2.3 System Identification: Linear methods

Prof. Davide Fissore
Politecnico di Torino, Italy
davide.fissore@polito.it

GRICU PhD School 2021

Digitalization Tools for the Chemical and Process Industries

March 12, 2021

Outline

1. Deterministic models vs. Empirical models
2. System identification: variables, regressors, function
3. ARX & ARMAX models
4. Box-Jenkins model
5. Identification procedure
6. Example

Deterministic models

Deterministic models are obtained from a deep knowledge of the process, and are based on **conservation laws**: mass, energy, momentum (force). They are also called “First principle based” models.

These models are usually expressed by means of differential equations, difference equations, algebraic equations and logical relations. Given the initial and the boundary conditions, it is possible to **predict** the evolution of the system in the future.

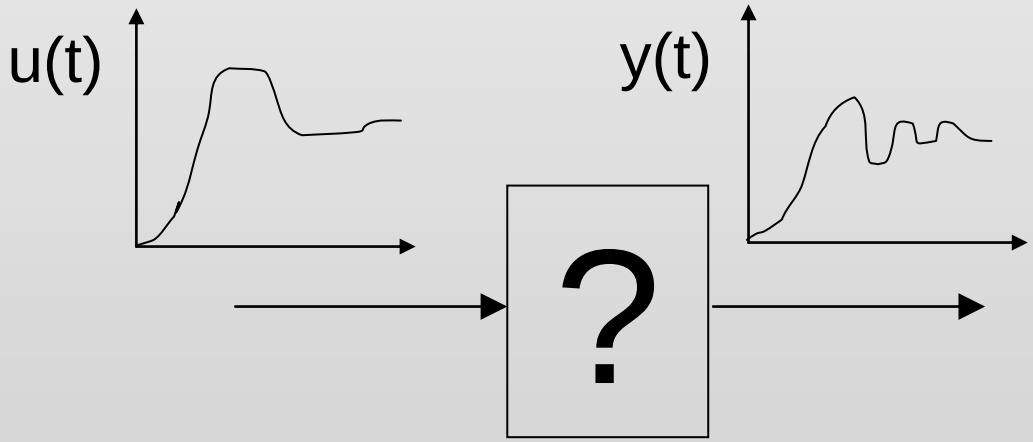
The resulting model is called **white box** or simulation model.

Moreover, it is possible to assess the **effect of the parameters** of the system (design variables) on some performance variables and thus, using this result, it is possible to **optimize** the process.

Empirical models

Empirical models do not consider the heat, mass, and momentum transfer processes occurring in a system, but they use **experiments** and **measurements** to get knowledge about the process.

In most cases **they consist of a mathematical relationship** (algebraic equations) **between the input and the output of the system**.



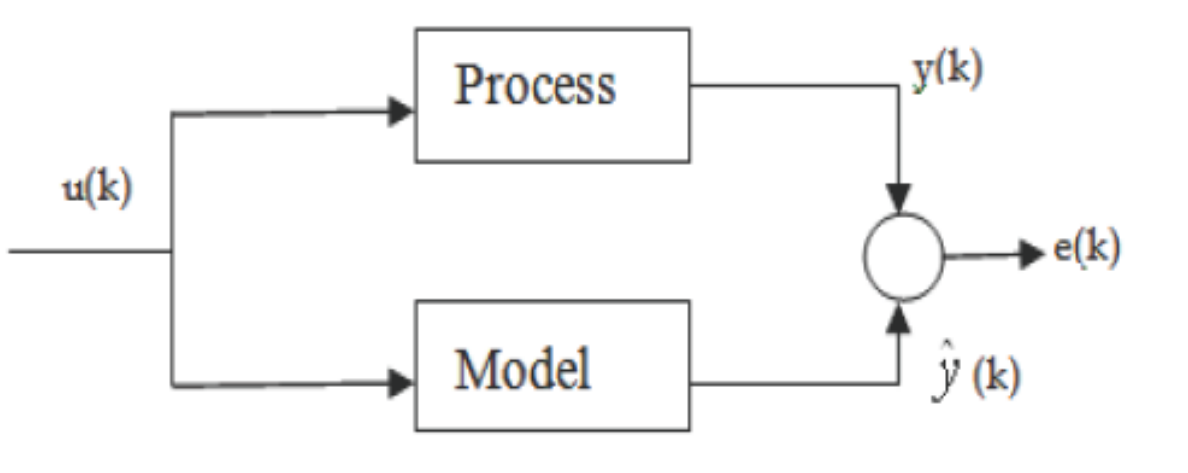
It can be even possible to build a **black box** model without having any specific knowledge about the process.

Empirical models

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

The process consists of one input signal ***u*** and one output signal ***y***. Here there two unknown parameters ***a*** and ***b***.

We want the model output to look like the process output as good as possible. The difference between the process and model outputs, the error ***e***, will be a measure to minimize to find the values of the parameters ***a*** and ***b***.



Empirical models

Black box models are useful in various cases:

- when there is **no information** about the process we would like to model
- when the process we are going to model is **too complex**
- when the time required for the calculations has to be **short**

System identification

System Identification (SI) is a methodology for building mathematical models of dynamic systems from experimental data, i.e., using measurements of the system input/output (IO) signals to estimate the values of adjustable parameters in a given model structure.

The process of SI requires some steps, such as

(i) measurement of the IO signals of the system,

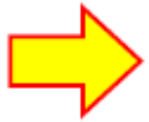
(ii) selection of a candidate model structure,

(iii) choice and application of a method to estimate the value of the adjustable parameters in the candidate model structure,

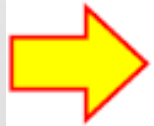
(iv) validation and evaluation of the estimated model to see if the model is right for the application needs, which should be done preferably with a different set of data.

Black box models

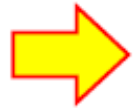
The structure of a black-box model is the following:

output = *f*(input)  $\mathbf{y} = \mathbf{f}(\mathbf{u})$

In case of a dynamic model, the output of the system is a function of the values of the input and of the output **in the past**:

output = *f*(old output, old input)  $\mathbf{y}_{now} = \mathbf{f}(\mathbf{y}_{old}, \mathbf{u}_{old})$

In order to determine the function ***f*** it is necessary to introduce the **adaptive parameters *p***:

output = *f*(old output, old input, parameters)  $\mathbf{y}_{now} = \mathbf{f}(\mathbf{y}_{old}, \mathbf{u}_{old}, \mathbf{p})$

e.g.

$$y(k) = 3y(k-1) + 2u(k-1) - 5u(k-2)$$

Black box models

A further improvement of the black-box model can be obtained if the errors ***e***, given by the difference between model calculations and experimental measurements (in the past), are considered:

output = *f*(old output, old input, old errors, parameters)

$$\mathbf{y}_{now} = \mathbf{f}\left(\mathbf{y}_{old}, \mathbf{u}_{old}, \mathbf{e}_{old}, \mathbf{p}\right)$$

e.g. $y(k) = 3y(k-1) + 2u(k-1) - 5u(k-2) + 7e(k-1)$

Black box models: variables

The model that has to be identified is thus:

$$\mathbf{y}(t) = \mathbf{f}[y_1(t-1),\ldots,y_1(t-n_{y_1}),\ldots,y_r(t-1),\ldots,y_r(t-n_{y_r}),$$
$$u_1(t-1),\ldots,u_1(t-n_{u_1}),\ldots,u_m(t-1),\ldots,u_m(t-n_{u_m}),$$
$$e_1(t-1),\ldots,e_1(t-n_{e_1}),\ldots,e_r(t-1),\ldots,e_r(t-n_{e_r})]$$

Black box models: variables

Let us define the array ϕ, whose components are named **regressors**:

$$\varphi(t) = [y_1(t-1), \ldots, y_1(t-n_{y_1}), \ldots, y_r(t-1), \ldots, y_r(t-n_{y_r}),$$
$$u_1(t-1), \ldots, u_1(t-n_{u_1}), \ldots, u_m(t-1), \ldots, u_m(t-n_{u_m}),$$
$$e_1(t-1), \ldots, e_1(t-n_{e_1}), \ldots, e_r(t-1), \ldots, e_r(t-n_{e_r})]^T$$

If *d* is the total number of variables of the system, then the size *l* of the array ϕ is defined as the **total order** of the model:

$$\ell = \sum_{i=1}^{d} n_i$$

Black box models: variables

Taking into account that the input variables can have a delayed effect on the output variables, it is possible to introduce the **delay times**, n_{ki}, one for each input variable, in the model of the process:

$$\varphi(t) = [y_1(t-1),\ldots,y_1\left(t-n_{y_1}\right),\ldots,y_r(t-1),\ldots,y_r\left(t-n_{y_r}\right),$$
$$u_1\left(t-n_{k_1}-1\right),\ldots,u_1\left(t-n_{k_1}-n_{u_1}\right),\ldots,u_m\left(t-n_{k_m}-1\right),\ldots,u_m\left(t-n_{k_m}-n_{u_m}\right),$$
$$e_1(t-1),\ldots,e_1\left(t-n_{e_1}\right),\ldots,e_r(t-1),\ldots,e_r\left(t-n_{e_r}\right)]^T$$

Black box models: function

The function ***f*** is able to map the array of the regressors ϕ into the output variables ***y***, using the parameters ***p***:

$$\mathbf{y}(t) = \mathbf{f}\left[\varphi(t), \mathbf{p}\right]$$

It is possible to use both **linear** and **non-linear** functions.

The simplest model for the function ***f*** is the following:

$$y(t) = \mathbf{p} \times \varphi(t)$$

(if ***p*** is a matrix, the array ***y*** is obtained).

ARX models

The **ARX model** is **linear** with respect to the regressors and to the parameters; ex. of ARX SISO:

$$y(t)+a_1 y(t-1)+a_2 y(t-2)+\ldots+a_{n_y} y(t-n_y)=b_1 u(t-1)+b_2 u(t-2)+\ldots+b_{n_u} u(t-n_u)$$

The ARX model is not able to describe those features that are characteristics of non-linear systems.

The time required to obtain a prediction is very short.

ARMAX models

The **ARMAX model** is **linear** with respect to the input and output variables, and to the errors; ex. of ARMAX SISO:

$$y(t)+a_1y(t-1)+a_2y(t-2)+\ldots+a_{n_y}y(t-n_y)=b_1u(t-1)+b_2u(t-2)+\ldots+b_{n_u}u(t-n_u)+$$
$$d_1e(t-1)+d_2e(t-2)+\ldots+d_{n_e}e(t-n_e)$$

Thanks to the presence of the error terms **e**, the accuracy of this model is expected to be higher than that of an ARX model.

Moreover, up to a certain extent, the error term is able to take into account process non-linearity, unmeasured disturbances, noisy measurements.

Box-Jenkins model

A linear system with additive disturbance $v(k)$ can be described as:

$$y(k) = G(z)u(k) + v(k)$$

where $G(z)$ is a transfer function

$$G(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \ldots + b_{n_b} z^{-n_b}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{n_a} z^{-n_a}}$$

Taking into account that $z^{-m}x(k) = x(k-m)$, the system can be described using a difference equation:

$$\begin{aligned} y(k) = &- a_1 y(k-1) - a_2 y(k-2) - \ldots - a_{n_a} y(k-n_a) \\ &+ b_0 u(k) + b_1 u(k-1) + \ldots + b_{n_b} u(k-n_b) + v(k) \end{aligned}$$

Box-Jenkins model

The disturbance $v(k)$ can be a white noise, or a colored noise:

$$v(k) = H(z)e(k)$$

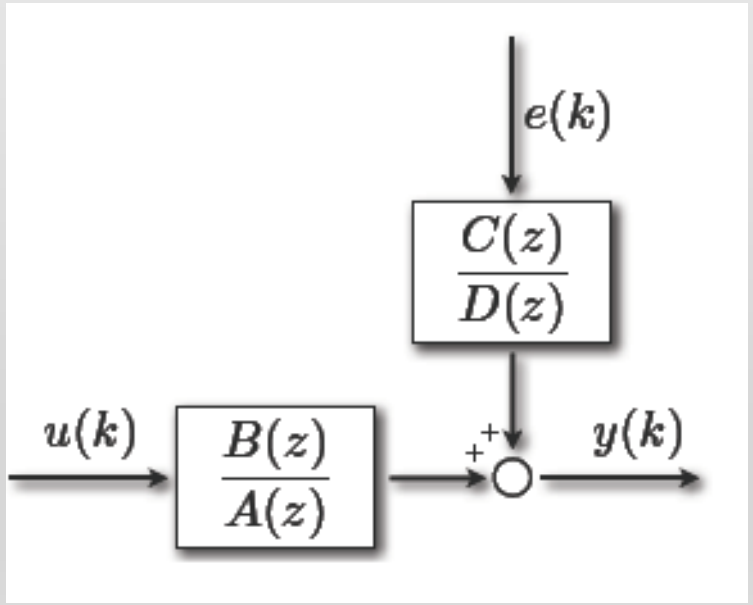
where $e(k)$ is a white noise and $H(z)$ is another transfer function:

$$H(z) = \frac{C(z)}{D(z)} = \frac{1 + c_1 z^{-1} + \ldots + c_{n_c} z^{-n_c}}{1 + d_1 z^{-1} + d_2 z^{-2} + \ldots + d_{n_d} z^{-n_d}}$$

Box-Jenkins model

The overall model is called **Box-Jenkins** (BJ) **model**:

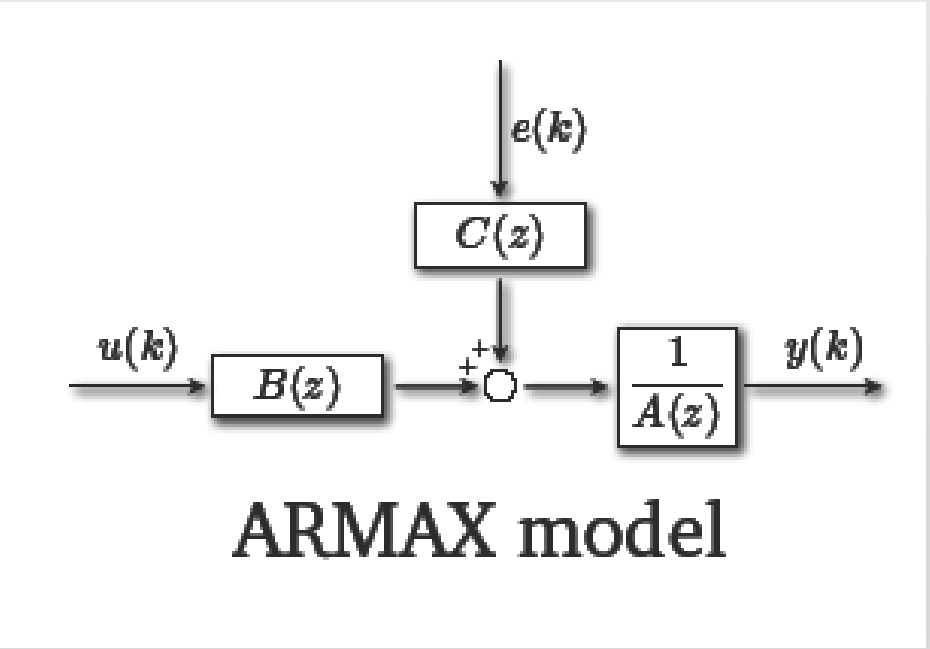
$$y(k) = \frac{B(z)}{A(z)}u(k) + \frac{C(z)}{D(z)}e(k)$$



Box-Jenkins model

In case $G(z)$ and $H(z)$ have the same denominator ($A(z) = D(z)$)

$$A(z)y(k) = B(z)u(k) + C(z)e(k)$$

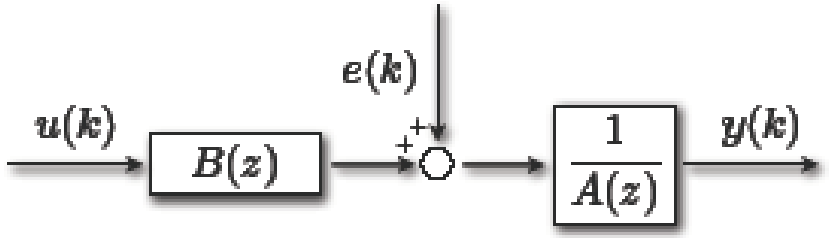


ARMAX model

Box-Jenkins model

In case $C(z) = I$ we get

$$A(z)y(k) = B(z)u(k) + e(k)$$



ARX model

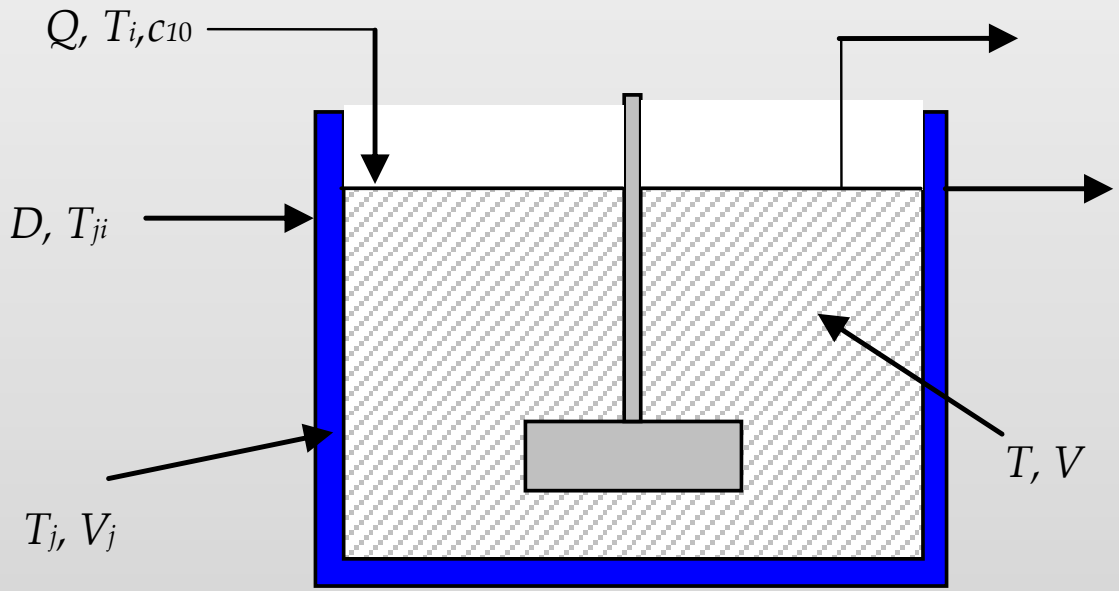
Identification procedure

1. Determination of the **input & output variables**, and of the **range of variability** of each variable.
2. Physical (or *in silico*) experiments: each input variables has to be “disturbed”, and the “answer” of the systems has to be measured (**learning set** and **cross-validation set**).

   → *Choice of the sampling time*
3. Selection of the **structure of the model**: size of the array of regressors, order of the model with respect to each variable, linear or non-linear model with respect to regressors and parameters.
4. **Parameters determination** (using the *learning set*).
5. **Model validation** (using the *cross-validation set*).

Example

Let us consider a perfectly mixed chemical reactor (CSTR) where an irreversible first order chemical reaction takes place: A → B.



The output variable of interest is the **concentration of the reactant**, and the manipulated variable is the **temperature of the fluid in the jacket.**

Identification procedure

(1) Input-output data for an identification/validation procedure are usually obtained disturbing the process, i.e. varying the input variables, in such a way that all the (operating) range of these variables is explored.

Pseudo-Random Binary Sequence (PRBS)

Let u be the input variable, and the values u_{MAX} and u_{MIN} be the maximum and minimum values of this variable. The variable u is then varied randomly between one of the two extreme values with a binary sequence:

$$(0 = x_{MIN}\,,\ 1 = x_{MAX})$$

Input variables can assume only two values, of the same magnitude, but with different sign, $\pm\Delta u$, with respect to steady-state values.

Identification procedure

Pseudo-Random Sequence (PRS)

The value of the input variable is calculated adding to the actual value a random value ***uniformly*** distributed in the interval [- Δu, + Δu].

Usually Δu is a fraction (15-20%) of the maximum variation of the input variable u.

Random Sequence (RS)

The value of the input variable is calculated adding to the actual value a random value in the interval [- Δu, + Δu].

Identification procedure

(2) The **sampling time** has to be carefully selected in order to allow for signal reconstruction.

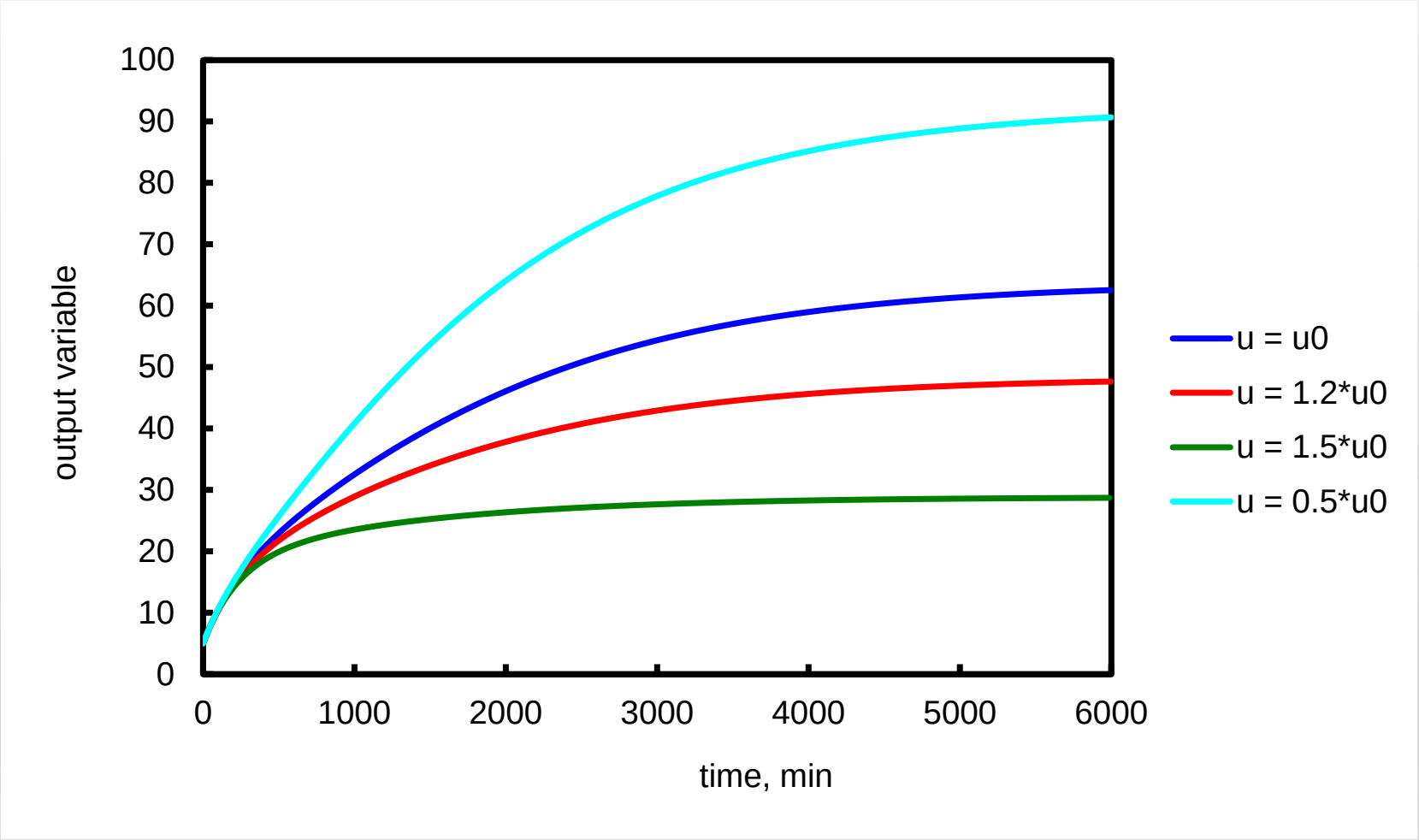
Shannon theorem: the sampling frequency should be twice the maximum frequency of the signal.

Generally, the sampling time should be a fraction (10-20%) of the main characteristic time of the process.

Input variables are varied every n sampling time t_s. Usually, the time nt_s is equal to 20% of the time required by the system to reach a steady-state value.

Example

Determine the time constant of the process



Identification procedure

If the sampling time is too small there are various drawbacks:

- there are **too much data** to be managed in the identification procedure;
- there is a strong risk to sample **noisy measures**;
- too similar data can cause problems to the algorithms used to identify model parameters;
- time required by calculations increases.

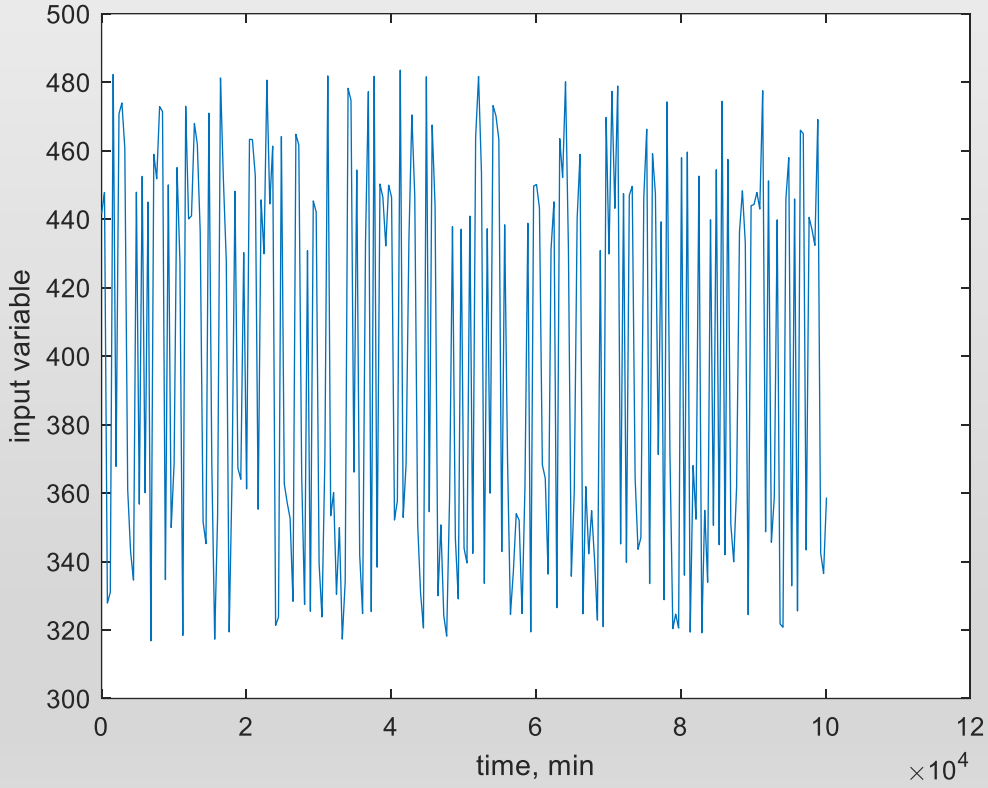
(3) Measured data can be **filtered**.

(4) **Outlier** can be eliminated by means of statistical techniques.

(5) **Detrend**: the mean (or steady-state) value can be subtracted from the measured data, so that sampled variables represent the deviation from the mean (or from the steady-state) value.

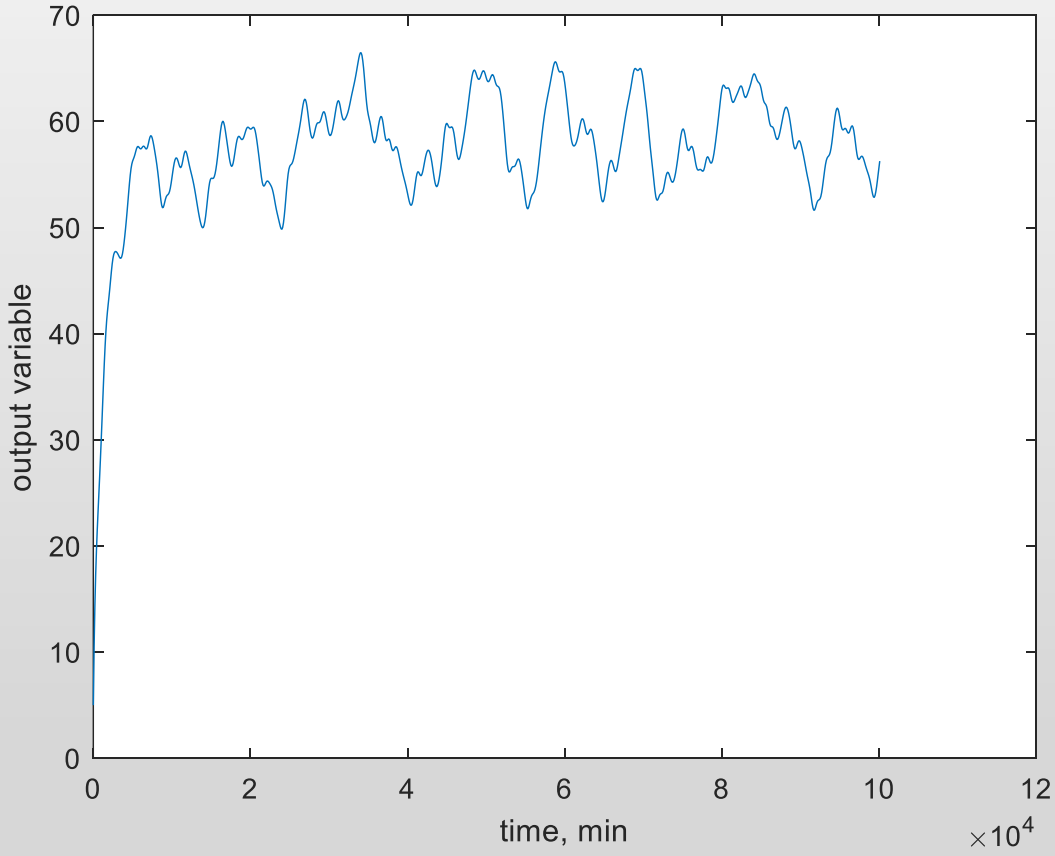
Example

Create a random sequence of input values for the identification procedure. In this framework it is possible to assume a variation of the input value of ±30% around the initial value.



Example

Create the set of input-output data required by the process identification algorithm.



Identification procedure

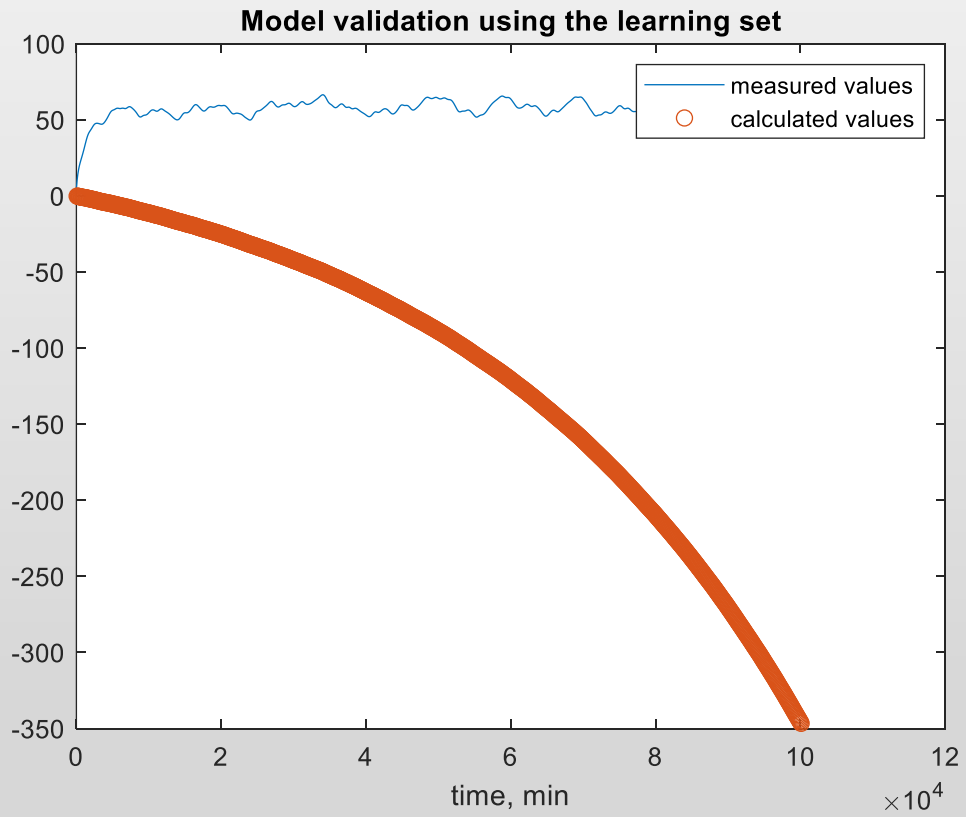
(6) The parameters **p** of the model are estimated by means of a regression procedure:

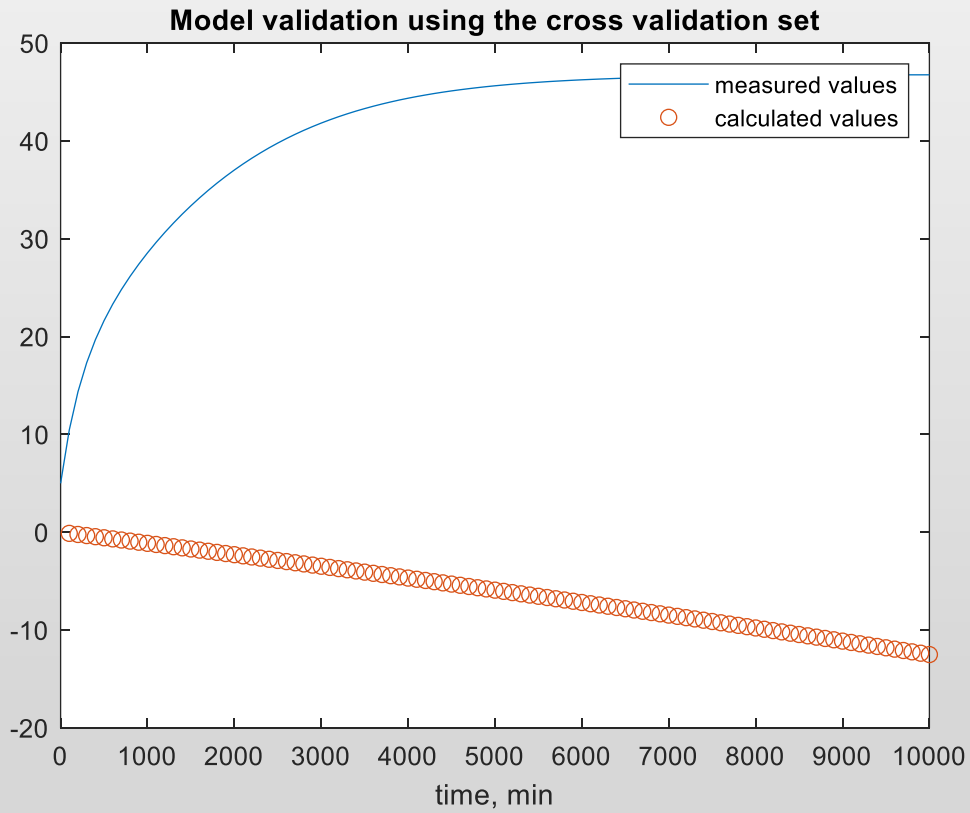
$$\min_{\mathbf{p}} \left\{ \sum_{i=1}^{n_y} \sum_{t=1}^{n_s} \left[y_i^{real}(t) - f_i(\varphi(t), \mathbf{p}) \right]^2 \right\}$$

where n_y is the number of output variables, and n_s is the number of samples.

Example

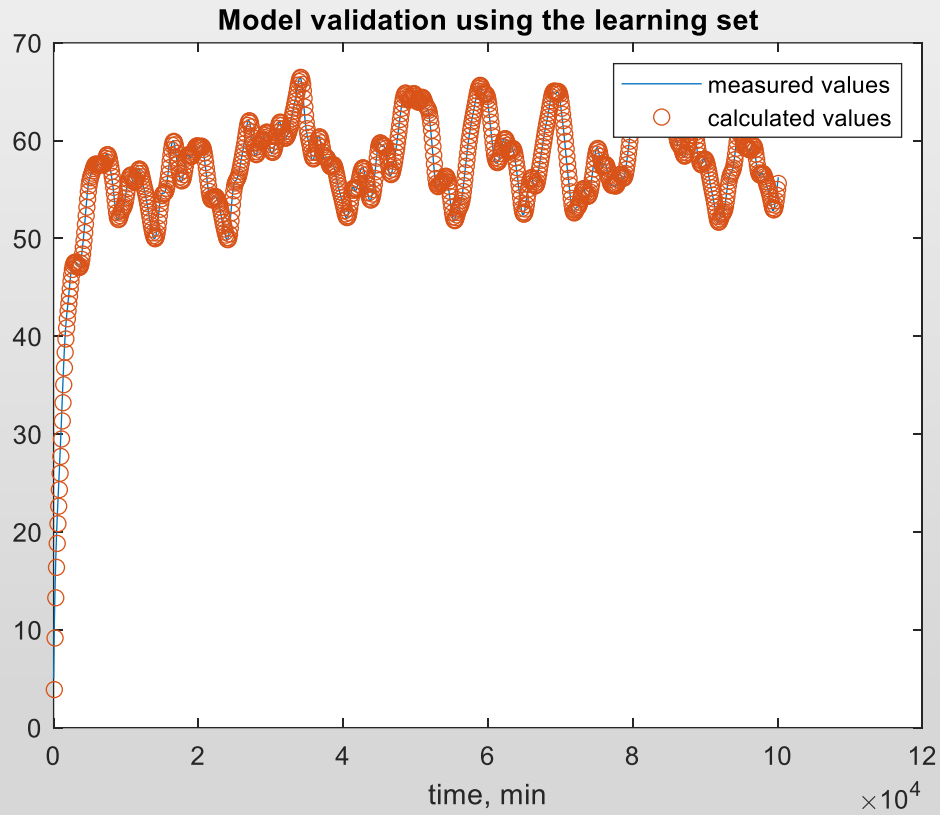
ARX (1,1)

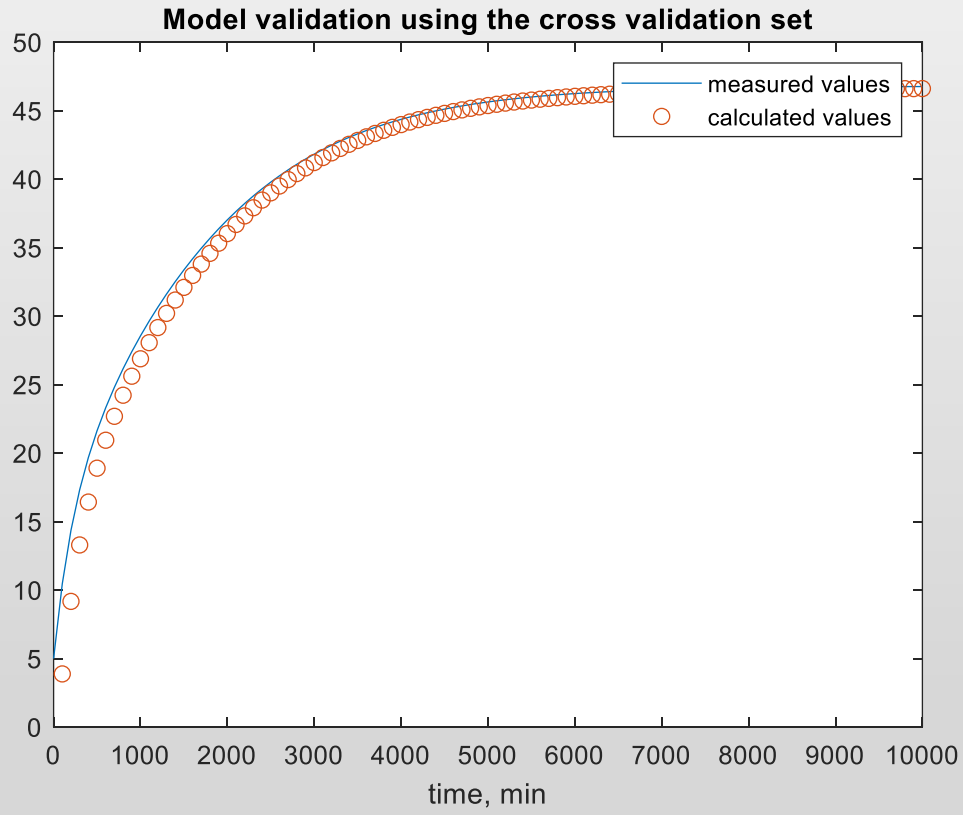




Example

ARMAX (1,1,1)





Identification procedure

(7) In order to asses the adequacy of the model to describe the process, we need a set of data (cross-validation set) different from that used to determine the parameters of the model (learning set).

We need to test the **overfitting**, i.e. the excessive specificity of the model with respect to learning data, and the capacity of extrapolation.

Cross validation index:

$$CVI = \frac{\sum_{i=1}^{n_{CVS}} \left(y_{real}(i) - y_{system}(i) \right)^2}{\sum_{i=1}^{n_{CVS}} \left(y_{system}(i) - y_{system}^{mean} \right)^2}$$

n_{CVS} is the number of data used for the validation.

Identification procedure

In order to find out the optimal order of a model, or to compare various models with different structures (ARX, ARMAX,…) it is possible to use **statistical criteria** that take into consideration various characteristics, e.g. order of the model, number of input and output variables, number of parameters…

AKAIKE INFORMATION CRITERION:

$$AIC = n_s \log\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(e(i)\right)^2\right) + 2n_p$$

FINAL PREDICTION ERROR CRITERION:

$$FPE = n_s \log\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(e(i)\right)^2\right) + n_s \log\left(\frac{n_s + n_p}{n_s - n_p}\right)$$

Identification procedure

BAYESIAN INFORMATION CRITERION:

$$BIC = n_s \log\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(e(i)\right)^2\right) + n_p \log\left(n_s\right)$$

LAW OF ITERATED LOGARITHMS CRITERION:

$$LILC = n_s \log\left(\frac{1}{n_s}\sum_{i=1}^{n_s}\left(e(i)\right)^2\right) + 2n_p \log\left(\log\left(n_s\right)\right)$$

n_s = number of samples

n_p = number of parameters

$$e(i) = y_{real}(i) - y_{system}(i)$$

References

1. Billings S. A., Woon W. S. F. A Prediction-Error and Stepwise-Regression Estimation Algorithm for Non-Linear Systems. *International Journal of Control*, 33(3), 803-822, 1986.
2. Hernando D., Desrochers A. A. Modeling of Non-Linear Discrete Time Systems for Input-Output Data. *Automatica*, 24(5), 629-641, 1988.
3. Hernandez W., Arcun Y, Control of Non-Linear Systems using Polynomial ARMA Models-. *AIChE Journal*, 39(3), 446-460, 1993.
4. Bosch P.P.J., van der Klauw A. C. *Modeling, Identification and Simulation of Dynamical Systems*. CRC Press, 1994.
5. Ljiung L. *System Identification: Theory for the User*. Prentice Hall, 1998.