UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Dipartimento
di Ingegneria Gestionale,
dell'Informazione e della Produzione

Lesson 14.

Fault diagnosis II

Model-based approaches

DATA SCIENCE AND AUTOMATION COURSE

MASTER DEGREE SMART TECHNOLOGY ENGINEERING

TEACHER
Mirko Mazzoleni

PLACE
University of Bergamo

Outline

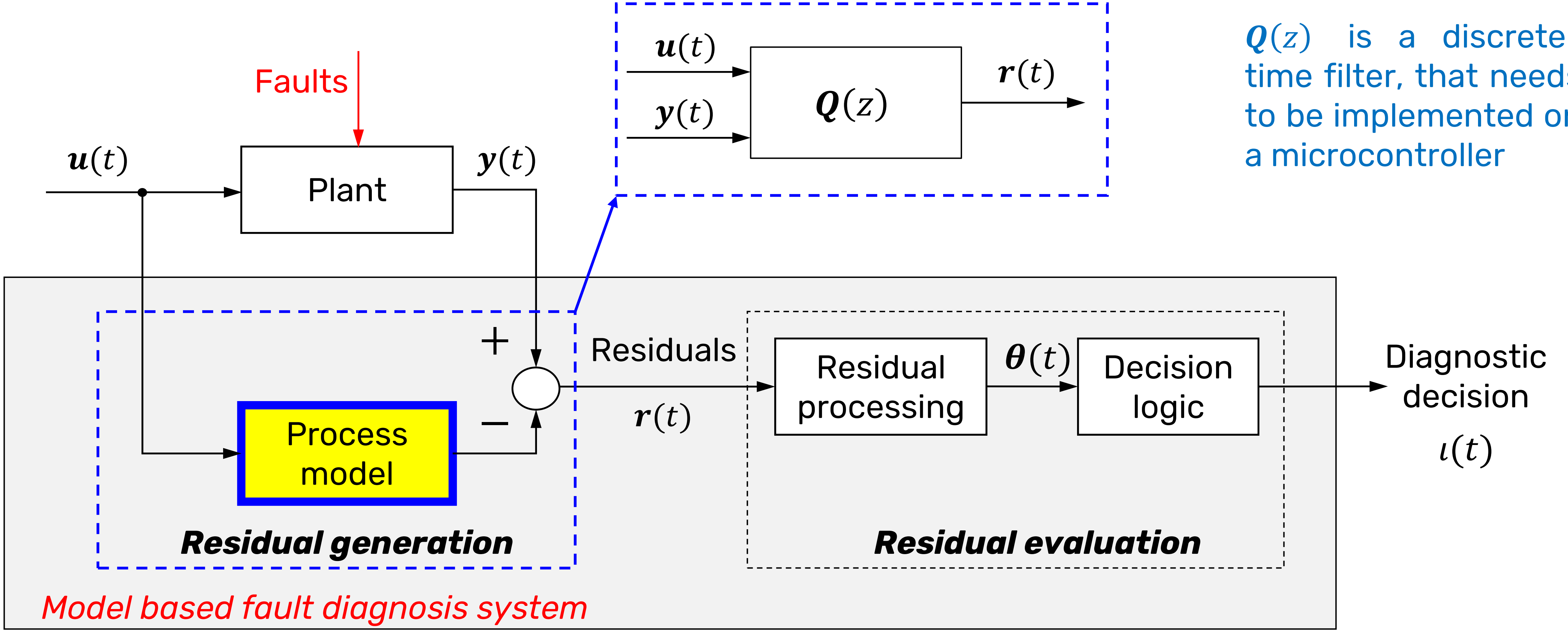
1. Schematic of the approach
2. Dynamical systems
3. Parity space approach
4. Diagnostic observer
5. Application to EMA fault detection

Outline

1. **Schematic of the approach**
2. Dynamical systems
3. Parity space approach
4. Diagnostic observer
5. Application to EMA fault detection

Model-based fault diagnosis

A model of the system is developed to design a **residual generator** $\boldsymbol{Q}(z)$

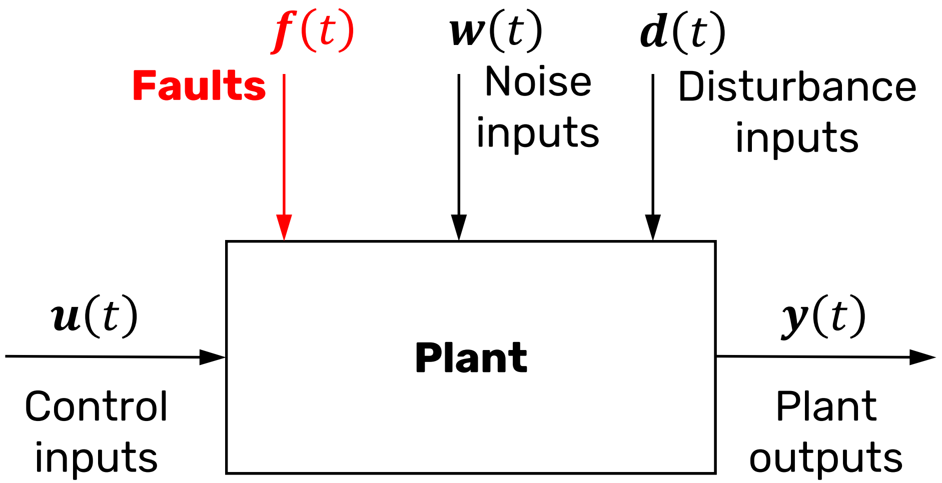


$\boldsymbol{Q}(z)$ is a discrete-time filter, that needs to be implemented on a microcontroller

Modeling a plant with faults

A **modeling scheme for fault diagnosis** considers the following external inputs:

- $\boldsymbol{y}(t)$: plant outputs (measureable)
- $\boldsymbol{u}(t)$: plant inputs (measureable)
- $\boldsymbol{d}(t)$: disturbance inputs *(not measureable)*
- $\boldsymbol{w}(t)$: noise inputs *(not measureable)*

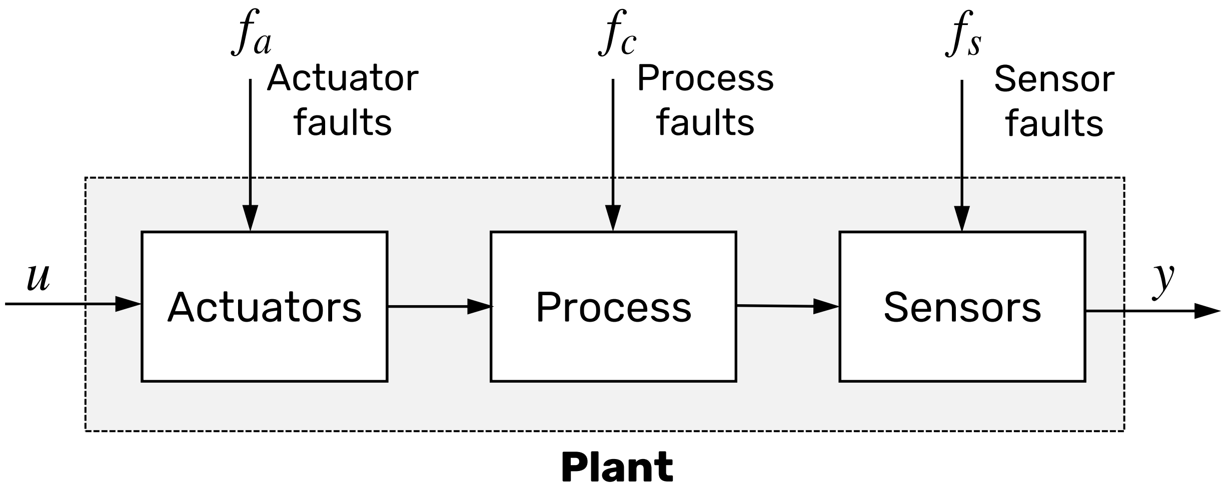


- **Disturbances:** include unknown uncontrollable inputs *(wind shears, crosswinds, load variations)*
  - ✓ If the transfer function from $\boldsymbol{d}$ to $\boldsymbol{y}$ is known, their effect on $\boldsymbol{y}$ can be **completely rejected**
- **Noises:** include uncertainties such as random noises or parametric model uncertainties
  - ✓ They can **only be attenuated** up to a certain extent

Classification of fault types

Faults can be classified based on their **location (physical classification)...**

- **Actuator faults:** a change in the characteristics of an actuator leading to a loss of efficiency or even to a complete breakdown *(jamming, runaway, floating, loss of effectiveness)*
- **Sensor faults:** erroneous measurements obtained with a defective sensor *(freezing, drift, bias, loss of accuracy)*
- **Process faults:** malfunction of an internal component due to excessive variation of some physical parameter *(structural damage, leakage, shortcut)*



Classification of fault types

…or based on how it is possible to **model their effect on the system:**

- **Additive faults:** fault modelled as a *fictive input* f, which acts independently on the plant inputs and outputs. The case $f = 0$ corresponds to *the fault-free case.* *(some types of **actuator faults** (jamming, runaway, oscillatory fault case), some types of **sensor faults** (bias, drift), operational wear and tear)*

- **Parametric faults:** fault whose effects on the plant depend on the magnitude of some internal signals or known inputs. *(**parametric faults**, some types of **actuator faults** (loss of efficiency, disconnection, stall load)*

The classification is important to adopt the **correct modeling** and **diagnosis technique**

Outline

Dynamical systems

A system (physical object that accepts inputs and produces outputs) is said to be **dynamical** if the output $y(t)$ at a certain time t **does not depend only** of the input $u(t)$ at the same time t, but also on the initial system condition $\boldsymbol{x}(0)$.

- In an **electromechanical motor**, the relation between the motor current and the motor speed can be described by a dynamical model
- The force generated by a **skeletal muscle** contraction will depend by the viscous damping given by the tissue and on the elastic storage properties by the sarcolemma and tendons
- **Flow equation** of blood through the vessels depend on pressures dynamics

An example of **static system** is the resistor: $i(t) = \Delta V(t)/R$,

$i(t) = \text{current in the resistor [A]}, \Delta V(t) = \text{voltage drop on the resistor [V]}, R = \text{resistance [}\Omega\text{]}.$

Mathematical models of dynamical systems

A **dynamical model** is **mathematical object** that can be used to analyze the behavior of a dynamical system

Tipically they are represented using a **set of equations** that explain the **relation between the variables** involved in the phenomenon, i.e. how the variables **evolve in time**

Models represent only a **simplified version** of the real phenomenon:

- «All models are wrong, but some are useful»

There are **different types** of dynamical models:

- Linear \ nonlinear
- Time invariant \ time variant
- Continuous \ discrete time

Example 1: Infectious disease

We want to model the **number of people infected** with a contagious disease

Suppose to **measure** the number of infected people one time every day **(sampling time)**

Let's define as $I(t)$ the **number of infected** people at day $t \in \mathbb{N}_{>0}$ (discrete-time case)

t is a multiple of the sampling period T_s

$$I(t+1) = I(t) + \lambda \cdot I(t)$$

Difference between new infected and deaths\recovered

- λ is the disease spreading rate (the higher λ, the faster the spreading. If negative, the epidemic subsides)

The number of infected people **at a certain day** depends on the number of infected in the **previous day** → The dynamical system has a **«memory» property**

Example 1: Simulation

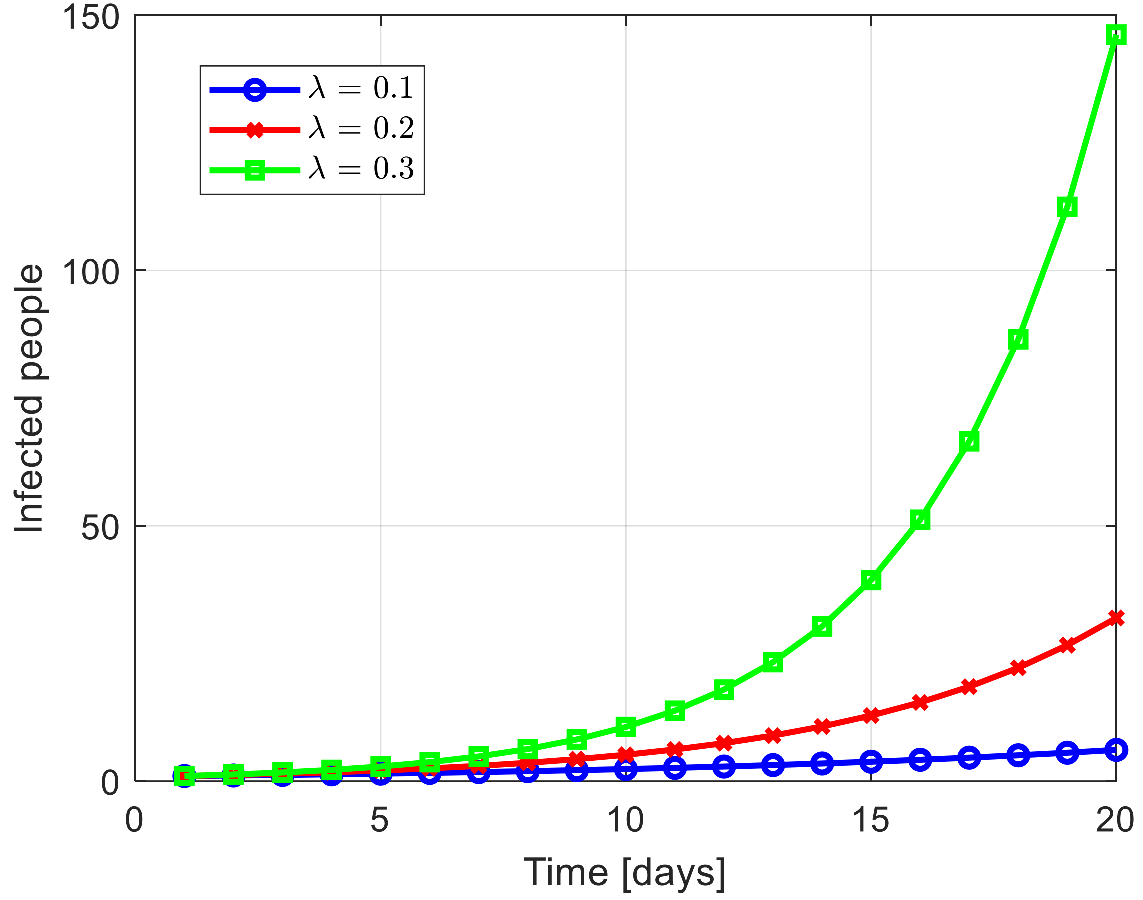
In order to analyze the phenomena, we can simulate the model. To simulate this model we have to know the **initial value** of infected people

```
I(1) = 1;        Initial condition
T = 20; lambda = 0.1;
for t = 2 : 1 : T
    I(t) = I(t-1) + lambda*I(t-1);
end
```

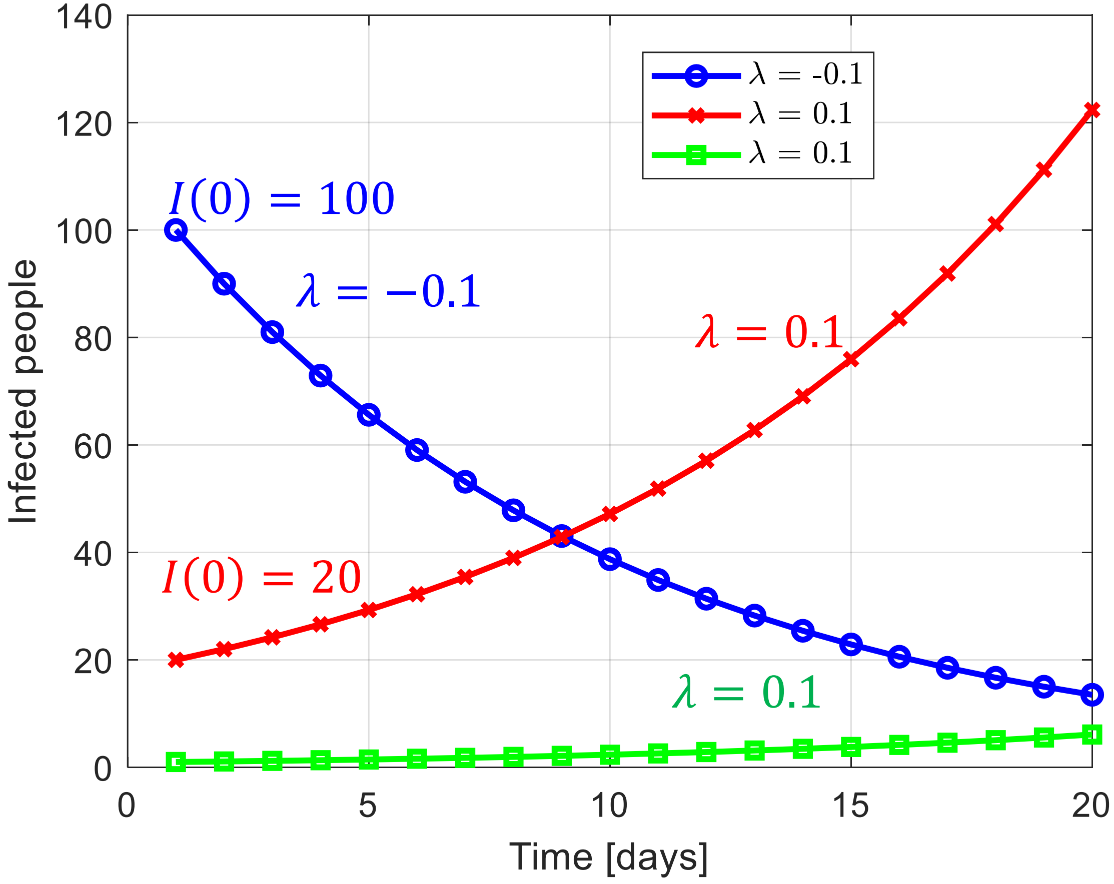
The number of infected people **at a certain day** depends on the number of infected in the **previous day** → The dynamical system has a **«memory» property**

Example 1: Simulation

Different rates λ, $I(0) = 1$



Different initial conditions $I(0)$



Example 1: Assumptions

The previous model is a **simplified** version of the phenomenon. It represents reality up to a certain extent

$$I(t+1) = I(t) + \lambda \cdot I(t)$$

A lot of **assumptions** are used, for example:

- The are **infinite** infectable people
- The number of people is a **real** number

If we want to relax these assumptions, and **more closely mimic reality**, we need a more complex model

Example 2: SIR

The **SIR** (Susceptible, Infected, Recovered) model is vastly used to model the **dynamics of an epidemic**

Each infected person will infect a proportion σ of the susceptible people ☹

$$\begin{cases} S(t+1) = S(t) - \sigma \cdot S(t) \cdot I(t) \\ I(t+1) = I(t) + \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\ R(t+1) = R(t) + \rho \cdot I(t) \end{cases}$$

A proportion ρ of the infected people recovers ☺

- σ: disease spreading rate, $\in [0,1]$
- ρ: recovery rate, $\in [0,1]$

- $S(t)$: number of people that are susceptible to the disease
- $I(t)$: number of infected people
- $R(t)$: number of people that recover for the disease and are not susceptible anymore

Example 2: SIR

$$\Delta S(t+1) = S(t+1) - S(t)$$

$$\begin{cases} S(t+1) = S(t) - \sigma \cdot S(t) \cdot I(t) \\ I(t+1) = I(t) + \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\ R(t+1) = R(t) + \rho \cdot I(t) \end{cases} \Rightarrow \begin{cases} \Delta S(t+1) = -\sigma \cdot S(t) \cdot I(t) \\ \Delta I(t+1) = \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\ \Delta R(t+1) = \rho \cdot I(t) \end{cases}$$

$$\Delta I(t+1) = \sigma S(t) I(t) - \rho I(t) \Rightarrow \Delta I(t+1) = I(t)\left[\frac{\sigma}{\rho} S(t) - 1\right]\rho = I(t)[R_0 \cdot S(t) - 1]\rho$$

Basic reproduction ratio $R_0 = \sigma/\rho$

Expected number of new infections from a single infection in a population where all subjects are susceptible

Given $I(t) > 0$:

- $\Delta I(t+1) > 0 \quad \text{IF} \quad R_0 > \frac{1}{S(t)}$
- $\Delta I(t+1) < 0 \quad \text{IF} \quad R_0 < \frac{1}{S(t)}$

$\Delta I(t) = 0$ if:

- $R_0 = \frac{1}{S(t)}$
- $I(t) = 0$

Example 2: Simulation

In order to analyze the phenomena, we can simulate the model

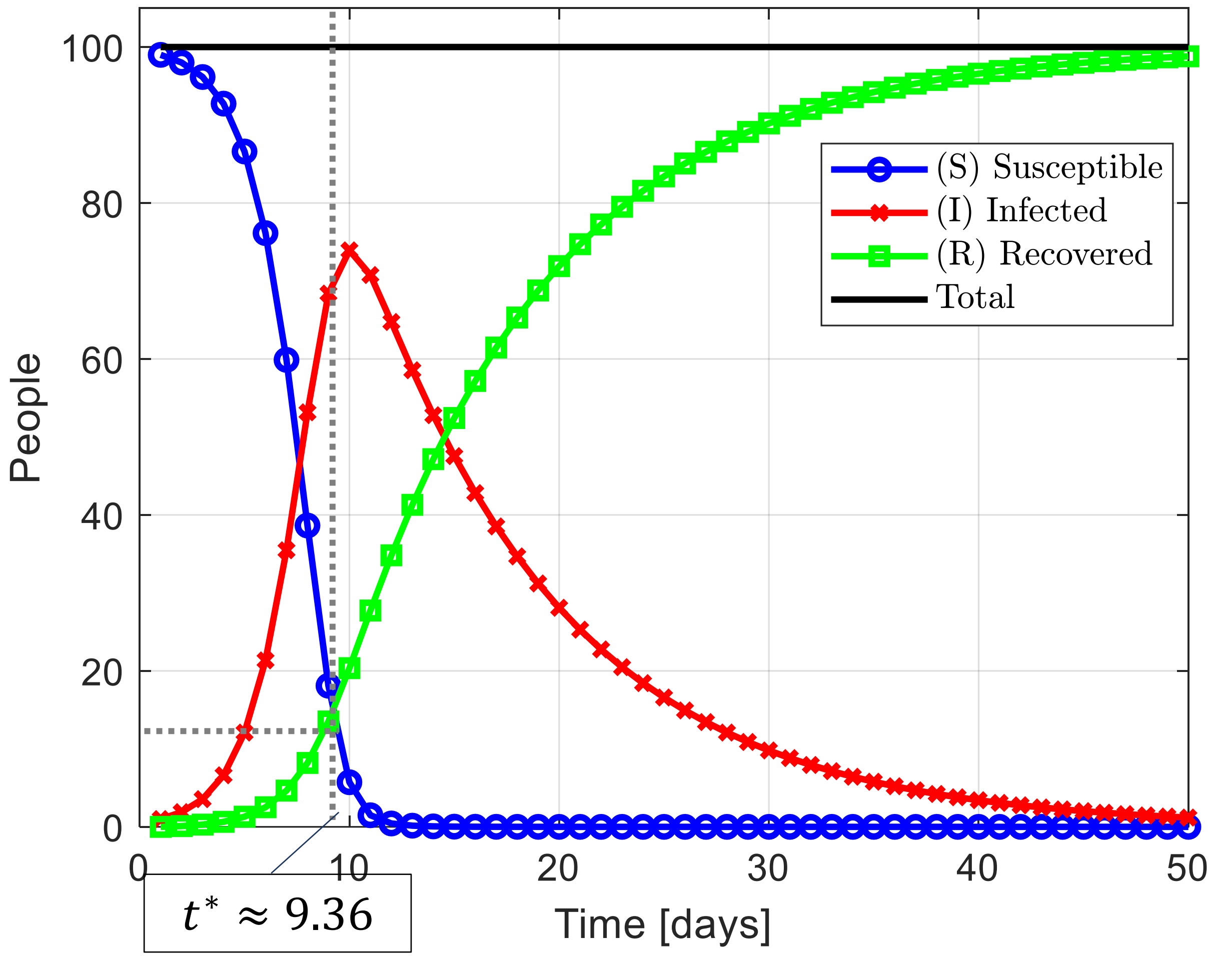
```
sigma = 0.01;
rho = 0.1;
T = 50;

S(1) = 99.9;
I(1) = 0.1;
R(1) = 0;

for t = 2 : 1 : T
    S(t) = S(t-1) - sigma * S(t-1)*I(t-1);
    I(t) = I(t-1) + sigma * S(t-1)*I(t-1) - rho * I(t-1);
    R(t) = R(t-1) + rho * I(t-1);
end
```

Example 1: Simulation

Population size: $100, \sigma = 0.01, \rho = 0.1$



$$R_0 = \frac{\sigma}{\rho} = 0.1 \qquad \frac{1}{S(t^*)} = R_0$$

$\underline{t < t^*}$

$$R_0 > \frac{1}{S(t)} \rightarrow \Delta I(t+1) > 0$$

Increase in the infected

$\underline{t > t^*}$

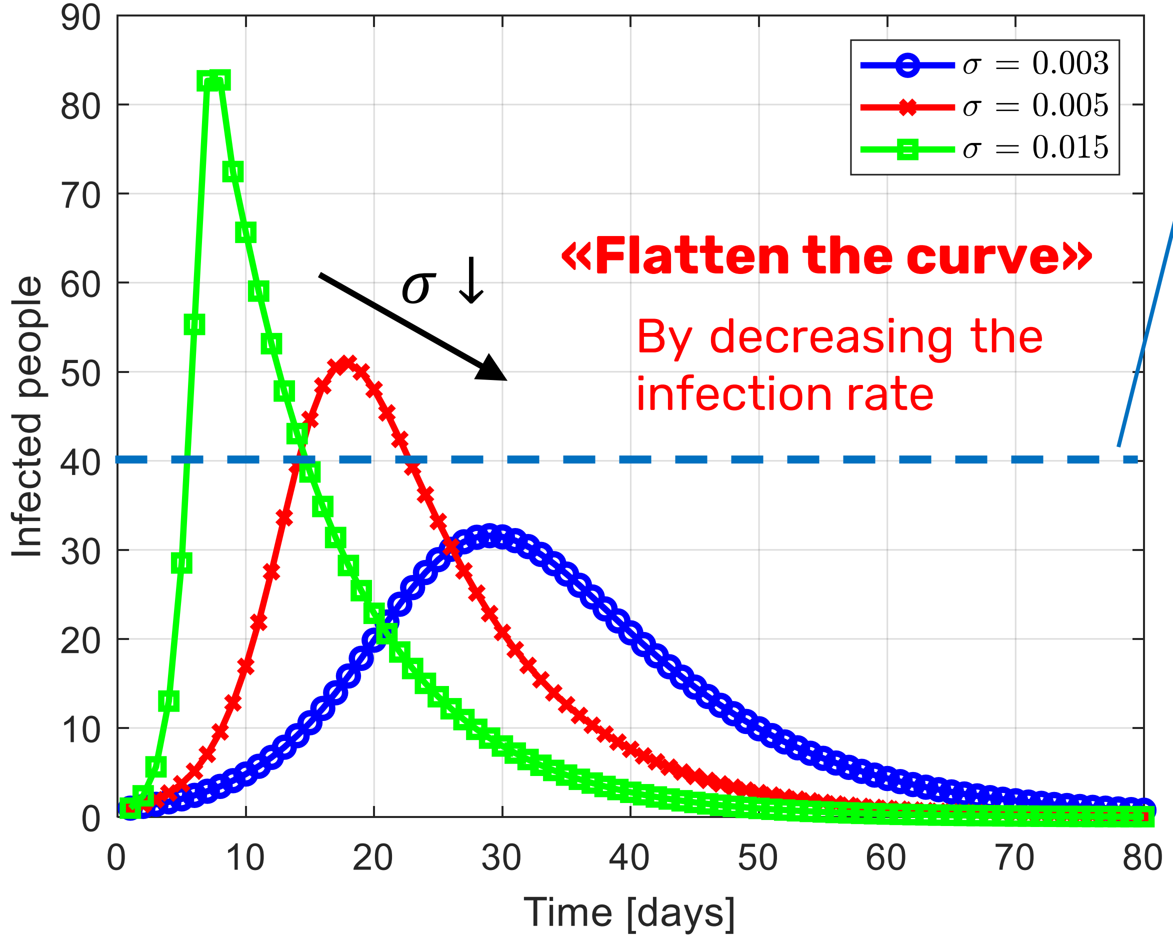
$$R_0 < \frac{1}{S(t)} \rightarrow \Delta I(t+1) < 0$$

Decrease in the infected

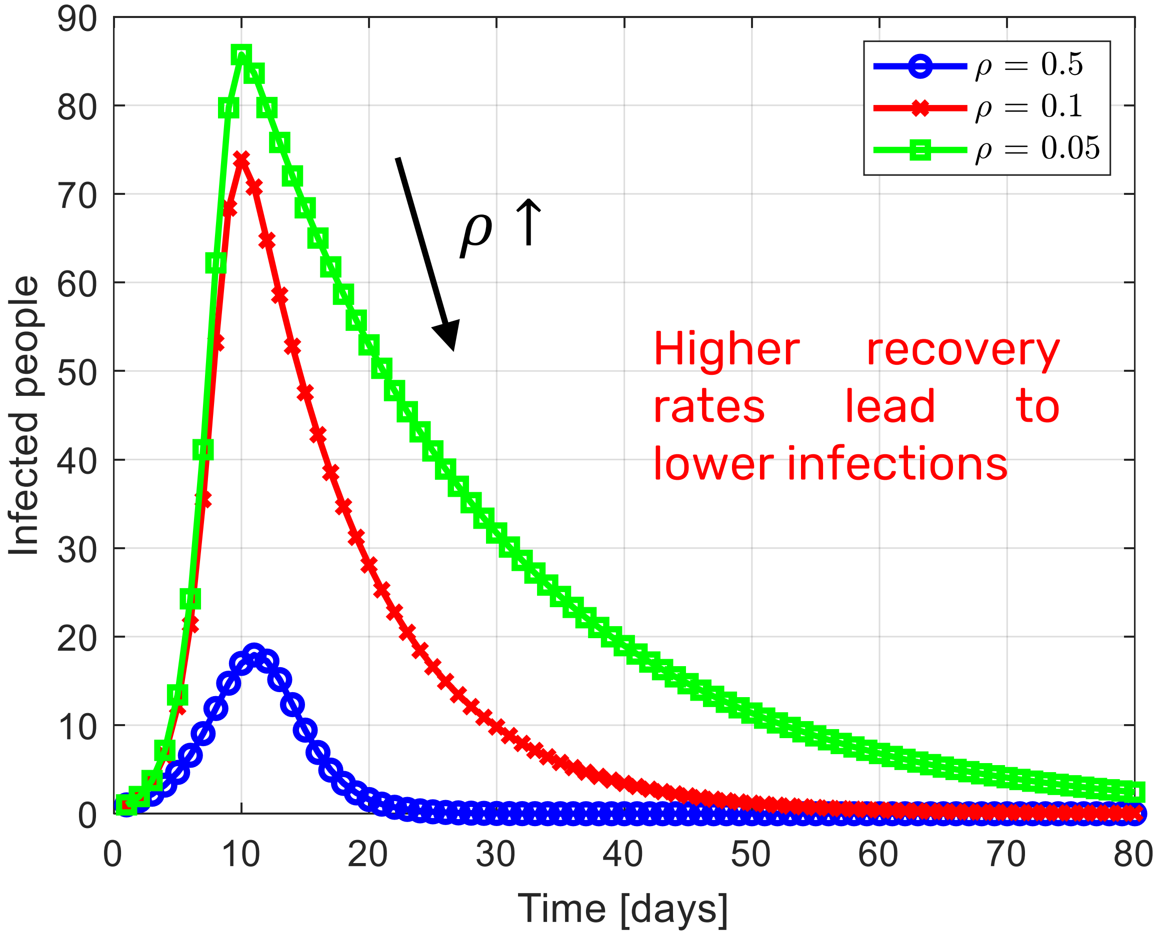
Example 1: Simulation

This could be denote the maximum capacity of the sanitary system

Population size: 100, Different σ, $\rho = 0.1$



Population size: 100, Different ρ, $\sigma = 0.01$



Example 2: Assumptions

The previous model presents the following assumptions:

- The number of persons is a **real number**
- The death/birth rate **is slower** than the infectious disease
- **Everyone can recover (or die)** from the disease

This model can be used to model the seasonal flu outbreak

It is possible to augment the model with **exogenous variables (Inputs)** that vary independently from the model dynamics.

The **inputs will affect** the system behaviour and are usually **known signals**

Example 3: SIR with vaccination

Let's define as $V(t)$ the percentage of **susceptible people that get vaccinated** at the day t.

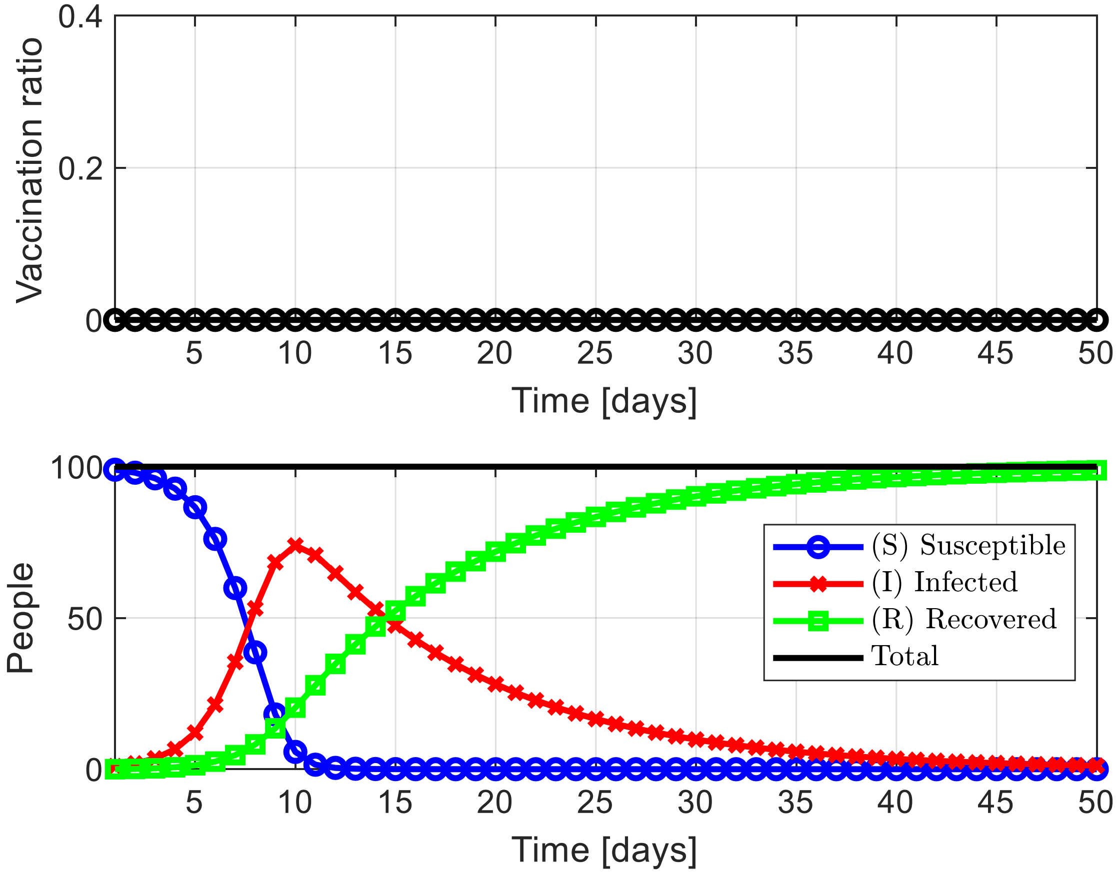
$$\begin{cases} S(t+1) = S(t) - \sigma \cdot S(t) \cdot I(t) - V(t) \cdot S(t) \\ I(t+1) \ = I(t) + \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\ R(t+1) = R(t) + \rho \cdot I(t) + V(t) \cdot S(t) \end{cases}$$

- σ: disease spreading rate, $\in [0,1]$
- ρ: recovery rate, $\in [0,1]$

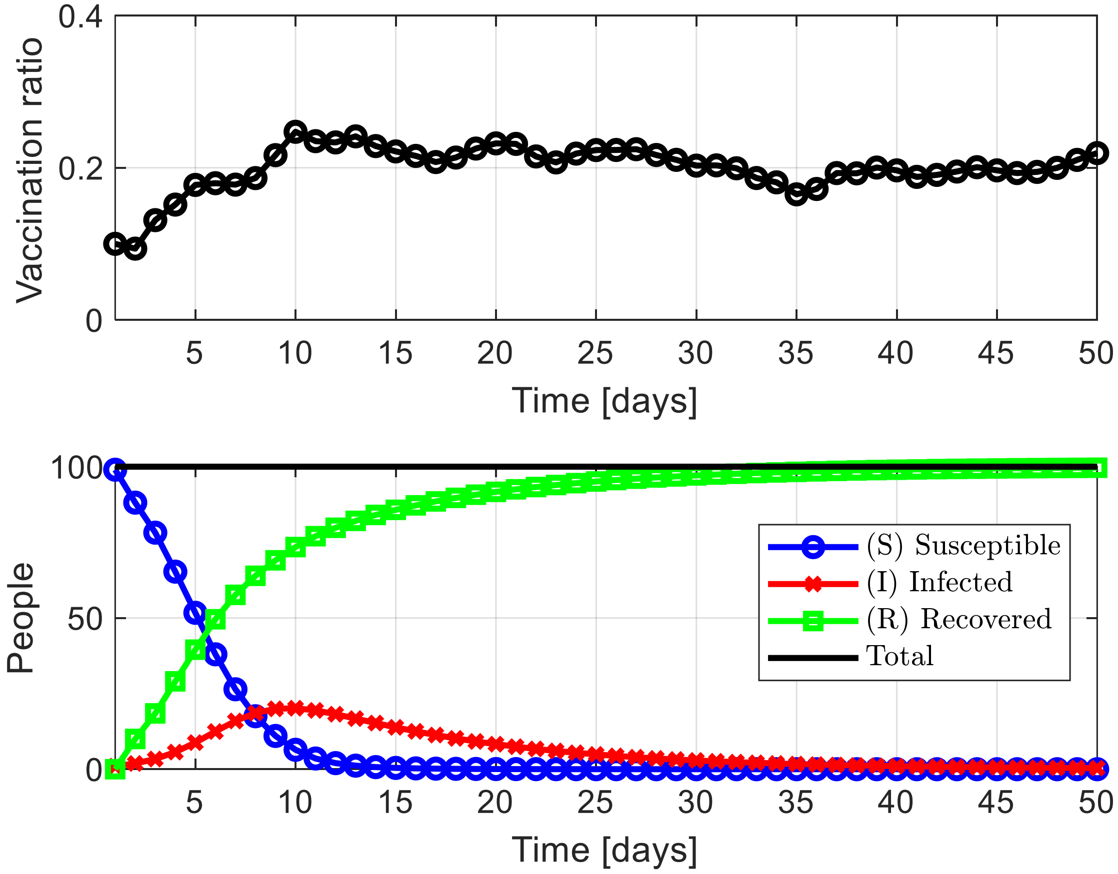
- $S(t)$: number of people that are susceptible to the disease
- $I(t)$: number of infected people
- $R(t)$: number of people that recover for the disease and are not susceptible anymore
- $V(t)$: is an **arbitrary signal** that perturbs the behavior of the model

Example 3: Simulation

Population size: $100, \sigma = 0.01, \rho = 0.1$
No vaccination input



Population size: $100, \sigma = 0.01, \rho = 0.1$
With vaccination input



Latent variables (state variables)

Usually, the system variables are not directly measurable. Thus, we can add an **output equation**, which specifies what we can actually measure.

Suppose that some authority **communicates the number** of infected people $Y(t)$

$$
\begin{cases}
S(t+1) = S(t) - \sigma \cdot S(t) \cdot I(t) - V(t) \cdot S(t) \\
I(t+1) = I(t) + \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\
R(t+1) = R(t) + \rho \cdot I(t) + V(t) \cdot S(t) \\
Y(t) = \alpha I(t) - \beta\big(S(t) + R(t)\big)
\end{cases}
$$

The output model is a static equation

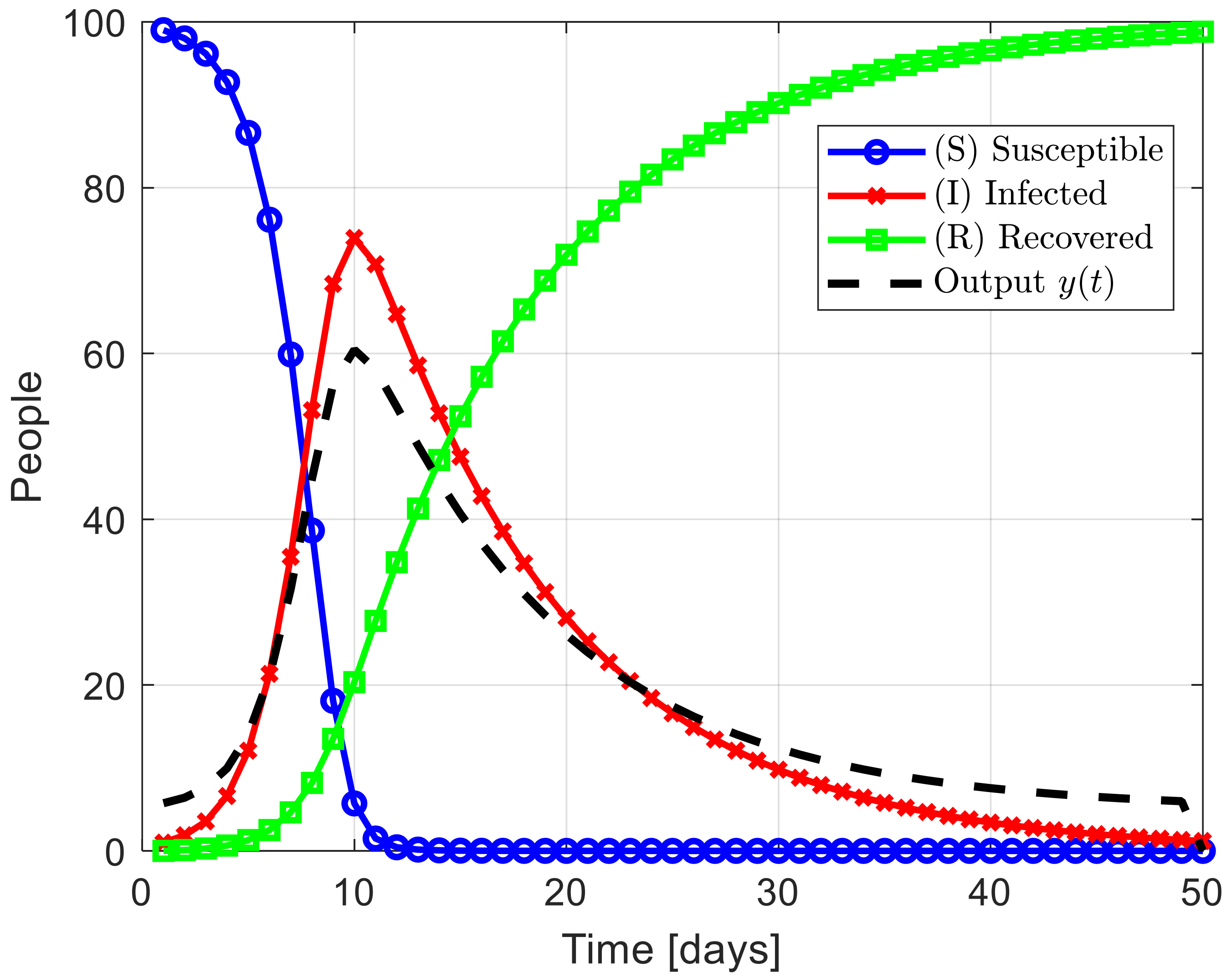
- α: rate of reported cases, $\in [0,1]$
- β: rate of diagnosis errors, $\in [0,1]$

The output at time t can be computed using only information up to time t, and depends on the **state variables** $S(t), I(t), R(t)$ (the model dynamics)

Example 4: Simulation

- **Population size:** $\mathbf{100}$
- $\boldsymbol{\sigma = 0.01, \rho = 0.1, \alpha = 0.8, \beta = 0.05}$
- **No vaccination input**

The output mimic the infected latent state variable $I(t)$, up to a certain **measurement error**



General representation

In general, a discrete-time dynamical system can be written as:

$$\begin{cases} \boldsymbol{x}(t+1) = \boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \\ \boldsymbol{y}(t) = \boldsymbol{g}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) \end{cases}$$

- $\boldsymbol{x}(t) \in \mathbb{R}^{n \times 1}$ are the **states** or latent variables
- $\boldsymbol{y}(t) \in \mathbb{R}^{p \times 1}$ are the **outputs** or measurements
- $\boldsymbol{u}(t) \in \mathbb{R}^{m_u \times 1}$ are the **inputs** or exogenous variables
- $\boldsymbol{f}(\cdot,\cdot) \in \mathbb{R}^{n \times 1}$ is the **process function**
- $\boldsymbol{g}(\cdot,\cdot) \in \mathbb{R}^{p \times 1}$ is the **output function**

SIR dynamical system example

- $\boldsymbol{x}(t) = \begin{bmatrix} S(t) \\ I(t) \\ R(t) \end{bmatrix}$
- $\boldsymbol{u}(t) = V(t)$
- $\boldsymbol{f}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) = \begin{bmatrix} S(t) - \sigma \cdot S(t) \cdot I(t) - V(t) \cdot S(t) \\ I(t) + \sigma \cdot S(t) \cdot I(t) - \rho \cdot I(t) \\ R(t) + \rho \cdot I(t) + V(t) \cdot S(t) \end{bmatrix}$
- $\boldsymbol{y}(t) = Y(t)$
- $\boldsymbol{g}\big(\boldsymbol{x}(t), \boldsymbol{u}(t)\big) = \alpha \cdot I(t) + \beta \cdot \big(S(t) + R(t)\big)$

Linear systems: general representation

When f and g are **linear functions** of the states $\boldsymbol{x}(t)$ and the inputs $\boldsymbol{u}(t)$, we talk of **Linear dynamical systems** *(we consider the discrete-time case)*

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

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix} \in \mathbb{R}^{n \times n} \quad B = \begin{bmatrix} b_{1,1} & & b_{1,m_u} \\ \vdots & & \\ b_{n,1} & & b_{n,m_u} \end{bmatrix} \in \mathbb{R}^{n \times m_u}$$

$$C = \begin{bmatrix} c_{1,1} & \cdots & c_n \\ & & \\ c_{p,1} & & c_{p,p} \end{bmatrix} \in \mathbb{R}^{p \times n} \quad D = \begin{bmatrix} d_{1,1} & \cdots & d_{1,m_u} \\ & & \\ d_{p,1} & & d_{p,m_u} \end{bmatrix} \in \mathbb{R}^{p \times m_u}$$

Example (SISO system $p = 1, m_u = 1$)

$$\begin{cases} x_1(t+1) = 0.5 \cdot x_1(t) + x_2(t) + 3 \cdot u(t) \\ x_2(t+1) = 0.1 \cdot x_2(t) \\ y(t) = x_1(t) + 3 \cdot x_2(t) + 5 \cdot u(t) \end{cases}$$

$$A = \begin{bmatrix} 0.5 & 1 \\ 0 & 0.1 \end{bmatrix} \in \mathbb{R}^{2 \times 2} \quad B = \begin{bmatrix} 3 \\ 0 \end{bmatrix} \in \mathbb{R}^{2 \times 1}$$

$$C = \begin{bmatrix} 1 & 3 \end{bmatrix} \in \mathbb{R}^{1 \times 2} \quad D = 5 \in \mathbb{R}$$

Z-transform

Given a discrete signal $s(t)\colon \mathbb{Z}^+ \to \mathbb{R}$ the Z –transform is the polynomial:

$$\boxed{\mathcal{Z}[s](z) = S(z) \equiv \sum_{t=0}^{\infty} s(t) \cdot z^{-t} = s(0) + s(1)z^{-1} + \cdots}$$

- $z \in \mathbb{C}$ is a complex number

If the series converges (for inputs s.t. $s(t) = 0, \forall t < 0$, this is always true), then $S(z)$ can be espressed as

$$S(z) = \frac{N(z)}{D(z)}$$

where $N(z)$ and $D(z)$ are finite degree polynomials

Delay

Given a signal $s(t)$, then the signal $w(t) = s(t-1)$ has Z-transform:

$$\mathcal{Z}[w](z) = W(z) = \sum_{t=0}^{\infty} w(t) \cdot z^{-t} = w(0) + w(1)z^{-1} + w(2)z^{-2} + \cdots$$

$$= s(-1) + s(0)z^{-1} + s(1)z^{-2} + \cdots$$

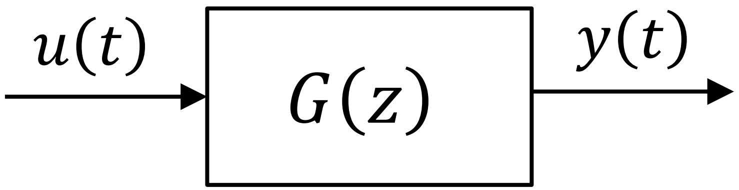
$$= 0 + z^{-1} \cdot (s(0) + s(1)z^{-1} + \cdots)$$

$$= z^{-1} \cdot \sum_{t=0}^{\infty} s(t) \cdot z^{-t} \boxed{= \; z^{-1} \cdot S(z)}$$

We can interpret z^{-1} has a **delay operator** (and z as a **forward operator**)

Linear systems: transfer function

Consider a **SISO** system. The **transfer function** $G(z)$ describes the relation between the **input and output** of a LTI (linear time invariant) dynamical system, when $\boldsymbol{x}(0) = 0$



In a SISO system, it is possible to express $G(z)$ as the **ratio** between the Z-transform of the input signal and the Z-transform of the output signal

$$G(z) = \frac{\mathcal{Z}[y](z)}{\mathcal{Z}[u](z)} = \frac{Y(z)}{U(z)}$$

$\Rightarrow$ Thus, $G(z)$ will be the ratio of **two rational polynomials**

Linear systems: transfer function

In general, if we apply the Z-transform to the state variables, to the inputs and to the outputs, we obtain that $\boldsymbol{G}(z)$ can be expressed as:

$$\boxed{\boldsymbol{G}(z) = C(zI_n - A)^{-1}B + D}$$

(discrete-time case)

$$\boldsymbol{u}(t) \longrightarrow \boxed{\boldsymbol{G}(z)} \longrightarrow \boldsymbol{y}(t)$$

The transfer function can be viewed as an object that **filters** the input $\boldsymbol{u}(t)$ to obtain the output $\boldsymbol{y}(t)$

The filtering behaviour of $\boldsymbol{G}(z)$ can be viewed in the **frequency domain**

The transfer function $\boldsymbol{G}(z)$ depends **only on the system** and not on the **input signal**

Linear systems: transfer function

$$A = \begin{bmatrix} 0.1 & 0.4 \\ 0.3 & 0.2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 3 & 1 \end{bmatrix} \qquad D = 0$$

$$G(z) = C(zI_n - A)^{-1}B + D$$

$$= \begin{bmatrix} 3 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} z & 0 \\ 0 & z \end{bmatrix} - \begin{bmatrix} 0.1 & 0.4 \\ 0.3 & 0.2 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 0 =$$

$$= \begin{bmatrix} 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} z - 0.1 & -0.4 \\ -0.3 & z - 0.2 \end{bmatrix}^{-1} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} =$$

$$\boxed{\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{a \cdot d - b \cdot c} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}$$

$$= \begin{bmatrix} 3 & 1 \end{bmatrix} \cdot \left(\frac{1}{z^2 - 0.3z - 0.1} \begin{bmatrix} z - 0.2 & 0.4 \\ 0.3 & z - 0.1 \end{bmatrix} \right) \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} =$$

Linear systems: transfer function

$$A = \begin{bmatrix} 0.1 & 0.4 \\ 0.3 & 0.2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad C = \begin{bmatrix} 3 & 1 \end{bmatrix} \qquad D = 0$$

$$G(z) = \frac{1}{z^2 - 0.3z - 0.1} \begin{bmatrix} 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} z - 0.2 & 0.4 \\ 0.3 & z - 0.1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} =$$

$$= \frac{1}{z^2 - 0.3z - 0.1} \begin{bmatrix} 3 & 1 \end{bmatrix} \cdot \begin{bmatrix} z - 0.2 \\ 0.3 \end{bmatrix} \Rightarrow$$

$$\boxed{G(z) = \frac{3z - 0.3}{z^2 - 0.3z - 0.1}}$$

The denominator is the **characteristic polynomial** of the matrix A

Linear systems: transfer function

Transfer function form

$$\boxed{G(z) = \frac{3z - 0.3}{z^2 - 0.3z - 0.1}}$$

With *a little abuse of notation*, we can write:

$$y(t) = G(z)u(t) = \frac{3z - 0.3}{z^2 - 0.3z - 0.1}u(t) \Rightarrow y(t) = \frac{3z^{-1} - 0.3z^{-2}}{1 - 0.3z^{-1} - 0.1z^{-2}}u(t)$$

$$\boxed{y(t) = 0.3y(t-1) + 0.1y(t-2) + 3u(t-1) - 0.3u(t-2)}$$

Recursive equation (filter) form

Linear systems representation summary

Summarizing, we can represent a **LTI SISO** discrete-time dynamica systems as:

1) State-space representation

$$\begin{cases} x_1(t+1) = 0.1x_1(t) + 0.4x_2(t) + u(t) \\ x_2(t+1) = 0.3x_2(t) + 0.2x_2(t) \\ y(t) = 3x_1(t) + x_2(t) \end{cases}$$

$C(zI_n - A)^{-1}B + D$

realization

2) Transfer function representation

$$G(z) = \frac{3z^{-1} - 0.3z^{-2}}{1 - 0.3z^{-1} - 0.1z^{-2}}$$

3) Recursive filter representation

$$y(t) = 0.3y(t-1) + 0.1y(t-2) + 3u(t-1) - 0.3u(t-2)$$

The state-space is the most complete representation. The transfer function form represents only the states that are reachable\observable from input\output signals, respectively

Linear systems: zeros and poles

The **transfer function polynomials** describe the properties of the dynamical system

- **Zeros:** roots of the numerator
- **Poles:** roots of the denominator (eigenvalues of the A **matrix)**

$$G(z) = \frac{3z - 0.3}{z^2 - 0.3z - 0.1}$$

A discrete-time LTI dynamical system is said to be **asymptotically stable** iff its **poles are in modulus less than 1**

$z^2 - 0.3z - 0.1 \rightarrow \text{Poles: } \; z_1 = 0.5; \;\; z_2 = -0.2$ $\qquad |z_1| < 1 \;\&\&\; |z_2| < 1 \rightarrow$ **asymptotically stable system**

- Asymptotic stability implies that the output of the system has a «bounded energy», given a «bounded energy» input
- If a system is in a stable equilibrium state, it will return to it after a perturbation

Frequency response

Consider a sampled sine wave with the sampling period T_s. The sampled values are:

$$s(t) = A \cdot \sin(2\pi f_0 \cdot t \cdot T_s + \varphi)$$

Amplitude Frequency Phase

With sampling period T_s, the **Nyquist frequency** is: $f_{Nyq} = \frac{f_s}{2} = \frac{1}{2 \cdot T_s}$

In order to sample correctly, we need to use a sufficiently high sampling frequency f_s

The sine frequency has to respect the **Nyquist criteria (sampling theorem)**

$$f_0 \leq f_{Nyq} = \frac{f_s}{2}$$

Frequency response theorem

Consider an asymptotically stable LTI SISO system with transfer function $G(z)$.

$$u(t) \rightarrow \boxed{G(z)} \rightarrow y(t)$$

Consider an input singal $u(t)$ such that: $u(t) = A \cdot \sin(2\pi T_s t \cdot f + \varphi)$

The output signal is: $y(t) = \tilde{y}(t) + \bar{A} \cdot \sin(2\pi T_s t \cdot f + \bar{\varphi})$

Transient effect

$$\boxed{\lim_{t \to \infty} \tilde{y}(t) = 0}$$

System Gain effect

$$\boxed{\bar{A} = A \cdot \left| G\left(e^{j \cdot 2\pi f T_s}\right) \right|}$$

System phase shift

$$\boxed{\bar{\varphi} = \varphi + \angle G\left(e^{j \cdot 2\pi f T_s}\right)}$$

Frequency response theorem

Defining the FRF (Frequency Response Function) of the SISO LTI system as:

$$\boxed{H_{T_S}(f) = G\left(e^{j\cdot 2\pi T_S\cdot f}\right)}$$

We can write the output $y(t)$ given a sine input $u(t) = A \cdot \sin(2\pi T_s t \cdot f + \varphi)$ as :

$$y(t) = A \cdot \left|H_{T_S}(f)\right| \cdot \sin\left(2\pi f \cdot t + \varphi + \angle H_{T_S}(f)\right)$$

The output of an LTI system to a sine wave input, after the transient, is another **sine wave with the same frequency**, but with phase and gain modified by the system

The FRF depends only on the system and the sampling period/frequency

Bode diagrams

The bode diagrams are composed by two graphs:

- The **magnitude graph**, that plots the **magnitude** of the system frequency response

$$\left|H_{T_s}(f)\right|$$

- The **phase graph**, that plots the **phase** of the system frequency response

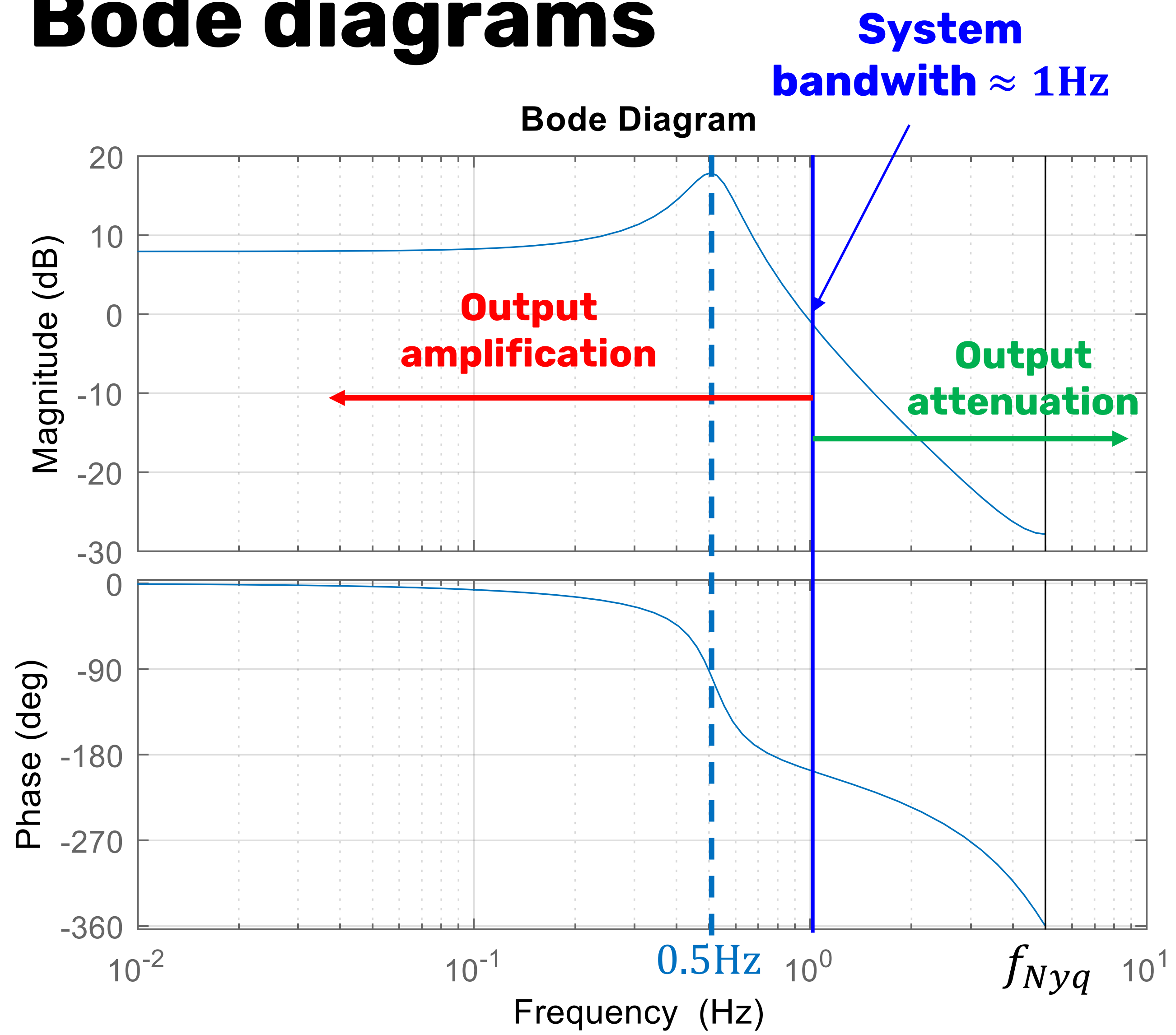
$$\angle H_{T_s}(f)$$

The **frequencies** are plotted in a **logarithmic scale**

The **magnitude** is expressed in **dB (decibels),** i.e. $\mathrm{dB}(f) = 20 \cdot \log\left|H_{T_s}(f)\right|$

The **phase** is usually expressed in **degrees**, but sometimes the radiants are used instead

Bode diagrams

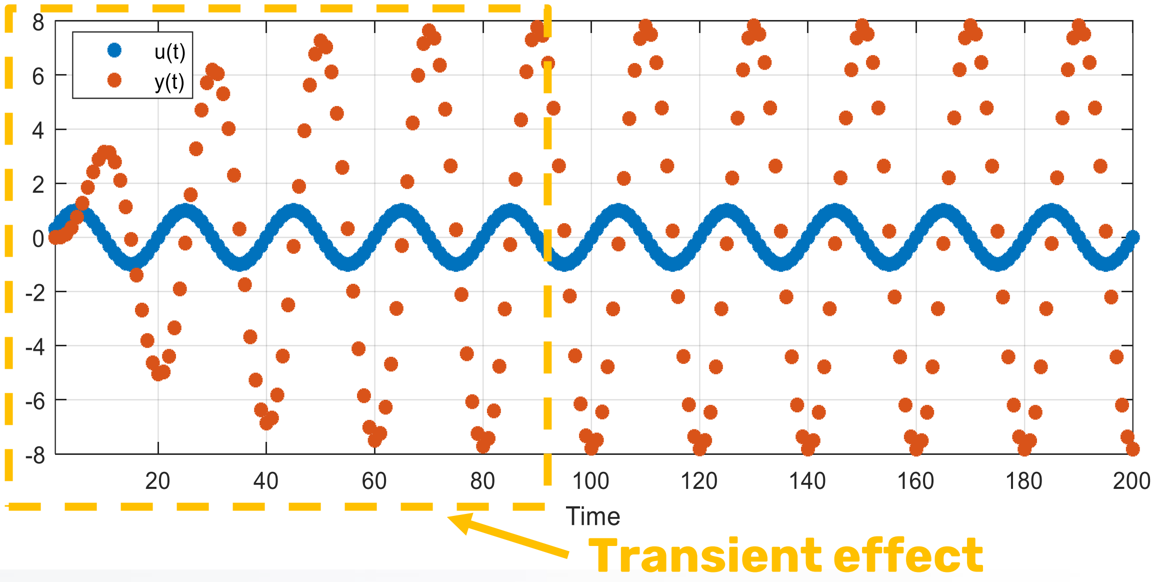


$$G(z) = \frac{0.05z + 0.2}{z^2 - 1.8z + 0.9}$$

$$T_s = 0.1s \Rightarrow f_{Nyq} = \frac{1}{2T_s} = \frac{1}{2 \cdot 0.1} = 5$$

Compute the output of the system with input (sine wave with frequency 0.5Hz):

$$u(t) = \sin(2\pi T_s t \cdot 0.5)$$



Linear systems: continuous time

Physical systems are naturally described in **continuous time**, i.e. $t \in \mathbb{R}_{>0}$, as opposite to discrete systems where $t \in \mathbb{N}_{>0}$. In this case, a linear dynamical model reads as:

$$\begin{cases} \dot{\boldsymbol{x}}(t) = A \cdot \boldsymbol{x}(t) + B \cdot \boldsymbol{u}(t) \\ \boldsymbol{y}(t) = C \cdot \boldsymbol{x}(t) + D \cdot \boldsymbol{u}(t) \end{cases} \xrightarrow[\mathcal{L}[\ \ ](s)]{\textbf{Laplace transform}} \begin{cases} s\boldsymbol{X}(s) = A \cdot \boldsymbol{X}(s) + B \cdot \boldsymbol{U}(s) \\ \boldsymbol{Y}(s) = C \cdot \boldsymbol{X}(s) + D \cdot \boldsymbol{U}(s) \end{cases}$$

The transfer function $\boldsymbol{G}(s)$ can be computer by resorting to the **Laplace transformation**, where $s \in \mathbb{C}$ is the Laplace variable. We then have that

(continuous-time case)

$$\boxed{\boldsymbol{G}(s) = C(sI_n - A)^{-1}B + D}$$

- The system $\boldsymbol{G}(s)$ is asymptotically stable iif the poles are < 0

How to get models of dynamical systems

There are several ways to define a model for a physical system:

1. **White-box models:** derive a continuous-time model from the physics of the system, by combining differential equation (usually conservation laws). Then, **discretize** the model with the sampling frequency f_s of your measured signals (Matlab `c2d`)
2. **Gray-box models:** derive the model structure (number of poles\zeros) from physical laws, but estimate its parameters (ex. The transfer fuction polynomial coefficients) from data
3. **Black-box models:** estimate both the model structure and the model parameters from data

IMAD (6 cfu) 1° year Master degree Computer Engineering
Dynamic systems identification (9 cfu) 1° year EMH

Outline

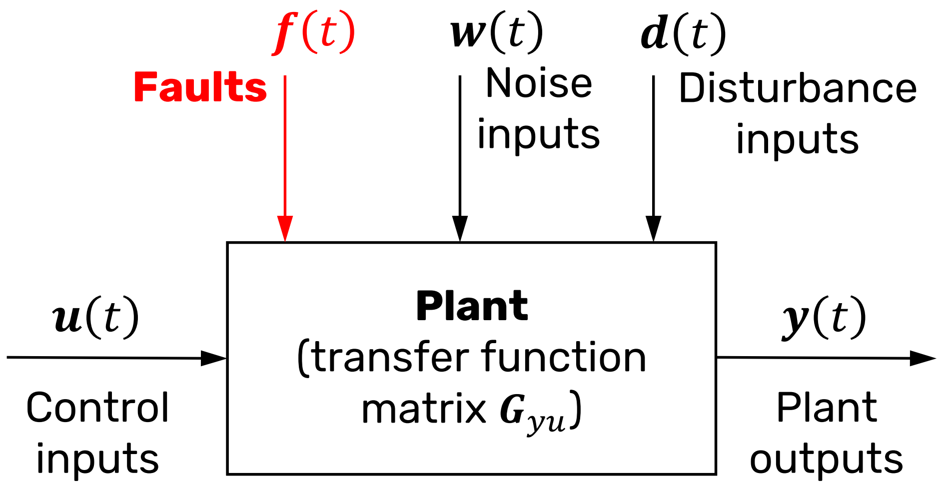
1. Schematic of the approach
2. Dynamical systems
3. **Parity space approach**
4. Diagnostic observer
5. Application to EMA fault detection

Modeling systems with faults

We consider faults that can be modeled as **additive signals**

- Many faults (on **actuators** and **sensors**) can be modeled in this way
- Process faults, usually modeled as multiplicative faults, can be **restated as additive**

$$\begin{cases} \underset{n\times 1}{\boldsymbol{x}(t+1)} = \underset{n\times n}{A} \cdot \boldsymbol{x}(t) + \underset{n\times m_u}{B} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{n\times m_d}{B_d} \cdot \underset{m_d\times 1}{\boldsymbol{d}(t)} + \underset{n\times m_f}{B_f} \cdot \underset{m_f\times 1}{\boldsymbol{f}(t)} \\ \underset{p\times 1}{\boldsymbol{y}(t)} = \underset{p\times n}{C} \cdot \boldsymbol{x}(t) + \underset{p\times m_u}{D} \cdot \boldsymbol{u}(t) + \underset{p\times m_d}{D_d} \cdot \boldsymbol{d}(t) + \underset{p\times m_f}{D_f} \cdot \boldsymbol{f}(t) \end{cases}$$



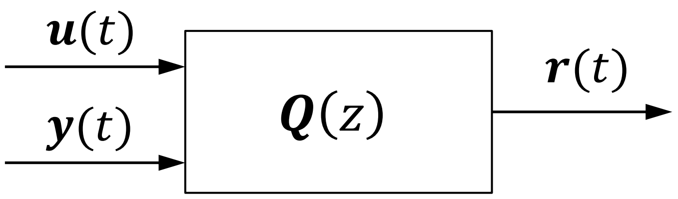
- $\boldsymbol{d}(t) \in \mathbb{R}^{m_d\times 1}$: additive unknown disturbances
- $\boldsymbol{f}(t) \in \mathbb{R}^{m_f\times 1}$: additive fault signals
- We assume no noise signals $\boldsymbol{w}(t)$ are present

The parity space framework

In the parity space FDI (Fault Detection and Isolation) framework the **dynamics** of the **residual signals** are presented in the form of **algebraic equations**

Thus, most the problem solutions are achieved with **linear algebra tools**

- the system designer is not required to have rich knowledge of advanced control theory
- most computations can be completed without complex and involved algorithms
- a great number of FDI methods and ideas have been first presented in the parity space framework, and later extended to other frameworks



Parity space idea

Suppose that $\boldsymbol{w}(t) = \mathbf{0}$, $\boldsymbol{d}(t) = \mathbf{0}$ and $\boldsymbol{f}(t) = \mathbf{0}$, i.e. the system is **not subject** to external noise, disturbances and faults. Then, with $s \geq 0$, the following relations hold:

- $\boldsymbol{y}(t-s) = C\boldsymbol{x}(t-s) + D\boldsymbol{u}(t-s)$

- $\boldsymbol{y}(t-s+1) = C\boldsymbol{x}(t-s+1) + D\boldsymbol{u}(t-s+1)$

$$= C \cdot \{A\boldsymbol{x}(t-s) + B\boldsymbol{u}(t-s)\} + D\boldsymbol{u}(t-s+1)$$

$$= CA^{1}\boldsymbol{x}(t-s) + CB\boldsymbol{u}(t-s) + D\boldsymbol{u}(t-s+1)$$

Parity space idea

- $\boldsymbol{y}(t-s+1) = C\boldsymbol{x}(t-s+1) + D\boldsymbol{u}(t-s+1)$

$$= C \cdot \{A\boldsymbol{x}(t-s) + B\boldsymbol{u}(t-s)\} + D\boldsymbol{u}(t-s+1)$$

$$= CA^{1}\boldsymbol{x}(t-s) + CB\boldsymbol{u}(t-s) + D\boldsymbol{u}(t-s+1)$$

- $\boldsymbol{y}(t-s+2) = C\boldsymbol{x}(t-s+2) + D\boldsymbol{u}(t-s+2)$

$$= C \cdot \{A\boldsymbol{x}(t-s+1) + B\boldsymbol{u}(t-s+1)\} + D\boldsymbol{u}(t-s+2)$$

$$= CA \cdot \{A\boldsymbol{x}(t-s) + B\boldsymbol{u}(t-s)\} + CB\boldsymbol{u}(t-s+1) + D\boldsymbol{u}(t-s+2)$$

$$= CA^{2}\boldsymbol{x}(t-s) + CA^{2-1}B\boldsymbol{u}(t-s) + CB\boldsymbol{u}(t-s+1) + D\boldsymbol{u}(t-s+2)$$

$t-s+(2-1)$ $\quad$ $t-s+(2)$

Parity space idea

- $\boldsymbol{y}(t-s+2) = C\boldsymbol{x}(t-s+2) + D\boldsymbol{u}(t-s+2)$

$$= C \cdot \{A\boldsymbol{x}(t-s+1) + B\boldsymbol{u}(t-s+1)\} + D\boldsymbol{u}(t-s+2)$$

$$= CA \cdot \{A\boldsymbol{x}(t-s) + B\boldsymbol{u}(t-s)\} + CB\boldsymbol{u}(t-s+1) + D\boldsymbol{u}(t-s+2)$$

$$= CA^2\boldsymbol{x}(t-s) + CA^{2-1}B\boldsymbol{u}(t-s) + CB\boldsymbol{u}(\underbrace{t-s+1}_{t-s+(2-1)}) + D\boldsymbol{u}(\underbrace{t-s+2}_{t-s+(2)})$$

$\vdots$

- $\boldsymbol{y}(t-s+s) = \boldsymbol{y}(t) = CA^s\boldsymbol{x}(t-s) + CA^{s-1}B\boldsymbol{u}(t-s) + \cdots + CB\boldsymbol{u}(\underbrace{t-1}_{t-s+(s-1)}) + D\boldsymbol{u}(\underbrace{t}_{t-s+(s)})$

UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Parity space idea

Define now the following quantities:

$$\underset{m_u(s+1)\times 1}{\boldsymbol{u}_s(t)} = \begin{bmatrix} \underset{m_u\times 1}{\boldsymbol{u}(t-s)} \\ \boldsymbol{u}(t-s+1) \\ \vdots \\ \boldsymbol{u}(t) \end{bmatrix} \qquad \underset{p(s+1)\times 1}{\boldsymbol{y}_s(t)} = \begin{bmatrix} \underset{p\times 1}{\boldsymbol{y}(t-s)} \\ \boldsymbol{y}(t-s+1) \\ \vdots \\ \boldsymbol{y}(t) \end{bmatrix}$$

$$\underset{p(s+1)\times m_u(s+1)}{H_{u,s}} = \begin{bmatrix} \overset{p\times m_u}{D} & 0 & \cdots & \cdots & 0 \\ CB & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ CA^{s-1}B & \cdots & \cdots & CB & D \end{bmatrix} \qquad \underset{p(s+1)\times n}{H_{o,s}} = \begin{bmatrix} \underset{p\times n}{C} \\ CA \\ \vdots \\ CA^s \end{bmatrix}$$

Parity space idea

Given the quantities $\boldsymbol{u}_s$, $\boldsymbol{y}_s$, $H_{o,s}$, $H_{u,s}$, we can write

$$\boxed{\boldsymbol{y}_s(t) = H_{o,s}\boldsymbol{x}(t-s) + H_{u,s}\boldsymbol{u}_s(t)}$$

Parity relation

$$\underbrace{\begin{bmatrix} \boldsymbol{y}(t-s) \\ \boldsymbol{y}(t-s+1) \\ \vdots \\ \boldsymbol{y}(t) \end{bmatrix}}_{p(s+1)\times 1} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix}}_{p(s+1)\times n} \cdot \underbrace{\boldsymbol{x}(t-s)}_{n\times 1} + \underbrace{\begin{bmatrix} D & 0 & \cdots & \cdots & 0 \\ CB & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ CA^{s-1}B & \cdots & \cdots & CB & D \end{bmatrix}}_{p(s+1)\times m_u(s+1)} \cdot \underbrace{\begin{bmatrix} \boldsymbol{u}(t-s) \\ \boldsymbol{u}(t-s+1) \\ \vdots \\ \boldsymbol{u}(t) \end{bmatrix}}_{m_u(s+1)\times 1}$$

Parity space idea

$$\boxed{\boldsymbol{y}_s(t) = H_{o,s}\boldsymbol{x}(t-s) + H_{u,s}\boldsymbol{u}_s(t)}$$

Parity relation

- Describes the **inputs and outputs relationship** based on the past state vector $\boldsymbol{x}(t-s)$
- $\boldsymbol{y}_s$ and $\boldsymbol{u}_s$ consist of the temporal and past outputs and inputs, and are **known**
- Matrices $H_{o,s}$ and $H_{u,s}$ are composite of system matrices A, B, C, D and are also **known**
- The only **unknown** variable is $\boldsymbol{x}(t-s)$

Parity space idea

The underlying idea of the parity relation based residual generation lies in the utilization of the that, for $s \geq n$, the following **rank condition** holds:

$$\operatorname{rank}\left(H_{o,s}\right) \leq n < \text{the number of rows of the matrix } H_{o,s}, \text{i.e. } p(s+1)$$

$$\underset{p(s+1)\times n}{H_{o,s}} = \begin{bmatrix} \underset{p\times n}{C} \\ CA \\ \vdots \\ CA^s \end{bmatrix}$$

Therefore, for $s \geq n$, there **exists** a vector $\boldsymbol{v}_s^T \in \mathbb{R}^{1\times p(s+1)}$, $\boldsymbol{v}_s \neq \mathbf{0}$, such that:

$$\boxed{\boldsymbol{v}_s^T \cdot H_{o,s} = \mathbf{0}}$$

The multiplication of a row vector for a matrix results into a linear combination of the matrix rows

that is, it is possible to express a row of $H_{o,s}$ as **linear combination** of other rows of $H_{o,s}$.

The vector $\boldsymbol{v}_s$ can be **found by solving** the above **linear system**. However, $\boldsymbol{v}_s$ is not guaranteed to be unique.

Parity space residual generator

From the observation that $\boldsymbol{v}_s^T H_{o,s} = \mathbf{0}$, a residual signal $r(t)$ is built as:

$$\boxed{r(t) = \underset{1 \times p(s+1)}{\boldsymbol{v}_s^T} \cdot \underset{p(s+1) \times 1}{\left(\boldsymbol{y}_s(t) - H_{u,s}\, \boldsymbol{u}_s(t)\right)}}$$

Residual generator

In the nominal case where $\boldsymbol{d}(t) = \mathbf{0}, \boldsymbol{f}(t) = \mathbf{0}$, we have that:

$$r(t) = \boldsymbol{v}_s^T \cdot \left(H_{o,s}\, \boldsymbol{x}(t-s)\right) = 0$$

In the nominal case, the **residual is zero**

Vectors satisfying $\boldsymbol{v}_s^T \cdot H_{o,s} = \mathbf{0}$ are called **parity vectors**. The set $P_s = \{\boldsymbol{v}_s \mid \boldsymbol{v}_s H_{o,s} = \mathbf{0}\}$ is called **parity space**

Parity space residual generator

$$\underset{p(s+1)\times n}{H_{o,s}} = \begin{bmatrix} \underset{p\times n}{C} \\ CA \\ \vdots \\ CA^s \end{bmatrix}$$

The main observation for building a parity vector was the fact that:

$$\mathrm{rank}\left(H_{o,s}\right) \leq n < \text{the number of rows of the matrix } H_{o,s}, \text{i. e. } p(s+1)$$

The previous statement can be proved by considering the square matrix $A \in \mathbb{R}^{n\times n}$, its associated **characteristic polynomial** (with $\lambda \in \mathbb{C}$)

$$\varphi(\lambda) = \lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \cdots + a_n$$

and **characteristic equation** $\varphi(\lambda) = 0$, whose solutions are the **eigenvalues** of A

From the **Cayley-Hamilton theorem**, it holds that:

$$\varphi(A) = A^n + a_1A^{n-1} + a_2A^{n-2} + \cdots + a_nI_n = 0$$

Parity space residual generator

$$\varphi(A) = A^n + a_1 A^{n-1} + a_2 A^{n-2} + \cdots + a_n I_n = 0$$

$$C \cdot \varphi(A) = CA^n + a_1 CA^{n-1} + a_2 CA^{n-2} + \cdots + a_n C = 0$$

As an example, consider a **2° order LTI SISO** system, i.e. with $A \in \mathbb{R}^{2\times 2}$ and $C \in \mathbb{R}^{1\times 2}$

- $C = [c_1 \quad c_2]$
- $CA = [w_1 \quad w_2]$

$$\underset{3\times 2}{H_{o,s}} = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix},$$

$$C \cdot \varphi(A) = CA^2 + a_1 CA + a_2 C = 0$$

$$CA^2 = -a_1 CA - a_2 C = -a_1 [w_1 \quad w_2] - a_2 [c_1 \quad c_2] = [-a_1 \quad -a_2] \begin{bmatrix} w_1 & w_2 \\ c_1 & c_2 \end{bmatrix}$$

$$= [-a_1 \quad -a_2] \begin{bmatrix} CA \\ C \end{bmatrix}$$

Thus, the rows of CA^2 can be expressed as a linear combination of the rows of C and CA, with $\mathrm{rank}(H_{o,s}) \leq 2$

Parity space: example

Consider the SISO nominal system model

$$Y(z) = G_{yu}(z)U(z) = \frac{b_n z^n + b_{n-1} z^{n-1} + \cdots + b_1 z + b_0}{z^n + a_{n-1} z^{n-1} + \cdots + a_1 z + a_0} U(z)$$

A trivial way to construct a parity space based residual generator is to:

1. rewrite the system into its minimum state space realization form
2. solve the linear system $\boldsymbol{v}_s^T \cdot H_{o,s} = \mathbf{0}$ for $\boldsymbol{v}_s$
3. construct the residual generator $r(t) = \boldsymbol{v}_s^T \cdot \left(\boldsymbol{y}_s(t) - H_{u,s}\, \boldsymbol{u}_s(t)\right)$

Parity space: example

On the other side, it follows from Cayley–Hamilton theorem that:

$$A^n + a_{n-1}A^{n-1} + a_{n-2}A^{n-2} + \cdots + a_0 I_n = \mathbf{0} \quad \Longrightarrow \quad [a_0 \quad \cdots \quad a_{n-1} \quad 1]\begin{bmatrix} C \\ CA \\ \vdots \\ CA^n \end{bmatrix} = \mathbf{0}$$

where A, C denote the system matrices of the minimum state space realization of $G_{yu}(z)$.

That means that:

$$\boldsymbol{v}_s = [a_0 \quad \cdots \quad a_{n-1} \quad 1]$$

is a parity space vector for $G_{yu}(z)$.

Parity space: example

The residual generator can be constructed as:

$$r(t) = \boldsymbol{v}_s^T \cdot \left(\boldsymbol{y}_s(t) - H_{u,s}\,\boldsymbol{u}_s(t)\right) = \boldsymbol{v}_s^T \boldsymbol{y}_s(t) - \boldsymbol{v}_s^T H_{u,s}\,\boldsymbol{u}_s(t) \ = [a_0 \quad \cdots \quad a_{n-1} \quad 1]\boldsymbol{y}_s(t) - \boldsymbol{v}_s^T H_{u,s}\,\boldsymbol{u}_s(t)$$

$$= a_0 y(t-s) + \cdots + a_1 y(t-s+1) + y(t) - \boldsymbol{v}_s^T H_{u,s}\,\boldsymbol{u}_s(t) = 0$$

For the equality $r(t) = 0$ to hold in the nominal case, it follows that

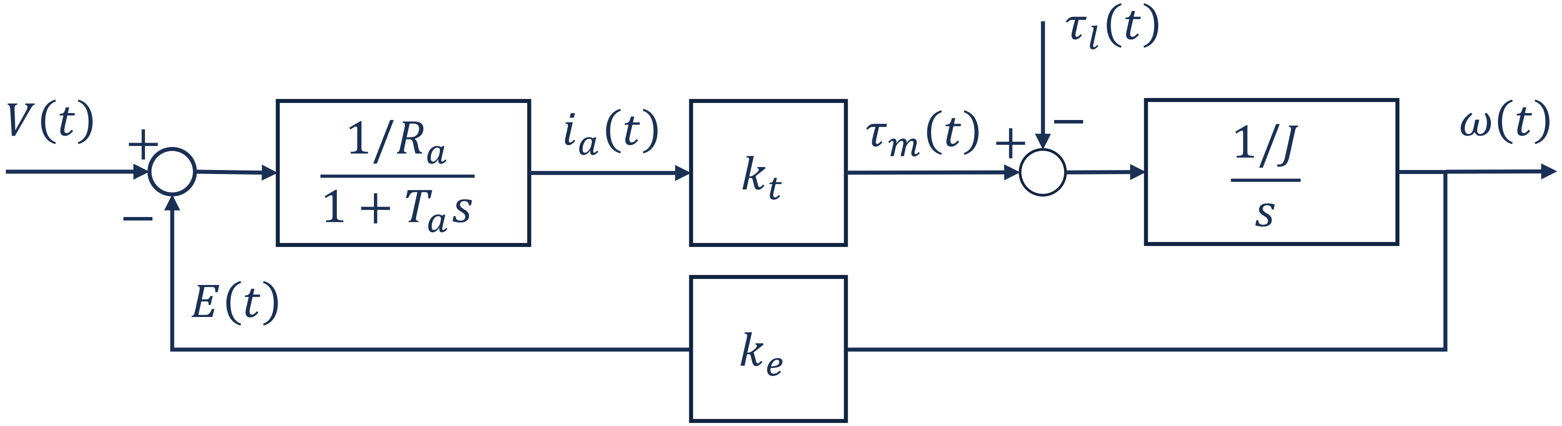
$$\boldsymbol{v}_s^T H_{u,s} = [b_0 \quad \cdots \quad b_{n-1} \quad b_n]$$

As a result, the residual generator, corresponding to the previous choice of $\boldsymbol{v}_s$, is given by

$$\boxed{r(t) = [a_0 \quad \cdots \quad a_{n-1} \quad 1] \cdot \boldsymbol{y}_s\,(t) - [b_0 \quad \cdots \quad b_{n-1} \quad b_n] \cdot \boldsymbol{u}_s\,(t)}$$

Parity space: DC motor example

Consider an open-loop model of a DC motor



- Model input $u(t) = V(t)$
- Model output $y(t) = \omega(t)$
- Disturbance $d(t) = \tau_l(t)$

- Total inertia J: $80.45 \cdot 10^{-6}\ \mathrm{Kg} \cdot \mathrm{m}^2$
- Motor electrical constant k_e: $6.27 \cdot 10^{-3}\ \mathrm{V/rpm}$
- Motor torque constant $k_t = 0.06\ \mathrm{Nm/A}$
- Motor coil inductance $L_a = 0.003\ \mathrm{H}$
- Motor coil resistance $R_a = 3.13\ \Omega$
- Electrical time constant $T_a = L_a/R_a$ s

$$G_{yu}(s) = \frac{1}{K_e \cdot \left(1 + J \cdot \frac{R_a}{K_t K_e} s + J \cdot T_a \cdot \frac{R_a}{K_t K_e} s^2\right)}$$

$$G_{yd}(s) = -\frac{R_a(1 + T_a s)}{K_t K_e \cdot \left(1 + J \cdot \frac{R_a}{K_t K_e} s + J \cdot T_a \cdot \frac{R_a}{K_t K_e} s^2\right)}$$

Parity space: DC motor example

Compute the transfer functions and convert to discrete time (with sampling freq. 100 Hz)

$$G_{yu}(s) = \frac{2.486 \cdot 10^5}{(s + 1042)(s + 1.496)} \xrightarrow{\texttt{c2d}} G_{yu}(z) = \frac{1.184\, z + 1.184}{z^2 - 0.9852\, z + 2.943 \cdot 10^{-5}}$$

$$G_{yd}(s) = \frac{-12430 \cdot (s + 1043)}{(s + 1042)(s + 1.496)} \xrightarrow{\texttt{c2d}} G_{yd}(z) = \frac{-123.6z + 0.003637}{z^2 - 0.9852\, z + 2.943 \cdot 10^{-5}}$$

A **valid parity vector** is:

$$\boldsymbol{v}_s^T = [2.943 \cdot 10^{-5} \quad -0.9852 \quad 1]$$

Parity space: DC motor example

A state space realization of $G_{yu}(z)$ leads to

$$G_{yu}(z) = \frac{1.184\,z + 1.184}{z^2 - 0.9852\,z + 2.943 \cdot 10^{-5}} \xrightarrow{\texttt{minreal(ss(G\_yu))}}$$

$$\underset{\substack{n \times n \\ 2 \times 2}}{A} = \begin{bmatrix} 0.9852 & -0.007535 \\ 0.003906 & 0 \end{bmatrix} \qquad \underset{\substack{n \times m_u \\ 2 \times 1}}{B} = \begin{bmatrix} 16 \\ 0 \end{bmatrix}$$

$$\underset{\substack{p \times n \\ 1 \times 2}}{C} = \begin{bmatrix} 0.07401 & 18.95 \end{bmatrix} \qquad \underset{\substack{p \times m_u \\ 1 \times 1}}{D} = 0$$

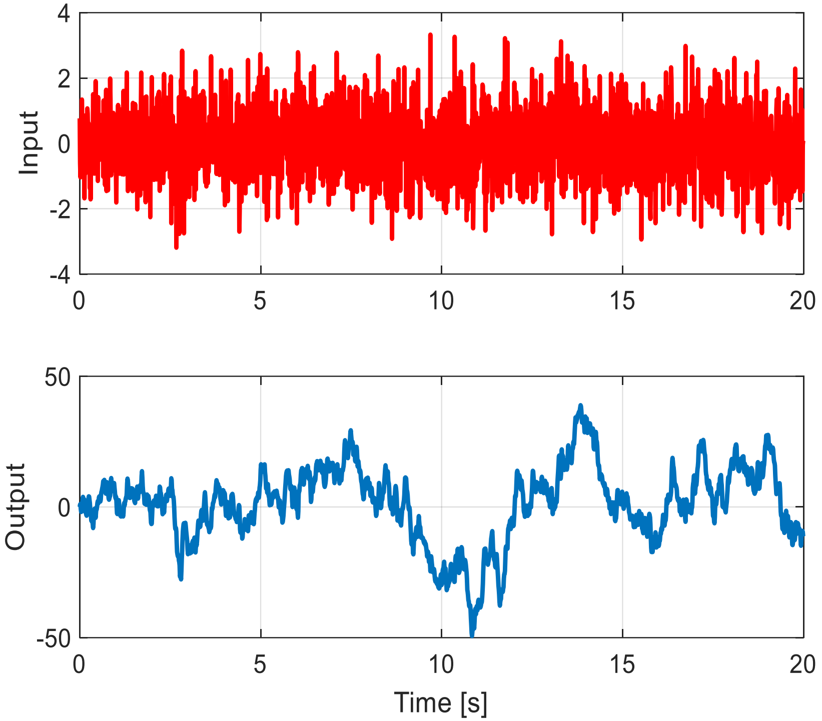
$$\underset{\substack{p(s+1) \times n \\ 1(2+1) \times 2}}{H_{o,s}} = \begin{bmatrix} 0.0740 & 18.9468 \\ 0.1469 & -0.0006 \\ 0.1447 & -0.0011 \end{bmatrix} \qquad \underset{\substack{p(s+1) \times m_u(s+1) \\ 1(2+1) \times 1(2+1)}}{H_{u,s}} = \begin{bmatrix} 0 & 0 & 0 \\ 1.1842 & 0 & 0 \\ 2.3508 & 1.1842 & 0 \end{bmatrix}$$

Always check the solution!

$$\boldsymbol{v}_s^T H_{o,s} = \begin{bmatrix} 2.943 \cdot 10^{-5} & -0.9852 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0.0740 & 18.9468 \\ 0.1469 & -0.0006 \\ 0.1447 & -0.0011 \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix}$$ **CORRECT!** ☺

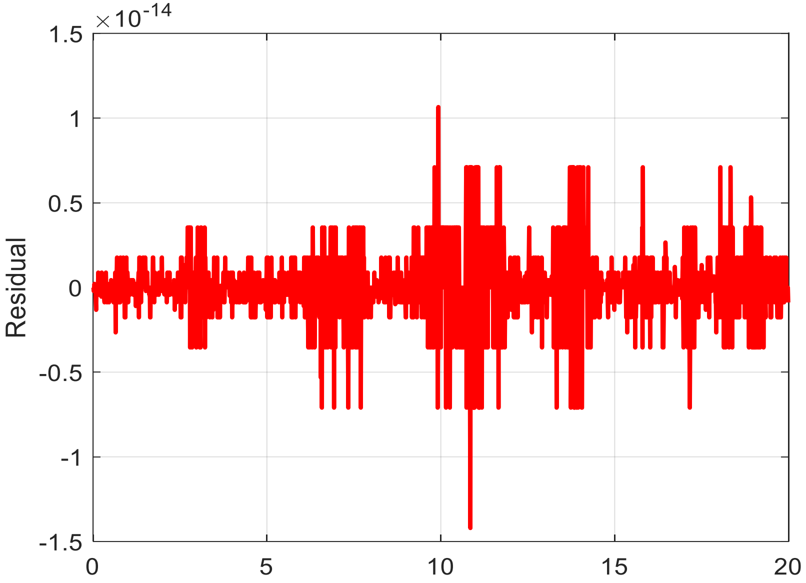
Parity space: DC motor example

1. Simulate data, $u(t) = WN(0,1)$



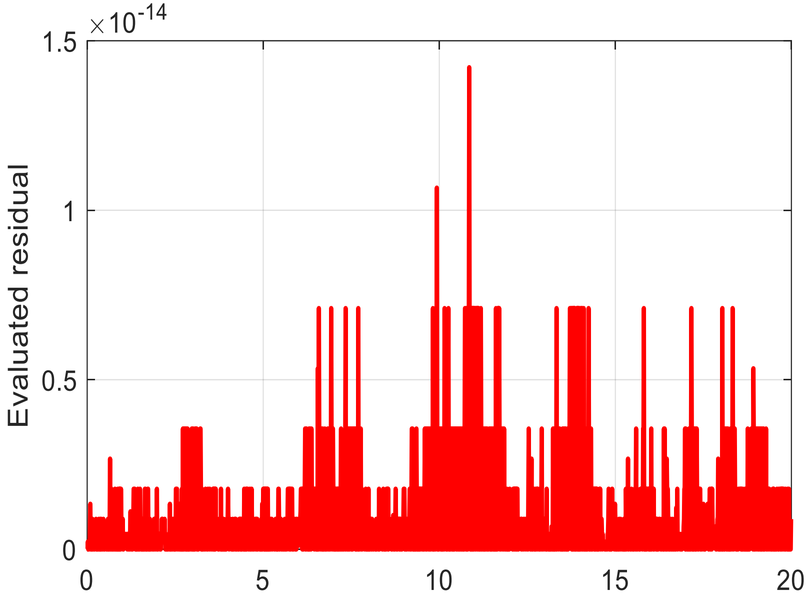
2. Compute the residual

$$r(t) = \boldsymbol{v}_s^T \cdot \left(\boldsymbol{y}_s(t) - H_{u,s}\,\boldsymbol{u}_s(t)\right)$$



3. Evaluate the residual

$$\theta(t) = |r(t)|$$



Parity space: effects of faults and disturbances

Suppose now that disturbances and faults affect the system, i.e. $\boldsymbol{d}(t) \neq 0$ and $\boldsymbol{f}(t) \neq 0$.

By defining the quantities:

$$\underset{m_f(s+1)\times 1}{\boldsymbol{f}_s(t)} = \begin{bmatrix} \underset{m_f \times 1}{\boldsymbol{f}(t-s)} \\ \boldsymbol{f}(t-s+1) \\ \vdots \\ \boldsymbol{f}(t) \end{bmatrix} \qquad \underset{p(s+1)\times m_f(s+1)}{H_{f,s}} = \begin{bmatrix} D_f & 0 & \cdots & \cdots & 0 \\ \underset{p \times m_f}{CB_f} & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ CA^{s-1}B_f & \cdots & \cdots & CB_f & D_f \end{bmatrix}$$

$$\underset{m_d(s+1)\times 1}{\boldsymbol{d}_s(t)} = \begin{bmatrix} \underset{m_d \times 1}{\boldsymbol{d}(t-s)} \\ \boldsymbol{d}(t-s+1) \\ \vdots \\ \boldsymbol{d}(t) \end{bmatrix} \qquad \underset{p(s+1)\times m_d(s+1)}{H_{d,s}} = \begin{bmatrix} D_d & 0 & \cdots & \cdots & 0 \\ \underset{p \times m_d}{CB_d} & \ddots & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ CA^{s-1}B_d & \cdots & \cdots & CB_d & D_d \end{bmatrix}$$

Parity space: effects of faults and disturbances

Given the quantities $\boldsymbol{u}_s$, $\boldsymbol{y}_s$, $\boldsymbol{f}_s$, $\boldsymbol{d}_s$, $H_{o,s}$, $H_{u,s}$, $H_{f,s}$, $H_{d,s}$ we can write

$$\boldsymbol{y}_s(t) = H_{o,s}\boldsymbol{x}(t-s) + H_{u,s}\boldsymbol{u}_s(t) + H_{d,s}\boldsymbol{d}_s(t) + H_{f,s}\boldsymbol{f}_s(t)$$

$$\begin{bmatrix} \boldsymbol{y}(t-s) \\ \boldsymbol{y}(t-s+1) \\ \vdots \\ \boldsymbol{y}(t) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix} \cdot \boldsymbol{x}(t-s) + \begin{bmatrix} D & 0 & \cdots & \cdots & 0 \\ CB & & & & \vdots \\ & & & & \vdots \\ \vdots & & & & 0 \\ CA^{s-1}B & \cdots & \cdots & CB & D \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{u}(t-s) \\ \boldsymbol{u}(t-s+1) \\ \vdots \\ \boldsymbol{u}(t) \end{bmatrix} +$$

$$+ \begin{bmatrix} D_d & 0 & \cdots & \cdots & 0 \\ CB_d & & & & \vdots \\ & & & & \vdots \\ \vdots & & & & 0 \\ CA^{s-1}B_d & \cdots & \cdots & CB_d & D_d \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{d}(t-s) \\ \boldsymbol{d}(t-s+1) \\ \vdots \\ \boldsymbol{d}(t) \end{bmatrix} + \begin{bmatrix} D_f & 0 & \cdots & \cdots & 0 \\ CB_f & & & & \vdots \\ & & & & \vdots \\ \vdots & & & & 0 \\ CA^{s-1}B_f & \cdots & \cdots & CB_f & D_f \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{f}(t-s) \\ \boldsymbol{f}(t-s+1) \\ \vdots \\ \boldsymbol{f}(t) \end{bmatrix}$$

Parity space: effects of faults and disturbances

The effects of the disturbances and faults on the residual are then given by:

$$r(t) = \boldsymbol{v}_s^T \cdot \left(\boldsymbol{y}_s(t) - H_{u,s}\, \boldsymbol{u}_s(t) \right)$$

$$= \boldsymbol{v}_s^T \cdot \left(H_{o,s} \boldsymbol{x}(t-s) + H_{d,s} \boldsymbol{d}_s(t) + H_{f,s} \boldsymbol{f}_s(t) \right)$$

$$\boxed{r(t) = \boldsymbol{v}_s^T \cdot \left(H_{d,s} \boldsymbol{d}_s(t) + H_{f,s} \boldsymbol{f}_s(t) \right)}$$

Parity space: effects of faults and disturbances

$$\boxed{r(t) = \boldsymbol{v}_s^T \cdot \left(H_{d,s}\boldsymbol{d}_s(t) + H_{f,s}\boldsymbol{f}_s(t)\right)}$$

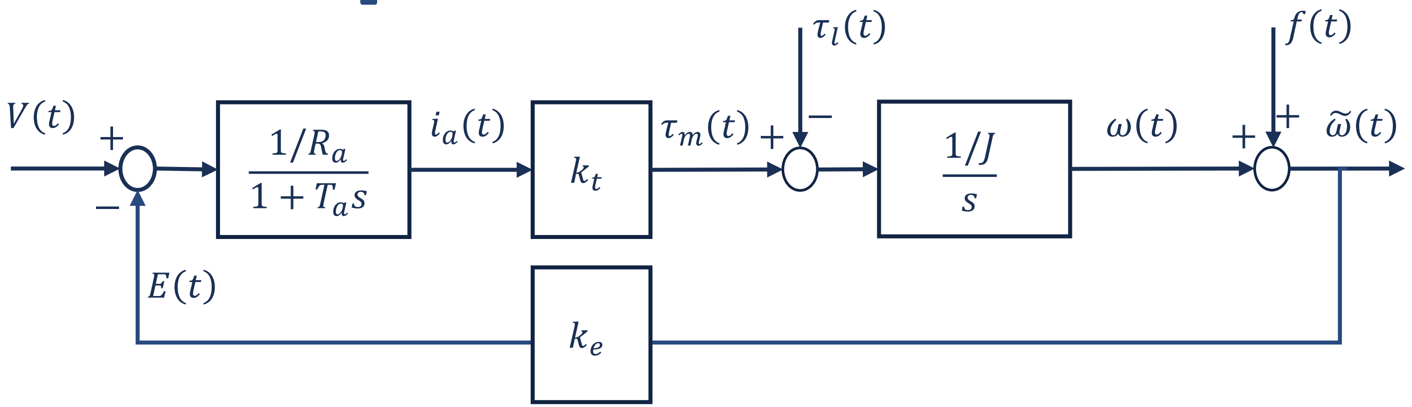
Ideally, if $\boldsymbol{v}_s^T H_{f,s} \neq \mathbf{0}$, the residual is **not zero** when a fault is present, and so **fault detection is achieved**. However, the residual is also sensitive to the disturbances.

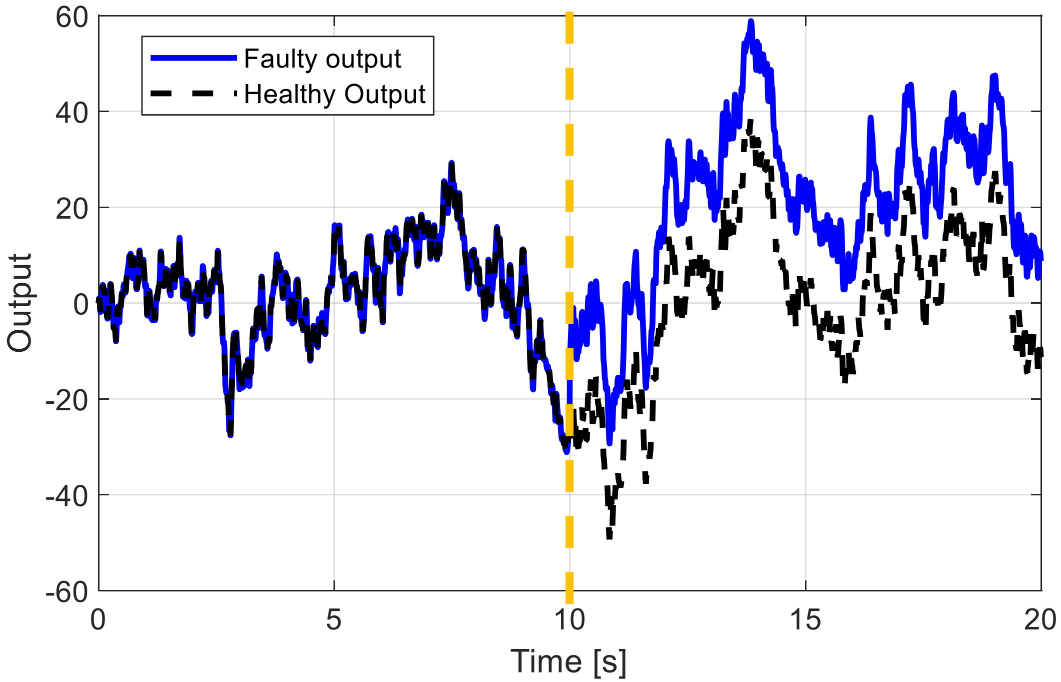
The **choice of the parity vector** has decisive impact on the performance of the residual generator. Its design can however be carried out in a straightforward manner.

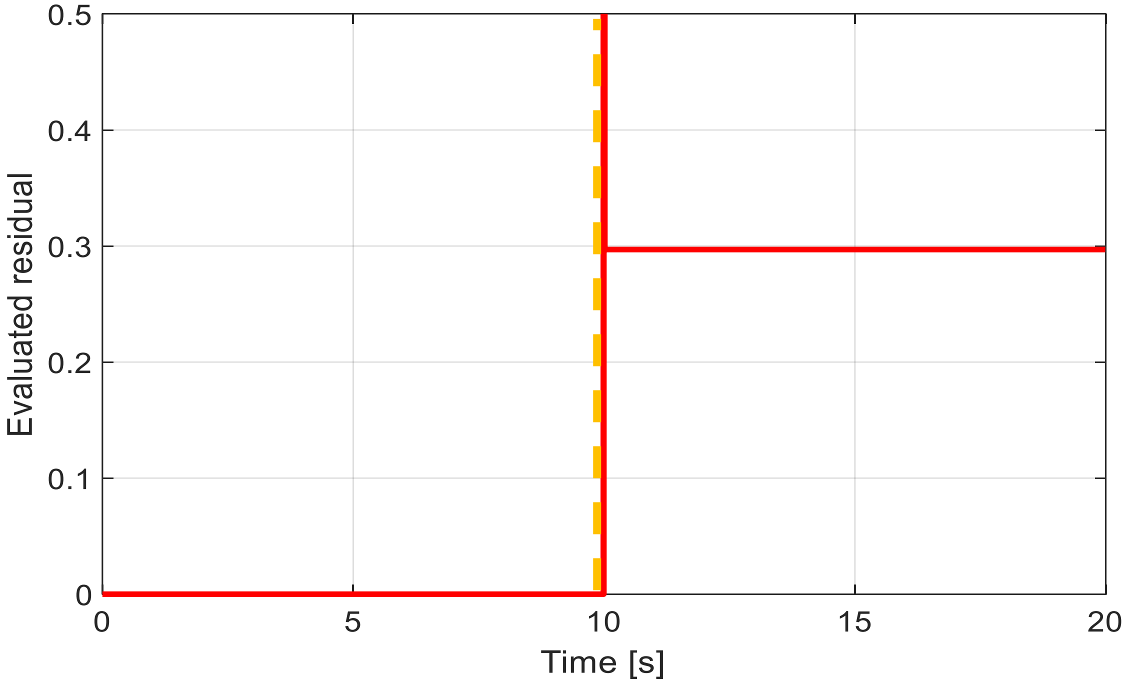
In against, the presented form of $r(t)$ is **not ideal for an on-line implementation**, since not only the actual, but also the past measurements and input data need to be recorded

Parity space: DC motor example

Suppose there is a **STEP fault**, $f(t) = 20 \cdot \mathrm{step}(t)$ on the **speed sensor** at time $t = 10$

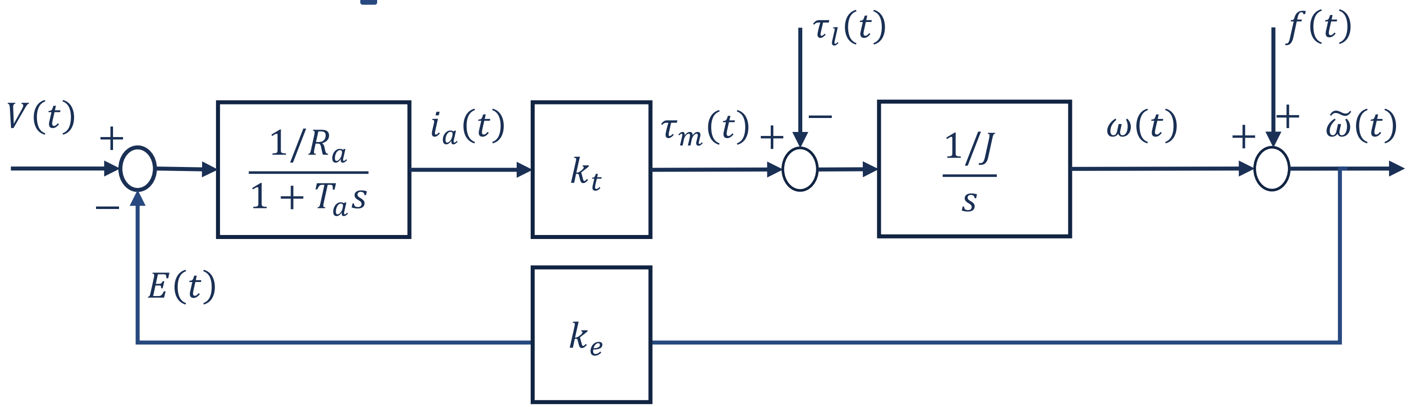


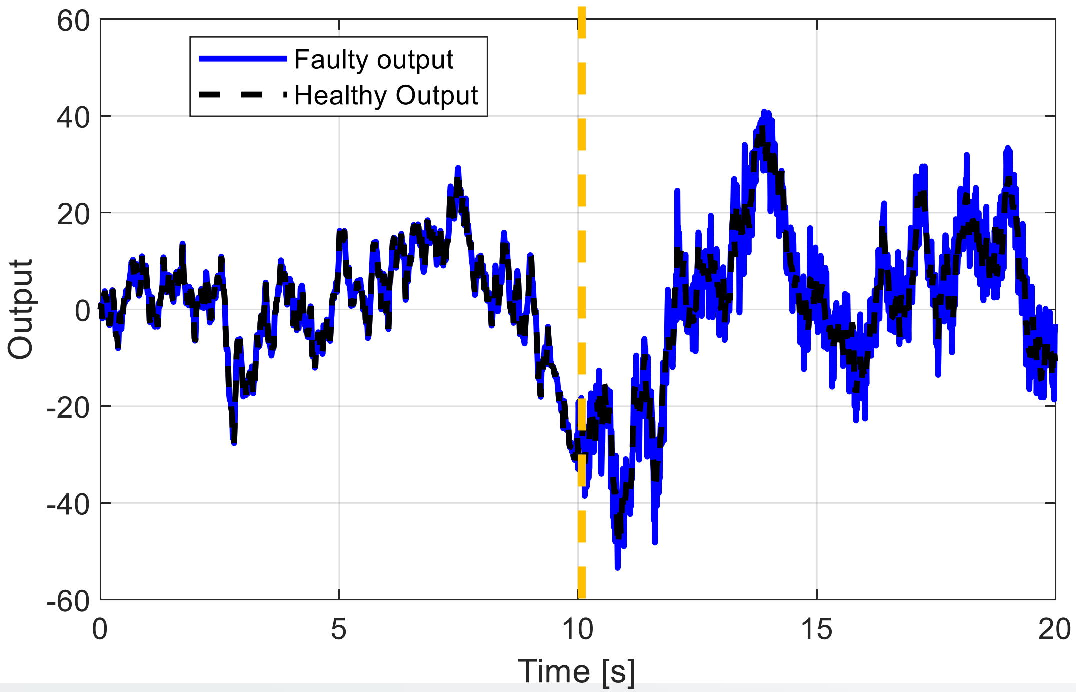


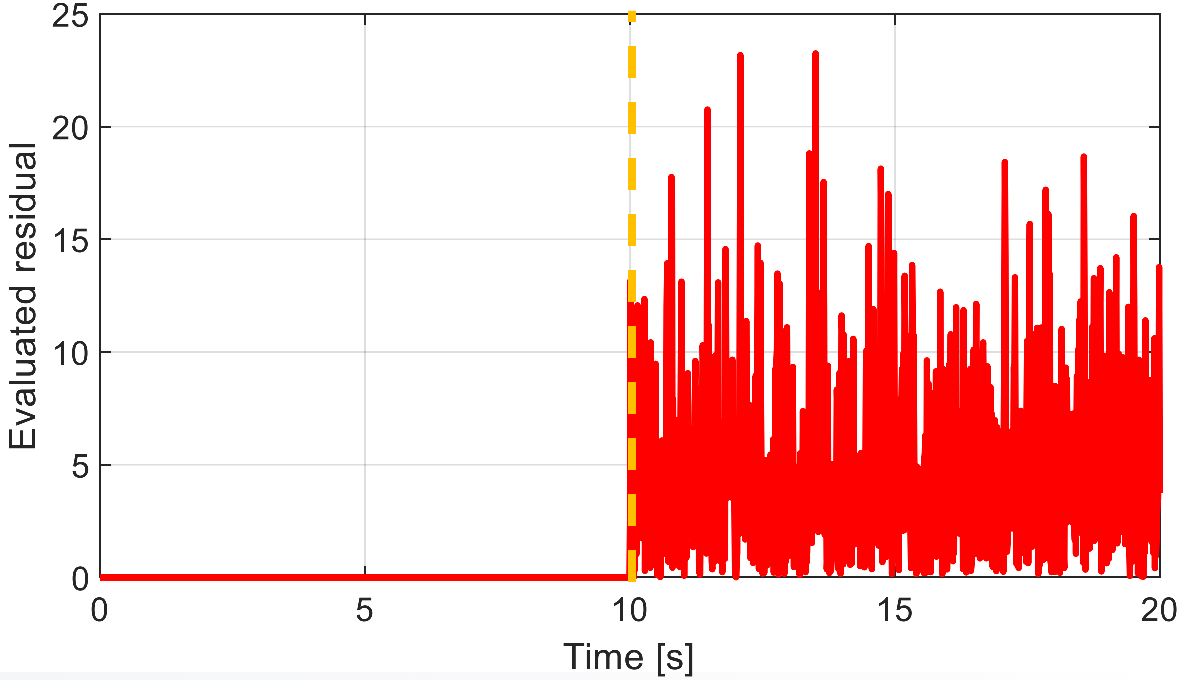


Parity space: DC motor example

Suppose there is a **NOISE fault**, $f(t) = WN(0, \lambda^2)$ with $\mathrm{SNR} = 10$ on the **speed sensor** at time $t = 10$







Parity space: input decoupling

For the fault detection purpose, an ideal residual generation would be a residual signal that **only depends on the faults** to be detected and is simultaneously **independent of the disturbances**

Recall the form of the parity space based residual generator, $\boldsymbol{v}_s \in P_s$:

$$r(t) = \boldsymbol{v}_s^T \cdot \left(H_{d,s}\boldsymbol{d}_s(t) + H_{f,s}\boldsymbol{f}_s(t) \right)$$

Thus, a **residual decoupled** from $\boldsymbol{d}_s(t)$ is delivered if and only if there exists $\boldsymbol{v}_s \in P_s$ s.t.

$\boldsymbol{v}_s^T H_{f,s} \neq \mathbf{0}$ **and** $\boldsymbol{v}_s^T H_{d,s} = \mathbf{0}$

Parity space: input decoupling

The previous condition can be equivalently restated as (with $\Delta \neq \mathbf{0} \in \mathbb{R}^{1\times m_f(s+1)}$)

$$\underbrace{\boldsymbol{v}_s^T}_{1\times p(s+1)} \cdot [\underbrace{H_{f,s}}_{p(s+1)\times m_f(s+1)} \quad \underbrace{H_{o,s}}_{p(s+1)\times n} \quad \underbrace{H_{d,s}}_{p(s+1)\times m_d(s+1)}] = [\underbrace{\Delta}_{1\times m_f(s+1)} \quad \mathbf{0} \quad \mathbf{0}]$$

So, the residual $r(t)$ is decoupled from $\boldsymbol{d}_s(t)$ **if and only if**

$$\text{rank}([H_{f,s} \quad H_{o,s} \quad H_{d,s}]) > \text{rank}([H_{o,s} \quad H_{d,s}])$$

Generally, this condition is fulfilled is there is a number of output measurements which is greater than the number of unobserved inputs, i.e. $p > (m_d + m_f)$

i.e. the colums of the matrix $H_{f,s}$ **are not linear combinations** of the columns of the matrix $[H_{o,s} \quad H_{d,s}]$

Parity space: input decoupling

The **parity space based residual** design algorithm is therefore:

1. Solve, for some s such that the input decoupling condition holds, the problem

$$\boldsymbol{v}_s^T H_{f,s} \neq \mathbf{0} \qquad \textbf{and} \qquad \boldsymbol{v}_s^T [H_{o,s} \quad H_{d,s}] = \mathbf{0}$$

2. Construct the residual generator as follows

$$r(t) = \boldsymbol{v}_s^T \cdot \left(\boldsymbol{y}_s(t) - H_{u,s}\, \boldsymbol{u}_s(t) \right)$$

This leads to a residue $r(t) = \boldsymbol{v}_s^T \cdot \left(H_{o,s}\boldsymbol{x}(t-s) + H_{d,s}\boldsymbol{d}_s(t) + H_{f,s}\boldsymbol{f}_s(t) \right) = H_{f,s}\boldsymbol{f}_s(t)$

Definition of B_f, D_f and B_d, D_d

Recall our model for a system with **additive faults and disturbances**

$$
\begin{cases}
\underset{n\times 1}{\boldsymbol{x}(t+1)} = \underset{n\times n}{A} \cdot \boldsymbol{x}(t) + \underset{n\times m_u}{B} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{n\times m_d}{B_d} \cdot \underset{m_d\times 1}{\boldsymbol{d}(t)} + \underset{n\times m_f}{B_f} \cdot \underset{m_f\times 1}{\boldsymbol{f}(t)} \\
\underset{p\times 1}{\boldsymbol{y}(t)} = \underset{p\times n}{C} \cdot \boldsymbol{x}(t) + \underset{p\times m_u}{D} \cdot \boldsymbol{u}(t) + \underset{p\times m_d}{D_d} \cdot \boldsymbol{d}(t) + \underset{p\times m_f}{D_f} \cdot \boldsymbol{f}(t)
\end{cases}
$$

In order to define B_d, D_d, B_f, D_d, it is useful (but not mandatory) to **start from the transfer functions** $\boldsymbol{G}_{yd}(z)$ and $\boldsymbol{G}_{yf}(z)$, that describe the effect of disturbances and faults on output signals, respectively. Then a **state space realization** can be performed to get B_d, D_d, B_f, D_d

Definition of B_f, D_f and B_d, D_d

We can express the relations of all the external inputs to the output signals as

$$\underset{p\times 1}{\boldsymbol{y}(t)} = \underset{p\times m_u}{\boldsymbol{G}_{yu}(z)} \cdot \boldsymbol{u}(t) + \underset{p\times m_d}{\boldsymbol{G}_{yd}(z)} \cdot \boldsymbol{d}(t) + \underset{p\times m_f}{\boldsymbol{G}_{yf}} \cdot \boldsymbol{f}(t)$$

Realization

$$\begin{cases} \underset{n\times 1}{\boldsymbol{x}(t+1)} = \underset{n\times n}{A} \cdot \boldsymbol{x}(t) + \underset{n\times m_u}{B} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{n\times m_d}{B_d} \cdot \underset{m_d\times 1}{\boldsymbol{d}(t)} + \underset{n\times m_f}{B_f} \cdot \underset{m_f\times 1}{\boldsymbol{f}(t)} \\ \underset{p\times 1}{\boldsymbol{y}(t)} = \underset{p\times n}{C} \cdot \boldsymbol{x}(t) + \underset{p\times m_u}{D} \cdot \boldsymbol{u}(t) + \underset{p\times m_d}{D_d} \cdot \boldsymbol{d}(t) + \underset{p\times m_f}{D_f} \cdot \boldsymbol{f}(t) \end{cases}$$

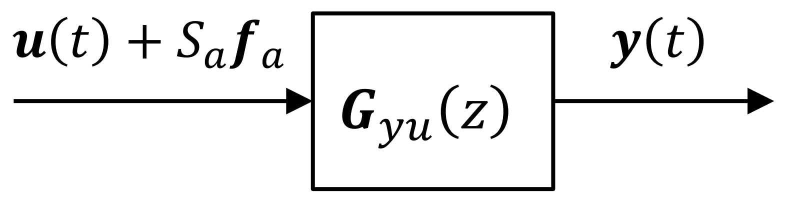
The transfer function $\boldsymbol{G}_{yd}$ **is defined by the physics** of the problem

Definition of B_f, D_f and B_d, D_d

The transfer function $\boldsymbol{G}_{yf}(z)$ **is implicitly defined** by how we choose to **model the fault**

Actuator faults $\boldsymbol{f}_a$

Modelling can be done by replacing $\boldsymbol{u}(t)$ by a perturbed input $\boldsymbol{u}(t) + S_a \boldsymbol{f}_a(t)$, with $S_a \in \mathbb{R}^{m_u \times m_f}$ a **fault distribution matrix**



$$Y(z) = \boldsymbol{G}_{yu}(z)U(z) + \boldsymbol{G}_{yu}(z)S_a \cdot F_a(z)$$

$$\boxed{\boldsymbol{G}_{yf}(z) = \boldsymbol{G}_{yu}(z)S_a}$$

Example

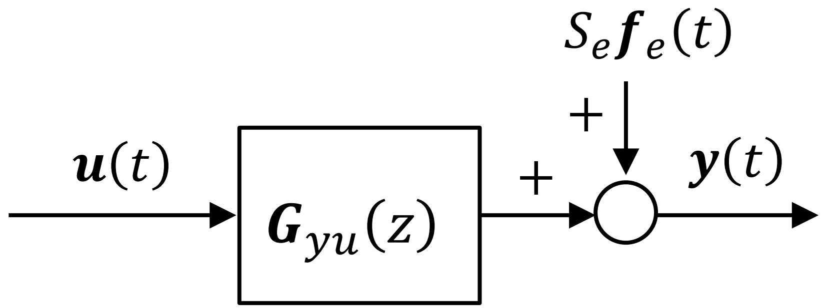
$$\underbrace{\begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}}_{\boldsymbol{u}(t)} = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{S_a} \underbrace{\begin{bmatrix} f_{a_1}(t) \\ f_{a_2}(t) \end{bmatrix}}_{\boldsymbol{f}_a(t)}$$

Definition of B_f, D_f and B_d, D_d

The transfer function $\boldsymbol{G}_{yf}(z)$ **is implicitly defined** by how we choose to **model the fault**

Sensor faults $\boldsymbol{f}_e$

Modelling can be done by replacing $\boldsymbol{y}(t)$ by a perturbed output $\boldsymbol{y}(t) + S_e \boldsymbol{f}_e(t)$, with $S_e \in \mathbb{R}^{p \times m_f}$ a **fault distribution matrix**



$$Y(z) = \boldsymbol{G}_{yu}(z)U(z) + S_e \cdot F_e(z)$$

$$\boxed{\boldsymbol{G}_{yf}(z) = S_e}$$

Example

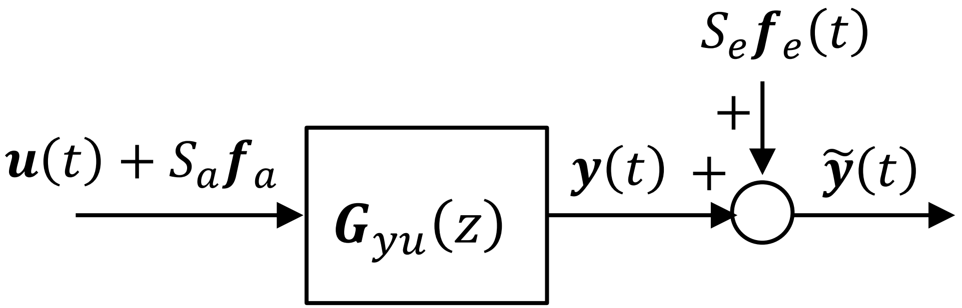
$$\underbrace{\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}}_{\boldsymbol{y}(t)} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{S_e} \underbrace{\begin{bmatrix} f_{e_1}(t) \\ f_{e_2}(t) \end{bmatrix}}_{\boldsymbol{f}_e(t)}$$

Definition of B_f, D_f and B_d, D_d

The transfer function $\boldsymbol{G}_{yf}(z)$ **is implicitly defined** by how we choose to **model the fault**

Actuator faults $\boldsymbol{f}_a$ + Sensor faults $\boldsymbol{f}_e$

The modeling is the union of the previous cases



$$Y(z) = \boldsymbol{G}_{yu}(z)U(z) + \boldsymbol{G}_{yu}S_a \cdot F_a(z)$$

$$\tilde{Y}(z) = Y(z) + S_eF_e(z) \quad \Rightarrow \quad \tilde{Y}(z) = Y(z) + S_eF_e(z) = \boldsymbol{G}_{yu}(z)U(z) + [\boldsymbol{G}_{yu}(z)S_a \quad S_e]\begin{bmatrix} F_a(z) \\ F_e(z) \end{bmatrix}$$

$$\boxed{\boldsymbol{G}_{yf}(z) = [\boldsymbol{G}_{yu}(z)S_a \quad S_e]}$$

$$\boldsymbol{f}(t) = \begin{bmatrix} \boldsymbol{f}_a(t) \\ \boldsymbol{f}_e(t) \end{bmatrix}$$

Faults vector

Parity space: DC motor example input decoupling

We want to build a parity vector **decoupled from the load disturbance** $\tau_l(t)$. In order to be able to do this, we suppose that a **2nd measurement output** is available, e.g. we measure the motor speed $\omega(t)$ with **another sensor** (with different gain)

Furthermore, suppose that we have a sensor fault, s.t. $Y(z) = \boldsymbol{G}_{yu}(z)U(z) + S_e \cdot F_e(z)$, with $\underset{\substack{p\times m_f \\ 2\times 1}}{S_e = \boldsymbol{G}_{yf}(z)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$,i.e. the fault acts only on the 1° output (1° speed sensor)

$$\underset{\substack{p\times m_u \\ 2\times 1}}{\boldsymbol{G}_{yu}(z)} = \begin{bmatrix} \dfrac{1.184\,z + 1.184}{z^2 - 0.9852\,z + 2.943\cdot 10^{-5}}\cdot 1 \\ \dfrac{1.184\,z + 1.184}{z^2 - 0.9852\,z + 2.943\cdot 10^{-5}}\cdot 0.99 \end{bmatrix} \qquad \underset{\substack{p\times m_d \\ 2\times 1}}{\boldsymbol{G}_{yd}(z)} = \begin{bmatrix} \dfrac{-123.6z + 0.003637}{z^2 - 0.9852\,z + 2.943\cdot 10^{-5}}\cdot 1 \\ \dfrac{-123.6z + 0.003637}{z^2 - 0.9852\,z + 2.943\cdot 10^{-5}}\cdot 0.99 \end{bmatrix}$$

Parity space: DC motor example input decoupling

A state space realization of $\boldsymbol{G}_{yu}(z)$ leads to

$$\underset{\substack{n \times n \\ 2 \times 2}}{A} = \begin{bmatrix} 0.9851 & 0.0021 \\ 0.0019 & 0 \end{bmatrix} \quad \underset{\substack{n \times m_u \\ 2 \times 1}}{B} = \begin{bmatrix} -0.3087 \\ -0.1275 \end{bmatrix} \quad \underset{\substack{p \times n \\ 1 \times 2}}{C} = \begin{bmatrix} -7.7413 & 18.95 \end{bmatrix} \quad \underset{\substack{p \times m_u \\ 2 \times 1}}{D} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\underset{\substack{n \times m_d \\ 2 \times 1}}{B_d} = \begin{bmatrix} 15.9988 \\ 0.0315 \end{bmatrix} \quad \underset{\substack{n \times m_f \\ 2 \times 1}}{B_f} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \underset{\substack{p \times m_d \\ 2 \times 1}}{D_d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \underset{\substack{p \times m_f \\ 2 \times 1}}{D_f} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\underset{\substack{p(s+1) \times n \\ 2(2+1) \times 2}}{H_{o,s}} = \begin{bmatrix} -7.7413 & 18.9468 \\ -7.6639 & 9.3634 \\ -7.6079 & -0.0156 \\ -7.5319 & -0.0154 \\ -7.4950 & -0.0156 \\ -7.4200 & -0.0154 \end{bmatrix} \quad H_{u,s} = \underset{\substack{p(s+1) \times m_u(s+1) \\ 2(2+1) \times 1(2+1)}}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1.1842 & 0 & 0 \\ 1.1723 & 0 & 0 \\ 2.3508 & 1.1842 & 0 \\ 2.3273 & 1.1723 & 0 \end{bmatrix}} \quad H_{f,s} = \underset{\substack{p(s+1) \times m_f(s+1) \\ 2(2+1) \times 1(2+1)}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}} \quad H_{d,s} = \underset{\substack{p(s+1) \times m_d(s+1) \\ 2(2+1) \times 1(2+1)}}{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -123.5527 & 0 & 0 \\ -122.3172 & 0 & 0 \\ -121.7180 & -123.5527 & 0 \\ -120.5008 & -122.3172 & 0 \end{bmatrix}}$$

Parity space: DC motor example input decoupling

We want to solve for $\boldsymbol{v}_s$ the following problem:

$$\boldsymbol{v}_s^T H_{f,s} \neq \boldsymbol{0} \qquad \textbf{and} \qquad \boldsymbol{v}_s^T [H_{o,s} \quad H_{d,s}] = \boldsymbol{0}$$

We can recast it in the standard form for solving linear systems: $[H_{o,s} \quad H_{d,s}]^T \boldsymbol{v}_s = \boldsymbol{0}$

This system is homogeneous admits always the particular solution $\boldsymbol{v}_s^o = \boldsymbol{0}$. If the condition $\mathrm{rank}([H_{f,s} \quad H_{o,s} \quad H_{d,s}]) > \mathrm{rank}([H_{o,s} \quad H_{d,s}])$ holds, one way to obtain a nicer solution is to add a solution from the **nullspace** of $[H_{o,s} \quad H_{d,s}]^T$

$$\boldsymbol{v}_s^{\mathrm{dec}} = \boldsymbol{v}_s^0 + \mathrm{null}([H_{o,s} \quad H_{d,s}]^T) \cdot [\alpha_1 \dots a_{ns}]^T$$

- $a \in \mathbb{R}$
- n_s: dimension of null space

Check a-posteriori if $\boldsymbol{v}_s^T H_{f,s} \neq \boldsymbol{0}$

Parity space: DC motor example input decoupling

Always check the solution!

$$\left(\boldsymbol{v}_s^{\text{dec}}\right)^T [H_{o,s} \quad H_{d,s}] =$$

$$= [-0.9900 \quad 1 \quad -1.9800 \quad 2 \quad -2.9700 \quad 3] \cdot \begin{bmatrix} -7.7413 & 18.9468 & 0 & 0 & 0 \\ -7.6639 & 9.3634 & 0 & 0 & 0 \\ -7.6079 & -0.0156 & -123.5527 & 0 & 0 \\ -7.5319 & -0.0154 & -122.3172 & 0 & 0 \\ -7.4950 & -0.0156 & -121.7180 & -123.5527 & 0 \\ -7.4200 & -0.0154 & -120.5008 & -122.3172 & 0 \end{bmatrix}$$

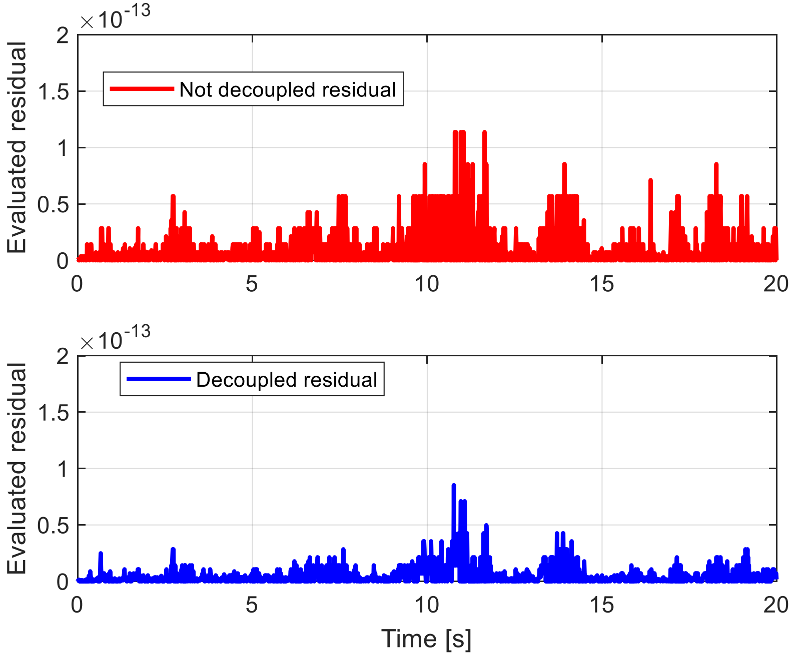
$$= [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$ **CORRECT! ☺**

$$\left(\boldsymbol{v}_s^{\text{dec}}\right)^T H_{f,s} = [-0.99 \quad 1 \quad -1.98 \quad 2 \quad -2.97 \quad 3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = [-0.99 \quad -1.98 \quad -2.97] \neq 0$$ **OK! ☺**

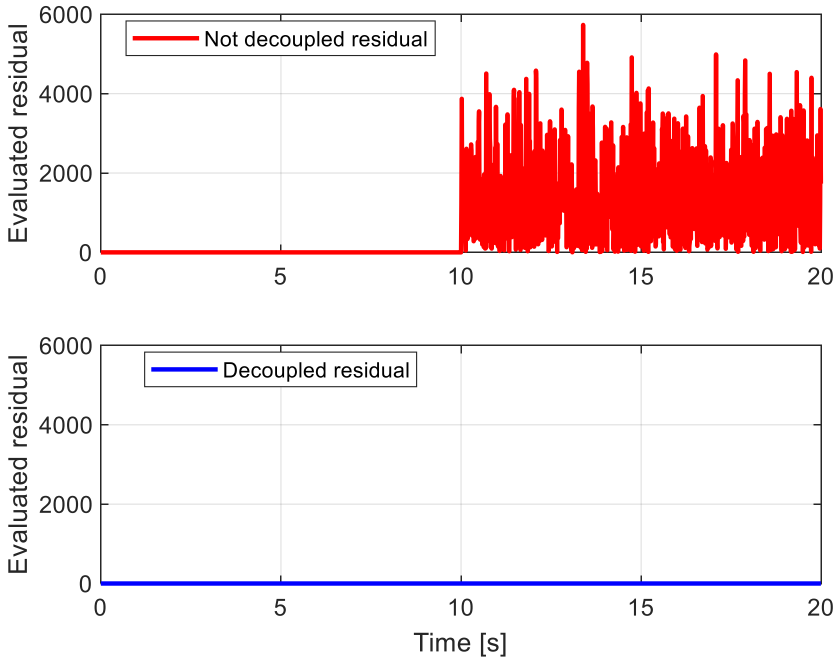
Parity space: DC motor example input decoupling

We can now compare the results of the **decoupled parity vector** v_s^{dec}, with respect to the **not decoupled parity vector** v_s

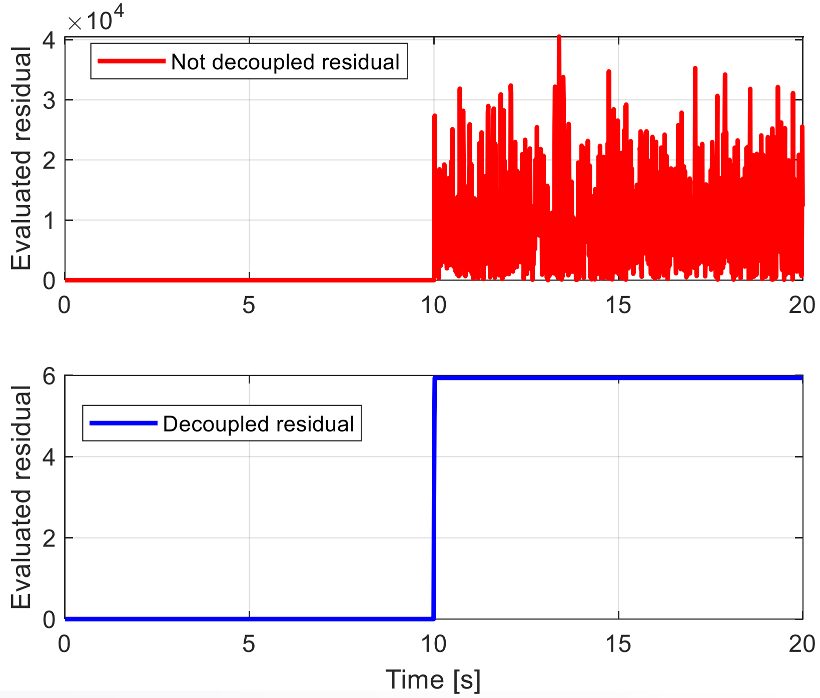
Residual with $d(t) = f(t) = 0$



Residual with $d(t) = WN(0, \lambda^2)$ λ^2 s.t. SNR $= 1$, $f(t) = 0$



Residual with $d(t) = WN(0, \lambda^2)$ λ^2 s.t. SNR $= 1$, $f(t) = 1 \cdot \mathrm{step}(t)$



Outline

1. Schematic of the approach
2. Dynamical systems
3. Parity space approach
4. **Diagnostic observer**
5. Application to EMA fault detection

Diagnostic observer

A **disadvantage** of the parity approach method is that it requires to **store a sample** of past inputs and outputs at time $t-1, \dots, t-s$, in order to compute the residual signal at the current time t. The implementation is **non-recursive**

It is possible to employ a different residual generator scheme, known as diagnostic observer, by **using the designed parity vector $\boldsymbol{v}_s$**

This allows to employ a **recursive implementation**, s.t. $r(t)$ is computed from $r(t-1)$

A common strategy based on **«parity space design, observer-based implementation»** is vastly used

Diagnostic observer

A **diagnostic observer** can be formulated as the following **dynamical system**

$$\begin{cases} \underset{s\times 1}{\boldsymbol{m}(t+1)} = \underset{s\times s}{G} \cdot \underset{s\times 1}{\boldsymbol{m}(t)} + \underset{s\times m_u}{H} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{s\times p}{L} \cdot \underset{p\times 1}{\boldsymbol{y}(t)} \\ \underset{1\times 1}{r(t)} = -\underset{1\times s}{w} \cdot \underset{s\times 1}{\boldsymbol{m}(t)} - \underset{1\times m_u}{q} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{1\times p}{v} \cdot \underset{p\times 1}{\boldsymbol{y}(t)} \end{cases}$$

We can interpret the quantity $-w \cdot \boldsymbol{m}(t) - q \cdot \boldsymbol{u}(t)$ as an **estimate** for $v\boldsymbol{y}(t)$. In this way, the residual signal reads as $r(t) = \hat{y}(t) - vy(t)$. Notice how $\hat{y}(t)$ depends on $\boldsymbol{m}(t)$, which in turn depends on $y(t-1)$.

For this reason, observer approaches are known as **closed-loop approaches.** This give them **more robustness** with respect to **modeling errors**

Diagnostic observer

Let $\boldsymbol{v}_s^T = \begin{bmatrix} \underbrace{\boldsymbol{v}_{s,0}^T}_{1\times p} & \boldsymbol{v}_{s,1}^T & \cdots & \boldsymbol{v}_{s,s}^T \end{bmatrix}$, be a parity vector, $\boldsymbol{v}_s \in \mathbb{R}^{p(s+1)\times 1}$, $\boldsymbol{v}_s \in P_s$. Then, we have:

$$\underbrace{G}_{s\times s-1} = [G_0 \quad \boldsymbol{g}] \qquad G_0 = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \qquad \boldsymbol{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_s \end{bmatrix} \qquad \underbrace{L}_{s\times p} = -\underbrace{\begin{bmatrix} \boldsymbol{v}_{s,0} \\ \vdots \\ \boldsymbol{v}_{s,s-1} \end{bmatrix}}_{s\times p} - \underbrace{\boldsymbol{g}}_{s\times 1} \cdot \underbrace{\boldsymbol{v}_{s,s}}_{1\times p}$$

$$\underbrace{\begin{bmatrix} \underbrace{H}_{s\times m_u} \\ \underbrace{q}_{1\times m_u} \end{bmatrix}}_{s+1\times m_u} = \begin{bmatrix} \overbrace{\boxed{\begin{matrix} \boldsymbol{v}_{s,0} + g_1\boldsymbol{v}_{s,s} & \boldsymbol{v}_{s,1} & \boldsymbol{v}_{s,2} & \cdots & \boldsymbol{v}_{s,s-1} & \boldsymbol{v}_{s,s} \\ \boldsymbol{v}_{s,2} + g_2\boldsymbol{v}_{s,s} & \boldsymbol{v}_{s,2} & \cdots & \cdots & \boldsymbol{v}_{s,s} & \mathbf{0} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \boldsymbol{v}_{s,s} + g_3\boldsymbol{v}_{s,s} & \boldsymbol{v}_{s,s} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{matrix}}}^{s\times p(s+1)} \\ \underbrace{\boxed{\begin{matrix} \boldsymbol{v}_{s,s} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{matrix}}}_{1\times p(s+1)} \end{bmatrix} \underbrace{\begin{bmatrix} D \\ CB \\ CAB \\ \vdots \\ CA^{s-2}B \\ CA^{s-1}B \end{bmatrix}}_{p(s+1)\times m_u}$$

$$\underbrace{\boldsymbol{w}}_{1\times s} = [0 \quad \cdots \quad 1]$$

$$\underbrace{\boldsymbol{v}}_{1\times p} = \boldsymbol{v}_{s,s}$$

Diagnostic observer

$$\begin{cases} \underset{s\times 1}{\boldsymbol{m}(t+1)} = \underset{s\times s}{G} \cdot \underset{s\times 1}{\boldsymbol{m}(t)} + \underset{s\times m_u}{H} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{s\times p}{L} \cdot \underset{p\times 1}{\boldsymbol{y}(t)} \\ \underset{1\times 1}{r(t)} = -\underset{1\times s}{\boldsymbol{w}} \cdot \underset{s\times 1}{\boldsymbol{m}(t)} - \underset{1\times m_u}{\boldsymbol{q}} \cdot \underset{m_u\times 1}{\boldsymbol{u}(t)} + \underset{1\times p}{\boldsymbol{v}} \cdot \underset{p\times 1}{\boldsymbol{y}(t)} \end{cases}$$

The vector $\boldsymbol{g}$ in the matrix $\boldsymbol{G}$ is an **additional degree of freedom** given by the observer formulation. It can be used to assign a **desired dynamic** to the observer

When $\boldsymbol{g} = \mathbf{0}$, then the DO formulation is **analogous** to the parity space design

The obtained residual generator is a **dynamical system** (a **filter)**, that takes as input the **system inputs** $u(t)$ **and outputs** $y(t)$, and get as output the **residual signal** $r(t)$

Diagnostic observer: DC motor

We now apply the DO scheme using the **decoupled parity vector** v_s^{dec} previously found.

Denote with $\boldsymbol{Q}(z)$ **the transfer function of the residual generator** , and let $\boldsymbol{g} = \boldsymbol{0}$

$$\underset{3\times 1}{\boldsymbol{Q}(z)} = \begin{bmatrix} \dfrac{-8.882\cdot 10^{-16}}{z^2} \\ \dfrac{2.97z^2 - 1.98z - 0.99}{z^2} \\ \dfrac{3z^2 - 2z + 1}{z^2} \end{bmatrix}$$

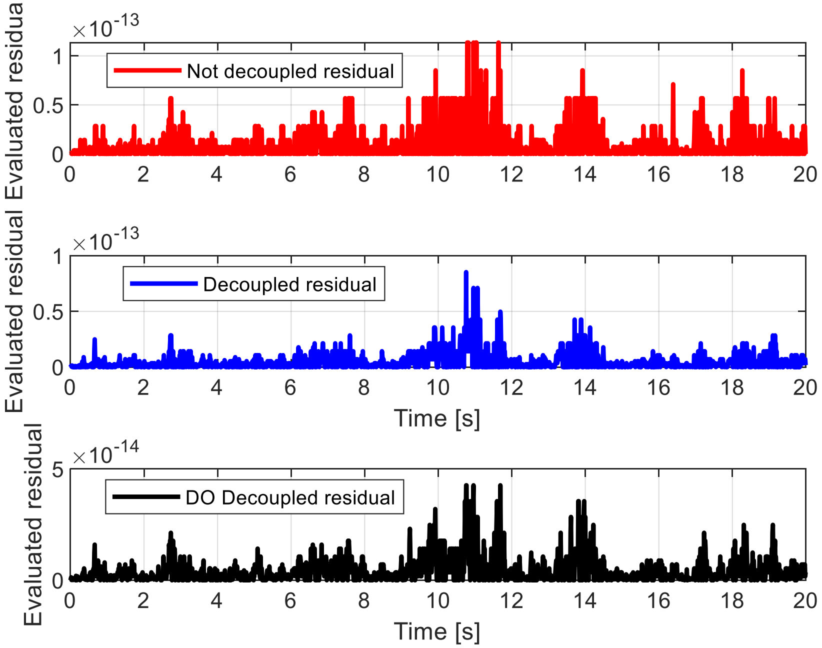
- This filter has **3 inputs** and **1 output**
- 1° input: $u(t)$ - 2nd input: $y_1(t)$ - 3rd input: $y_2(t)$,

  where $\boldsymbol{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$
- Output: residual signal $r(t)$

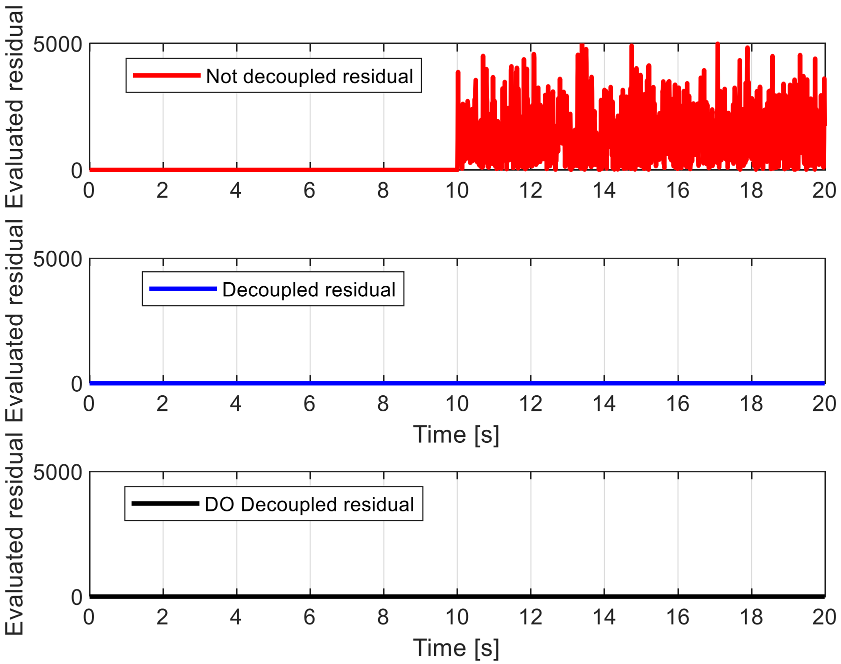
Diagnostic observer: DC motor

Compare the results with the parity base approach

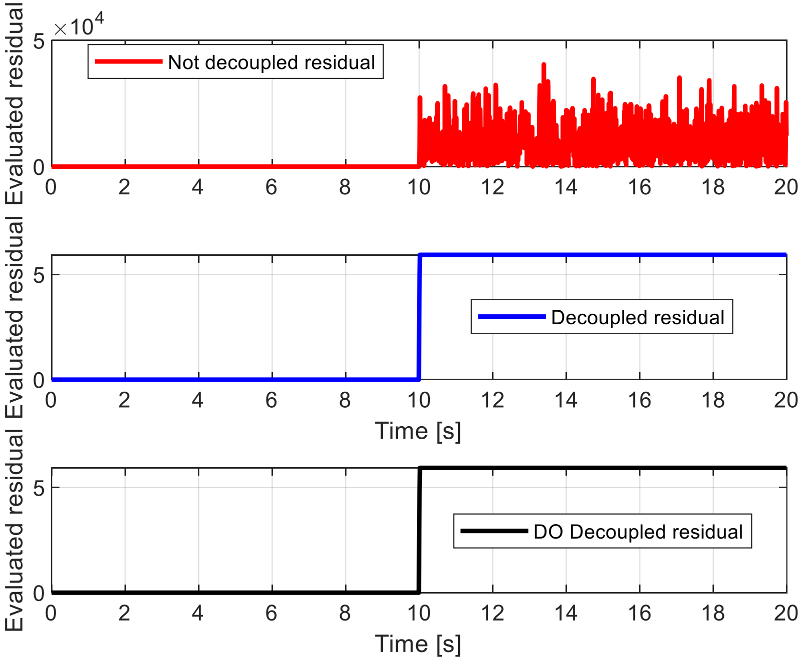
Residual with $d(t) = f(t) = 0$



Residual with $d(t \geq 10) = WN(0, \lambda^2)$
λ^2 s.t. SNR $= 1$, $f(t) = 0$



Residual with $d(t \geq 10) = WN(0, \lambda^2)$
λ^2 s.t. SNR $= 1$, $f(t \geq 10) = 1 \cdot \mathrm{step}(t)$

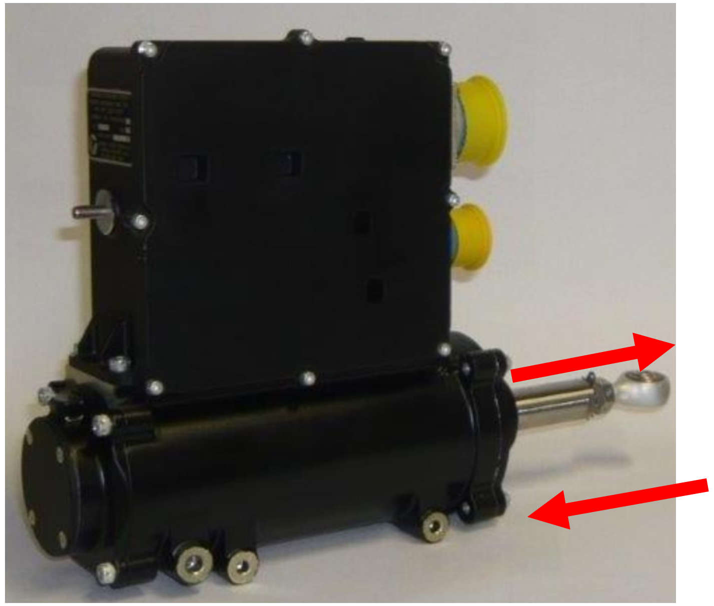


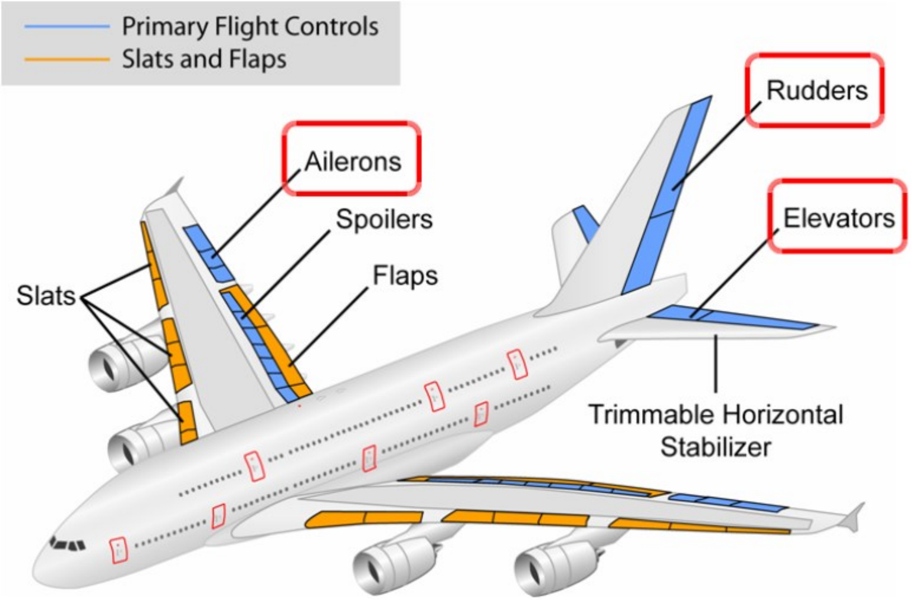
Outline

The REPRISE project

The *Reliable Electromechanical actuator for PRImary SurfacE with health monitoring* is a H2020 project which aims to **design new EMA** for flight actuation systems **with HM capabilities**

This project has received funding from the the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 717112 (project acronym: REPRISE).





The REPRISE project

The following activities were carried out to develop a **model-based fault diagnosis** method:

1. Experimental data acquisition (with **endurance** and degradation of **ballscrew** component)

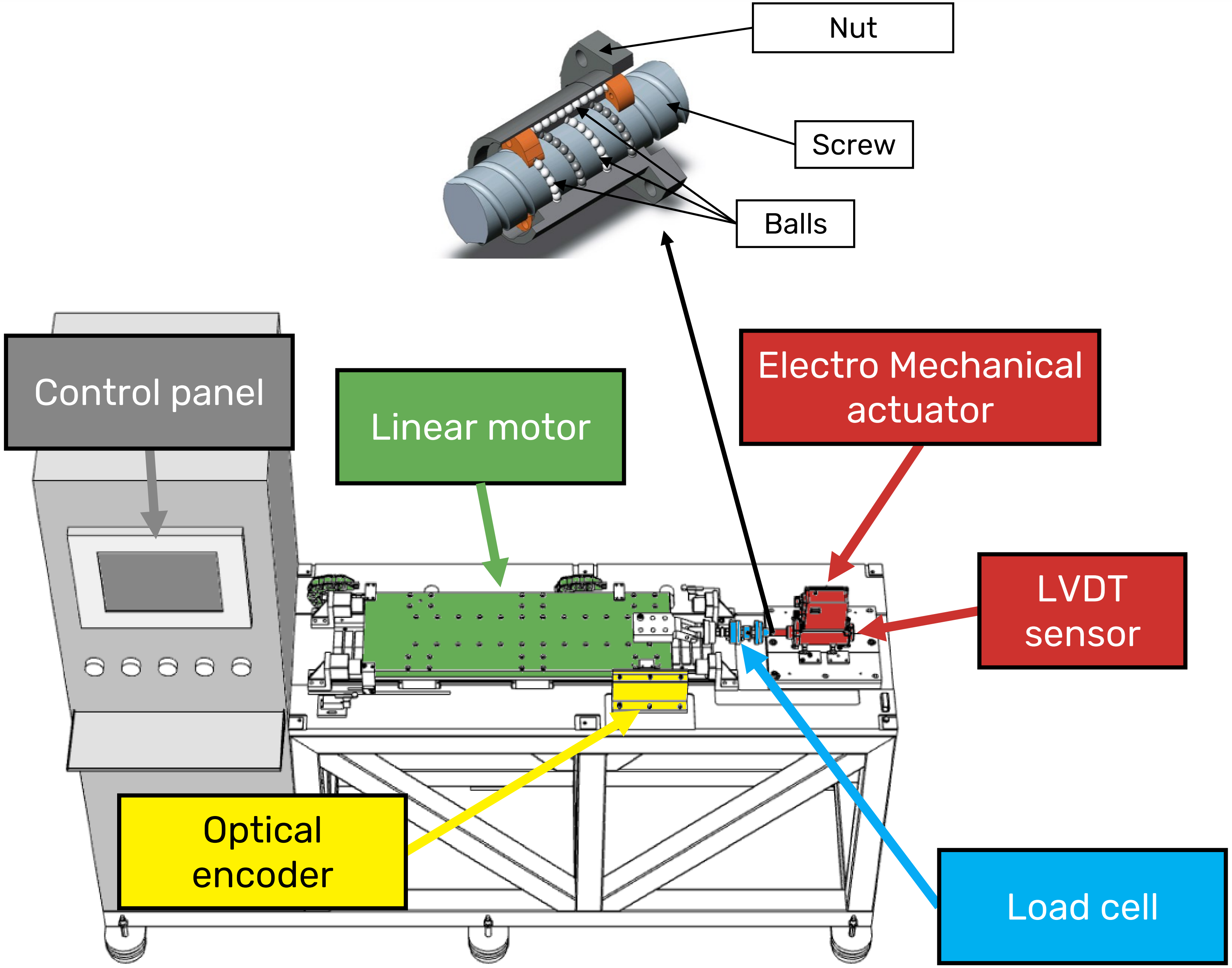
2. Estimation of the EMA **closed-loop** transfer function $G_{yu}(z) = \frac{X(z)}{\bar{X}(z)}$
   - $\bar{x}(t)$: position reference
   - $x(t)$: measured position

3. Linear **residual generator design** based on parity space and diagnostic observer

Test bench layout

Measurements:

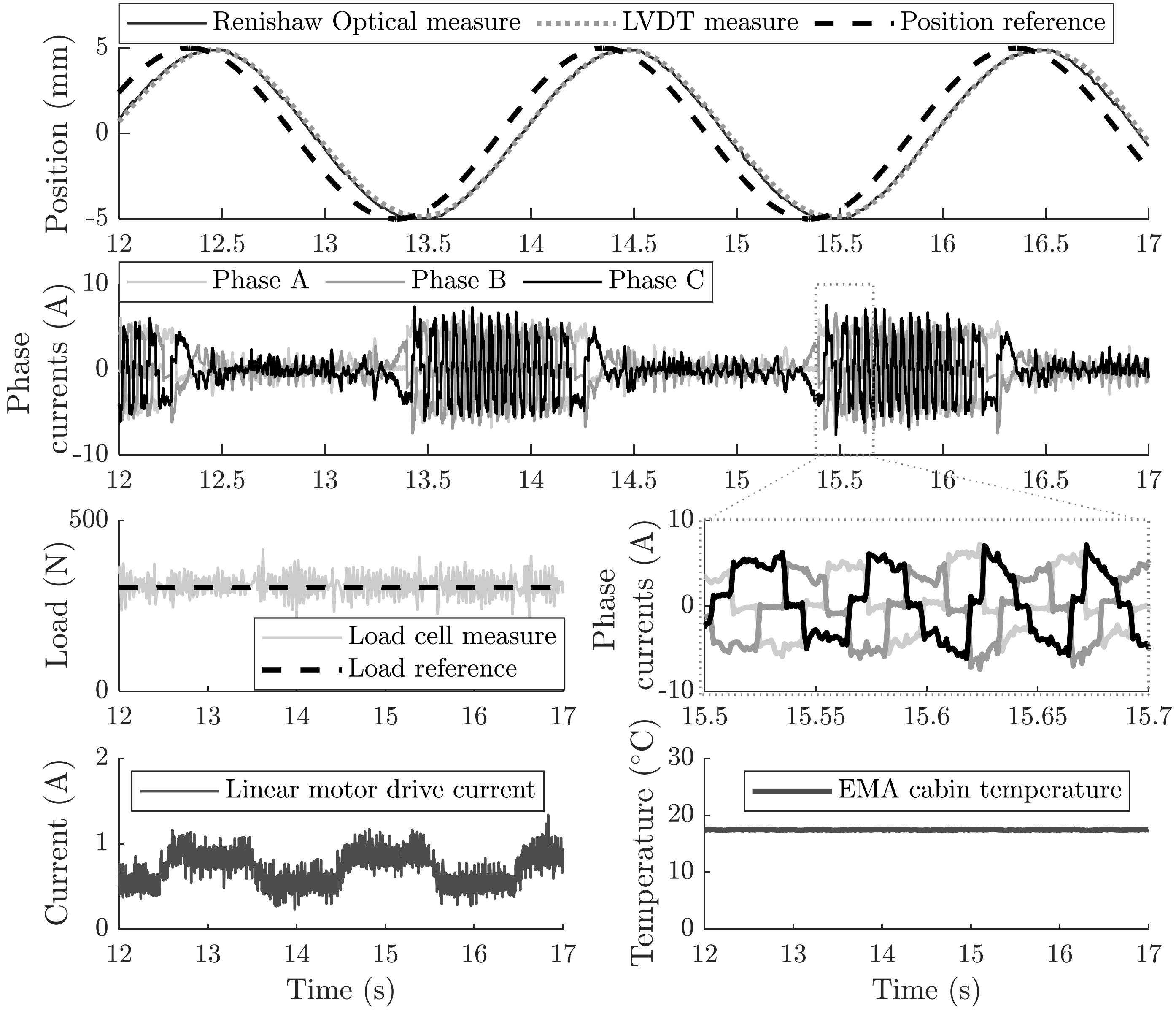
1. EMA Phase currents
2. EMA LVDT position
3. Linear motor position
4. EMA Reference position
5. Load cell



Test bench layout

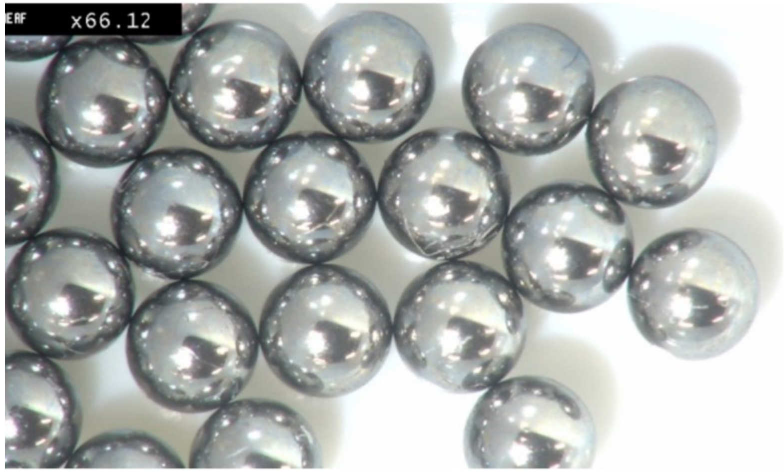
Measurements:

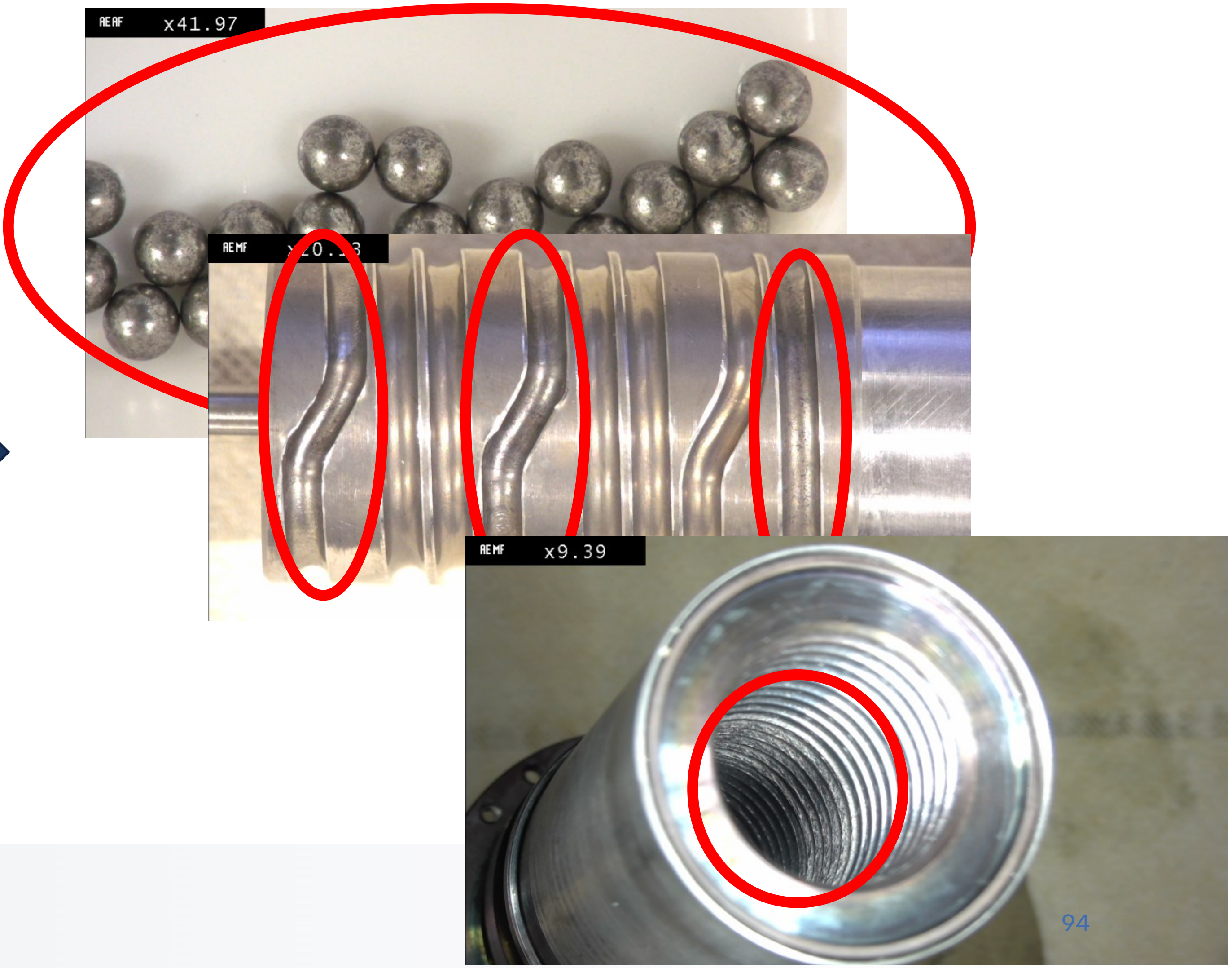
1. EMA Phase currents
2. EMA LVDT position
3. Linear motor position
4. EMA Reference position
5. Load cell



Ballscrew degradation

“Hammered” **balls**, “indented” **recirculation circuits** in ballscrew,“scratched” **nut thread**





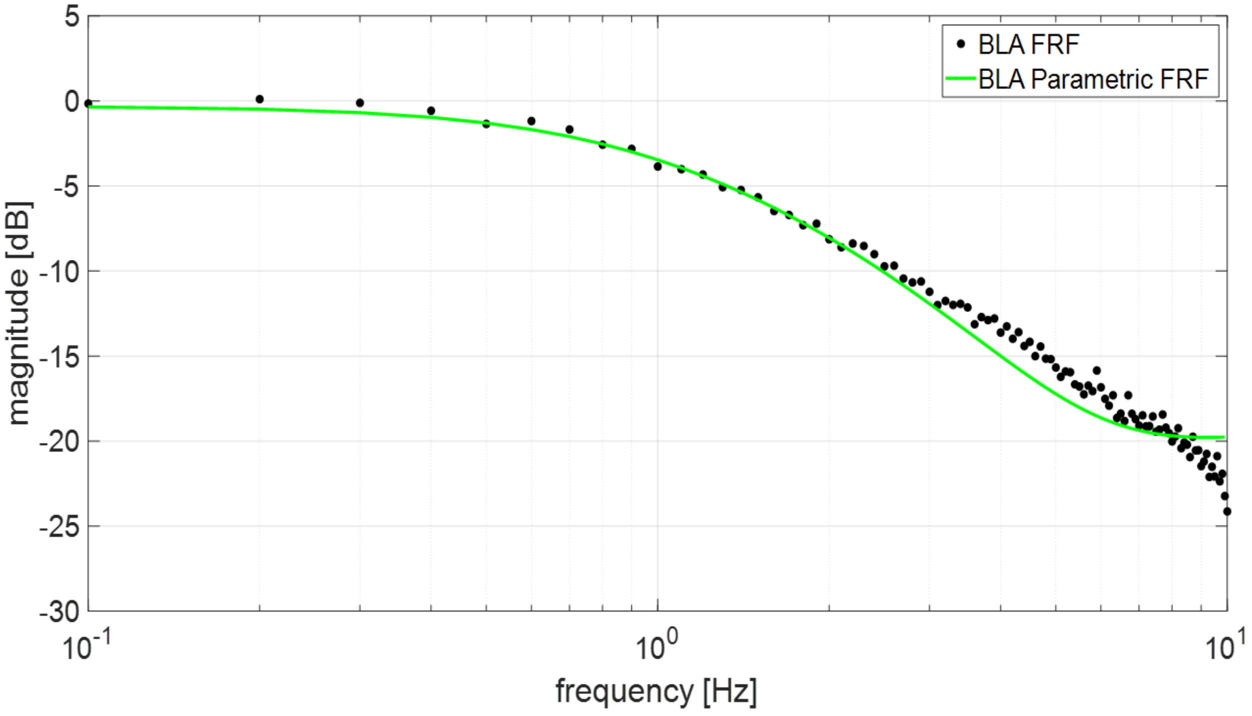
EMA closed-loop model

Identification method:

1. **Multisine** excitation

2. **Nonparametric estimate** of the Best Linear Approximation (BLA)

3. **Parametric estimate** fitting FRF data



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

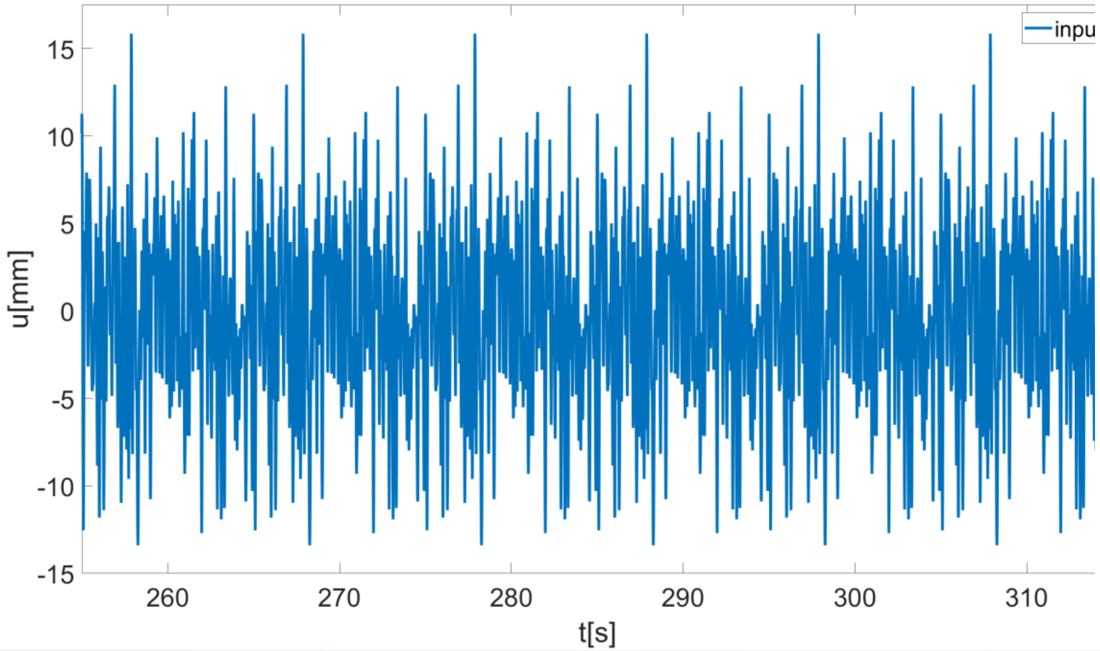
- $\bar{x}(t)$: position reference
- $x(t)$: position measure LVDT

Fault behaviour

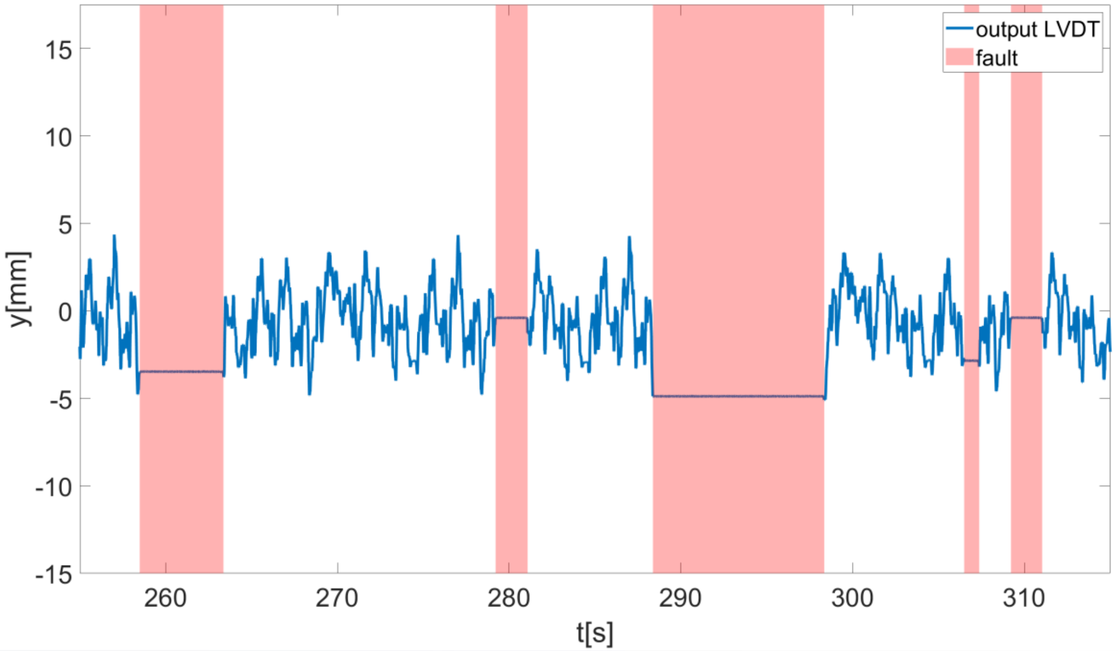
When the degradation took over, the **EMA undergone small-jams** during the operation

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

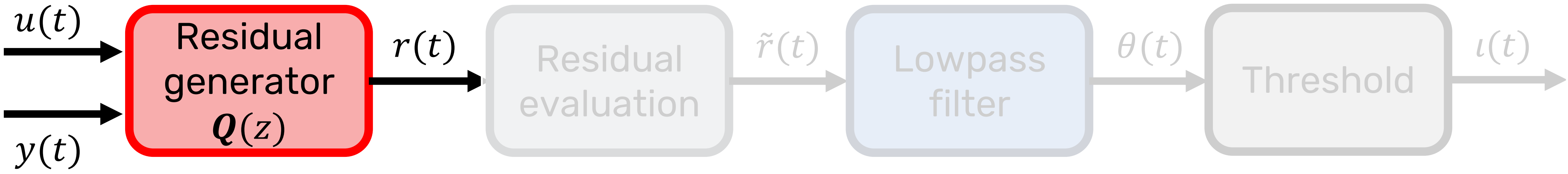
Multisine position reference

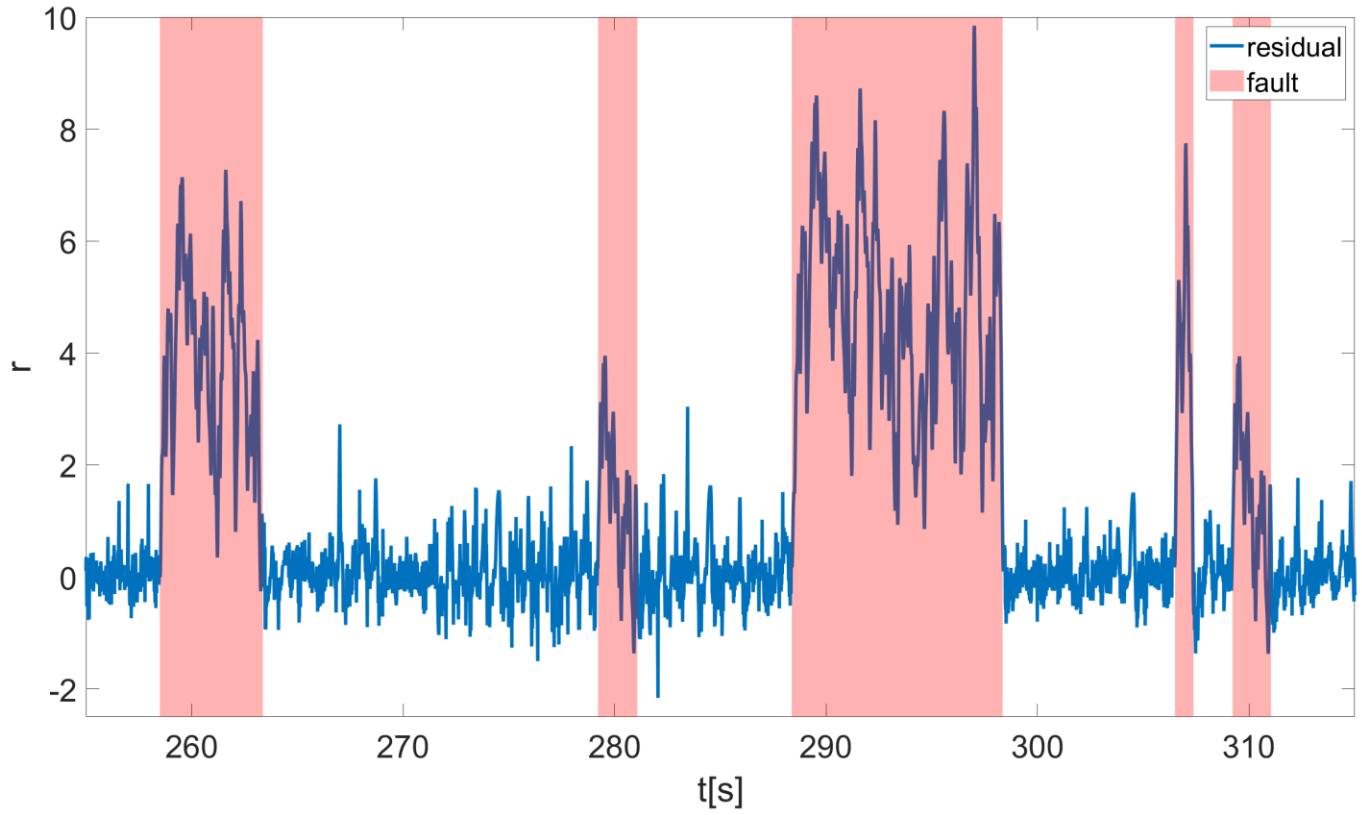


Output position (LVDT)

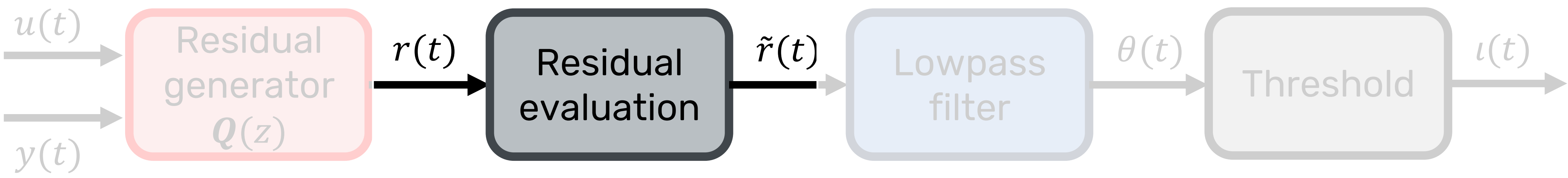


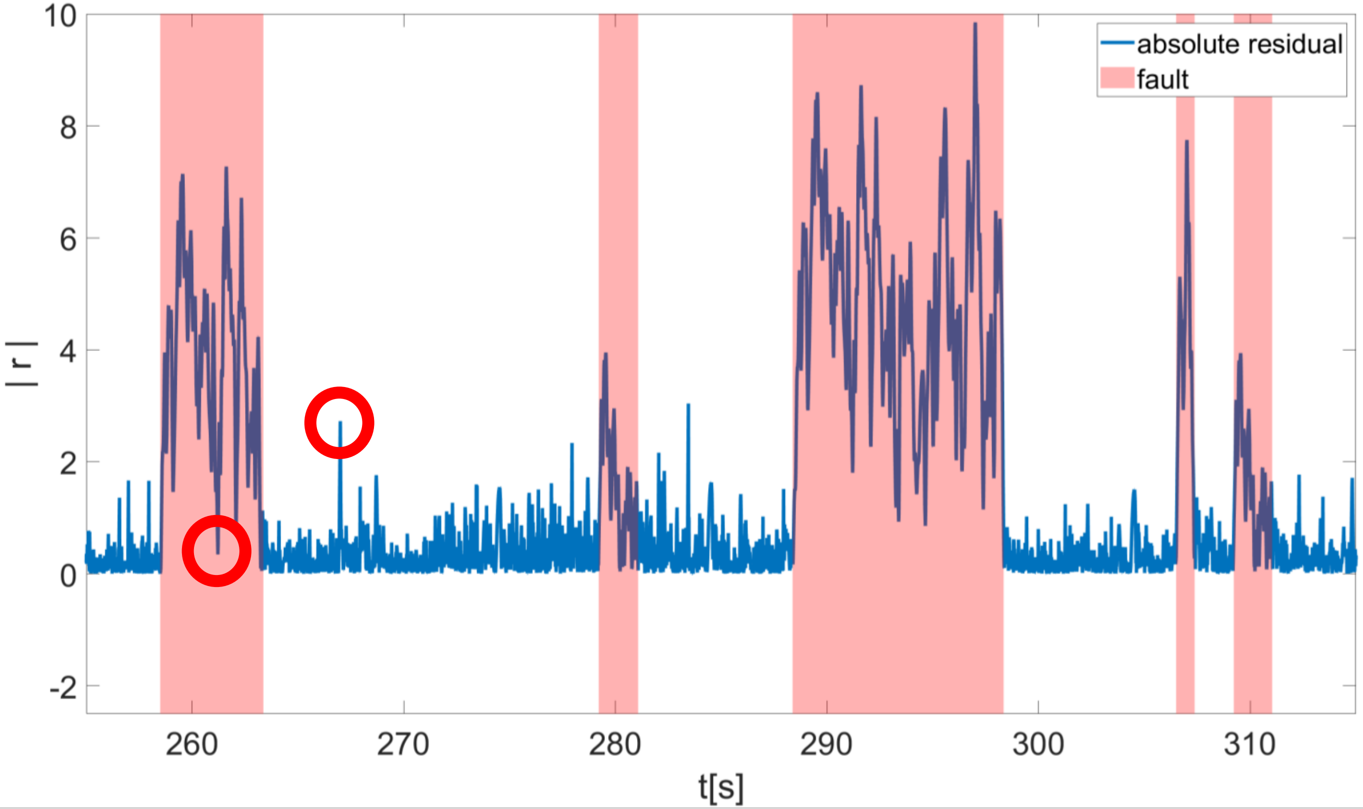
Residual generator



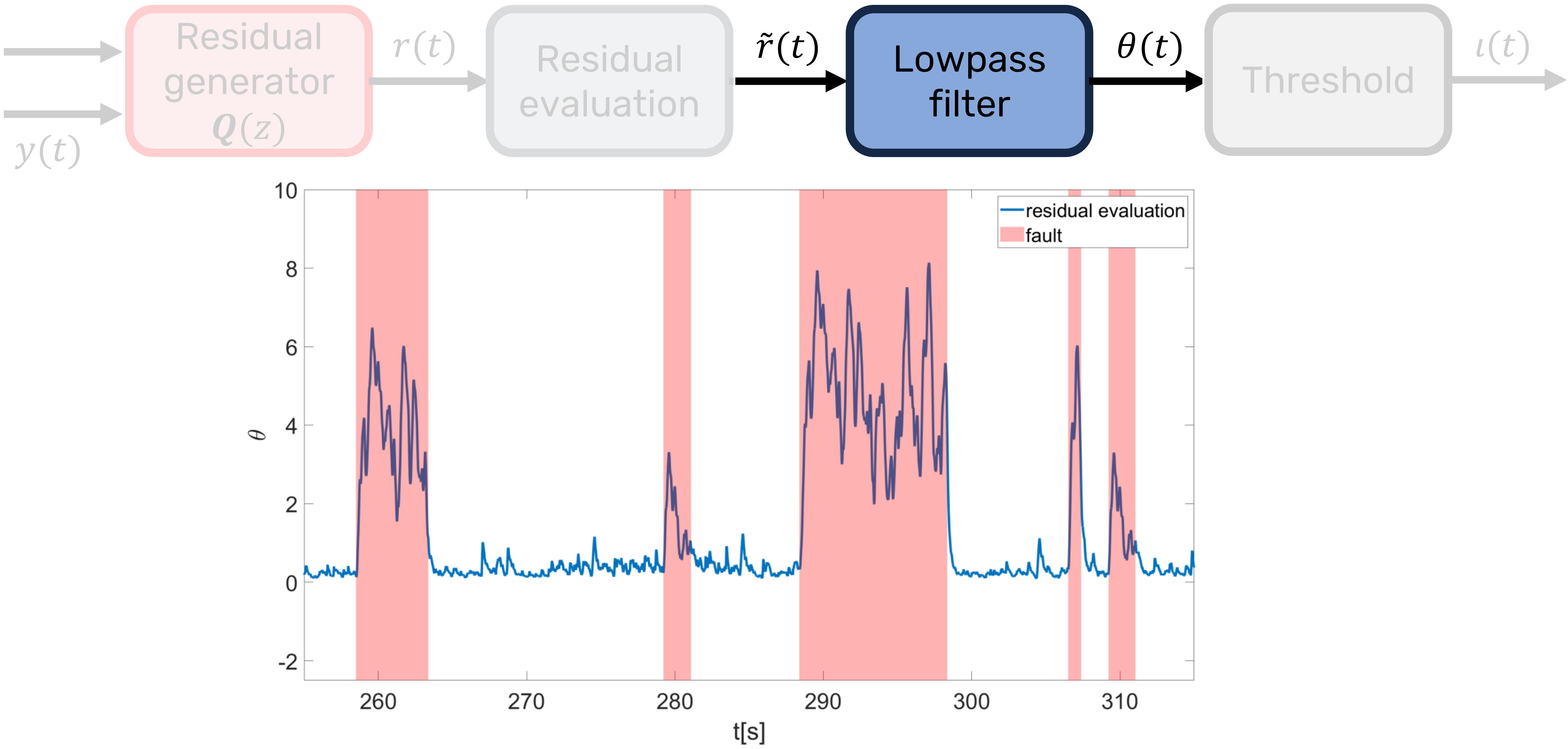


Residual generator

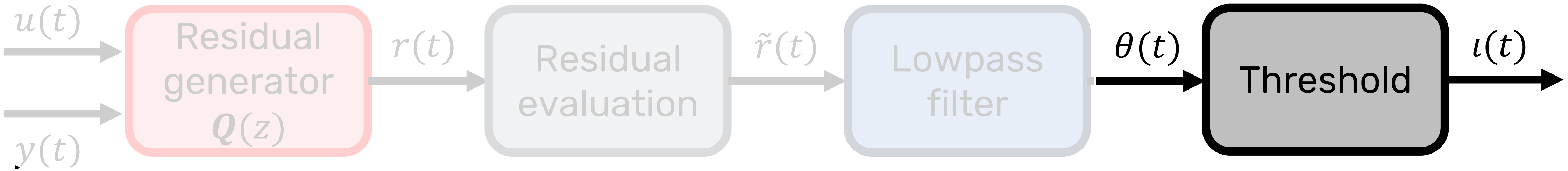


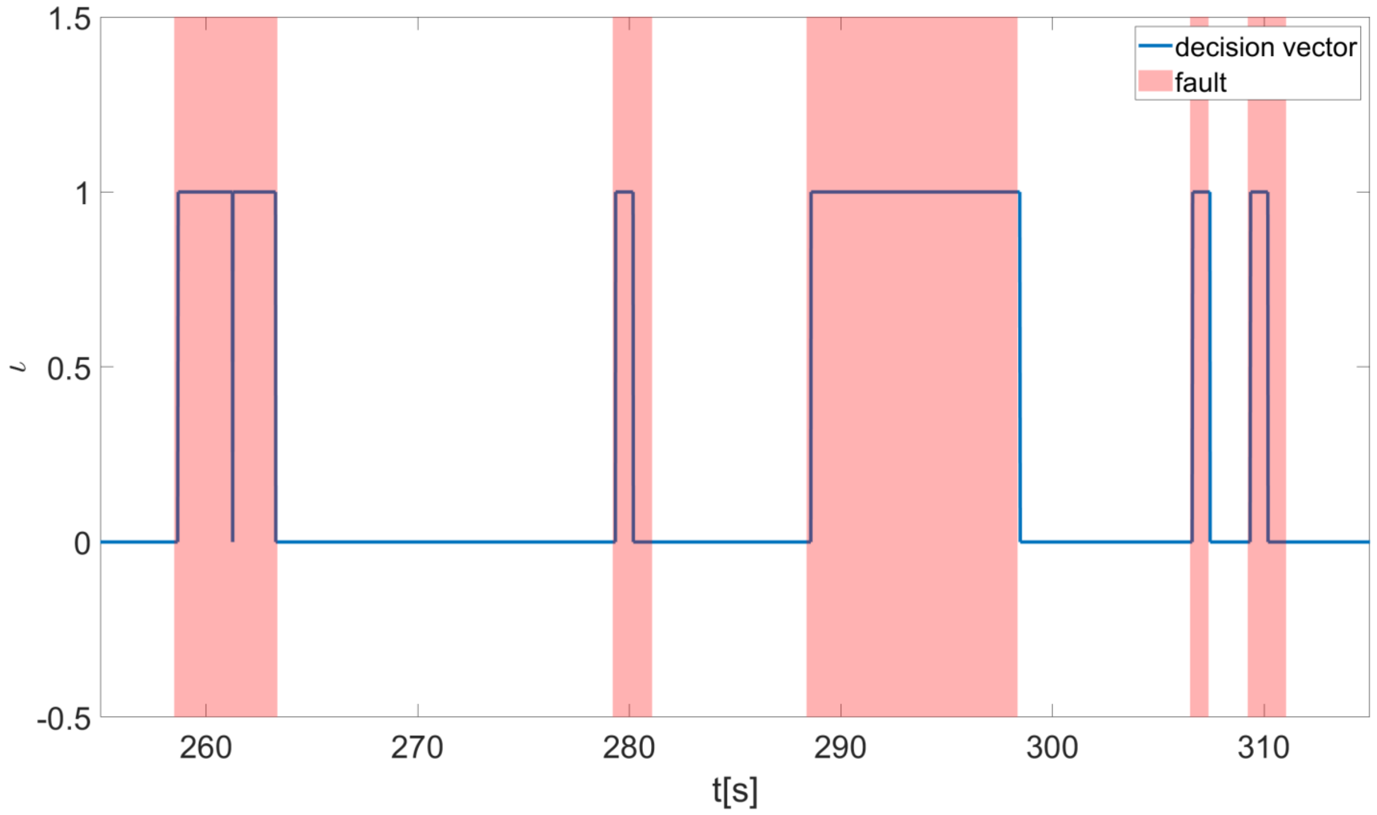


Residual generator

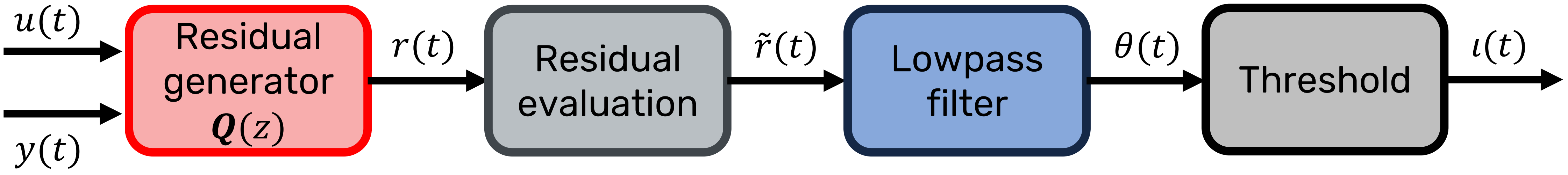


Residual generator





Residual generator



$\boldsymbol{Q}(z)$

$\tilde{r}(t) = |r(t)|$

$$H(z) = \frac{0.1139}{z - 0.8861}$$

$$\omega_t = 9.38 \text{ rad/s}$$

$\tau = 1.55$

The cut frequency of the filter and the length of the parity vector were **optimized** by minimizing a cost function, over a certain time, of the type:

$$J(\omega_t, s) = \text{false\_alarms}(\omega_t, s) \cdot \alpha + \text{missed\_alarms}(\omega_t, s) \cdot \beta$$

References

1. Ding, Steven X. “Model-based fault diagnosis techniques: design schemes, algorithms, and tools”. Springer Science & Business Media, 2008.
2. Andreas Varga “Solving Fault Diagnosis Problems - Linear Synthesis Techniques”, vol. 84 of Studies in Systems, Decision and Control, Springer International Publishing, 2017, xxviii+394 pages, hardcover: ISBN 978-3-319-51558-8, ebook: 978-3-319-51559-5.