# System Identification

Control Engineering EN, $3^{rd}$ year B.Sc.
Technical University of Cluj-Napoca
Romania

Lecturer: Lucian Buşoniu

# Part I

# Introduction to System Identification

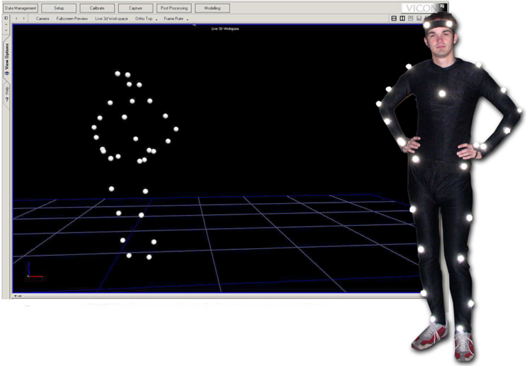
# Overall objective

System identification is the process of creating a *model* to describe the behavior of a *dynamical system*, from *experimental data*.

# An informal example

Motion capture is one example where:

- The *system* is the human
- The *data* comprises the measured trajectories of the markers
- The *model* consists of representations of these trajectories (e.g. splines)

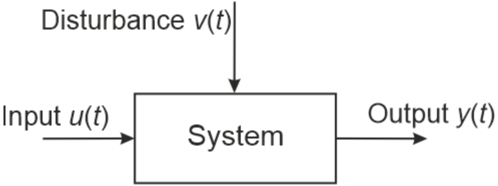


# Table of contents

# System concept

## An informal definition

A system is a part of the world with a well-defined interface, which is acted on by *input* and *disturbance* signals, and produces in response *output* signals.



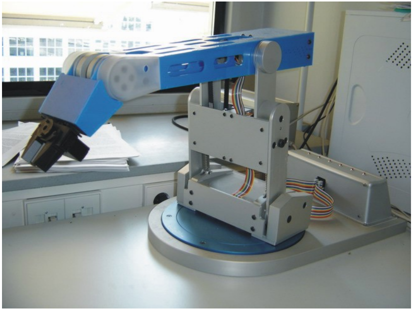

The input can be controlled, but not the disturbance; often the disturbance cannot be measured, either. Note that the signals are functions of time, so the system evolves in time (it is *dynamical*).

# System example: car

Consider the longitudinal (forward) motion of the car.

**Inputs:** Acceleration and brake pedal positions, possibly gear.
**Output:** Velocity.
**Disturbance:** Friction with varying road surfaces.

# System example: robot arm



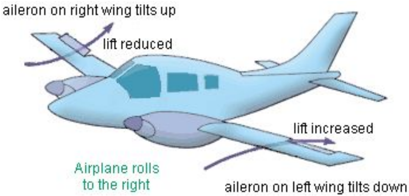
Consider a robot arm that, for example, must perform pick-and-place tasks.

**Inputs:** DC voltages on the link and end-effector motors.

**Outputs:** Positions of the links and of the end-effector.

**Disturbances:** Mass of the picked-up object (load), friction.

# System example: aircraft




Consider the roll motion of the aircraft.

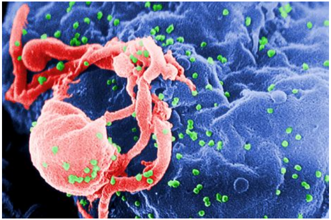
**Input:** Aileron deflection angle.

**Output:** Aircraft roll angle.

**Disturbances:** Wind, inputs from other control surfaces, etc.

Note that only a part of the overall system dynamics is studied. This type of simplification is commonly used.

# System example: HIV infection



Inputs: Quantities of applied drugs (e.g. protease inhibitors, reverse transcriptase inhibitors).

Output: Counts of infected and healthy target cells; virus count; immune effector count (per milliliter).

Disturbances: Other infections, patient characteristics.

# Other domains

The HIV example illustrates that the utility of system modeling and identification goes beyond typical cases studied in control (electrical, mechanical, hydraulic, pneumatic, including the systems exemplified above).

Other application fields include:

- Chemical industry.
- Energy, transport, and water infrastructure.
- Signal processing.
- Economy.
- Social sciences (e.g. dynamics of social networks).
- Etc.

# Table of contents

# Model concept

**An informal definition**

A model is a description of the system that captures its essential behavior.

Crucial feature: the model is always an *approximation* (idealization, abstractization) of the real system.

This feature is necessary and also desirable: exact models are unfeasible, simpler models are easier to understand and use.

# Non-math example: Mental/verbal model of a car

The model consists of verbal rules such as:

- Turning the wheel causes the car to turn.
- Pressing the gas pedal makes the car accelerate.
- Pressing the brake pedal makes the car slow down.
- ...

# Taxonomy of mathematical models

## By number of parameters:

1. Parametric models: have a fixed form (mathematical formula), with a known, often small number of parameters
2. Nonparametric models: cannot be described by a fixed, small number of parameters
   Often represented as graphs or tables

## By amount of prior knowledge ("color"):

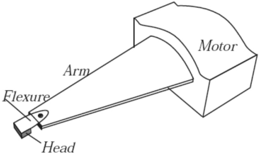
1. First-principles, white-box models: fully known in advance
2. Black-box models: entirely unknown
3. Gray-box models: partially known

System identification is used to find (the unknown parts) of black-box and gray-box models.

Examples and details follow.

# Example for (non)parameteric: hard drive

Consider a hard drive read-write head, with input = motor voltage, and output = head position.



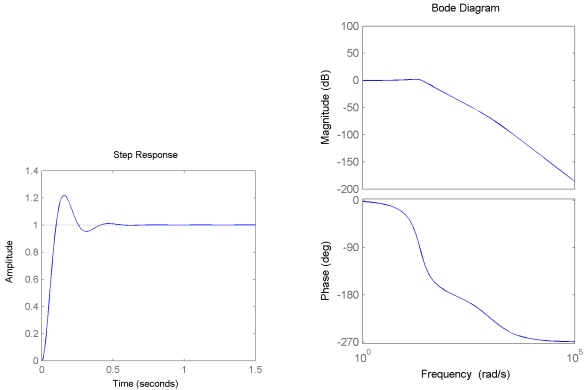
# Parametric model example: Transfer function

$$H(s) = \frac{b_0}{a_3 s^3 + a_2 s^2 + \ldots + a_1 s + a_0} = \frac{500}{0.001 s^3 + 1.02 s^2 + 20 s + 500}$$

The form (mathematical formula) is given, and it depends on a fixed number of parameters (polynomial coefficients $b_0, a_3, \ldots, a_0$) which must be set to obtain the fully specified model.

Connection: System Theory.

# Nonparametric model example: Graphs




The model represents the system behavior in graph form, such as *step response* or *frequency response* (Bode diagram).

Connection: System Theory (recall step & impulse responses of 1st and 2nd order systems, Bode diagrams).

# First-principles, white-box models

Physical laws are used to write down equations describing the system (e.g., force or mass balance equations). All the parameters in the equations are known. Models are usually continuous-time differential equations involving the inputs and outputs.

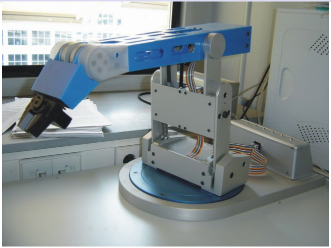
Since everything is known / we can “see inside the box”, first-principles models are also called white-box.

Characteristics:

- Remain valid for every operating point.
- Offer significant insight into the system’s behavior.
- Unfeasible if the system is too complex or poorly understood.

Connection: Process Modeling.

# White-box example



$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau$$

**Inputs:** Motor torques in the joints, collected in the vector $\tau \in \mathbb{R}^n$. $n$ denotes the number of joints.

**Output:** Angles of the links, collected in the vector $\theta \in [-\pi, \pi)^n$.

Note $\tau$ and $\theta$ are functions of time; for conciseness, argument $t$ is not given. Also, the dot denotes differentiation w.r.t. time, e.g. $\dot{\theta} = d\theta/dt$.

$M$: mass matrix, $C$: matrix of centrifugal and Coriolis forces, $G$: gravity vector known (we do not give their expressions here).

# Black-box models

Obtained numerically from experimental data collected on the system, without any prior information about the system.

Characteristics (compared to white-box models):

- Usually valid *locally*, around an operating point.
- Give less physical insight.
- Easy to construct and use, the only option in many applications.

Main focus of this system identification course.

A detailed example will be given in the next section of the lecture.

# Gray-box models

Gray-box models – as their color suggests – are a middle-ground between black-box and white-box models: the form of the model can be obtained from first principles, but some parameters are unknown and must be identified from experiments.

Example: equation $M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau$ for the robot arm available, but the friction coefficients in the joints are unknown (a typical case in practice, as friction models are very tough to obtain).

# Using the model

Models are useful for many purposes, some of the more important ones being:

- Analyze the model (e.g. to find features such as stability, time constants etc.)
- Simulate of the system's response in new scenarios (e.g. how the car patient will respond to drugs). Allows studying scenarios that might be dangerous or expensive in the real system.
- Predict the system's future output (e.g. weather prediction).
- Design a controller for the system, in order to achieve good behavior (e.g. fast response, small overshoot).
- Design the system itself, by obtaining insight into how the yet-unbuilt system will behave. (Since system is not available, requires first-principles modeling.)

Control design is the most relevant for us, as control engineers.

Connection: Control Engineering discipline (this year)

# Summary of connections with other disciplines

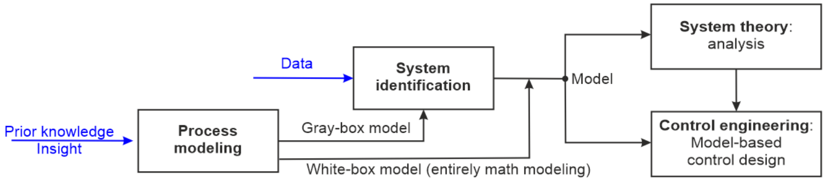
System identification uses knowledge from:

- Linear algebra
- Numerical calculus
- Process modeling
- System theory
- Optimization

and is useful for:

- Control engineering
- Continuous plant control
- Robot control systems
- etc.

# Key connections and model color

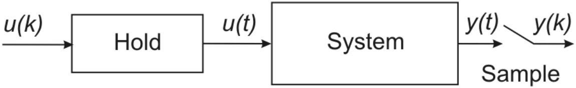


# Table of contents

# Workflow example

System identification often works in discrete time, as we will work in this example.

Usual scheme for identification in discrete-time:

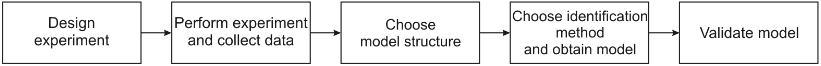



Here, we consider a flexible robot arm, $u$ = torque, $y$ = arm acceleration. The data is obtained from the Daisy database (`http://homes.esat.kuleuven.be/~smc/daisy/`).

# Workflow 0: Establish model purpose

Goal: Simulate the response of the flexible robot arm.

# Workflow 1: Experiment design

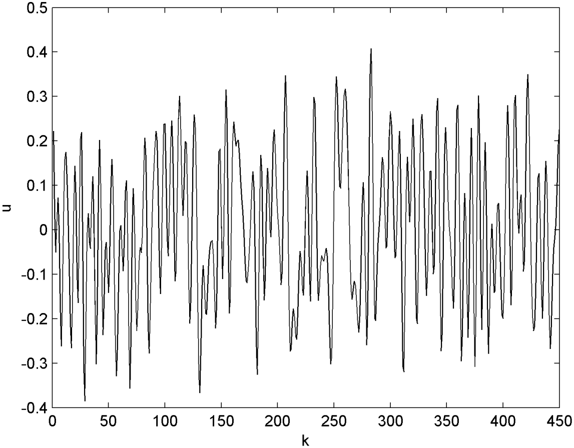



A main part of experiment design consists of selecting the input signal (duration, sampling interval, shape). This signal should be sufficiently rich to bring out the interesting behavior in the system.

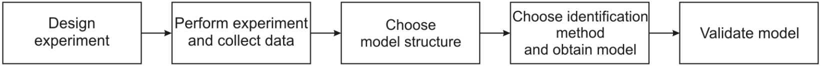
There are usually constraints: the system cannot be placed in dangerous conditions, cannot deviate too much from a profitable operating point, etc.

# Workflow 1: Experiment design: Example

Input signal: $u(k)$, $k = 0, 1, 2, \ldots, N$

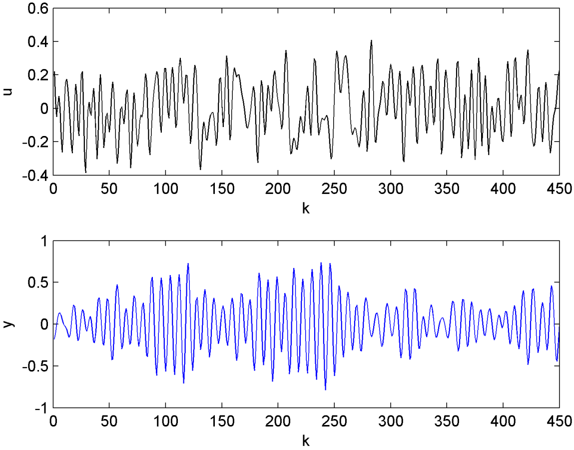


# Workflow 2: Experiment




The experiment is performed and the output data is recorded.

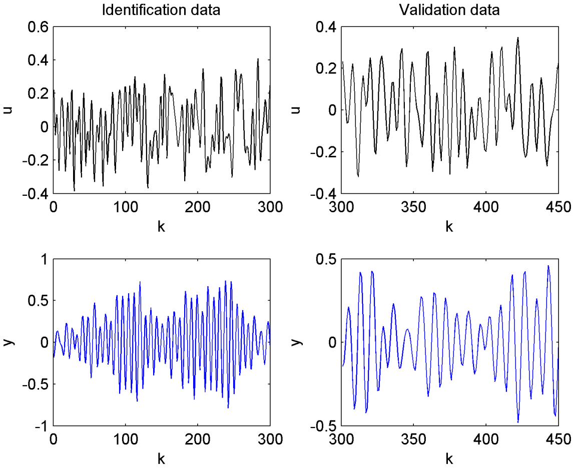
# Workflow 2: Experiment: Example



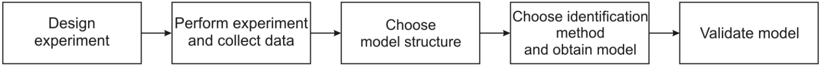

$y(k)$, $k = 0, 1, 2, \ldots, N$

# Workflow 2: Experiment: Example (continued)

We split the data into an *identification* set and a *validation* set (important later).



# Workflow 3: Structure choice




The structure of the model is chosen: graphical model, or mathematical model.

Any knowledge and intuition about the system should be exploited to choose an appropriate structure: it should be flexible enough to lead to an accurate model, but simple enough to keep the estimation task well-conditioned.

# Workflow 3: Structure choice: Example

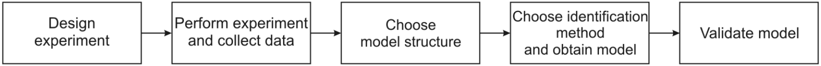
We choose a so-called 'ARX' model structure, where the output $y(k)$ at the current discrete time step is computed based on to the previous inputs and outputs:

$$\begin{aligned} y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) \\ = b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3) + e(k) \\ \text{equivalent to} \\ y(k) = -a_1 y(k-1) - a_2 y(k-2) - a_3 y(k-3) \\ + b_1 u(k-1) + b_2 u(k-2) + b_3 u(k-3) + e(k) \end{aligned}$$

$e(k)$ is the error made by the model at step $k$. The order of the model is 3.

Model parameters: $a_1, a_2, a_3$ and $b_1, b_2, b_3$.
(Recall $y$ and $u$ are *data*.)

# Workflow 4: Model estimation




A method is chosen and applied to identify the parameters of the structure. Of course, which methods are appropriate depends on the structure chosen.

# Workflow 4: Model estimation: Example

Identification consists of finding the parameters $a_1, a_2, a_3, b_1, b_2, b_3$. We choose a method that minimizes the sum of the squared errors $\sum_{k=1}^{300} e^2(k)$ on the identification data. The actual algorithm will be presented later in the course.

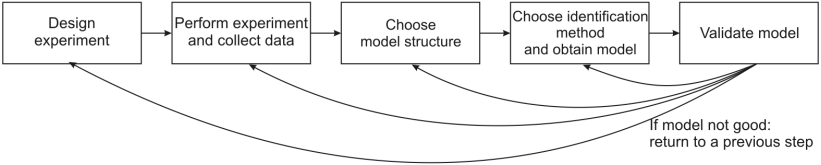
The solution is:

$$a_1 = -2.24, a_2 = 2.17, a_3 = -0.83,$$
$$b_1 = -0.06, b_2 = 0.02, b_3 = -0.05$$

which replaced in the structure gives the approximate model:

$$\begin{aligned} y(k) &= 2.24y(k-1) - 2.17y(k-2) + 0.83y(k-3) \\ &\quad - 0.06u(k-1) + 0.02u(k-2) - 0.05u(k-3) \end{aligned}$$

# Workflow 5: Model validation

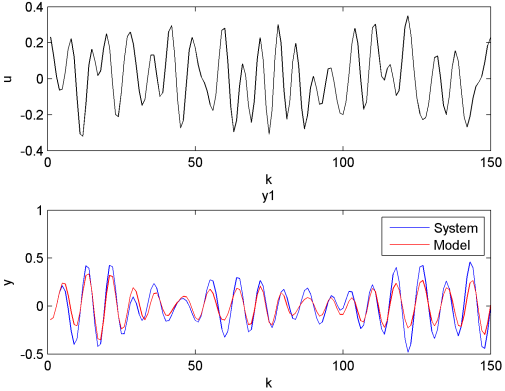



Validation is a crucial step: the model must be good enough *for our purposes*. If validation is unsuccessful, some or all of the previous steps must be repeated.

E.g., the response (outputs) of the obtained model can be compared with the true response of the system, on validation data. This validation dataset should preferably be different from the set used for identification. (Either a different experiment is performed, or the experimental data is split into testing and validation subsets.)

# Workflow 4: Model validation: Example

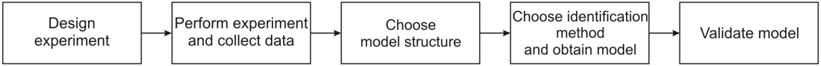
We use the validation data that we kept separate from the start:




Our goal of simulating the system output is reasonably achieved (for inputs that are “well represented” by the experimental input chosen).

# Summary (of the technical part)

- Overall objective: model of a dynamical system from data.
- System: part of the world acted upon by inputs and disturbances, producing outputs. Examples.
- Model: (usually mathematical) description of a system capturing its essential behavior.
- Taxonomy of models: parametric/nonparametric, first-principles (white-box)/black-box/gray-box.
- Examples and usage of models.
- System identification workflow (see below), with method details to be worked out later.



# Table of contents

# Prerequisites and literature

## Prerequisites:

Dynamical systems and models, linear algebra, numerical methods, statistics, Matlab
(brief introductions to the required topics will be given along the way)

## Literature

- Mandatory: lecture slides (written down in detail to give a self-contained, complete picture).
- Students may also optionally consult the book: T. Söderström and P. Stoica. *System Identification*. Prentice Hall, 1989, on which this course is based. Full text available free of charge at: `http://user.it.uu.se/~ts/bookinfo.html`.

Credit for some ideas goes to the SysID course at Uppsala University, by K. Pelckmans.
`http://www.it.uu.se/edu/course/homepage/systemid/vt12`

# Grading and platforms

## Grading

- 30% labs: 2x15% for 2 lab tests.
- 10% quizzes at the start of each lab, graded starting with lab 2.
- 30% project: 2x15% for 2 parts.
- 30% final written exam.
- 10% quizzes during each lecture.

## Platforms

- Microsoft Teams, for questions and updates (and classes only if needed).
- Matlab, for developing labs solutions.
- ClassMarker, for lab and lecture quizzes.

# Labs

- Each lab (except lab 1) starts with a quiz.
- Validated solutions are required to mark the lab as done, attendance is not sufficient. Two-step validation: correctness and code originality check by TA; then semi-automated plagiarism check.
- First copying attempt invalidates that lab, second attempt forfeits the whole course and you take it next year.
- Two 1-hour long lab tests, one mid-semester and one at the end. Must apply (randomly chosen) methods studied at previous labs.

# Project

Part 1: Polynomial function approximation
Part 2: TBA

- Deliverables: Matlab code and presentation for each part. Presentation defended in interactive session including thorough question-answer part.
- Originality checked both by direct questions and semi-automatically, copying forfeits course.

# Website, contact

http://busoniu.net/teaching/sysid2023

Email lecturer: lucian@busoniu.net

On the website you can find:

- Detailed schedule
- Lecture slides
- Lab rules and material
- Project information
- etc.