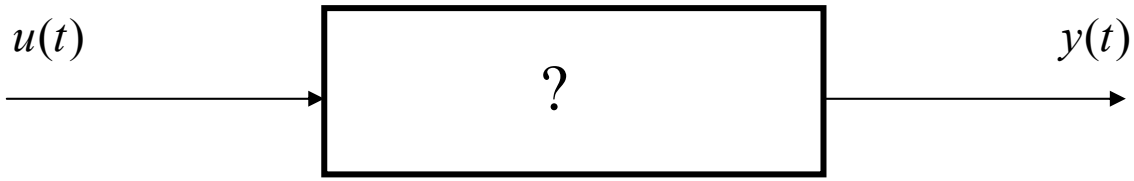
System Identification

System Identification



Difference Equations

Many systems can be described by differential equations. When a differential equation takes a discrete form, it generates a difference equation:

$$\frac{dy(t)}{dt} + ay(t) = bu(t) \qquad u(t) - input \qquad y(t) - output$$

In discrete cases

$$\frac{dy(t)}{dt} = \frac{y(t) - y(t-T)}{T}$$

t becomes discrete time, $T = 1$

$$\frac{dy(t)}{dt} = y(t) - y(t-1)$$

then $y(t) - y(t-1) + ay(t) = bu(t)$

or

$$(1+a)y(t) - y(t-1) = bu(t)$$

or

$$y(t) + a_1 y(t-1) = b_1 u(t)$$

Difference Equations

For 2nd order differential equation

$$\frac{d^2y(t)}{dt^2} + a_1\frac{dy(t)}{dt} + a_2 y(t) = bu(t)$$

$$y(t) + \tilde{a}_1 y(t-1) + \tilde{a}_2 y(t-2) = b\,u(t)$$

General cases: Linear difference equation

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = b_1 u(t-1) + \ldots + b_m u(t-m)$$

or

$$y(t) = -a_1 y(t-1) - \ldots - a_n y(t-n) + b_1 u(t-1) + \ldots + b_m u(t-m)$$

System Identification: the least squares method

Simple example (downsized case)

$$y(t) + a\ y(t-1) = b\ u(t-1)$$

We have measured samples

$$u(0), u(1), \ldots, u(N-1)$$

$$y(0), y(1), \ldots, y(N)$$

We use estimate

$$\hat{y}(t) = -ay(t-1) + b\ u(t-1)$$

The estimate error

$$V = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \hat{y}(t)\right)^2 = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) + ay(t-1) - b\ u(t-1)\right)^2$$

Let

$$\frac{\partial V}{\partial a} = \frac{1}{N}\sum_{t=1}^{N} 2y(t-1)\left(y(t) + ay(t-1) - b\ u(t-1)\right) = 0$$

$$\frac{\partial V}{\partial b} = \frac{1}{N}\sum_{t=1}^{N} -2u(t-1)\left(y(t) + ay(t-1) - b\ u(t-1)\right) = 0$$

System Identification: the least squares method

i.e

$$a\sum_{t=1}^{N} y^2(t-1) \; -b\sum_{t=1}^{N} y(t-1)u(t-1) = -\sum_{t=1}^{N} y(t-1)y(t)$$

$$-a\sum_{t=1}^{N} y(t-1)u(t-1) + b\sum_{t=1}^{N} u^2(t-1) = \sum_{t=1}^{N} u(t-1)y(t)$$

i.e

$$\begin{bmatrix} \sum_{t=1}^{N} y^2(t-1) & -\sum_{t=1}^{N} y(t-1)u(t-1) \\ -\sum_{t=1}^{N} y(t-1)u(t-1) & \sum_{t=1}^{N} u^2(t-1) \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix} = \begin{Bmatrix} -\sum_{t=1}^{N} y(t-1)y(t) \\ \sum_{t=1}^{N} u(t-1)y(t) \end{Bmatrix}$$

i.e

$$\begin{Bmatrix} a \\ b \end{Bmatrix} = \begin{bmatrix} \sum_{t=1}^{N} y^2(t-1) & -\sum_{t=1}^{N} y(t-1)u(t-1) \\ -\sum_{t=1}^{N} y(t-1)u(t-1) & \sum_{t=1}^{N} u^2(t-1) \end{bmatrix}^{-1} \begin{Bmatrix} -\sum_{t=1}^{N} y(t-1)y(t) \\ \sum_{t=1}^{N} u(t-1)y(t) \end{Bmatrix}$$

System Identification: the least squares method

Alternative expression

$$y(t) = -a\ y(t-1) + b\ u(t-1)$$

$$\theta = \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a & b \end{bmatrix}^T \qquad -\text{parameters}$$

$$\varphi(t) = \begin{bmatrix} -y(t-1) \\ u(t-1) \end{bmatrix} = \begin{bmatrix} -y(t-1) & u(t-1) \end{bmatrix}^T$$

Therefore

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

Calculated value of output depends on parameters θ

$$\hat{y}(t|\theta) = \varphi^T(t)\theta$$

For measured samples

$$u(0), u(1), \ldots, u(N-1)$$

$$y(0), y(1), \ldots, y(N)$$

The error defined as

$$V_N(\theta) = \frac{1}{\mathrm{N}} \sum_{\mathrm{t=1}}^{\mathrm{N}} (y(t) - \hat{y})^2 = \frac{1}{\mathrm{N}} \sum_{\mathrm{t=1}}^{\mathrm{N}} (y(t) - \varphi^T(t)\theta)^2$$

System Identification: the least squares method

$$\frac{\partial V_N(\theta)}{\partial \theta} = \frac{1}{\mathrm{N}} \sum_{\mathrm{t=1}}^{\mathrm{N}} \varphi(t)(y(t) - \varphi^T(t)\theta) = 0$$

$$\sum_{\mathrm{t=1}}^{\mathrm{N}} \varphi(t) y(t) - \sum_{\mathrm{t=1}}^{\mathrm{N}} \varphi(t)\varphi^T(t)\theta = 0$$

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

where

$$\sum_{\mathrm{t=1}}^{\mathrm{N}} \varphi(t)\varphi^T(t) = \sum_{\mathrm{t=1}}^{\mathrm{N}} \begin{bmatrix} -y(t-1) \\ u(t-1) \end{bmatrix} [-y(t-1)\ \ u(t-1)] = \begin{bmatrix} \sum_{t=1}^{N} y^2(t-1) & -\sum_{t=1}^{N} y(t-1)u(t-1) \\ -\sum_{t=1}^{N} y(t-1)u(t-1) & \sum_{t=1}^{N} u^2(t-1) \end{bmatrix}$$

$$\sum_{\mathrm{t=1}}^{\mathrm{N}} \varphi(t) y(t) = \sum_{\mathrm{t=1}}^{\mathrm{N}} \begin{bmatrix} -y(t-1) \\ u(t-1) \end{bmatrix} y(t) = \begin{bmatrix} \sum_{t=1}^{N} -y(t-1)y(t) \\ \sum_{t=1}^{N} u(t-1)y(t) \end{bmatrix}$$

System Identification: the least squares method

$$y(t) = -a_1 y(t-1) - \ldots - a_n y(t-n) + b_1 u(t-1) + \ldots + b_m u(t-m)$$

$$\theta = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_m \end{bmatrix} = \left[a_1 \ldots a_n b_1 \ldots b_m \right]^T \qquad - \text{parameters}$$

$$\varphi(t) = \begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n) \\ u(t-1) \\ \vdots \\ u(t-m) \end{bmatrix} = \left[-y(t-1) \ldots -y(t-n) \;\; u(t-1) \ldots u(t-m) \right]^T$$

Therefore

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

Calculated value of output depends on parameters θ

$$\hat{y}(t|\theta) = \varphi^T(t)\theta$$

System Identification: the least squares method

Suppose the parameters

θ is unknown

the record of input and output are known

$u(1),\ldots,u(N)$

$y(1),\ldots,y(N)$

define

$$Z^N = \{u(1), y(1),\ldots,u(N), y(N)\}$$

Least square method:

$$\min_{\theta} V_N(\theta, Z^N)$$

where

$$V_N(\theta, Z^N) = \frac{1}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}(y(t)-\hat{y})^2 = \frac{1}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}(y(t)-\varphi^T(t)\theta)^2$$

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta, Z^N)$$

i.e. the estimated parameters $\hat{\theta}_N$ is the value of θ which minimizes V_N

System Identification: the least squares method

To find the minimum value of $V_N(\theta, Z^N)$,

Let

$$\frac{d}{d\theta}V_N(\theta, Z^N) = \frac{2}{\mathrm{N}}\sum_{\mathrm{t=1}}^{\mathrm{N}}\varphi(t)(y(t)-\varphi^T(t)\theta) = 0$$

Gives

$$\sum_{\mathrm{t=1}}^{\mathrm{N}}\varphi(t)y(t) = \sum_{t=1}^{N}\varphi(t)\varphi^T(t)\theta = 0$$

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

System Identification: the least squares method

First order difference equation

$$y(t)+ay(t-1)=bu(t-1)$$

The estimated parameters

$$\begin{bmatrix}\hat{a}_N\\ \hat{b}_N\end{bmatrix}=\begin{bmatrix}\sum y^2(t-1) & -\sum y(t-1)u(t-1)\\ -\sum y(t-1)u(t-1) & \sum u^2(t-1)\end{bmatrix}^{-1}\begin{bmatrix}-\sum y(t)y(t-1)\\ \sum y(t)u(t-1)\end{bmatrix}$$

where

$$\sum=\sum_{t=1}^{N}$$

For example: Let $N=5, t=1,...,5$

Input $u(0), u(1), u(2), u(3), u(4)$ are observed

Output $y(0), y(1), y(2), y(3), y(4), y(5)$ are observed

$$\sum y^2(t-1)=y^2(0)+y^2(1)+y^2(2)+y^2(3)+y^2(4)$$

$$\sum u^2(t-1)=u^2(0)+u^2(1)+u^2(2)+u^2(3)+u^2(4)$$

$$\sum y(t-1)u(t-1)=y(0)u(0)+y(1)u(1)+y(2)u(2)+y(3)u(3)+y(4)u(4)$$

$$\sum y(t)y(t-1)=y(1)y(0)+y(2)y(1)+y(3)y(2)+y(4)y(3)+y(5)y(4)$$

$$\sum y(t)u(t-1)=y(1)u(0)+y(2)u(1)+y(3)u(2)+y(4)u(3)+y(5)u(4)$$

Linear Regressions ARX

Linear regression model

$\hat{y}(t|\theta) = \varphi^T(t)\theta$ - is called linear regression model in statistics

where

$\varphi^T(t)$ – called regression vector, contains old values of $y(t)$

Regression means trying to calculate $y(t)$ by going back to $\varphi(t)$

$$y(t) + a_1 y(t-1) + \ldots + a_n y(t-n) = b_1 u(t-1) + \ldots + b_m u(t-m)$$

It is Auto Regression with eXtra inputs = ARX

Model Quality

Assume (finite impulse response) the model

$$y(t) = b_1 u(t-1) + \ldots + b_m u(t-m)$$

The observed data has noise

$$y(t) = b_1^0 u(t-1) + \ldots + b_m^0 u(t-m) + e(t)$$

where independent noise *e(t)* has zero mean and variance λ but otherwise unknown.

$$y(t) = \varphi^T(t)\theta_0 + e(t)$$

$$\hat{\theta}_N = \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right)^{-1} \sum_{t=1}^{N} \varphi(t) y(t) = \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right)^{-1} \sum_{t=1}^{N} \varphi(t)\left(\varphi^T(t)\theta_0 + e(t) \right)$$

$$= \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right)^{-1} \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\theta_0 + \sum_{t=1}^{N} \varphi(t)e(t) \right)$$

$$= \theta_0 + \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t) \right)^{-1} \sum_{t=1}^{N} \varphi(t)e(t)$$

Model Quality

Mathematical expectation:

$$y(t) = \varphi^T(t)\theta_0 + e(t)$$

$$E\left(\hat{\theta}_N - \theta_0\right) = E\left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right)^{-1} \sum_{t=1}^{N} \varphi(t)e(t) = \left(\sum_{t=1}^{N} \varphi(t)\varphi^T(t)\right)^{-1} \sum_{t=1}^{N} \varphi(t)Ee(t) = 0$$

The estimate is unbiased

Model Quality

The covariance matrix

$$E\left(\hat{\theta}_N-\theta_0\right)\left(\hat{\theta}_N-\theta_0\right)^T=E\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\sum_{t=1}^{N}\varphi(t)e(t)\left(\left(\sum_{t=1}^{N}\varphi(s)\varphi^T(s)\right)^{-1}\sum_{t=1}^{N}\varphi(s)e(s)\right)^T$$

$$=E\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\sum_{t=1}^{N}\varphi(t)\varphi^T(s)\left(\sum_{t=1}^{N}\varphi(s)\varphi^T(s)\right)^{-1}e(t)e(s)$$

$$=\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\sum_{t=1}^{N}\varphi(t)\varphi^T(s)\left(\sum_{t=1}^{N}\varphi(s)\varphi^T(s)\right)^{-1}Ee(t)e(s)$$

$$=\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\sum_{t=1}^{N}\varphi(t)\varphi^T(s)\left(\sum_{t=1}^{N}\varphi(s)\varphi^T(s)\right)^{-1}\lambda\delta(t-s)$$

$$=\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}\lambda\sum_{t=1}^{N}\varphi(t)\varphi^T(s)\left(\sum_{t=1}^{N}\varphi(s)\varphi^T(s)\right)^{-1}$$

$$=\lambda\left(\sum_{t=1}^{N}\varphi(t)\varphi^T(t)\right)^{-1}$$

i.e. the covariance depends on input property and noise level.

Time-Invariant Linear Systems

$$y(t) = \int_0^\infty g(\tau)u(t-\tau)d\tau$$



Discret form

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k)$$

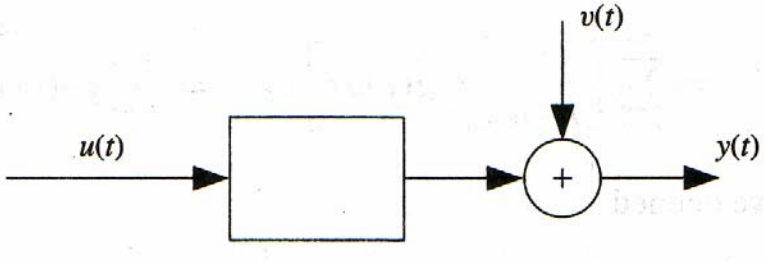
with disturbance

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + v(t)$$

Let

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k)$$

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + \sum_{k=0}^{\infty} h(k)e(t-k)$$



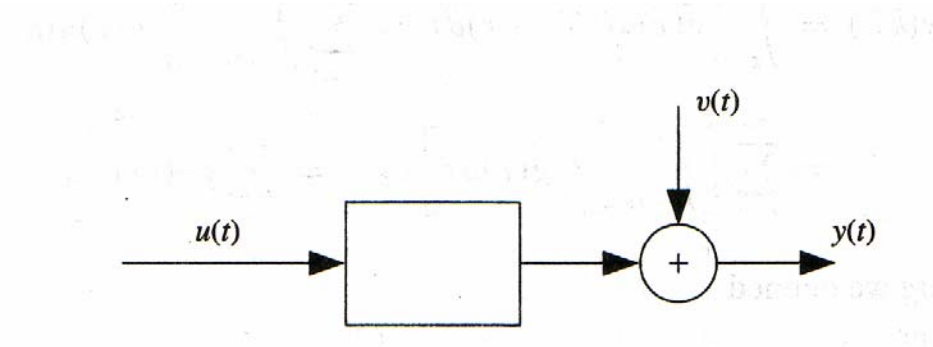
Let transfer operator

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k}, \qquad H(q) = \sum_{k=0}^{\infty} h(k)q^{-k} = 1 + \sum_{k=1}^{\infty} h(k)q^{-k}$$

$$y(t) = G(q)u(t) + H(q)e(t)$$

Disturbances

$$v(t) = \sum_{k=1}^{\infty} h(k)e(t-k)$$



Assume Gaussian

$e(t) \in N(0, \lambda)$

$$Ev(t) = E\sum_{k=0}^{\infty} h(k)e(t-k) = \sum_{k=0}^{\infty} h(k)Ee(t-k) = 0$$

Convariance function of the process $v(t)$:

$$R_v(\tau) = Ev(t)v(t-\tau) = \sum_{k=0}^{\infty}\sum_{s=0}^{\infty} h(k)h(s)Ee(t-k)e(t-\tau-s)$$

$$= \sum_{k=0}^{\infty}\sum_{s=0}^{\infty} h(k)h(s)\delta(k-\tau-s)\lambda$$

$$= \lambda\sum_{k=0}^{\infty} h(k)h(k-\tau)$$

Continuous-time representation

$$Y(s) = G_c(s)U$$

$$y(t) = G_c(p)u(t)$$

p – diffrentiation operator

$$y(t) = G_c(p)u(t) + H(q)e(t)$$

$$G_c(p) \rightarrow G_T(q)$$

Euler approximation

$$G_T(q) \approx G_c\left(\frac{q-1}{T}\right)$$

Tustin's formula

$$G_T(q) \approx G_c\left(\frac{2}{T}\frac{q-1}{1+q}\right)$$

DFT and Inverse DFT

$u(t) \qquad t = 1,2,\ldots,N$

DFT

$$U_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{N} u(t) e^{-i\omega t}$$

Inverse DFT

$$u(t) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} U_N\left(\frac{2\pi k}{N}\right) e^{-i\frac{2\pi k t}{N}}$$

Parseval's Relationship

$$\sum_{k=1}^{N} \left| U_N\left(\frac{2\pi k}{N}\right) \right|^2 = \sum_{t=1}^{N} u^2(t)$$

Non-parametric Time-and Frequency-Domain Methods

$$y(t) = \sum_{t=1}^{\infty} g(k)u(t-k) + v(t)$$

or

$$y(t) = G_0(q)u(t) + v(t)$$

if u(t) is an impulse

$$u(t) = \begin{cases} \alpha & t = 0 \\ 0 & t \neq 0 \end{cases}$$

$$y(t) = \alpha g_0(t) + v(t)$$

where

$g_0(t) - impulse\ response$

Estimate of impulse response : (when noise is low)

$$\hat{g}(t) = \frac{y(t)}{\alpha}$$

Estimate of step *response* : (when noise is low)

$$u(t) = \begin{cases} \alpha & t \geq 0 \\ 0 & t < 0 \end{cases}$$

$$\hat{g}(t) = \frac{y(t) - y(t-1)}{\alpha}$$

Non-parametric Time-and Frequency-Domain Methods

$$y(t) = \sum_{t=1}^{\infty} g(k)u(t-k) + v(t)$$

or

$$y(t) = G_0(q)u(t) + v(t)$$

if u(t) is an impulse

$$u(t) = \begin{cases} \alpha & t = 0 \\ 0 & t \neq 0 \end{cases}$$

1. If input is chosen as white noise

$$\mathrm{R}_u(\tau) = \alpha\delta_{\tau 0}$$

$$g_0(\tau) = \frac{R_{yu}(\tau)}{\alpha}$$

$$\hat{R}_{yu}^N(\tau) = \frac{1}{N}\sum_{t=\tau}^{N} y(t)u(t-\tau)$$

2. If input is not white noise

$$\hat{R}_u^N(\tau) = \frac{1}{N}\sum_{t=\tau}^{N} u(t)u(t-\tau)$$

solve $\hat{R}_{yu}^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} \hat{g}(k)\hat{R}_u^N(k-\tau)$ *for* $g(k)$

Non-parametric Time-and Frequency-Domain Methods

$$U_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{\infty} u(t) e^{-i\omega t}$$

$$Y_N(\omega) = \frac{1}{\sqrt{N}} \sum_{t=1}^{\infty} y(t) e^{-i\omega t}$$

Frequency response estimate

$$\hat{G}(\omega) = \frac{Y_N(\omega)}{U_N(\omega)}$$

Spectra

Signal $s(t)$ $\quad t \in [1, \infty]$

Covariance function

$$R_s(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} s(t)s(t-\tau)$$

Defination of Spectrum

$$\Phi_s(\omega) = \sum_{\tau=-\infty}^{\infty} R_s(\tau) e^{-i\tau\omega}$$

Spectrum of disturbance

$$\Phi_v(\omega) = \sum_{\tau=-\infty}^{\infty} R_v(\tau) e^{-i\tau\omega} = \sum_{\tau=-\infty}^{\infty} \lambda e^{-i\tau\omega} \sum_{k=0}^{\infty} h(k)h(k-\tau)$$

$$= \lambda \sum_{\tau=-\infty}^{\infty} \sum_{k=0}^{\infty} h(k)e^{-ik\omega} h(k-\tau)e^{-i(k-\tau)\omega} = \lambda \sum_{s=0}^{\infty} h(s)e^{-is\omega} \sum_{k=0}^{\infty} h(k-\tau)e^{-ik\omega} = \lambda \left| H(e^{i\omega}) \right|^2$$

Spectrum of output

$$\Phi_y(\omega) = \left| G(e^{i\omega}) \right|^2 \Phi_u(\omega) + \lambda \left| H(e^{i\omega}) \right|^2$$

ARX model

$$y(t)+a_1 y(t-1)+\ldots+a_{n_a} y(t-n_a)=b_1 u(t-1)+\ldots+b_{n_b} u(t-n_a)+e(t)$$

Let

$$A(q)=1+a_1 q^{-1}+\ldots+a_{n_a} q^{-n_a}$$

$$B(q)=b_1 q^{-1}+\ldots+b_{n_b} q^{-n_b}$$

$$A(q)y(t)=B(q)u(t)+e(t)$$

ARMAX model

$$y(t)+a_1y(t-1)+\ldots+a_{n_a}y(t-n_a)=b_1u(t-1)+\ldots+b_{n_b}u(t-n_a)$$
$$+e(t)+c_1e(t-1)+\ldots+c_{n_c}e(t-n_c)$$

Let

$$A(q)=1+a_1q^{-1}+\ldots+a_{n_a}q^{-n_a}$$

$$B(q)=b_1q^{-1}+\ldots+b_{n_b}q^{-n_b}$$

$$C(q)=c_1q^{-1}+\ldots+c_{n_c}q^{-n_c}$$

$$A(q)y(t)=B(q)u(t)+C(q)e(t)$$

Let

Let

$$G(q)=\frac{B(q)}{A(q)},\qquad H(q)=\frac{C(q)}{A(q)}$$

$$y(t)=G(q)u(t)+H(q)e(t)$$

Assume the mean and variance, respectively

$$Ee(t)=0$$

$$Ee^2(t)=\lambda$$

ARMAX model

The filter G(q) is stable if

$$\sum_{k=1}^{\infty} |g(k)| < \infty$$

Simulation

$$y(t) = G(q)u(t) + H(q)e(t)$$

No noise case

$$y^*(t) = G(q)u^*(t)$$

Simulation is to produce

$$e^*(t) \qquad t = 1,2,\ldots,N$$
$$v^*(t) = H(q)e^*(t)$$

to present realization of $y^*(t)$

Prediction

$$y(t) = G(q)u(t) + H(q)e(t)$$

Noise prediction

$$v(t) = \sum_{k=0}^{\infty} h(k)e(t-k) = e(t) + \sum_{k=1}^{\infty} h(k)e(t-k)$$

$v(t)$ is estimatedat $t-1$

$$\hat{v}(t|t-1) = \sum_{k=1}^{\infty} h(k)e(t-k) = \sum_{k=1}^{\infty} h(k)q^{-k}e(t) = [H(q)-1]e(t) = [H(q)-1]\frac{v(t)}{H(q)} = [1-H^{-1}(q)]v(t)$$

Given $u(s)$ and $y(s)$, $s \le t-1$, one-step-ahead prediction of $y(t)$ is

$$\begin{aligned}
\hat{y}(t|t-1) &= G(q)u(t) + \hat{v}(t|t-1) \\
&= G(q)u(t) + [1-H^{-1}(q)]y(t) \\
&= G(q)u(t) + [1-H^{-1}(q)][y(t) - G(q)u(t)] \\
&= G(q)u(t) + [1-H^{-1}(q)]y(t) - [1-H^{-1}(q)]G(q)u(t) \\
&= H^{-1}(q)G(q)u(t) + [1-H^{-1}(q)]y(t)
\end{aligned}$$

i.e.

$$\hat{y}(t|t-1) = H^{-1}(q)G(q)u(t) + [1-H^{-1}(q)]y(t)$$

or

$$H(q)\hat{y}(t|t-1) = G(q)u(t) + [H(q)-1]y(t)$$

Linear Models and Sets of Linear Models

Linear model

$$y(t) = G(q)u(t) + H(q)e(t)$$

A set of linear models

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t)$$

where

θ – parameters to be determined

$\theta \in D_M \subset R^d$ (Subset of R^d)

One –step-prediction

$$\hat{y}(t|\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + \left[1 - H^{-1}(q,\theta)\right]y(t)$$

ARX model

$$A(q)y(t) = B(q)u(t) + e(t)$$

$$G(q,\theta) = \frac{B(q)}{A(q)}, H(q,\theta) = \frac{1}{A(q)}$$

$$\hat{y}(t|\theta) = B(q)u(t) + \left[1 - A(q)\right]y(t)$$

Linear Models and Sets of Linear Models

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t)$$

Let

$$\theta = [a_1 \ldots a_{n_a} b_1 \ldots b_{n_b}]$$

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

$$\hat{y}(t|\theta) = [1 - A(q)]y(t) + B(q)u(t) = \varphi^T(t)\theta$$

i.e. linear regression

$$\hat{y}(t|\theta) = \varphi^T(t)\theta$$

Linear Models and Sets of Linear Models

ARMAX model pseudo-linear regression

$$A(q)y(t) = B(q)u(t) + C(q)e(t)$$

$$A(q) = 1 + a_1 q^{=1} + \ldots + a_{n_a} q^{=n_a}$$

$$B(q) = b_1 q^{=1} + \ldots + b_{n_b} q^{=n_b}$$

$$C(q) = 1 + c_1 q^{=1} + \ldots + c_{n_c} q^{=n_c}$$

$$G(q,\theta) = \frac{B(q)}{A(q)}, H(q,\theta) = \frac{C(q)}{A(q)}$$

Let

$$\theta = [a_1 \ldots a_{n_a} b_1 \ldots b_{n_b} c_1 \ldots c_{n_c}]$$

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)}u(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t)$$

or

$$C(q)\hat{y}(t|\theta) = B(q)u(t) + [C(q) - A(q)]y(t)$$

Linear Models and Sets of Linear Models

$$C(q)\hat{y}(t|\theta) = B(q)u(t) + [C(q) - A(q)]y(t)$$

Adding
$[1 - C(q)]\hat{y}(t|\theta)$

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t) + [C(q) - 1][y(t) - \hat{y}(t|\theta)]$$

Introduce prediction error
$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|\theta)$$

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t) + [C(q) - 1]\varepsilon(t,\theta)$$

Let

$$\theta = [a_1 \ldots a_{n_a} b_1 \ldots b_{n_b} c_1 \ldots c_{n_c}]$$

$$\varphi(t) = [-y(t-1) \ldots - y(t-n_a)\ \ u(t-1) \ldots u(t-n_b)\ \ \varepsilon(t-1,\theta) \ldots \varepsilon(t-n_c,\theta)]$$

i.e. pseudo - linear regression

$$\hat{y}(t|\theta) = \varphi^T(t,\theta)\theta$$

General Family of Model Structures

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t)$$

The predictor

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)F(q)}u(t) + \left[1 - \frac{D(q)A(q)}{C(q)}\right]y(t)$$

B	FIR (finite impulse response)
AB	ARX
ABC	ARMAX
AC	ARMA
ABD	ARARX
$ABCD$	ARARMAX
BF	OE (output error)
$BFCD$	BJ (Box-Jenkins)

State Space Models

$$\dot{x}(t) = F(\theta)x(t) + G(\theta)u(t)$$

$$\dot{x}(t)_{n\times 1} = F(\theta)_{n\times n}x(t)_{n\times 1} + G(\theta)_{n\times m}u(t)_{m\times 1}$$

Let p be the differentiation operator

$$[pI - F(\theta)]x(t) = G(\theta)u(t)$$

Let noise-free observation $\eta(t) = Hx(t)$

$$x(t) = [pI - F(\theta)]^{-1}G(\theta)u(t)$$

i.e.

$$\eta(t) = H[pI - F(\theta)]^{-1}G(\theta)u(t)$$

$$\eta(t) = G_c(p,\theta)u(t)$$

where

$$G_c(p,\theta) = H[pI - F(\theta)]^{-1}G(\theta)$$

$$x(kT+T) = A_T(\theta)x(kT) + B_T(\theta)u(kT)$$

$$\eta(kT) = H[qI - A_T(\theta)]^{-1}B_T(\theta)u(kT)$$

$$y(kT) = Hx(kT) + v_T(kT)$$

State Space Models

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t)$$
$$y(t) = C(\theta)x(t) + v(t)$$

Corresponding to

$$y(t) = G(q,\theta)u(t) + v(t)$$
$$G(q,\theta) = C(\theta)\left[qI - A(\theta)\right]^{-1}B(\theta)$$

Kalman Filter

Noise $v(t) = H(q,\theta)e(t), \qquad w(t)$

Assume

$$Ew(t)w^T(t) = R_1(\theta), \qquad Ev(t)v^T(t) = R_2(\theta), \qquad Ew(t)v^T(t) = R_{12}(\theta),$$

$$x(t+1) = A(\theta)x(t) + B(\theta)u(t) + w(t)$$

$$y(t) = C(\theta)x(t) + v(t)$$

$y(t)$ is expected as

$$\hat{x}(t+1,\theta) = A(\theta)\hat{x}(t,\theta) + B(\theta)u(t) + K(\theta)\left[y(t) - C(\theta)\hat{x}(t,\theta)\right]$$

$$\hat{y}(t|\theta) = C(\theta)\hat{x}(t,\theta)$$

where

$$K(\theta) = \left[A(\theta)\overline{P}(\theta)C^T(\theta) + R_{12}(\theta)\right]\cdot\left[C(\theta)\overline{P}(\theta)C^T(\theta) + R_2(\theta)\right]^{-1}$$

$\overline{P}(\theta)$is the solution of Riccati equation

$$\overline{P}(\theta) = A(\theta)\overline{P}(\theta)A^T(\theta) + R_1(\theta)$$
$$-\left[A(\theta)\overline{P}(\theta)C^T(\theta) + R_{12}(\theta)\right]\cdot\left[C(\theta)\overline{P}(\theta)C^T(\theta) + R_2(\theta)\right]^{-1}\left[A(\theta)\overline{P}(\theta)C^T(\theta) + R_{12}(\theta)\right]^T$$

$$\hat{y}(t|\theta) = C(\theta)[qI - A(\theta) + K(\theta)C(\theta)]^{-1}B(\theta)u(t) + C(\theta)[qI - A(\theta) + K(\theta)C(\theta)]^{-1}K(\theta)y(t)$$

$$\overline{P}(\theta) = \overline{E}\left[x(t) - \hat{x}(t,\theta)\right]\left[x(t) - \hat{x}(t,\theta)\right]^T$$

Parameter Estimation Methods

The least-squares method

The maximum likelihood method

Instrumental-variable methods

Frequency domain data fitting

Numerical solution by iterative search methods

Recursive estimation methods

Practical considerations

The goal of identification is to obtain good and reliable with reasonable amount of work

Experiment design: which signal to measure, how to condition the signals before sampling

Pre-process data: remove drifts and trends

Pre-filtering the input and output data though the same filter will not change the input-output relation for a linear system.

Model Validation

Validation- to test whether a model is appropriate
1. to choose type of model set
2. to choose the size of model set
3. to choose the model parametrization

Validation questions:
Dose the model agree sufficiently well with the observed data?
Is the model good enough fo the propose?
Does the model describe the true system?

Quality of a model can be measured by a mean square error criterion

Residual analysis

$$\varepsilon(t) = \varepsilon(t, \hat{\theta}_N) = y(t) - y\left(t \middle| \hat{\theta}_N\right)$$

Software and Examples

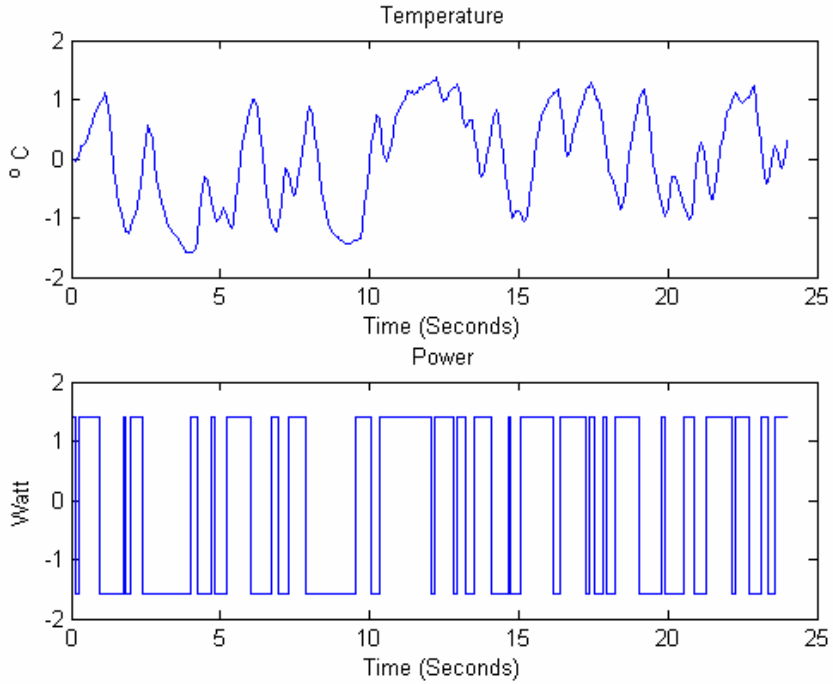
MATLAB toolbox: system identification

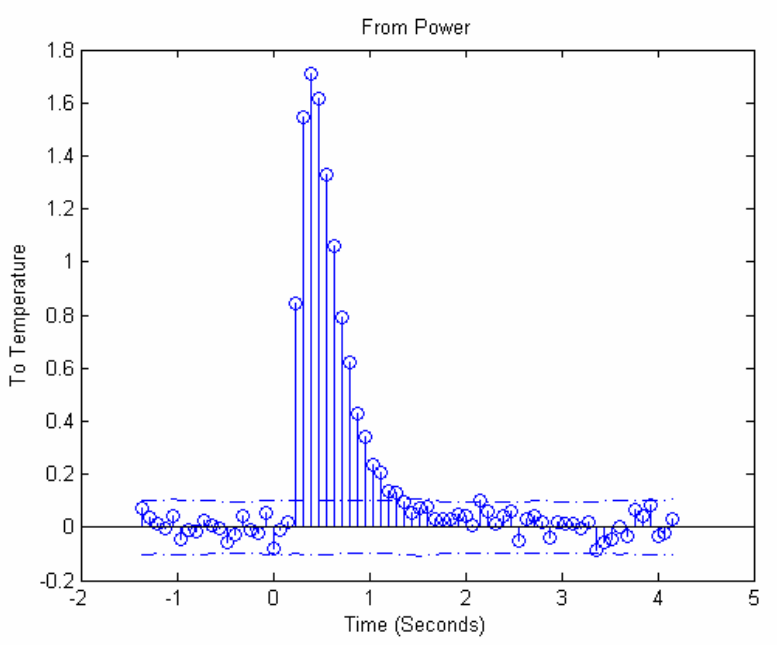
1. Hairdryer

2. A fighter aircraft

3. Buffer vessel dynamics

Hairdryer





Estimating Non-Parametric Model

impulse(ze,'sd',3); %impulse response
with 3 standard deviations confidence region

Estimating Parametric Model

To inspect the properties of the estimated model, just enter the model-name at the command windo

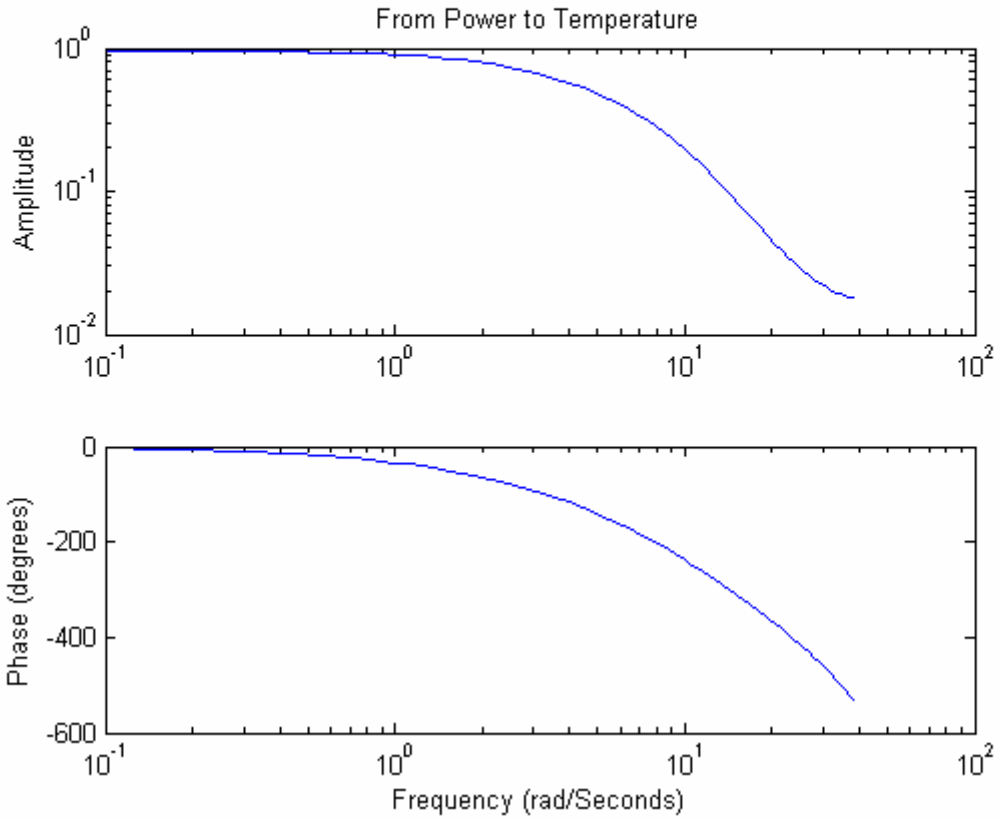
```
m1
```

```
State-space model:  x(t+Ts) = A x(t) + B u(t) + K e(t)
                       y(t) = C x(t) + D u(t) + e(t)

A =
                      x1          x2          x3
           x1      0.95423    -0.21174     0.06628
           x2      0.24784      0.6466     0.23888
           x3     -0.04934    -0.65945     0.13319


B =
                    Power
           x1   0.00023682
           x2     0.016849
           x3     0.051534


C =
                      x1          x2          x3
  Temperature     -14.059    0.094448    0.043373


D =
                    Power
  Temperature           0


K =
               Temperature
           x1     -0.06584
           x2    0.0090332
           x3      0.10811


x(0) =

           x1            0
           x2            0
```

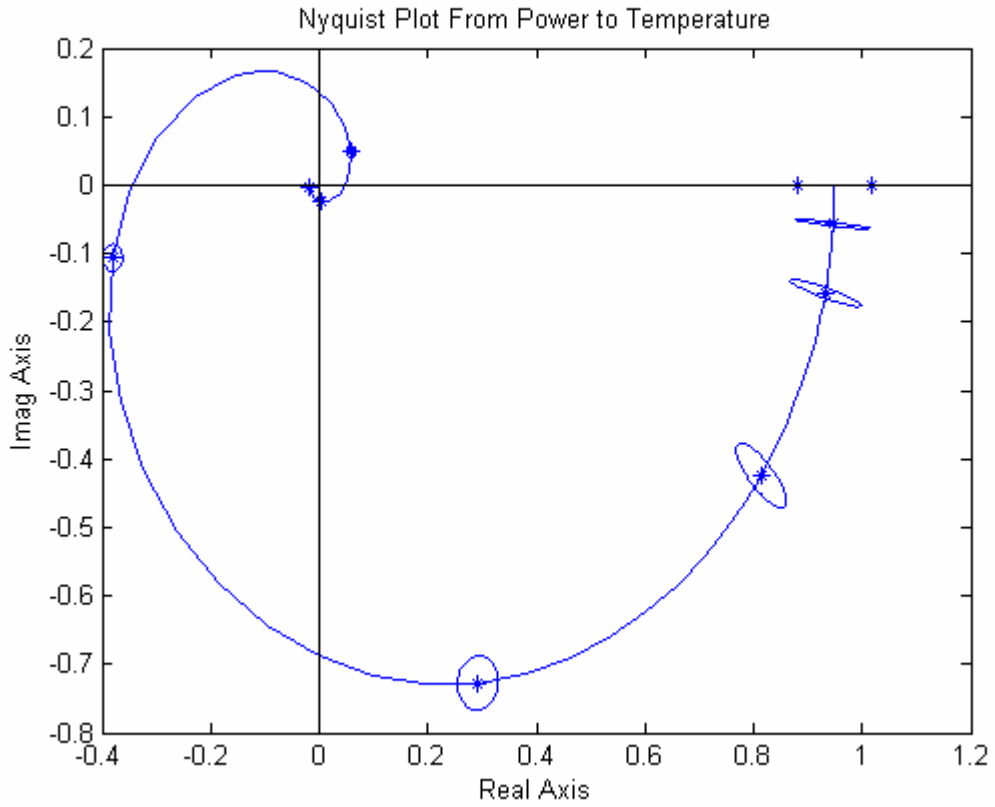
Analyzing the Estimated Model

The Bode plot of the generated model can be obtained using the bode function as shown below:

```
bode(m1)
```



```
nyquist(m1,'sd',3)
```



```
step(m1,'b',ze,'r');
```

```
Warning: Different TimeUnit in different systems. Time Scale not reliable.
```

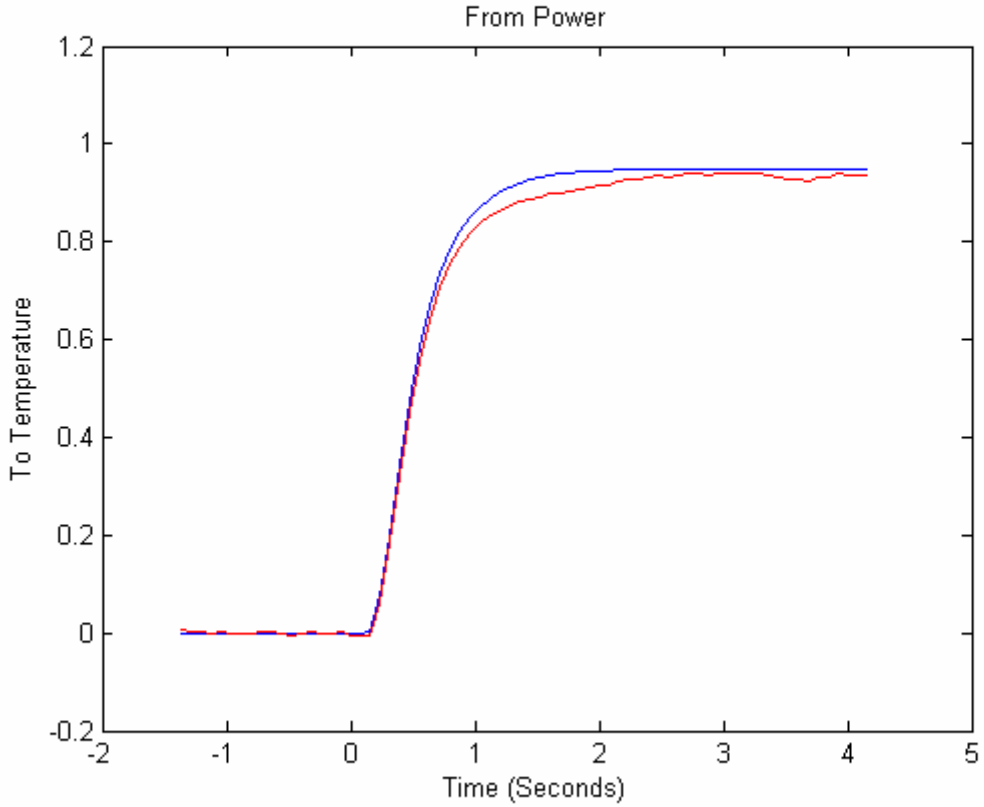
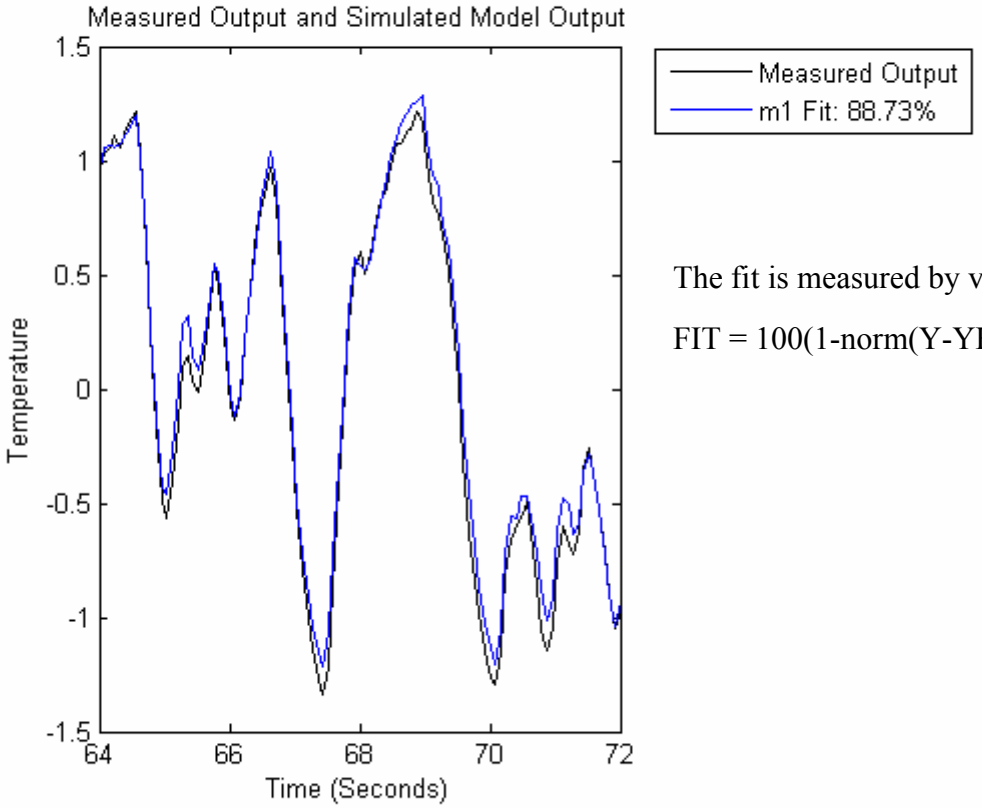


Figure 6: Step response of the estimated model compared against the non-parametric step response

```
zv = dry(800:900); % select an independent data set for validation
zv = detrend(zv); % preprocess the validation data
compare(zv,m1); % perform comparison of simulated output
```



The fit is measured by vector's norm.

FIT = 100(1-norm(Y-YHAT)/norm(Y-mean(Y))) (in %)

Figure 7: Simulation of model output compared against a validation data set.