UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Advanced methods for system identification Ph.D. course

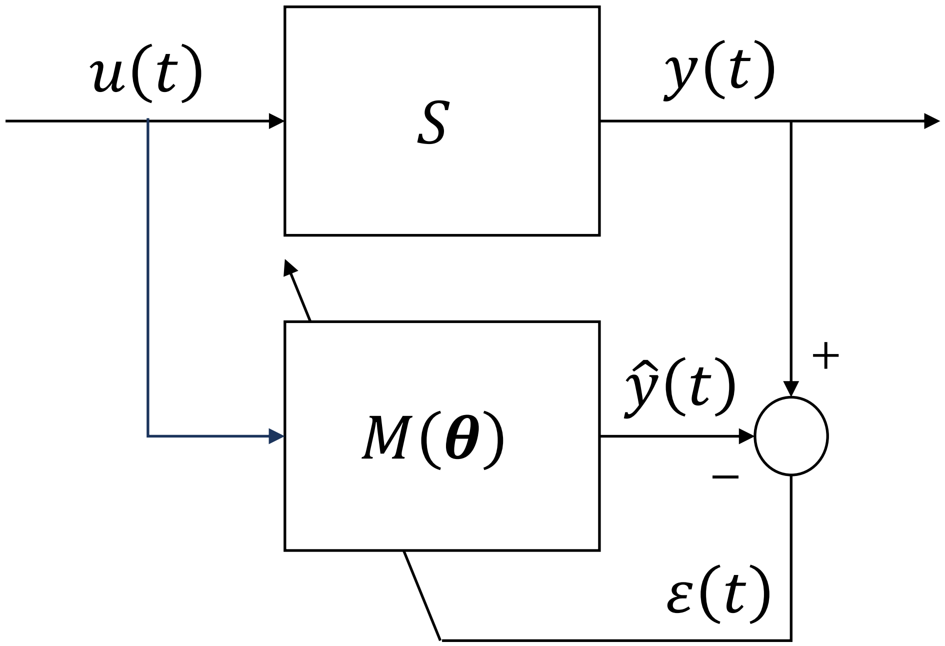
Lesson 1: Introduction

Ph.D. IN ENGINEERING AND APPLIED SCIENCES

TEACHER
Mirko Mazzoleni

PLACE
University of Bergamo

What we have learnt so far…



Identification methods for **linear SISO stationary** systems

Stochastic input-output linear models:

- **Time series models** (AR, ARMA)

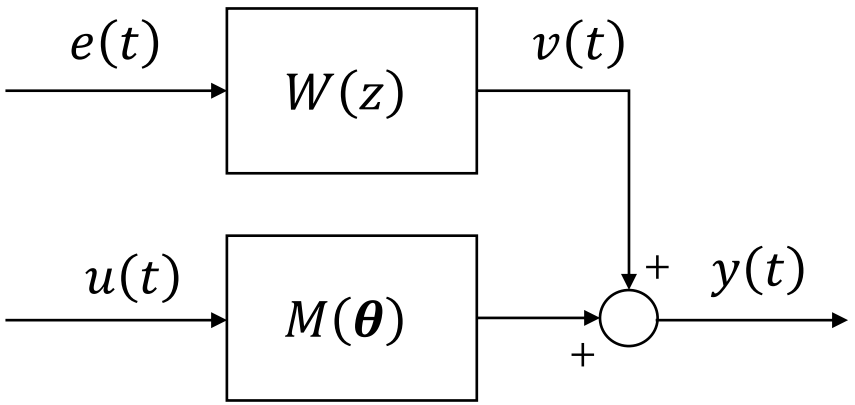
$$y(t) = a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) + e(t) + c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c)$$

- **Input\output models** (ARX, ARMAX)

$$y(t) = a_1 y(t-1) + \cdots + a_{n_a} y(t-n_a) + b_1 u(t-1) + \cdots + b_{n_b} u(t-n_b) +$$

$$+ e(t) + c_1 e(t-1) + \cdots + c_{n_c} e(t-n_c)$$

What we have learnt so far...



General model form (in operational notation)

- $e(t)$ is **a white noise** signal, introduced to account for all **unmodeled dynamics** and the effects of **external unmeasured disturbances**
- $W(z)$ and $G(z)$ are suitable **transfer functions**

$$M(\boldsymbol{\theta}): y(t) = G(z, \boldsymbol{\theta})u(t-1) + W(z, \boldsymbol{\theta})e(t), \qquad e(t) \sim \mathrm{WN}(0, \lambda^2)$$

Model identification: typically based on prediction error minimization (PEM) approach

Kolmogorov-Wiener theory: optimal prediction of linear stationary stochastic processes

What we have learnt so far...

Prediction form model $M(\boldsymbol{\theta})\colon \hat{y}(t|t-1) = \dfrac{G(z)}{W(z)} u(t-1) + \left(1 + \dfrac{1}{W(z)}\right) y(t)$

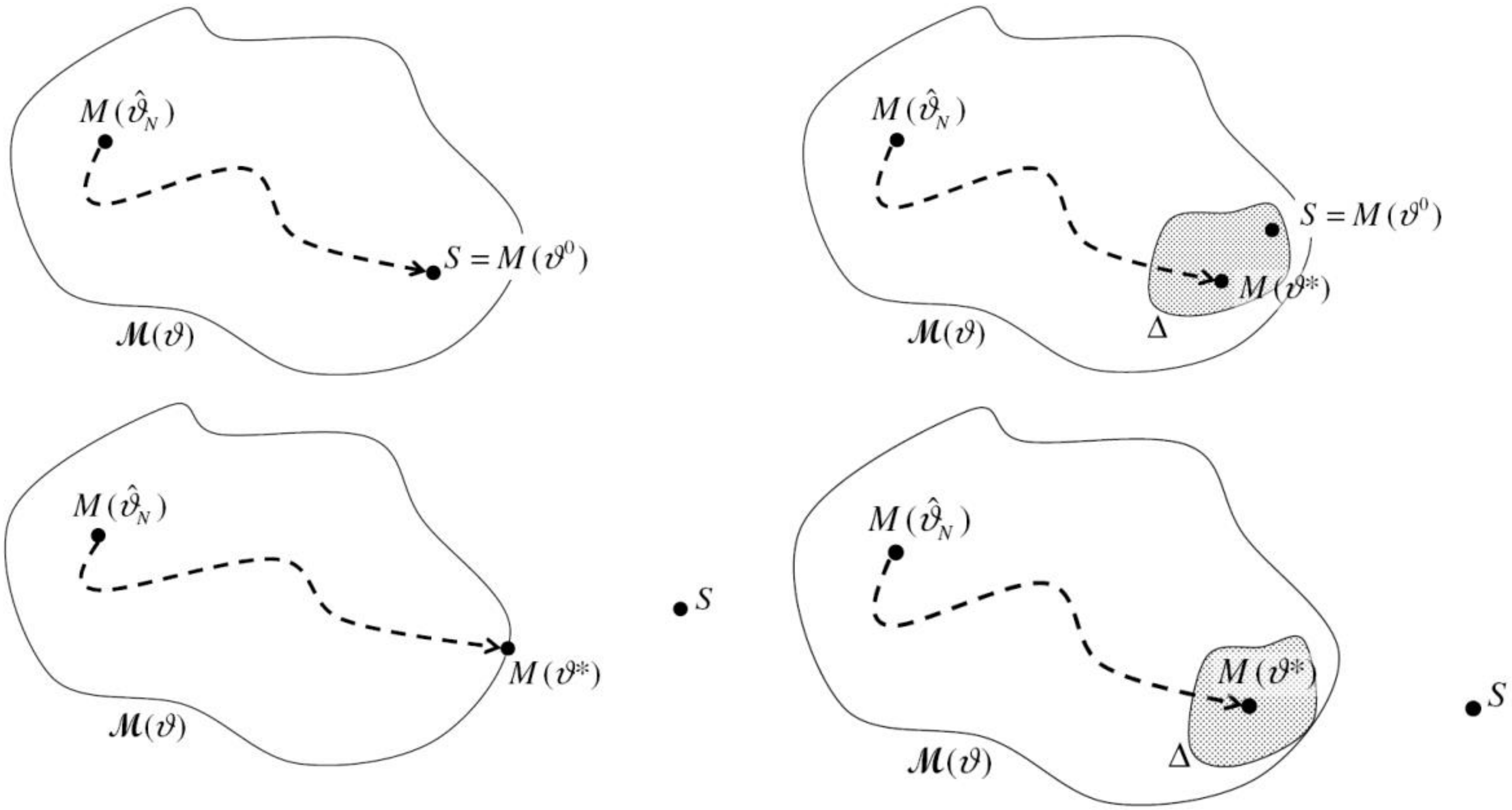
Given a model family $\{M(\boldsymbol{\theta})\ |\boldsymbol{\theta} \in \Theta\}$ the best model is selected as the one that **minimizes the mean square** (one-step ahead prediction) **error** over the available data:

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon(t,\boldsymbol{\theta})^2, \qquad \varepsilon(t,\boldsymbol{\theta}) = y(t) - \hat{y}(t|t-1;\boldsymbol{\theta})$$

In other words

$$M^{\text{opt}} = M(\boldsymbol{\theta}^{\text{opt}}), \qquad \boldsymbol{\theta}^{\text{opt}} = \min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

What we have learnt so far…



What we have learnt so far...

PEM identification algorithms

- **AR(X):** linear regression, Least Squares
- **ARMA(X):** pseudo-linear regression, Maximum Likelihood (quasi-Newton optimization)

Other issues

- **model validation**
- **structure selection** (FPE, AIC, MDL)
- **data pre-processing** (detrending, outlier detection and removal)
- **structural identifiability** (is the best model approximating the data unique in the considered model family?), under- and over-parametrization
- **experimental identifiability** (is there a unique minimizing model parametrization compatible with the data?), persistency of excitation of the input signal, input design
- **frequency weighting** through prefiltering

…and what we still have to learn! Part 1

We have focused on SISO systems, but many problems are actually **multivariable** and hardly amenable to decoupling.

Examples

- Helicopter control (4×6 model)
- Gas turbine engine (3×6 model)
- Distillation column (5×5 model)

see Skogestad and Postlethwaite, "Multivariable Feedback Control"

Define the following quantities:

- m: number of inputs
- p: number of outputs
- n: number of states

...and what we still have to learn! Part 1

Problems

- Which representation? **Input-output** or **state-space** models?
  - ✓ Suppose $m = 40, p = 28, n = 22$
    - ❑ **I/O model:** 112000 parameters (each I/O relation modeled by a 100° order FIR)

$$\text{number of params} = m \cdot p \cdot 100 = 40 \cdot 28 \cdot 100 = 112000$$

    - ❑ **State-space:** 3100 parameters ➡ **Subspace identification methods**

$$\begin{cases} \underset{n\times 1}{\boldsymbol{x}(t+1)} = \underset{n\times n}{A} \cdot \boldsymbol{x}(t) + \underset{n\times m}{B} \cdot \underset{m\times 1}{\boldsymbol{u}(t)} \\ \underset{p\times 1}{\boldsymbol{y}(t)} = \underset{p\times n}{C} \cdot \boldsymbol{x}(t) + \underset{p\times m}{D} \cdot \boldsymbol{u}(t) \end{cases}$$

$$\text{number of params} = n \cdot n + n \cdot m + n \cdot p + p \cdot m$$
$$= 22 \cdot 22 + 22 \cdot 40 + 22 \cdot 28 + 28 \cdot 40$$
$$= 3100$$

…and what we still have to learn! Part 2

In many applications data are not available all together, but they are **collected one by one on-line**. Can we still identify a model in these conditions?

Recursive identification methods

Sometimes the system generating the data is **not time-invariant** (or it is nonlinear).

Can we recompute the model on-line adapting it to the current behavior of the system?

Adaptive identification methods

…and what we still have to learn! Part 3

The standard system identification setup assumes that the **noise enters on the output** of the system. In some cases, this assumption might not hold, and also the **input is noisy.**

Learning algorithms such as linear regression using Least Square formulation lead to **biased estimates**. Is there a way to correct this problem?

Instrumental variables methods

…and what we still have to learn! Part 4

Up to this point, we covered methods based on a time-domain formulation. A characteristic feature of dynamical system is their frequency-domain interpretation (e.g. Bode plots, transfer functions, …)

It is possible to formulate an identification problem in frequency domain? What are the pros\cons of this approach?

Frequency-domain system identification

…and what we still have to learn! Part 5

Traditionally, the problem of model **complexity selection** in system identification has been tackled by resorting to:

- Complexity **criteria** (AIC, BIC, FPE)
- **Validation** techniques based on a held-out dataset

Another way to deal with the approximation-generalization tradeoff is by using **regularization**

The linear modeling paradigm is generally good as a first approximation, but the world is … **nonlinear**!

Kernel methods