

# SYSTEM IDENTIFICATION AND DATA ANALYSIS

*Silvio Simani*

Office: Building A, 3rd floor, n. 344

Email: [ssimani@ing.unife.it](mailto:ssimani@ing.unife.it)

Phone: +39 0532 97 4844

URL: [www.ing.unife.it/simani](http://www.ing.unife.it/simani)

## General Course Information

- Lectures: Monday, 11:30-13:30, Info Lab or lecture room 9;  
Tuesday, 9:00-11:30, lecture room 9;  
Thursday, 10:30-12:30, lecture room 9.
- Instructor: *Silvio Simani*
- Textbook:  
Lennart Ljung, *System Identification: Theory for the User*, 2nd Edition, Prentice-Hall, 1999 (Book's web page: <http://www.control.isy.liu.se/~ljung/sysid>)
- Reference books:
  1. L. Ljung and T. Glad, *Modeling of Dynamic Systems*, Prentice Hall, 1994
  2. T. Soderstrom and P. Stoica, *System Identification*, Prentice Hall International (UK) Ltd, 1989
- Course web-page:  
[www.ing.unife.it/simani/lessons.html](http://www.ing.unife.it/simani/lessons.html)

# Course Outline

---

1. Introduction and overview on system identification
2. Non-recursive (off-line) identification methods
3. Non-recursive and recursive (on-line) identification methods
4. Recursive identification methods
5. Practical aspects and applications of system identification

## Associated Reading in the Textbook

---

1. Introduction and overview on system identification ([Ch. 1](#); [4.1-4.3](#); [Ch. 6](#))
2. Non-recursive (off-line) identification methods ([Ch. 7](#))
3. Non-recursive and recursive (on-line) identification methods ([Ch. 10](#); [Ch. 11](#))
4. Recursive identification methods ([Ch. 11](#))
5. Practical aspects and applications of system identification ([Ch. 13](#), [14](#), [16](#), [17](#))

# System Identification and Data Analysis

## Lecture 1

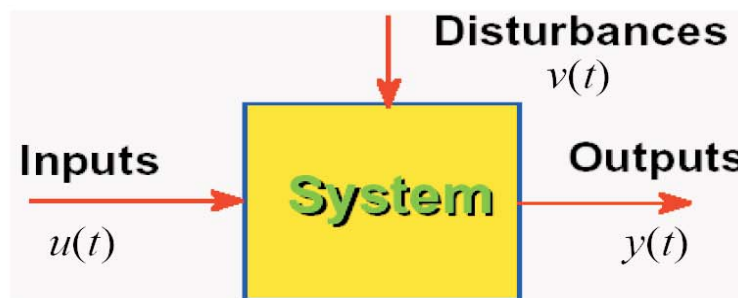
### Introduction and Overview

- What is System Identification (SI)?
- Introduction to systems and models
- Procedure of system identification
- Methods of system identification
- Review on topics covered in course "Automatica I (Laboratorio)"
- Examples of system identification

## System Identification

"Identification is the determination, on the basis of input and output, of a system within a specified class of systems, to which the system under test is equivalent."

- L. Zadeh, (1962)



System identification is the field of *modeling* dynamic systems from *experimental data*

# Systems

**System:** A collection of components which are coordinated together to perform a function.

A system is a defined part of the real world. Interactions with the environment are described by inputs, outputs, and disturbances.

**Dynamic system:** A system with a memory, i.e., the input value at time  $t$  will influence the output at future instants.

**Examples of dynamic system:** (pp. 2-6, textbook)

- Example 1.1 A Solar-Heated House
- Example 1.2 A Military Aircraft
- Example 1.3 Speech

## Ex. A Solar Heated House

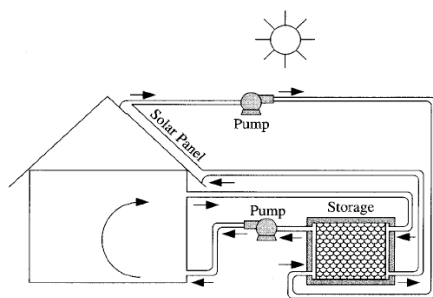


Figure 1.2 A solar-heated house.

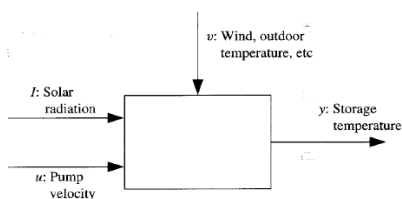
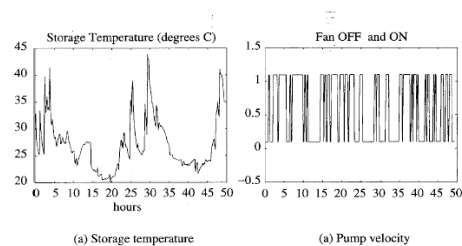
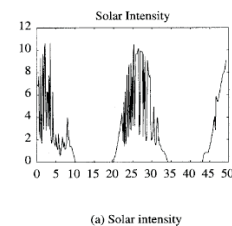


Figure 1.3 The solar-heated house system:  $u$ : input;  $I$ : measured disturbance;  $y$ : output;  $v$ : unmeasured disturbances.



(a) Storage temperature (b) Pump velocity



(a) Solar intensity

Figure 1.4 Storage temperature  $y$ , pump velocity  $u$ , and solar intensity  $I$  over a 50-hour period. Sampling interval: 10 minutes.

## Ex. Speech

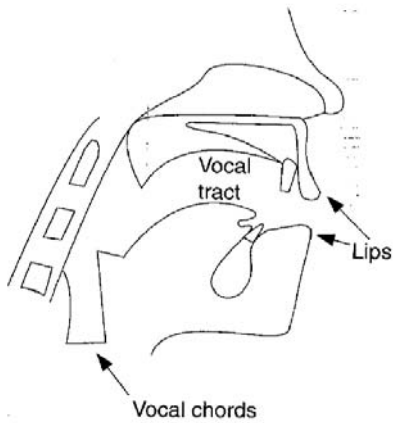


Figure 1.7 Speech generation.

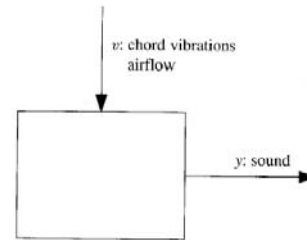


Figure 1.8 The speech system:  $y$ : output;  $v$ : unmeasured disturbance.

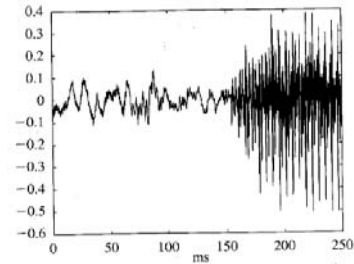
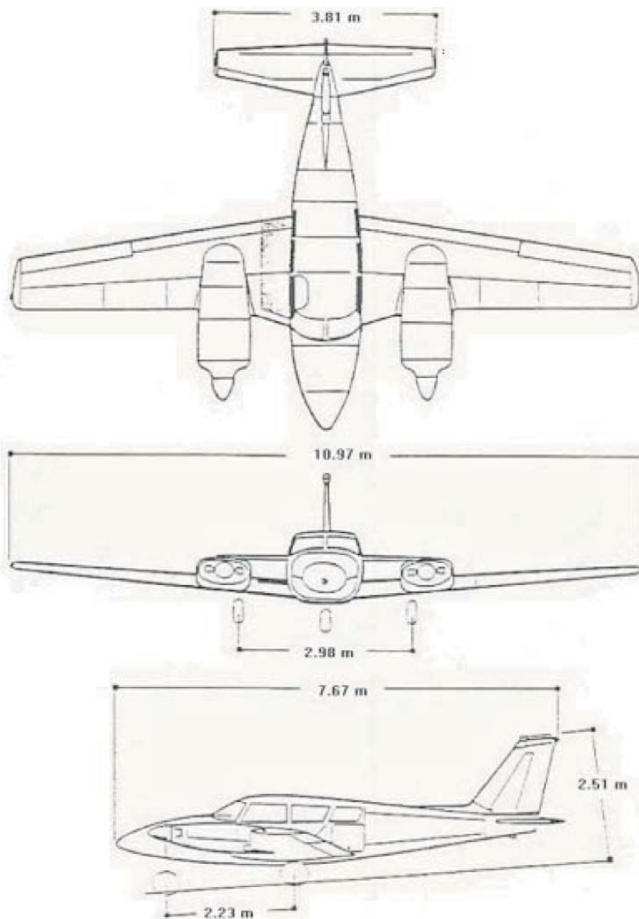


Figure 1.9 The speech signal (air pressure). Data sampled every 0.125 ms. (8 kHz sampling rate).



## Aircraft Model

Symbol	Sensor Variable
$\delta_e$	Elevator deflection angle
$\delta_a$	Aileron deflection angle
$\delta_r$	Rudder deflection angle
$\delta_{th}$	Throttle aperture %
$V$	True Air Speed
$Q$	Pitch Rate
$\theta$	Elevation Angle
$H$	Altitude
$P$	Roll Rate
$R$	Yaw Rate
$\phi$	Bank Angle
$\psi$	Heading Angle
$n$	Engine Angular Rate

# Aircraft Mathematical Model

$$\dot{V} = F_x \frac{\cos \alpha \cos \beta}{m} + F_y \frac{\sin \beta}{m} + F_z \frac{\sin \alpha \cos \beta}{m}$$

$$\dot{\alpha} = \frac{-F_x \sin \alpha + F_z \cos \alpha}{mV \cos \beta} + Q - (P \cos \alpha + R \sin \alpha) \tan \beta$$

$$\dot{\beta} = \frac{-F_x \cos \alpha \sin \beta + F_y \cos \beta - F_z \sin \alpha \sin \beta}{mV} + P \sin \alpha - R \cos \alpha$$

$$\dot{P} = \frac{M_x I_z + M_z I_{xz} + PQ I_{xz} (I_x - I_y + I_z)}{I_x I_z - I_{xz}^2} + \frac{QR (I_y I_z - I_{xz}^2 - I_z^2)}{I_x I_z - I_{xz}^2}$$

$$\dot{Q} = \frac{M_y + PR (I_z - I_x) - P^2 I_{xz} + R^2 I_{xz}}{I_y}$$

$$\dot{R} = \frac{M_x I_{xz} + M_z I_x + PQ (I_x^2 - I_x I_y + I_{xz}^2)}{I_x I_z - I_{xz}^2} + \frac{QR I_{xz} (-I_x + I_y - I_z)}{I_x I_z - I_{xz}^2}$$

$$\dot{\phi} = P + Q \sin \phi \tan \theta + R \cos \phi \tan \theta$$

$$\dot{\theta} = Q \cos \phi - R \sin \phi$$

$$\dot{\psi} = \frac{Q \sin \phi + R \cos \phi}{\cos \theta}$$

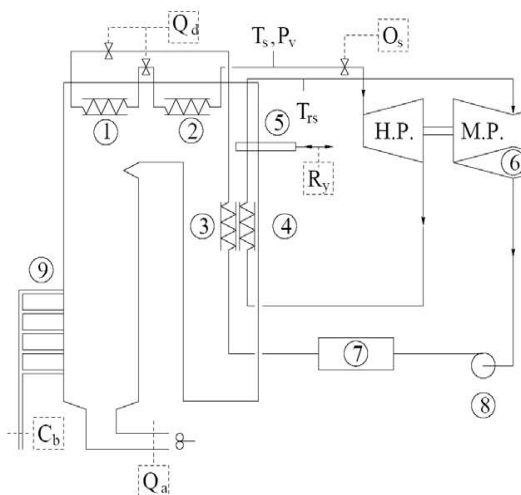
$$\dot{H} = V \cos \alpha \cos \beta \sin \theta - V \cos \theta (\sin \beta \sin \phi + \sin \alpha \cos \beta \cos \phi) - V_{Az}$$

# 120 MW Power Plant "Pont sur Sambre"

## Process Description



3 major components: the reactor, turbine, & condenser



- $u_1(t)$ :  $C_b$  gas flow
- $u_2(t)$ :  $O_s$  turbine valves opening
- $u_3(t)$ :  $Q_d$  super heater spray flow
- $u_4(t)$ :  $R_y$  gas dampers
- $u_5(t)$ :  $Q_a$  air flow

- $y_1(t)$ :  $P_v$  steam pressure
- $y_2(t)$ :  $T_s$  main steam temperature
- $y_3(t)$ :  $T_{rs}$  reheat steam temperature

# Models

---

**Model:** A description of the system. The model should capture the essential information about the system.

<b>Systems</b>	<b>Models</b>
Complex  Building/Examining systems is expensive, dangerous, time consuming, etc.	Approximative (However, model should capture the relevant information of the system)  Models can answer many questions about the system.

## Types of Models

---

- Mental, intuitive or verbal models
  - e.g., driving a car
- Graphs and tables
  - e.g., Bode plots and step responses
- Mathematical models
  - e.g., differential and difference equations, which are well-suited for modeling dynamic systems

# Mathematical Models and Benefits

- Do not require a physical system
  - Can treat new designs/technologies without prototype
  - Do not disturb operation of existing system
- Easier to work with than real world
  - Easy to check many approaches, parameter values, ...
  - Flexible to time-scales
  - Can access un-measurable quantities
- Support safety
  - Experiments may be dangerous
  - Operators need to be trained for extreme situations
- Help to gain insight and better understanding

## Mathematical Models

### Model descriptions

- Transfer functions
- State-space models
- Block diagrams

### Notation for continuous-time and discrete-time models

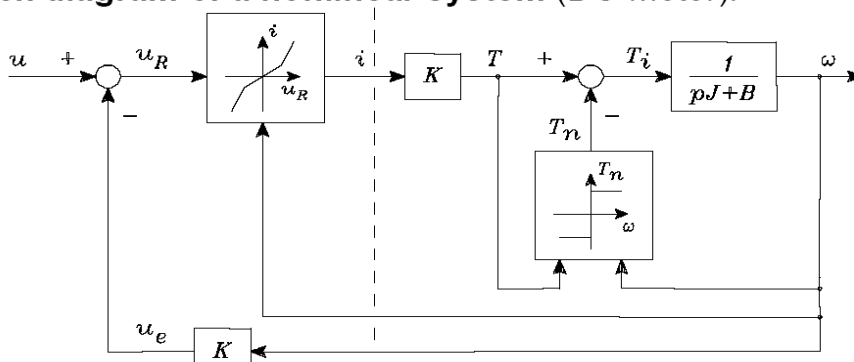
Complex Laplace variable  $s$  and differential operator  $p$ :

$$\dot{x}(t) = \partial x(t) / \partial t = px(t)$$

Complex z-transform variable  $z$  and shift operator  $q$ :

$$x(k+1) = qx(k)$$

### Block diagram of a nonlinear system (DC-motor):





# Type of Models and System Modeling

---

## Models

mathematical – other  
parametric – nonparametric  
continuous-time – discrete-time  
input/output – state-space  
linear – nonlinear  
dynamic – static  
time-invariant – time-varying  
SISO – MIMO

## Modeling/System Identification

theoretical (physical) – experimental  
white-box – grey-box – black-box  
structure determination – parameter estimation  
time-domain – frequency-domain  
direct – indirect

# Types of Models

## - Parametric and Non-parametric Models

---

Many approaches to system identification, depending on model class

- linear/nonlinear
- parametric/nonparametric

Non-parametric methods try to estimate a generic model of a signal or system.

– step responses, impulse responses, frequency responses, etc.

Parametric methods estimate parameters in a user-specified model

– parameters in transfer functions, state-space matrices of given order, etc.

# Types of Models

## - Linear and Nonlinear Models

---

The system identification methods are characterized by model type:

**A. Linear discrete-time model:** Classical system identification

**B. Neural network:** Strongly non-linear systems with complicated structures – no relation to the actual physical structures/parameters (will not be covered)

**C. General simulation model:** Any mathematical model, that can be simulated e.g. with Matlab\Simulink. It requires a realistic physical model structure, typically developed by theoretical modelling

## Types of Models – Cont'd

---

Models can also be classified according to purpose:

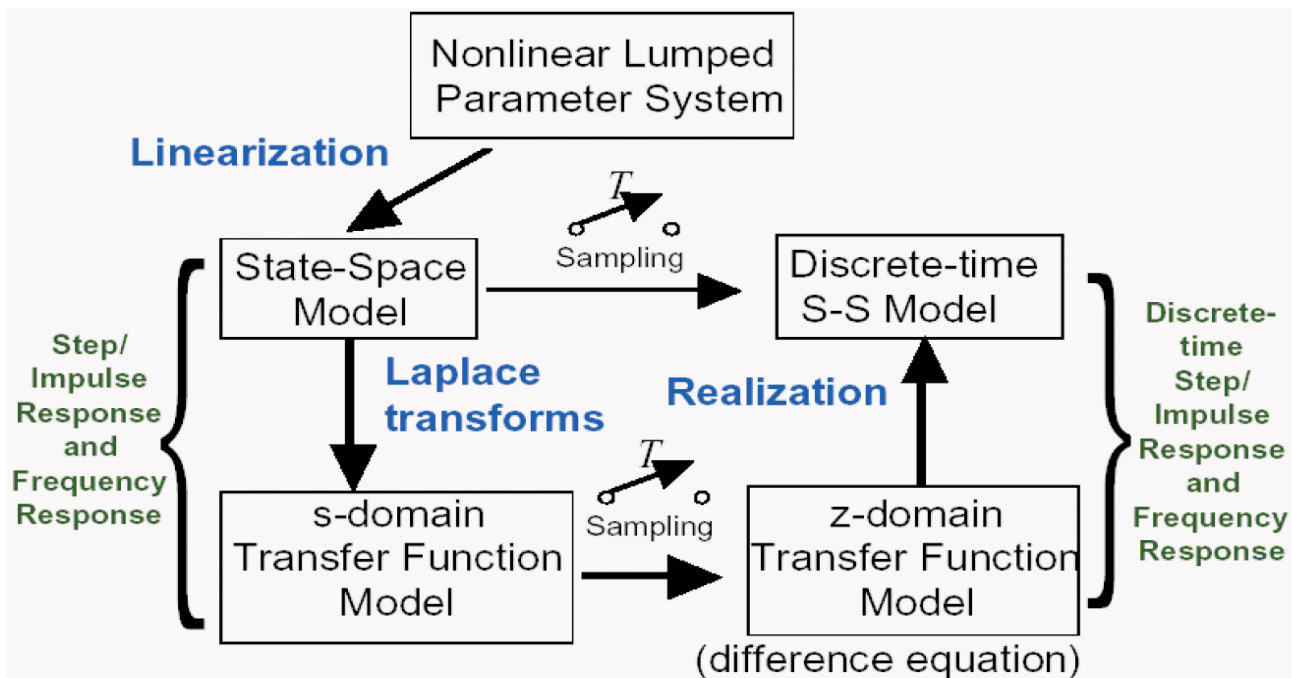
- **Models to assist plant design and operation**

- Detailed, physically based, often **non-dynamic models** to assist in fixing plant dimensions and other basic parameters
- Economic models allowing the size and product mix of a projected plant to be selected
- Economic models to assist decisions on plant renovation

- **Models to assist control system design and operation**

- Fairly complete **dynamic model**, valid over a wide range of process operation to assist detailed quantitative design of a control system
- Simple models based on crude approximation to the plant, but including some economically quantifiable variables, to allow the scope and type of a proposed control system to be decided
- Reduced dynamic models for use on-line as part of a control system

# Systems/Models Representations



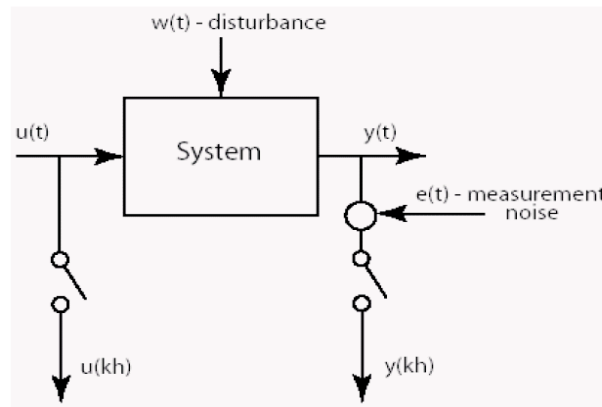
## How to Build Mathematical Models?

Two basic approaches:

- **Physical modeling**
  - Use first principles, laws of nature, etc. to model components
  - Need to understand system and master relevant facts!
- **System identification - Experimental modeling**
  - Use experiments and observations to deduce model
  - Need prototype or real system!

# Principle of System Identification

**Basic Idea:** estimate system from measurement of  $u(t)$  and  $y(t)$

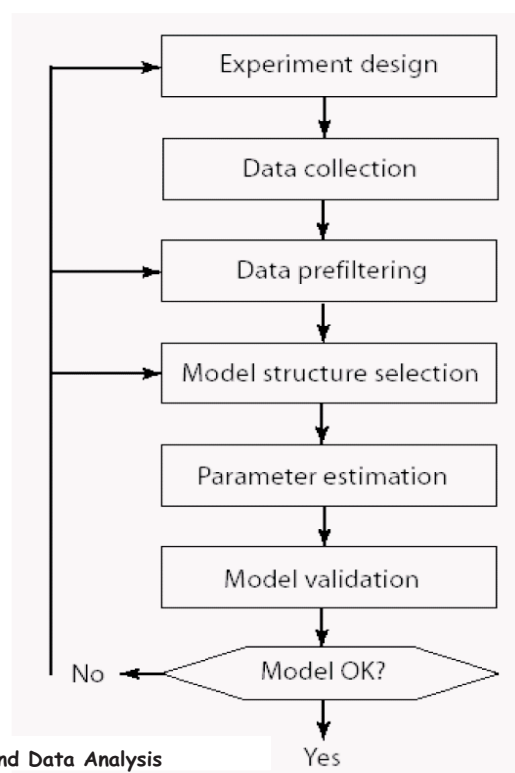


Issues:

- Choice of sampling frequency, input signal (experimental conditions)
- What class of models – how to model disturbances?
- Estimating model parameters from sampled, finite and noisy data

## Procedure of System Identification

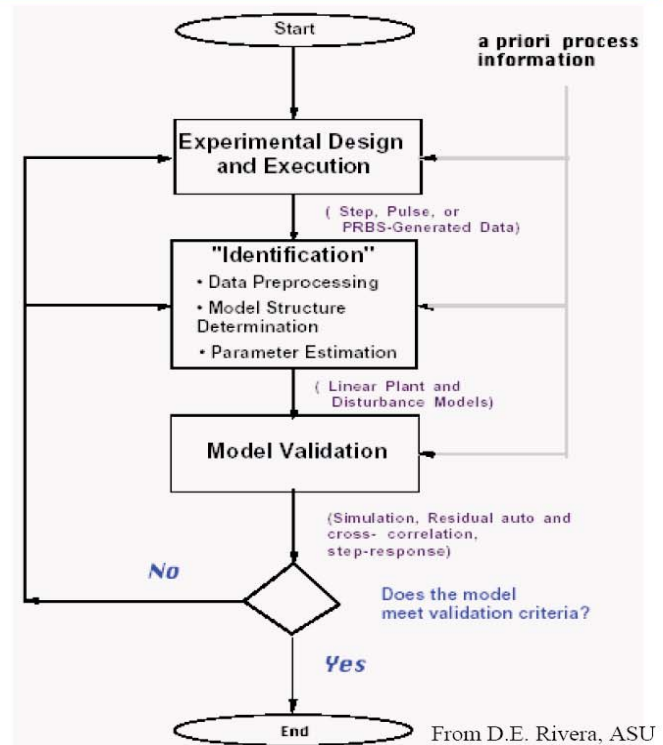
- Experiment design and data collection
- Data preprocessing
- Model structure selection
- Parameter estimation
- Model validation



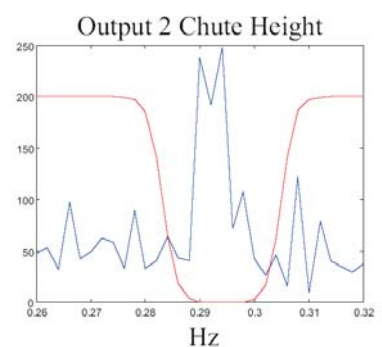
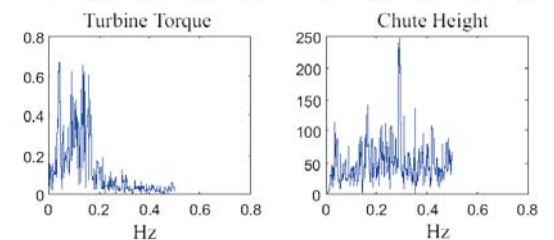
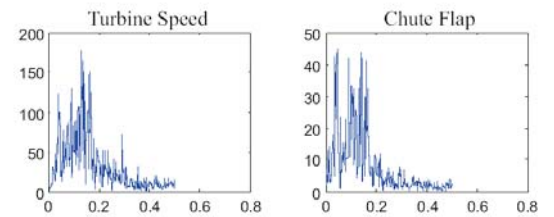
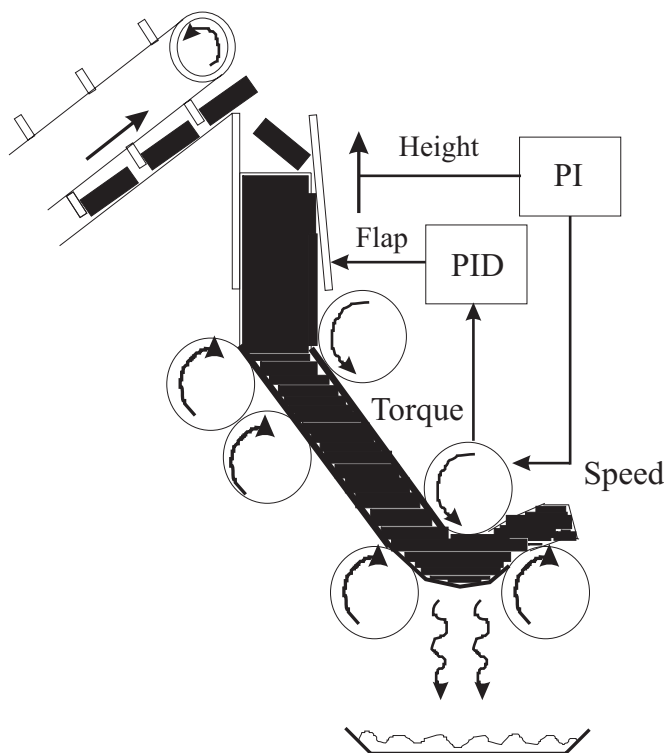
**An iterative procedure !**

# Procedure of System Identification – I

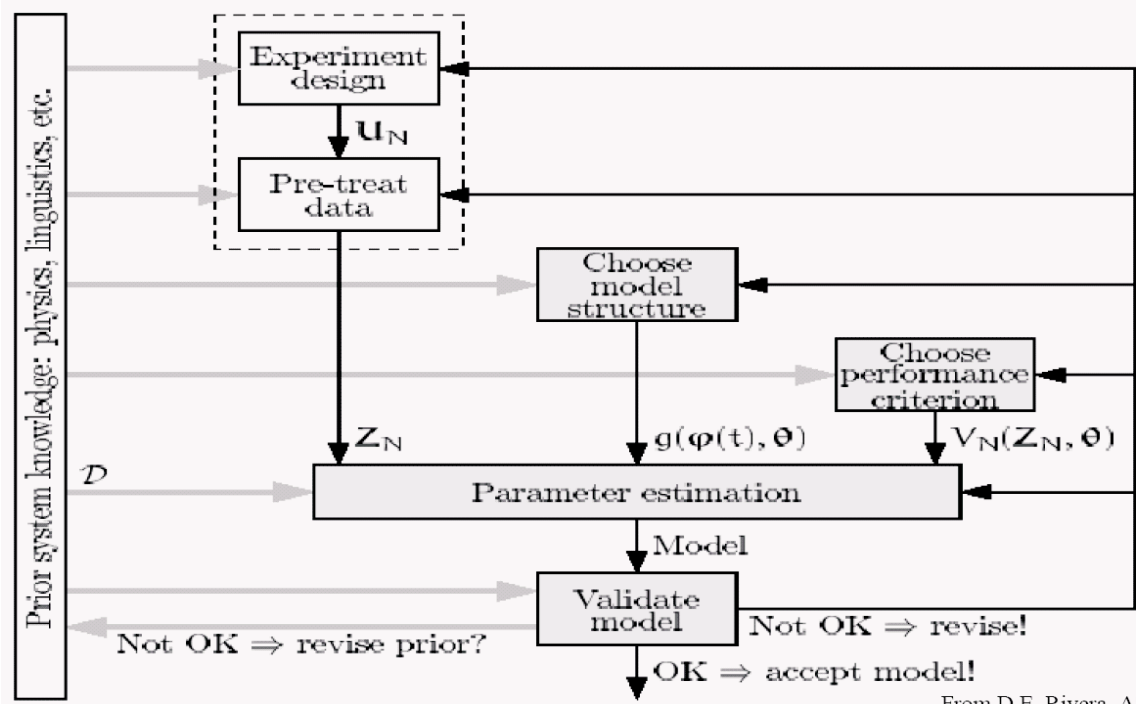
- Experimental design and execution
- Data preprocessing
- Model structure determination
- Parameter estimation
- Model validation



## Sugar Cane Crushing Process



# Procedure of System Identification – II



From D.E. Rivera, ASU;  
Originally from P. Lindskog

## Experiments and Data Collection

Often good to use a two-stage approach

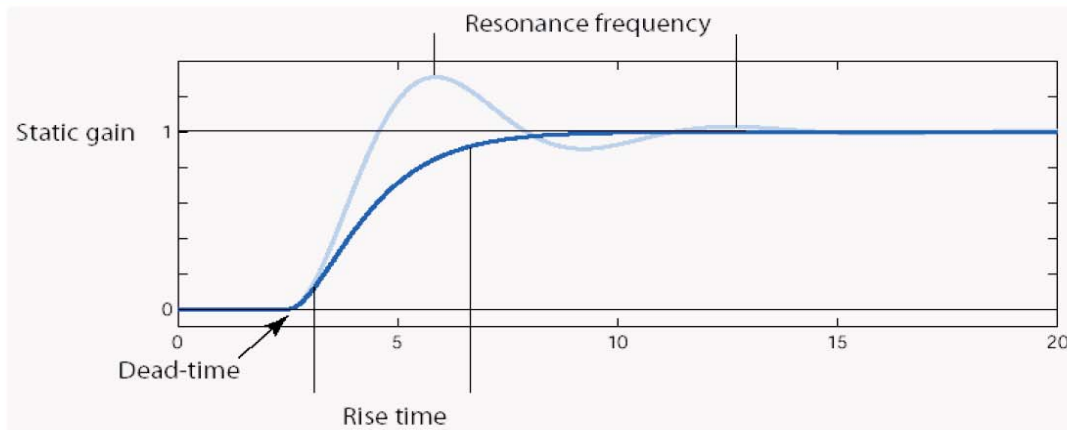
### 1. Preliminary experiments

- step/impulse response tests to get basic understanding of system dynamics
- linearity, static gains, time delays, time constants, sampling interval

### 2. Data collection for model estimation

- carefully designed experiment to enable good model fit
- operating point, input signal type, number of data points to collect, etc.

# Preliminary Experiments: Step Response Experiment



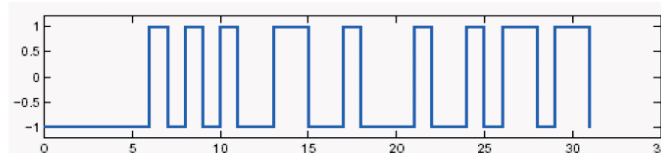
Useful for obtaining qualitative information about system

- Indicates dead-times, static gain, time constants and resonance frequency etc.
- Aids sampling time selection (rule-of-thumb: 4-10 sampling points over the rise time)

## Designing Experiment for Model Estimation

### Input signal should excite all relevant frequencies

- estimated model are more accurate in frequency ranges where input has high energy
- a good choice is often a binary sequence with random “hold times” (e.g., PRBS – Pseudo-Random Binary Sequence)

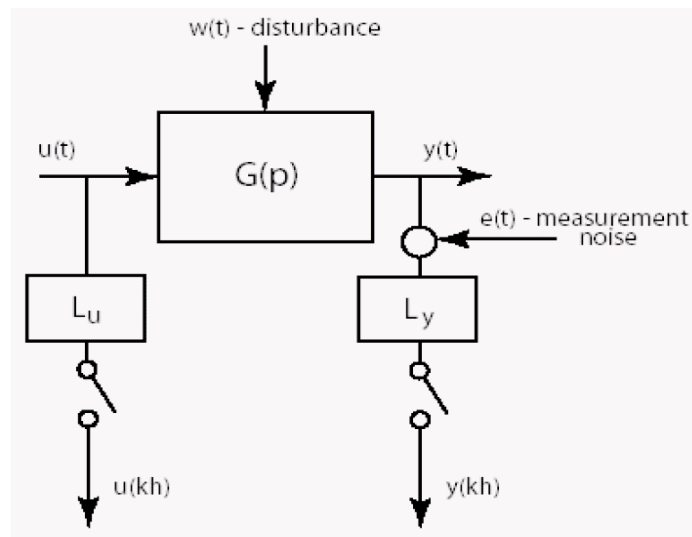


### Trade-off in selection of signal amplitude

- large amplitude gives high signal-to-noise ratio (SNR), low parameter variance
- most systems are nonlinear for large input amplitudes

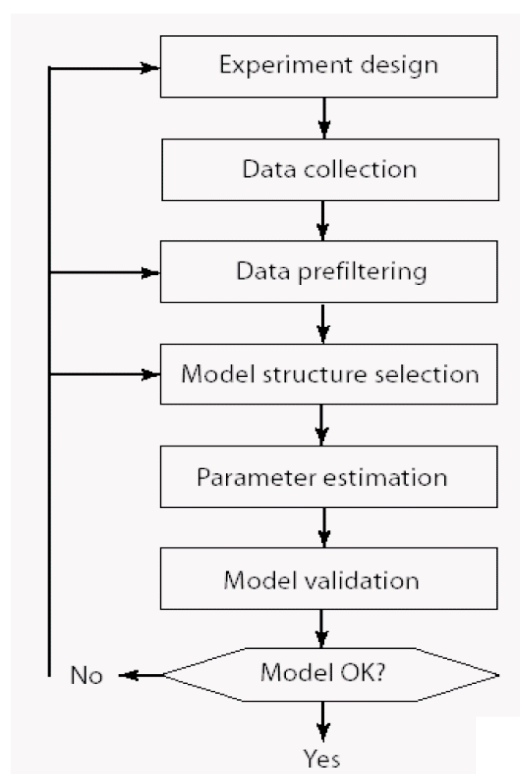
**Many pitfalls if estimating a model of a system under closed-loop control !**

# Data Collection



Sampling time selection and anti-alias filtering are central !

# Procedure of System Identification



**An iterative procedure !**



# Prefiltering of Data

## Remove

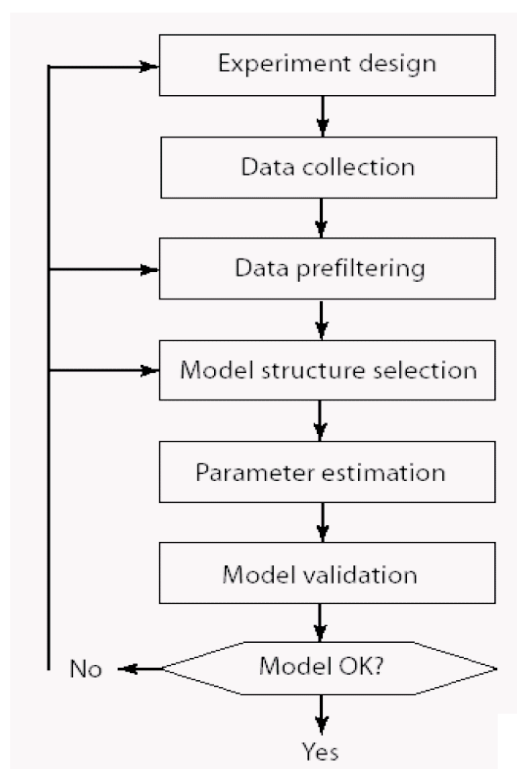
- transients needed to reach desired operating point
- mean values of input and output signals, *i.e.*, work with

$$\Delta u[t] = u[t] - \frac{1}{N} \sum_{t=1}^N u[t]$$

$$\Delta y[t] = y[t] - \frac{1}{N} \sum_{t=1}^N y[t]$$

- trends (use `detrend` in MATLAB)
- outliers (“obviously erroneous data points”)

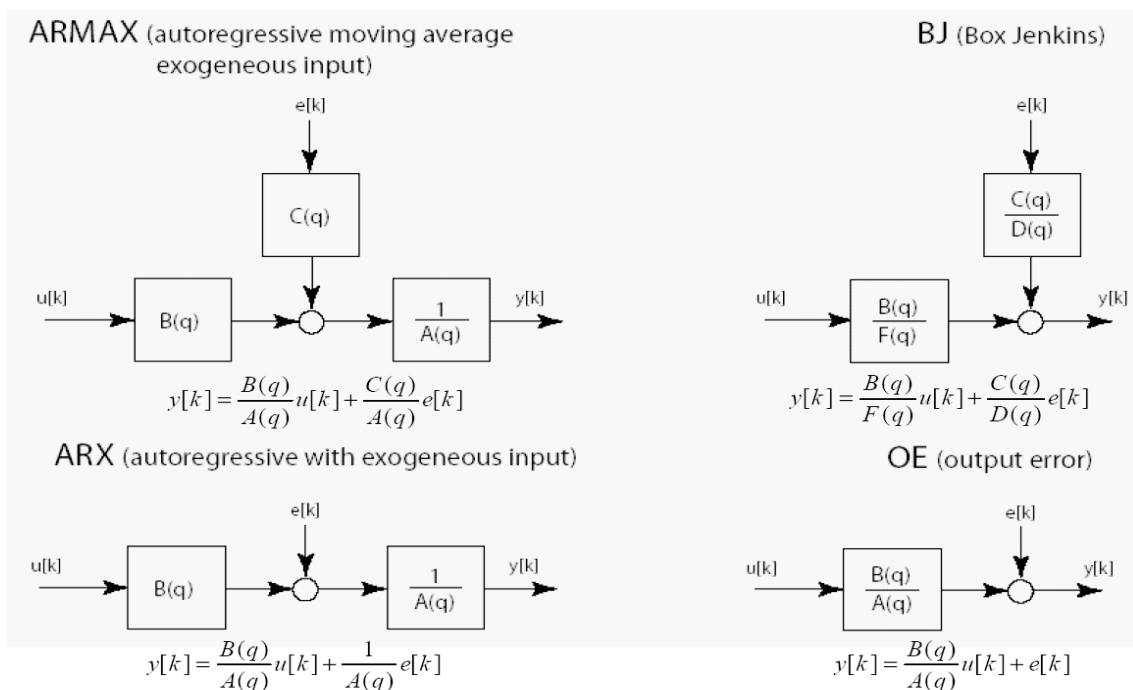
# Procedure of System Identification



**An iterative procedure !**

# Model Structures

Model structures commonly used (BJ includes all others as special cases)



## Model Structures - Cont'd

### • Model structures Based on Input-Output

Model	$\tilde{p}(q)$	$\tilde{p}_e(q)$
ARX	$\frac{B(q)}{A(q)}$	$\frac{1}{A(q)}$
ARMAX	$\frac{B(q)}{A(q)}$	$\frac{C(q)}{A(q)}$
FIR	$B(q)$	1
Box-Jenkins	$\frac{B(q)}{F(q)}$	$\frac{C(q)}{D(q)}$
Output Error	$\frac{B(q)}{F(q)}$	1

$$A(q)y[k] = \frac{B(q)}{F(q)}u[k] + \frac{C(q)}{D(q)}e[k] \quad \text{or} \quad y[k] = \tilde{p}(q)u[k] + \tilde{p}_e(q)e[k]$$

### • Model structures Based on State-Space Representation

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] & \text{or} & \quad x[k+1] = A(\theta)x[k] + B(\theta)u[k] \\ y[k+1] &= Cx[k+1] + Du[k+1] & & \quad y[k+1] = C(\theta)x[k+1] + D(\theta)u[k+1] \end{aligned}$$

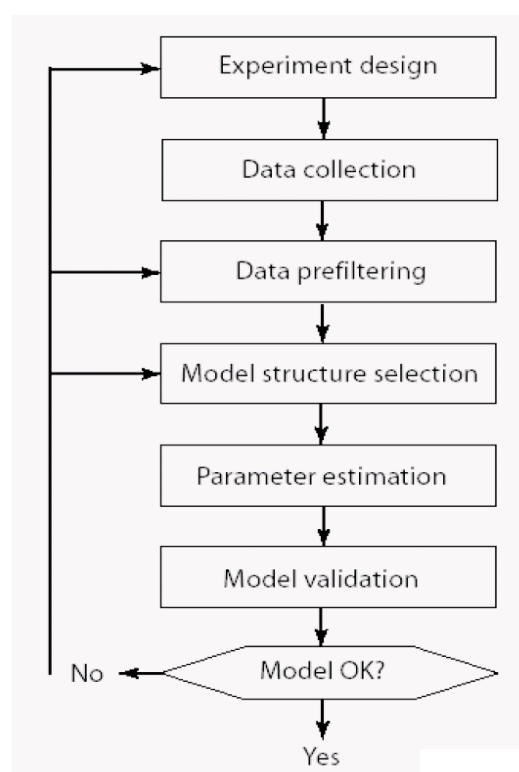
# Choice of Model Structure

1. Start with non-parametric estimates (correlation analysis, spectral estimation)
  - give information about model order and important frequency regions
2. Prefilter input/output data to emphasize important frequency ranges
3. Begin with ARX (AutoRegressive with eXogeneous input) models
4. Select model orders via
  - cross-validation (simulate model and compare with new data)
  - Akaike's Information Criterion (AIC), *i.e.*, pick the model that minimizes

$$\left(1 + 2 \frac{d}{N}\right) \sum_{t=1}^N \varepsilon[t; \theta]^2$$

(where  $d$  is the number of estimated parameters in the model)

# Procedure of System Identification



**An iterative procedure !**

# Nonparametric Estimation Methods

---

## Nonparametric methods

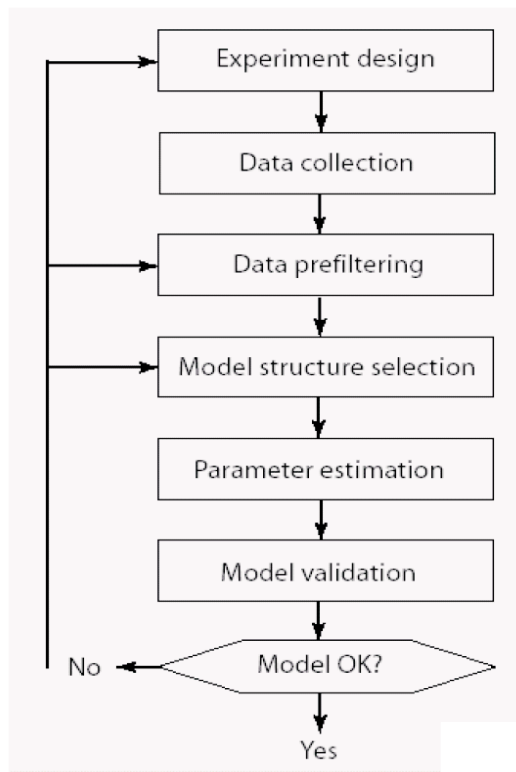
- Transient response
  - Correlation analysis
  - Frequency responses analysis and Fourier analysis
  - Spectral analysis
- Discussed in the “Automatica I (Laboratorio)” course, will not be elaborated further in this course

# Parametric Estimation Methods

---

- **Non-recursive/Batch (off-line) methods**
  - Linear regression and (block) least squares methods
  - Prediction error methods
  - Instrumental variable methods
  - Subspace methods *(If possible, few details)*
- **Recursive (on-line) methods**
  - Recursive Least Squares (RLS) methods
  - Forgetting factor techniques and time-varying systems identification methods

# Procedure of System Identification



**An iterative procedure !**

Lecture 1

Lecture Notes on System Identification and Data Analysis

35

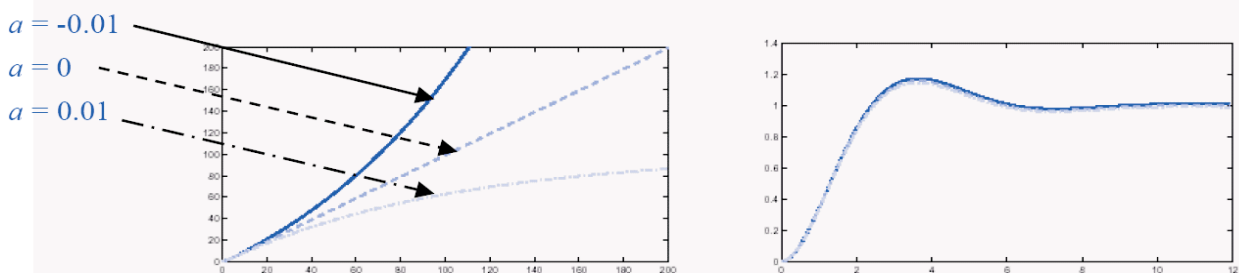
## Model Validation

A critical evaluation: “is model good enough”?  
– typically depends on the purpose of the model

### Example

$$G(s) = \frac{1}{(s+1)(s+a)}$$

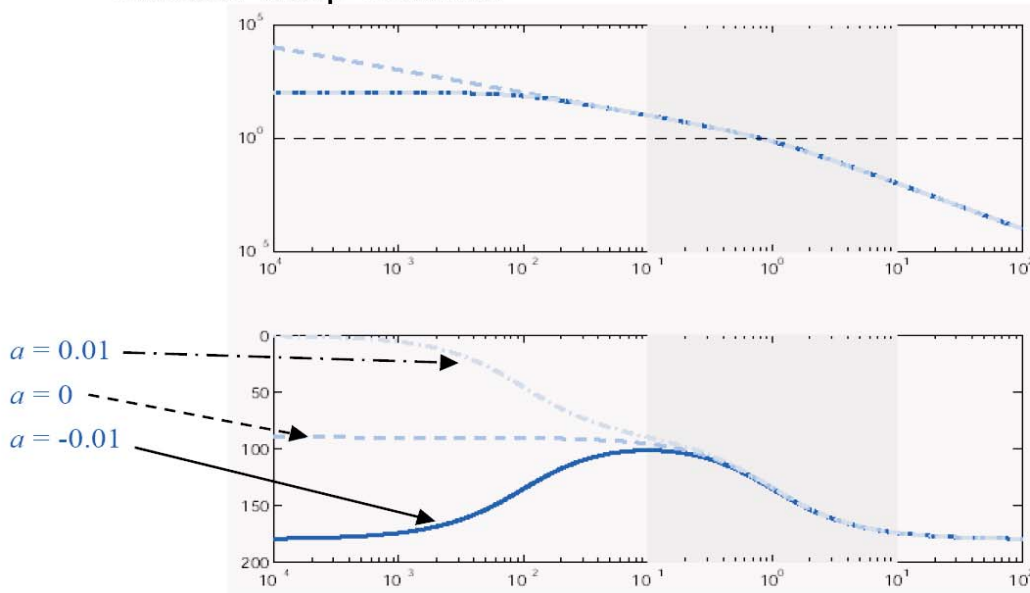
Open- and closed-loop responses for  $a = -0.01, 0, 0.01$



Insufficient for open-loop prediction, good enough for closed-loop control.

# Model Validation – cont'd

- Bode diagrams reveal why model is good enough for closed-loop control



- Different low-frequency behavior, similar responses around cross-over frequency

Lecture 1

Lecture Notes on System Identification and Data Analysis

37

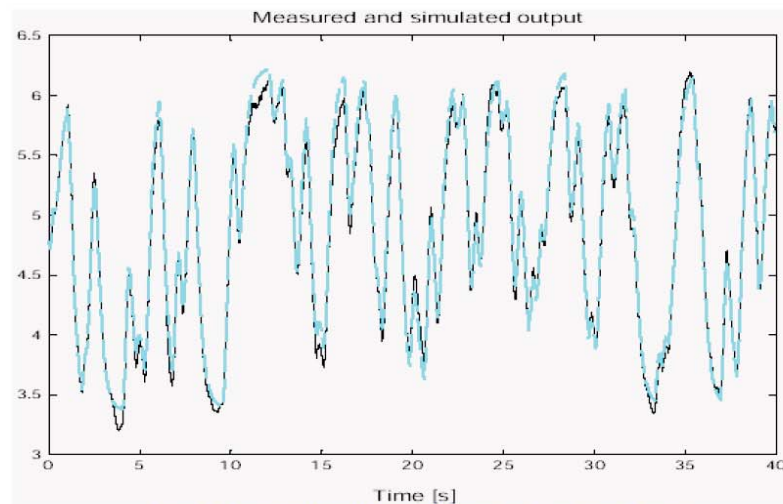
## Principle of Model Validation

1. Compare model simulation/prediction with real data – **in time domain**
2. Compare estimated model's frequency response and spectral analysis result – **in frequency domain**
3. Perform statistical tests on prediction errors

# Validation: simulation and prediction

---

- Split data into two parts: one for estimation and one for validation
- Apply input signal in validation data set to estimated model
- Compare simulated output with output stored in validation data set



## Statistical Model Validation

---

If we fit the parameters of the model

$$y[t] = G(q; \theta)u[t] + H(q; \theta)e[t]$$

to data, the *residuals*

$$\varepsilon[t] = H(q; \theta)^{-1} \{y[t] - G(q; \theta)u[t]\}$$

represent a disturbance that explains mismatch between model and observed data.

If the model is correct, the residuals should be

- white, and
- uncorrelated with  $u$

# Statistical Model Validation – cont'd

To test if the residuals  $\varepsilon[t]$  are **white**, we compute the auto-covariance function

$$\hat{R}_\varepsilon(\tau) = \frac{1}{N} \sum_{t=1}^N \varepsilon[t] \varepsilon[t + \tau]$$

and verify that its components lie within a 95% confidence region around zero.

- large components indicate un-modelled dynamics

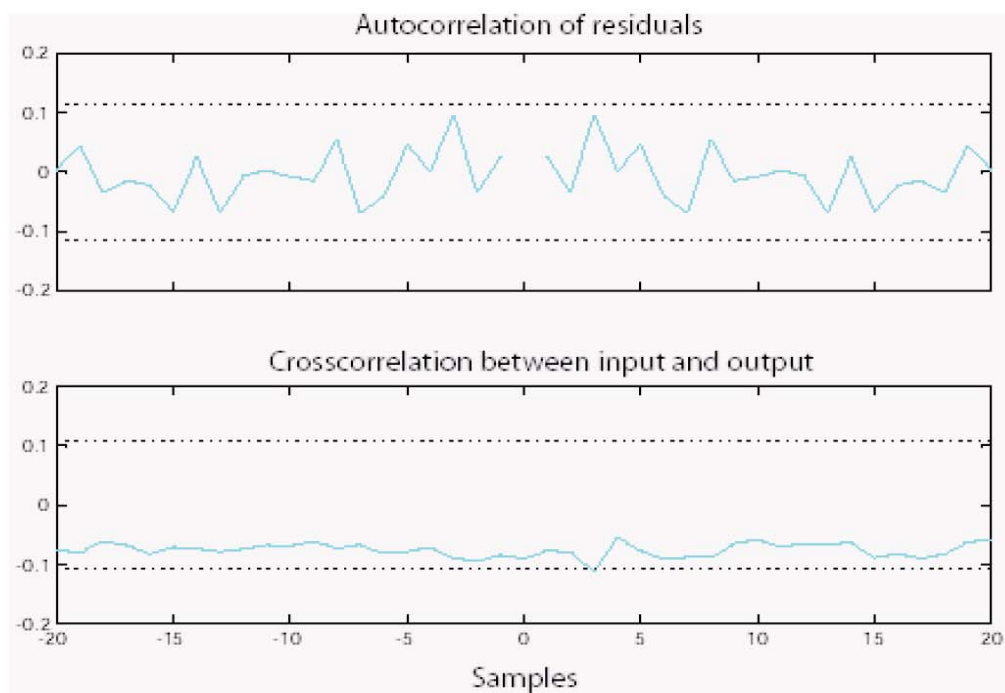
**Independence** tested by verifying that cross-correlation function

$$\hat{R}_{\varepsilon u}(\tau) = \frac{1}{N} \sum_{t=1}^N \varepsilon[t + \tau] u[t]$$

lie within a 95% confidence region around zero.

- large components indicate un-modelled dynamics,
- $\hat{R}_{\varepsilon u}(\tau)$  nonzero for  $\tau < 0$  (non-causality) indicate the presence of feedback

# Statistical Model Validation – cont'd





# Software Tools

## - MATLAB Toolbox: System Identification

---

>> help ident

System Identification Toolbox.

Version 5.0.1 (R12.1) 18-May-2001

### Simulation and prediction.

- predict - M-step ahead prediction.
- pe - Compute prediction errors.
- sim - Simulate a given system.

### Data manipulation.

- iddata - Construct a data object.
- detrend - Remove trends from data sets.
- idfilt - Filter data through Butterworth filters.
- idinput - Generates input signals for identification.
- merge - Merge several experiments.
- misdata - Estimate and replace missing input and output data.
- resample - Resamples data by decimation and interpolation.

# Software Tools

## - MATLAB Toolbox: System Identification – cont'd

---

### Nonparametric estimation.

- covf - Covariance function estimate for a data matrix.
- cra - Correlation analysis.
- etfe - Empirical Transfer Function Estimate and Periodogram.
- impulse - Direct estimation of impulse response.
- spa - Spectral analysis.
- step - Direct estimation of step response.

### Parameter estimation.

- ar - AR-models of signals using various approaches.
- armax - Prediction error estimate of an ARMAX model.
- arx - LS-estimate of ARX-models.
- bj - Prediction error estimate of a Box-Jenkins model.
- ivar - IV-estimates for the AR-part of a scalar time series.
- iv4 - Approximately optimal IV-estimates for ARX-models.
- n4sid - State-space model estimation using a sub-space method.
- oe - Prediction error estimate of an output-error model.
- pem - Prediction error estimate of a general linear model.

# Software Tools

## - MATLAB Toolbox: System Identification – cont'd

---

### Model structure creation.

- idpoly - Construct a model object from given polynomials.
- idss - Construct a state space model object.
- idarx - Construct a multivariable ARX model object.
- idgrey - Construct a user-parameterized model object.

### Model conversions.

- arxdata - Convert a model to its ARX-matrices (if applicable).
  - polydata - Polynomials associated with a given model.
  - ssdata - IDMODEL conversion to state-space.
  - tfdata - IDMODEL conversion to transfer function.
  - zpkdata - Zeros, poles, static gains and their standard deviations.
  - idfrd - Model's frequency function, along with its covariance.
  - idmodred - Reduce a model to lower order.
  - c2d, d2c - Continuous/discrete transformations.
  - ss, tf, zpk, frd - Transformations to the LTI-objects of the CSTB.
- Most CSTB conversion routines also apply to the model objects of the Identification Toolbox.

# Software Tools

## - MATLAB Toolbox: System Identification – cont'd

---

### Model presentation.

- bode - Bode diagram of a transfer function or spectrum (with uncertainty regions).
- ffplot - Frequency functions (with uncertainty regions).
- plot - Input - output data for data objects.
- present - Display the model with uncertainties.
- pzmap - Zeros and poles (with uncertainty regions).
- nyquist - Nyquist diagram of a transfer function (with uncertainty regions).
- view - The LTI viewer (with the Control Systems Toolbox for model objects).

### Model validation.

- compare - Compare the simulated/predicted output with the measured output.
- pe - Prediction errors.
- predict - M-step ahead prediction.
- resid - Compute and test the residuals associated with a model.
- sim - Simulate a given system (with uncertainty).

### Model structure selection.

- aic, fpe - Compute Akaike's information and final prediction criteria
- arxstruc - Loss functions for families of ARX-models.
- selstruc - Select model structures according to various criteria.
- struc - Typical structure matrices for ARXSTRUC.

# Software Tools

## - MATLAB Toolbox: System Identification – cont'd

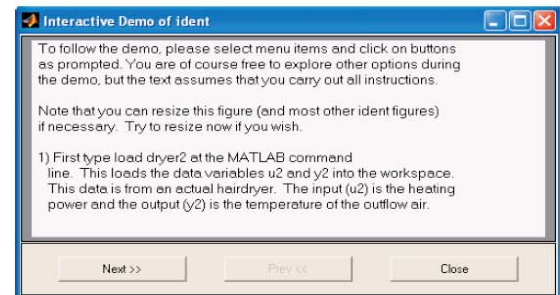
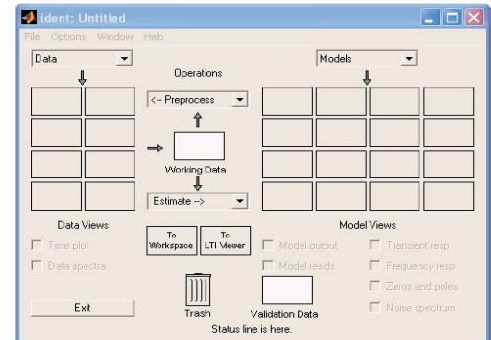
Practice yourself using Matlab System Identification toolbox demonstrations: “iddemo”

```
>> iddemo
```

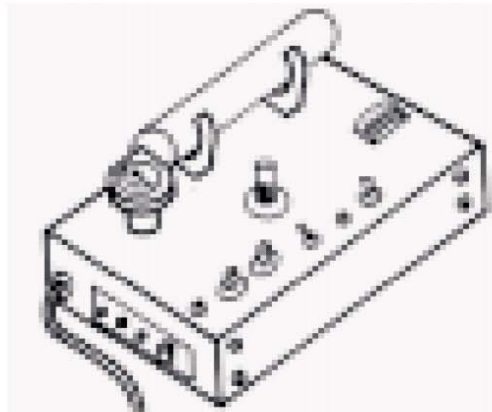
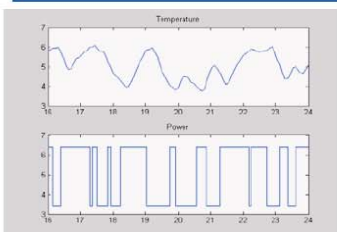
The SYSTEM IDENTIFICATION TOOLBOX is an analysis module that contains tools for building mathematical models of dynamical systems, based upon observed input-output data. The toolbox contains both PARAMETRIC and NON-PARAMETRIC MODELING methods.

Identification Toolbox demonstrations:

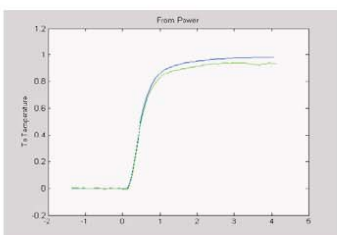
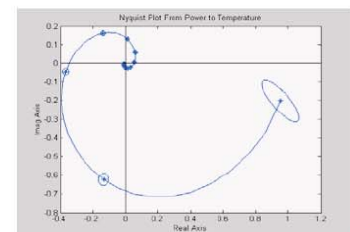
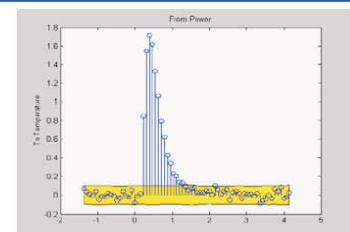
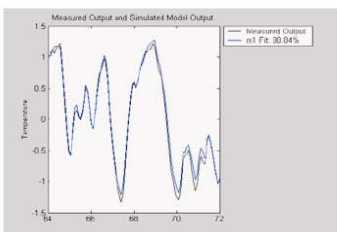
- 1) The Graphical User Interface (ident): A guided Tour.
- 2) Build simple models from real laboratory process data.
- 3) Compare different identification methods.
- 4) Data and model objects in the Toolbox.
- 5) Dealing with multivariable systems.
- 6) Building structured and user-defined models.
- 7) Model structure determination case study.
- 8) How to deal with multiple experiments.
- 9) Spectrum estimation (Marple's test case).
- 10) Adaptive/Recursive algorithms.
- 11) Use of SIMULINK and continuous time models.
- 12) Case studies.



## A System Identification Example: Hairdryer

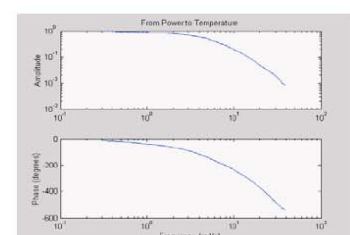


Feedback's Process Trainer PT326



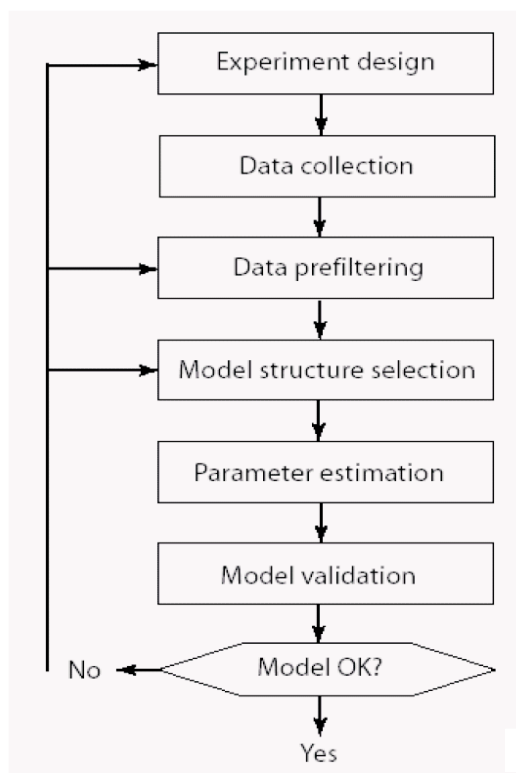
"Hairdryer" process: input is the voltage over the heating device; output is outlet temperature

Matlab: "iddemo" (demonstration 2)



# Main Focus in This Course

---



**An iterative procedure !**

Lecture 1

Lecture Notes on System Identification and Data Analysis

49

## Reading and Exercise

---

- **Reading:** Textbook, Chapter 1; Sections 4.1-4.3
- **Further Reading (Master's Theses):**
  - L. Ljung, *From Data to Model: A Guided Tour of System Identification*, Report No. LiTH-ISY-R-1652, Linköping University, Sweden, 1994.
- **Exercise:** None

# Exams Procedure

---

- Data Selection and System Identification
- System Identification Toolbox in Matlab
  - Report preparation
- Oral examination

## Lecture 1:

### System Identification and Data Analysis

*Any question, comment or suggestion?*