Table of Contents I

notes

- this is the table of contents of this document; each section corresponds to a specific part of the course

Introduction to System Identification

notes

Contents map

developed content units	taxonomy levels
system identification	u1, e1
parameter estimation	u1, e1
least squares	u1, e1

prerequisite content units	taxonomy levels
impulse response	u1, e1

notes

Main ILO of sub-module "Introduction to System Identification"

describe the underlying concepts behind system identification

describe the role of system identification in model-based control

notes

- By the end of this module, you should be able to explain why we need good models for control, and describe how to fit a linear model using least squares.

What does "identifying a system" mean?

in a nutshell: to build mathematical models of
a dynamic system from its input-output data

Typical ingredients:

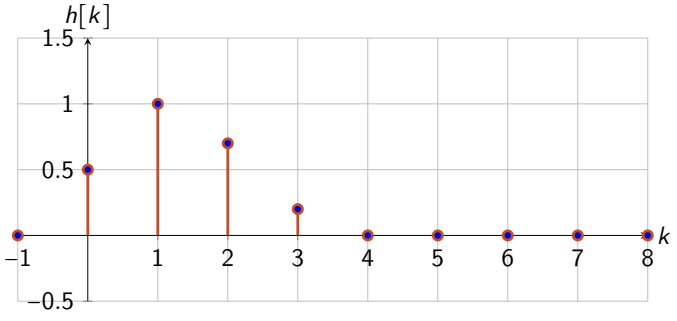
- measurements (inputs and outputs)
- model structure (e.g., impulse response).

notes

- Model = structure + parameters. Structure: e.g. FIR or ARX; Parameters: the coefficients we estimate.
- We'll estimate parameters that shape an impulse response from measurements.

Identification in this course

focus on estimating the coefficients
that define the impulse response
of causal FIR discrete-time LTI systems
(likely less than 1% of what sysid encompasses)



notes

- here we will focus on something that is extremely narrow - the goal is eventually to be exposed to the philosophy behind sysid

More precisely

available dataset: $\mathcal{D} = \{y[k], u[k]\}_{k \in \mathcal{K}}$

assumed model: $y[k] = \sum_{i=0}^{n} h[i]u[k-i] + e[k]$

with

- n to be estimated
- $h[0], h[1], \ldots, h[n]$ to be estimated
- $e[k]$ noise / modeling error with potentially unknown statistics

notes

- This is the model structure well use: convolution of input with the impulse response.
- Our task is: given input-output data, find the best h_i that match the system.

Does identification matter for control?

. . . how can you do model-based control without a model? Caveats, though:

- end goal of this course = MPC
- bad model $\Longrightarrow$ bad model based controller
- bad dataset $\Longrightarrow$ bad model *(filtering may help!)*

notes

- Emphasize: we want to control real systems (robots, motors, drones) and we often cant derive perfect models from first principles.
- Model-based control assumes the model is good enough to reflect how the system reacts.

Next modules

- least squares
- regularization

notes

- We'll derive and code these two technologies in Python soon

Summarizing

describe the underlying concepts behind system identification

describe the role of system identification in model-based control

- need a model structure
- need input-output data
- need an estimation algorithm

notes

- You should now understand how and why we estimate parameters from data.
- Next step: code something in Python and apply to real and simulated systems

Most important python code for this sub-module

notes

A very widely used library

`https://scikit-learn.org/stable/`

notes

- this is quite a library! It has a lot of stuff in it

Self-assessment material

notes

Question 1

Why do we need system identification before applying model-based control techniques like MPC?

Potential answers:

I: **(wrong)** To make the system faster
II: **(wrong)** To reduce noise in sensors
III: **(correct)** To estimate a model of the system from data
IV: **(wrong)** I do not know

Solution 1:

MPC and other model-based controllers require a mathematical model of the system. If we don't have it from physics, we estimate it from data.

notes

- see the associated solution(s), if compiled with that ones :)

Question 2

What is the primary purpose of system identification in control engineering?

Potential answers:

I: **(wrong)** To increase the processing speed of the control system
II: **(correct)** To construct a mathematical model of a dynamic system using data
III: **(wrong)** To eliminate measurement noise entirely from sensors
IV: **(wrong)** To design the controller directly without requiring a model
V: **(wrong)** I do not know

Solution 1:

System identification aims to build mathematical models from input-output data, which are essential for model-based control techniques. Noise reduction and controller design are separate processes.

notes

- see the associated solution(s), if compiled with that ones :)

Question 3

Why is input-output data critical in system identification?

Potential answers:

I: **(wrong)** To determine the physical dimensions of system components
II: **(correct)** To estimate model parameters that best explain the observed behavior
III: **(wrong)** To validate the controller's performance in real-time
IV: **(wrong)** To replace the need for mathematical modeling entirely
V: **(wrong)** I do not know

Solution 1:

Input-output data enables parameter estimation by linking the model's predictions to actual measurements. Physical dimensions and controller validation are separate concerns.

notes

- see the associated solution(s), if compiled with that ones :)

Question 4

What is a major risk of using a poorly identified model in Model Predictive Control (MPC)?

Potential answers:

I: **(wrong)** The system's hardware may suffer physical damage
II: **(correct)** The controller may perform inadequately due to inaccurate predictions
III: **(wrong)** Increased computational load during controller operation
IV: **(wrong)** The need for more frequent sensor calibrations
V: **(wrong)** I do not know

Solution 1:

MPC relies on model predictions for control decisions. A poor model leads to erroneous predictions and degraded performance. Physical damage or sensor issues are not direct consequences.

notes

- see the associated solution(s), if compiled with that ones :)

Question 5

Which three elements are fundamentally required for system identification?

Potential answers:

I: **(wrong)** Actuators, sensors, and a power supply
II: **(wrong)** Controller design, simulation software, and data storage
III: **(correct)** Input-output data, estimation algorithm, and if available, model structure
IV: **(wrong)** Noise filters, feedback loops, and setpoints
V: **(wrong)** I do not know

Solution 1:

System identification requires a model structure to define the mathematical form, input-output data to fit parameters, and an estimation algorithm (e.g., least squares) to compute optimal values. Hardware components and controller design are separate.

notes

- see the associated solution(s), if compiled with that ones :)

Recap of sub-module "Introduction to System Identification"

- Model-based control requires accurate models
- System identification builds models from data
- There are several tools to estimate model parameters, in this course we only scratch the surface

notes

- The most important takeaway is the link between model quality and control performance.