POLITECNICO DI MILANO

Output-error model identification: linear time-invariant systems

Marco Lovera

Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano

The OE method for LTI systems

- For linear time-invariant systems the OE problem can be formulated either in the time-domain or in the frequency-domain.

- For the time-domain case, simplifications to the general approach can be made by exploiting linearity.

- While the frequency-domain approach is conceptually possible also in the nonlinear case, it is usually considered practical in the LTI case only.

The OE method for LTI systems

Model class:

$$\mathcal{M}(\theta): \quad \begin{aligned} \dot{x} &= A(\theta)x + B(\theta)u, \quad x(0) = x_0 \\ y &= C(\theta)x + D(\theta)u \end{aligned}$$

Assumptions:

- The model class is structurally identifiable
- $\mathcal{S} \in \mathcal{M}(\theta)$ or equivalently $\exists \theta^\star : \quad \mathcal{S} = \mathcal{M}(\theta^\star)$
- $\theta \in \mathbb{R}^{n_\theta}$.
- The available dataset guarantees experimental identifiability.

Measurement model:

- Measurements are discrete
- Sampling is uniform and defined by

$$t_k = t_0 + kT_s, \quad k = 1, \dots, K.$$

- Measurement equation:

$$y_m(k) = y(k) + v(k)$$

  - $y(k) = y(kT_s)$
  - $v(k) = G(0, \sigma^2), \quad E[v(i)v(j)] = 0, \quad i \neq j$

- The general derivation of the OE method can be repeated, to get to the same cost function:.

$$J(\theta) = \frac{1}{2\sigma^2}\sum_{k=1}^{K}(y_m(k) - y(k;\theta))^2 = \frac{1}{2\sigma^2}\sum_{k=1}^{K} e(k;\theta)^2.$$

- Linearity of the model class makes the computations more efficient during the optimisation, as sensitivities can be computed analytically.

Indeed, following the analytical approach, starting from the model equations:

$$\dot{x} = A(\theta)x + B(\theta)u, \quad x(0) = x_0$$
$$y = C(\theta)x + D(\theta)u$$

and differentiating with respect to a component of the parameter vector we get for the state equation

$$\frac{\partial}{\partial\theta_j}\frac{dx}{dt} = A(\theta)\frac{\partial x}{\partial\theta_j} + \frac{\partial A(\theta)}{\partial\theta_j}x + \frac{\partial B(\theta)}{\partial\theta_j}u, \quad \frac{\partial x(0)}{\partial\theta_j} = 0.$$

Interchanging derivatives on the left-hand side we get:

$$\frac{d}{dt}\left(\frac{\partial x}{\partial \theta_j}\right) = A(\theta)\frac{\partial x}{\partial \theta_j} + \frac{\partial A(\theta)}{\partial \theta_j}x + \frac{\partial B(\theta)}{\partial \theta_j}u, \quad \frac{\partial x(0)}{\partial \theta_j} = 0, \quad j = 1, \ldots, n_\theta$$

and similarly for the output equation

$$\left(\frac{\partial y}{\partial \theta_j}\right) = C(\theta)\frac{\partial x}{\partial \theta_j} + \frac{\partial C(\theta)}{\partial \theta_j}x + \frac{\partial D(\theta)}{\partial \theta_j}u, \quad j = 1, \ldots, n_\theta.$$

Therefore, it is possible to compute the required sensitivities by simulating the state space models defined by the above state and output equations.

Note that:

$$\frac{d}{dt}\left(\frac{\partial x}{\partial \theta_j}\right) = A(\theta)\frac{\partial x}{\partial \theta_j} + \frac{\partial A(\theta)}{\partial \theta_j}x + \frac{\partial B(\theta)}{\partial \theta_j}u, \qquad \frac{\partial x(0)}{\partial \theta_j} = 0, \qquad j = 1, \ldots, n_\theta$$

$$\left(\frac{\partial y}{\partial \theta_j}\right) = C(\theta)\frac{\partial x}{\partial \theta_j} + \frac{\partial C(\theta)}{\partial \theta_j}x + \frac{\partial D(\theta)}{\partial \theta_j}u, \qquad j = 1, \ldots, n_\theta.$$

is a set of n_θ LTI models, with inputs *x* and *u* in which:

- *u* is measured;
- *x* can be obtained by simulation of the identified model;
- sensitivities of *A*, *B*, *C*, *D* can be computed before the fact;
- LTI simulations can be carried out efficiently and (almost) exactly.

Practical issues

In multiple-output problems, scaling is an important factor which has been already mentioned.

Other important issues:

- Choice of sampling interval
- Prefiltering
- Stability

Choice of sampling interval

The discrete measurement model

$$t_k = t_0 + kT_s, \qquad k = 1, \ldots, K.$$

$$y_m(k) = y(k) + v(k)$$

$$y(k) = y(kT_s)$$

assumes a suitable choice for the sampling interval used in collecting data.

Therefore, guidelines for the choice of this parameter are needed.

Choice of sampling interval

As a minimum sampling must be «fast» with respect to the relevant dynamics (Shannon-Nyquist Theorem).

In practice this means T_s should be 5-10 times shorter than the fastest dynamics of interest.

Choosing T_s arbitrarily small is not necessarily a good idea as this:

- Increases the number of data-points to be processed;
- With little or no gain in terms of information on the parameters.

Choice of sampling interval

Note that in some problems *multirate* sampling may occur, *i.e.*, measurements form different sensors are received at different sampling rates.

Example: in flight control

- IMU measurements at 100 Hz (or more)
- GPS measurements at 10 Hz (or less).

The OE algorithm must be modified to compute the contributions to the cost at the appropriate rate.

$$J_i(\theta) = \frac{1}{2\sigma_i^2} \sum_{k=1}^{K} e_i(k;\theta)^2, \quad i = 1, \dots, p.$$

Prefiltering

- The main weakness of the OE method is the assumption

$$\mathcal{S} \in \mathcal{M}(\theta)$$

  which is hardly ever true in practice.

- Therefore, if the data contains contributions from modes or dynamics not included in the model class, bias in the estimates can be expected.

- A way of mitigating this issue is by introducing *error prefiltering*, which corresponds to a modification of the cost function as follows.

Prefiltering

- Instead of minimising the cost

$$J(\theta) = \frac{1}{2\sigma^2} \sum_{k=1}^{K} (y_m(k) - y(k;\theta))^2 = \frac{1}{2\sigma^2} \sum_{k=1}^{K} e(k;\theta)^2.$$

- We define the modified cost

$$J_f(\theta) = \frac{1}{2\sigma^2} \sum_{k=1}^{K} e_f(k;\theta)^2.$$

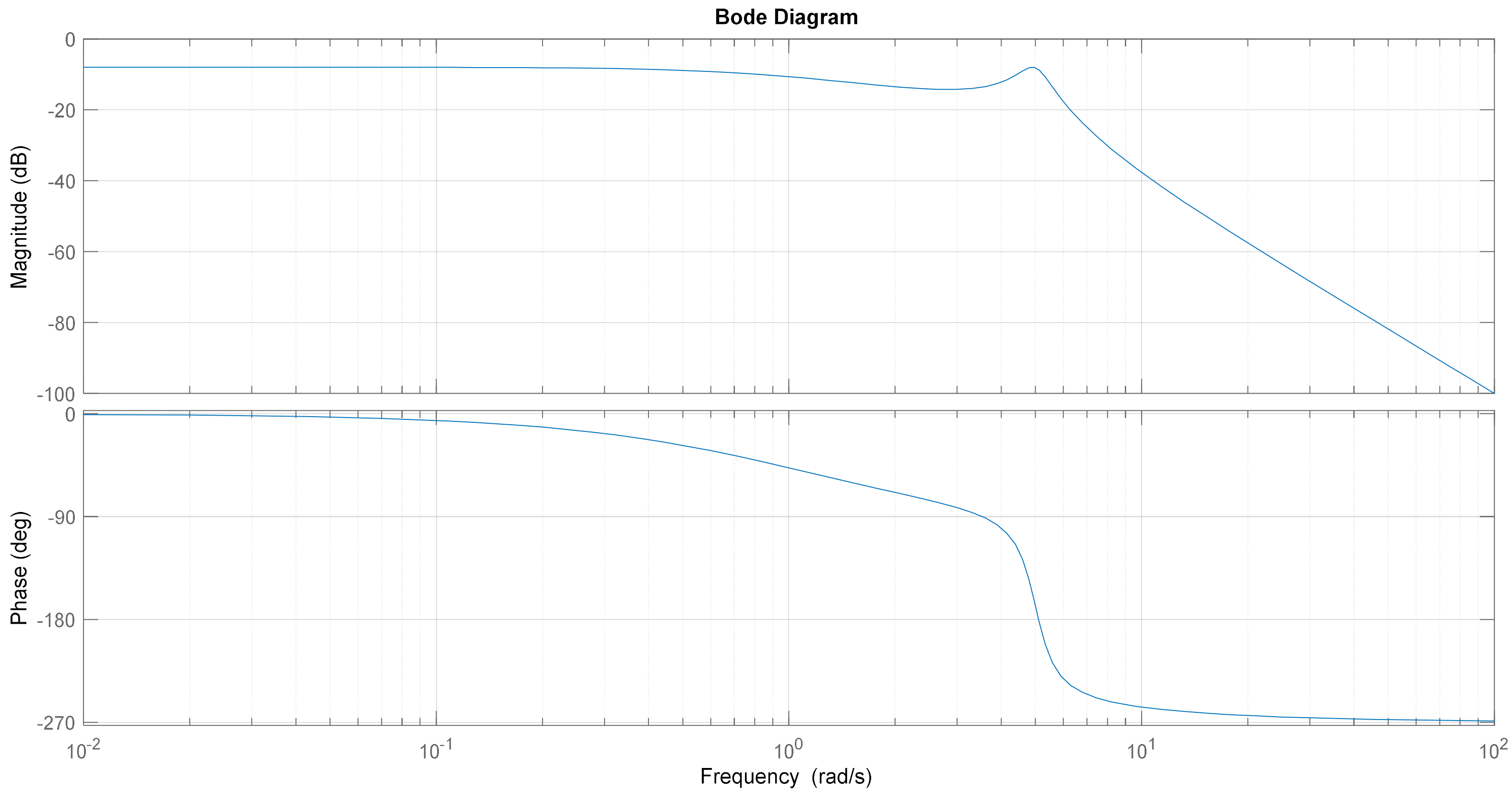
- Where the *filtered error* is defined as $e_f(k;\theta) = L(z)e(k;\theta)$

- And *L*(*z*) is a linear digital filter design to attenuate the contribution of undesired dynamics.

Prefiltering

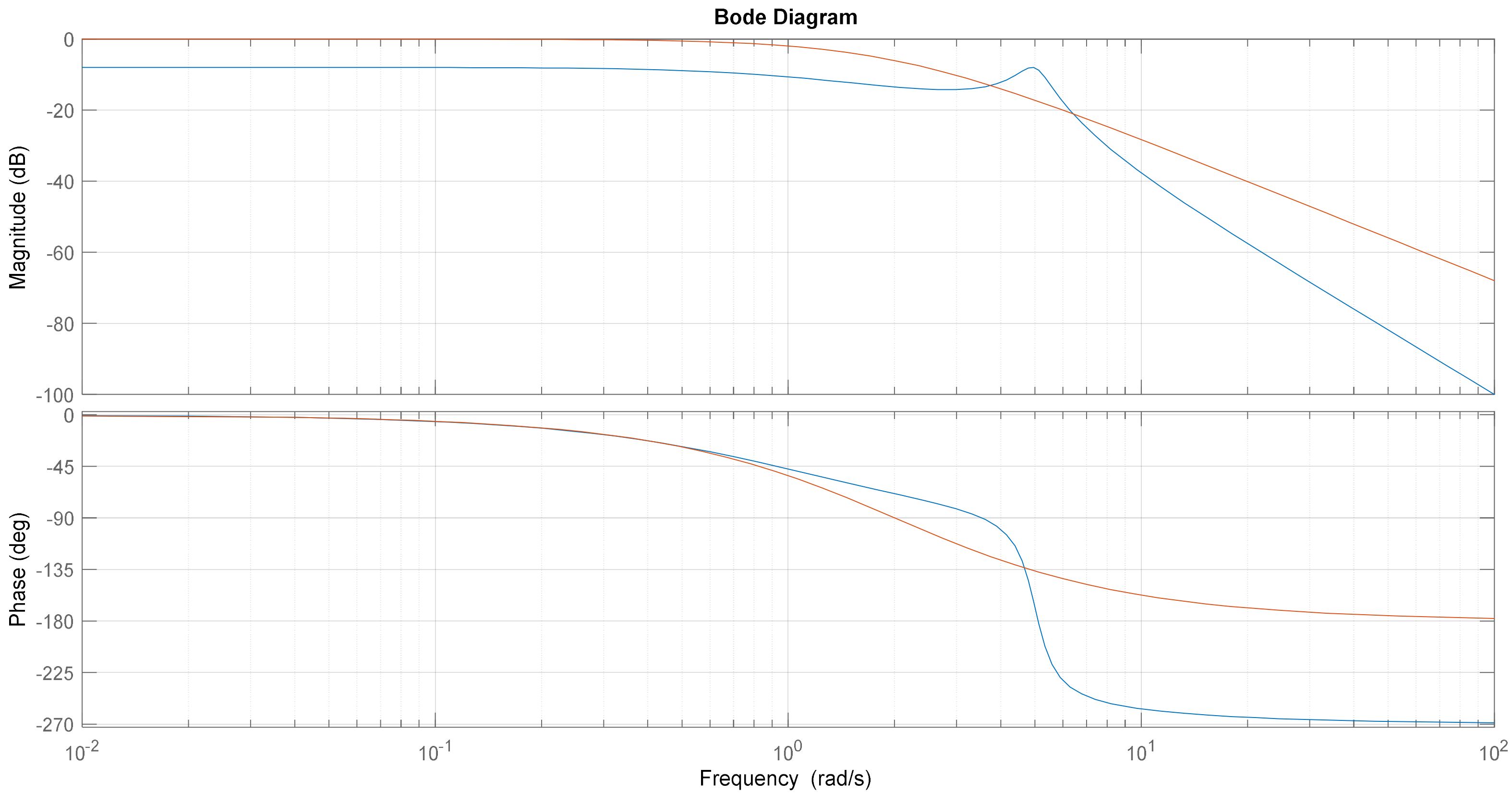
Example: the true system is $G(s) = \frac{\mu}{\tau s + 1} \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$

with frequency response function



Prefiltering

The model class is instead defined as $M(s;\mu,\tau)=\dfrac{\mu}{\tau s+1}$.
The contribution of the underdamped mode can be reduced by using a prefilter such that

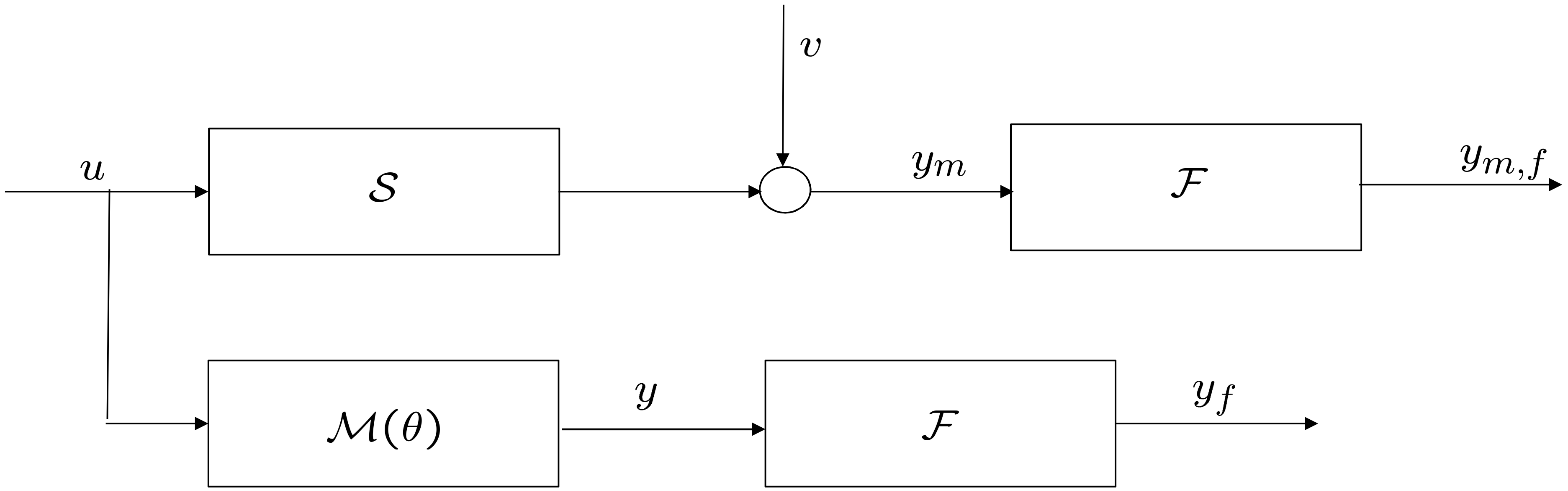


From error prefiltering to data prefiltering

- Error filtering as we defined it should be repeated at every evaluation of the cost function.
- Therefore it represents a significant increase in the cost of the optimisation procedure.
- For linear systems however filtering can be carried out before the fact on the data, and the filter can be removed from the optimisation process, as follows.

From error prefiltering to data prefiltering

Consider the block diagram:

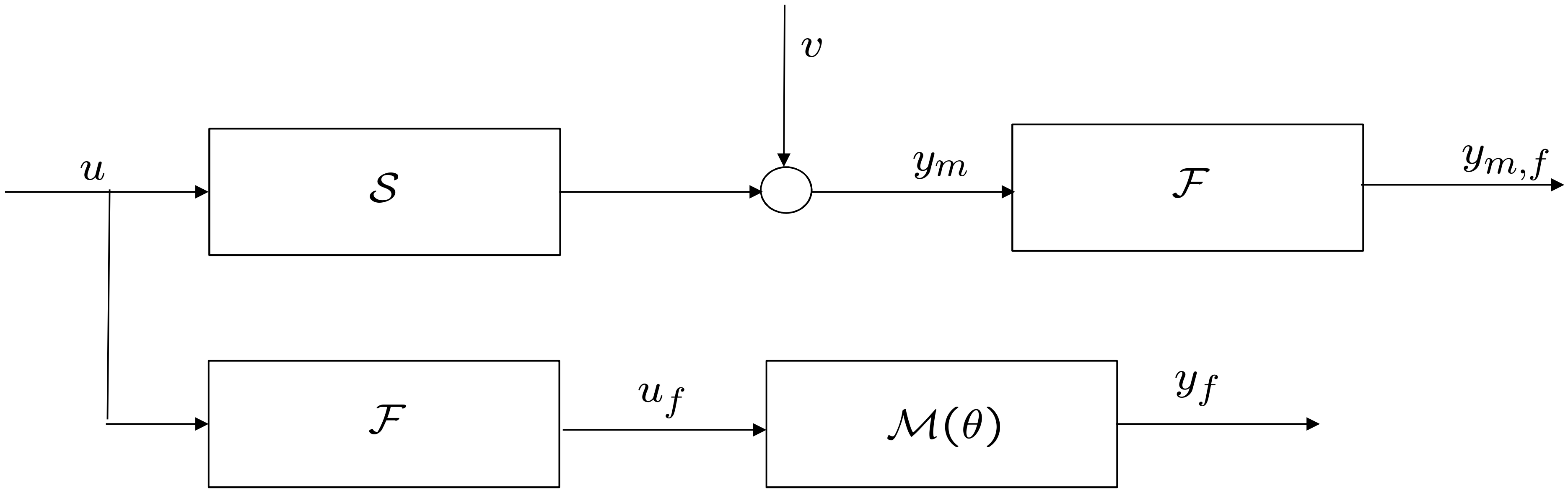


Then the filtered error can be written as:

$$e_f(k;\theta) = L(z)e(k;\theta) = L(z)y_m(k) - L(z)y(k;\theta) =$$
$$= y_{m,f}(k) - y_f(k;\theta)$$

From error prefiltering to data prefiltering

But all the blocks are linear so they can be interchanged:



Therefore instead of filtering *e* one can just filter the data *before* starting the optimisation and apply the usual OE method to the filtered data.

From error prefiltering to data prefiltering

- *Caveat*: data and error prefiltering are equivalent only in the linear case.
- Using data prefiltering in nonlinear problems can lead to unpredictable results.
- Note that some existing software tools for system identification allow you to do it!

The model class is given by

$$\mathcal{M}(\theta): \quad \begin{array}{l} \dot{x} = A(\theta)x + B(\theta)u, \quad x(0) = x_0 \\ y = C(\theta)x + D(\theta)u \end{array}$$

and we have seen that at each iteration simulations of the model evaluated for different choices of the parameter must be carried out.

Assume now that the true system is asymptotically stable.

Clearly, if the parameter is such that the model becomes unstable at some point during the optimisation, this is likely to lead to numerical issues and failure of the algorithm.

Stability

The problem can be handled in many different ways:

- Use a *constrained* optimisation algorithm to enforce bounds on the parameters.

- Check stability at each iteration and *modify* the current estimates to lead to a stable system.

Note that the problem exists also in the nonlinear version of OE but it is a lot harder to handle.

This issue with time-domain OE is the main argument used by proponents of frequency-domain versions.

The frequency-domain OE method

Time-domain OE has two main downsides:

- a large number of data-points need to be processed.
- Potential stability issues may arise in the optimisation.

In many applications however it is possible to:

- test the true system using individual sinusoids, or sums of sinusoids or sine sweeps;
- construct point-wise estimates of the frequency response function of the system;
- treat the estimated samples of the frequency response function as if they were measurements.

This leads to the formulation of the frequency-domain OE method.

For the moment we will consider the estimated frequency response function as data, this estimation problem will be studied separately later.

The model class is a MIMO model structure in state space form:

$$\dot{\boldsymbol{x}} = A(\theta)\boldsymbol{x} + B(\theta)\boldsymbol{u}$$
$$\boldsymbol{y} = C(\theta)\boldsymbol{x} + D(\theta)\boldsymbol{u}$$

Where the A, B, C, D matrices depend on the vector of parameters θ.

From the state space form, we are able to get the frequency response function matrix:

$$G(j\omega, \theta) \in \mathbb{C}^{p\times m}$$

Which we consider in this framework as the “true output”.

The frequency-domain OE method

We have K measured samples of the frequency response function for a generic MIMO LTI system with p outputs and m inputs:

$$G_m(j\omega_k) \text{ as } k = 1, \dots, K$$

$$G_m(j\omega_k) \in \mathbb{C}^{p \times m}$$

Data is corrupted by *circular* white Gaussian measurement noise:

$$G_m(j\omega_k) = G(j\omega_k) + V(j\omega_k) \text{ where } V(j\omega_k) \in \mathbb{C}^{p \times m}$$

and

$$E[V(j\omega_k)] = 0 \text{ as } k = 1,2, \dots$$

$$E[V(j\omega_k)\bar{V}(j\omega_k)] = \Lambda^2 \text{ (diagonal real matrix) as } k = 1,2, \dots, K$$

$$E[V(j\omega_k)\bar{V}(j\omega_i)] = 0 \quad \forall\, k \neq i$$

The frequency-domain OE method

In order to build the Likelihood function we have to organize the transfer matrices in row vectors:

$$G(j\omega_k) = \begin{bmatrix} g_{11}(j\omega_k) & \cdots & g_{1m}(j\omega_k) \\ \vdots & \ddots & \vdots \\ g_{p1}(j\omega_k) & \cdots & g_{pm}(j\omega_k) \end{bmatrix}$$

$$\boldsymbol{G}(j\omega_k) = \begin{bmatrix} g_{11}(j\omega_k) \\ g_{12}(j\omega_k) \\ \vdots \\ g_{pm-1}(j\omega_k) \\ g_{pm}(j\omega_k) \end{bmatrix}$$

And we double each vector in order to separate the real from the imaginary part of each element:

$$\boldsymbol{G}(j\omega_k) = \begin{bmatrix} g_{11}(j\omega_k) \\ g_{12}(j\omega_k) \\ \vdots \\ g_{pm-1}(j\omega_k) \\ g_{pm}(j\omega_k) \end{bmatrix}$$

$$Y_k = \begin{bmatrix} Re\{\boldsymbol{G}(j\omega_k)\} \\ Im\{\boldsymbol{G}(j\omega_k)\} \end{bmatrix}$$

We do the same thing with measurements:

$$Y_{mk} = \begin{bmatrix} Re\{\boldsymbol{G}_m(j\omega_k)\} \\ Im\{\boldsymbol{G}_m(j\omega_k)\} \end{bmatrix}$$

And we organize the noise variance matrix Λ^2 as:

$$R = \begin{bmatrix} \frac{\Lambda^2}{2} & 0 \\ 0 & \frac{\Lambda^2}{2} \end{bmatrix}$$

So we can write that:

- $E[Y_{mk}] = Y_k$ (because the noise is zero mean)
- $Var(Y_{mk}) = R$

We assumed Gaussian noise, so

$$Y_{mk} \sim G(Y_k, R)$$

and the probability density function of the data is

$$f_{Y_{mk}} = \frac{1}{\sqrt{\det R\,(2\pi)}} e^{-\frac{1}{2}(q-Y_k)^T R^{-1}(q-Y_k)}$$

Hence, we are able to find the Likelihood function:

$$L\left(Y_{mk}\Big|_1^K \setminus \theta\right) = \frac{1}{\left(\det R\,(2\pi)^K\right)^{\frac{1}{2}}} e^{-\sum_{k=1}^{K}\frac{1}{2}(Y_{mk}-Y_k)^T R^{-1}(Y_{mk}-Y_k)}$$

We have to find that value of the parameter θ that maximises L, which is equivalent to minimizing the cost function:

$$J(\theta) = \frac{1}{2}\sum_{k=1}^{K}(Y_{mk} - Y_k)^T R^{-1}(Y_{mk} - Y_k)$$

The goal of our process is to find:

$$\min_{\theta} J(\theta)$$

In order to minimize the cost function, we may apply the so called **modified Newton Raphson Method.**

For the sake of simplicity we rename:

$$e(k) = Y_{mk} - Y_k$$

And we estimate the R matrix:

$$R = \frac{1}{K}\sum_{k=1}^{K} e(k)e(k)^T$$

Only the diagonal elements of R are estimated, enforcing an assumption that the measurement noise sequences for the measured outputs are uncorrelated with one another.

The frequency-domain OE method

To get the minimum value of J we run the iterative algorithm:

1. Choose an initial guess θ_0
2. Compute $G = \left[\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\right]$
3. Compute $H = \left[\frac{\partial^2 J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T}|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}\right]$
4. Get the increment $\Delta\boldsymbol{\theta} = -H^{-1}G$
5. Get the updated $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}$
6. Update the R matrix

Stopping criteria:

$$\frac{\|\boldsymbol{\theta}_{\boldsymbol{k+1}} - \boldsymbol{\theta}_{\boldsymbol{k}}\|}{\|\boldsymbol{\theta}_{\boldsymbol{k}}\|} < 10^{-5}$$

$$\frac{\|\boldsymbol{J}(\boldsymbol{\theta}_{\boldsymbol{k+1}}) - \boldsymbol{J}(\boldsymbol{\theta}_{\boldsymbol{k}})\|}{\|\boldsymbol{J}(\boldsymbol{\theta}_{\boldsymbol{k}})\|} < 10^{-5}$$

$$\|G\| < 10^{-5}$$

Ill-conditioning of the Hessian may occur, due to:

1. Overparameterization
2. Mis-specification of the model class
3. Insufficient information content in the data

Main consequences:

1. M (the information matrix) may be negative definite. This results in a cost increase: $J(\theta_{k+1}) > J(\theta_k)$
2. The step size $\Delta\theta$ can be large in one or more directions

Treatment: regularisation

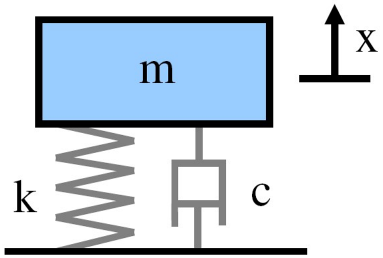
$$M^{-1} = (M_0 + kA)^{-1}$$

Example 1

We have a mass-spring-damper linear SISO system.
In state-space form:

$$\begin{bmatrix}\dot{x}\\ \ddot{x}\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{k}{m} & -\frac{c}{m}\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{m}\end{bmatrix}[F]$$

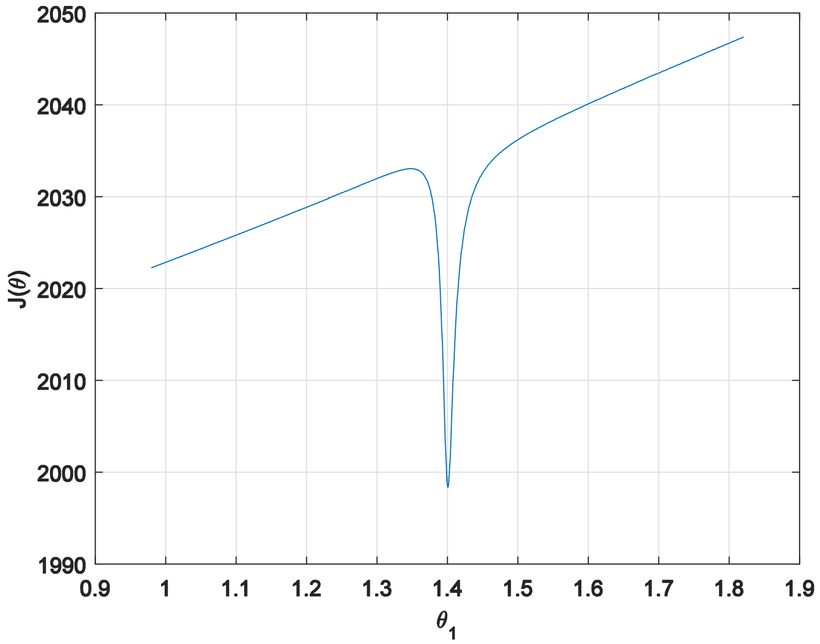
$$y = 100\begin{bmatrix}x\\ \dot{x}\end{bmatrix} = 100\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\end{bmatrix}$$



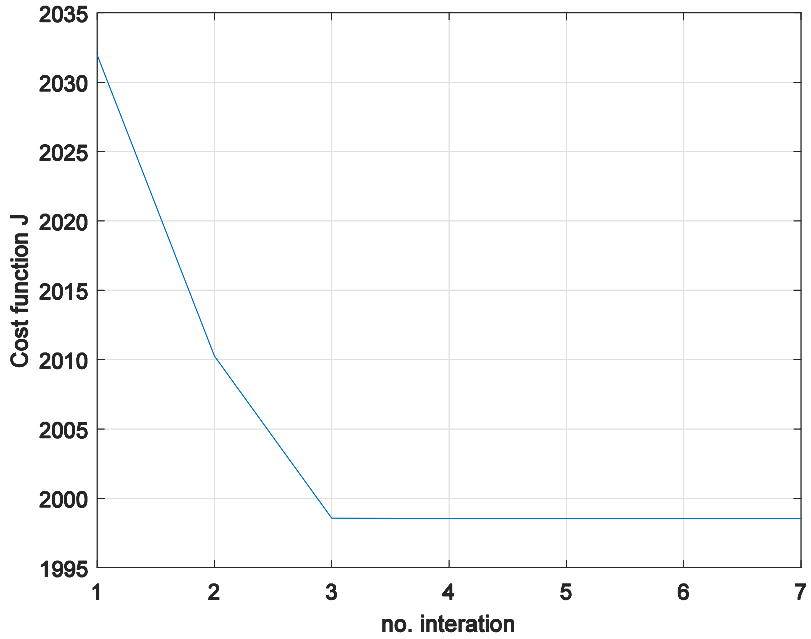
In practice, we are able to easily compute the mass and the spring stiffness. However, the parameter c is representative of many effects which can be only estimated through experiments.

To this purpose measure the system response and we apply the output error process to estimate that parameter.

Example 1

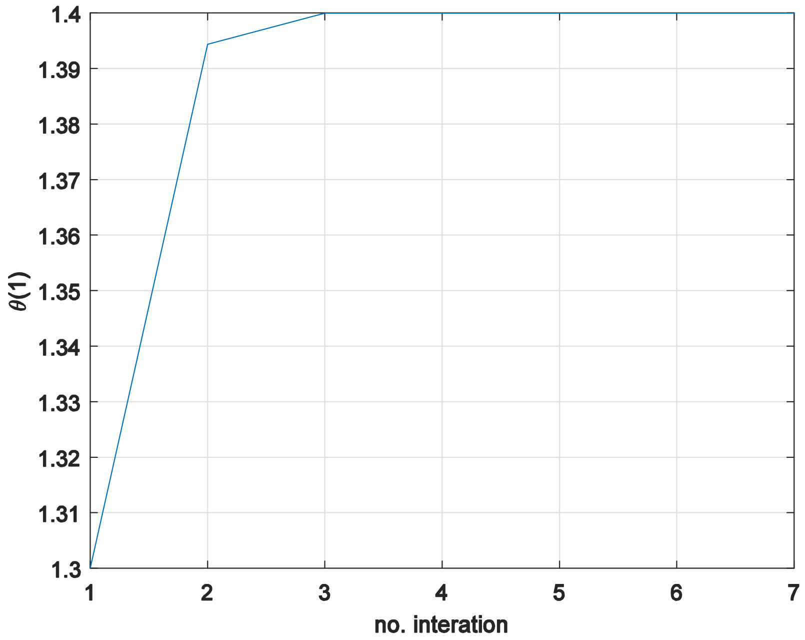


2050
2040
2030
2020
2010
2000
1990
J(θ)
0.9
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
θ1

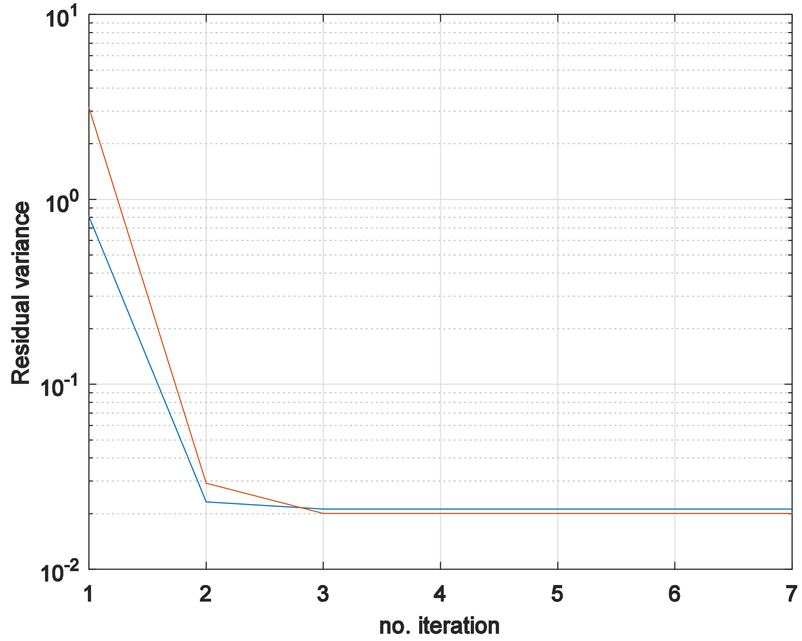


2035
2030
2025
2020
2015
2010
2005
2000
1995
Cost function J
1
2
3
4
5
6
7
no. interation

Example 1



1.4
1.39
1.38
1.37
1.36
1.35
1.34
1.33
1.32
1.31
1.3
θ(1)
1
2
3
4
5
6
7
no. interation

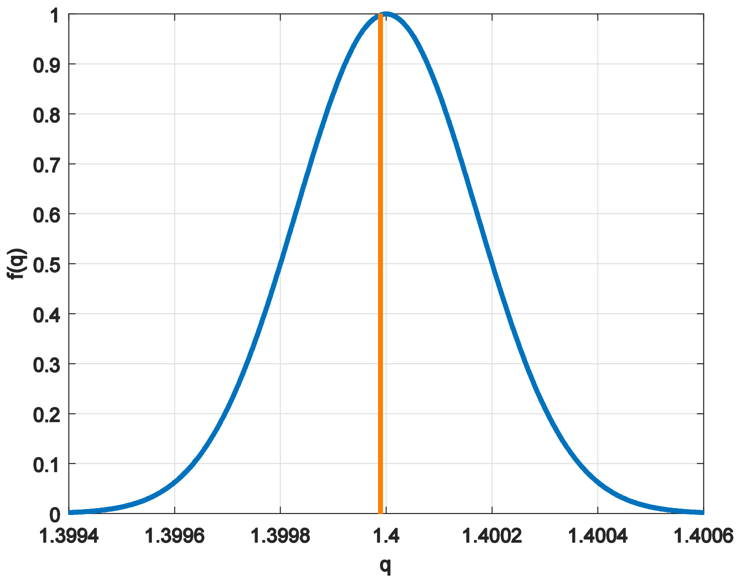


10^1
10^0
10^{-1}
10^{-2}
Residual variance
1
2
3
4
5
6
7
no. iteration

Example 1

Parameter	True Value	Estimated Value	$s(\theta)$	$\frac{100s(\theta)}{\theta}$
c	1.4	1.39999	0.00017	0.01202

- No. iterations: 7



Example 2

We can conveniently rewrite the system of Example 1 using ω_n (natural frequency) and ξ (damping):

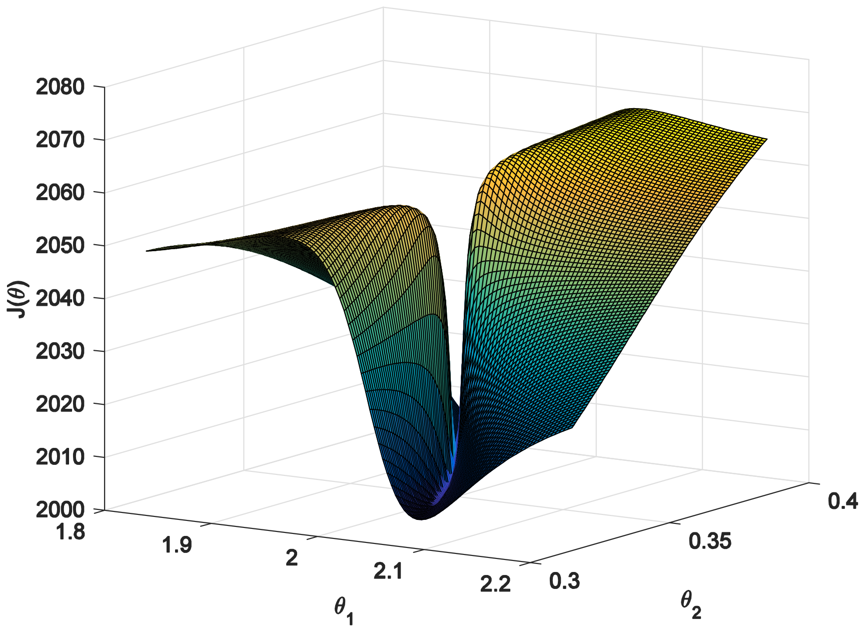
$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\omega_n\xi \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} [F]$$

$$y = 100 \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = 100 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

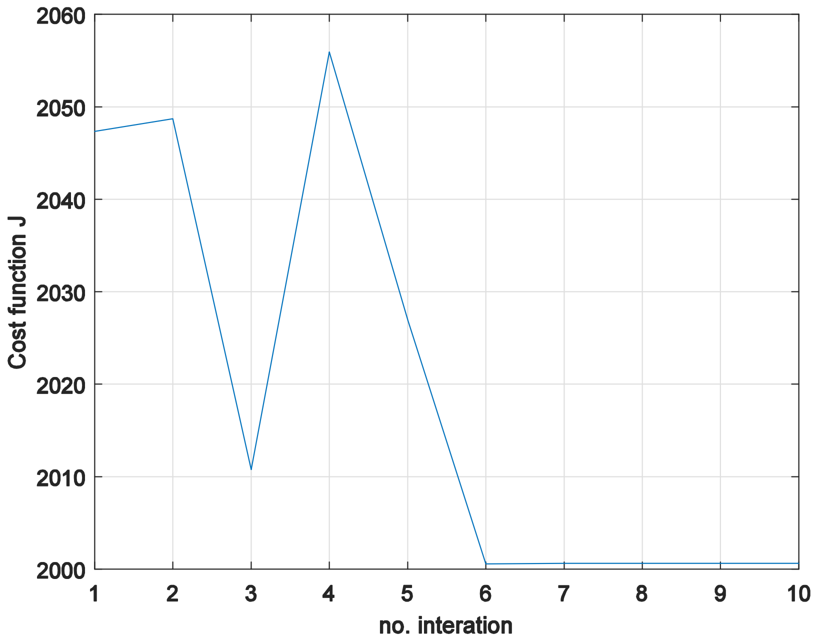
Suppose that we do not know ω_n and ξ. We are able to identify them using the output error method.

The only difference with the previous case is that we have to estimate two parameters instead than just one.

Example 2

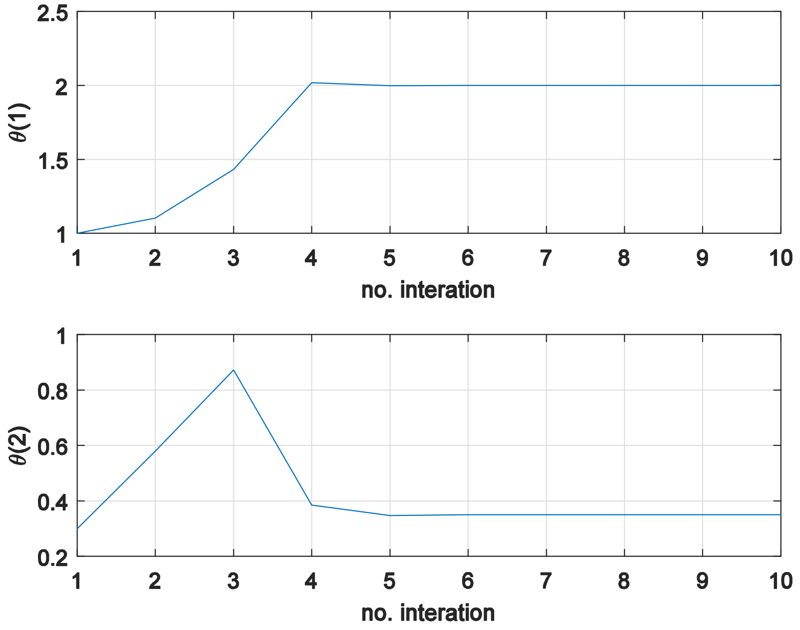


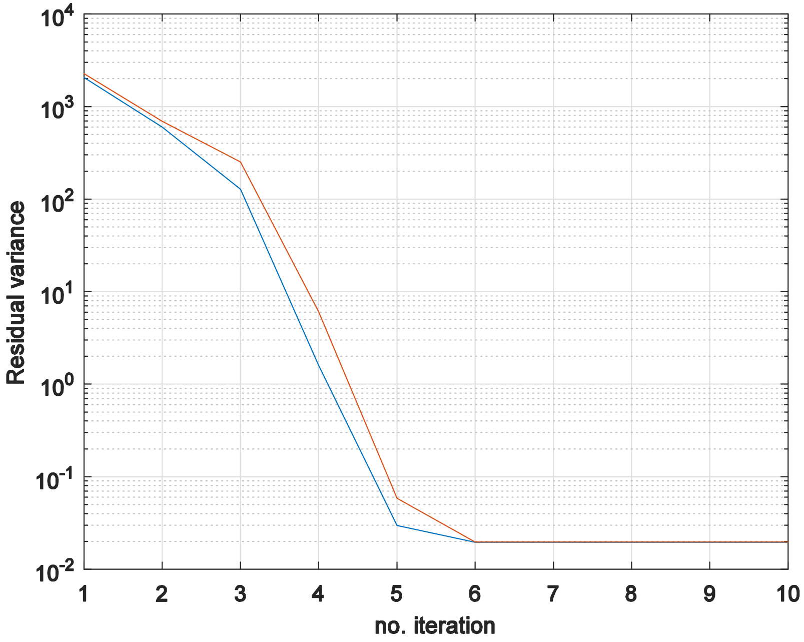
2080
2070
2060
2050
2040
2030
2020
2010
2000
J(θ)
1.8
1.9
2
2.1
2.2
θ1
0.3
0.35
0.4
θ2



2060
2050
2040
2030
2020
2010
2000
Cost function J
1
2
3
4
5
6
7
8
9
10
no. interation

Example 2


θ(1)
no. interation
θ(2)
no. interation

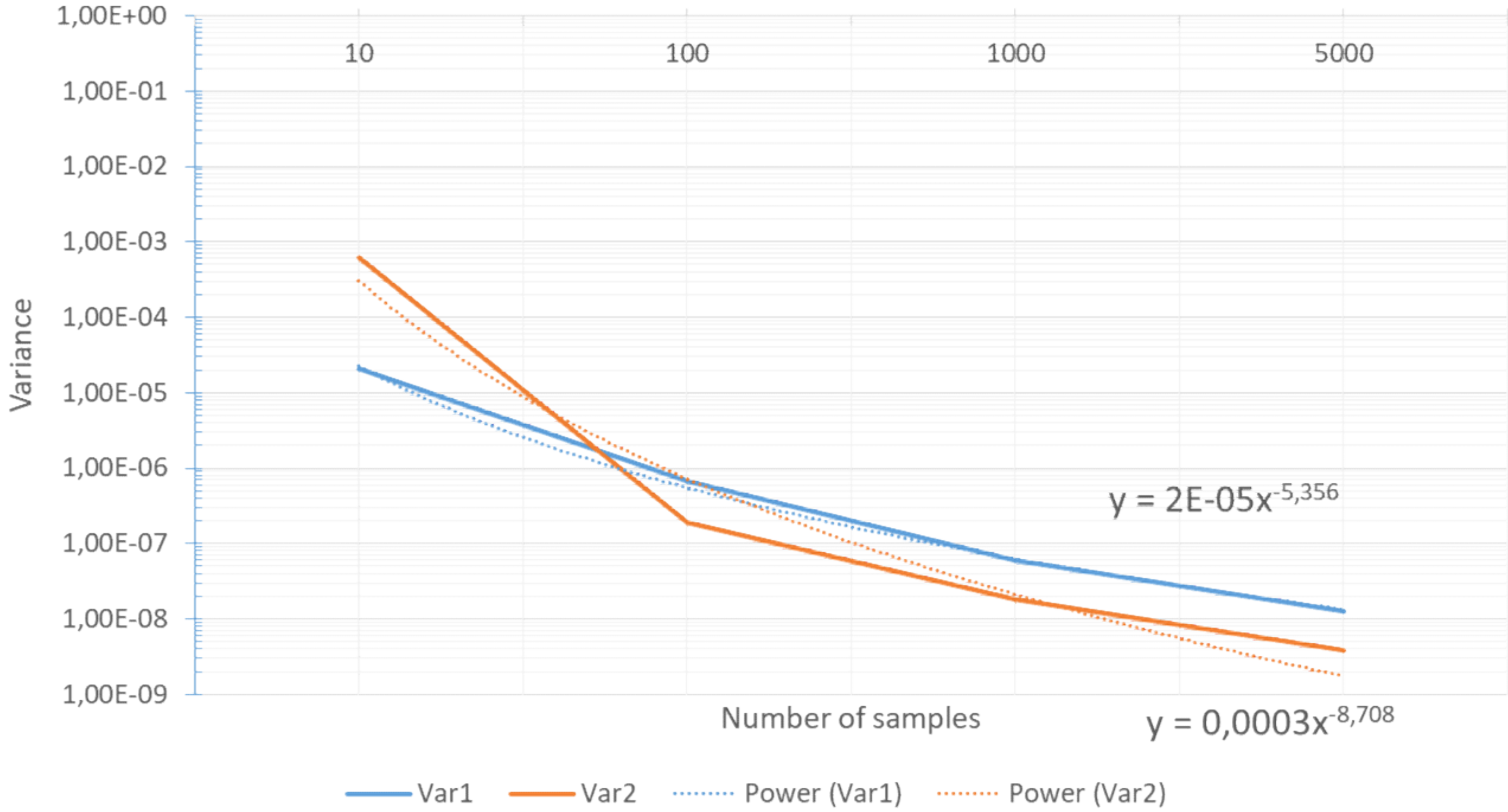

Residual variance
no. iteration

Example 2

Parameter	True Value	Estimated Value	$s(\theta)$	$\frac{100s(\theta)}{\theta}$
ω_n	2	1.99991	7.89189×10^{-5}	0.003946
ξ	0.35	0.35002	4.35355×10^{-5}	0.012438

- No. iterations: 10

Example 2


1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
1,00E-07
1,00E-08
1,00E-09
Variance
10
100
1000
5000
Number of samples
y = 2E-05x-5,356
y = 0,0003x-8,708
Var1
Var2
Power (Var1)
Power (Var2)

Aircraft flight dynamics can be described using a nonlinear, 6 degrees of freedom model, given by the equations:

Force equations:

$$\dot{u} = rv - qw + \frac{\bar{q}S}{m} C_X - g \sin\theta + \frac{T}{m}$$

$$\dot{v} = pw - ru + \frac{\bar{q}S}{m} C_Y + g \cos\theta \sin\phi$$

$$\dot{w} = qu - pv + \frac{\bar{q}S}{m} C_Z + g \cos\theta \cos\phi$$

Moment equations:

$$\dot{p} - \frac{I_{xz}}{I_x}\dot{r} = \frac{\bar{q}Sb}{I_x} C_l - \frac{(I_z - I_y)}{I_x} qr + \frac{I_{xz}}{I_x} qp$$

$$\dot{q} = \frac{\bar{q}S\bar{c}}{I_y} C_m - \frac{(I_x - I_z)}{I_y} pr - \frac{I_{xz}}{I_y}(p^2 - r^2) + \frac{I_p}{I_y}\Omega_p r$$

$$\dot{r} - \frac{I_{xz}}{I_z}\dot{p} = \frac{\bar{q}Sb}{I_z} C_n - \frac{(I_y - I_x)}{I_z} pq - \frac{I_{xz}}{I_z} qr - \frac{I_p}{I_z}\Omega_p q$$

Kinematic equations:

$$\dot{\phi} = p + \tan\theta (q \sin\phi + r \cos\phi)$$

$$\dot{\theta} = q \cos\phi - r \sin\phi$$

$$\dot{\psi} = \frac{q \sin\phi + r \cos\phi}{\cos\theta}$$

Example 3

In order to focus on particular aircraft motions, we can linearize the system using as reference condition steady, wings-level flight with no sideslip.

We can also decouple the system into the longitudinal and the lateral motion:

Longitudinal equations:

$$\dot{V} = -\frac{\bar{q}_o S}{m}\left(C_{D_V}\frac{\Delta V}{V_o} + C_{D_\alpha}\Delta\alpha + C_{D_q}\frac{q\bar{c}}{2V_o} + C_{D_\delta}\Delta\delta\right)\Delta C_D$$

$$- g\cos\gamma_o(\Delta\theta - \Delta\alpha) - \frac{T_o\sin\alpha_o}{m}\Delta\alpha$$

$$\dot{\alpha} = -\frac{\bar{q}_o S}{mV_o}\left(C_{L_V}\frac{\Delta V}{V_o} + C_{L_\alpha}\Delta\alpha + C_{L_{\dot{\alpha}}}\frac{\dot{\alpha}\bar{c}}{2V_o} + C_{L_q}\frac{q\bar{c}}{2V_o} + C_{L_\delta}\Delta\delta\right)$$

$$+ q - \frac{g\sin\gamma_o}{V_o}(\Delta\theta - \Delta\alpha) - \frac{T_o\cos\alpha_o}{mV_o}\Delta\alpha$$

$$\dot{q} = \frac{\bar{q}_o S\bar{c}}{I_y}\left(C_{m_V}\frac{\Delta V}{V_o} + C_{m_\alpha}\Delta\alpha + C_{m_{\dot{\alpha}}}\frac{\dot{\alpha}\bar{c}}{2V_o} + C_{m_q}\frac{q\bar{c}}{2V_o} + C_{m_\delta}\Delta\delta\right)$$

$$\dot{\theta} = q$$

Lateral equations:

$$\dot{\beta} = \frac{\bar{q}_o S}{mV_o}\left(C_{Y_\beta}\beta + C_{Y_p}\frac{pb}{2V_o} + C_{Y_r}\frac{rb}{2V_o} + C_{Y_\delta}\delta\right)$$

$$+ p\ \sin\alpha_o - r\ \cos\alpha_o + \frac{g\cos\theta_o}{V_o}\ \phi$$

$$\dot{p} - \frac{I_{xz}}{I_x}\dot{r} = \frac{\bar{q}_o Sb}{I_x}\left(C_{l_\beta}\beta + C_{l_p}\frac{pb}{2V_o} + C_{l_r}\frac{rb}{2V_o} + C_{l_\delta}\delta\right)$$

$$\dot{r} - \frac{I_{xz}}{I_z}\dot{p} = \frac{\bar{q}_o Sb}{I_z}\left(C_{n_\beta}\beta + C_{n_p}\frac{pb}{2V_o} + C_{n_r}\frac{rb}{2V_o} + C_{n_\delta}\delta\right)$$

$$\dot{\phi} = p + \tan\theta_o\ r$$

$$\dot{\psi} = \sec\theta_o\ r$$

Example 3

We focus on the linearized longitudinal equations:

$$\dot{\alpha} = Z_\alpha \alpha + \left(1 + Z_q\right) q + Z_\delta \delta$$
$$\dot{q} = M_\alpha \alpha + M_q q + M_\delta \delta$$
$$a_z = \frac{V_0}{g}\left(Z_\alpha \alpha + Z_q q + Z_\delta \delta\right)$$

where all the states are perturbation quantities.

In state space form:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix}=\begin{bmatrix}Z_\alpha & 1\\ M_\alpha & M_q\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix}+\begin{bmatrix}Z_\delta\\ M_\delta\end{bmatrix}\delta$$

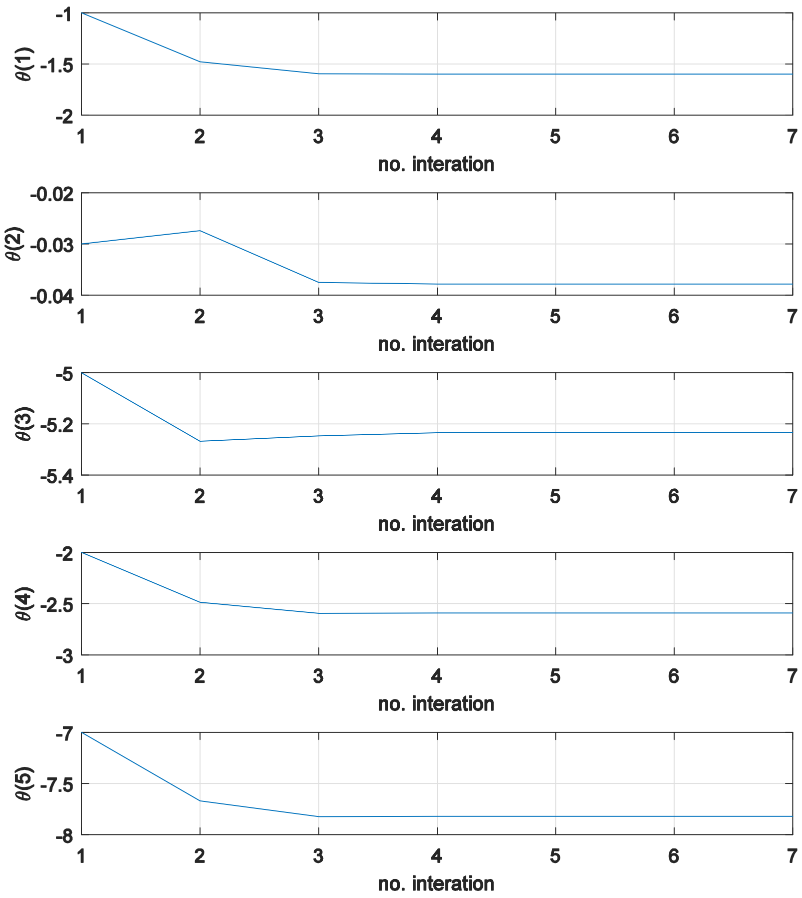
$$y=\begin{bmatrix}\alpha\\ q\\ a_z\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\\ \frac{V_0}{g}Z_\alpha & 0\end{bmatrix}\begin{bmatrix}\alpha\\ q\end{bmatrix}+\frac{V_0}{g}Z_\delta\delta$$

The goal is to estimate the following parameters

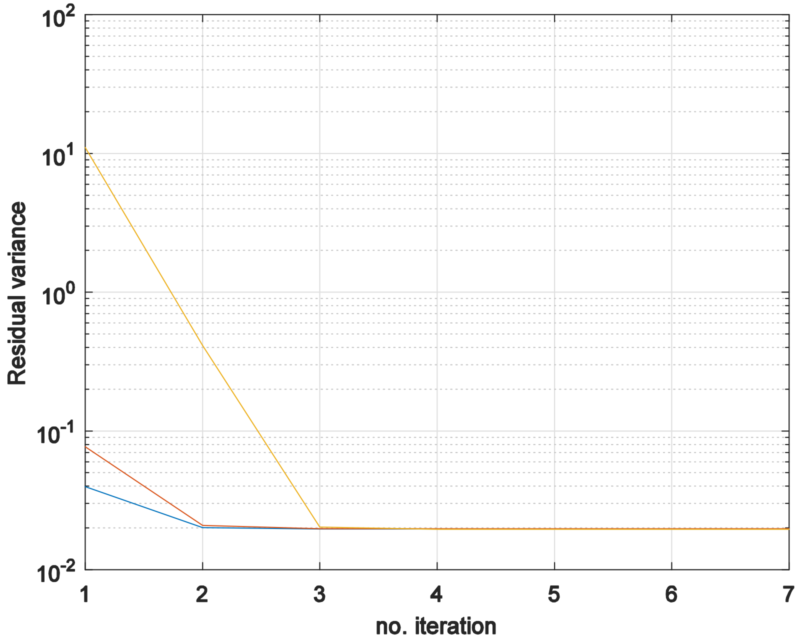
$$\boldsymbol{\theta}=[Z_\alpha \quad Z_\delta \quad M_\alpha \quad M_q \quad M_\delta]^{\mathrm{T}}$$

from samples of the freqeuncy response function.

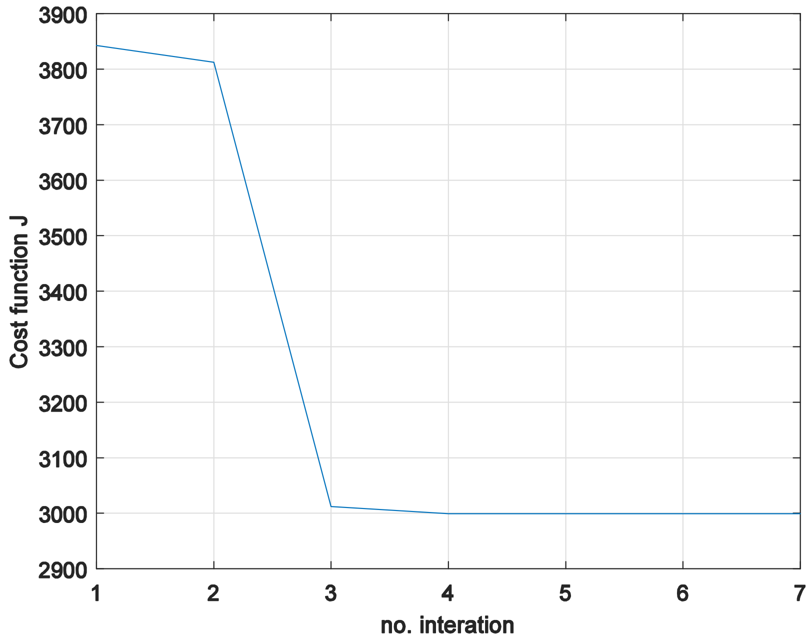
Example 3


θ(1)
-1
-1.5
-2
θ(2)
-0.02
-0.03
-0.04
θ(3)
-5
-5.2
-5.4
θ(4)
-2
-2.5
-3
θ(5)
-7
-7.5
-8
1 2 3 4 5 6 7
no. interation

Example 3



Residual variance
10^2
10^1
10^0
10^-1
10^-2
1 2 3 4 5 6 7
no. iteration



Cost function J
3900
3800
3700
3600
3500
3400
3300
3200
3100
3000
2900
1 2 3 4 5 6 7
no. interation

Example 3

Parameter	True Value	Estimated Value	$s(\theta)$	$\frac{100s(\theta)}{\theta}$
Z_α	-1.589	-1.5976	0.0036	0.2236
Z_δ	-0.038	-0.0379	0.0002	0.4991
M_α	-5.245	-5.2345	0.0045	0.0861
M_q	-2.598	-2.5907	0.0043	0.1665
M_δ	-7.852	-7.8202	0.0179	0.2294

- No. of iterations: 7

A DC Motor can be modeled as shown in the picture below.

v and i are the tension and the current within the motor armature.

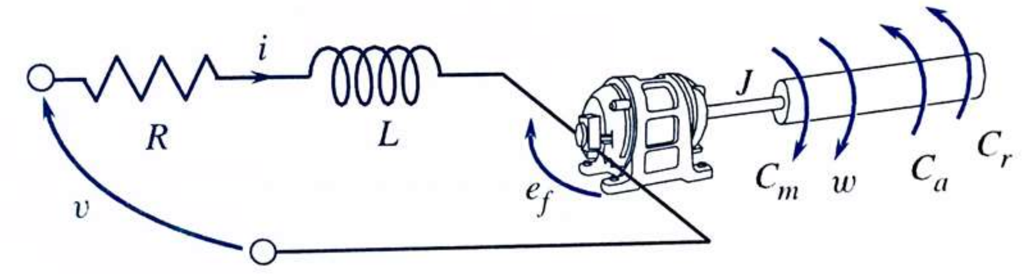
R and L are the circuit resistance and inductance

e_f is the so called counter-electromotive force. It is such that: $e_f = K_t\omega$ where ω is the motor rotating speed

$C_m = K_t i$ is the motor torque

$C_a = h\omega$ is the torque due to friction

C_r is the resisting torque



Example 4

For the sake of semplicity, let us make a change of variables. Choose:

$$x_1 = y_1 = i$$
$$x_2 = y_2 = \omega$$

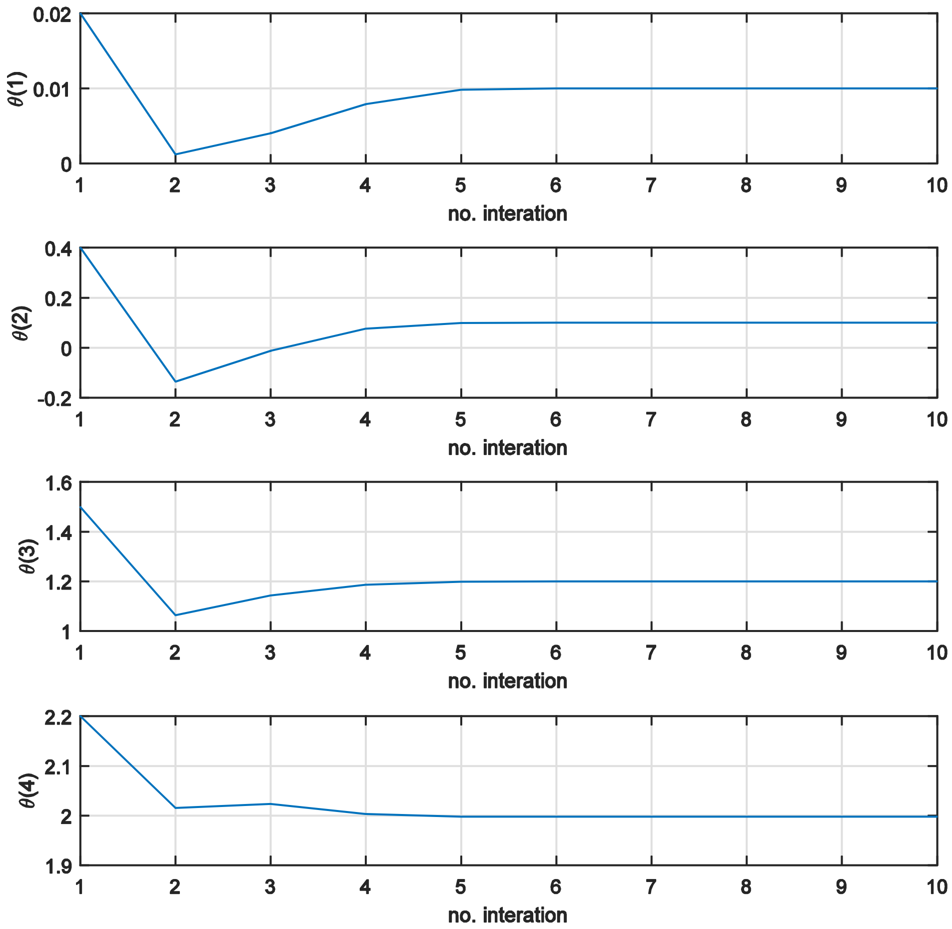
as state variables and:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_t}{L} \\ K_t & h \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} [$$

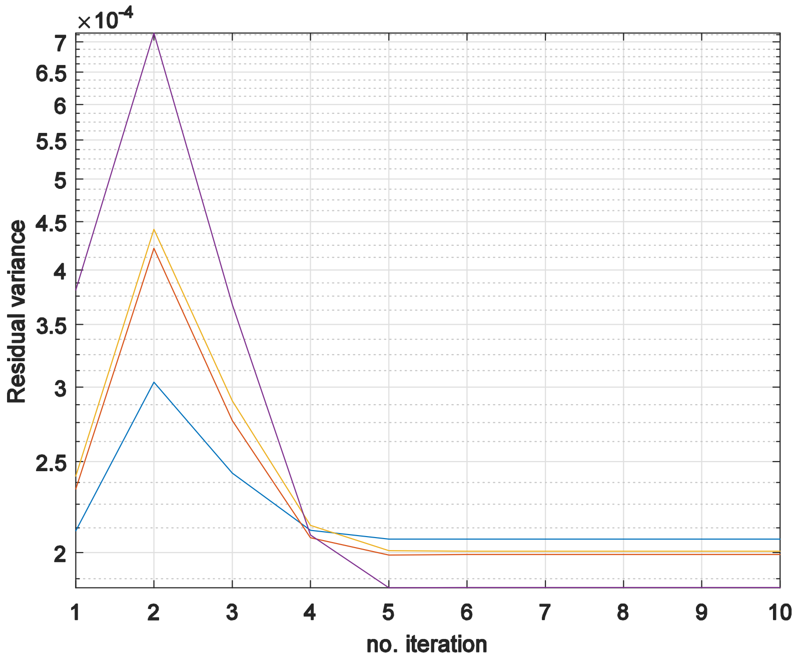
$$\boldsymbol{\theta} = [J \quad h \quad K_t$$

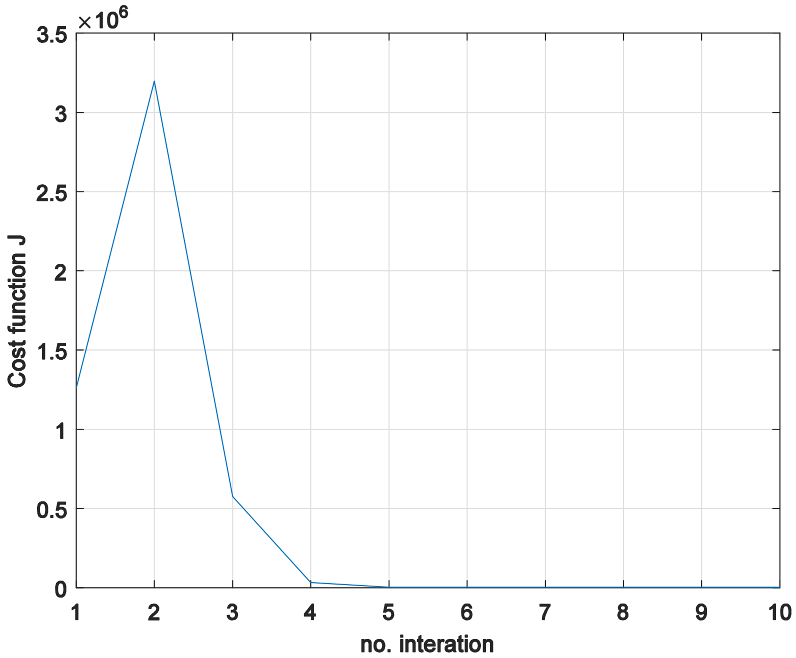
Example 4



θ(1)
0.02
0.01
0
θ(2)
0.4
0.2
0
-0.2
θ(3)
1.6
1.4
1.2
1
θ(4)
2.2
2.1
2
1.9
1 2 3 4 5 6 7 8 9 10
no. interation

Example 4


×10^-4
Residual variance
7
6.5
6
5.5
5
4.5
4
3.5
3
2.5
2
1 2 3 4 5 6 7 8 9 10
no. iteration


×10^6
Cost function J
3.5
3
2.5
2
1.5
1
0.5
0
1 2 3 4 5 6 7 8 9 10
no. interation

Example 4

Parameter	Real Value	Estimated Value	$s(\theta)$	$\frac{100s(\theta)}{\theta}$
J	0.01000	0.01002	0.00003	0.26676
h	0.10000	0.10035	0.00047	0.46413
K_t	1.20000	1.19894	0.00082	0.06824
R	2.00000	1.99776	0.00172	0.08630

- No. of iterations: 10