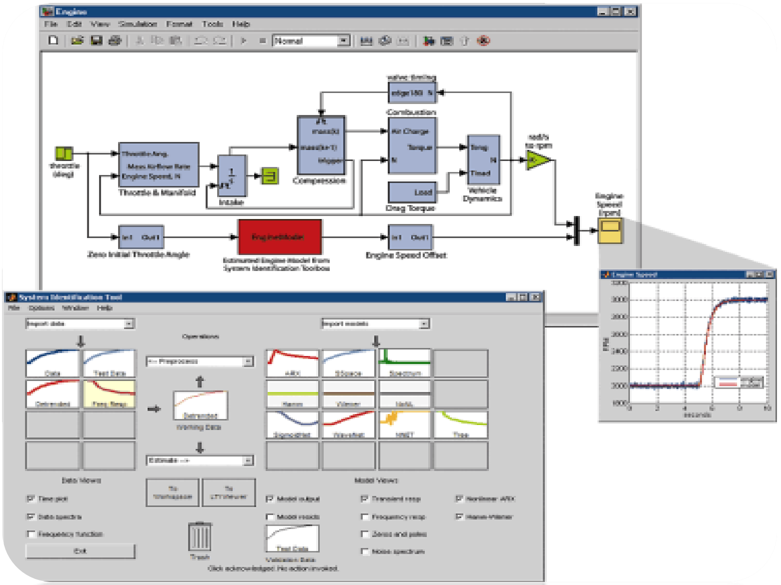


System identification

Davide Manca

Lesson 4 of “Dynamics and Control of Chemical Processes” – Master Degree in Chemical Engineering

POLITECNICO
MILANO 1863

Introduction

- The mathematical model of a process unit or of a whole process is of paramount importance for engineering applications.
- The numerical model of a process can be applied to chemistry, electronics, mechanics, economics, …

- A model can be used to find quantitative answers without measuring real processes or making **experiments**.

- Another important feature consists in the capability of **predicting** the future response of the system → model-based multivariable control; process optimization.

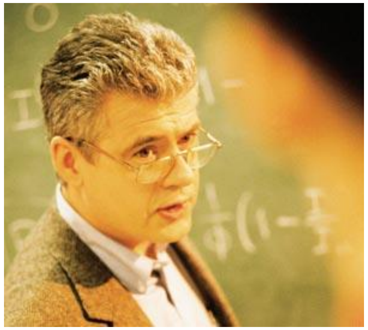
Some questions

- How can we build a mathematical model for prediction purposes?
- How to use the experimental data (if available) for the model synthesis?
- How to assess the reliability and consistency of the model?

- The mathematical models can be classified into two classes:
  - Basic models;
  - Empirical models.
- **BASIC MODELS**: they are based on an in-depth knowledge of the physical features of the system. Conservation laws: mass, energy, momentum. First-principle models. Deterministic models.
- **EMPIRICAL MODELS**: the process is not described by any physical laws. Conversely, it is defined by means of quantitative observations, experiments, and measures.

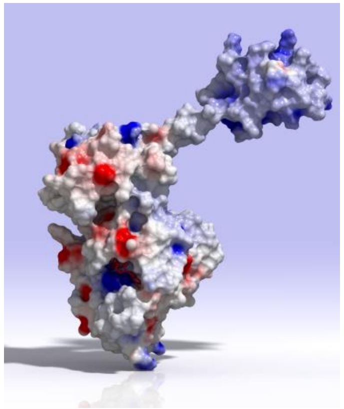
Black-box models

- A black-box model is completely independent from the physics of the process to be identified.
- Theoretically, it is possible to build (i.e. identify) a black-box model without knowing anything of the process to be modeled.
- Obviously, an **in-depth knowledge** of the process allows increasing the quality of the identified model.
- Usually, it is necessary to **find a balance** between the prediction quality of the model and its complexity.



When is the identification recommended?

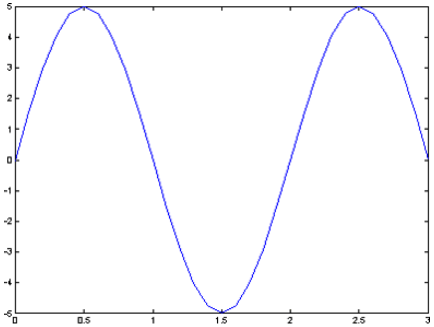
- If one is **ignorant** about the process to be modeled;
- If the **complexity** of the process to be modeled is high;
- If we need **fast** solutions:
  - to carry out the model;
  - in the simulation of the model, *i.e.* CPU time.

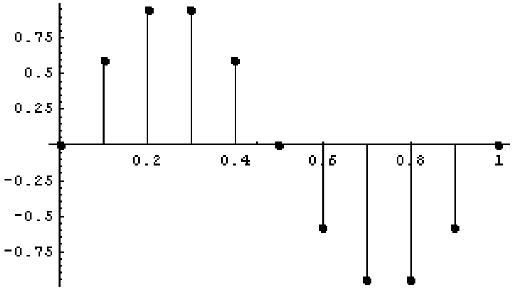


- If we need a model that does not produce mathematical errors (**floating point exceptions**).

Signals and systems

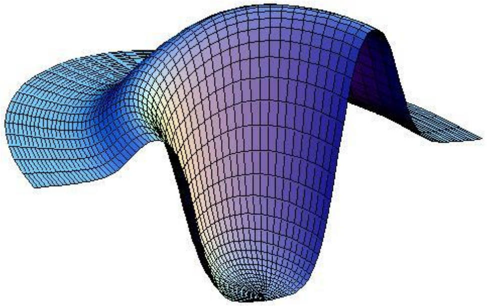
- It is worth introducing the concepts of **signal** and **system** according to the nomenclature introduced by Bosh e Klaw, 1994.
- **SIGNAL**
  - It is something holding/bearing **information**.
  - **Deterministic** signals: they are completely defined for instance by a mathematical expression;
  - **Stochastic** signals: the exact future value of the signal cannot be predicted. The signal is described by a statistical approach, for instance in terms of mean value and standard deviation.
  - A signal may be **continuous** or **discrete**.





Signals and systems

- **SYSTEM**
  - It is a set of **relationships between variables and signals**.
  - The system is an ideal representation of the real process.



- **STATE-SPACE MODELS**
  - The state of a dynamic system contains the whole past history of the process;
  - Therefore, it is possible to predict the future behaviour of the system without having to know the past history of the process.
  - Mathematically, this means that the system features a number of **state variables**, x_i, that contribute to its dynamic description by means of a system of n differential equations of the first order.
  - n is the "order of the system".

Signals and systems

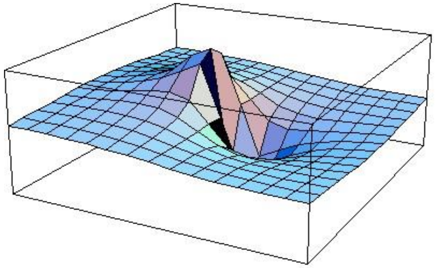
- **STATE-SPACE MODELS**
  - Besides the state variables, there are the **input variables**, u_i, which are the so called “driving forces” of the system.
  - The solution of the system of n differential equations provides the whole picture of the future behaviour of the system.
  - To know the x_i values may be of reduced interest as the state variables are not necessarily observable.
  - The most interesting variables, y_i, are the **output variables** that depend on the x_i and u_i variables.
  - The **deterministic** and **continuous** formulation of a **state-space** system is:

$$\dot{\mathbf{x}}(t) = \mathbf{G}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

$$\mathbf{y}(t) = \mathbf{H}\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

Signals and systems

- STATE-SPACE MODELS
  - The **deterministic** and **discrete** of a **state-space** system is:



$$\mathbf{x}(t+1)=\mathbf{g}\left(\mathbf{x}(t),\mathbf{u}(t)\right)$$
$$\mathbf{y}(t)=\mathbf{h}\left(\mathbf{x}(t),\mathbf{u}(t)\right)$$

- BLACK-BOX MODELS
  - As aforementioned, the black-box models exchange information with the **observer** only by means of signals. Therefore, the **observer** cannot know the $\mathbf{g}$ and $\mathbf{h}$ functions that rule/characterize the system.
  - Given a **black-box model**, the only pieces of information available to the observer are the **input and output signals**. Consequently, the state variables, $\mathbf{x}$, are unknown.
  - A black-box model does not use any state variables.

Black-box models

- A black-box model is characterized by the following mathematical formulation:

$$\text{system output} = \mathbf{f}\left(\text{system input}\right) \quad \Rightarrow \quad \mathbf{y} = \mathbf{f}\left(\mathbf{u}\right)$$

- If the model is dynamic then the system output depends on the past history of the inputs and outputs:

$$\text{present outputs} = \mathbf{f}\left(\text{past outputs, past inputs}\right) \quad \Rightarrow \quad \mathbf{y}_{now} = \mathbf{f}\left(\mathbf{y}_{old}, \mathbf{u}_{old}\right)$$

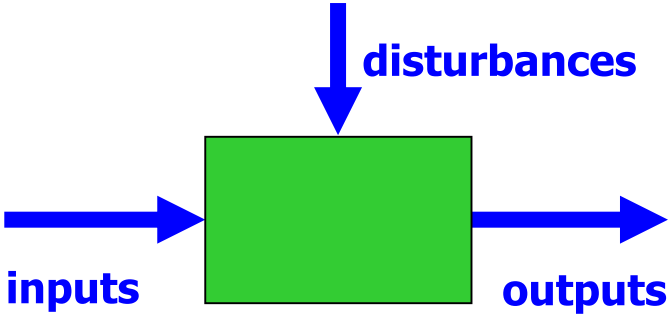
- In order to successfully carry out the identification procedure of the functional dependency, **f**, it is worth introducing some **adaptive parameters**, **p**.

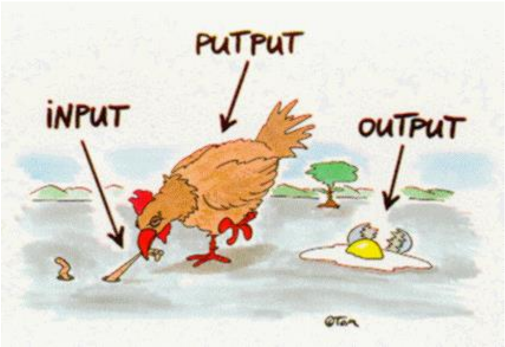
$$\text{present outputs} = \mathbf{f}\left(\text{past outputs, past inputs, parameters}\right) \quad \Rightarrow \quad \mathbf{y}_{now} = \mathbf{f}\left(\mathbf{y}_{old}, \mathbf{u}_{old}, \mathbf{p}\right)$$

Black-box models

- The $\mathbf{p}$ parameters may be used as independent variables of the identification procedure (*i.e.* degrees of freedom) with the *objective* that the black-box model describes in the best way the input-output data coming from the real process.
- A possible improvement of the black-box model, $\mathbf{f}$, consists in accounting for the error, $\mathbf{e}$, which measures the distance of the identified system from the real process.

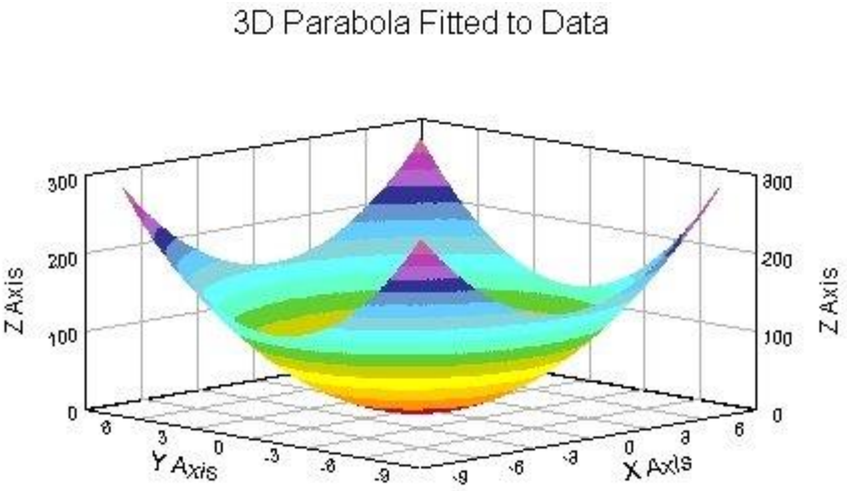
$$\text{present outputs} = \mathbf{f}\begin{pmatrix}\text{past outputs, past inputs,} \\ \text{past errors, parameters}\end{pmatrix} \quad \Rightarrow \quad \mathbf{y}_{now} = \mathbf{f}\left(\mathbf{y}_{old}, \mathbf{u}_{old}, \mathbf{e}_{old}, \mathbf{p}\right)$$





System identification

- The system identification calls for three steps:
  1. **Selection** of the set of inputs, $\mathbf{u}$, outputs, $\mathbf{y}$, errors, $\mathbf{e}$, in terms of number of unknowns and length of the time interval;
  2. Selection of the **regressor**, $\mathbf{f}$;
  3. **Model identification** in terms of model regression respect to the observations. This is done by means of suitable parameters that play the role of independent variables (i.e. degrees of freedom).



Regressors

- In general, a model has not to necessarily consider all the observable/measurable inputs and outputs of the process.
- In case of a system to be used for **control purposes** may be suitable to consider just the **controlled**, **manipulated** variables and possibly the **measurable disturbances**.
- Usually, we have:
  - r output variables, $\mathbf{y}$;
  - m input variables, $\mathbf{u}$.

- The error vector is: $\mathbf{e} = \mathbf{y}_{real} - \mathbf{y}$
- The $r + m$ variables of the model are sampled (i.e. measured and stored) every sampling time, t_s.
- These variables describe the system history (and take into account the attenuation/fading effect of the signal with time).

Regressors

- The system to be identified has the following formulation:

$$\begin{aligned}\mathbf{y}(t)=\mathbf{f}[&y_1(t-1),\dots,y_1\left(t-n_{y_1}\right),\dots,y_r(t-1),\dots,y_r\left(t-n_{y_r}\right),\\ &u_1(t-1),\dots,u_1\left(t-n_{u_1}\right),\dots,u_m(t-1),\dots,u_m\left(t-n_{u_m}\right),\\ &e_1(t-1),\dots,e_1\left(t-n_{e_1}\right),\dots,e_r(t-1),\dots,e_r\left(t-n_{e_r}\right)]\end{aligned}$$

- We introduce the φ vector whose components are called **regressors**:

$$\begin{aligned}\boldsymbol{\varphi}(t)=[&y_1(t-1),\dots,y_1\left(t-n_{y_1}\right),\dots,y_r(t-1),\dots,y_r\left(t-n_{y_r}\right),\\ &u_1(t-1),\dots,u_1\left(t-n_{u_1}\right),\dots,u_m(t-1),\dots,u_m\left(t-n_{u_m}\right),\\ &e_1(t-1),\dots,e_1\left(t-n_{e_1}\right),\dots,e_r(t-1),\dots,e_r\left(t-n_{e_r}\right)]^T\end{aligned}$$

- If d is the total number of system variables, then the length ℓ of the φ vector is the **total order** of the model:

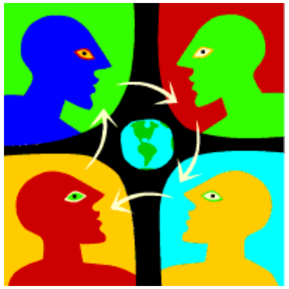
$$\ell=\sum_{i=1}^{d}n_i$$

Regressors

- It is worth considering that the inputs may have a delayed effect on the outputs. This can be accounted for by introducing the so-called **time delays**, n_{ki}, in the system model for each m input:

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

Function $\mathbf{f}$



- The $\mathbf{f}$ function, through the $\mathbf{p}$ parameters, **maps** the regressors vector into the $\mathbf{y}$ variables:

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

- We can have either **linear** or **non-linear** mapping regressors (i.e. functions).
- The simplest model for the $\mathbf{f}$ function is:

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

N.B.: $\mathbf{p}$ is a row vector whilst $\boldsymbol{\varphi}$ is a column vector.

- Likewise, if $\mathbf{p}$ is a matrix then $\mathbf{y}$ is a vector.
- If we hypothesize that the output vector, $\mathbf{y}$, is the sum of two terms referred to a **deterministic contribution** (not disturbed) ρ and a **disturbance contribution** $\boldsymbol{\omega}$, we have:

$$\mathbf{y}(t)=\boldsymbol{\rho}(t)+\boldsymbol{\omega}(t)$$

Function $\mathbf{f}$

- In the formulation $\mathbf{y}(t)=\boldsymbol{\rho}(t)+\boldsymbol{\omega}(t)$ the $\boldsymbol{\omega}$ term represents the contribution to the output variable, $\mathbf{y}$, which cannot be deterministically modelled.
- $\boldsymbol{\omega}$ produces a stochastic effect on the system. $\boldsymbol{\omega}$ is the noise and/or the system deviation from the ideal linearity.
- The deterministic contribution to the model can be expresses as follows:

$$\boldsymbol{\rho}(t)=\mathbf{G}(q,\mathbf{p})\mathbf{u}(t)$$

  - where $\mathbf{G}$ is a **rational transfer function** matrix in the **translation operator** q:

$$\mathbf{G}(q,\mathbf{p})=\frac{\mathbf{B}(q)}{\mathbf{A}(q)}=\frac{\mathbf{b}_1 q^{-\mathbf{n}_k}+\mathbf{b}_2 q^{-\mathbf{n}_k-1}+\ldots+\mathbf{b}_{n_u} q^{-\mathbf{n}_k-\mathbf{n}_u}}{1+\mathbf{a}_1 q^{-1}+\ldots+\mathbf{a}_{n_\rho} q^{-\mathbf{n}_\rho}}$$

  - The plain representation of the problem is:

Time delay

$$\boldsymbol{\rho}(t)+\mathbf{a}_1\boldsymbol{\rho}(t-1)+\ldots+\mathbf{a}_{n_\rho}\boldsymbol{\rho}\left(t-\mathbf{n}_\rho\right)=\mathbf{b}_1\mathbf{u}\left(t-\mathbf{n}_k\right)+\ldots+\mathbf{b}_{n_u}\mathbf{u}\left(t-\mathbf{n}_k-\mathbf{n}_u\right)$$

Function f

- Likewise, it is possible to model the disturbance contribution:

$$\boldsymbol{\omega}(t)=\mathbf{H}(q,\mathbf{p})\mathbf{e}(t)$$

  - where $\mathbf{H}$ is a **rational transfer function** matrix in the **translation operator** q:

$$\mathbf{H}(q,\mathbf{p})=\frac{\mathbf{D}(q)}{\mathbf{C}(q)}=\frac{\mathbf{d}_1 q^{-1}+\mathbf{d}_2 q^{-2}+\ldots+\mathbf{d}_{n_e} q^{-\mathbf{n}_e}}{1+\mathbf{c}_1 q^{-1}+\ldots+\mathbf{c}_{n_\omega} q^{-\mathbf{n}_\omega}}$$

  - The plain representation of the problem is:

$$\boldsymbol{\omega}(t)+\mathbf{c}_1\boldsymbol{\omega}(t-1)+\ldots+\mathbf{c}_{n_\omega}\boldsymbol{\omega}(t-\mathbf{n}_\omega)=\mathbf{d}_1\mathbf{e}(t-1)+\ldots+\mathbf{d}_{n_e}\mathbf{e}(t-\mathbf{n}_e)$$

- Eventually, the general formulation of the problem becomes:

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

BOX-JENKINS formulation

Function f

- It is worth observing that the Box-Jenkins formulation depends on 5 structural parameters: $\mathbf{n}_u \quad \mathbf{n}_e \quad \mathbf{n}_\omega \quad \mathbf{n}_\rho \quad \mathbf{n}_k$
- and on 4 adaptive parameters: $\mathbf{a} \quad \mathbf{b} \quad \mathbf{c} \quad \mathbf{d}$

- A simplification of the Box-Jenkins model can be obtained by imposing that:

$$\mathbf{A}(q)=\mathbf{C}(q)$$

- The model becomes: $\mathbf{A}(q)\mathbf{y}(t)=\mathbf{B}(q)\mathbf{u}(t)+\mathbf{D}(q)\mathbf{e}(t)$ **ARMAX model**
- The **ARMAX** acronym derives from:
  - AUTOREGRESSIVE $\mathbf{A}(q)\mathbf{y}(t)$
  - MOVING AVERAGE $\mathbf{D}(q)\mathbf{e}(t)$
  - EXOGENOUS INPUT $\mathbf{B}(q)\mathbf{u}(t) \longleftarrow$ This is an “extra input” that in economic terms is defined as “Exogenous Input”

Regressors

- The regressors vector in case of **ARMAX** models is:

$$\boldsymbol{\varphi}_{ARMAX}(t)=[\mathbf{y}(t-1),\ldots,\mathbf{y}(t-\mathbf{n}_y);\ \mathbf{u}(t-\mathbf{n}_k-1),\ldots,\mathbf{u}(t-\mathbf{n}_k-\mathbf{n}_u);\ \mathbf{e}(t-1),\ldots,\mathbf{e}(t-\mathbf{n}_e)]$$

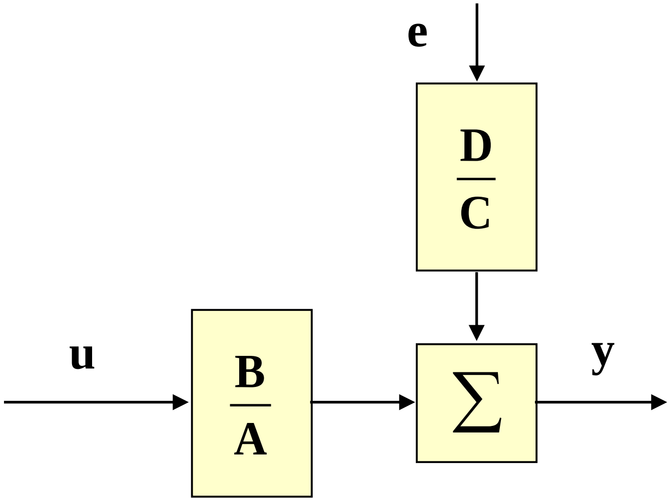
- If we remove the moving average term we get the **ARX** model:

$$\boldsymbol{\varphi}_{ARX}(t)=[\mathbf{y}(t-1),\ldots,\mathbf{y}(t-\mathbf{n}_y);\ \mathbf{u}(t-\mathbf{n}_k-1),\ldots,\mathbf{u}(t-\mathbf{n}_k-\mathbf{n}_u)]$$

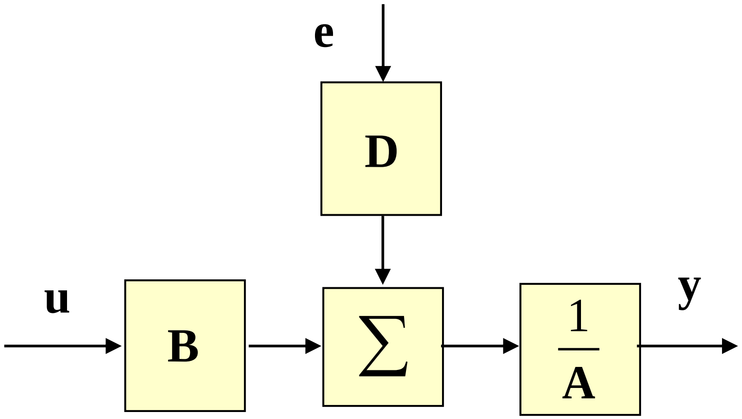
- Eventually, if we set $n_y = 0$ in the ARX model we get a **FIR** (**F**inite **I**mpulse **R**esponse) model:

$$\boldsymbol{\varphi}_{FIR}(t)=[\mathbf{u}(t-\mathbf{n}_k-1),\ldots,\mathbf{u}(t-\mathbf{n}_k-\mathbf{n}_u)]$$

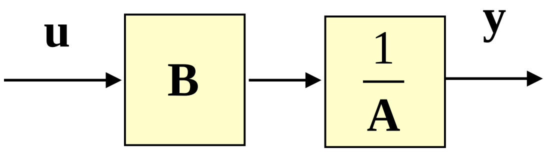
Identification models



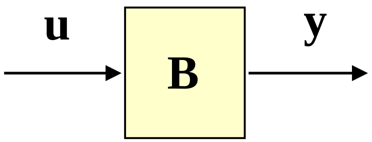
BOX-JENKINS



ARMAX



ARX



FIR

ARX models

- The mathematical models presented so far may have a scalar, vector, or mixed structure:
    - **SISO**: Single Input – Single Output
    - **MISO**: Multiple Input – Single Output
    - **MIMO**: Multiple Input – Multiple Output

- **FEATURES**
    - The ARX model is **linear** both in the **regressors** and **parameters**.
    - As such, it cannot describe **multiple steady states**;
    - By definition, it can **not** describe **non-linear** dynamics;
    - Its identification is rather **simple**;
    - The **CPU time** for a model prediction is **moderate**.

- **Example of an ARX SISO**

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

ARMAX models

- **FEATURES**
  - An ARMAX model is **linear** in the input, output, and error **variables**.
  - The prediction capability of ARMAX models is better than the ARX ones thanks to the presence of the error terms, $\mathbf{e}$;
  - The error term can, somehow, account for:
    - Process non-linearity;
    - Unmeasured disturbances;
    - Measures noise.
  - It cannot account for **multiple steady states**;
  - The model outputs are evaluated with the scalar product between the regressors vector and the parameters one;

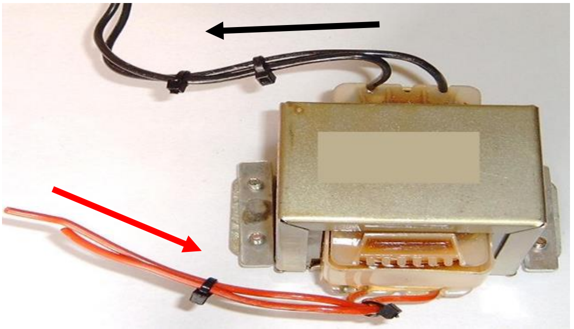
$$y(t) = \mathbf{p} \times \boldsymbol{\varphi}(t, \mathbf{p})$$

  - The ARMAX model is **not linear** in the regression **parameters**.

ARMAX models

- **FEATURES**
  - The evaluation of the model parameters, $\mathbf{p}$, calls for a **non-linear regression** procedure → higher CPU time.
  - The CPU time for a prediction is higher than that of an ARX one.
  - The presence of the error term $\mathbf{e} = \mathbf{y}_{real} - \mathbf{y}$ calls for the in-the-field measurement of the real output variables. As such, an ARMAX model is not recommended/suitable for the prediction on n steps forward.
- **Example of an ARMAX SISO**

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



Non-linear models

- Going on with the concept of function $\mathbf{f}$ interpreted as the **map** of the regressors vector $\boldsymbol{\varphi}$ in the output variables $\mathbf{y}$ through the parameters $\mathbf{p}$:

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

  we can implement/adopt any non-linear $\mathbf{f}$ function.

- Usually, it is worth adopting the expansion of a base function $f_k\left(\boldsymbol{\varphi}(t)\right)$ such that:

$$\mathbf{f}\left[\boldsymbol{\varphi}(t),\mathbf{p}\right]=\sum_k p_k f_k\left(\boldsymbol{\varphi}(t)\right)$$

- One of the most used forms is the polynomial expansion of the regressor elements:

$$\mathbf{y}(t)=\sum_{k=1}^{N}\alpha_k\varphi_k+\sum_{k=1}^{N}\sum_{l\geq k}^{N}\beta_k\varphi_k\varphi_l+\sum_{k=1}^{N}\sum_{l\geq k}^{N}\sum_{m\geq l}^{N}\gamma_k\varphi_k\varphi_l\varphi_m+\ldots$$

N.B.: this expansion is **linear** in the **parameter** but is **non-linear** in the N **regressors**.

Non-linear models

- Another alternative is to use functions f_k obtained from the parameterization of a single **basis mother function**, κ.
- Usually, κ is further parameterized by means of two distinct parameters:
  - β_k refers to either a scale or directional property;
  - γ_k refers to either a position or a translation.

$$f_k\left(\varphi(t)\right)=\kappa\left(\varphi(t),\beta_k,\gamma_k\right)=\kappa\left(\beta_k\left(\varphi-\gamma_k\right)\right)$$

- For instance, if $\kappa = \cos(x)$, the expansion becomes a **Fourier series** where β_k are the frequencies and γ_k the phases:

$$\mathbf{f}\left[\varphi(t),\mathbf{p}\right]=\sum_k p_k f_k\left(\varphi(t)\right)=\sum_k \alpha_k\kappa\left(\varphi(t),\beta_k,\gamma_k\right)=\sum_k \alpha_k \cos\left(\beta_k\left(\varphi(t)-\gamma_k\right)\right)$$

- Another basis mother function usually used is the **sigmoid**:

$$\mathbf{f}\left[\varphi(t),\mathbf{p}\right]=\sum_k p_k f_k\left(\varphi(t)\right)=\sum_k \alpha_k\kappa\left(\varphi(t),\beta_k,\gamma_k\right)=\sum_k \alpha_k \frac{1}{1+e^{\left(\beta_k\left(\varphi(t)-\gamma_k\right)\right)}}$$

NARX models

- Similarly to the linear models, it is possible to introduce the non-linear models: **NARMAX**, **NARX**, **NFIR**.
- As far as the **NARX** models are concerned we can *build* them starting from both the input and output vectors trough two consecutive mappings:
  - Firstly, we build a **non-linear polynomial mapping**, g, of the regressors vector:

$$\boldsymbol{\eta}(t)=g\left(\boldsymbol{\varphi}(t)\right) \quad \Rightarrow$$

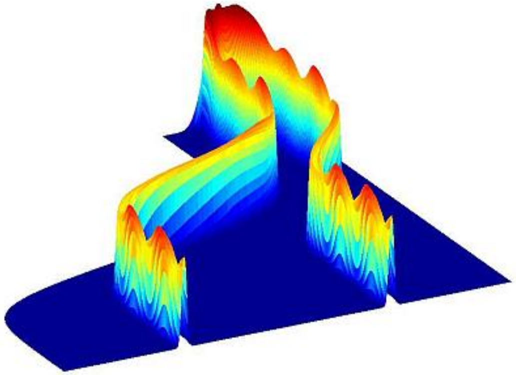
$$\begin{aligned}
\eta_1 &= \varphi_1 \\
&\vdots \\
\eta_N &= \varphi_N \\
\eta_{N+1} &= \varphi_1\varphi_1 \\
\eta_{N+2} &= \varphi_1\varphi_2 \\
&\vdots \\
\eta_{N+N} &= \varphi_1\varphi_N \\
\eta_{2N+1} &= \varphi_2\varphi_2 \\
&\vdots \\
\eta_{2N+N-1} &= \varphi_2\varphi_N \\
&\vdots
\end{aligned}$$

  - Secondly, the new regressors vector, $\boldsymbol{\eta}$, is **linearly mapped** in the output vector, $\mathbf{y}$:

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

NARX models

- **FEATURES**
  - Since the model structure is **linear in the parameters** then it is sufficient to use an algorithm based on the minimization of the squared errors to determine the vector $\mathbf{p}$;
  - Therefore, the **computation effort** for the identification is absolutely analogous to that done in the case of the ARX models;
  - Thanks to its intrinsic non-linearity a NARX model can describe **multiple steady states**;
  - it does **not** call for any **in-the-field measures** of the output vector, $\mathbf{y}$, for prediction purposes as it does not implement the error terms, $\mathbf{e}$;
  - A NARX model, if not properly structured, may produce **unbounded output** with an evident decay of the predictive capabilities of the model.



NARX models

- **Example of a NARX SISO**
  - Let us consider as an example a NARX SISO based on the quadratic polynomial expansion of a regressors vector with four elements (*i.e.* two inputs and two outputs): $y(t-1), y(t-2), u(t-1), u(t-2)$

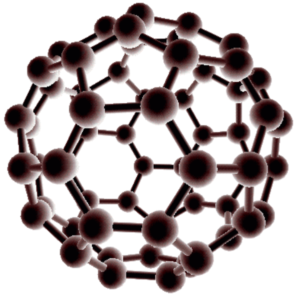
    the NARX system is:

$$\begin{aligned}y(t) = {} & a_1 y(t-1) + a_2 y(t-2) + a_3 u(t-1) + a_4 u(t-2) + a_5 y(t-1) y(t-1) + a_6 y(t-1) y(t-2) + \\ & a_7 y(t-1) u(t-1) + a_8 y(t-1) u(t-2) + a_9 y(t-2) y(t-2) + a_{10} y(t-2) u(t-1) + \\ & a_{11} y(t-2) u(t-2) + a_{12} u(t-1) u(t-1) + a_{13} u(t-1) u(t-2) + a_{14} u(t-2) u(t-2)\end{aligned}$$

  - The model is based on **14** parameters despite the reduce dimensions.
  - There are problems of **overparameterization**. For instance, a third-order (i.e. cubic) polynomial expansion with a regressors vector of 6 elements would produce a model with **83** parameters.
  - It is highly recommended to identify suitable computational algorithms capable of **removing the negligible coefficients** → Stepwise algorithms.

Identification procedure

- Assessment of the system limits and the necessary variables: as a function of the model specifications we can define the **the exact number of input and output variables**. We also define/identify their **variability range** to organize/create a suitable *learning* domain for the following identification procedure. The variables are selected as a function of the physical/empirical knowledge of the process, correlation indexes, and trial & error.
- Design of experiments: once the model variables are defined we have to assign the **sampling frequency**. Moreover, it is worth that all the input variables can be disturbed. It is also worth considering if it is possible to disturb the variables that not directly affect the operation of the system to be identified.
- Selection of the model structure: we have to define the **length** of the regressors vector, the model **order** respect to any model variable, the **linearity** or **non**-**linearity** of the model respect to both the regressors and the parameters.



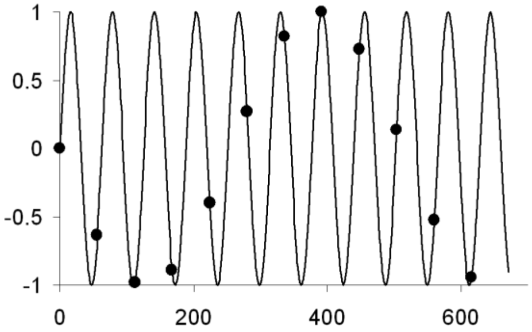
Identification procedure

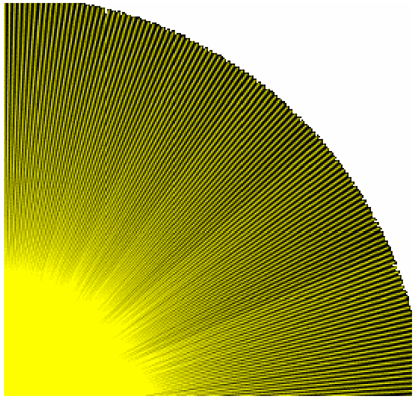
- **Parameter identification**: it is now necessary to identify/define the **numerical algorithm** for the evaluation of the model parameters. We have to differentiate between **deterministic** (minimization of the computed error) and **stochastic** models (method of maximum likelihood).
- **Simulation and validation**: once the model is identified, it is worth **testing** its predictive capability and by providing a set of input data not previously used. Indeed, the model validation procedure uses a set of validation data (**cross-validation set**) that is suitably chosen *a priori* and kept separate from the training set (**learning set**).

Sampling time

- To produce and store the two sets of input data for the system identification and validation it is necessary to define the time interval when to measure those data.
- Generally, the sampling time is a fraction, 5%-20%, of the characteristic time of the system.
- **Shannon theorem**: if one samples a signal at a frequency at least double than the maximum frequency of the system, it is possible to reconstruct the original signal without any information loss.
- If the Shannon theorem is not respected then it is possible to incur in **aliasing problems**. This means that it is not possible to reconstruct the original signal from the sampled data.

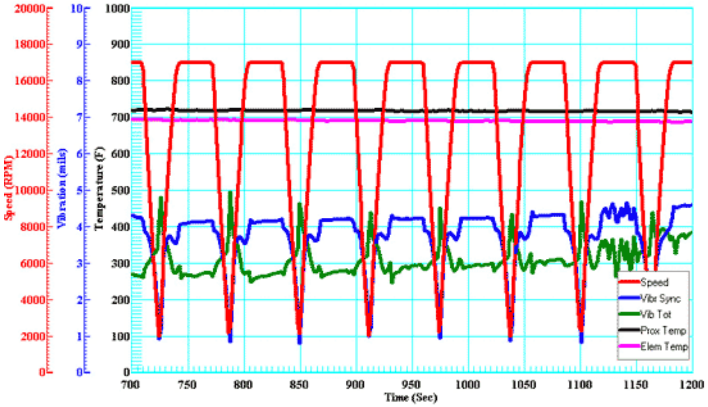
Claude Shannon
1916-2001





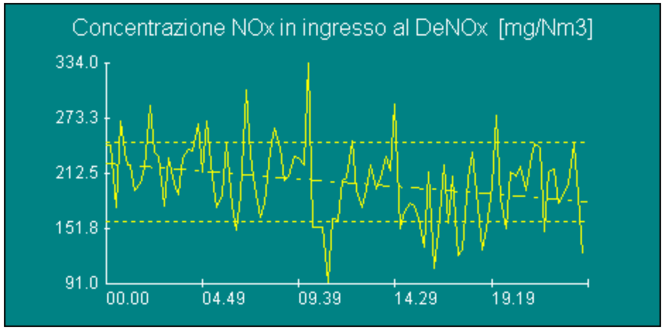
Sampling time

- Having defined the sampling time, t_s, of the signal, if it is **too high** then it is not possible to describe the real process dynamics.
- If t_s is **too short**:
  - there is probably a large amount of process data that must be manipulated without achieving any improvement in the information they yield;
  - there is the risk of increasing the sampling of the noise;
  - too similar data induce numerical problems in the identification algorithm;
  - the CPU time increases for both the identification and prediction procedures;
  - The in-the-field measuring cost may increase.

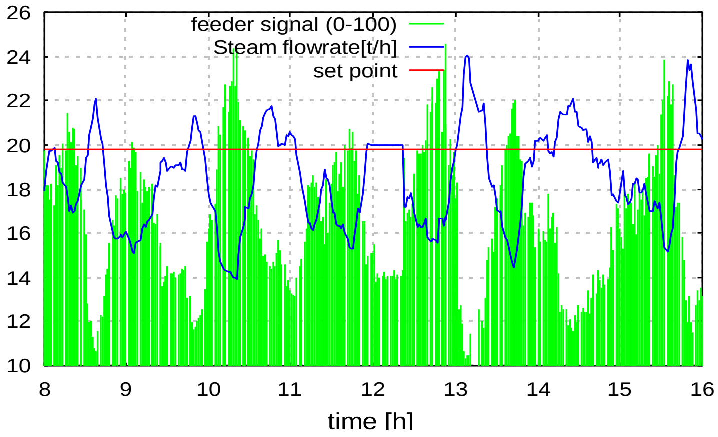


Data preprocessing

- Once the in-the-field data have been sampled and stored it is possible to apply some suitable mathematical operators to dampen the excessive oscillations (for instance **moving average**).
- It possible to apply low-pass and high-pass **filters** to remove the sudden variations beyond the standard operating limits.
- In addition, it is possible to identify and remove the so-called **outliers** by suitable techniques of statistical analysis.

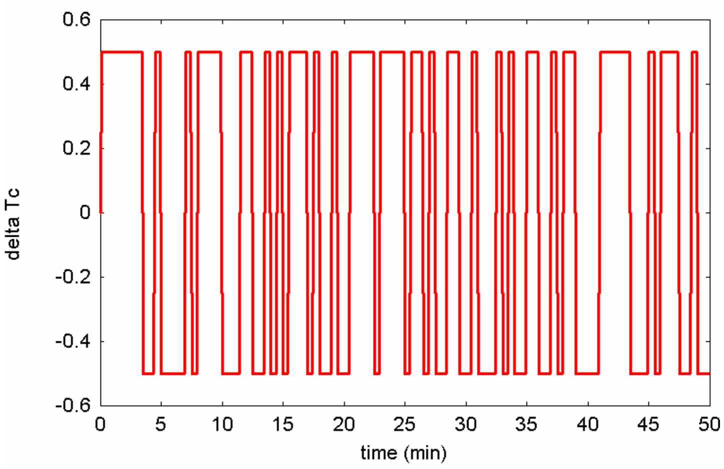


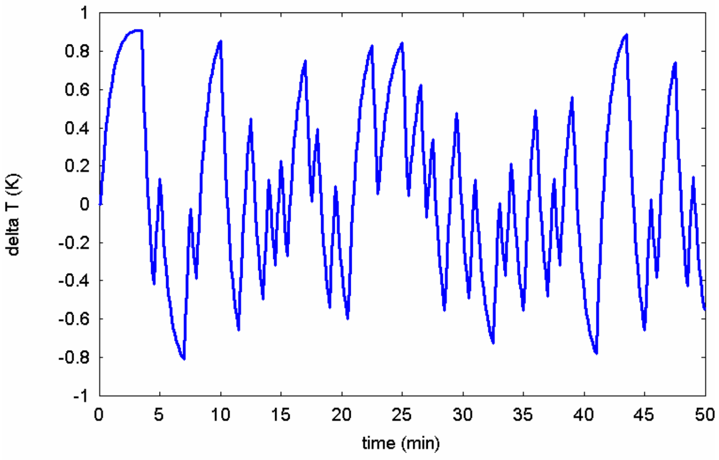
- **DETREND**: the mean value is subtracted to the data. By doing so, the sampled variables show only the deviation from the stationary values or more in general from the average operating conditions. This allows also using the model for other stationary points (although the quality of the model may decrease).



Disturbance sequence

- The input-output data collection for both the identification and validation procedures happens by disturbing the process input variables according to different techniques that try to properly cover the operating conditions.
- **PRBS**: Pseudo Random Binary Sequence. One has to choose two band limits, x_{MIN} , x_{MAX}, of the variable to be disturbed, x, and these values are randomly varied between such values with a binary sequence ($0 = x_{MIN}$, $1 = x_{MAX}$). The output vector is then measured after that sequence.

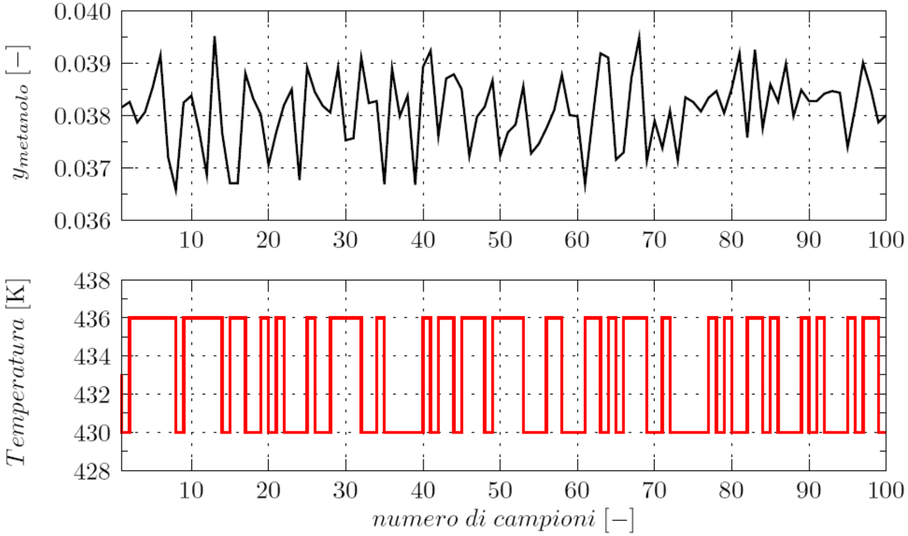




PRBS

1. The input variables may take **only two values**, with the same amplitude and opposite sign, $\pm\Delta u$, respect to the stationary conditions (assumed as reference values at 0 after the detrend procedure).
2. Moving from a **positive** to a **negative** condition, and *vice versa*, happens in a **random** way so to produce in the sequence the so-called white noise (*i.e.* with null average).
3. The disturbance on the input variables is given every n sampling time, t_s, with a 0.5 probability of changing the sign.
4. Usually, the interval $n \cdot t_s$ is equal to 20% of the time taken by the system to reach a steady-state condition (*i.e.* complete the transient).
5. The amplitude Δu of the disturbance must be sufficiently high to avoid any measurement disturbance induced by the system noise.

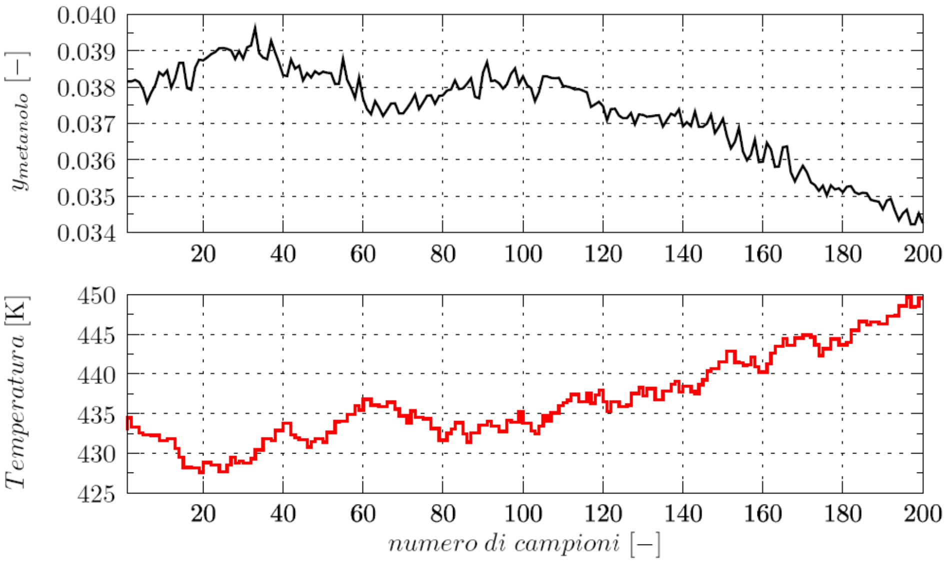
Recommended for ARX



PRS

- The main characteristic that differentiates a linear system from a non-linear one is the proportional dependency of its response after a given disturbance amplitude.
- In case of linear system, if the input amplitude doubles then also the output amplitude doubles. Not the same in case of non-linear system.
- In a **PRS**, **P**seudo **R**andom **S**equence, the input variable value is assigned by summing at the present value a random quantity whose distribution is uniform in the $[-\Delta u, +\Delta u]$ interval.
- The input variables undergo a random disturbance that is a **constant fraction** (in absolute value) of the maximum variability interval (~15-20%).

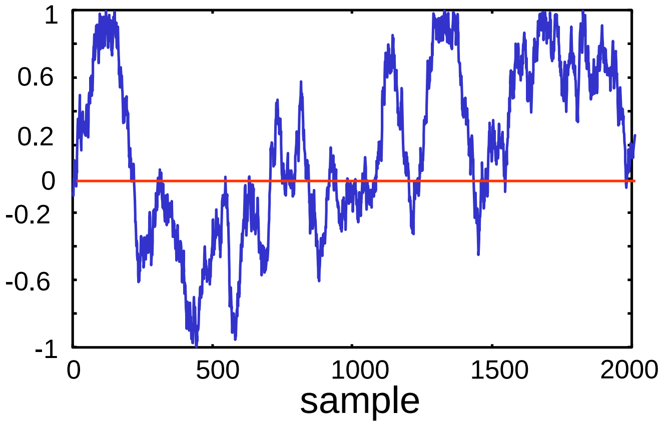
Recommended for ARMAX

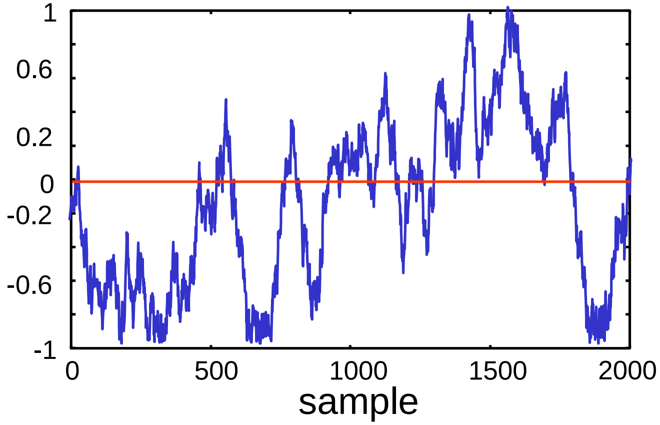


Completely random sequences

- A further generalization of the PRS sequence is assigning to the input variable some random increments within a predefined variability range $[-\Delta u, +\Delta u]$.
- There are few precautions to be respected:
  - limit the maximum variation increment for each disturbance;
  - keep the input variables within the range of consistent operability (*i.e.* lower and upper bounds).

Recommended for NARX and ANN



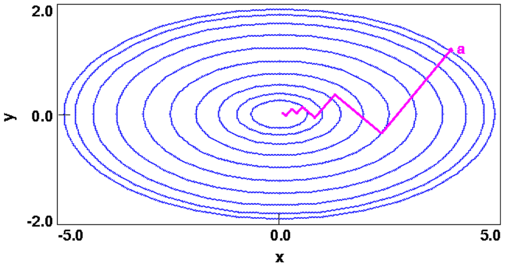


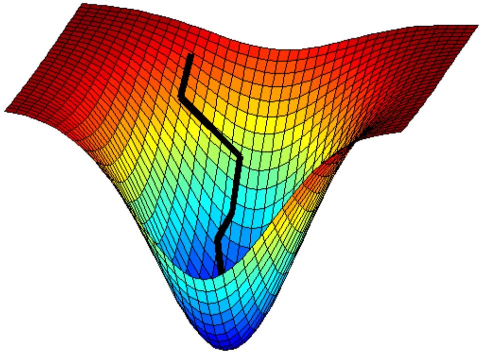
Parameters regression

- The model parameters, $\mathbf{p}$, are evaluated by a regression procedure.
- In case of deterministic signals, the numerical problem to be solved is:

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

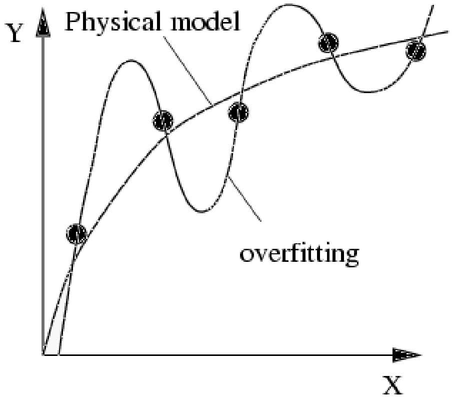
where n_y is the number of output variables and n_s is the sampling number.





Model validation

- Once the model parameters have been evaluated, it is necessary to assess the system quality respect to a data set different from the one used in the identification procedure.
- This is the **cross-validation** procedure.



- In addition, it is necessary to examine any possible system **overfitting** that is an excessive model specificity to the learning data set.
- It is worth testing also the extrapolation capability of the model.
- The validation procedure can be carried out as follows:
  - **One step ahead**: the procedure uses the outputs of the real process step by step;
  - **Predictive mode**: the system (*i.e.* model) outputs are used to carry out the simulations step by step without relying on the real process data (**pure extended simulation**).

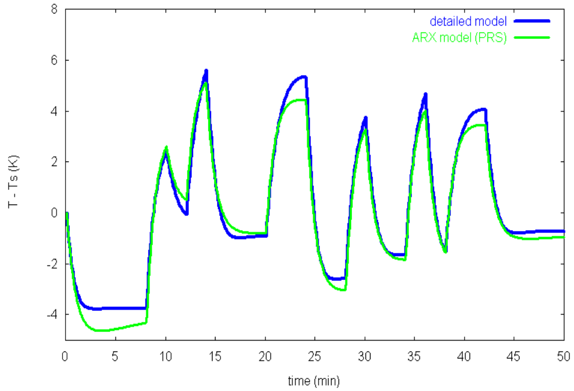
Model validation

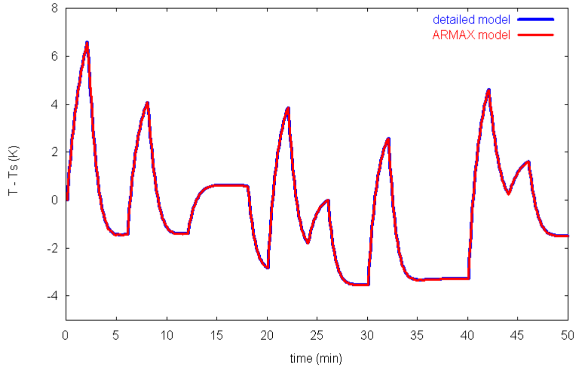
- The goodness of a model is assessed respect to:
  - the capability of reproducing the real process **dynamics**;
  - The model **stability** respect to external disturbances.

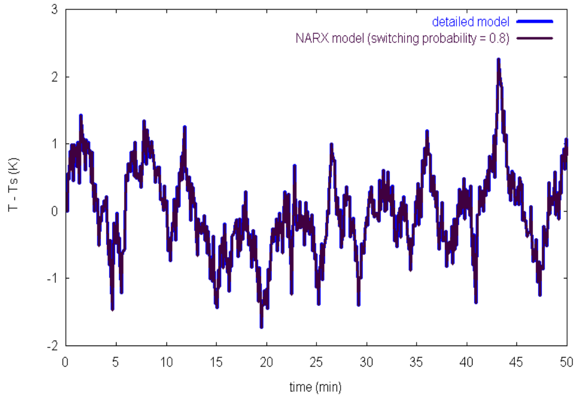
- **CROSS VALIDATION INDEX**: allows to assess the goodness of the model (n_{CVS} = number of validation data)

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

- The more CVI is near to zero, the better the model validation.

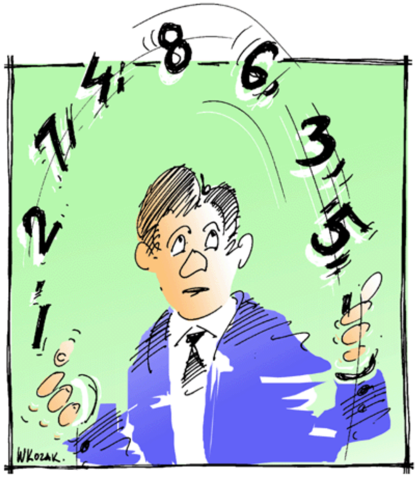






Models comparison

- To identify/define the optimal order of the model (and avoid any overfitting phenomenon) or carry out comparisons among models with different structures (*e.g.*, ARX, ARMAX, NARX, …) it is possible to adopt some **statistical criteria** that account for and find a compromise among:
  - **model order**;
  - **number of input and output variables**;
  - **parameters number**.



- The four most known comparison criteria are:
  - **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**:

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

Models comparison

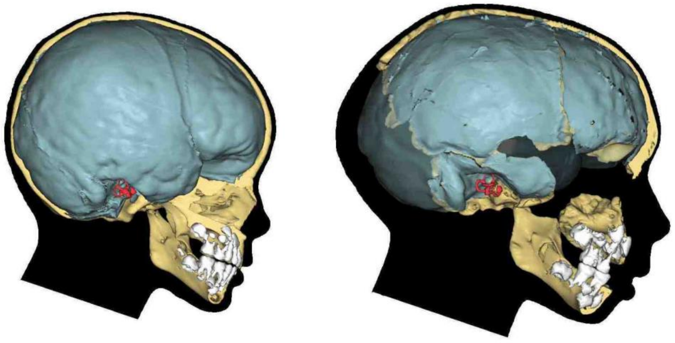
- **BAYESIAN INFORMATION CRITERION**:

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

- **LAW OF ITERATED LOGARITHMS CRITERION**:

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

n_s sampling number, n_p parameters number and $e(i) = y_{real}(i) - y_{system}(i)$



Neanderthal Sapiens

Bibliography

- Billings S.A., W.S.F. Woon, *"A Prediction – Error and Stepwise – Regression Estimation Algorithm for Non-linear Systems"*, Int. J. Control, Vol 44, no 3, pp 803-822, 1986
- Bosch P.P.J, A.C. van der Klauw, *"Modeling, Identification, and Simulation of Dynamical Systems"*, CRC Press, 1994
- Henson M.A, D.E. Seborg, *"Nonlinear Process Control"*, Prentice Hall, New Jersey, 1997
- Hernández E., Y. Arcun, *"Control of Non-linear Systems using Polynomial ARMA Models"*, AIChE, Vol 39, no 3, pp 446-460, 1993
- Hernando D., A. A. Desrochers, *"Modeling of Non-linear Discrete Time Systems from Input–Output Data "*, Automatica, Vol 24, no 5, pp 629-641, 1988
- Juditsky et. al, *"Nonlinear Black-box Modeling in System Identification: Mathematical Foundations"*, Automatica, Vol 31, no 12, pp 1725-1750, 1995
- Leontaritis I.J., S.A. Billings, *"Input-output Parametric Models for Non-linear Systems. Part I: Deterministic Non-linear Systems"*, Int. J. Control, Vol 41, no 2, pp 303-328, 1985
- Leontaritis I.J., S.A. Billings, *"Input-output Parametric Models for Non-linear Systems. Part II: Stochastic Non-linear Systems"*, Int. J. Control, Vol 41, no 2, pp 329-344, 1985
- **Ljung L., *"System Identification: Theory for the User"*, Prentice Hall, 1998**
- Shannon C.E. , "*A Mathematical Theory of Communication*", The Bell System Technical Journal, Vol 27, 379–423, 623–656, 1948
- Sjöberg J., et al., *"Nonlinear Black-box Modeling in System Identification: a Unified Overview"*, Automatica, Vol 31, no 12, pp 1691-1724, 1995
- Sriniwas G.R., et. Al, *"Non-linear Identification and Control of a High-purity Distillation Column"*, J. Proc. Cont., Vol 5, no 3, pp 149-162, 1995