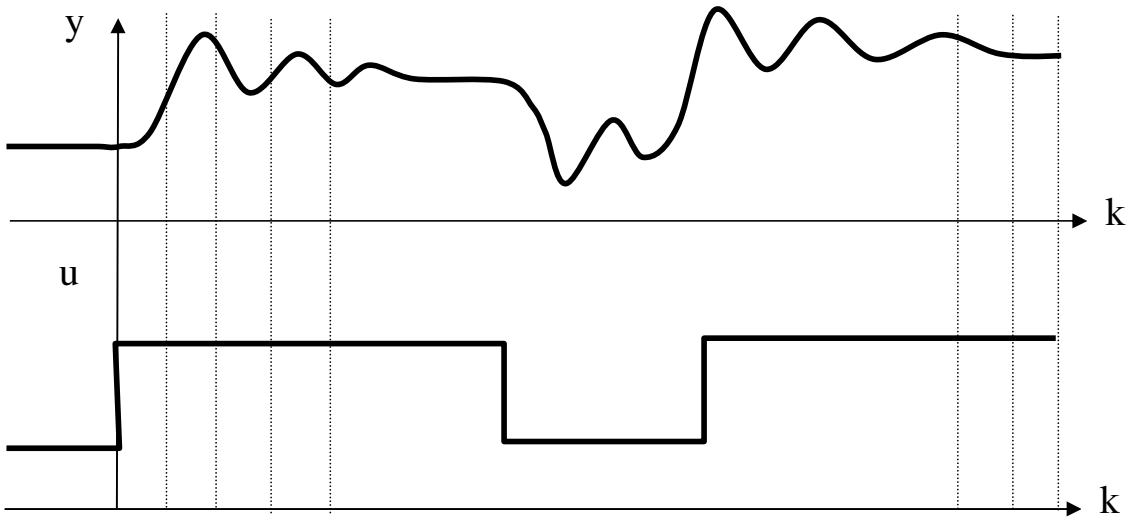
- A transfer function:

8.2 The Least Squares Algorithm

Consider that the SISO process can be represented by the transfer function:

$$\frac{Y(z)}{U(z)} = G(z) = \frac{z^{-d}\left(b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}\right)}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_n z^{-n}}$$

Carry out the experiment – excite the process, collect N_1 data pairs:



Specify the candidate model:

$$\hat{y}(k+1) = \hat{a}_1 y(k) + \hat{a}_2 y(k-1) + \cdots \hat{a}_n y(k-n+1) + \hat{b}_1 u(k-d) + \hat{b}_2 u(k-d-1) + \cdots \hat{b}_m u(k-d-m+1)$$

The first valid "test" equation is then:

$$\hat{y}(m+d) = \hat{a}_1 y(m+d-1) + \hat{a}_2 y(m+d-2) + \cdots \hat{a}_n y(m+d-n) + \hat{b}_1 u(m-1) + \hat{b}_2 u(m-2) + \cdots \hat{b}_m u(0)$$

We can repeat this, to generate output estimates over the valid dataset:

$$\hat{y}(m+d)=\hat{a}_1 y(m+d-1)+\hat{a}_2 y(m+d-2)+\cdots\hat{a}_n y(m+d-n)+\hat{b}_1 u(m-1)+\hat{b}_2 u(m-2)+\cdots\hat{b}_m u(0)$$
$$\hat{y}(m+d+1)=\hat{a}_1 y(m+d)+\hat{a}_2 y(m+d-1)+\cdots\hat{a}_n y(m+d-n+1)+\hat{b}_1 u(m)+\hat{b}_2 u(m-1)+\cdots\hat{b}_m u(1)$$
$$\hat{y}(m+d+2)=\hat{a}_1 y(m+d+1)+\hat{a}_2 y(m+d)+\cdots\hat{a}_n y(m+d-n+2)+\hat{b}_1 u(m+1)+\hat{b}_2 u(m)+\cdots\hat{b}_m u(2)$$
$$\hat{y}(m+d+3)=\hat{a}_1 y(m+d+2)+\hat{a}_2 y(m+d+1)+\cdots\hat{a}_n y(m+d-n+3)+\hat{b}_1 u(m+2)+\hat{b}_2 u(m+1)+\cdots\hat{b}_m u(3)$$
$$\vdots \qquad \vdots$$
$$\vdots \qquad \vdots$$
$$\hat{y}(N_1-1)=\hat{a}_1 y(N_1-2)+\hat{a}_2 y(N_1-3)+\cdots\hat{a}_n y(N_1-1-n)+\hat{b}_1 u(N_1-d-2)+\hat{b}_2 u(N_1-d-3)+\cdots\hat{b}_m u(N_1-d-1-m)$$

Could be written in matrix form as:

$$\begin{bmatrix} \hat{y}(m+d) \\ \hat{y}(m+d+1) \\ \hat{y}(m+d+2) \\ \vdots \\ \hat{y}(N_1-2) \\ \hat{y}(N_1-1) \end{bmatrix} = \begin{bmatrix} y(m+d-1) & \cdots & y(m+d-n) & u(m-1) & \cdots & u(0) \\ y(m+d) & \cdots & y(m+d-n+1) & u(m) & \cdots & u(1) \\ y(m+d+1) & \cdots & y(m+d-n+2) & u(m+1) & \cdots & u(2) \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ y(N_1-3) & \ddots & y(N_1-n-2) & u(N1-d-3) & \cdots & u(N_1-d-m-2) \\ y(N_1-2) & \cdots & y(N_1-n-1) & u(N_1-d-2) & \cdots & u(N_1-d-m-1) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \\ \hat{a}_n \\ \hat{b}_1 \\ \hat{b}_2 \\ \\ \hat{b}_m \end{bmatrix}$$

This could be rewritten as:

$$\underline{\hat{Y}}_N = \Phi\underline{\hat{\theta}}$$

where:

We need a measure of how good our candidate model fits the data-set.

Define the Least-Squares cost function:

$$J = \sum_{i=m+d}^{N_1-1} e^2(i)$$

where:

Now we will define the error vector $\underline{E}$ over the valid data-set:

$$\underline{E} = \begin{bmatrix} e(m+d) \\ e(m+d+1) \\ \vdots \\ e(N_1-1) \end{bmatrix} = \begin{bmatrix} y(m+d) \\ y(m+d+1) \\ \vdots \\ y(N_1-1) \end{bmatrix} - \begin{bmatrix} \hat{y}(m+d) \\ \hat{y}(m+d+1 \\ \vdots \\ \hat{y}(N_1-1) \end{bmatrix}$$

Then we can write the least-squares cost function as:

$$J = \sum_{i=m+d}^{N_1-1} e^2(i) =$$

Could be rewritten as:

$$J = \left(\underline{Y}_N - \underline{\hat{Y}}\right)^T \left(\underline{Y}_N - \underline{\hat{Y}}\right)$$

But we know:

$$\underline{\hat{Y}}_N = \Phi \underline{\hat{\theta}}$$

Revision:

The cost can be expanded as follows:

$$J = \left(\underline{Y}_N{}^T - \underline{\hat{\theta}}^T \Phi^T \right) \left(\underline{Y}_N - \Phi \underline{\hat{\theta}} \right) =$$

Which could be rearranged to yield:

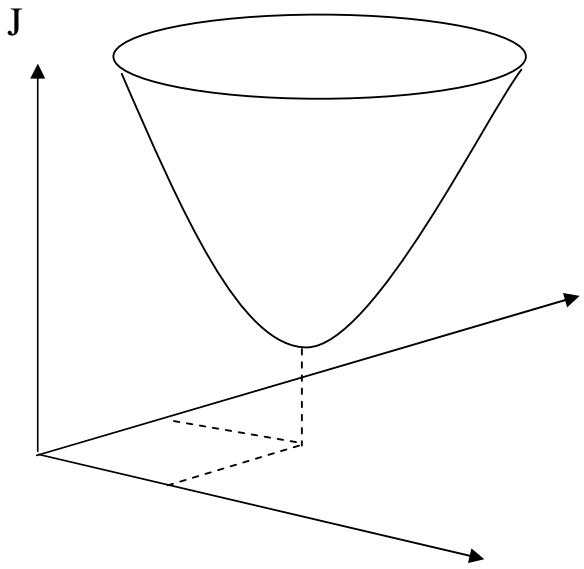
$$J = \underline{\hat{\theta}}^T \Phi^T \Phi \underline{\hat{\theta}} - 2 \underline{Y}_N{}^T \Phi \underline{\hat{\theta}} + \underline{Y}_N{}^T \underline{Y}_N$$

This is a quadratic function of the parameter vector:-

Consider a simple example where we are trying to identify the parameters of the following first order model:

$$G(z) = \frac{b_1 z^{-1}}{1 - a_1 z^{-1}}$$

The least-squares cost function could be plotted for various choices of parameters:



Solution by completing the square:

First consider

$$J = x^2 + 3x + 1$$

Completing the square yields,

The minimum occurs at:

Now consider:

$$J = 5x^2 + 3x + 1$$

Try the following candidate for completing the square:

Multiplying out yields,

Now consider the following cost based on vector $\underline{x}$

$$J = \underline{x}^T M \underline{x} - G\underline{x} + J_0 \qquad (1)$$

Consider the candidate for completing the square:

$$J = (\underline{x} - \underline{\alpha})^T M (\underline{x} - \underline{\alpha}) + \beta \qquad (2)$$

The minimum of this cost function then occurs when:

Multiplying out equation (2) yields:

$$J = \underline{x}^T M \underline{x} - \underline{\alpha}^T M \underline{x} - \underline{x}^T M \underline{\alpha} + \underline{\alpha}^T M \underline{\alpha} + \beta \qquad (3)$$

Comparing equation (3) with the cost equation (1):

$$J = \underline{x}^T M \underline{x} - G\underline{x} + J_0 = \underline{x}^T M \underline{x} - 2\underline{\alpha}^T M \underline{x} + \left(\underline{\alpha}^T M \underline{\alpha} + \beta\right)$$

Hence the cost of equation (1) is a minimum when:

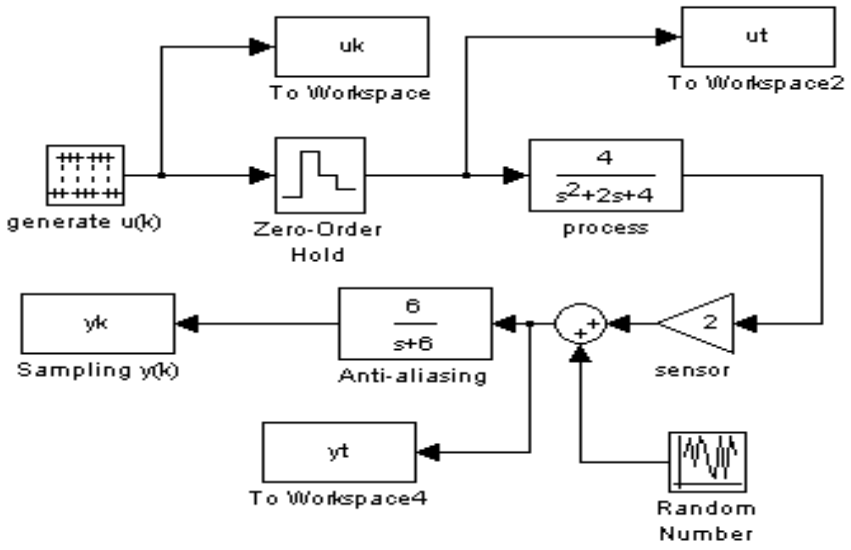
Now we will return to the least squares cost for system identification:

$$J = \underline{\hat{\theta}}^T \Phi^T \Phi \underline{\hat{\theta}} - 2\underline{Y}_N \Phi \underline{\hat{\theta}} + \underline{Y}_N{}^T \underline{Y}_N$$

This will then be minimised when:

$$\underline{\hat{\theta}}_{LS}^T = \tfrac{1}{2} G M^{-1} =$$

EXAMPLE:



The antialisaing filter is chosen as:

The transfer function G(z) is then determined as:

$$G(z) = Z\left\{\frac{1-e^{-sT}}{s}\frac{4}{s^2+2s+4}\,2\,\frac{6}{s+6}\right\}$$

$$= \frac{0.3975z^{-1} + 0.652z^{-2} + 0.0565z^{-3}}{1 - 0.835z^{-1} + 0.407z^{-2} - 0.0183z^{-3}}$$

Hence:

$$\underline{\theta}_{true} = \begin{bmatrix} a_1 & a_2 & a_3 & b_1 & b_2 & b_3 \end{bmatrix}^T =$$
$$= \begin{bmatrix} 0.835 & -0.407 & 0.0183 & 0.3975 & 0.652 & 0.0565 \end{bmatrix}^T$$

The first valid equation is:

$$\hat{y}k(3) = \hat{a}_1 yk(2) + \hat{a}_2 yk(1) + \hat{a}_3 yk(0) + \hat{b}_1 u(2) + \hat{b}_2 u(1) + \hat{b}_3 u(0)$$

The regressor matrix Φ is:

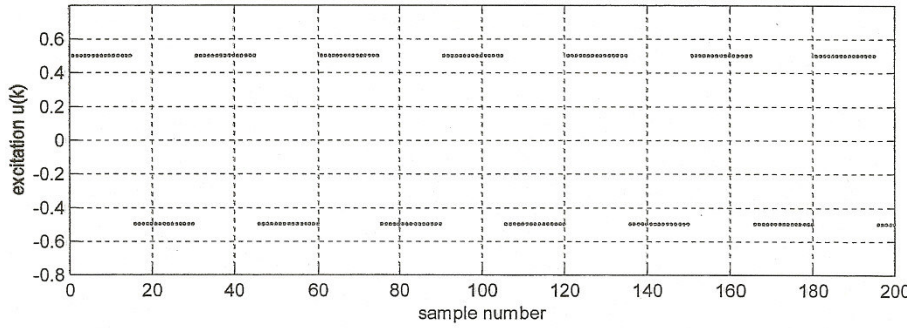
$$\Phi = \left[\begin{array}{ccc|ccc} yk(2) & yk(1) & yk(0) & u(2) & u(1) & u(0) \\ yk(3) & yk(2) & yk(1) & u(3) & u(2) & u(1) \\ yk(4) & yk(3) & yk(2) & u(4) & u(3) & u(2) \\ yk(5) & yk(4) & yk(3) & u(5) & u(4) & u(3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}\right]$$

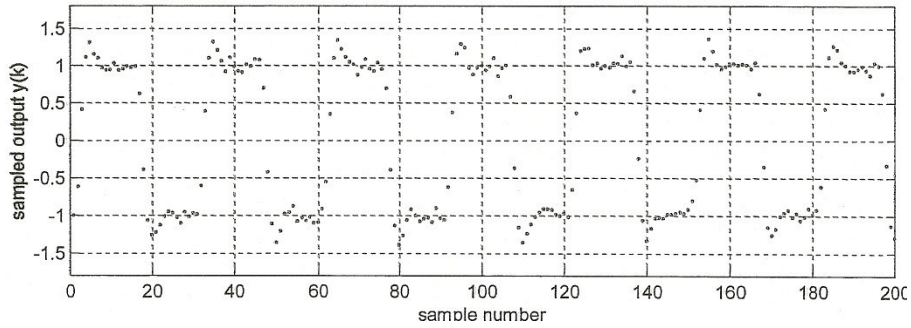
The least squares solution could be implemented as an M file:

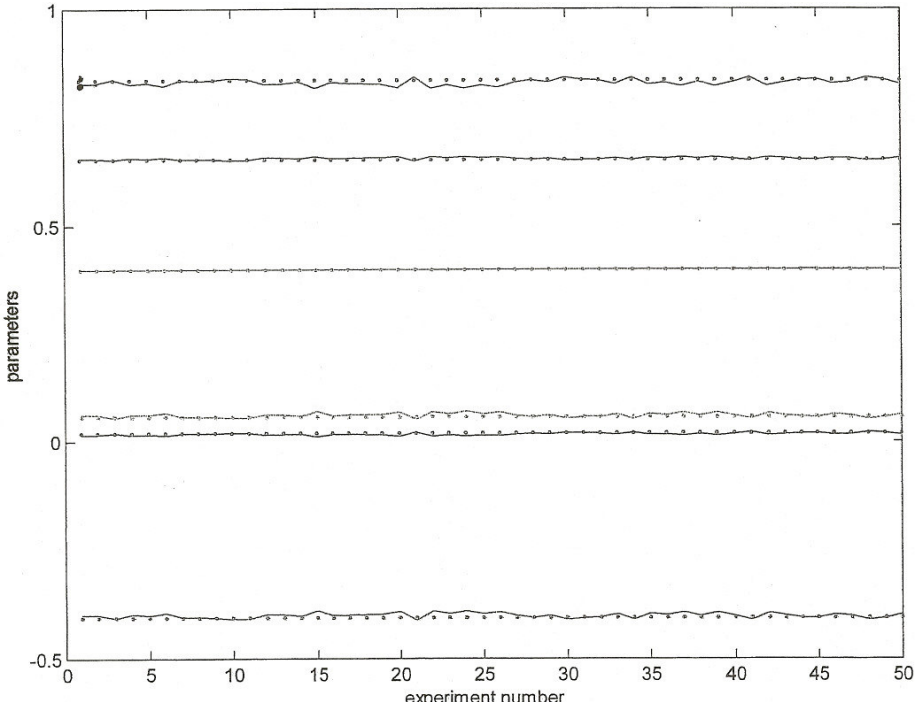
```
n=length(yk);
y=yk-1;
u=uk-0.5;
y3=y(1:n-3);
y2=y(2:n-2);
y1=y(3:n-1);
u3=u(1:n-3);
u2=u(2:n-2);
u1=u(3:n-1);
phi=[y1,y2,y3,u1,u2,u3];
yn=y(4:n);
theta=inv(phi'*phi)*phi'*yn
```

Expt 1: Low standard deviation noise

- 50 different data sets collected each consisting of 200 points
- Parameter vector estimated for each data-set:

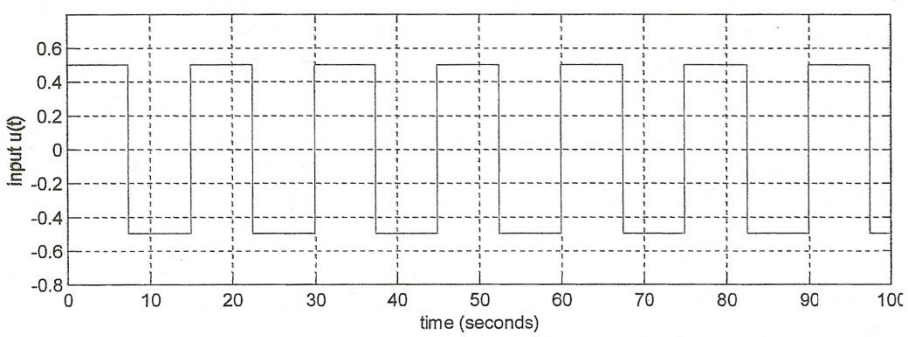


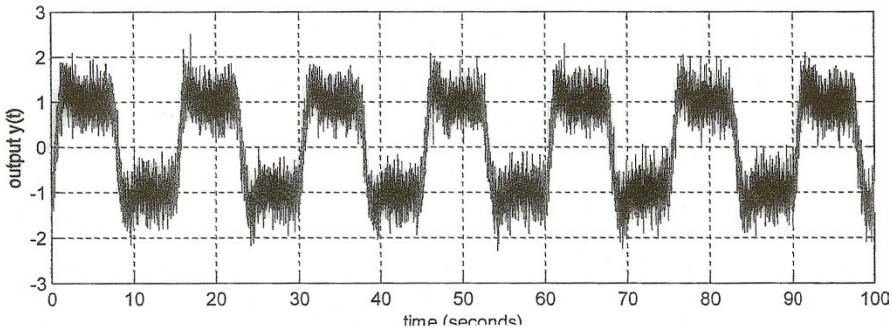


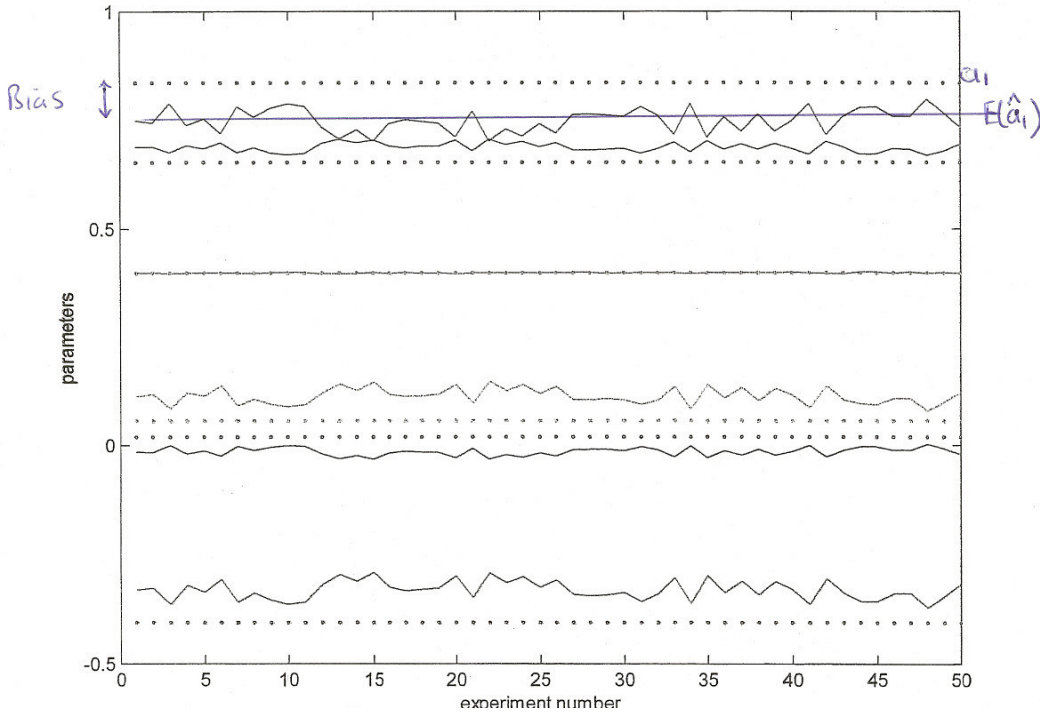


Expt 2: Large standard deviation noise

- 50 different data sets collected each consisting of 200 points
- Parameter vector estimated for each data-set:

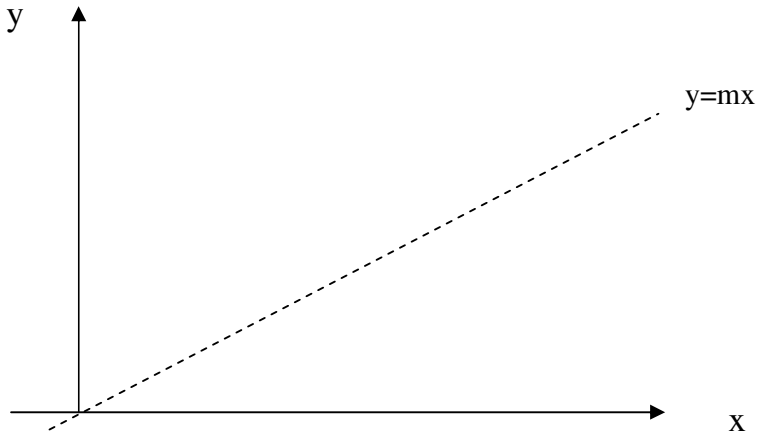






8.2.1 A Note on Bias

Consider the simple line fitting problem:



The true system is y=mx, but the data is corrupted by noise, hence the measurements x and y are related by:

$$y = mx + \varepsilon$$

If the measurement noise is random with zero mean:

Now consider the system identification problem of obtaining unbiased estimates of the A(z) and B(z) polynomials.

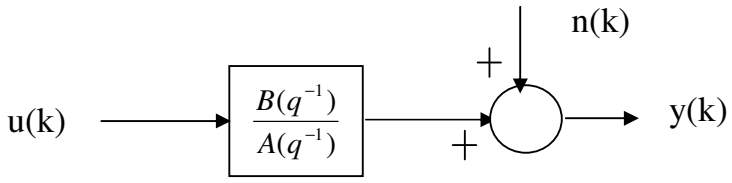
If the measurements are related via the ARX equation:

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

and the residual obeys:

then the parameter estimates will be unbiased.

Note: Consider the effect of sensor measurement noise:



Then:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})}u(k) + n(k)$$

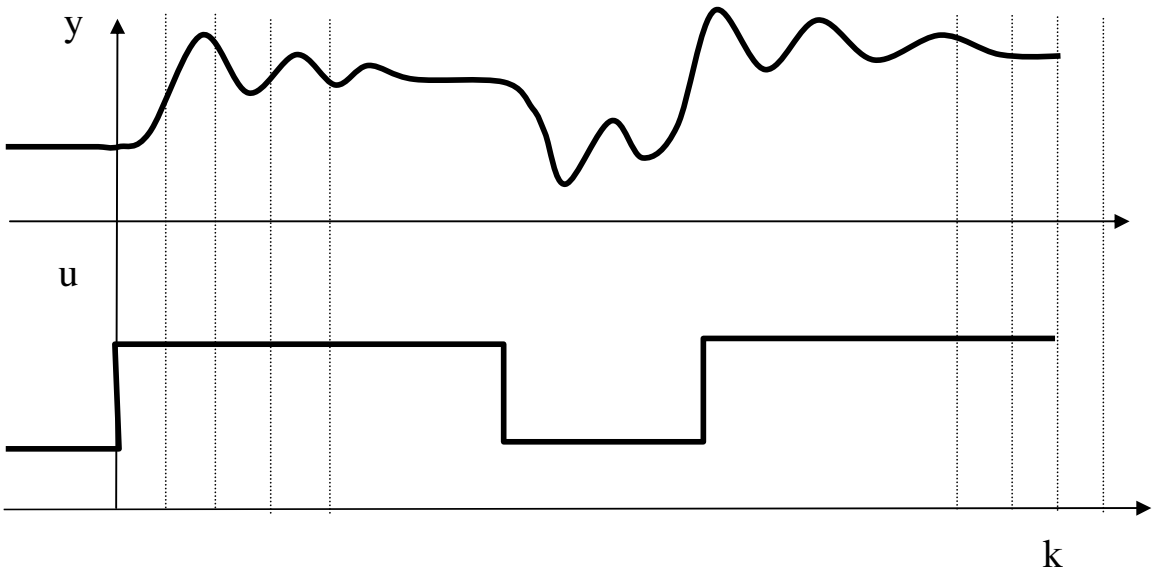
Hence the estimates will be biased:

8.3 Recursive Least Squares

The least squares method is a batch algorithm:

Recursive least Squares (RLS) is an on-line recursive algorithm – which will use each new sample pair {u(k+1), y(k+1)} to generate an updated estimate for the parameter vector.

Consider that data has been collected up to the k^{th} sample:



Define the measurement vector at time k:

$$\underline{Y}(k) = \begin{bmatrix} y(m+d) & y(m+d+1) & \cdots & y(k-1) & y(k) \end{bmatrix}^T$$

Now define the regressor matrix Φ(k) as being formed from all the valid input and output data up to time k.

Hence the estimator equation becomes:

$$\underline{\hat{Y}}(k) = \begin{bmatrix} \hat{y}(m+d) \\ \hat{y}(m+d+1) \\ \vdots \\ \hat{y}(k) \end{bmatrix} = \Phi(k)\underline{\hat{\theta}}$$

And the least squares solution to the parameters at time k is :

$$\underline{\hat{\theta}}_{LS}(k) = \left(\Phi(k)^T \Phi(k)\right)^{-1} \Phi(k)^T \underline{Y}(k)$$

Now consider that we have available the input/output pair sampled at the $(k+1)^{th}$ sample.

The measurement vector then becomes:

$$\underline{Y}(k+1) = \begin{bmatrix} y(m+d) \\ y(m+d+1) \\ \vdots \\ y(k) \\ \hline y(k+1) \end{bmatrix} = \begin{bmatrix} \text{----} \\ \\ \end{bmatrix}$$

The estimated output vector is:

$$\underline{\hat{Y}}(k+1) = \begin{bmatrix} \hat{y}(m+d) \\ \hat{y}(m+d+1) \\ \vdots \\ \hat{y}(k) \\ \hline \hat{y}(k+1) \end{bmatrix} = \begin{bmatrix} \underline{\hat{Y}}(k) \\ \hline \hat{y}(k+1) \end{bmatrix}$$

Where:

$$\hat{y}(k+1) = \hat{a}_1 y(k) + \cdots \hat{a}_n y(k-n+1) + \hat{b}_1 u(k-d) + \cdots \hat{b}_m u(k-d-m+1)$$

Or :

$$\hat{y}(k+1) = \underline{\psi}^T(k+1)\,\underline{\hat{\theta}}$$

The estimator equation can then be written as:

$$\begin{bmatrix} \hat{y}(m+d) \\ \hat{y}(m+d+1) \\ \vdots \\ \vdots \\ \hat{y}(k) \\ \hat{y}(k+1) \end{bmatrix} = \begin{bmatrix} y(m+d-1) & \cdots & y(m+d-n) & u(m-1) & \cdots & u(0) \\ y(m+d) & \cdots & y(m+d-n+1) & u(m) & \cdots & u(1) \\ y(m+d+1) & \cdots & y(m+d-n+2) & u(m+1) & \cdots & u(2) \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ y(k-1) & \ddots & y(k-n) & u(k-d-1) & \cdots & u(k-d-m) \\ y(k) & \cdots & y(k-n+1) & u(k-d) & \cdots & u(k-d-m+1) \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_n \\ \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_m \end{bmatrix}$$

Which could be written as:

$$\underline{\hat{Y}}(k+1) = \begin{bmatrix} \underline{\hat{Y}}(k) \\ \hat{y}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi(k) \\ \underline{\psi}^T(k+1) \end{bmatrix} \underline{\hat{\theta}}$$

The updated parameter vector could be calculated as:

$$\underline{\hat{\theta}}_{LS}(k+1) = \left(\Phi(k+1)^T \Phi(k+1)\right)^{-1} \Phi(k+1)^T \underline{Y}(k+1)$$

Consider:

$$\Phi(k+1)^T \Phi(k+1) = \begin{bmatrix} \Phi(k) \\ \hline \underline{\psi}^T(k+1) \end{bmatrix}^T \begin{bmatrix} \Phi(k) \\ \hline \underline{\psi}^T(k+1) \end{bmatrix} =$$

Now we will define:

$$P(k) = \left(\Phi(k)^T \Phi(k)\right)^{-1}$$
$$P(k+1) = \left(\Phi(k+1)^T \Phi(k+1)\right)^{-1}$$

Hence we can now write:

$$P(k+1) = \left[\Phi(k)^T \Phi(k) + \underline{\psi}(k+1)\underline{\psi}^T(k+1)\right]^{-1}$$

Householder's Matrix Inversion lemma states:

$$\left(A + BCD\right)^{-1} = A^{-1} - A^{-1}B\left(C^{-1} + DA^{-1}B\right)^{-1} DA^{-1}$$

Choose:

Then:

$$P(k+1) = \left[P(k)^{-1} + \underline{\psi}(k+1)\underline{\psi}^T(k+1)\right]^{-1}$$

Could be written as:

This can be simplified to:

$$P(k+1) = P(k) - \frac{P(k)\underline{\psi}(k+1)\underline{\psi}^T(k+1)P(k)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)}$$

The updated parameter vector is then generated from:

$$\underline{\hat{\theta}}_{LS}(k+1) = \left(\Phi(k+1)^T\Phi(k+1)\right)^{-1}\Phi(k+1)^T\underline{Y}(k+1)$$

But:

$$\Phi(k+1)^T\underline{Y}(k+1) = \left[\begin{array}{c}\Phi(k)\\ \hline \underline{\psi}^T(k+1)\end{array}\right]^T\left[\begin{array}{c}\underline{Y}(k)\\ \hline y(k+1)\end{array}\right] =$$

The parameter estimate can then be written as:

$$\underline{\hat{\theta}}_{LS}(k+1) = \left[P(k) - \frac{P(k)\underline{\psi}(k+1)\underline{\psi}^T(k+1)P(k)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)}\right]\left[\Phi(k)^T\underline{Y}(k) + \underline{\psi}(k+1)y(k+1)\right]$$

Multiplying this out:

$$\underline{\hat{\theta}}_{LS}(k+1) = P(k)\Phi(k)^T\underline{Y}(k) + P(k)\underline{\psi}(k+1)y(k+1)$$
$$-\frac{P(k)\underline{\psi}(k+1)\underline{\psi}^T(k+1)P(k)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)}\Phi(k)^T\underline{Y}(k) - \frac{P(k)\underline{\psi}(k+1)\underline{\psi}^T(k+1)P(k)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)}\Phi(k)^T\underline{\psi}(k+1)y(k+1)$$

TUTORIAL:

Show that the parameter update algorithm can be expressed as:

$$L(k+1) = \frac{P(k)\underline{\psi}(k+1)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)}$$

$$\hat{y}(k+1) = \underline{\psi}^T(k+1)\underline{\hat{\theta}}_{LS}(k)$$

$$\underline{\hat{\theta}}_{LS}(k+1) = \underline{\hat{\theta}}_{LS}(k) + L(k+1)\left(y(k+1) - \hat{y}(k+1)\right)$$

$$P(k+1) = \left[P(k) - \frac{P(k)\underline{\psi}(k+1)\underline{\psi}^T(k+1)P(k)}{1+\underline{\psi}^T(k+1)P(k)\underline{\psi}(k+1)} \right]$$

8.3.1 A Note on the Choice of Excitation Signal

The need to excite the dynamics of the process over the frequency range of interest – consider the solution to batch least squares:

$$\underline{\hat{\theta}}_{LS} = \left(\Phi^T\Phi\right)^{-1}\Phi^T\underline{Y}$$

If the data collected is not rich enough in information then:

Choices of excitation include:

8.3.2 Adaptive Pole Placement Control

Use RLS algorithm to provide on-line updates of the parameters – hopefully this will allow controller to track slow changes in the process.

Each new update of the parameter vector is then used, to design the controller polynomials, based on the polynomial pole-placement technique.