UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Control Automation Lab

ADAPTIVE LEARNING, ESTIMATION AND SUPERVISION OF DYNAMICAL SYSTEMS (ALES)

Lecture 1: Introduction

Master Degree in COMPUTER ENGINEERING

Data Science and Data Engineering Curriculum

SPEAKER
Prof. Mirko Mazzoleni

PLACE
University of Bergamo

Who I am

- **Name:** Mirko Mazzoleni

- **Currently:** Assistant Professor (RTD-B), **Control and Automation Lab (CAL UniBG)**

✓ *Research:* System identification, machine learning, fault diagnosis

✓ *Teaching:*
1. Identificazione dei Modelli e Analisi dei Dati (IMAD)
2. Adaptive learning, estimation and supervision of dynamical systems (ALES)
3. Sustainable and Industrial System laboratory

- **Other:** Co-founder of AISent srl startup https://aisent.io/ AISENT

- **Contacts**

✓ mirko.mazzoleni@unibg.it

✓ https://mirkomazzoleni.github.io/

✓ http://cal.unibg.it/ **Website CAL UniBG**

✓ https://www.facebook.com/ControlAutomationLabUnibg/

Control and Automation Laboratory (CAL) @ University of Bergamo, Italy

People:

- 3 professors
- 1-3 visiting professors
- 6 PhD students

- **Teaching** (control systems)
- **Research** (advanced control, optimization, fault diagnosis, identification)
- **Industrial projects** (manufacturing, aerospace, packaging,...)

Outline

1. Presentation of the Adaptive Learning, Estimation and Supervision of dynamical systems (ALES) course

2. What we have learnt so far…

3. …and what we still have to learn!

Outline

1. **Presentation of the Adaptive Learning, Estimation and Supervision of dynamical systems (ALES) course**

2. What we have learnt so far…

3. …and what we still have to learn!

Prerequisites for the course

It is **strongly suggested to have** good foundations in the following topics:

- Systems identification (IMAD)
- Control systems
- Linear algebra
- Calculus 1 and 2

How to «refresh» the prerequisites?

- Systems identification (IMAD) ⇒ Unibg course
- Control systems ⇒ Unibg courses
- Linear algebra ⇒ Unibg course, Gilbert Strang course @MIT on Youtube
- Calculus I and II ⇒ UniBg course

Exam

- Short **written** exam **1 hour + coding project**
- **Open questions** about the theory
- **Matlab implementation** of a technique\algorithm from the literature + oral power point presentation

How to be prepared for the exam?

- Follow the lessons
- Study the theory
- Implement by yourself the algorithms presented at lesson

Educational objectives of the course

At the end of the course, you will be able to:

- **Employ** *recursive* and *adaptive estimation methods*, without and with constraints on the estimation variables

- **Identify** models of dynamical systems in *state-space form*, with *multiple-inputs and multiple-outputs* (MIMO), and in *closed-loop* settings

- **Define** and **solve** industrial *fault diagnosis problems*, identify their main components and envisage a possibile solution using different approaches

Teaching materials

Materials provided by the teacher

- Lectures slides
- **Sorry. No code** ☹*:* you have to implement the algorithms by your own MATLAB®

Interaction and feedback

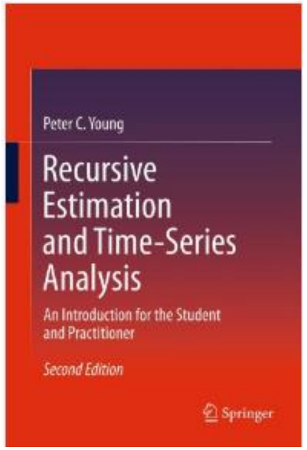
- During the week I will give you **activities** to do and **tests** to answer. They are **optional** but they help you to understand the degree of learning before the exam. In addition, they will contribute to giving a bonus of **+3 points** to the final grade

We will use **MS Teams activities**

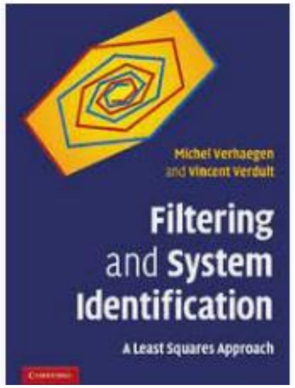
Teaching materials

Suggested textbooks

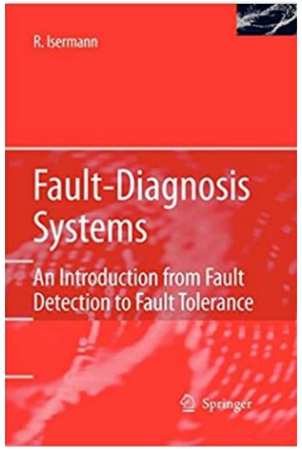
- Peter C. Young, **Recursive Estimation and Time-Series Analysis**, 2° ed. Springer (2011)



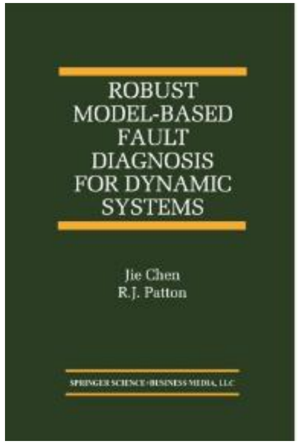
- Michel Verhaegen, **Filtering and system identification: a least squares approach**, Cambridge University Press (2007)



- Rolf Isermann, **Fault-Diagnosis Systems: An Introduction From Fault Detection To Fault Tolerance**, Springer (2005)



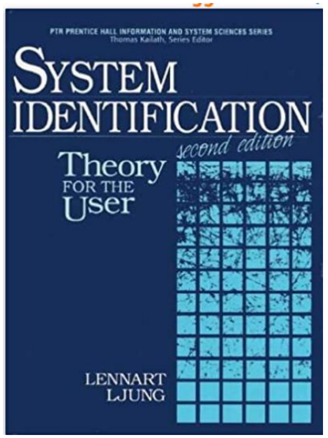
- Jie Chen, Ron J. Patton, **Robust Model-Based Fault Diagnosis for Dynamic Systems**, Springer (1999)



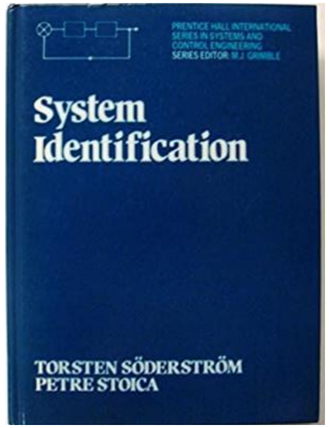
Teaching materials

Other suggested textbooks

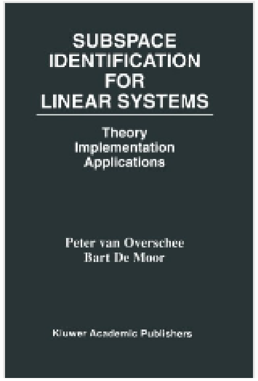
- Lennart Ljung, **System Identification: Theory for the User**, Pearson (1998)



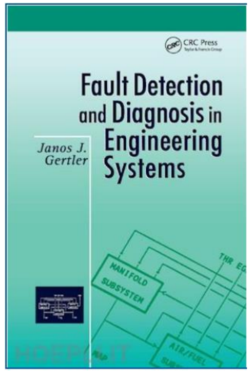
- Torsten Soderstrom, Petre Stoica, **System identification** Prentice Hall international (2001)



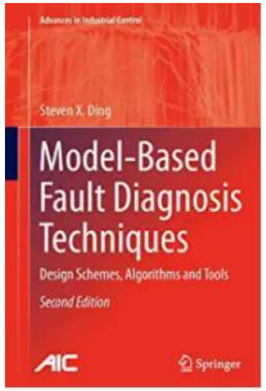
- Peter Van Overschee, Bart De Moor, **Subspace identification for linear systems**, Springer (1996)



- Janos Gertler, **Fault Detection and Diagnosis in Engineering Systems**, CRC Press, (1998)



- Steven X. Ding, **Model-baed fault diagnosis techniques**, 2° ed, Springer (2013)



Syllabus

1. **Recursive and adaptive identification**
   1.1 Recursive ARX estimation (RLS)
   1.2 Least Mean Squares (LMS)
   1.3 Instrumental Variables (IV)

2. **Subspace and MIMO identification**
   2.1 Singular Value Decomposition
   2.2 Impulse data: Ho-Kalman, Kung algorithms
   2.3 Generic I/O data: the MOESP algorithm

3. **Closed-loop identification**

4. **Supervision of dynamical systems**
   4.1 Introduction to fault diagnosis
   4.2 Model-based fault diagnosis
   4.3 Parity space approaches
   4.4 Observer-based approaches
   4.5 Signal-based fault diagnosis
   4.6 Knowledge-based fault diagnosis

CASE STUDIES

- Virtual Reference Feedback Tuning
- Nuclear particles classification
- Leak detection in an industrial valve
- Bearing fault identification

Outline

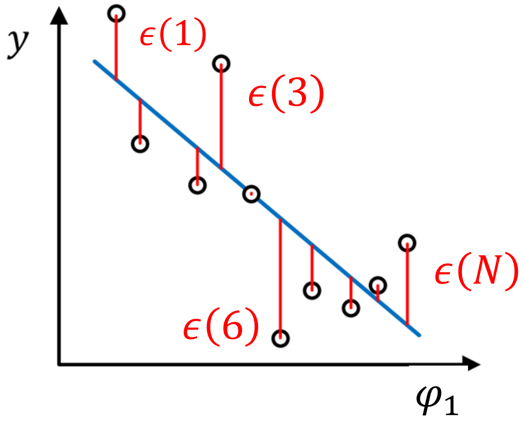
1. Presentation of the Adaptive Learning, Estimation and Supervision of dynamical systems (ALES) course

2. **What we have learnt so far…**

3. …and what we still have to learn!

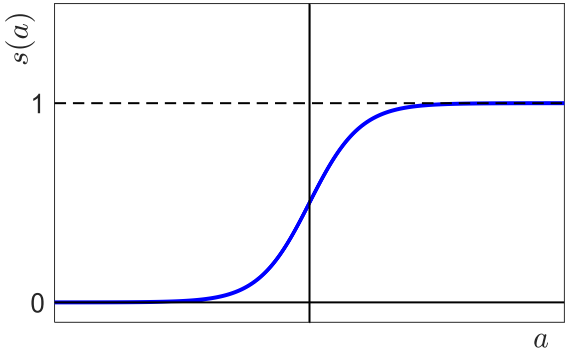
Modeling static systems and machine learning

Linear regression model



$$y(i) = \boldsymbol{\varphi}^\top(i)\boldsymbol{\theta} + \epsilon(i)$$

Logistic regression model



$$P(y = 1|\boldsymbol{\varphi}) = \frac{1}{1 + e^{-\boldsymbol{\varphi}^\top\boldsymbol{\theta}}}$$

Linear regression algorithm

$$J(\boldsymbol{\theta}) = \frac{1}{N}(X\boldsymbol{\theta} - Y)^\top(X\boldsymbol{\theta} - Y)$$

$$\hat{\boldsymbol{\theta}} = (X^\top X)^{-1}X^\top Y$$

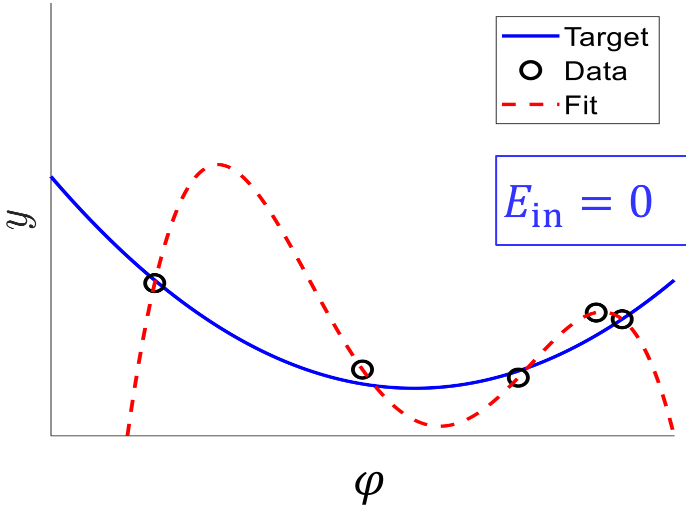
Logistic regression algorithm

$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\left(y(i)\ln\pi(i) + (1 - y(i)\ln[1 - \pi(i)]\right)$$

Gradient descent $\hat{\boldsymbol{\theta}}^{(k+1)} = \hat{\boldsymbol{\theta}}^{(k)} - \alpha \cdot \nabla J(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}^{(k)}}$

Modeling static systems and machine learning

Overfitting

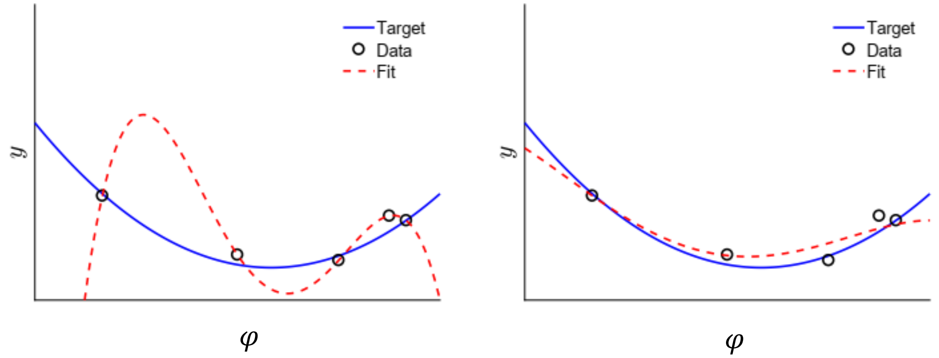


$E_{\text{in}} = 0 \qquad E_{\text{out}} = \textbf{huge}$

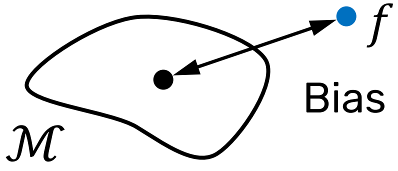
Regularization

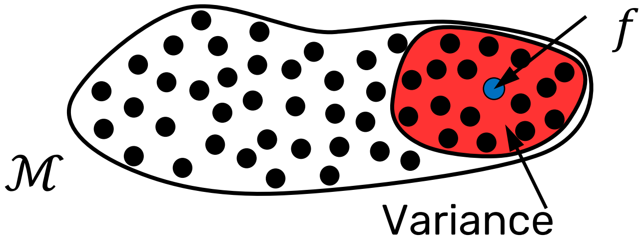
$$E_{\text{aug}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\left(y(i) - h(\boldsymbol{\varphi}(i);\,\boldsymbol{\theta})\right)^2 + \lambda_{\text{reg}} \cdot \sum_{j=0}^{d-1}\left(\theta_j\right)^2$$

λ_{reg_1} $\qquad$ $\lambda_{\text{reg}_2} > \lambda_{\text{reg}_1}$



Bias and variance



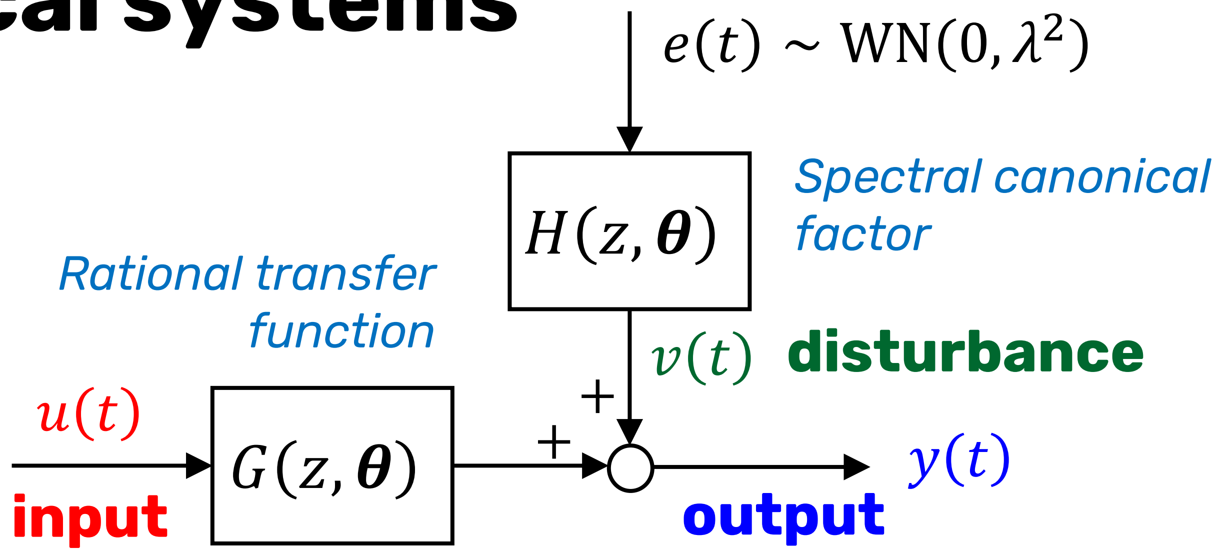


Validation and cross-validation

Train (60%)	Validation (20%)	Test (20%)

Modeling LTI SISO dynamical systems

In the IMAD course, we studied **Prediction Error Methods** (PEM) for **SISO LTI** dynamical systems



The **general form** of the models was:

- $e(t)$ is **a white noise** signal, introduced to account for all **unmodeled dynamics** and the effects of **external unmeasured disturbances**

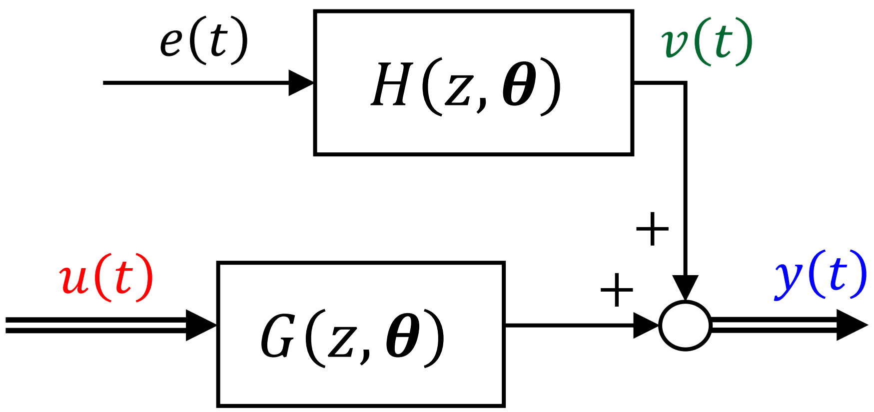
- $G(z; \boldsymbol{\theta})$ and $H(z; \boldsymbol{\theta})$ are suitable **transfer functions** that depend on parameters $\boldsymbol{\theta}$

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

Modeling LTI SISO dynamical systems

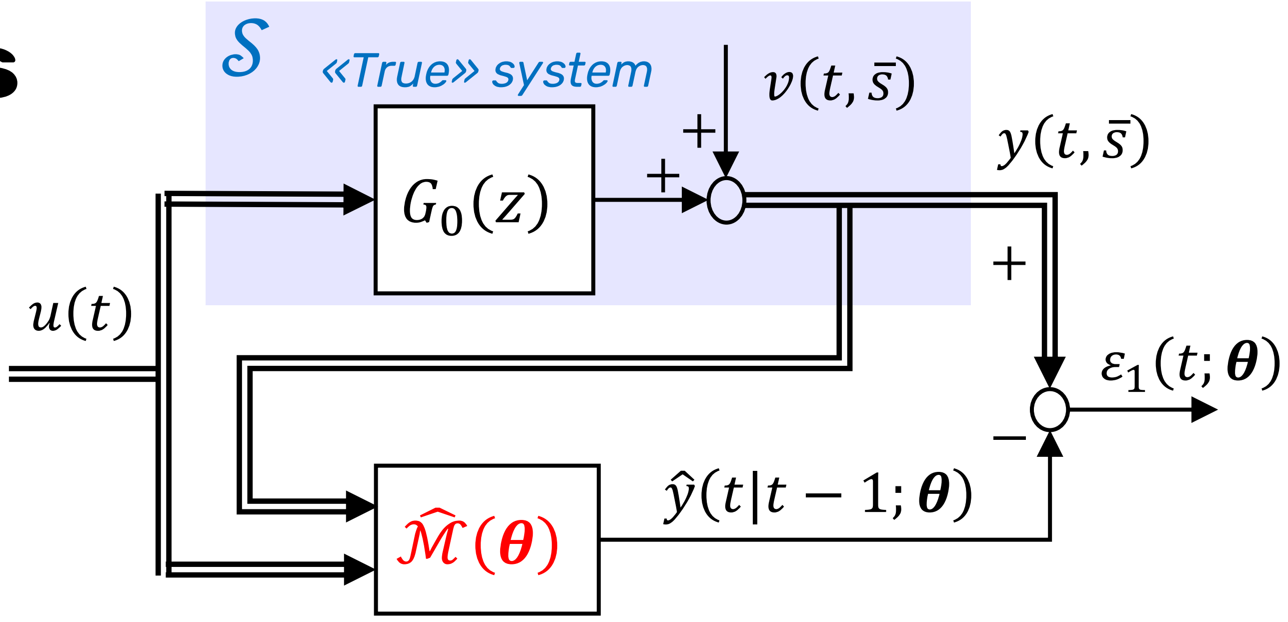
Family of models	$G(z,\boldsymbol{\theta})$	$H(z,\boldsymbol{\theta})$
ARX	$\frac{B(z,\boldsymbol{\theta})}{A(z,\boldsymbol{\theta})} z^{-k}$	$\frac{1}{A(z,\boldsymbol{\theta})}$
ARMAX	$\frac{B(z,\boldsymbol{\theta})}{A(z,\boldsymbol{\theta})} z^{-k}$	$\frac{C(z,\boldsymbol{\theta})}{A(z,\boldsymbol{\theta})}$
OE	$\frac{B(z,\boldsymbol{\theta})}{F(z,\boldsymbol{\theta})} z^{-k}$	1
FIR	$B(z,\boldsymbol{\theta}) z^{-k}$	1
BJ	$\frac{B(z,\boldsymbol{\theta})}{F(z,\boldsymbol{\theta})} z^{-k}$	$\frac{C(z,\boldsymbol{\theta})}{D(z,\boldsymbol{\theta})}$

The $\boldsymbol{\theta}$ vector represents the **model parameters** (value of the coefficients of the $A(z), B(z), C(z)$ polynomials)



Prediction Error Methods

Given a model family $\mathcal{M}(\boldsymbol{\theta})$, the estimate of $\boldsymbol{\theta}$ was performed by minimizing variance of the **one-step prediction error** $\varepsilon_1(t;\boldsymbol{\theta})$



Model in «prediction form»

Model in prediction form

$$\widehat{\mathcal{M}}(\boldsymbol{\theta}):\ \hat{y}(t|t-1;\boldsymbol{\theta}) = H^{-1}(z,\boldsymbol{\theta})G(z,\boldsymbol{\theta})\cdot u(t) + [1 - H^{-1}(z,\boldsymbol{\theta})]\cdot y(t)$$

PEM estimate

$$\varepsilon_1(t;\boldsymbol{\theta}) = y(t) - \hat{y}(t|t-1;\boldsymbol{\theta})$$

$$J_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{t=1}^{N}\varepsilon_1(t;\boldsymbol{\theta})^2 \qquad \widehat{\boldsymbol{\theta}}_N = \arg\min_{\boldsymbol{\theta}\in\Theta} J_N(\boldsymbol{\theta})$$

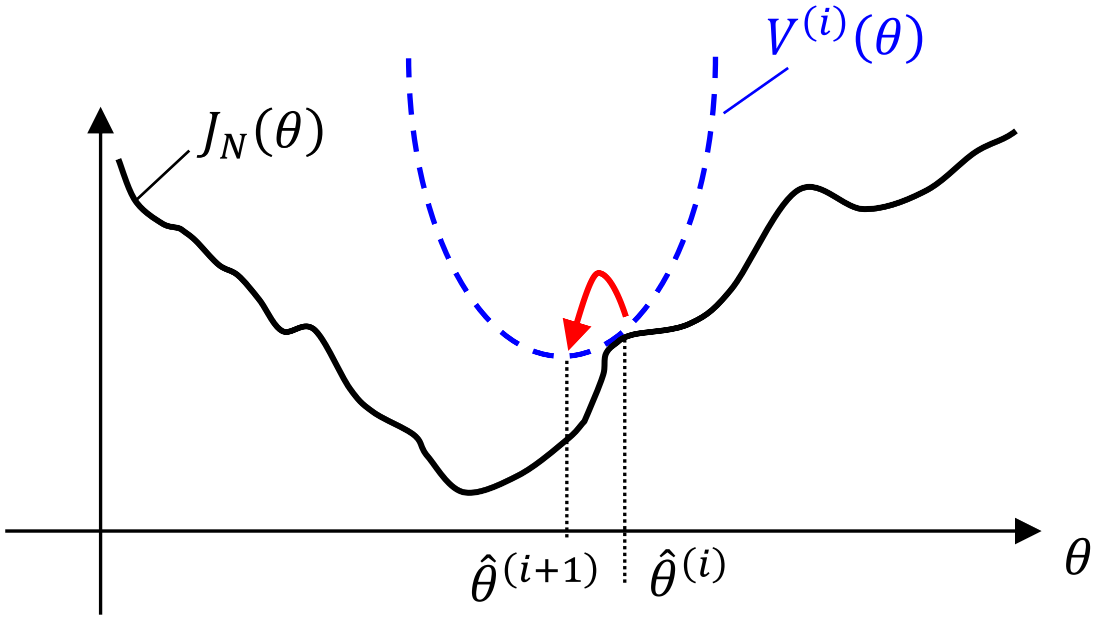
Prediction Error Methods: estimation algorithms

- **ARX, FIR: convex** cost function (global minimum) ⇨ **Least squares**
- **ARMAX, BJ, OE: not convex** cost function (local minima) ⇨ **Quasi-Newton iterative schemes**

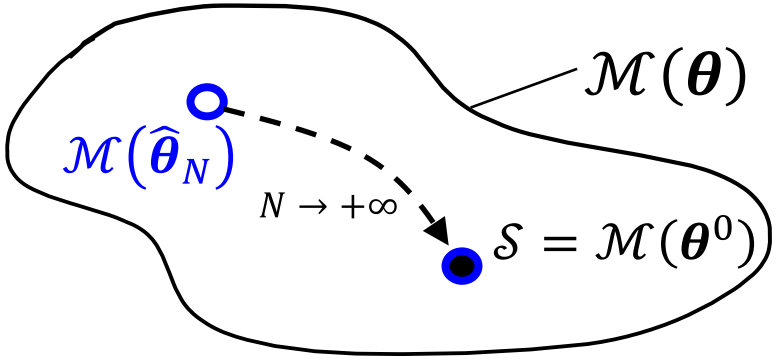
ARX, FIR

$$\hat{\boldsymbol{\theta}}_N = \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^\top(t)\right]^{-1} \cdot \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)y(t)\right]$$

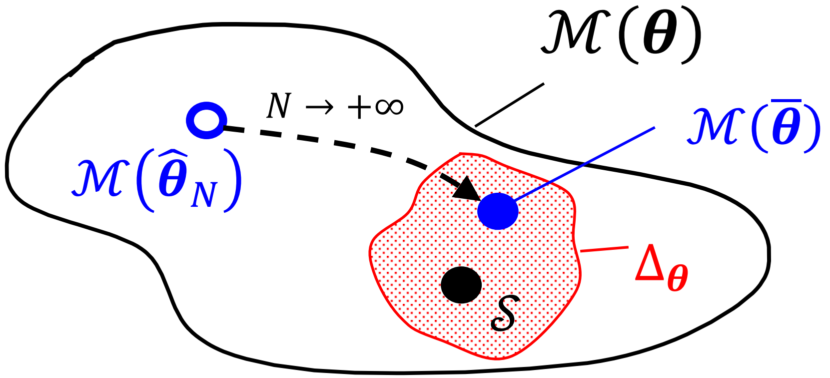
ARMAX, BJ, OE



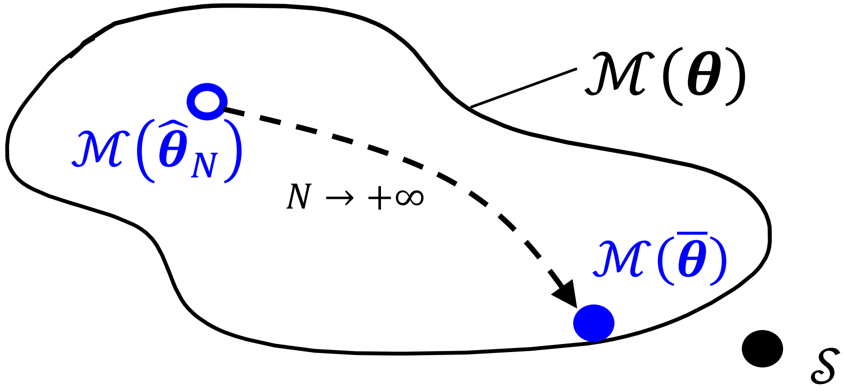
Prediction Error Methods: asymptotic properties



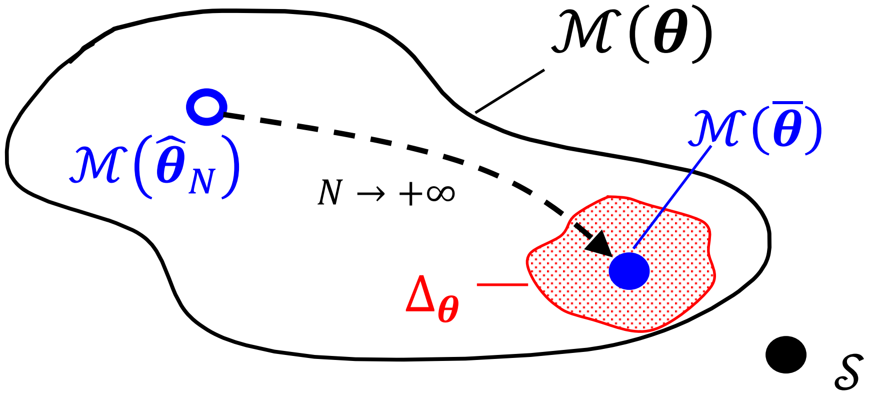
$$\hat{\boldsymbol{\theta}}_N \xrightarrow[N \to +\infty]{} \bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^0$$



$$\hat{\boldsymbol{\theta}}_N \xrightarrow[N \to +\infty]{} \bar{\boldsymbol{\theta}} \neq \boldsymbol{\theta}^0$$



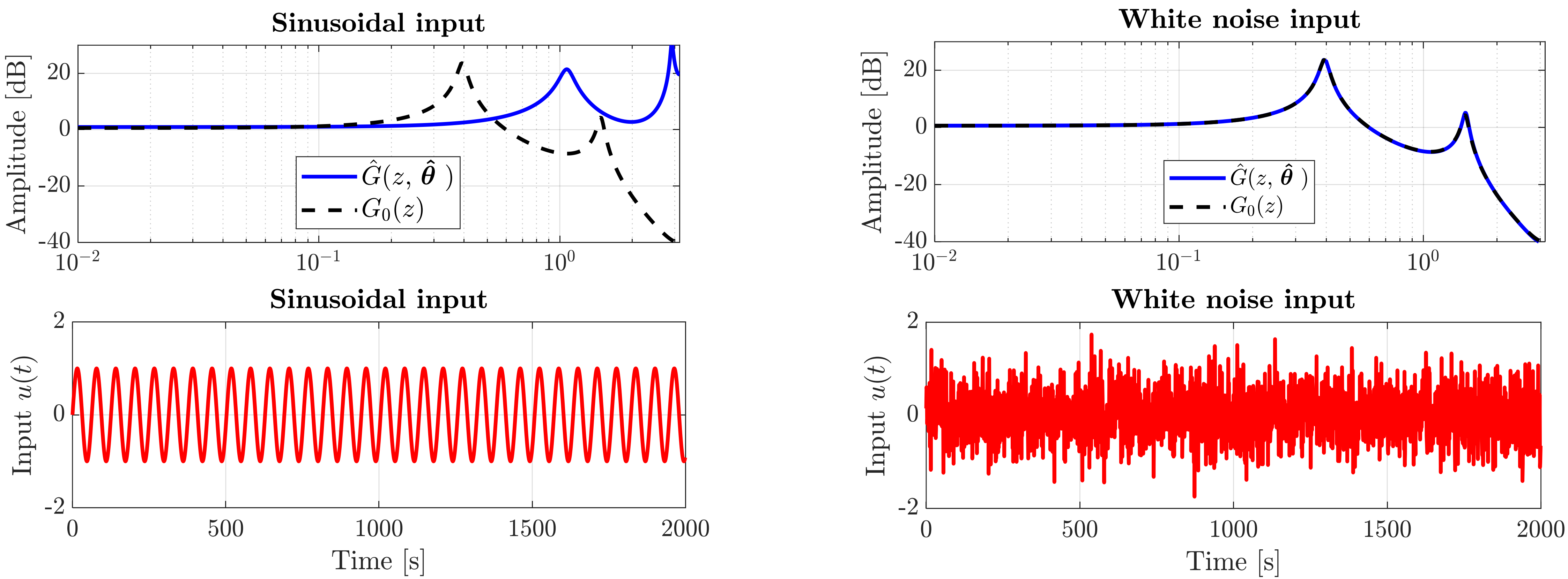
$$\hat{\boldsymbol{\theta}}_N \xrightarrow[N \to +\infty]{} \bar{\boldsymbol{\theta}} \neq \boldsymbol{\theta}^0$$



$$\hat{\boldsymbol{\theta}}_N \xrightarrow[N \to +\infty]{} \bar{\boldsymbol{\theta}} \neq \boldsymbol{\theta}^0$$

Prediction Error Methods: persistent excitation

Experimental identifiability requires the input signal to be **persistently exciting** of enough order, as **white noise, Pseudo-Random Binary Signal (PRBS), Multisine**

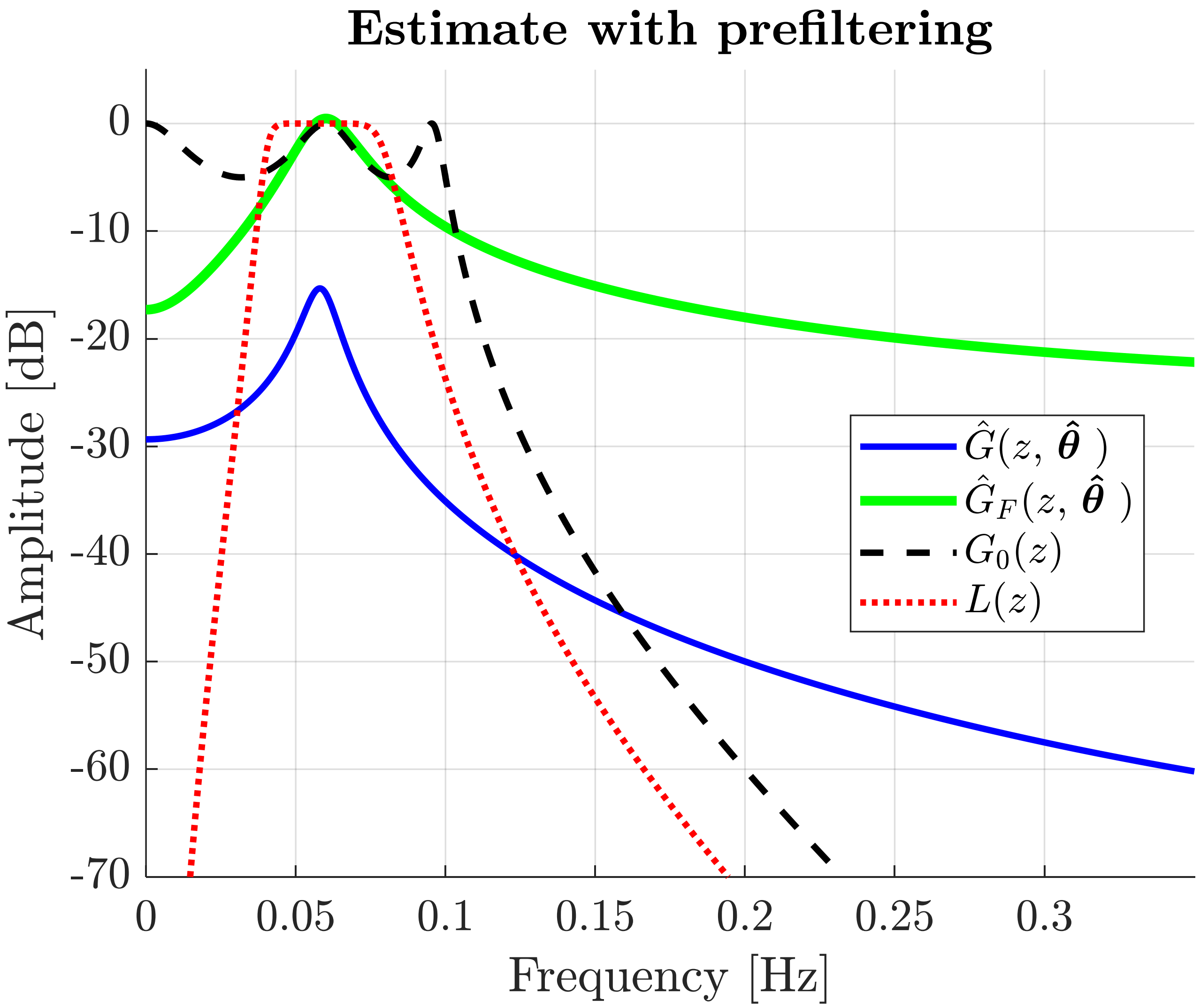


Prediction Error Methods: robustness

When the chosen model is not able to exactly represent the true system, we can **prefilter the data** to perform **bias-shaping**

Focus the modelling efforts in a selected bandwidth

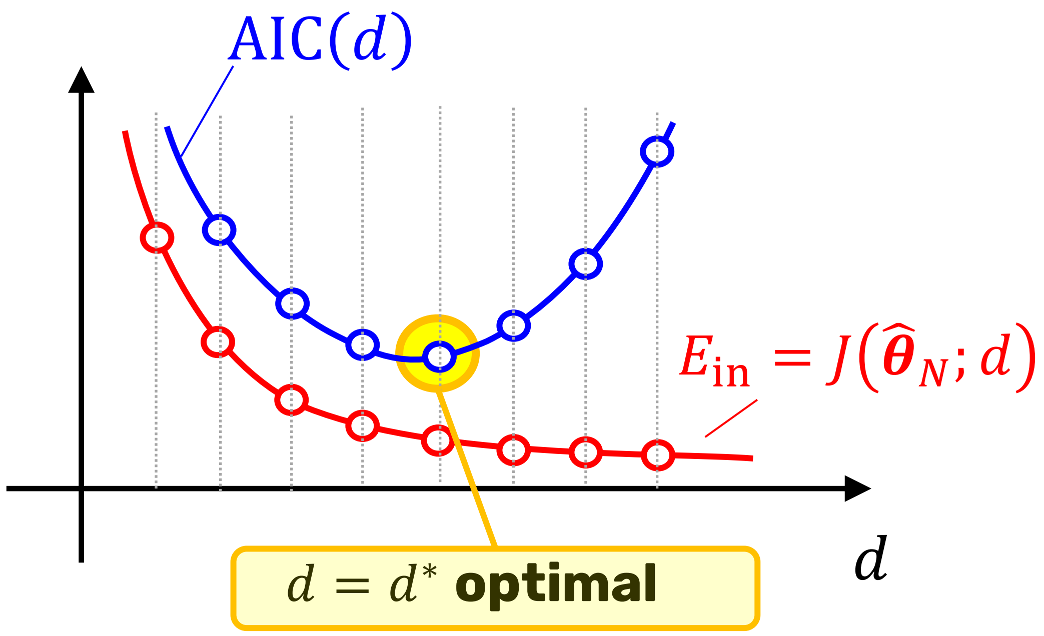
$$\bar{J}(\boldsymbol{\theta}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left|\Delta G\left(e^{j\omega}, \boldsymbol{\theta}\right)\right|^2 \cdot g_F(\omega, \boldsymbol{\theta})\, d\omega$$



Prediction Error Methods: validation techniques

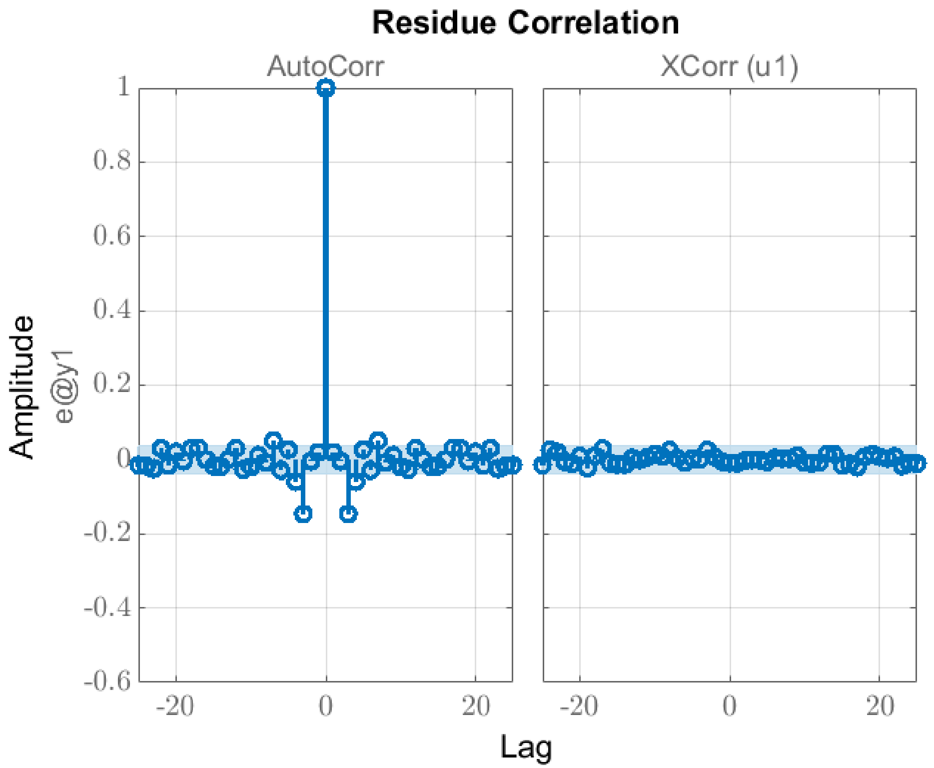
Choosing the model complexity

- Complexity formulas
- Validation



Assessing model goodness

- Residual analysis
- Uncertainty of the estimates
- Prediction and simulation fitness



Empirical Transfer Function Estimate (ETFE)

Suppose we divide the signals $u(t)$ and $y(t)$, both multisines, into sub-sequences $u^{[p]}, y^{[p]}$ which contain the different $p = 1, \dots, P$. We define:

$\breve{U}^{[p]} = \mathrm{DFT}\left(u^{[p]}\right)$ **input DFT**

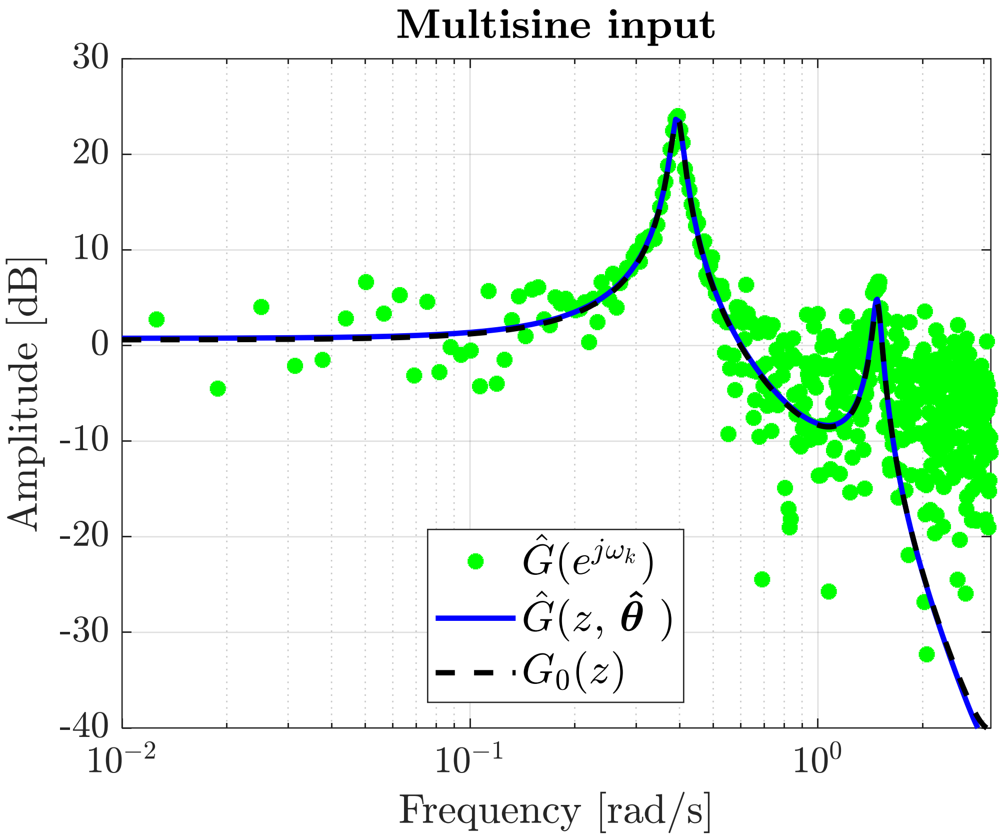
$\breve{Y}^{[p]} = \mathrm{DFT}\left(y^{[p]}\right)$ **output DFT**

$$\widehat{U}(k) = \frac{1}{P} \sum_{p=1}^{P} \breve{U}^{[p]}(k)$$

$$\widehat{Y}(k) = \frac{1}{P} \sum_{p=1}^{P} \breve{U}^{[p]}(k)$$

$$\Rightarrow \quad \widehat{G}(k) = \frac{\widehat{Y}(k)}{\widehat{U}(k)}$$

Empirical Transfer Function Estimate (ETFE)



Outline

1. Presentation of the Adaptive Learning, Estimation and Supervision of dynamical systems (ALES) course

2. What we have learnt so far…

3. **…and what we still have to learn!**

Adaptive Learning, Estimation and Supervision of dynamical systems

A) ADAPTIVE LEARNING

The identification algorithms we have learnt so far process all available data as a **single batch.** This approach has several drawbacks:

- **memory occupancy:** we need to store all data in the computing device
- **computational load:** we may need to perform a big matrix inversion
- **time delay:** if we need to take quick decisions, an immediately available - albeit approximate – model, may be more useful than a perfect model available only after a batch of data has been collected

Adaptive Learning, Estimation and Supervision of dynamical systems

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?

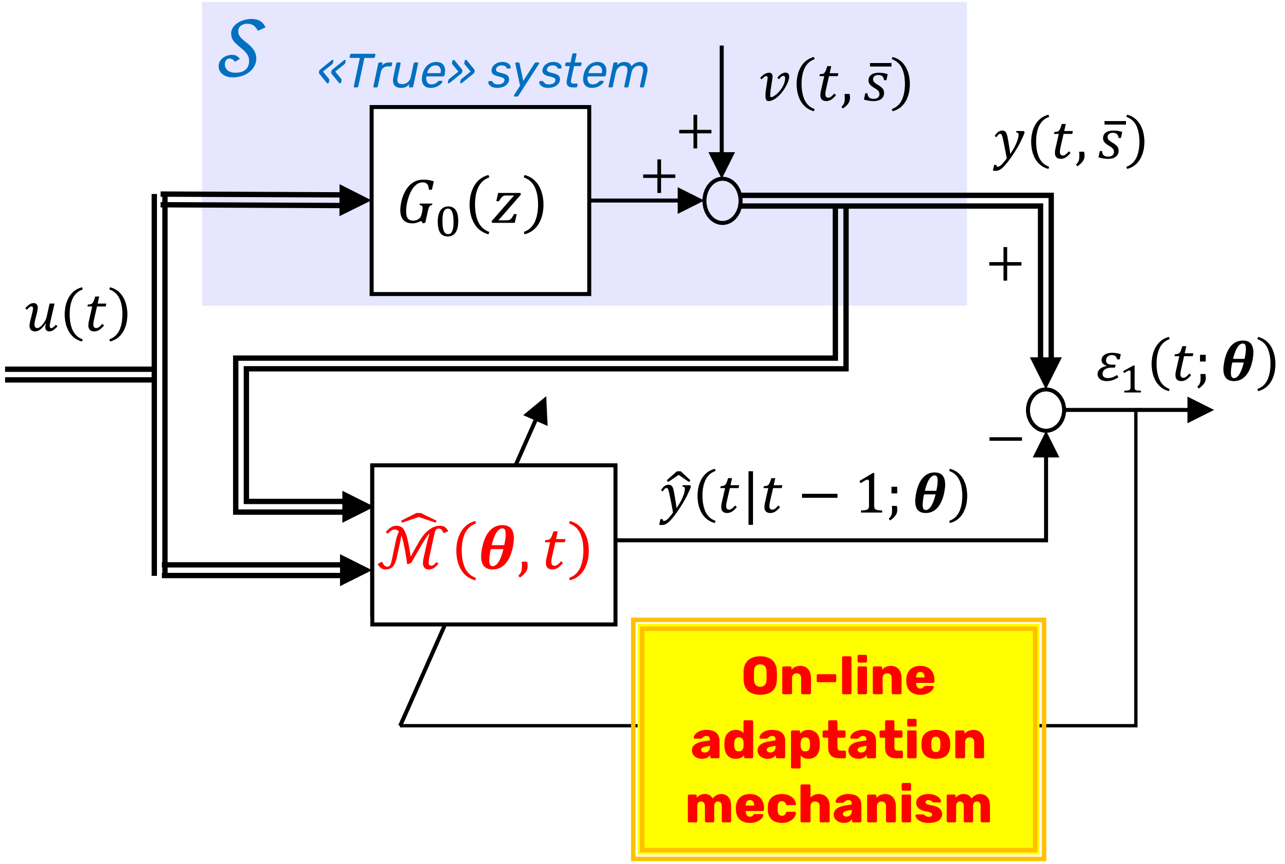
ADAPTIVE IDENTIFICATION METHODS

Adaptive Learning, Estimation and Supervision of dynamical systems

When a model is:

- time-varying, or
- Nonlinear but linearly approximated

the linear model has only **local validity** and must be **online updated** to the most recent data available *(model tracking)*



Adaptive Learning, Estimation and Supervision of dynamical systems

The main applications of **recursive and adaptive learning methods** are:

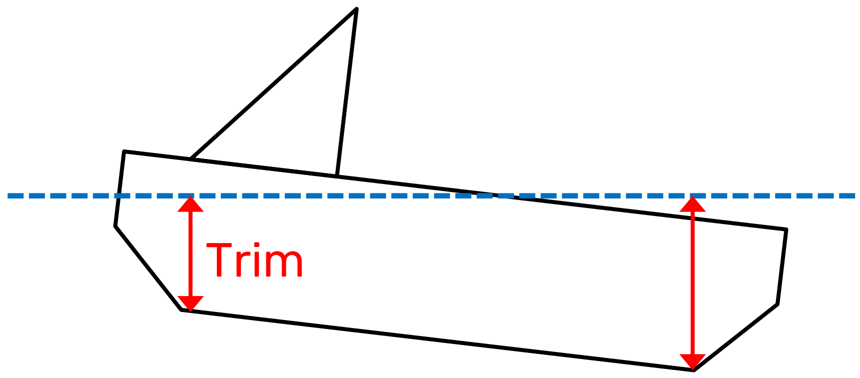
1. **Adaptive filtering and control**: a time-varying controller or filter is tuned to the **currently estimated** model of the process *(iterative learning control, extremum-seeking control, model reference adaptive control, adaptive filtering and equalization)*

2. **Fault diagnosis**: a model of the system is continuously updated to rapidly identify significant **changes in its parameters**. For LTI systems, a change in a parameter might be a symptom of a system degradation due to a fault *(diagnosis of parametric faults)*

Example: ship steering adaptive control

A **ship's heading angle** and position is controlled using the **rudder angle**. For large ships, its response to a change in rudder angle is so **slow** that it is affected by random components as **wind** and **wave motion**

The steering dynamics of the ship depends on its shape, size, loading, trim, and water depth. In particular, **loading** and **water depth** may **vary** with time



The rudder angle controller must be **continuously retuned** to match the dynamics of the system

Example: self-tuning predictive control

Let the system to be controlled an $\mathrm{ARX}(n_a = 1, n_b = 2, k = 1)$ so that:

$$y(t) = ay(t-1) + b_0u(t-1) + b_1u(t-2) + e(t) \qquad e(t) \sim \mathrm{WN}(0, \lambda^2)$$

Under the standard hypotheses for **Minimum Variance (MV) control**, the MV control law for a reference signal $y^0(t)$ is:

Controlli automatici - 6 CFU

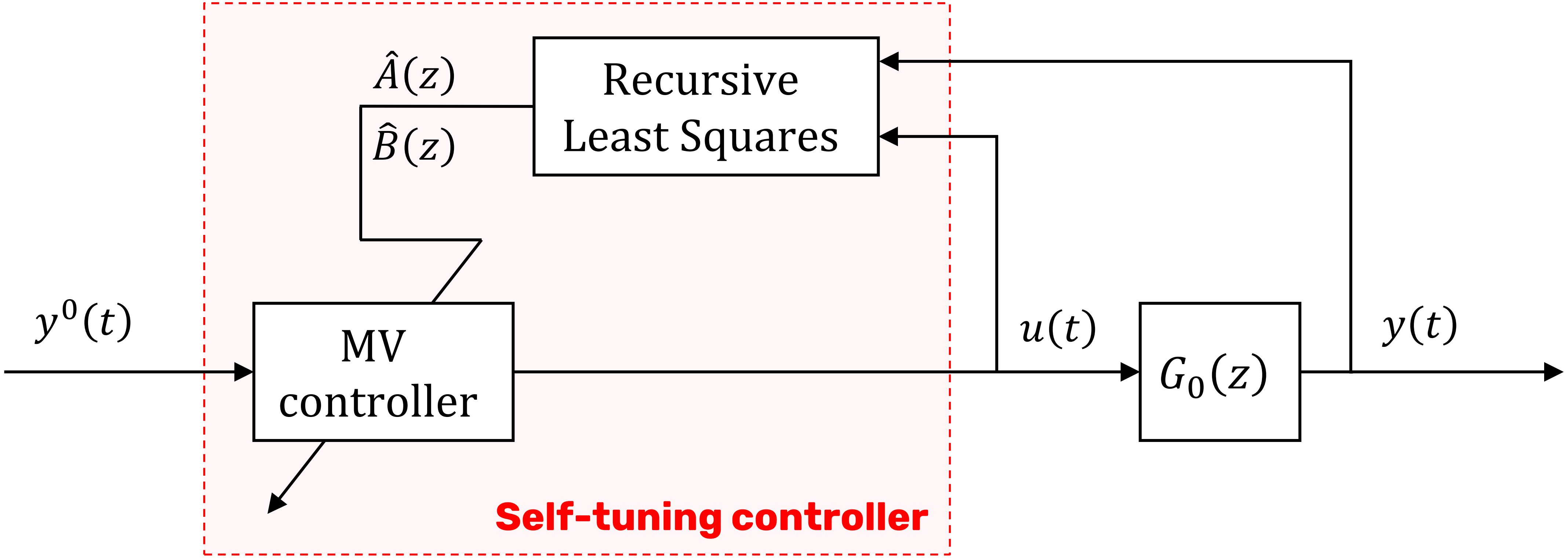
$$u(t) = \frac{1}{b_0}\left(-b_1u(t-1) - ay(t-1) + y^0(t)\right)$$

If the system changes its parameters in time, this control law will be **no more «optimal»** in terms of the **tracking error variance**

$$\mathbb{E}\left[\left(\hat{y}(t|t-1) - y^0(t)\right)^2\right]$$

Example: self-tuning predictive control

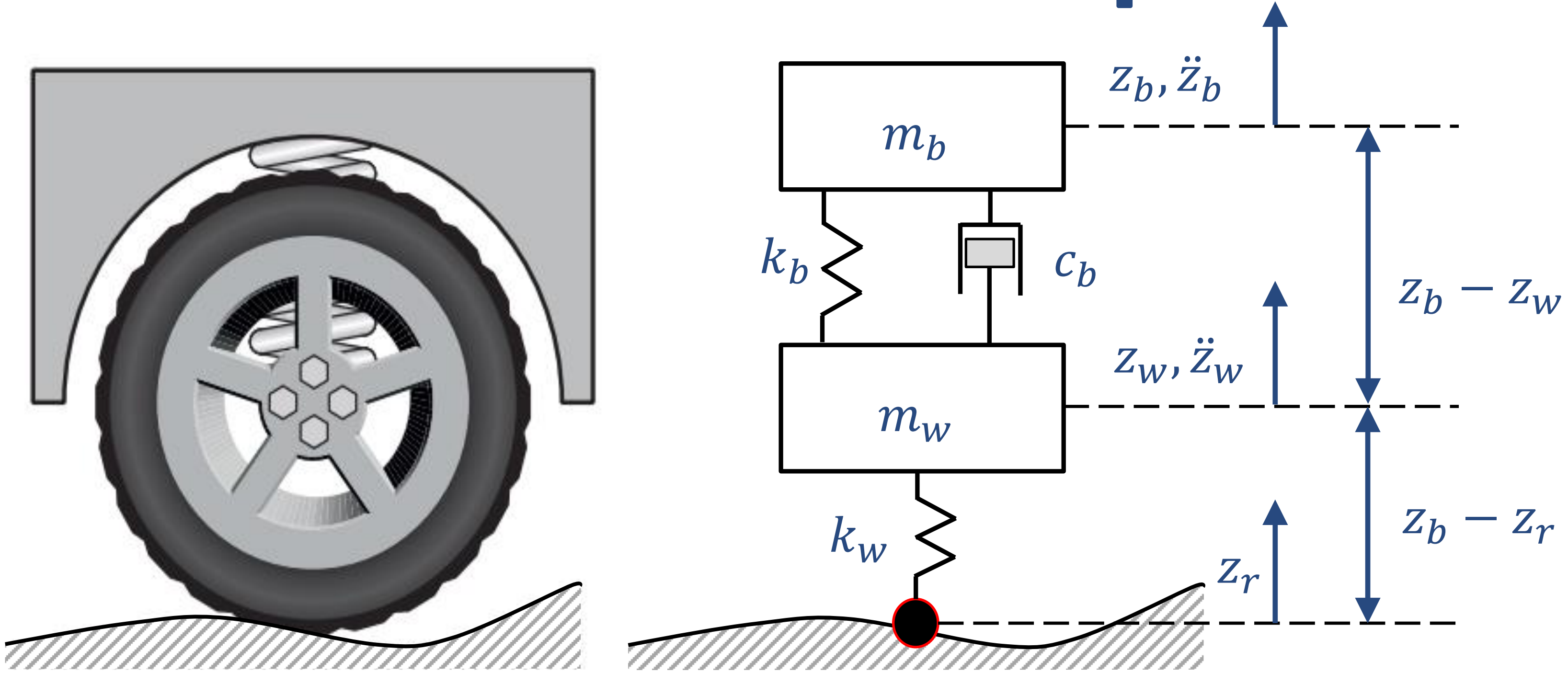
Thus, an **adaptive (self-tuning) control scheme** is envisaged:



The estimation algorithm will provide the estimates $\hat{a}(t), \hat{b}_0(t), \hat{b}_1(t)$ at **each time** t

Example: supervision of an automotive suspension

The **suspension** and the **tire pressure** have a large influence on the vehicle dynamics and are thus highly **safety critical**



Consider a (simplified) **quarter-car** model, where:

- m_b, m_w: body and wheel mass
- $z_b - z_w$: suspension deflection
- k_w: wheel rigidity
- k_b, c_w: suspension rigidity and damping coefficients
- z_r: road level

Example: supervision of an automotive suspension

Force balances lead to:

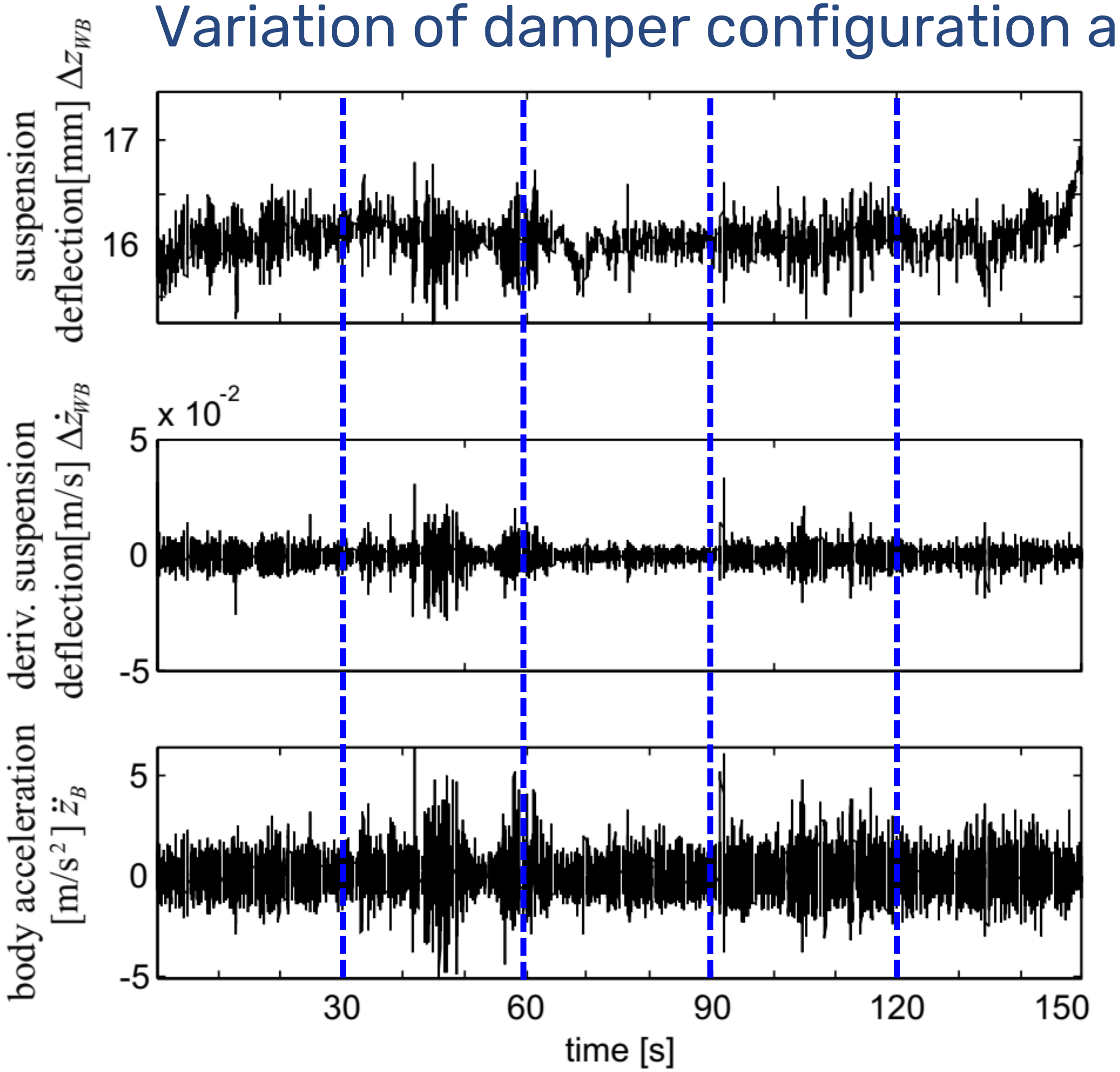
$$\begin{cases} m_b \ddot{z}_b(t) = -k_b\big(z_b(t) - z_w(t)\big) - c_b\big(\dot{z}_b(t) - \dot{z}_w(t)\big) - m_b g \\ m_w \ddot{z}_w(t) = k_b\big(z_b(t) - z_w(t)\big) + c_b\big(\dot{z}_b(t) - \dot{z}_w(t)\big) - k_w\big(z_w(t) - z_r(t)\big) - m_w g \end{cases}$$

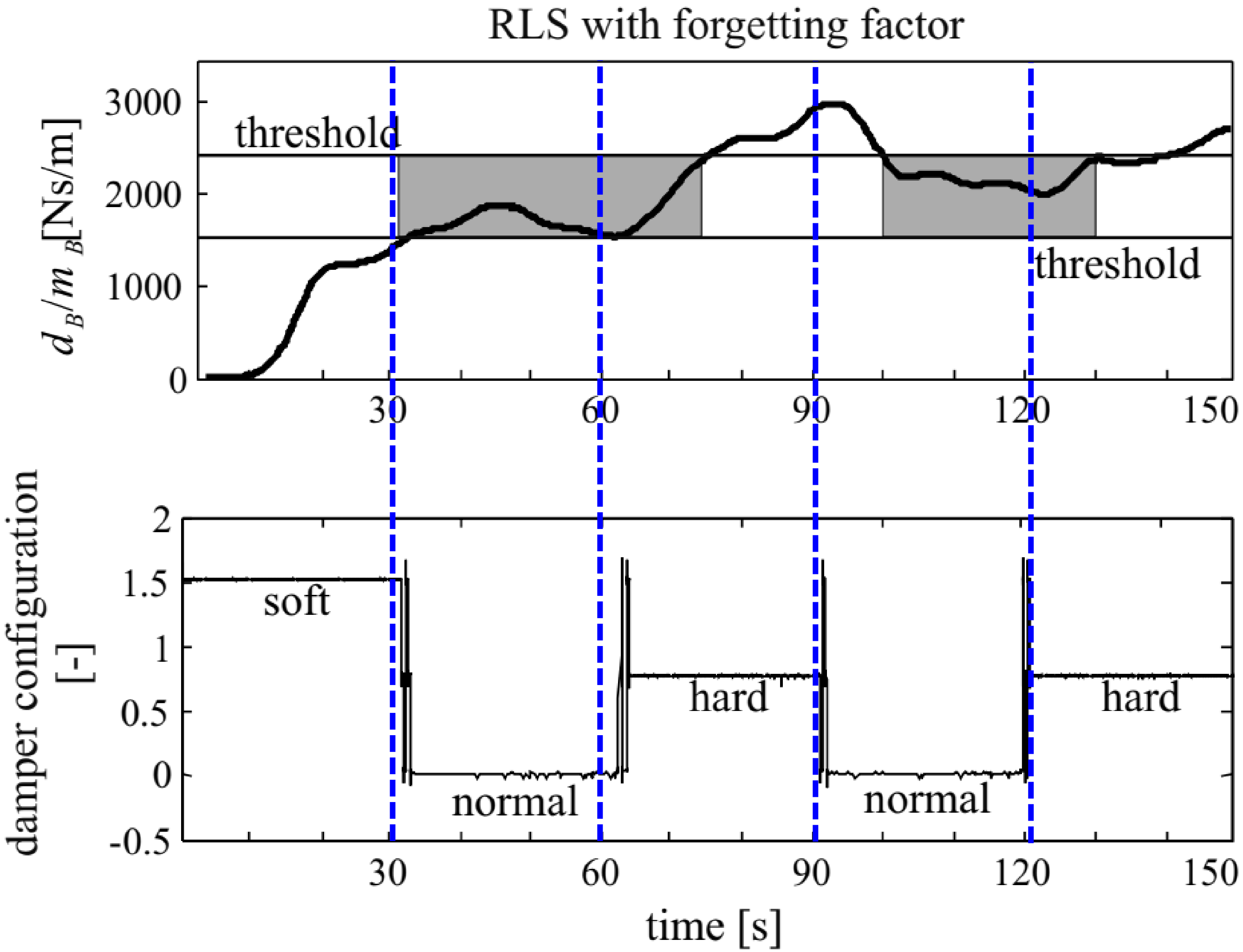
Assume that the body mass m_b and damper spring rigidity k_b are known. It is possible to **estimate the suspension viscous coefficient** c_b with **linear regression** measuring the body acceleration $\ddot{z}_b$ and the suspension deflection $z_b - z_w$

However, for a **fault detection** purpose (e.g. a **worn out of the suspension** which cause a change in the damping coefficient) **recursive methods** are needed

Example: supervision of an automotive suspension

Variation of damper configuration at $t = 30, 60, 90, 120\ s$





Adaptive Learning, Estimation and Supervision of dynamical systems

B) ESTIMATION AND IDENTIFICATION

Until now, we focused on **Prediction Error Methods (PEM)** for **SISO** LTI system identification. These methods are optimal if some assumptions holds:

- The input and output signals are assumed to be **stationary** stochastic processes (ssp), so that **no transients** must be present in the signals
- The input has to **be persistently exciting** of a sufficient order
- Only **input-output models** can be identified

Furthermore, they may optimize a **non-convex** cost function

Adaptive Learning, Estimation and Supervision of dynamical systems

However, sometimes we:

- may have at disposal only signals **with transients**, **not stationary** and **low-order exciting** inputs

- may need a **state-space models** instead of an input-output one (e.g. for Kalman filtering advanced control as **LQR** or **MPC**, model-based **fault diagnosis** approach)

Advanced and Multivariable control - 6 CFU

SUBSPACE IDENTIFICATION METHODS

Adaptive Learning, Estimation and Supervision of dynamical systems

Subspace identification approaches are not based on optimization, and they are often used as an **initialization step** for PEM approaches

However, subspace methods allow a simpler identification scheme for **MIMO systems**, as the **number of parameters** to be estimated is often **lower** with respect to PEM approaches

MIMO SYSTEM IDENTIFICATION

Example: estimate a large MIMO system

Suppose that $m_u = 40$ inputs, $p = 28$ outputs, $n = 22$ states

Let an **Input-Output model** assume that each I/O relation is modeled by a **100° order FIR** model. Then, the **total number of parameters** to be estimated is

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

State-space model

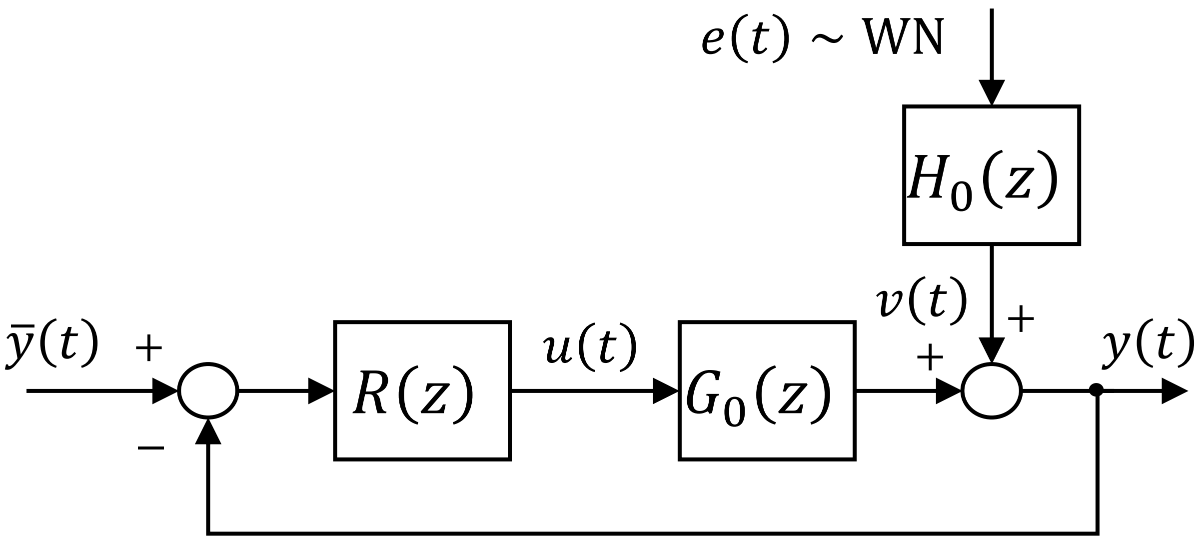
$$\begin{cases} \underset{n \times 1}{\boldsymbol{x}(t+1)} = \underset{n \times n}{A} \cdot \boldsymbol{x}(t) + \underset{n \times m_u}{B} \cdot \underset{m_u \times 1}{\boldsymbol{u}(t)} \\ \underset{p \times 1}{\boldsymbol{y}(t)} = \underset{p \times n}{C} \cdot \boldsymbol{x}(t) + \underset{p \times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

$$\begin{aligned} \text{number of params} &= n \cdot n + n \cdot m + n \cdot p + p \cdot m \\ &= 22 \cdot 22 + 22 \cdot 40 + 22 \cdot 28 + 28 \cdot 40 \\ &= \boxed{3100} \end{aligned}$$

Adaptive Learning, Estimation and Supervision of dynamical systems

Consider a **closed-loop** SISO control system:

- Usual operation of many plants, in particular if $G_0(z)$ is **open-loop unstable**



- Sometimes the controller is **intrinsically present** (e.g. biomedical, economic systems)

- The controller can have the effect of **«linearizing»** a nonlinear plant

Main (fundamental) difference with the open-loop setting: $u(t)$ and $v(t)$ are **correlated**

Adaptive Learning, Estimation and Supervision of dynamical systems

In most situations the goal of **closed-loop identification** is to estimate $G_0(z)$ and possibly $H_0(z)$. Sometimes, also the estimate of $R(z)$ is of interest. Questions that arise are:

- Are the estimates of $G_0(z)$ and $H_0(z)$ **consistent** when $\mathcal{S} \in \mathcal{M}(\boldsymbol{\theta})$?
- Are the estimates of $G_0(z)$ and $H_0(z)$ **consistent** when only $G_0 \in \mathcal{G}(\boldsymbol{\theta})$?
- Is it possible to **tune the bias** of the estimate?
- What is the **variance** of the estimates?

CLOSED-LOOP IDENTIFICATION METHODS

Adaptive Learning, Estimation and Supervision of dynamical systems

C) SUPERVISION OF DYNAMICAL SYSTEMS

The **supervision (**or **diagnosis)** of dynamical systems aim to **discover anomalies** in the plant, its actuators and sensors, using **measurements** and a **model** of the plant

- How to **detect** if a process component is «healthy» or «faulty»?
- How to **isolate** which component is faulty?
- How to be **robust** despite of *noises*, *disturbances* or *model inaccuracies*?

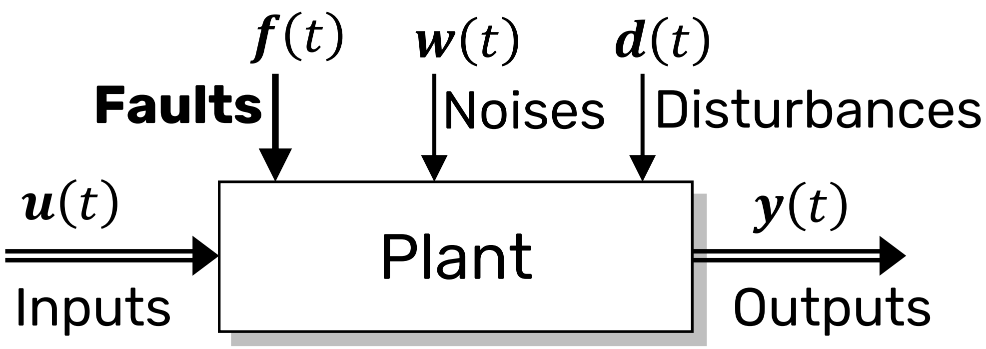
FAULT DIAGNOSIS METHODS

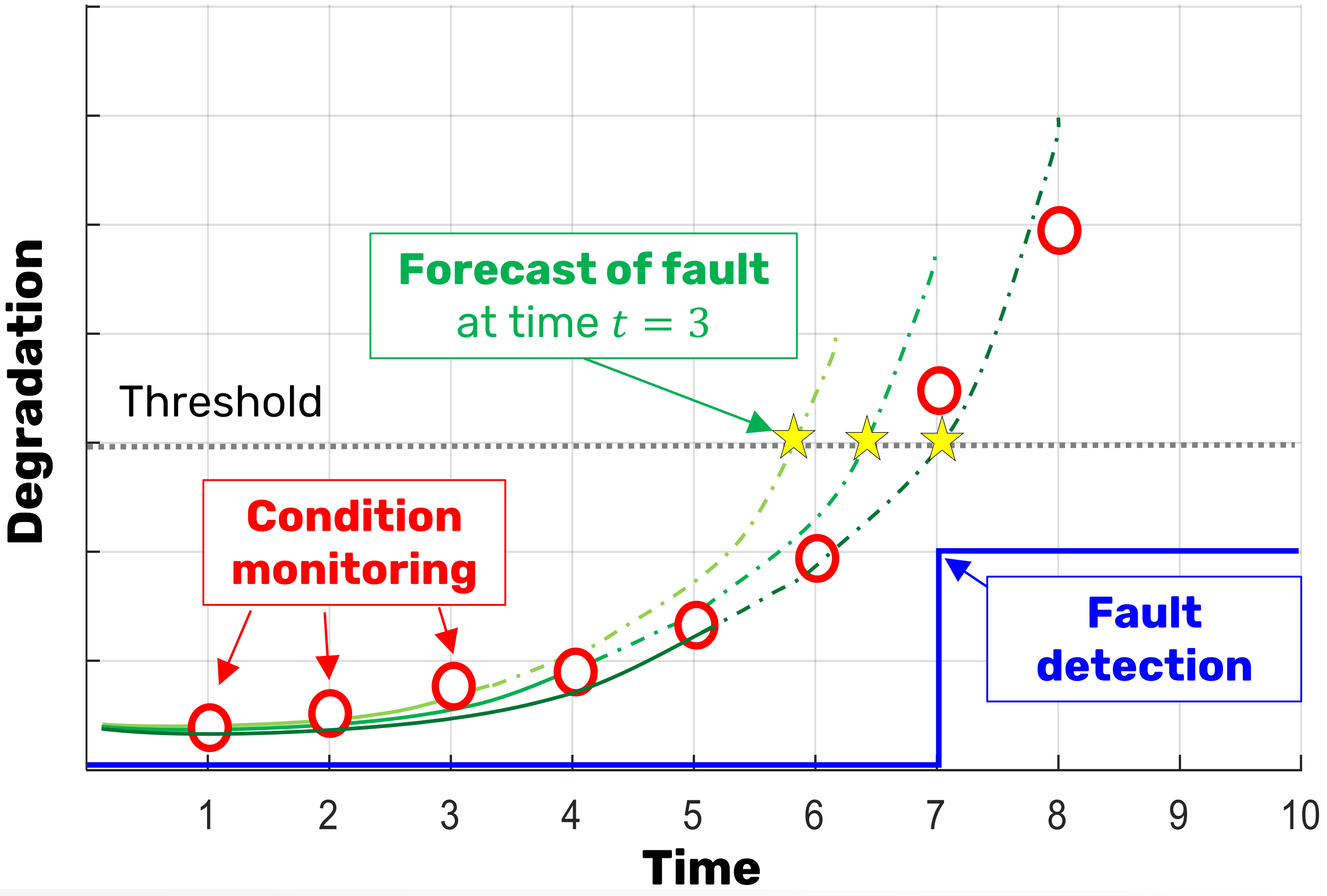
Adaptive Learning, Estimation and Supervision of dynamical systems

1. Fault **detection** and **isolation**

2. Condition **monitoring**

3. **Prognosis**

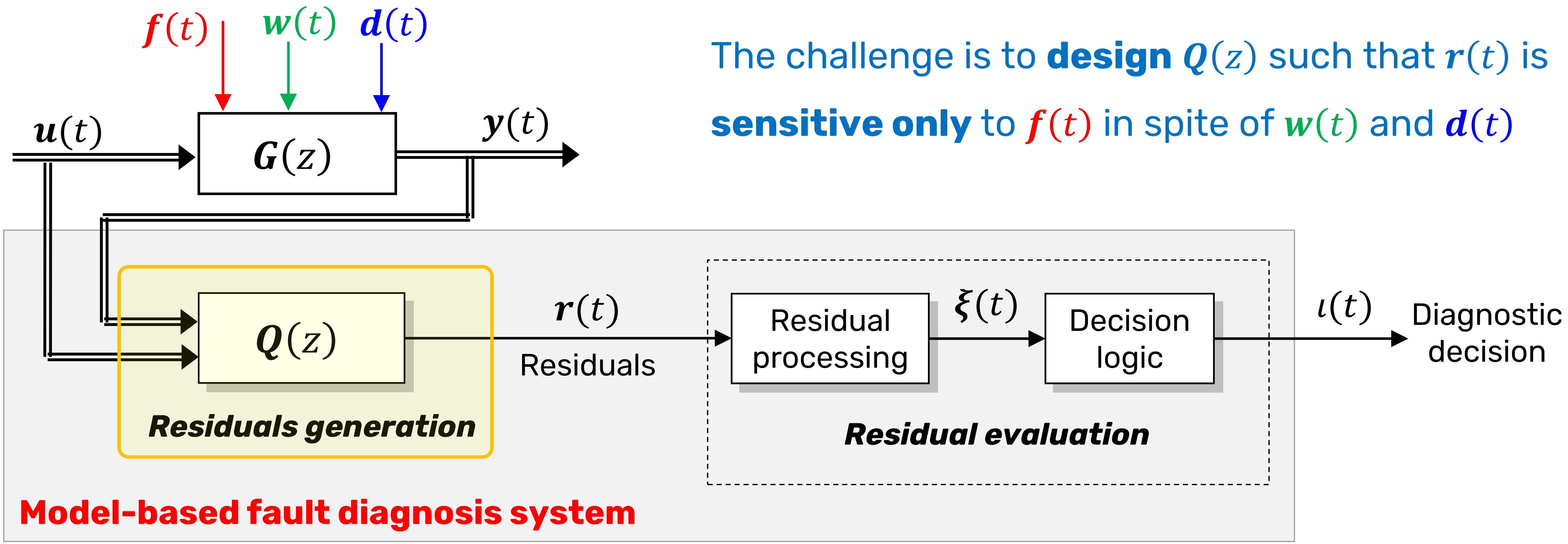




Adaptive Learning, Estimation and Supervision of dynamical systems

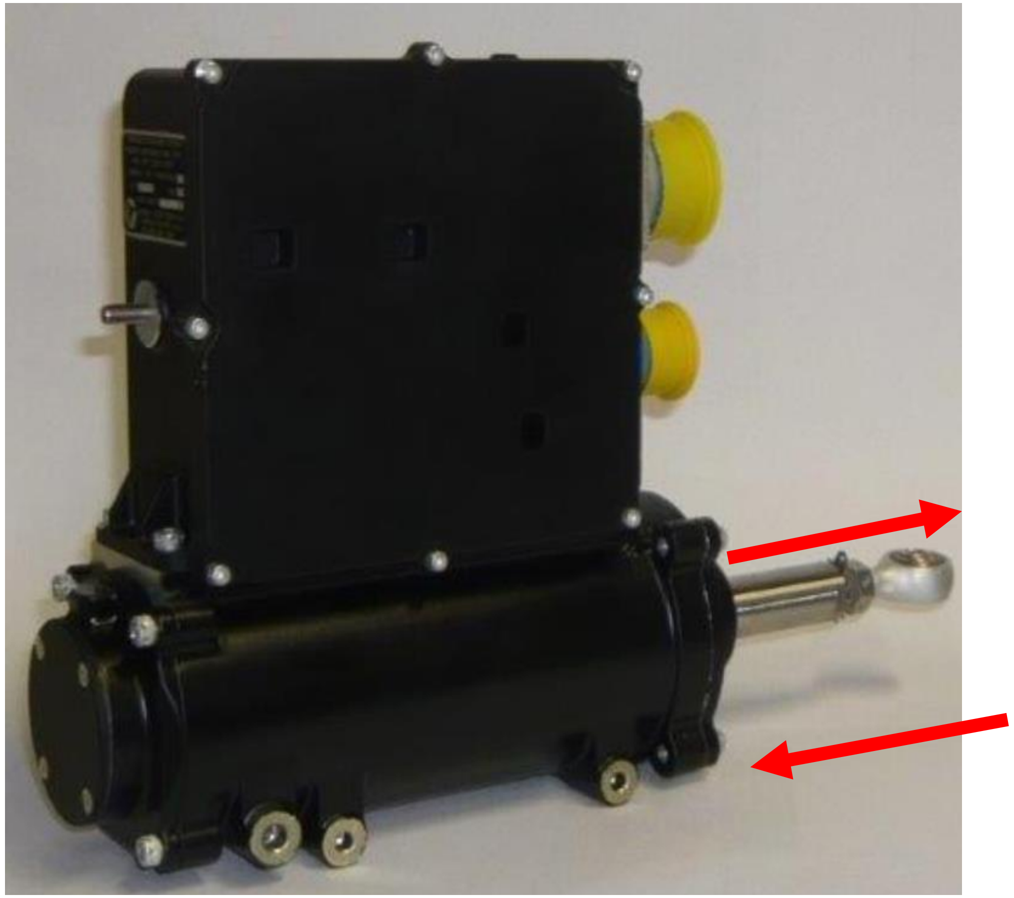
We will mainly focus on the **model-based approach** to fault diagnosis

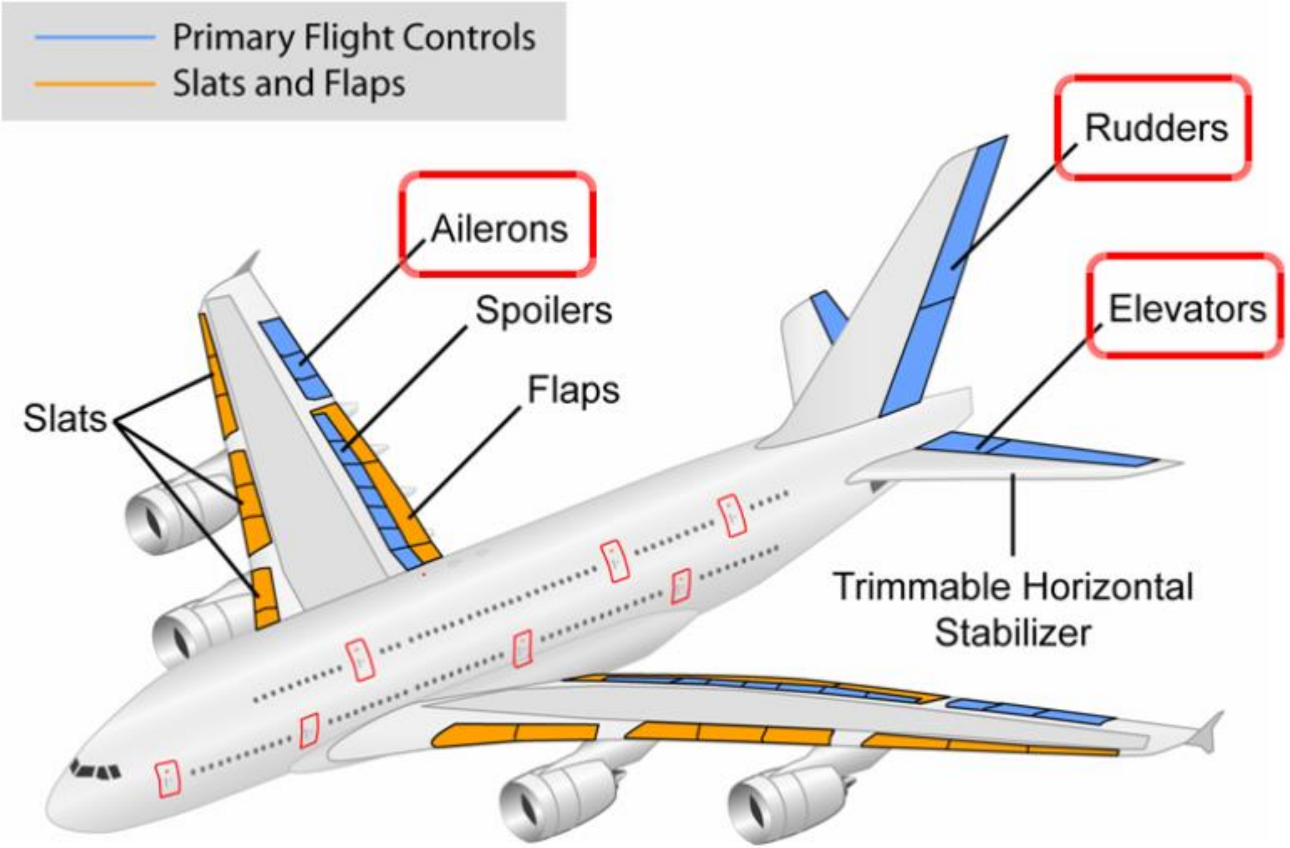
The challenge is to **design** $\boldsymbol{Q}(z)$ such that $\boldsymbol{r}(t)$ is **sensitive only** to $\boldsymbol{f}(t)$ in spite of $\boldsymbol{w}(t)$ and $\boldsymbol{d}(t)$



Example: model-based EMA jamming detection

Consider an EMA actuating **primary flight surfaces:**

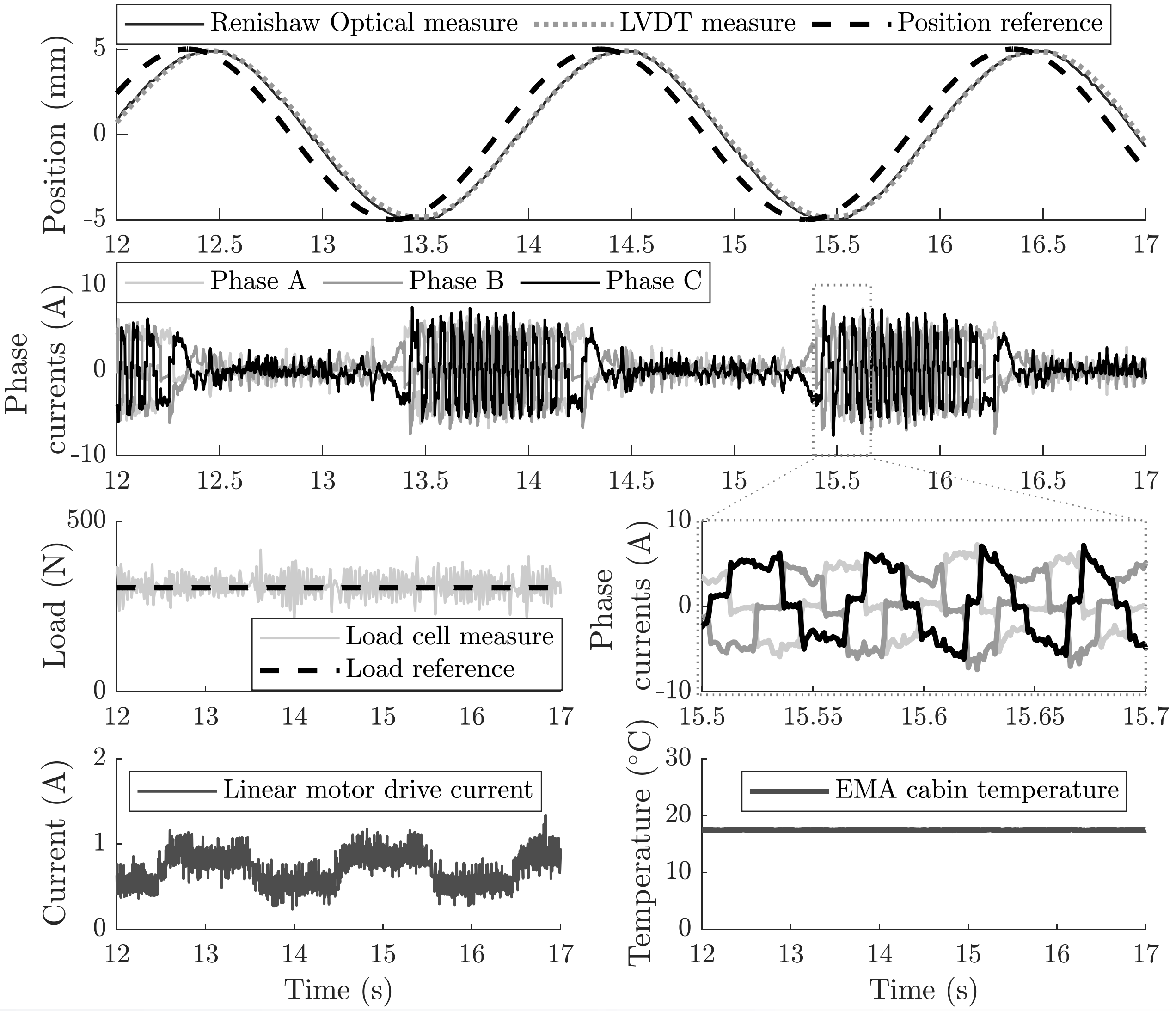




Example: model-based EMA jamming detection

Test bench measurements

1. EMA Phase currents
2. EMA LVDT position
3. Linear motor position
4. EMA Reference position
5. Load cell



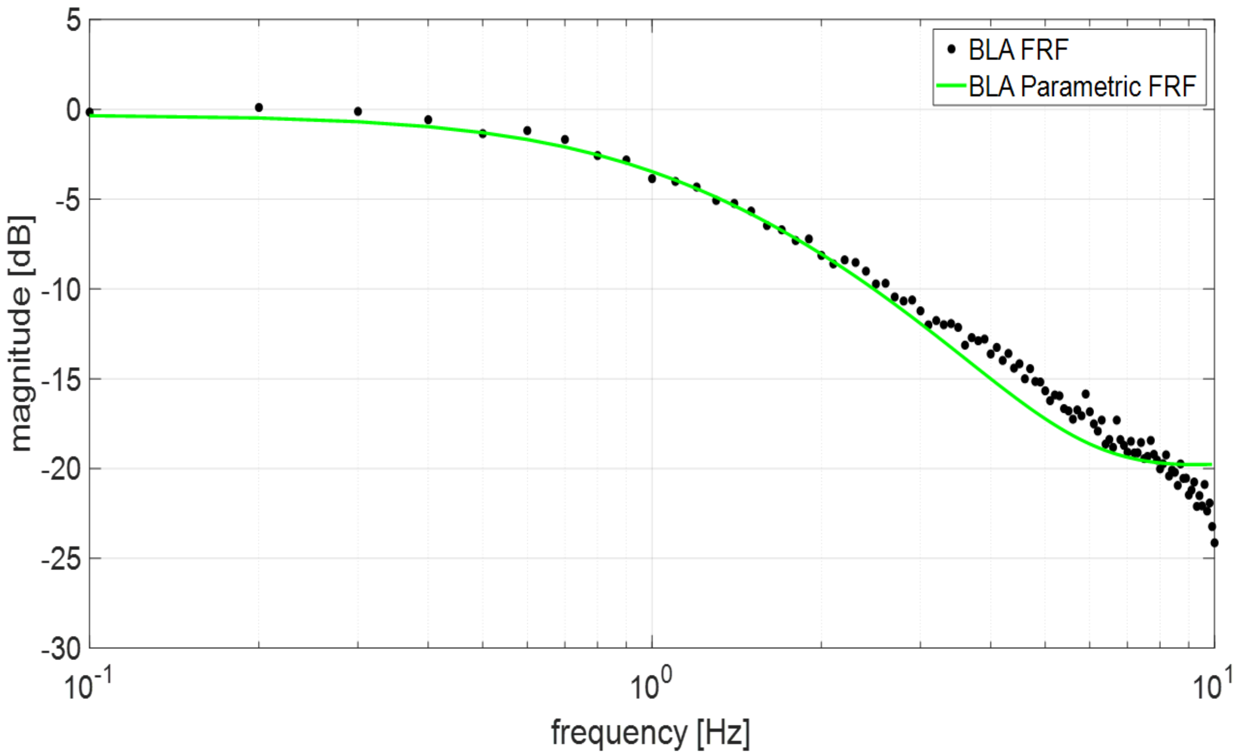
Example: model-based EMA jamming detection

Identification method:

1. **Multisine** excitation

2. **Nonparametric estimate** of the closed-loop transfer function

3. **Parametric estimate** fitting FRF data



$$\frac{X(z)}{\bar{X}(z)} = G(z) = \frac{0.06892 - 0.1732z^{-1} + 0.1266z^{-2}}{1 - 1.624z^{-1} + 0.6467z^{-2}}$$

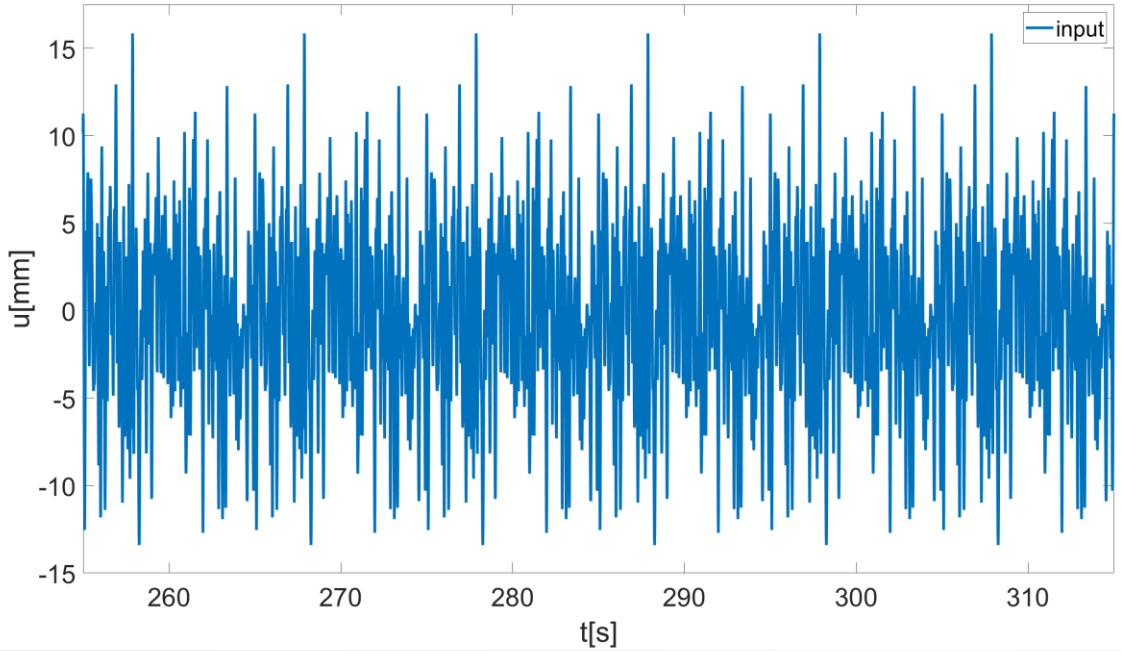
- $\bar{x}(t)$: position reference
- $x(t)$: position measure LVDT

Example: model-based EMA jamming detection

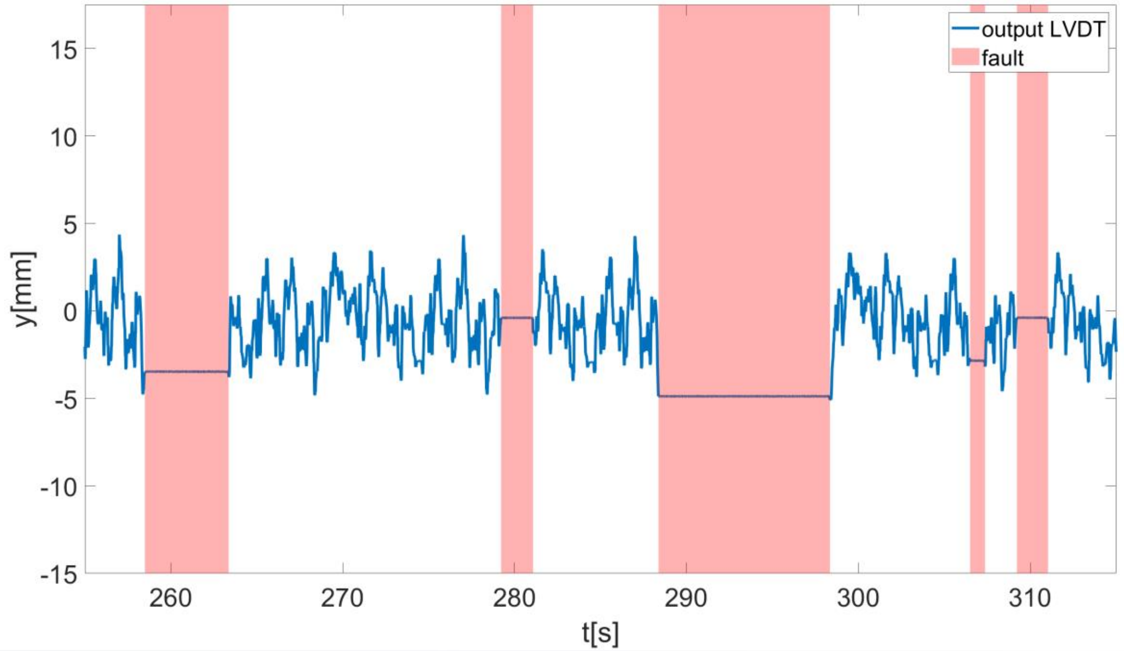
When the degradation took over, the **EMA undergone small-jams** during the operation.

The output position signal, as measured by the LVDT sensor, shows **constant value.** We modeled the fault as an **actuator fault**.

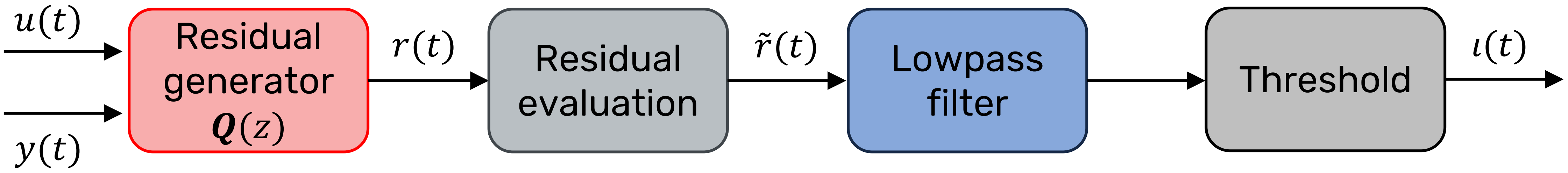
Multisine position reference

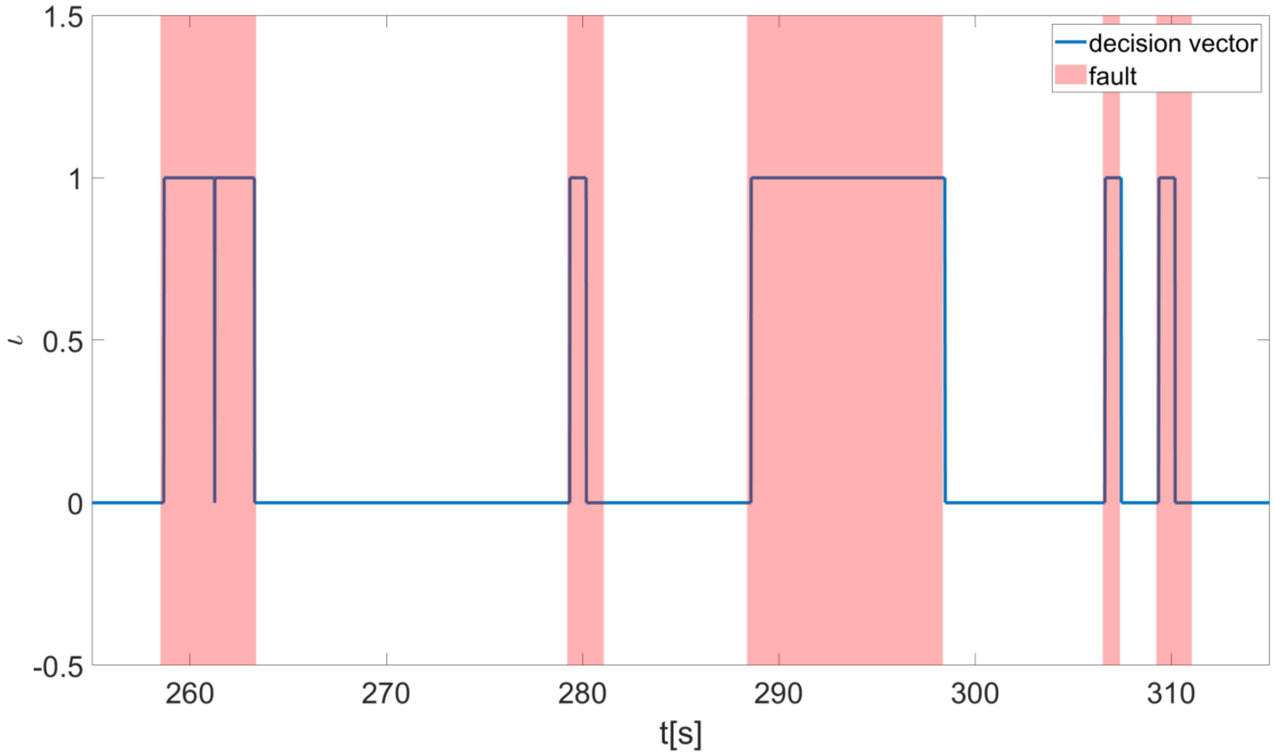


Output position (LVDT)



Example: model-based EMA jamming detection





UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione