Model-Based and Intelligent Fault Detection and Isolation: Residual Generation, Observers and Industrial Case Studies

Silvio Simani

Email: silvio.simani@unife.it

URL: www.silviosimani.it/talks.html

28/11/2025

TALK STRUCTURE

- Definitions
- Methods of residual generation
- Fault detectability and isolability
- Fault diagnosis using:
  - (Dynamic Observers) UIO
  - Kalman Filters
  - Neural Networks
- Case Studies
- Concluding discussion

Definitions
Residual Generation Method
Fault Detectability and Isolability

Need for Fault Diagnosis

- Safer
- Reliable
- Avoid system shutdown, breakdown
- Catastrophes

involving human fatalities and material damage

Monitoring

- Continuous real-time task
- Determination of the conditions of a physical system
- By recording information, recognising and indicating anomalies in the behaviour

PROBLEM STATEMENT

- ***What is a fault?***

- ***An unexpected change in a system, such as a component malfunction and variations in operating condition, that tend to degrade overall system performance***

- **We use the term "fault" rather than "failure" to denote a malfunction rather than a catastrophe. The term failure suggests a complete breakdown, whilst a fault may denote something tolerable**

Fault Classification

- **Classification according to location at which each fault affects the system:**

  - **Sensor fault**
  - **Actuator fault (valve, motor)**
  - **Component fault (intermediate valve)**

Classification by Fault Type

- bias
- drift
- slow varying fault
- abrupt changes
- stochastic faults

<u>Note</u>: Faults are distinguished from noise

Noise can be considered in advance, but faults are "unexpected".
Noise is often tolerable, but faults are not

Incipient Faults

Why do we need fault diagnosis?

- **Early indication of *incipient faults* can help avoid major plant breakdowns and catastrophes**

- **Fault detection and isolation has become a critical issue in the operation of high-integrity and fault-tolerant systems**

Tasks of Fault Diagnosis

- **Fault detection: i.e., the indication that something is going wrong in the system**
- **Fault isolation: i.e., the determination of the exact location of the fault**
- **Fault identification: i.e., the determination of the size and type a nature of the fault**
- **Fault accommodation: i.e., the reconfiguration of the system using healthy components**

- **We focus attention on Fault Detection and Isolation (FDI)**

Traditional Approaches to Fault Diagnosis

Installation of multiple sensors?

(“physical redundancy” or “hardware redundancy!”):

- **Back up safety-critical hardware and software using triplex or quadruplex arrangement, *e.g. as in “fly-by-wire” aircraft***
- **The measure is aimed especially at detecting and isolating sensor faults.**
- **Measurements of the same variable from different sensors are compared. Any serious *discrepancy* is an *indication of a fault* in at least one sensor**

Traditional Methods

Limit checking:

Plant measurement compared with preset limits; exceedance of a limit can indicate a fault situation.

Installation of special sensors:

Limit sensors – basically, performance limit-checking in hardware (e.g., limit temperature or pressure) or measuring some special variables (sound, efficiency, vibration, ...)

Frequency spectrum analysis:

Some plant measurements have a typical frequency spectrum under normal operating conditions; any deviation from this is an indication of abnormality.

Characteristic signature used to isolate faults.

MODEL-BASED APPROACHES TO FAULT DIAGNOSIS

Analytical (Functional) Redundancy

- **Makes explicit use of mathematical model of system, often known as the "model-based" approach.**
- **Computer implementation allows the development of fault detection, fault isolation and fault identification based on analytical rather than hardware redundancy.**
- **We now give a more detailed treatment of this approach.**

Functional (Analytical) Redundancy

Provides independent way of detecting and accommodating faults

- **Requires a “functional” model - either “explicit” or “implicit”**
- **Suffers from system non-linearity and modelling errors**
- **Sometimes saves on weight, space, cost**

E.g. Dynamic Observers, UIO and Kalman filters

Knowledge-based Approach

- Makes explicit use of human knowledge of fault and qualitative reasoning.

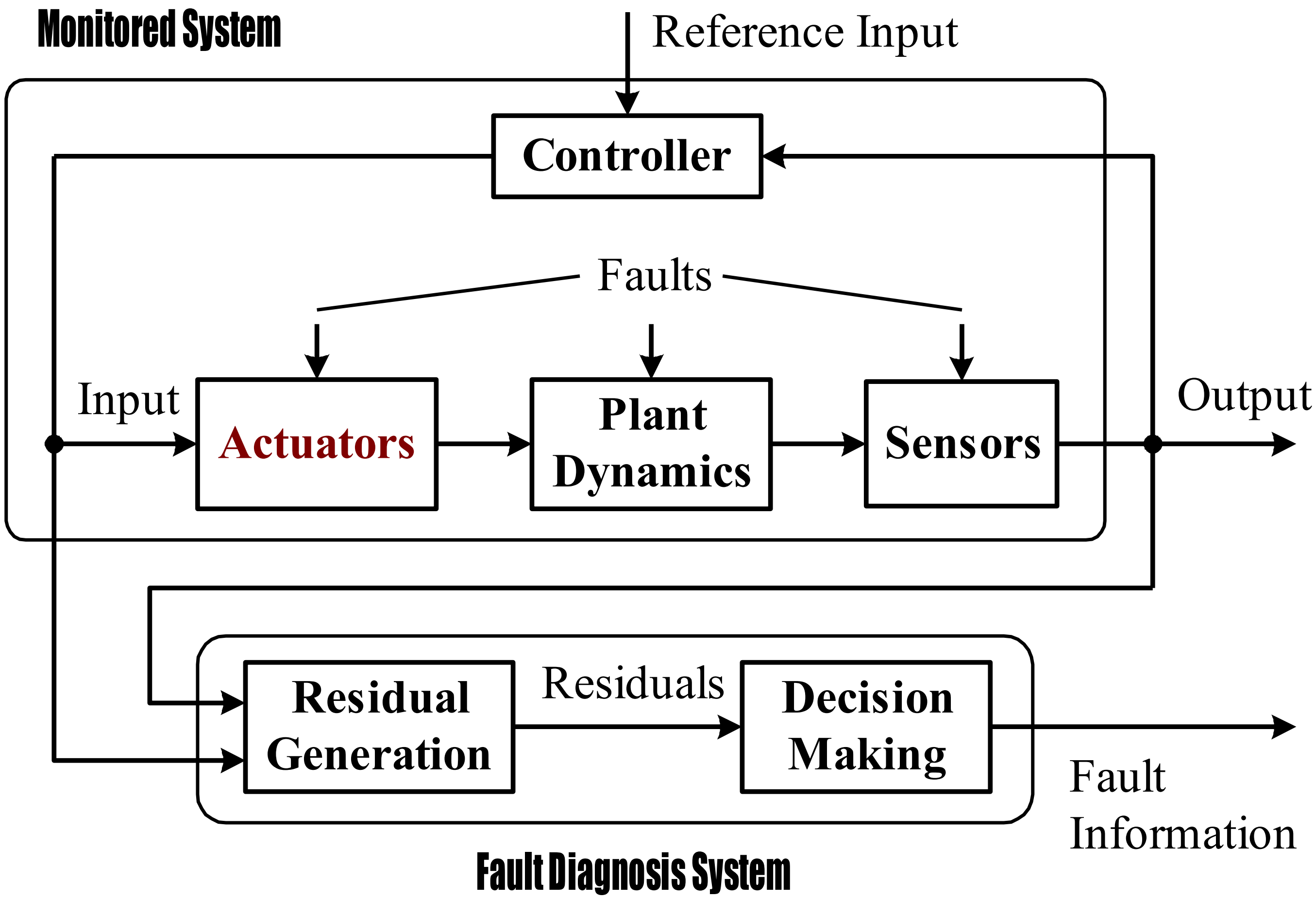
- Provides method of combining numeric and symbolic models for performing the FDI task.

- **Combination of all above approaches:**

  EXPERT SYSTEM FOR FDI

E.g. Neural Networks (and Fuzzy Systems)

Basic Principles of Model-Based Fault Diagnosis



Two-Stages of Model-Based FDI

- **Residual generation**
  - **Generate residual signals which are <u>indicators of faults</u>.**

- **Decision making**
  - **Make decisions based on residuals and decision rules.**

STRUCTURE OF RESIDUAL GENERATION

- **Residuals are INDICATORS OF FAULTS that are <u>independent of the system operating state</u> under nominal conditions.**

  - **Zero** **for fault-free case**
  - **Non-zero** **for faulty case**

  $\underline{r} \neq 0$ **if and only if** $\underline{f} \neq 0$

Decision rule:

$J(\underline{r}) \leq J_{th}$ **for** $\underline{f}(t) = 0$

$J(\underline{r}) > J_{th}$ **for** $\underline{f}(t) \neq 0$

where $J(\underline{r})$ **residuals evaluation function**

J_{th} **threshold**

FAULT ISOLATION

- **Fault Isolability:** ***Fault distinguishable from other faults using residual set (or residual vector).***
- **Structured residual set:** ***Residual set which has required sensitivity to specific faults and insensitivity to other faults.***

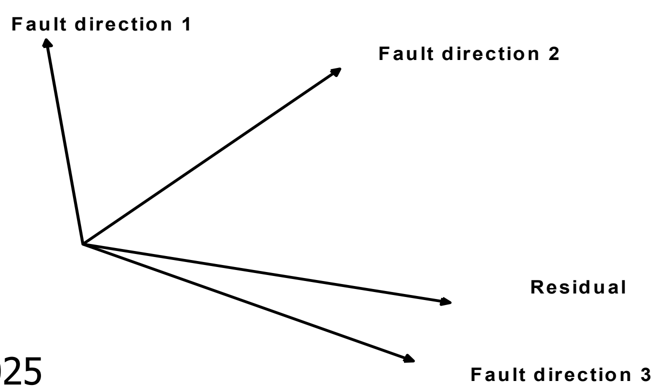
Method 1

	r_1	r_2	r_3
f_1	0	1	1
f_2	1	0	1
f_3	1	1	0

Method 2

	r_1	r_2	r_3
f_1	1	0	0
f_2	0	1	0
f_2	0	0	1

- **Fixed direction residual vector:**



Residual Generation Based on (State) Output Estimation

State estimation approach ***restricted*** **to on-line reconstruction of sets (or subsets) of state variables (or measured variables) using mathematical model.**

Using observers or Kalman filters whose estimation error vector then used for residual generation.

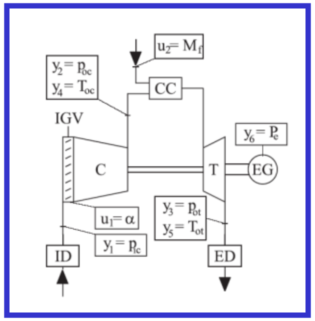
Most processes are nonlinear and linear state space models are only appropriate for small variations.

When a fault occurs, the variations can be large!!

Perhaps, multiple-models should be used - based on different points of operation.

Our Case Studies

- Simulated case studies
- Real processes
- Research works



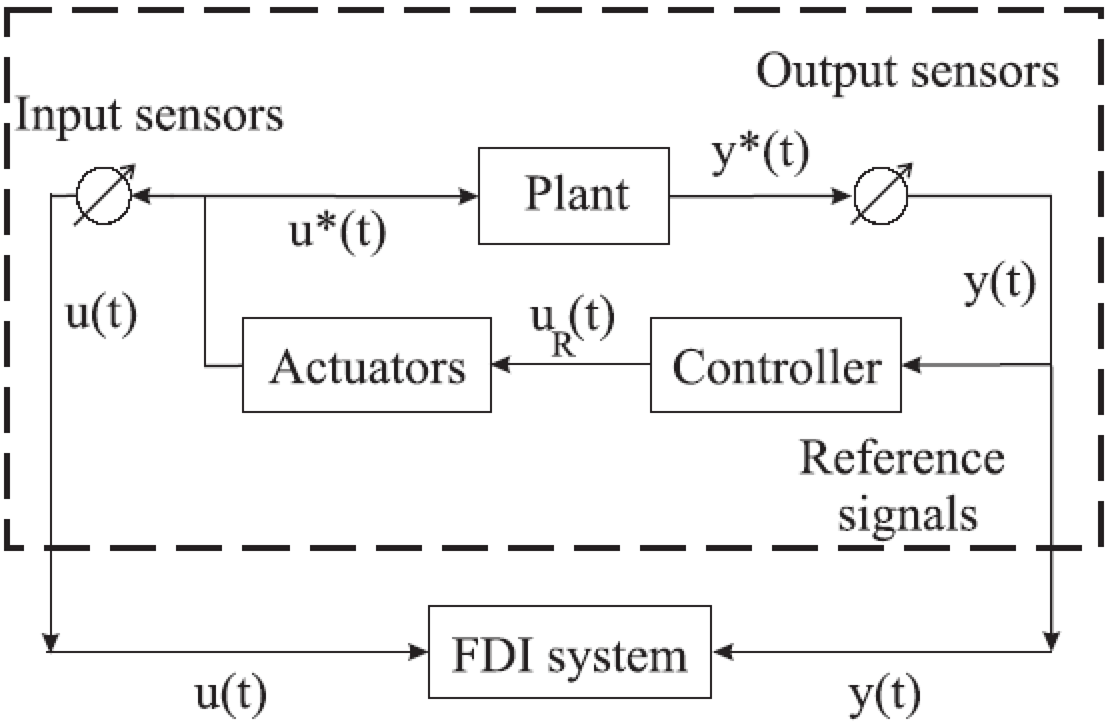
Piper Malibu

- **Aircraft/Aerospace Applications**
- **Industrial Simulated Example**

Fault and System

Models

Model-based FDI Techniques



The fault diagnosis scheme

Model-based FDI Techniques (Cont'd)

Fault and System Modelling

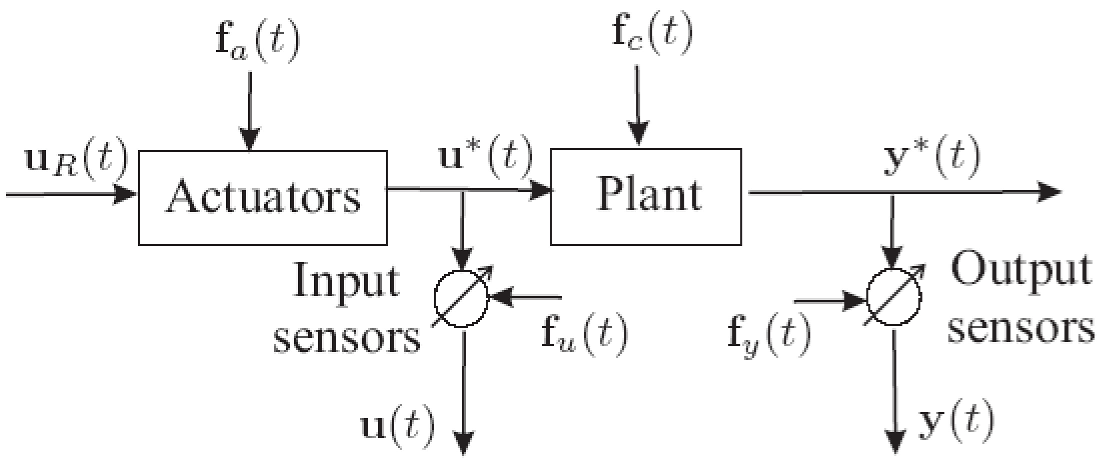


Figure 2.5: The monitored system and fault topology.

$$\begin{cases} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}^*(t) \\ \mathbf{y}^*(t) &= \mathbf{C}\mathbf{x}(t) \end{cases}$$

Model-based FDI Techniques (Cont'd)

Fault and System Modelling

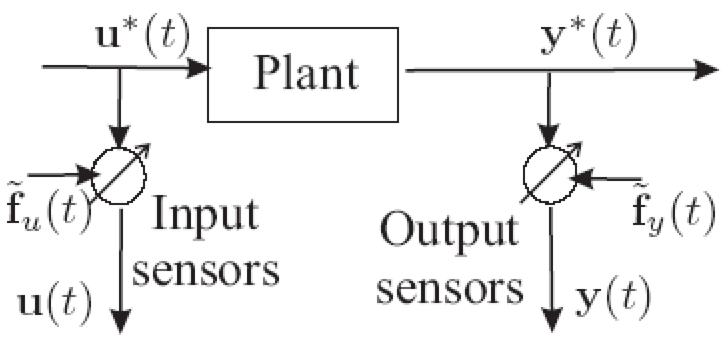


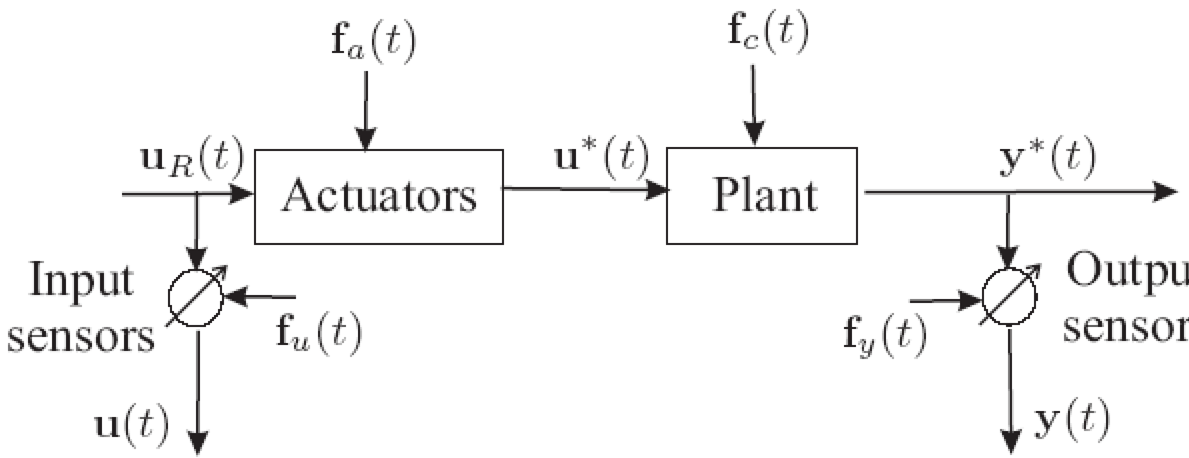
Figure 2.6: The structure of the plant sensors.

$$\begin{cases} \mathbf{u}(t) = \mathbf{u}^*(t) + \tilde{\mathbf{u}}(t) \\ \mathbf{y}(t) = \mathbf{y}^*(t) + \tilde{\mathbf{y}}(t) \end{cases}$$

$$\begin{cases} \mathbf{u}(t) = \mathbf{u}^*(t) + \tilde{\mathbf{u}}(t) + \mathbf{f}_u(t) \\ \mathbf{y}(t) = \mathbf{y}^*(t) + \tilde{\mathbf{y}}(t) + \mathbf{f}_y(t) \end{cases}$$

Model-based FDI Techniques (Cont'd)

Fault and System Modelling



$$\mathbf{u}^*(t) = \mathbf{u}_R(t) + \mathbf{f}_a(t)$$

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{f}_c(t) + B\mathbf{u}^*(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathbf{f}_y(t) \\ \mathbf{u}(t) & = & \mathbf{u}^*(t) + \mathbf{f}_u(t) \end{cases}$$

Model-based FDI Techniques (Cont'd)

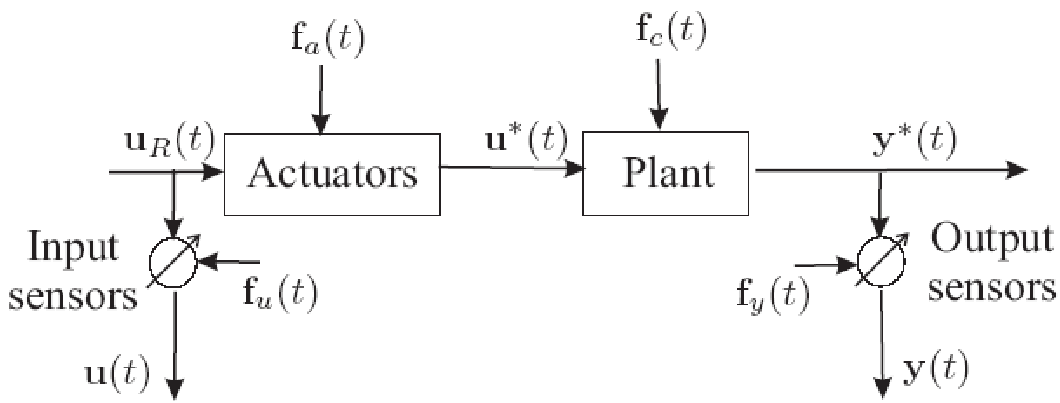


Figure 2.7: Fault topology with actuator input signal measurement.

- Modelling of Faulty Systems

$$\mathbf{f}(t) = [\mathbf{f}_a^T,\ \mathbf{f}_u^T,\ \mathbf{f}_c^T,\ \mathbf{f}_y^T]^T \in \Re^k$$

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{f}_c(t) + \mathbf{B}\mathbf{f}_a(t) + \mathbf{B}\mathbf{u}^*(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathbf{f}_y(t) \\ \mathbf{u}(t) & = & \mathbf{u}^*(t) + \mathbf{f}_u(t) \end{cases}$$

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}^*(t) + \mathbf{L}_1\mathbf{f}(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathbf{L}_2\mathbf{f}(t) \\ \mathbf{u}(t) & = & \mathbf{u}^*(t) + \mathbf{L}_3\mathbf{f}(t) \end{cases}$$

Residual Generation

Residual Evaluation

Residual Generator Structure

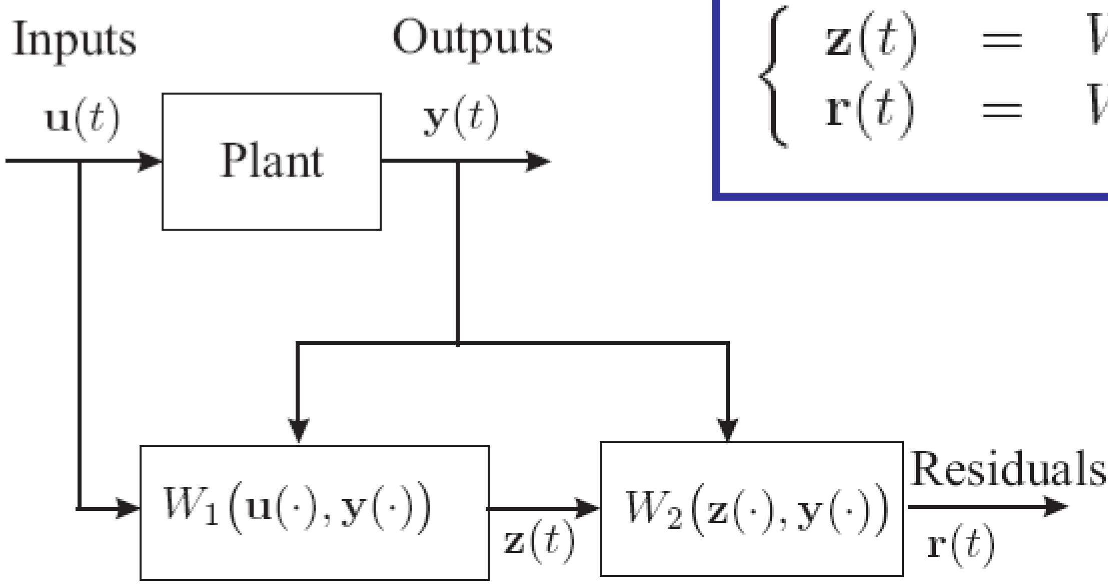


Figure 2.8: Residual generator general structure.

$$\begin{cases} \mathbf{z}(t) & = & W_1(\mathbf{u}(\cdot), \mathbf{y}(\cdot)) & & \\ \mathbf{r}(t) & = & W_2(\mathbf{z}(\cdot), \mathbf{y}(\cdot)) & = & \mathbf{0}. \end{cases}$$

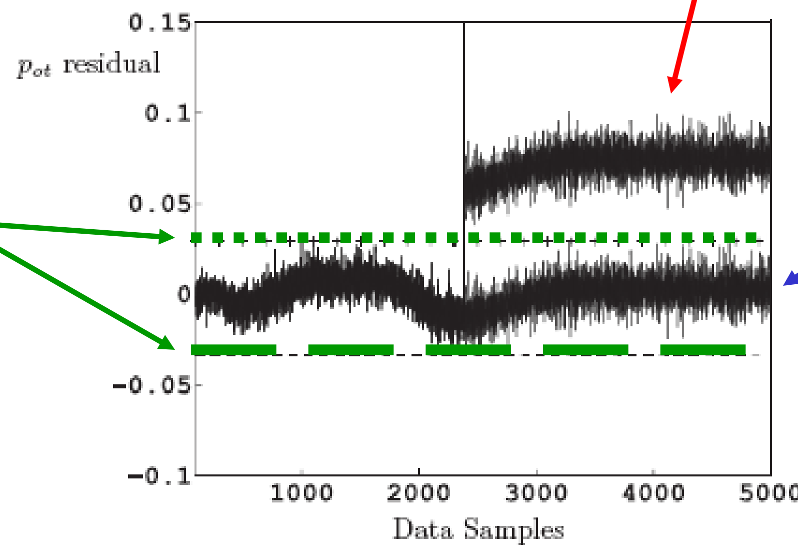
Under fault-free assumptions, the residual signal *r(t)* is "almost zero"

General Residual Evaluation

$$\begin{cases} J\left(\mathbf{r}(t)\right) \leq \varepsilon(t) & \text{for} \quad \mathbf{f}(t) = 0 \\ J\left(\mathbf{r}(t)\right) > \varepsilon(t) & \text{for} \quad \mathbf{f}(t) \neq 0 \end{cases}$$

Faulty residual

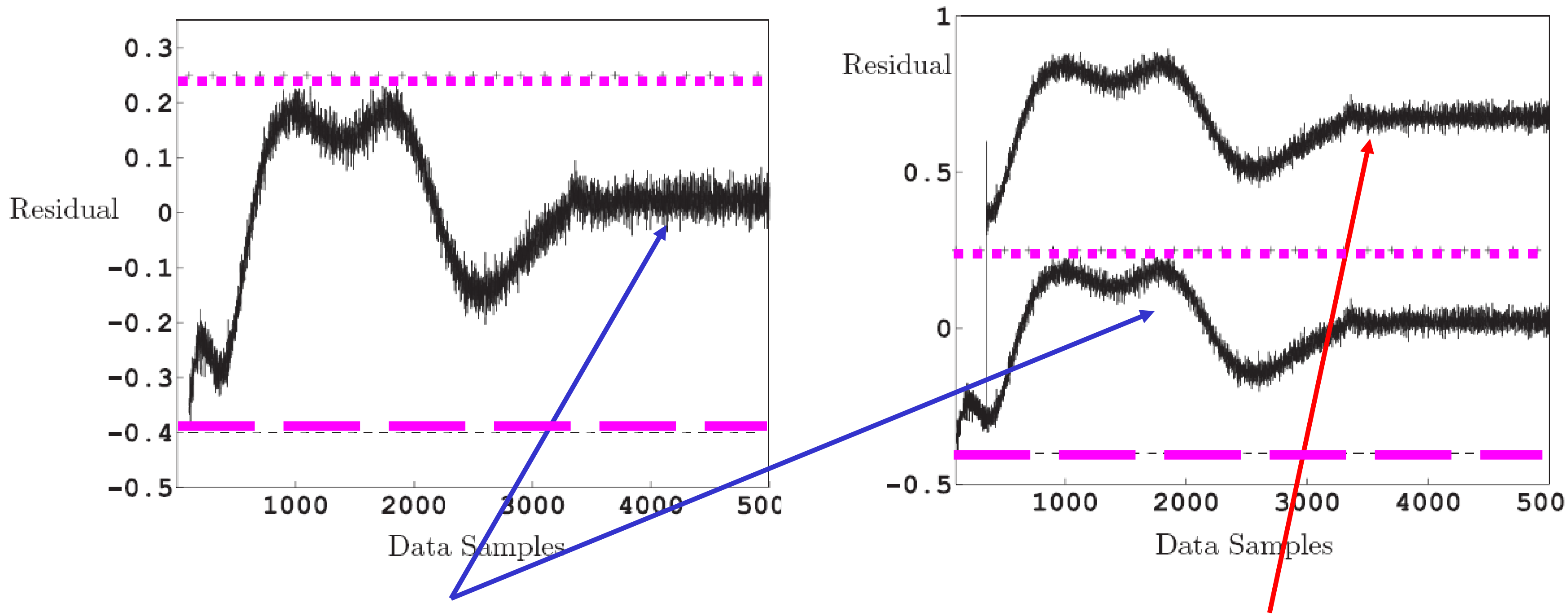
Detection thresholds
ε(t)



Fault free residual

General Residual Evaluation *(example)*

Detection thresholds



Fault free residual **Fault-free & faulty residuals**

Change Detection & Residual Evaluation

$$\left\{ \begin{array}{llll} J\left(\mathbf{r}(t)\right) \leq \varepsilon(t) & \text{for} & \mathbf{f}(t) = 0 \\ J\left(\mathbf{r}(t)\right) > \varepsilon(t) & \text{for} & \mathbf{f}(t) \neq 0 \end{array} \right.$$

$$\boxed{J\left(r(t)\right) \equiv \left|r(t)\right|}$$

Detection thresholds

ε(t)

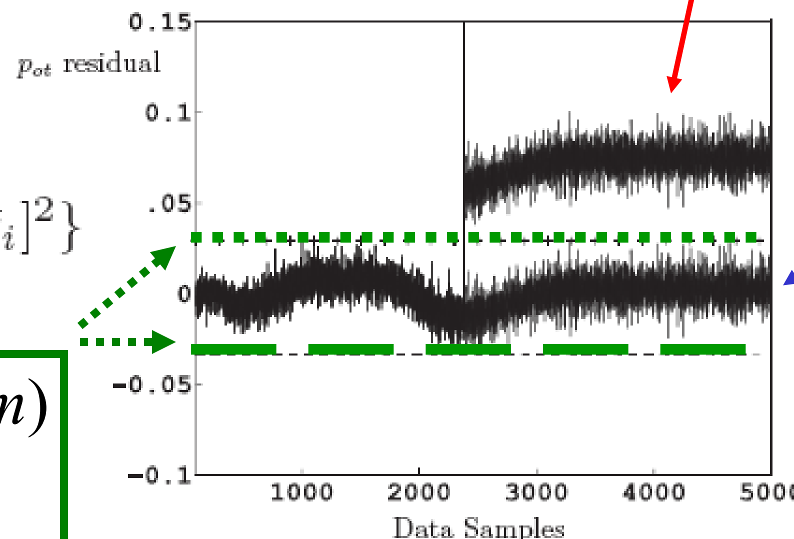
$$\bar{r}_i = E\{r_i(t)\}; \qquad \bar{\sigma}_i^2 = E\{[r_i(t) - \bar{r}_i]^2\}$$

$$\varepsilon(t) = \bar{r}_i \pm \delta \times \bar{\sigma}_i \qquad (i = 1, \cdots, m)$$

with $\delta \geq 2$

Faulty residual

Fault free residual



Unknown Input Observers

Unknow Input Observer (UIO)

Disturbance Distribution Matrix (Known)

$$\left.\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{d}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned}\right\}$$

System with disturbance "*Unknown Input*"

Unknown Input

Definition: An observer is defined as an *Unknown Input Observer* for the system with disturbance (above), if its state estimation error vector $e_x(t)$ approaches zero asymptotically, regardless of the presence of the unknown input term in the system.

UIO Model

Given:

$$\left.\begin{aligned}\mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{d}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)\end{aligned}\right\}$$

The full–order UIO has the following mathematical form

$$\left.\begin{aligned}\mathbf{z}(t+1) &= \mathbf{F}\mathbf{z}(t) + \mathbf{T}\mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{y}(t)\\ \hat{\mathbf{x}}(t) &= \mathbf{z}(t) + \mathbf{H}\mathbf{y}(t)\end{aligned}\right\}$$

where $\mathbf{z}(t) \in \Re^n$ is the state of the UIO, $\hat{\mathbf{x}}(t)$ the estimated state vector $\mathbf{x}(t)$, whilst $\mathbf{F}$, $\mathbf{T}$, $\mathbf{H}$ and $\mathbf{K}$ are matrices to be designed to achieve the unknown input de–coupling .

UIO Structure

Plant Model

$$\left.\begin{array}{lcl}\mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t)+\mathbf{E}\mathbf{d}(t)\\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t)\end{array}\right\}$$

UIO Model

$$\left.\begin{array}{lcl}\mathbf{z}(t+1) & = & \mathbf{F}\mathbf{z}(t)+\mathbf{T}\mathbf{B}\mathbf{u}(t)+\mathbf{K}\mathbf{y}(t)\\ \hat{\mathbf{x}}(t) & = & \mathbf{z}(t)+\mathbf{H}\mathbf{y}(t)\end{array}\right\}$$

Observer Design???

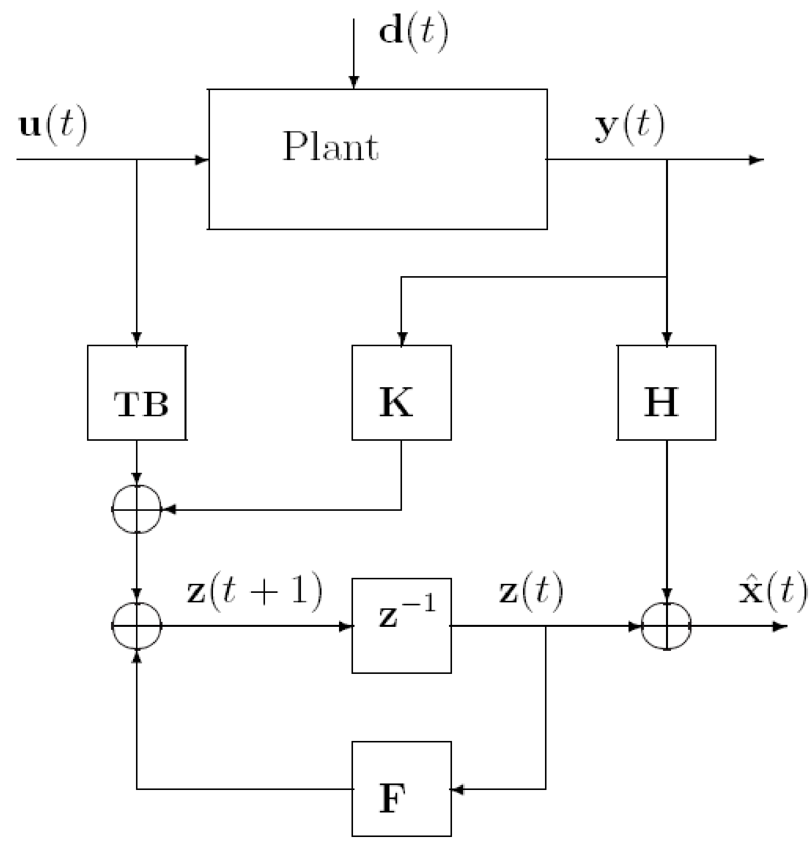


Figure 4.3: The UIO structure.

UIO Design

$$\left.\begin{array}{lcl}\mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t)+\mathbf{E}\mathbf{d}(t)\\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t)\end{array}\right\}$$

Plant Model

$$\left.\begin{array}{lcl}\mathbf{z}(t+1) & = & \mathbf{F}\mathbf{z}(t)+\mathbf{T}\mathbf{B}\mathbf{u}(t)+\mathbf{K}\mathbf{y}(t)\\ \hat{\mathbf{x}}(t) & = & \mathbf{z}(t)+\mathbf{H}\mathbf{y}(t)\end{array}\right\}$$

UIO Model

$$e_x(t+1)=x(t+1)-\hat{x}(t+1)$$

State estimation error

$$\boxed{\begin{array}{lcl}\mathbf{e}_x(t+1) & = & [\mathbf{A}-\mathbf{HCA}-\mathbf{K}_1\mathbf{C}]\mathbf{e}_x(t)+\left[\mathbf{F}-(\mathbf{A}-\mathbf{HCA}-\mathbf{K}_1\mathbf{C})\right]\mathbf{z}(t)\\ & + & \left[\mathbf{K}_2-(\mathbf{A}-\mathbf{HCA}-\mathbf{K}_1\mathbf{C})\right]\mathbf{y}(t)\\ & + & \left[\mathbf{T}-(\mathbf{I}-\mathbf{HC})\right]\mathbf{B}\mathbf{u}(t)+(\mathbf{HC}-\mathbf{I})\mathbf{E}\mathbf{d}(t)\end{array}}$$

UIO Design (Cont'd)

$$
\begin{aligned}
\mathbf{e}_x(t+1) &= [\mathbf{A} - \mathbf{HCA} - \mathbf{K}_1\mathbf{C}]\mathbf{e}_x(t) + \big[\mathbf{F} - (\mathbf{A} - \mathbf{HCA} - \mathbf{K}_1\mathbf{C})\big]\mathbf{z}(t) \\
&+ \big[\mathbf{K}_2 - (\mathbf{A} - \mathbf{HCA} - \mathbf{K}_1\mathbf{C})\big]\mathbf{y}(t) \\
&+ \big[\mathbf{T} - (\mathbf{I} - \mathbf{HC})\big]\mathbf{Bu}(t) + (\mathbf{HC} - \mathbf{I})\mathbf{Ed}(t)
\end{aligned}
$$

$$
\left.
\begin{aligned}
\mathbf{z}(t+1) &= \mathbf{Fz}(t) + \mathbf{TBu}(t) + \mathbf{Ky}(t) \\
\hat{\mathbf{x}}(t) &= \mathbf{z}(t) + \mathbf{Hy}(t)
\end{aligned}
\right\}
$$

$$
\begin{aligned}
(\mathbf{HC} - \mathbf{I})\mathbf{E} &= \mathbf{0} \\
\mathbf{I} - \mathbf{HC} &= \mathbf{T} \\
\mathbf{A} - \mathbf{HCA} - \mathbf{K}_1\mathbf{C} &= \mathbf{F} \\
\mathbf{FH} &= \mathbf{K}_2
\end{aligned}
$$

the state estimation error will then be:

$$\mathbf{e}_x(t+1) = \mathbf{Fe}_x(t).$$

UIO for Fault (Detection) + Isolation

$$\left.\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{E}\mathbf{d}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) \end{aligned}\right\}$$

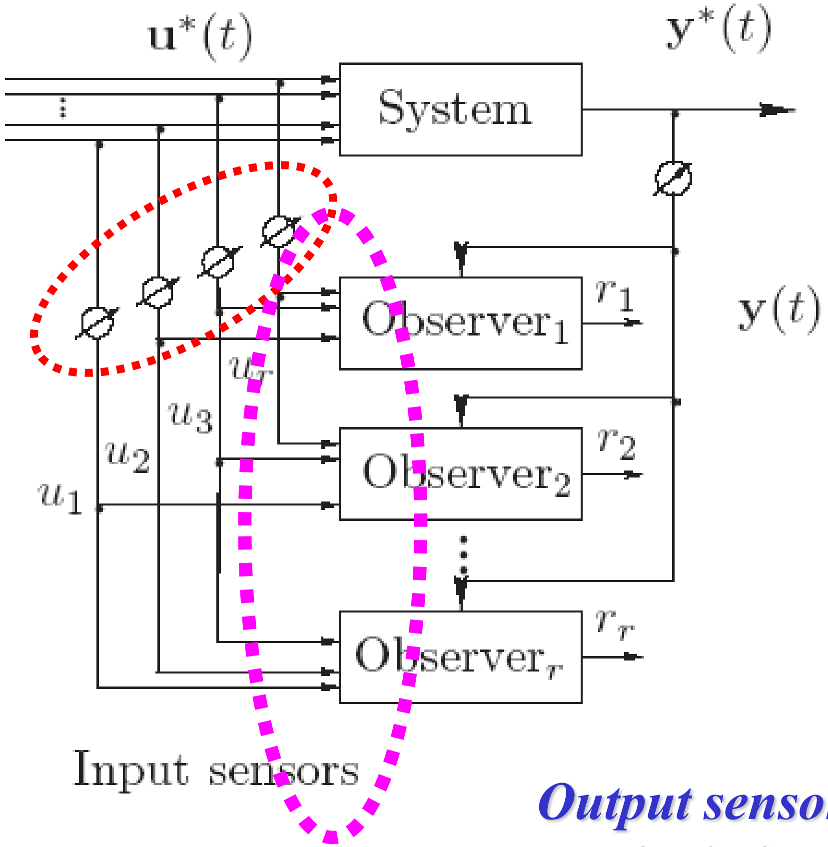
General system

$$\left.\begin{aligned} \mathbf{z}(t+1) &= \mathbf{F}\mathbf{z}(t) + \mathbf{T}\mathbf{B}\mathbf{u}(t) + \mathbf{K}\mathbf{y}(t) \\ \hat{\mathbf{x}}(t) &= \mathbf{z}(t) + \mathbf{H}\mathbf{y}(t) \end{aligned}\right\}$$

UIO model

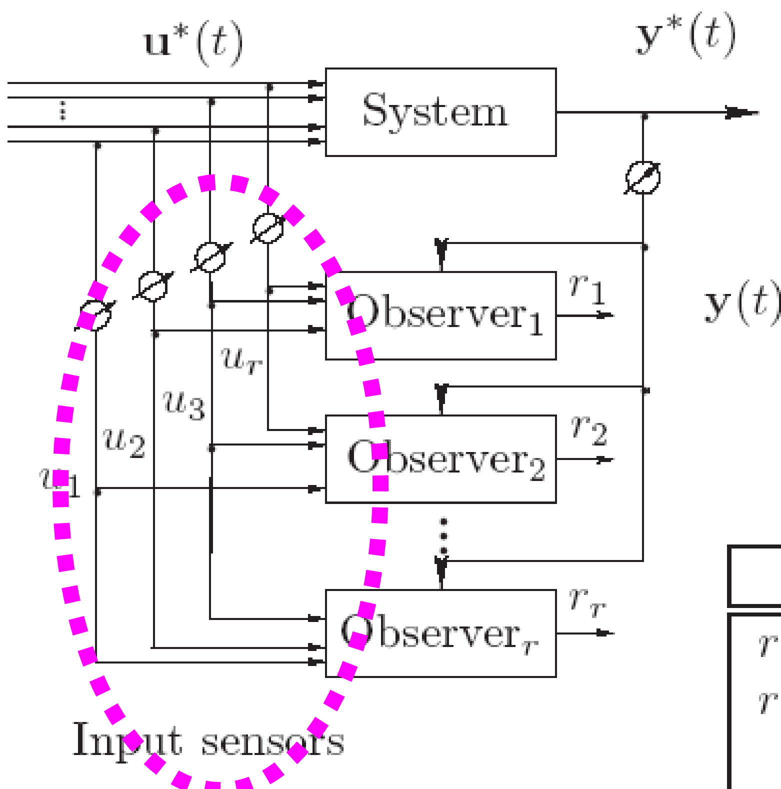
$$\left.\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{B}\mathbf{f}_u(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{f}_y(t) \end{aligned}\right\}$$

Process with input faults



Output sensors are fault-free

Input Fault Isolation with UIO



Each observer is insensitive to one input sensor:

Table 4.1: Fault signatures.

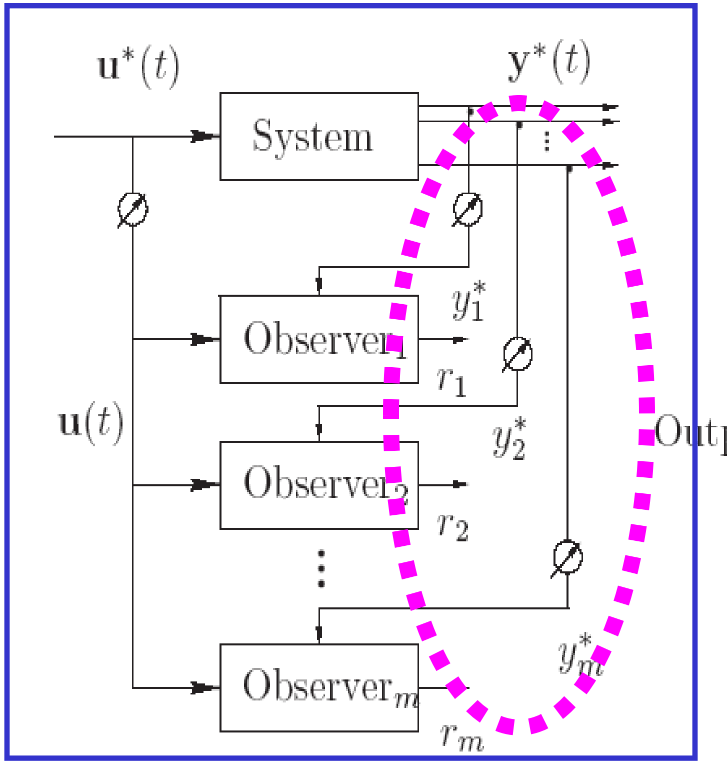
	u_1	u_2	$\dots$	u_r	y_1	y_2	$\dots$	y_m
r_{UIO_1}	0	1	$\dots$	1	1	1	$\dots$	1
r_{UIO_2}	1	0	$\dots$	1	1	1	$\dots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{UIO_r}	1	1	$\dots$	0	1	1	$\dots$	1

UIO for *Output Fault* Isolation

Table 4.1: Fault signatures.

	u_1	u_2	$\dots$	u_r	y_1	y_2	$\dots$	y_m
r_{O_1}	1	1	$\dots$	1	1	0	$\dots$	0
r_{O_2}	1	1	$\dots$	1	0	1	$\dots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{O_m}	1	1	$\dots$	1	0	0	$\dots$	1

Bank of output observers



Fault-free case:

$$\lim_{t\to\infty} r_i(t) = \lim_{t\to\infty} \left(y_i(t) - C^i\mathbf{x}^i(t)\right) = 0$$

Faulty case

$$y_i(t) = y_i^*(t) + f(t)$$

$$\lim_{t\to\infty} r_i(t) \neq 0$$

Residual Disturbance *Robustness*

- Residuals decoupled from disturbance
- Robust residual generator
- Disturbance effect minimisation
- Measurement errors

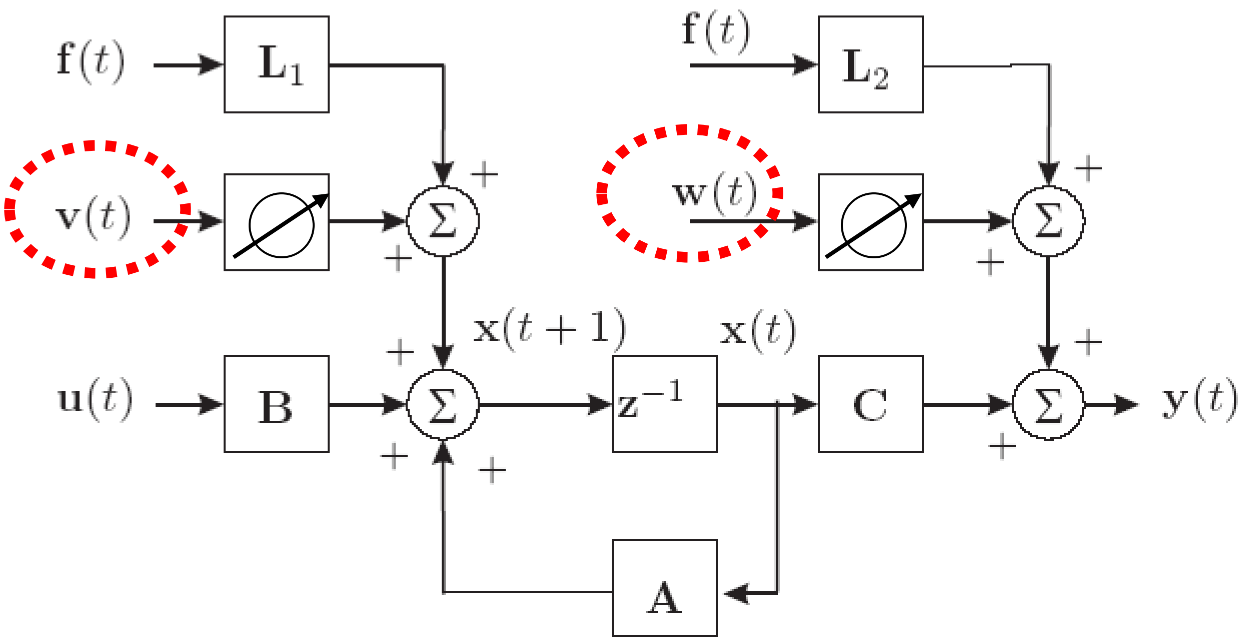


Figure 2.14: MIMO process with faults and noises.

FDI with *Noisy Measurements*

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w(t)} \quad + \mathbf{L}_1\mathbf{f}(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathrm{v(t)} \quad + \mathbf{L}_2\mathbf{f}(t) \end{cases}$$

✓ Model with fault and noise

$$\boxed{\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w(t)} \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathrm{v(t)} \end{cases}}$$

➢ Model with noise only: **Kalman filter!**

Kalman Filter

Idea…

What is a Kalman Filter?

- *Optimal recursive* *data fusion* algorithm
- Predictor-Corrector style algorithm
- Processes all available sensor measurements in estimating the value of parameters of interest using:
  - Knowledge of system and sensor dynamics
  - Statistical models reflecting uncertainty in system noise and sensor dynamics
  - Any information regarding initial conditions

What is a Kalman Filter (cont'd)?

- *Optimal* in the sense that for systems which can be described by a linear model, *e.g.*

$$\boxed{\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ z_k &= C\,x_k + v_k \end{aligned}}$$

  and for which the process and measurement noises w_k and v_k are *normally distributed*, the Kalman filter is the provably optimal estimator (estimate has minimum error variance)

- In our case, "process noise" corresponds to uncertainty in the process model, measurement noise is from uncertainty in the sensing model, ***x*** denotes the state being estimated and ***z*** the sensor measurements

Time Update (Predictor)

- Update expected value of **x**

$$\boxed{\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1} + \mathbf{B}\mathbf{u}_k}$$

- Update error covariance matrix **P**

$$\boxed{\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}}$$

- Previous statements were simplified versions of the same idea:

$$\boxed{\hat{x}(t_3^-) = \hat{x}(t_2) + u[t_3 - t_2]}$$

$$\boxed{\sigma^2(t_3^-) = \sigma^2(t_2) + \sigma_\varepsilon^2[t_3 - t_2]}$$

Measurement Update (Corrector)

- Update expected value

$$\boxed{\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-)}$$

  - *innovation* is $\boxed{\mathbf{z}_k - \mathbf{C}\hat{\mathbf{x}}_k^-}$

- Update error covariance matrix

$$\boxed{\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{C})\mathbf{P}_k^-}$$

- Compare with previous form

$$\boxed{\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)(z_3 - \hat{x}(t_3^-))}$$

$$\boxed{\sigma^2(t_3) = (1 - K(t_3))\sigma^2(t_3^-)}$$

The Kalman Gain

- The optimal Kalman gain $\mathbf{K}_k$ is

$$\boxed{\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^T (\mathbf{C}\,\mathbf{P}_k^-\,\mathbf{C}^T + \mathbf{R})^{-1}}$$

$$\boxed{= \frac{\mathbf{P}_k^-\,\mathbf{C}^T}{\mathbf{C}\,\mathbf{P}_k^-\,\mathbf{C}^T + \mathbf{R}}}$$

- Compare with previous form

$$\boxed{K(t_3) = \frac{\sigma^2(t_3^-)}{\sigma^2(t_3^-) + \sigma_3^2}}$$

Kalman Filter for FDI

Design…

Kalman Filter Design

$$* \quad \begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w(t)} \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathrm{v(t)} \end{cases}$$

With reference to the time–invariant, discrete–time, linear dynamic system described by Equation * the i-th KF for the i-th output has the structure [Jazwinski, 1970]:

$$\boxed{\begin{aligned} \mathbf{x}_F^i(t+1|t) &= \mathbf{A}\mathbf{x}_F^i(t|t) + \mathbf{B}\mathbf{u}(t) \\ y_F^i(t+1|t) &= \mathbf{C}_i\mathbf{x}_F^i(t+1|t) \end{aligned}}$$

(State and Output Prediction)

Kalman Filter Design (Cont'd)

$$\mathbf{P}(t+1|t) = \mathbf{A}\mathbf{P}(t|t)\mathbf{A}^T + \mathbf{Q}$$

$$\mathbf{K}_i(t+1) = \mathbf{P}(t+1|t)\mathbf{C}_i^T\left[\mathbf{C}_i\mathbf{P}(t+1|t)\mathbf{C}_i^T + \mathbf{R}\right]^{-1}$$

$$\mathbf{x}_F^i(t+1|t+1) = \mathbf{x}_F^i(t+1|t) + \mathbf{K}_i(t+1)\left[y_i(t+1) - \hat{y}_F^i(t+1|t)\right]$$

$$\begin{aligned}\mathbf{P}(t+1|t+1) \; = \; & \left[\mathbf{I} - \mathbf{K}_i(t+1)\mathbf{C}_i\right]\mathbf{P}(t+1|t)\left[\mathbf{I} - \mathbf{K}_i(t+1)\mathbf{C}_i\right]^T + \\ & +\mathbf{K}_i(t+1)\mathbf{R}\mathbf{K}_i^T(t+1).\end{aligned}$$

(Vector Updates)

The variables $\mathbf{x}_F^i(t+1|t)$ and $y_F^i(t+1|t)$ are the one step prediction of the state and of the output of the process, respectively. $\mathbf{x}_F^i(t|t)$ is the state estimation given by the filter, $\mathbf{C}_i$ the i-th row of the output distribution matrix $\mathbf{C}$, $\mathbf{P}(t+1|t)$ is the covariance matrix of the one step prediction error $\mathbf{x}(t+1) - \mathbf{x}_F^i(t+1|t)$ whilst $P(t|t)$ is the covariance matrix of the filtered state error $\mathbf{x}(t) - \mathbf{x}_F^i(t|t)$.

Kalman Filter Design (Cont'd)

$$\mathbf{P}(t+1|t) = \mathbf{A}\mathbf{P}(t|t)\mathbf{A}^T + \mathbf{Q}$$

$$\mathbf{K}_i(t+1) = \mathbf{P}(t+1|t)\mathbf{C}_i^T\left[\mathbf{C}_i\mathbf{P}(t+1|t)\mathbf{C}_i^T + \mathbf{R}\right]^{-1}$$

$$\mathbf{x}_F^i(t+1|t+1) = \mathbf{x}_F^i(t+1|t) + \mathbf{K}_i(t+1)\left[y_i(t+1) - \hat{y}_F^i(t+1|t)\right]$$

$$\mathbf{P}(t+1|t+1) = \left[\mathbf{I} - \mathbf{K}_i(t+1)\mathbf{C}_i\right]\mathbf{P}(t+1|t)\left[\mathbf{I} - \mathbf{K}_i(t+1)\mathbf{C}_i\right]^T + \\ +\mathbf{K}_i(t+1)\mathbf{R}\mathbf{K}_i^T(t+1).$$

(Vector Updates)

$\mathbf{Q}$ is the covariance matrix of the input vector noise $\tilde{\mathbf{u}}(t)$ and $\mathbf{R}$ is the variance of the i-th component of the output noise $\tilde{\mathbf{y}}(t)$. $\mathbf{K}_i(t+1)$ is the time-variant gain of the filter and $y_i(t)$ is the i-th component of the measured output $\mathbf{y}(t)$.

$$\begin{cases} \mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{Q}\mathbf{v}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{R}\mathbf{w}(t) \end{cases}$$

Kalman Filtering for FDI

$$
* \quad \begin{cases} \mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w}(t) + \mathbf{L}_1\mathbf{f}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathrm{v}(t) + \mathbf{L}_2\mathbf{f}(t) \end{cases}
$$

1) It can be proved that the innovation $e_i(t+1) = y_i(t+1) - y_F^i(t+1|t) = y_i(t+1) - \mathbf{C}_i\mathbf{x}_F^i(t+1|t)$ is a zero-mean white process when all the assumptions regarding the system * and the statistical characteristics of the noises described by Equation * are completely fulfilled. A Riccati equation is ob-

2) In the presence of a fault on the i-th output $(f_{y_i}(t) \neq 0)$, the stochastic properties (mean–value, variance and whiteness, etc) of the innovation process $e_i(t)$ change abruptly so that the fault detection can be based on these variations [Basseville, 1988].

3) Finally, note how multiple faults in outputs can be isolated since a fault on the i-th output affects only the innovation of the KF driven by the i-th output and all the innovation of the filters with unknown input.

Kalman Filtering for FDI (Cont'd)

$$\begin{cases} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w}(t) + \mathbf{L}_1\mathbf{f}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathrm{v}(t) + \mathbf{L}_2\mathbf{f}(t) \end{cases}$$

$$\begin{cases} \mathbf{x}(t+1) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathrm{v}(t) \end{cases}$$

Innovation $\boxed{e_i(t+1) = y_i(t+1) - y_F^i(t+1) = y_i(t+1) - C_i\, x_F^i(t+1 \mid t)}$

(i) Because of the linear property of the identified model and because of the additive effect of the faults on the system, it may easily be shown that the effect of the change on the innovation is also additive.

(ii) Any abrupt change in measurements due to a fault is reflected in a change in the mean value and in the standard deviation of innovations.

Kalman Filtering for FDI (Cont'd)

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w}(t) \quad + \mathbf{L}_1\mathbf{f}(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathrm{v}(t) \quad + \mathbf{L}_2\mathbf{f}(t) \end{cases}$$

$$\begin{cases} \mathbf{x}(t+1) & = & \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathrm{w}(t) \\ \mathbf{y}(t) & = & \mathbf{C}\mathbf{x}(t) + \mathrm{v}(t) \end{cases}$$

Innovation $\boxed{e_i(t+1) = y_i(t+1) - y_F^i(t+1) = y_i(t+1) - C_i\, x_F^i(t+1 \mid t)}$

In particular, since the KF produces zero–mean and independent white residuals with the system in normal operation, a method for FDI consists of testing how much the sequence of innovations has deviated from the white noise hypothesis.

KF Residual Evaluation...

Which thresholds???

$$\begin{cases} J\left(\mathbf{r}(t)\right) \leq \varepsilon(t) & \text{for} \quad \mathbf{f}(t) = \mathbf{0} \\ J\left(\mathbf{r}(t)\right) > \varepsilon(t) & \text{for} \quad \mathbf{f}(t) \neq \mathbf{0} \end{cases}$$

Innovation

$$\boxed{e_i(t+1) = y_i(t+1) - y_F^i(t+1) = y_i(t+1) - C_i\, x_F^i(t+1\,|\,t)}$$

Kalman Filtering for FDI (Cont'd)

$$\boxed{r(t) \equiv e_i(t+1) = y_i(t+1) - y_F^i(t+1) = y_i(t+1) - C_i\, x_F^i(t+1 \mid t)}$$

Innovation or Residual r(t)

(i) Statistical Tests

Zero-mean

$$\overline{r}(t) = E[r(t)] = \frac{1}{t}\sum_{j=1}^{t} r(j)$$

&

variance

$$\sigma_r^2(t) = E[r^2(t)] = \frac{1}{t}\sum_{j=1}^{t} r^2(j)$$

Kalman Filtering for FDI (Cont'd)

$$\boxed{r(t) \equiv e_i(t+1) = y_i(t+1) - y_F^i(t+1) = y_i(t+1) - C_i\, x_F^i(t+1\,|\,t)}$$

Innovation or Residual r(t)

(ii) Statistical Tests

Whiteness test

$$R_\tau^t(\tau) = \frac{1}{t}\sum_{j=1}^{t} r(j)r(j+\tau),$$

χ^2 − type

$$\zeta_\tau^M(t) = \frac{t}{R_\tau^t(0)^2}\sum_{\tau=1}^{M}\left(R_\tau^t(\tau)\right)^2$$

which are computed in a growing window. The parameter $\zeta_\tau^M(t)$ is a chi–squared random variable with M degrees of freedom.

Kalman Filtering for FDI (Cont'd)

Whiteness test

$$R_r^t(\tau) = \frac{1}{t}\sum_{j=1}^{t} r(j)r(j+\tau),$$

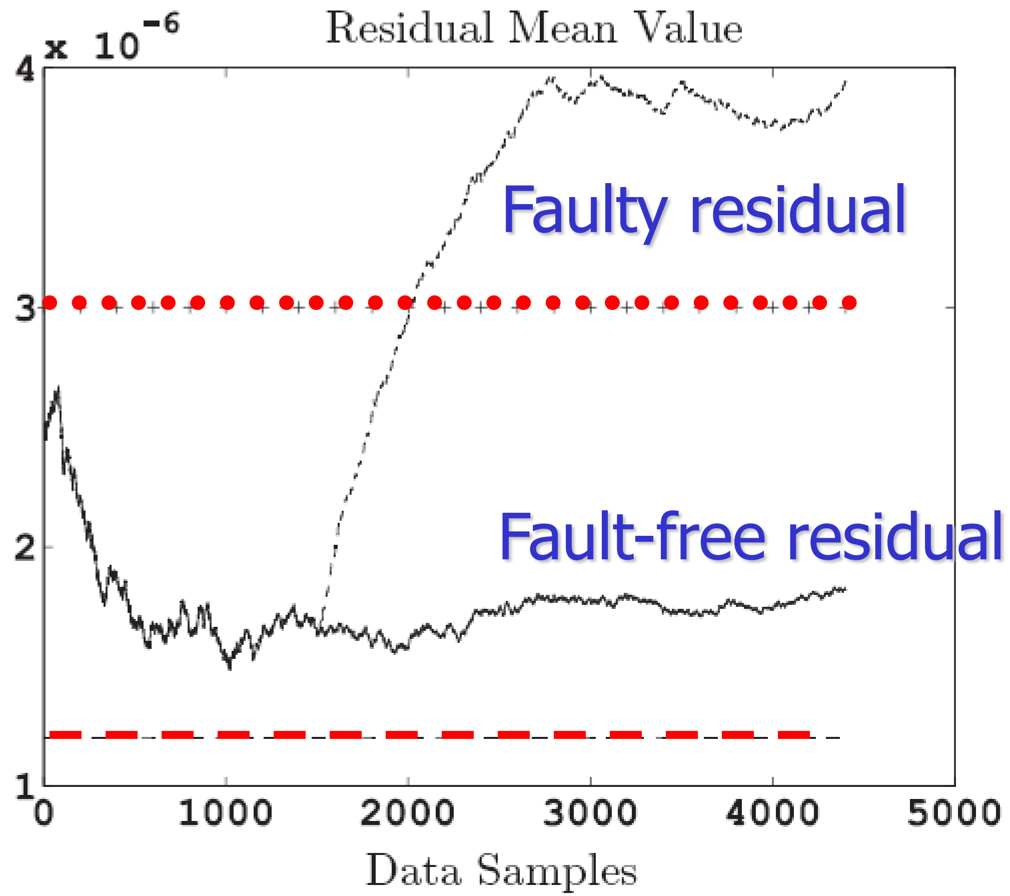
χ^2 – type

$$\zeta_r^M(t) = \frac{t}{R_r^t(0)^2}\sum_{\tau=1}^{M}\left(R_r^t(\tau)\right)^2$$

If a system abnormality occurs, the statistics of $r(t)$ change, so the comparison of $\overline{r}(t)$ and $\zeta_r^M(t)$ with a threshold ϵ fixed under no faults conditions, becomes the detection rule . In particular, such a threshold can be settled as previously seen or, with the aid of chi–squared tables, $\epsilon = \chi_\beta^2(M)$ can be computed as a function of the false–alarms probability β and of the window size M.

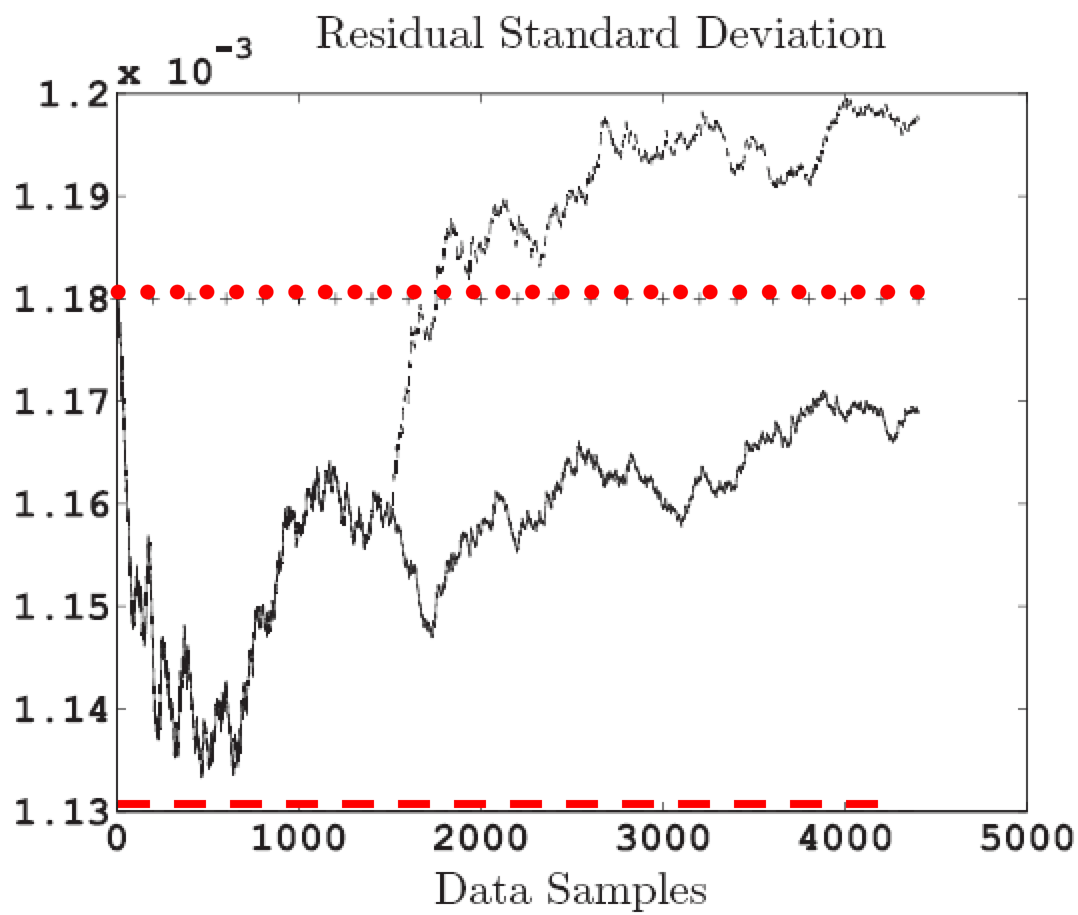
KF Residuals: Mean-value (example)

Fault-free & faulty residuals



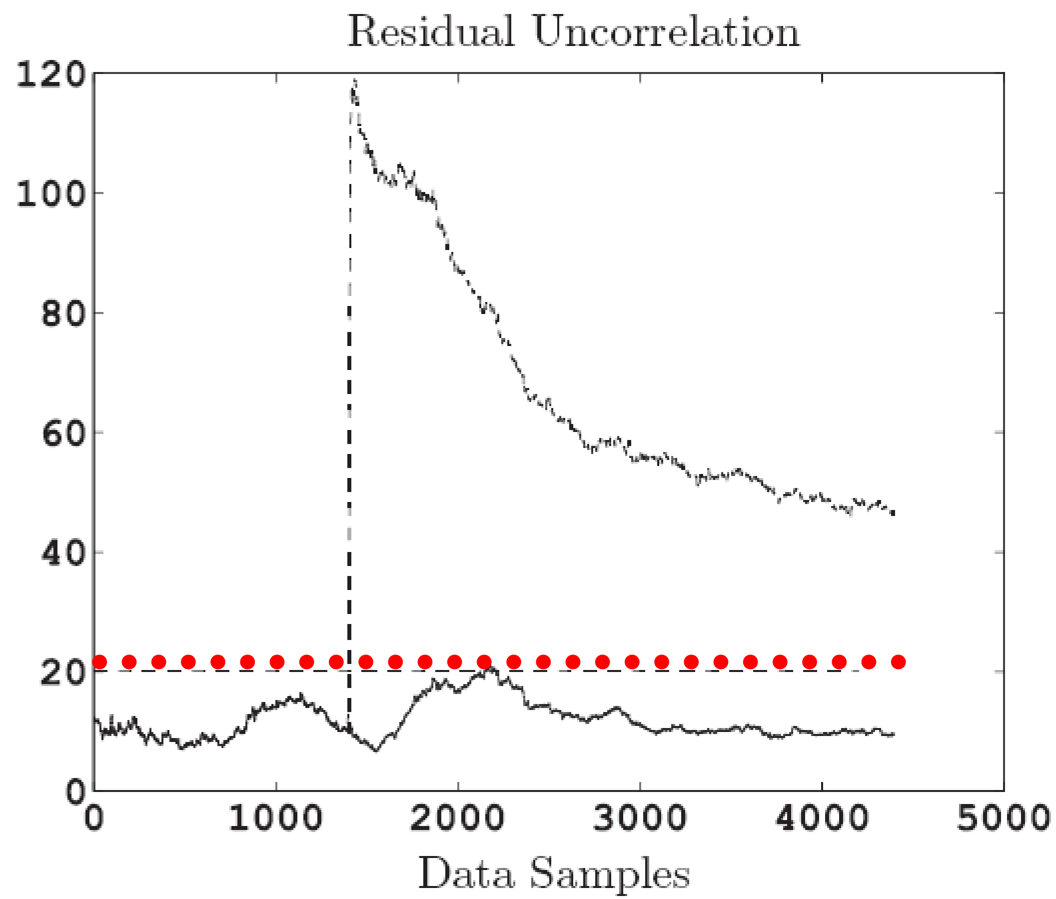
KF Residuals: Standard deviation (example)

Fault-free & faulty residuals



KF Residuals: Whiteness test (example)

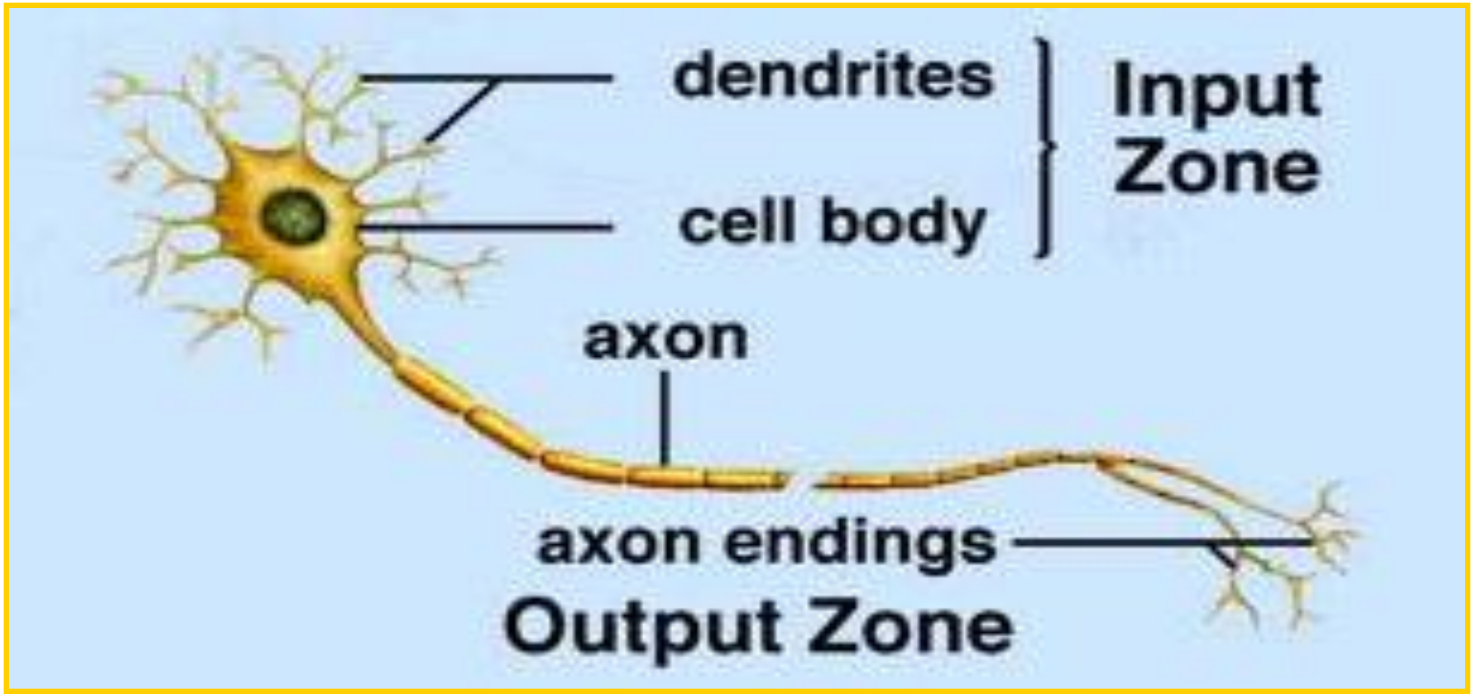
Fault-free & faulty residuals



Neural Networks for FDI

Brain

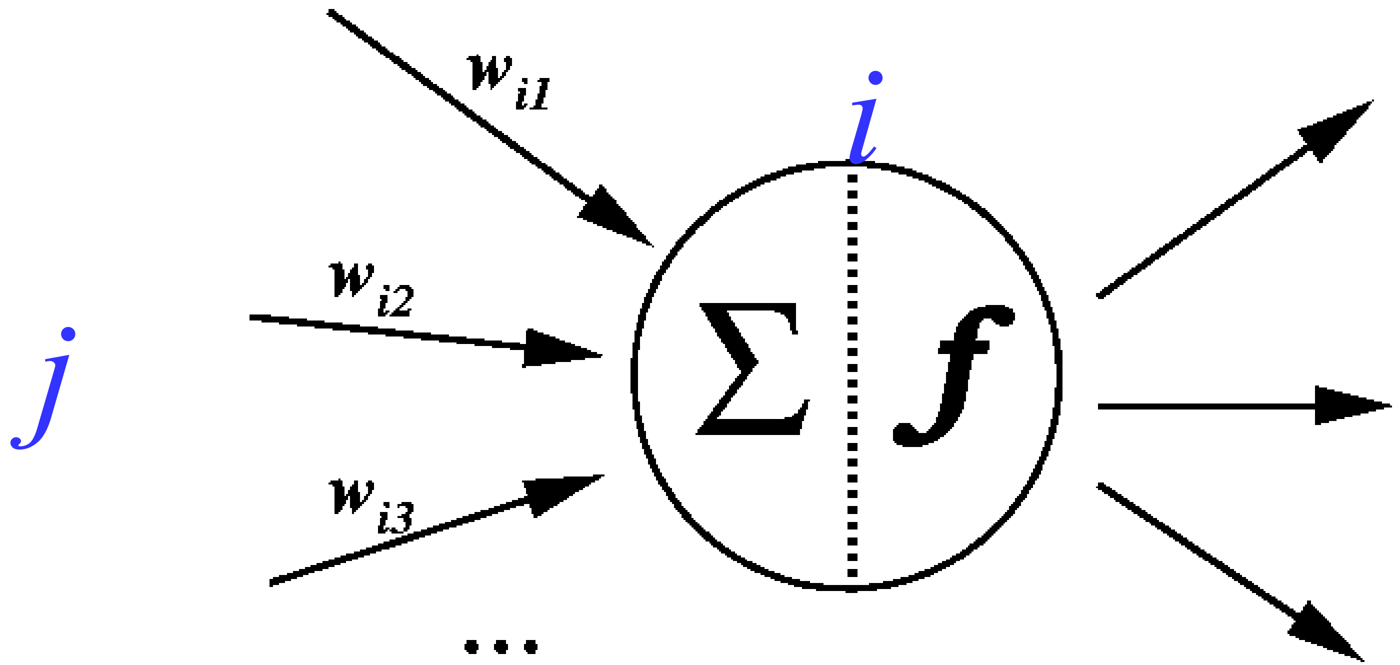
- 10^{11} neurons (processors)
- On average 1000-10000 connections



Artificial Neuron

bias

$$net_i = \sum_j w_{ij} y_j + b$$

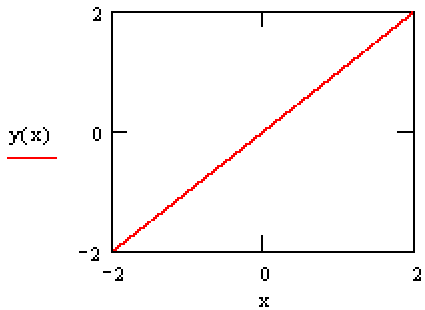


$$y_i = f(net_i)$$

The function *f* is the unit's activation function.

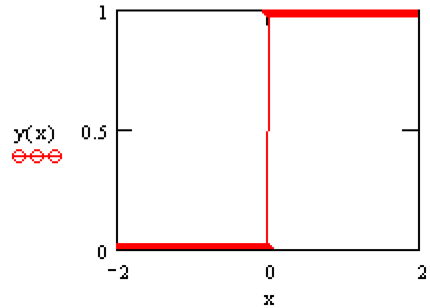
Activation Functions

$y(x) := x$



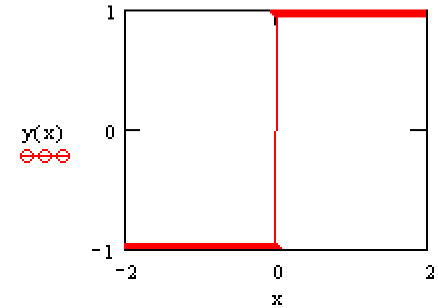
Identity function

$$y(x) := \begin{cases} temp \leftarrow 1 & \text{if } x>0 \\ temp \leftarrow 0 & \text{otherwise} \end{cases}$$



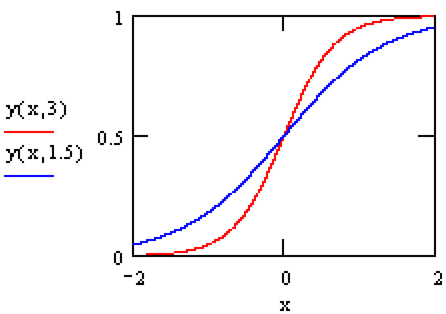
Binary Step function

$$y(x) := \begin{cases} temp \leftarrow 1 & \text{if } x>0 \\ temp \leftarrow -1 & \text{otherwise} \end{cases}$$



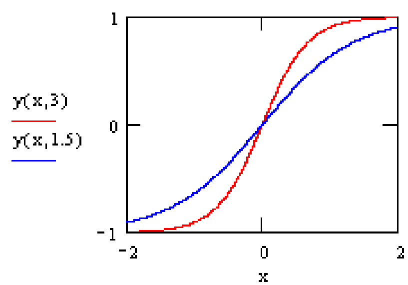
Bipolar Step function

$$y(x, \sigma) := \frac{1}{1 + \exp(-\sigma \cdot x)}$$



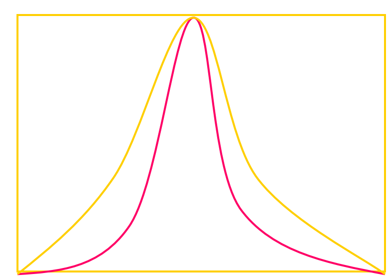
Sigmoid function

$$y(x, \sigma) := \frac{2}{1 + \exp(-\sigma \cdot x)} - 1$$



Bipolar Sigmoid function

$$y(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$



Gaussian function

When Should ANN Solution Be Considered ?

- The solution to the problem cannot be explicitly described by an algorithm, a set of equations, or a set of rules.
- There is some evidence that an input-output mapping exists between a set of input and output variables.
- There should be a large amount of data available to train the network.

NN Main Features

- NN Nonlinear Structure
  - Multi-Layer Perceptron (MLP)
- NN Estimation Error:

$$E(w(t)) = \frac{1}{2}\sum_{i=1}^{p}\left[d(i) - f\left(w(t)\cdot x(i)\right)\right]^2$$

- Gradient Descending Learning

Gradient Descent Method

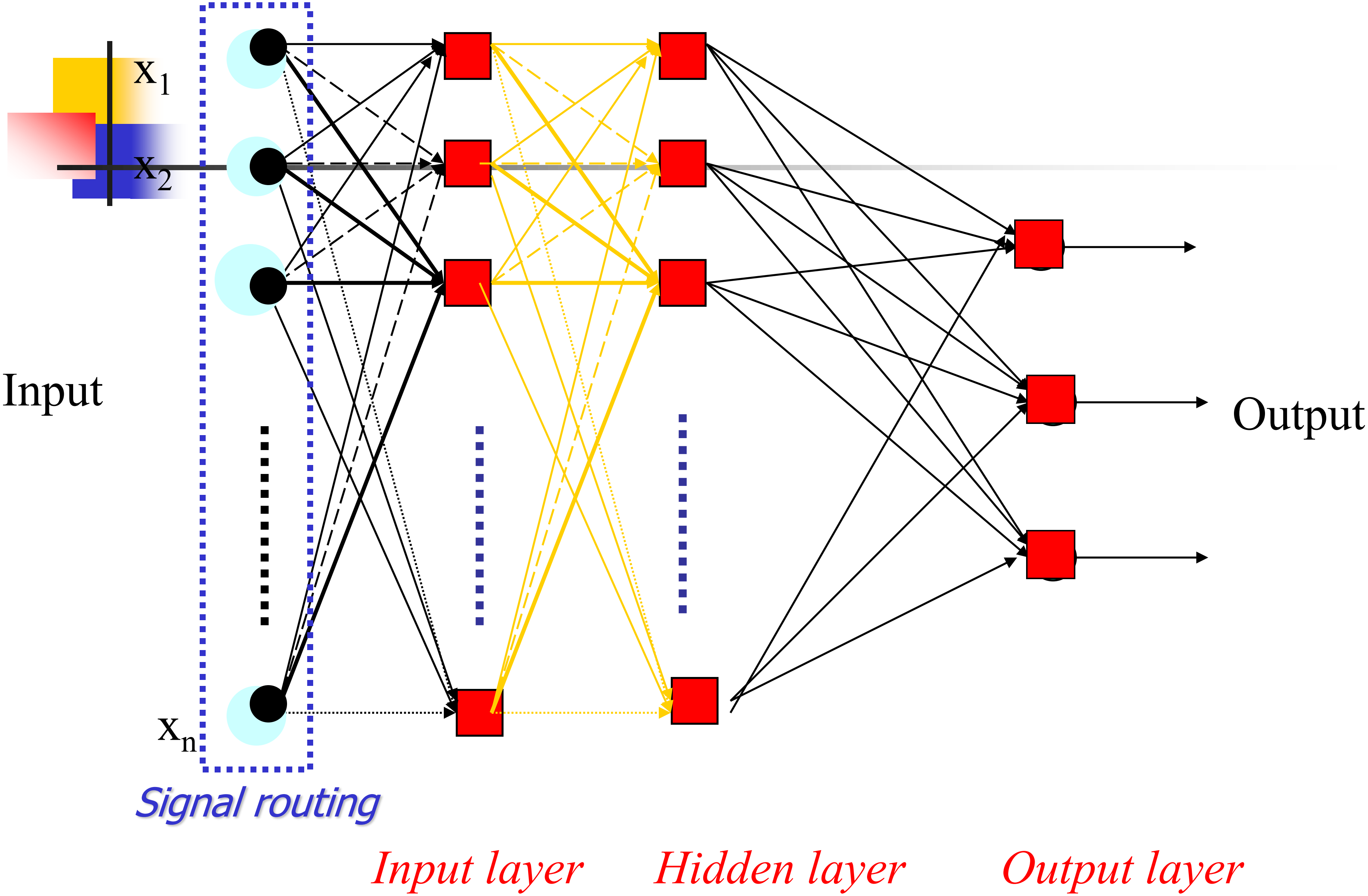
To find g

$$\underline{w}(t+1) = \underline{w}(t)+g(\ E(\underline{w}(t))\ )$$

so that $\underline{w}$ automatically tends to the global minimum of E(w).

$$\underline{w}(t+1) = \underline{w}(t) - E'(\underline{w}(t))\eta(t)$$

Example: *Three-layer* networks



Neural Networks

- Main Ability of NN: Learn from examples
- Trained to represent relationships b/w past values of residual data and faults
- No mathematical model is needed if sufficient training data is available.
- Generalise when presented with I/P not appearing in the training data
- Decision-making in noisy or corrupted data

Neural Networks : Strategies for FDI

- Pattern Recognition
- Model based

Residual generation: Residual ***r*** determined in order to characterise each fault. Ideally, the NN models identify all classes of system behaviour.

Decision-making: Process the residual ***r*** to determine the location and occurrence time of each fault

Neural Networks : Strategies for FDI

- A single NN can be used for both stages simultaneously on cost of less transparency but improved training time
- Fault isolation: Requires training data available for all expected faults in terms or residual values or system measurements
- Used for classification in conjunction with other residual generating methods e.g. non-linear observers.
- Online FDI

Residual Evaluation with NN

Neural Network for residual evaluation!

Fault-free and faulty residual signals (*NN input*)

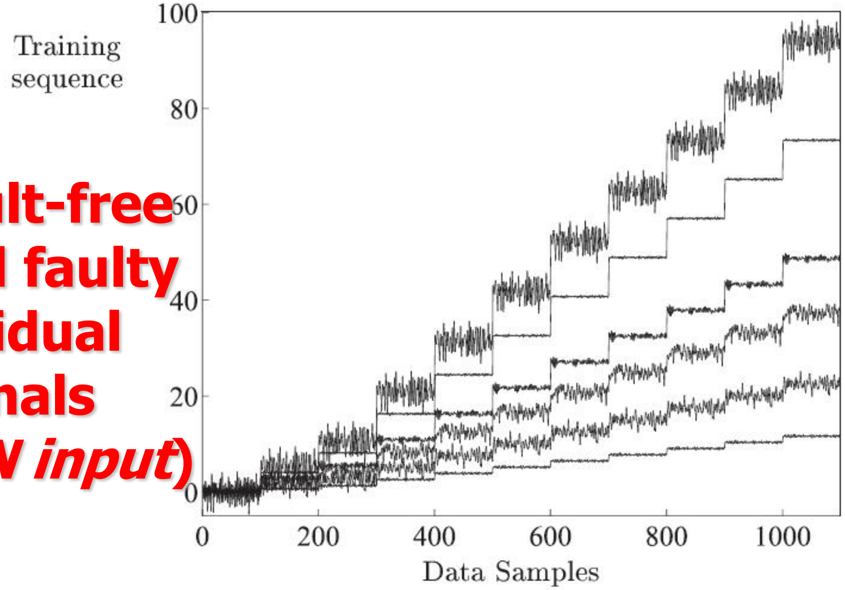
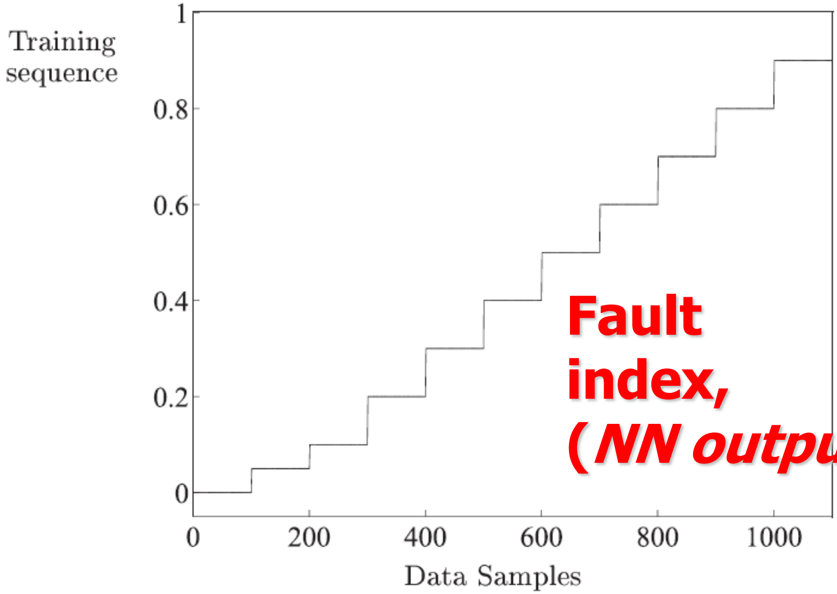


Fig. 5.27. NN input pattern.

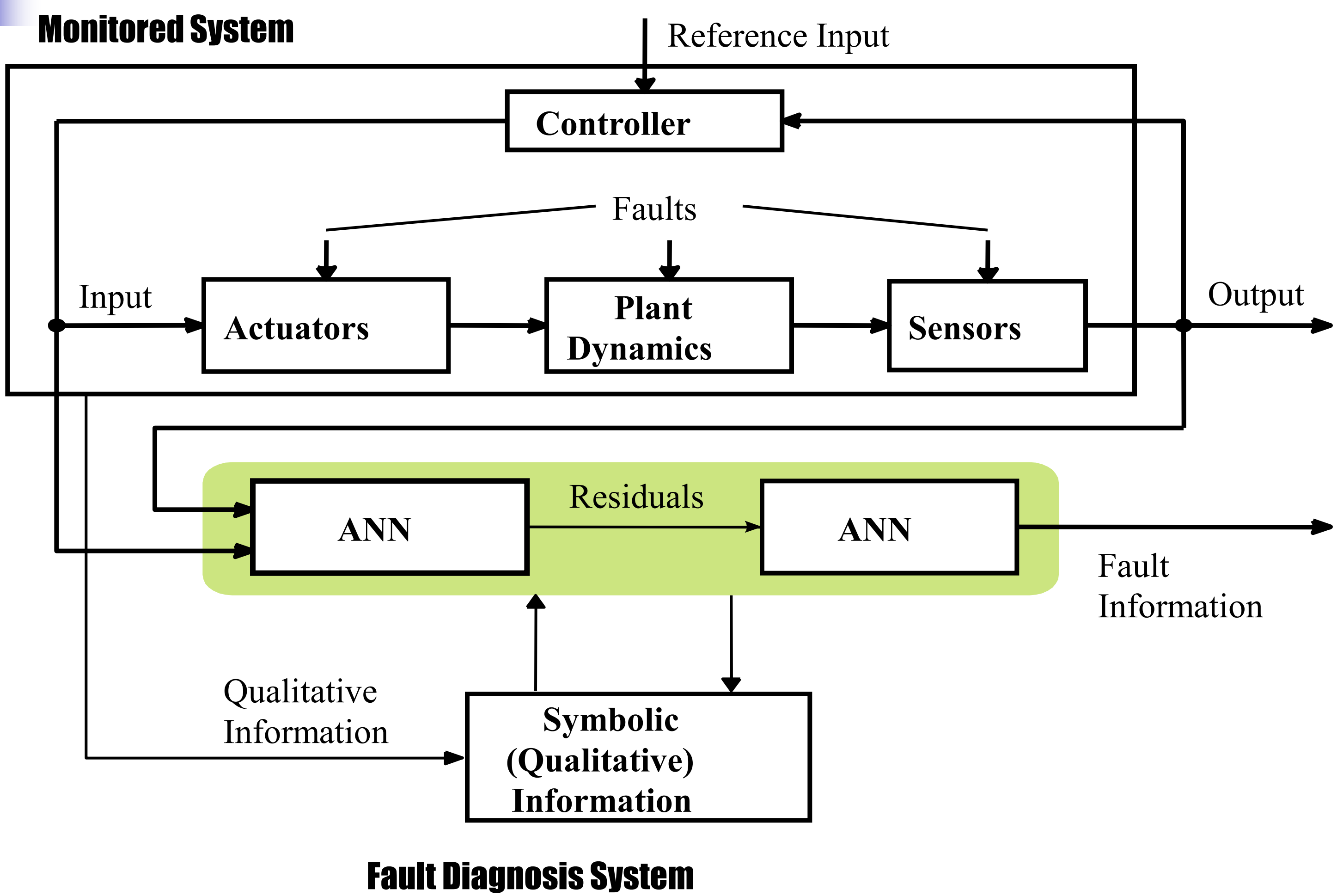


Fault index, (*NN output*)

3. Output pattern of the NN.

Neural Network Training Sequences

Neural Network Scheme for FDI



Application Examples

- **Industrial Process**
  - Gas Turbine Prototype
- **Aircraft/Aerospace Studies**
  - Small Commercial Aircraft
  - Mars Express Satellite/Probe

Simulated Gas Turbine

Prototype Model

Simulated Application Example (Cont'd)

Simulated Gas Turbine (SIMULINK®)

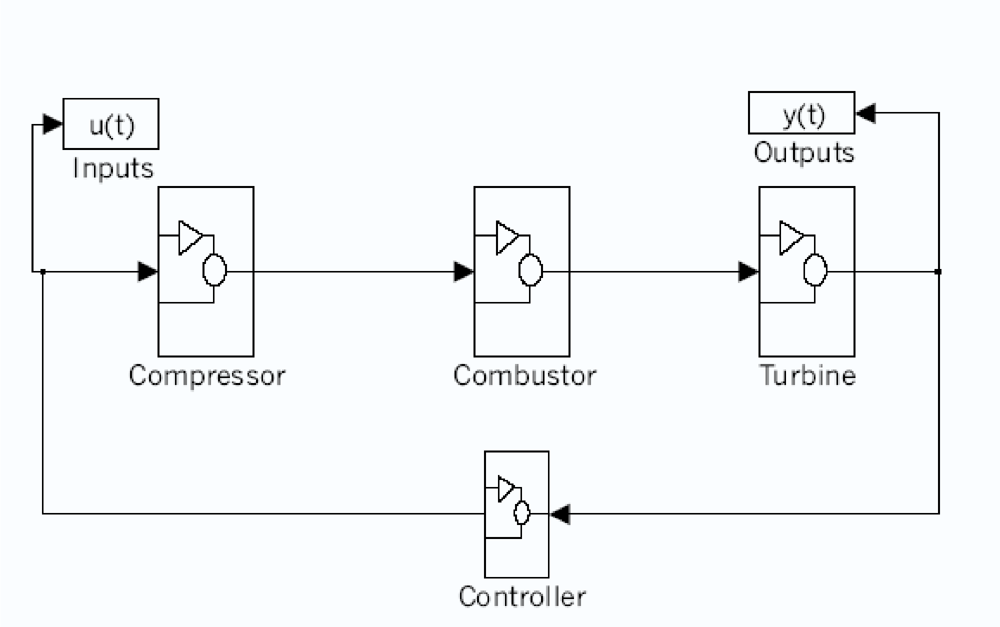


Figure 5.7: SIMULINK block diagram of the process.

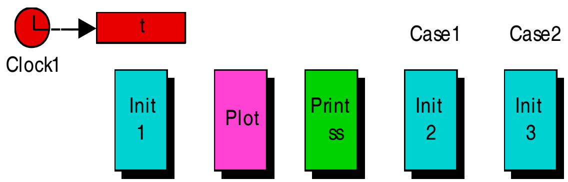
Gas turbine main cycle parameters (ISO design conditions).

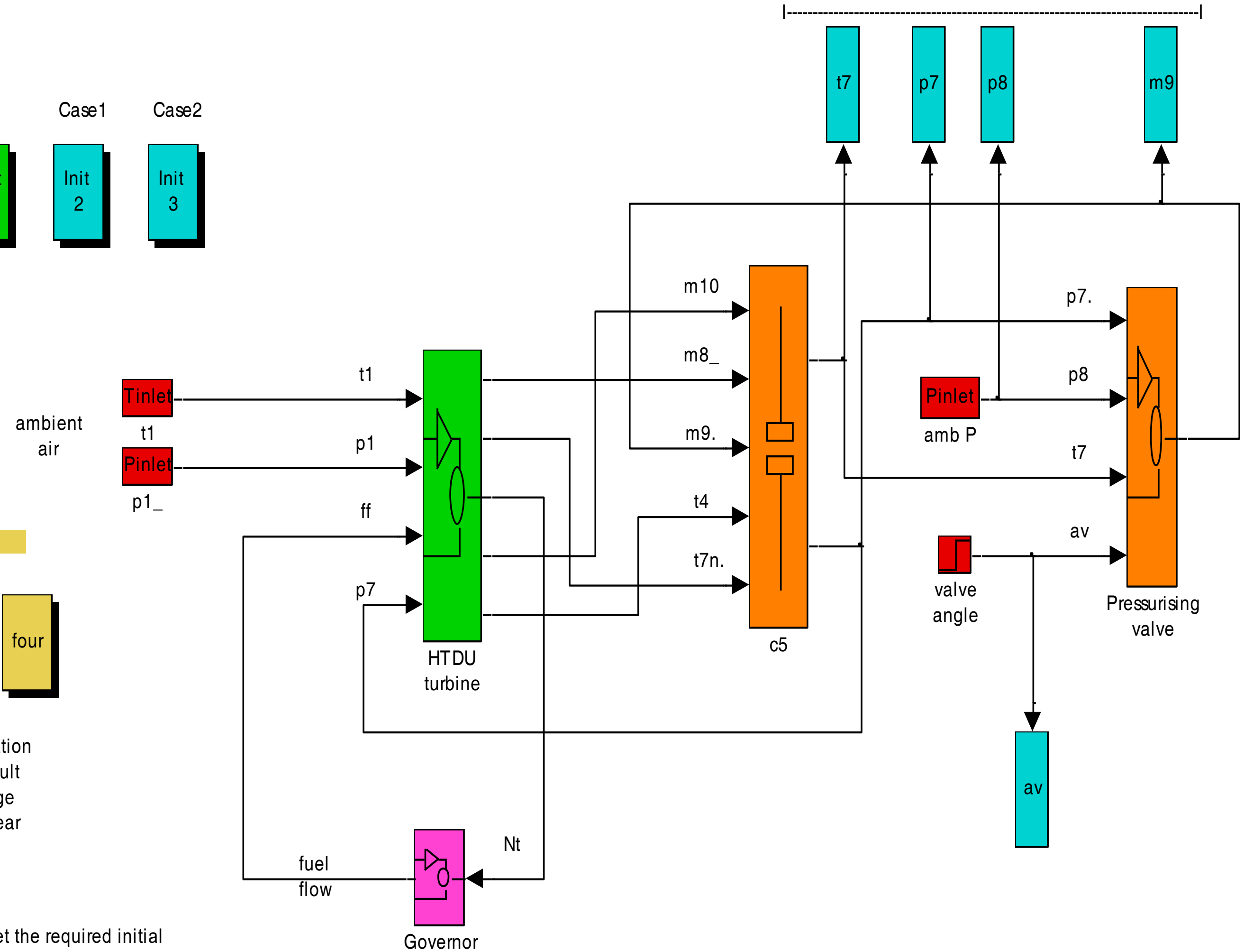
Air mass flow rate [kg/s]	24.4
Cycle pressure ratio (P_{oc}/P_{ic})	9.1
Electrical power (P_e) [kW]	5220
Exhaust temperature (T_{ot})[K]	796
Fuel mass flow rate (M_f) [kg/s]	0.388
IGV angle range ($\Delta\alpha$) [deg]	17

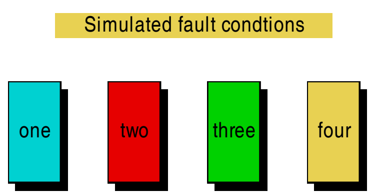
Gas Turbine FDI

- Work started in 1999 (UK):
  - Ron J. Patton & Mike Grimble,
  - Steve Daley & Andrew Pike
- Residual generation:
  - Kalman filters.
  - Fuzzy logic
- Residual evaluation:
  - Geometrical or statistical tests
  - Neural Networks

HTDU model
R B P 17/9/97
mods by AJLO 4/1/99







one = Compressor contamination
two = Thermcouple sensor fault
three = HP turbine seal damage
four = Fuel actuator friction wear

NOTE:

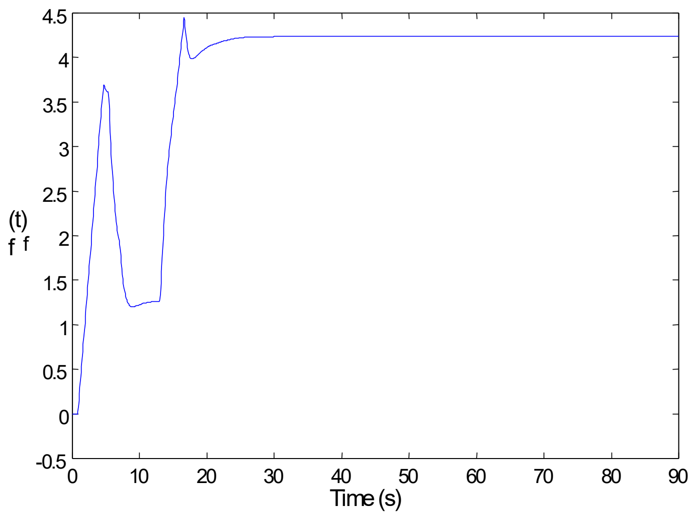
To run the with a fault condition. Set the required initial conditions (init 1, 2 or 3) then double click the required fault condition(s) before running the simulation. To reset to a no fault condition, simply reset the inital condtions.

Turbine Model

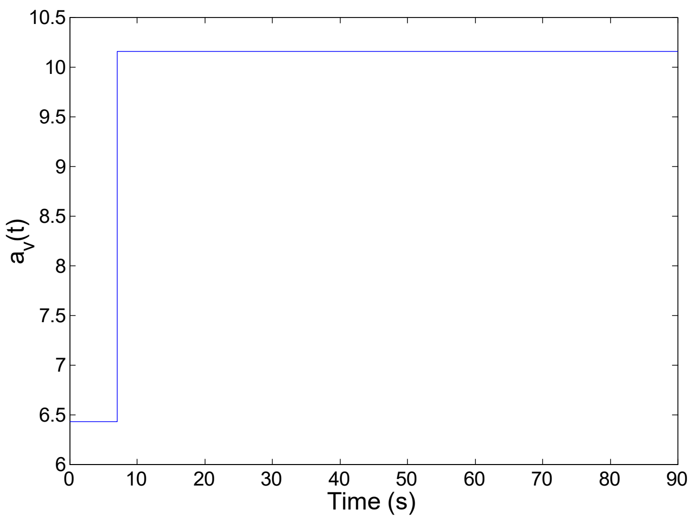
- Simulink dynamic model supplied, based upon an ABB ALSTOM experimental test rig.
- 1%-5% model accuracy.
- Steady-state validation.

Turbine Inputs & Outputs

Measure	$a_v(t)$	Torque	$f_f(t)$	Temp.	Press.	Mass
Accuracy	$\pm 2\%$	$\pm 1\%$	$\pm 5\%$	$\pm 1.5^0$	$\pm 1\%$	$\pm 5\%$



$f_f(t)$



$a_v(t)$

Fault Conditions

Four gradually developing faults:

1) Compressor contamination (*core engine* performance deterioration)
2) Thermocouple *sensor* fault
3) High Pressure turbine seal damage (*core engine* performance deterioration
4) Fuel *actuator* friction wear

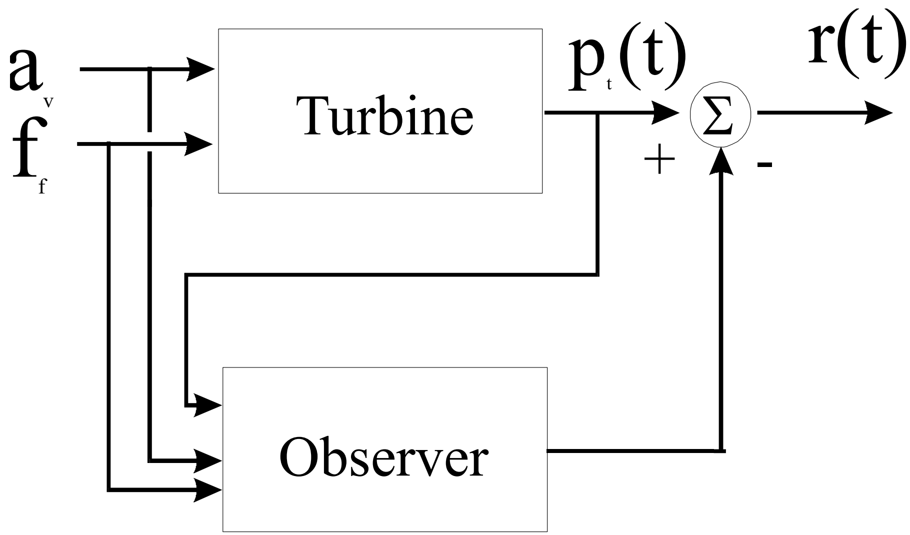
(realistic fault conditions!)

Compressor Contamination (1)

- **Fault 1: Compressor contamination**
- It represents fouling of the surfaces of the compressor blades.
- The failure is modeled as a gradual decrease in mass flow rate for a given pressure ratio.

Compressor Contamination (1)

- Mainly affects p_3 , p_4 , p_5 , p_t
- $\mathbf{p_t(t)}$ output observer
- Observer inputs: $\mathbf{f_f(t), a_v(t), p_t(t)}$
- residual generation $\mathbf{r_{13}(t)}$



Thermocouple Sensor Fault (2)

- **Fault 2: output sensor fault**

- Failure case 2 represents the malfunctioning of a thermocouple ($\mathbf{t_{3n}}$) in the gas path.

- It leads to a slowly increasing or decreasing reading over time.

High Pressure Turbine Seal Damage (3)

- **Failure case 3: failure of an HP turbine seal.**
- This results in a reduction in turbine efficiency.
- The fault is modeled as a gradual reduction in turbine efficiency over time.

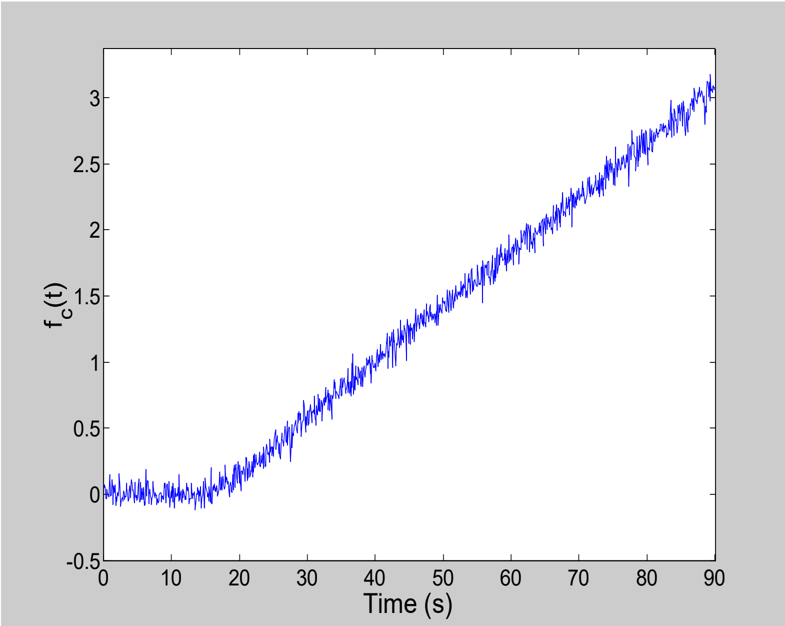
Fuel *Actuator* Friction Wear (4)

- **Failure case 4: loss of performance due to wear of the fuel valve actuator.**
- The effect of actuator wear causes slower response to demanded flow rates.
- It is modeled as a simple first order lag.

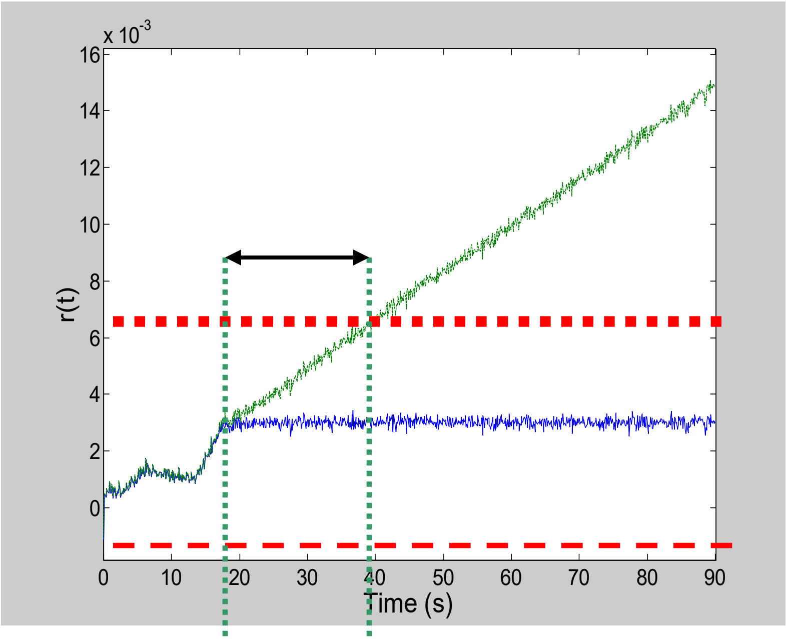
  The time constant increases linearly with time to represent progressive wear damage to the actuator.

Compressor Contamination (Case 1)

$p_t(t)$ residual generation using a Kalman filter



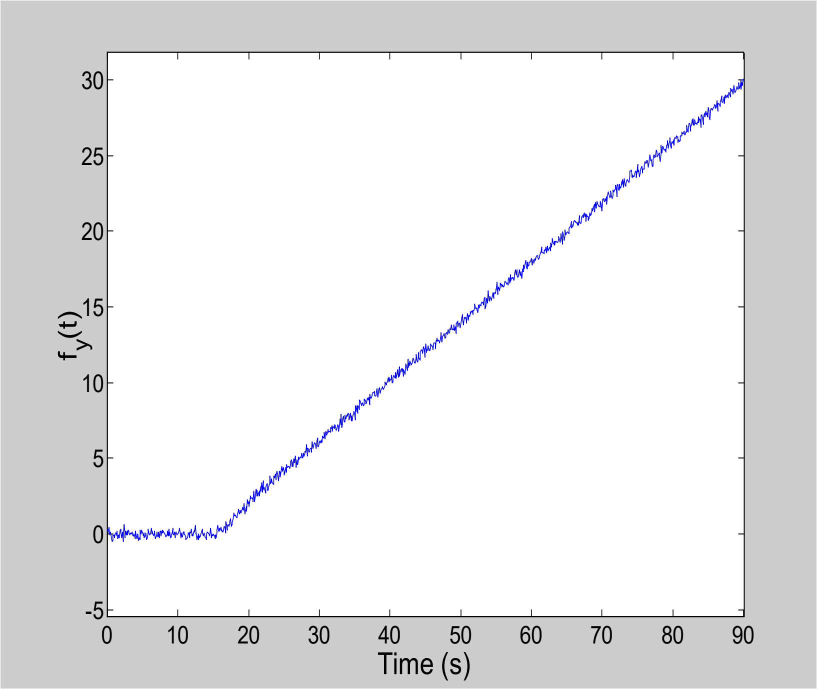
Faulty signal

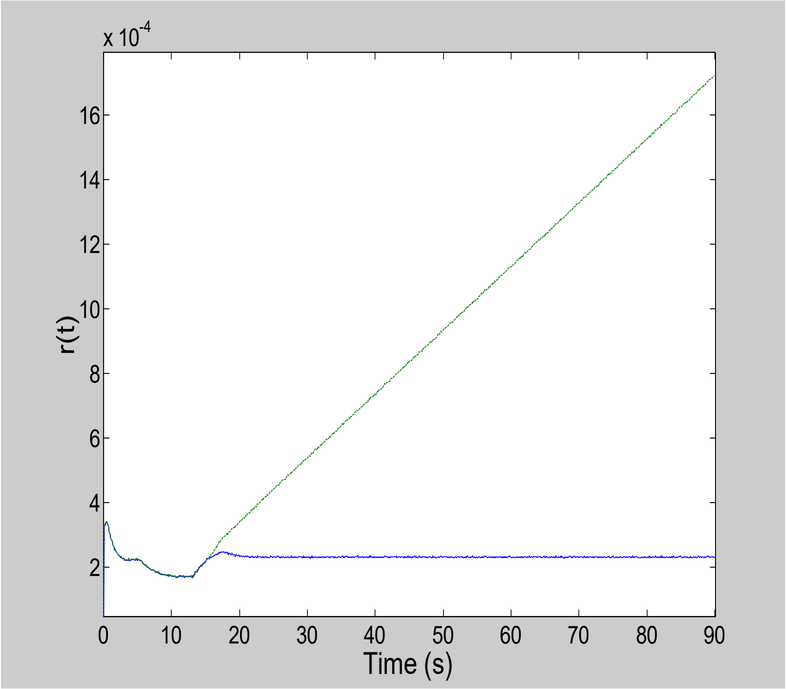


Fault free and faulty residual

Thermocouple Sensor (Case 2) Fault

t_{3n} residual generation using a Kalman filter



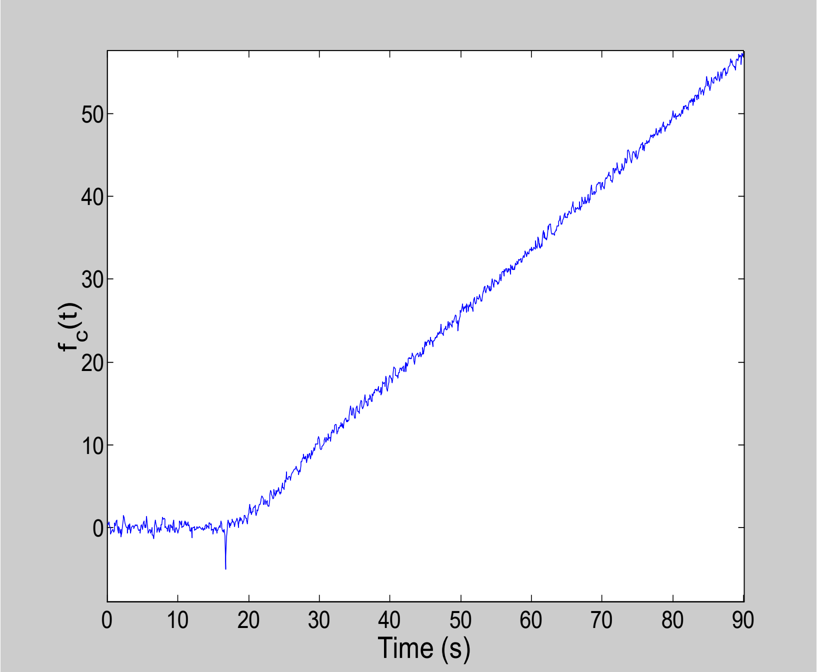


Faulty signal

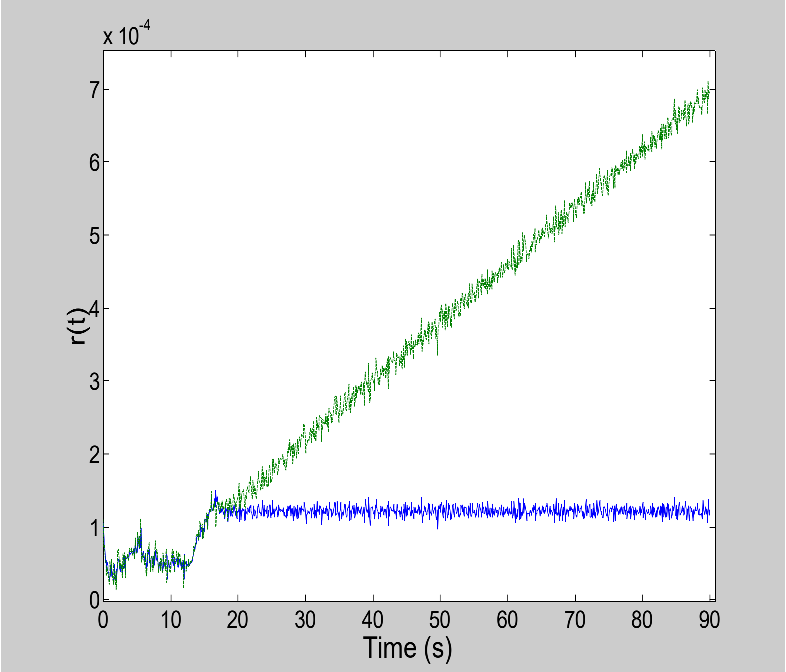
Fault free and faulty residual

High Pressure Turbine Seal Damage (Case 3)

$p_5(t)$ residual generation using a Kalman filter



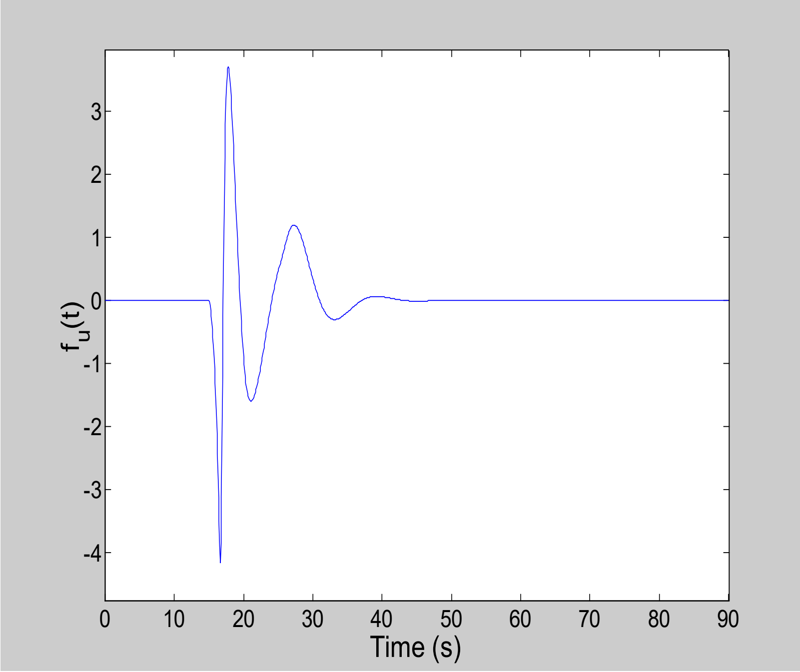
Faulty signal

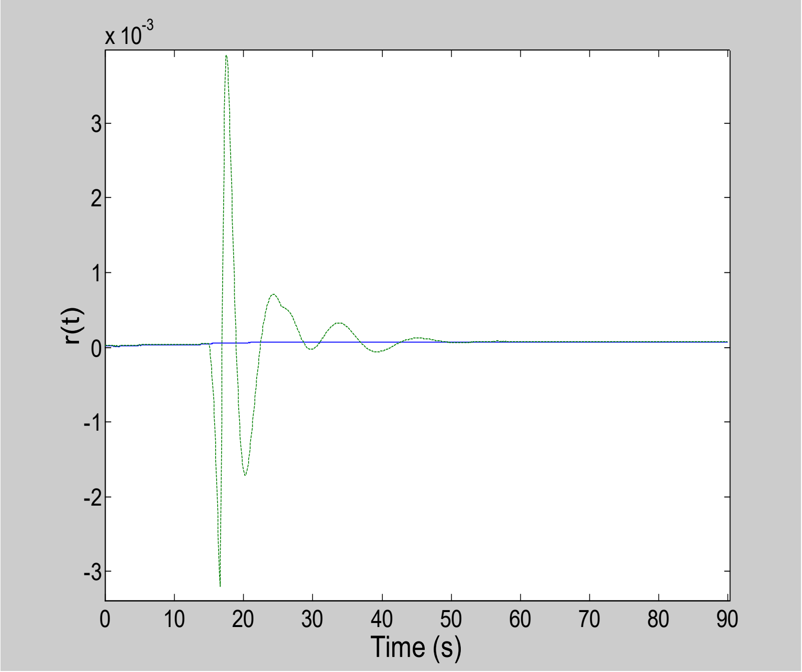


Fault free and faulty residual

Fuel *Actuator* Friction Wear (Case 4)

$q_t(t)$ residual generation using a Kalman filter





Faulty signal

Fault free and faulty residual

Fault Isolability

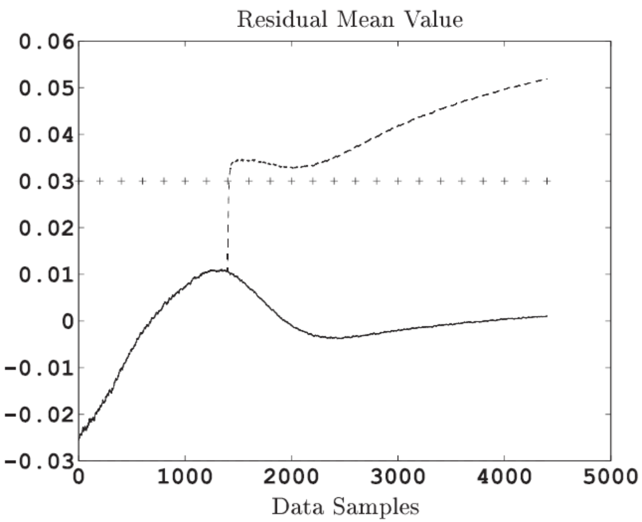
Fault signature: the most sensitive measurement

Fault/$r(t)$	p_3	p_4	p_5	p_7	p_t	q_a	q_c	q_t	t_{3n}	t_5 t_6	
Case 1	1	1	1	0	1	0	0	0	0	0	0
Case 2	0	0	0	0	0	0	0	0	1	0	0
Case 3	1	1	1	1	1	0	0	0	0	1	1
Case 4	1	1	1	0	1	1	1	1	0	0	0

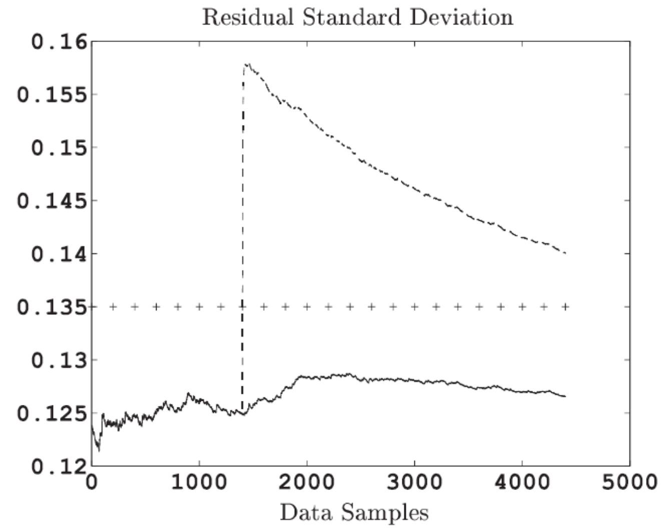
‘0’ if residual is not sensitive to a fault

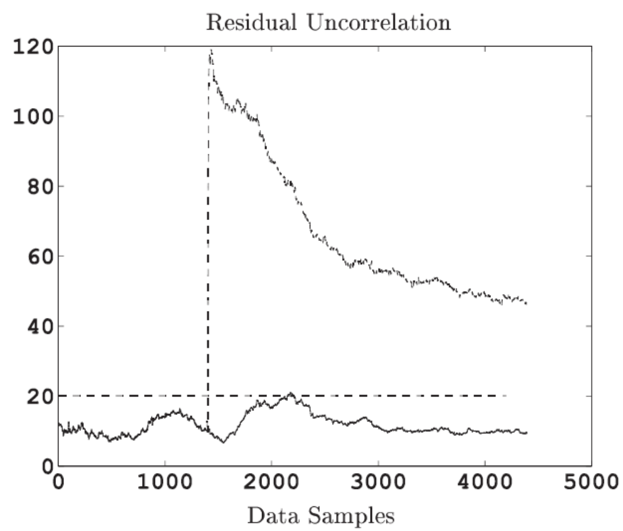
‘1’ if residual is sensitive to a fault

Residual Statistical Analysis



Mean value of the p_{ic} residual computed by using a growing





Minimum Detectable Faults

Fault case	Deterministic environment	Stochastic environment
Case 1	5%	11%
Case 2	5%	8%
Case 3	5%	9%
Case 4	5%	8%

- Faults expressed as per cent of the signals.
- Minimum delay FDI

NN Training for FDI

Neural Network for residual evaluation!

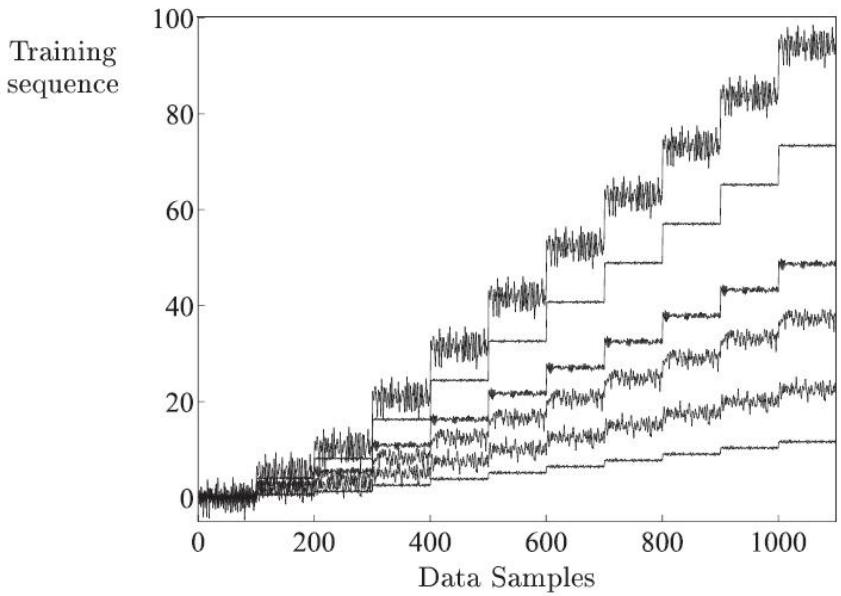
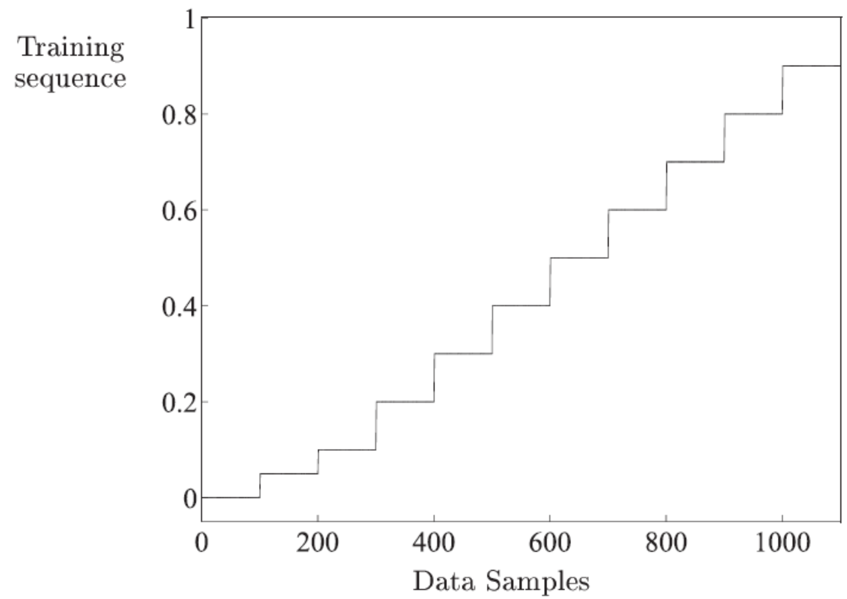


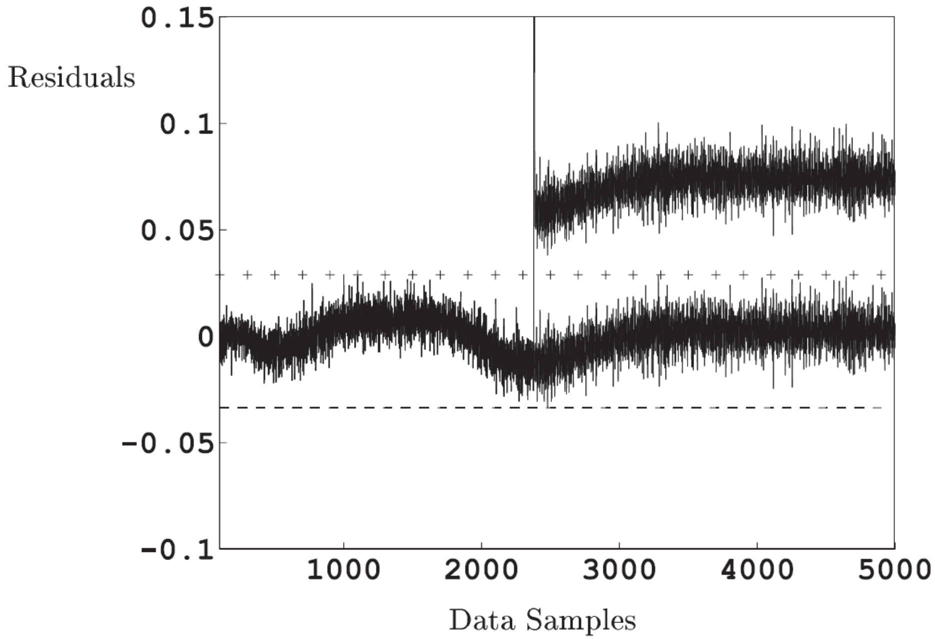
Fig. 5.27. NN input pattern.



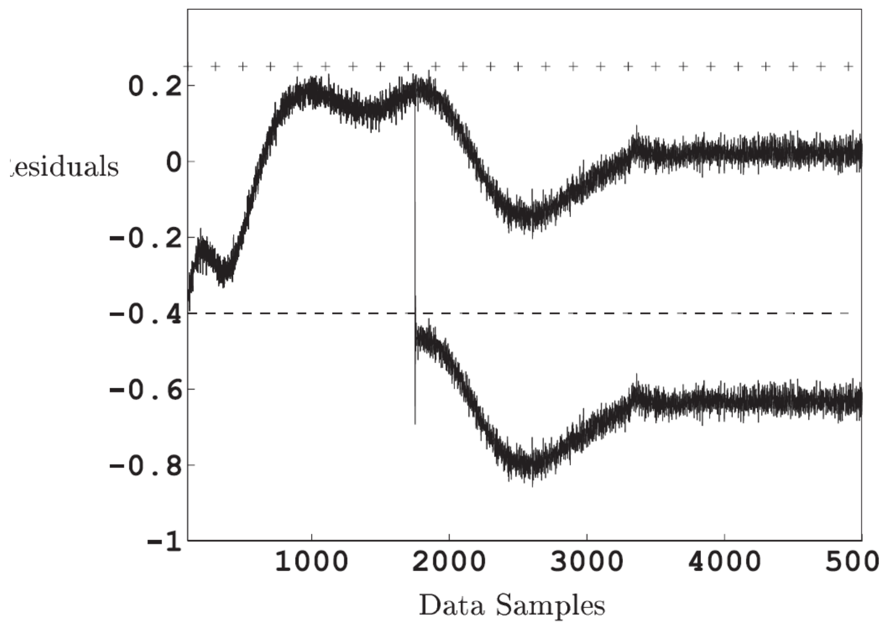
3. Output pattern of the NN.

Neural Network Training Sequences

NN Residuals for FDI



$p_{ot}(t)$ fault signal.



$\alpha(t)$ fault signal.

Training results concerning the IGV sensor.

Input layer	Hidden layer	SSE after 20000 epochs
15	15	0.17
15	20	0.24
20	30	**0.108**

Minimal detectable step faults.

Method	M_f	IGV
(NN)	3%	2.5%

Small Commercial Aircraft

PIPER PA-30

Simulated Application Example

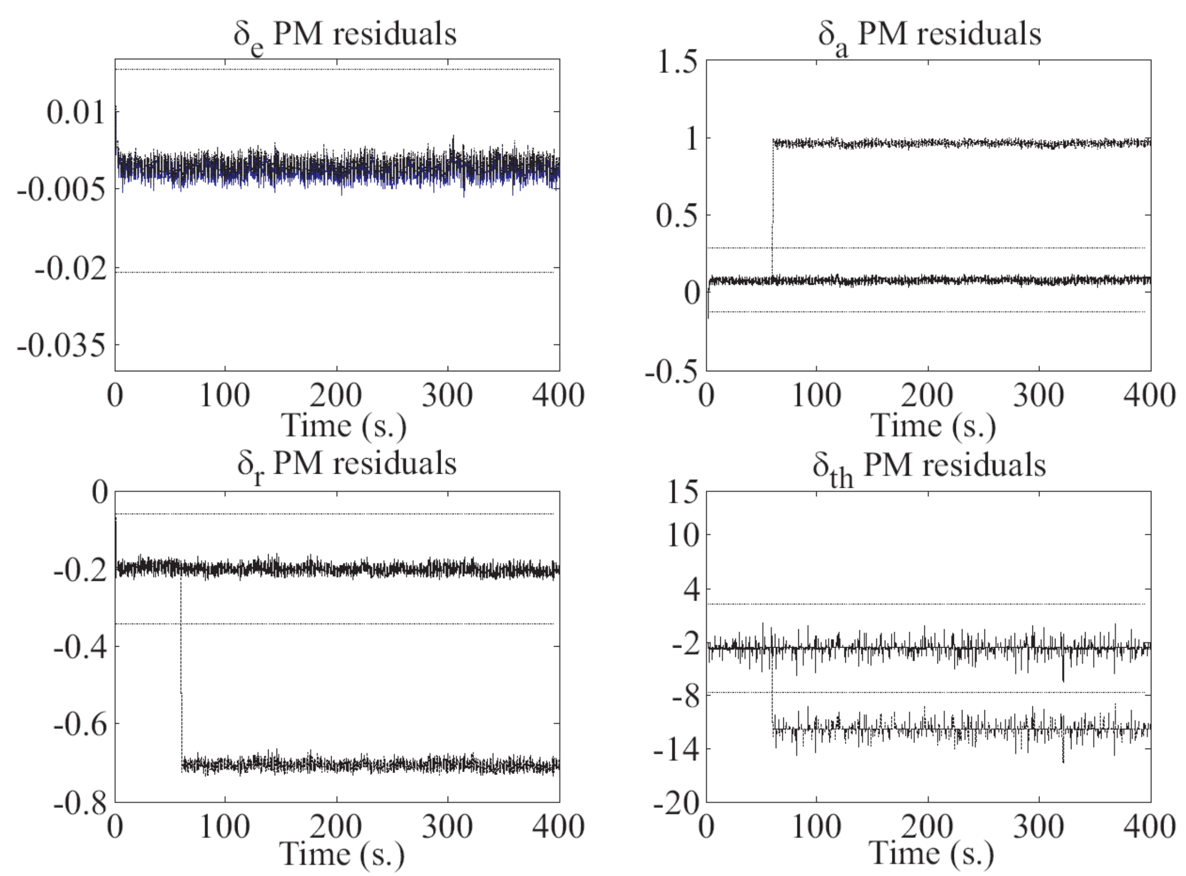
Small Aircraft Model

Piper Malibu

Table 1: Nomenclature

V	True Air Speed (TAS)	H	altitude
α	angle of attack	δ_e	elevator deflection angle
β	angle of sideslip	δ_a	aileron deflection angle
P	roll rate	δ_r	rudder deflection angle
Q	pitch rate	δ_{th}	throttle aperture percentage
R	yaw rate	γ	flight path angle
ϕ	bank angle	g	acceleration of gravity
θ	elevation angle	m	airplane mass
ψ	heading angle	I_x, I_y, I_z	principal-axis inertia moments
n	engine shaft angular rate	d_t	distance of c.g. from the Thrust line

Residual Signals (inputs)



UIO/ Kalman filters as residual generators

Fault Severity

- δ_e, δ_r, δ_a and δ_{th};
- V, ϕ, θ and n;
- ψ and H;
- P, Q and R.

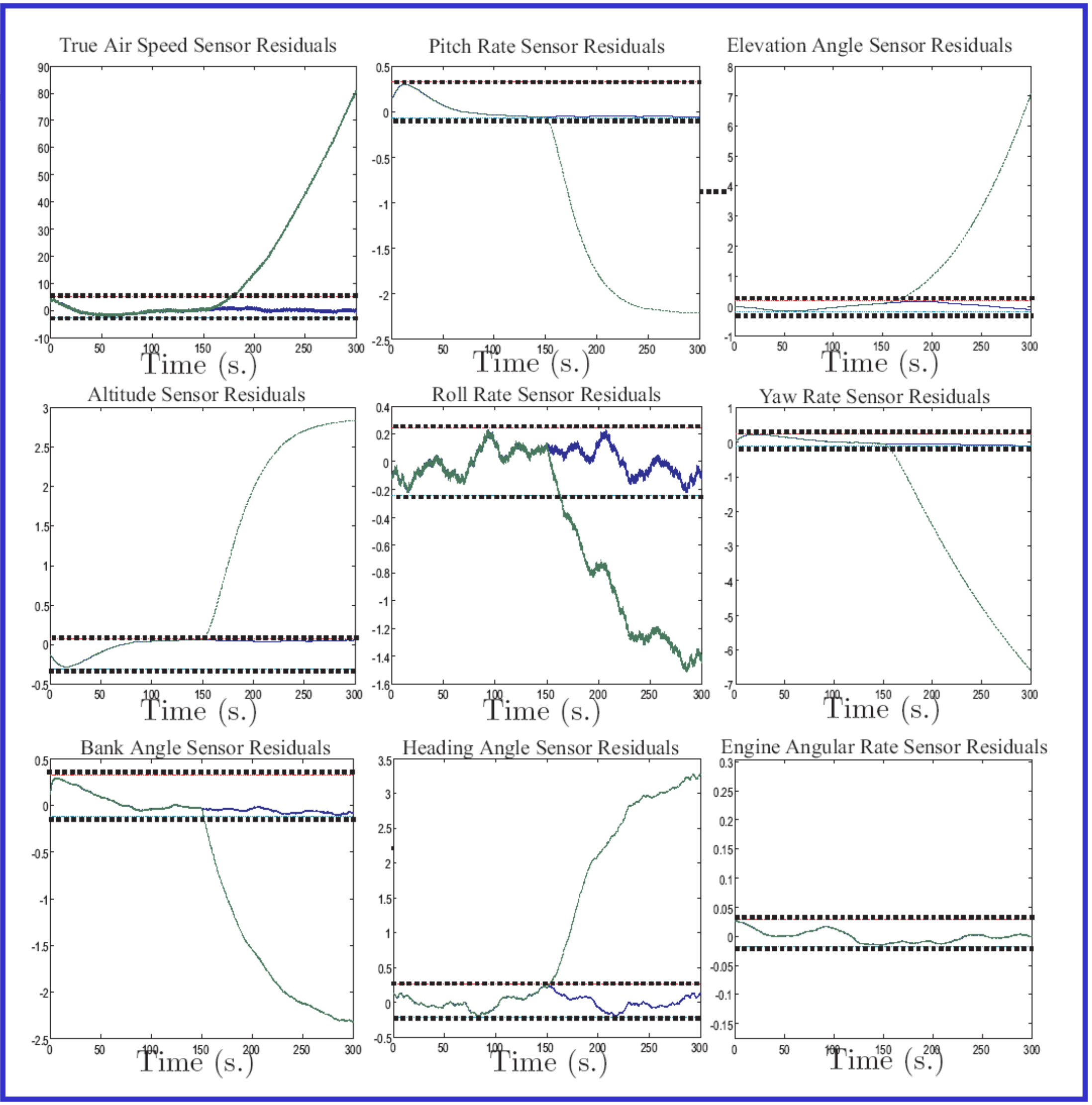
FDI of the 1st input sensor (elevator)

Residual Signals (outputs)

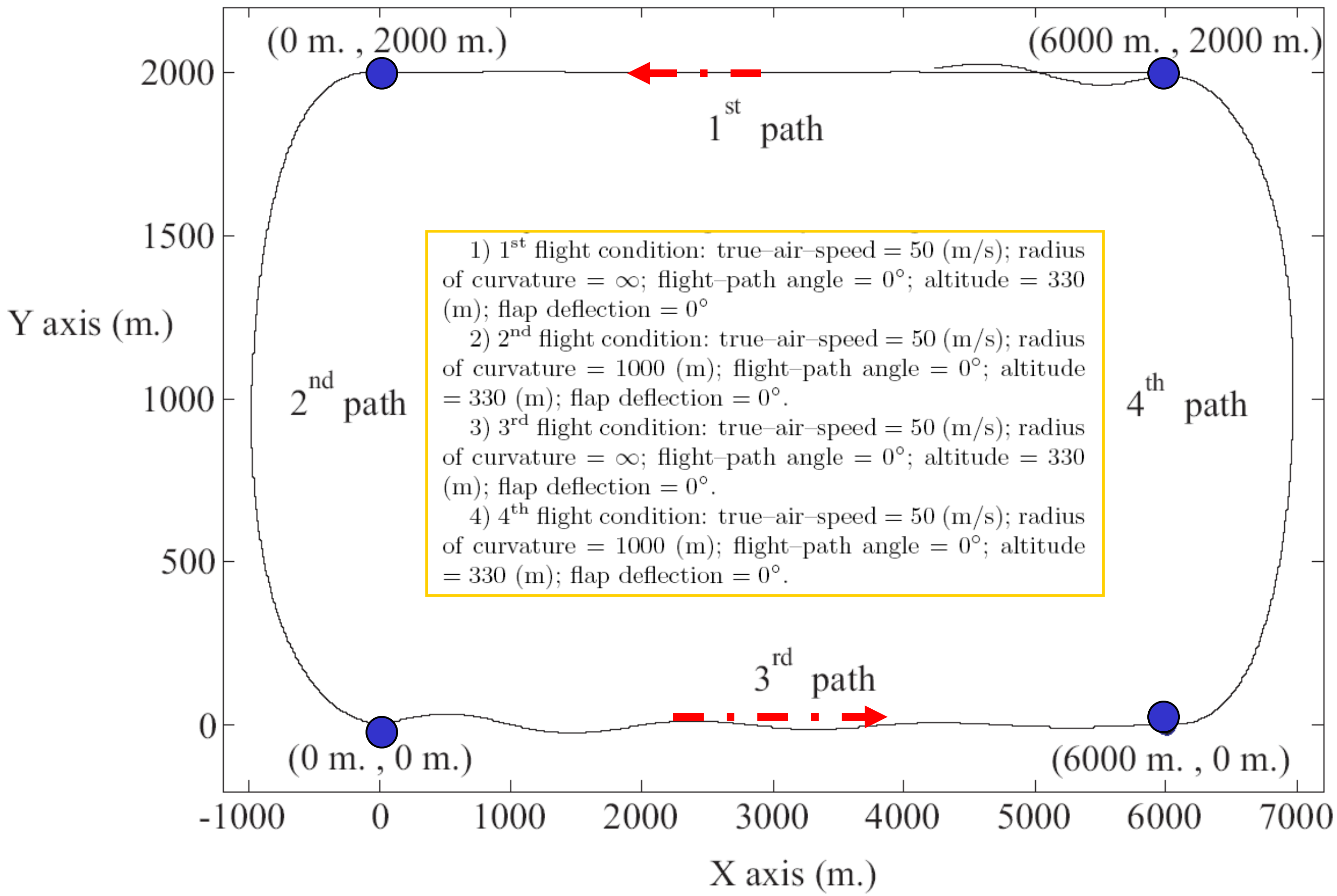
FDI of the 9th output sensor (engine speed)

Fault Severity

- δ_e, δ_r, δ_a and δ_{th};
- V, ϕ, θ and n;
- ψ and H;
- P, Q and R.



Validation Trajectory



Minimum Detectable Faults

Input sensor $c_i(t)$	Variable	Fault size	Detection delay (s)
Elevator deflection angle	δ_e	2°	18
Aileron deflection angle	δ_a	3°	6
Rudder deflection angle	δ_r	4°	8
Throttle aperture (%)	δ_{th}	2%	15

Output sensor $y_i(t)$	Variable	Fault size	Detection delay (s)
True air speed	V	8 m/s	9
Pitch rate	Q	3°/s	22
Elevation angle	θ	5°	10
Altitude	H	8 m	12
Roll rate	P	2°/s	24
Yaw rate	R	3°/s	29
Bank angle	ϕ	5°	5
Heading angle	ψ	6°	20
Engine angular rate	n	20 RPM	25

Minimum detectable faults and detection delays

Reliability Analysis

Monte–Carlo analysis by monitoring the PM residuals

Faulty sensor	r_{fa}	r_{mf}	r_{td}, r_{ti}	τ_{md}, τ_{mi}
δ_e	0.002	0.003	0.997	27s
δ_a	0.001	0.001	0.999	18s
δ_r	0.002	0.003	0.997	25s
δ_{th}	0.003	0.002	0.998	35s

Experimental Robustness/Reliability Assessment

Mars Express (MEX)

FDI

MEX Project

- “Robust Estimation for Failure Detection”
  - ESA (Holland), Astrium (Toulouse, France)
  - Denis Fertin, Bernard Polle
- University of Hull, Hull (UK)
  - Ron J. Patton, Faisal J. Uppal
- Università di Ferrara
  - Silvio Simani

Simulated Application Example

6.1 SATELLITE OVERVIEW

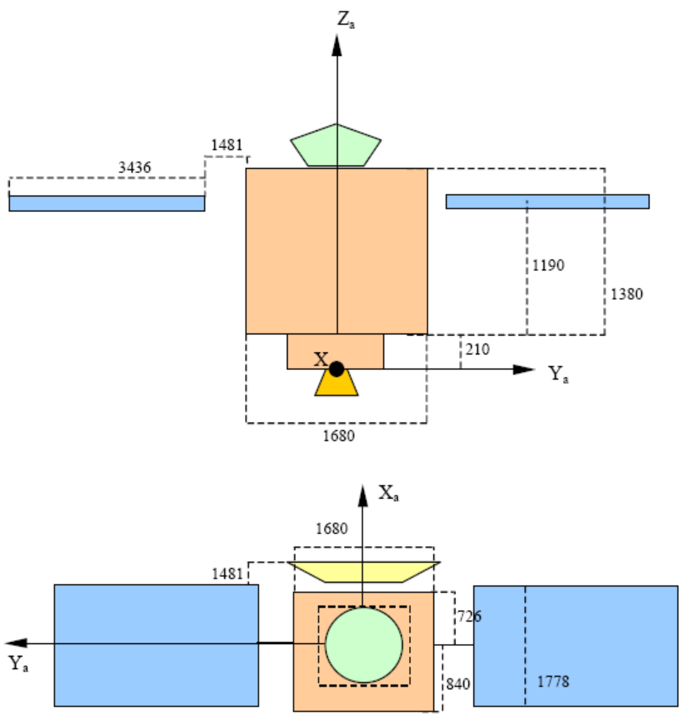


Figure 6-1: Satellite dimensions

6.4 SOLAR ARRAYS DYNAMICS

A GLOBALSTAR solar array has been selected. Its dimensions are recalled on the Figure 6-2.

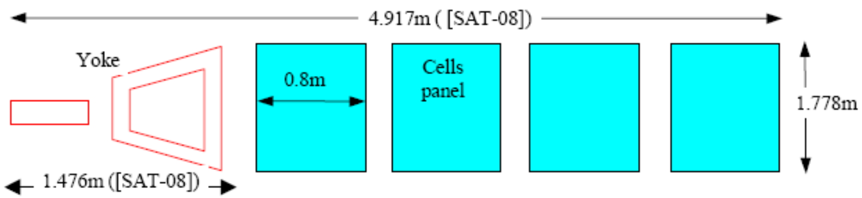


Figure 6-2 : MARS-EXPRESS solar array

Simple SA model

Aerospace Satellite

Simulated Application

Aerospace Satellite

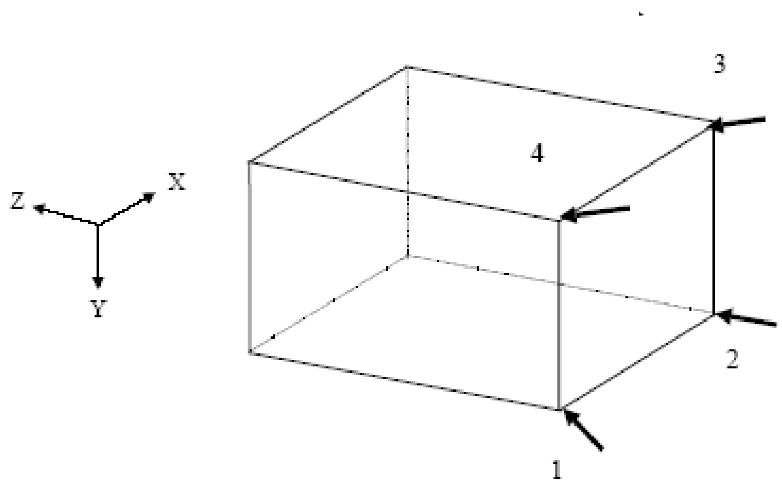


Figure 6-3 : Thrusters implementation

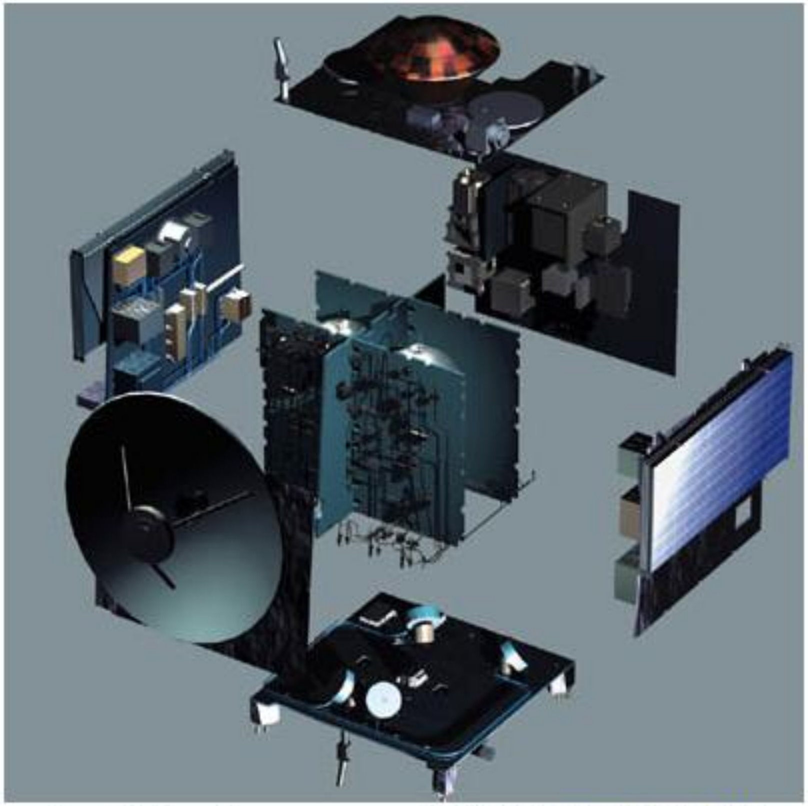
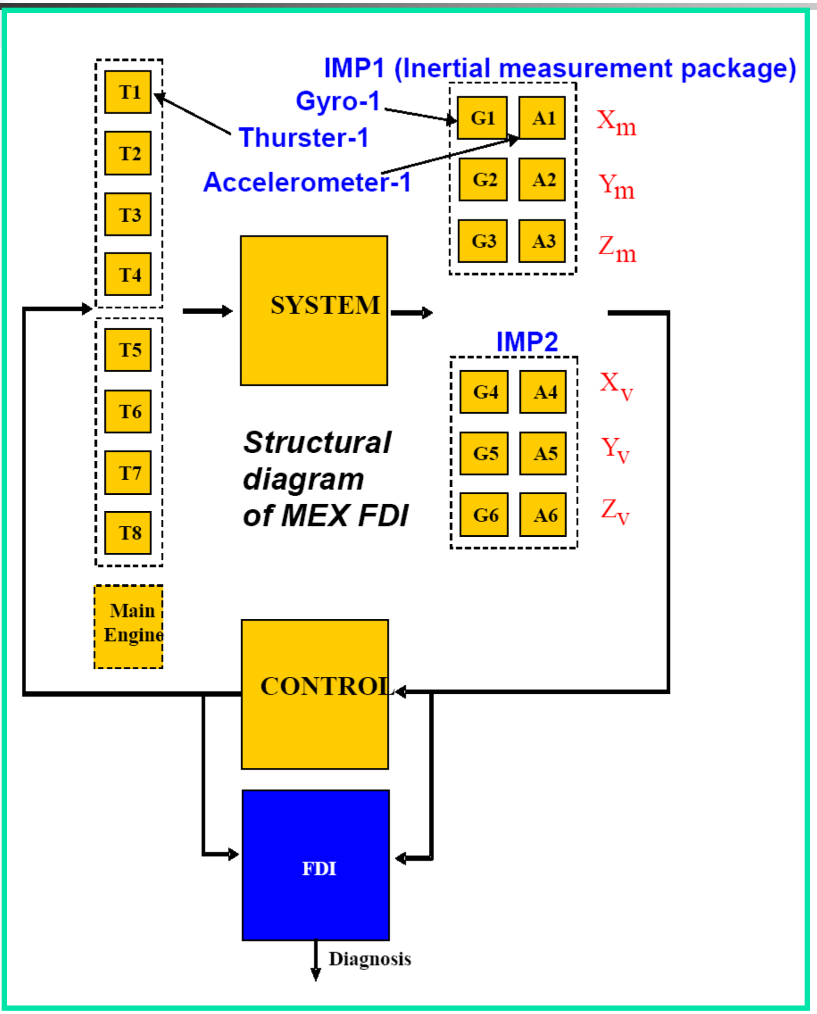
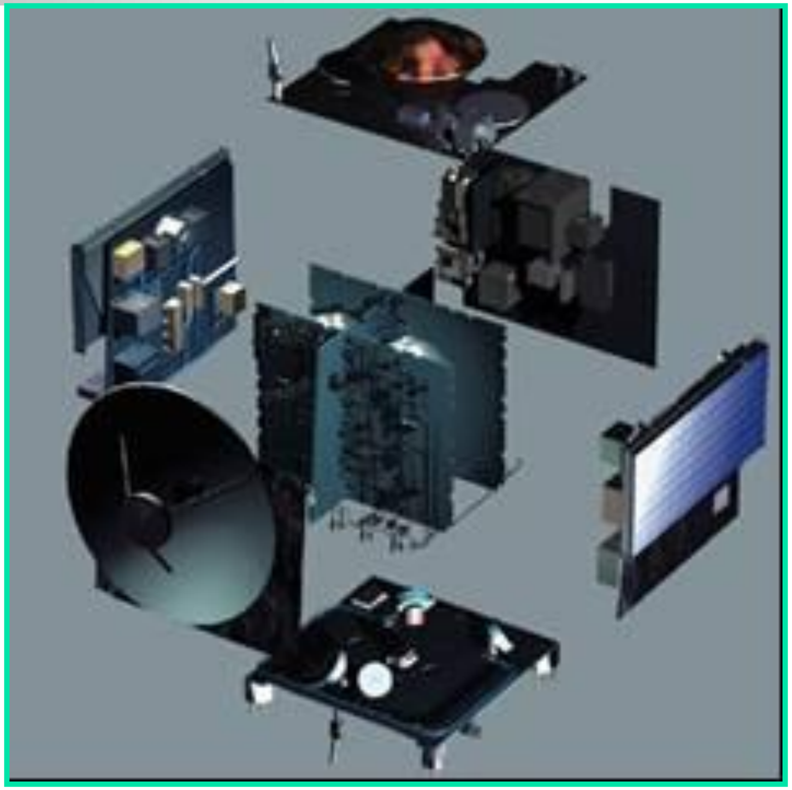


Fig. 1: Mars Express spacecraft (MEX) (www.esa.int)

Diagram of the MEX System

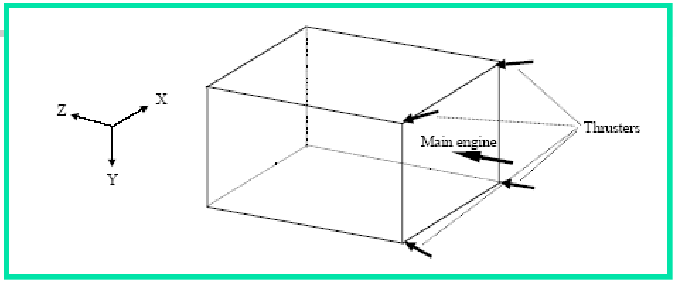


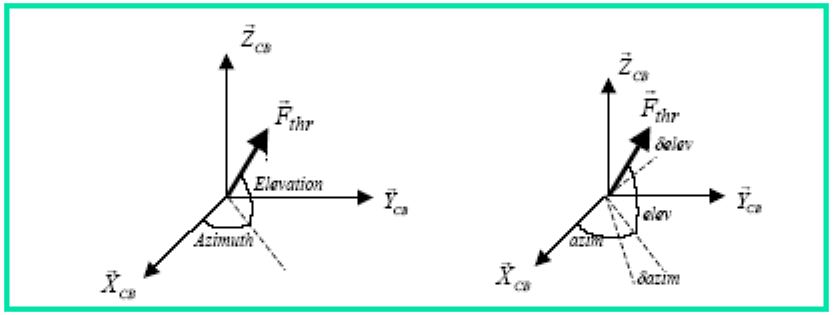


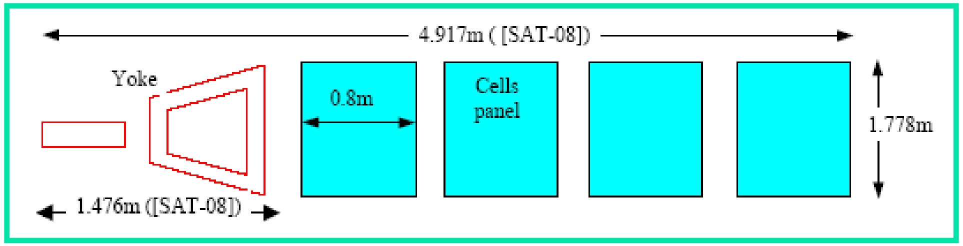
Mars Express Satellite

Process Modelling Problems

- Solar array dynamics
- Command input nonlinearity
- Centre of Gravity uncertainty
- Thrusters' misalignments
- IMU/IMP/Gyro misalignment errors
- Engine disturbance torques







Suggested Methodologies

1. **Satellite Dynamic Model**
   - ✓ Kalman filter design for residual generation;
   - ✓ Fixed thresholds for FDI

2. **UIO Observers**
   - ✓ Designed from the MEX model
   - ✓ Fault Isolation

Uncertain System

$$\begin{cases} x_p(t+1) = A_p\,\theta\;\; x_p(t) + B_p\,\theta\;\; u(t) \\ y(t) = C_p(\theta)\, x(t) \end{cases}$$

MEX process

u(t), y(t) — *System Identification*

$$\begin{cases} x(t+1) = A\; x(t) + B\; u(t) + H\;\; e(t) \\ y(t) = C\; x(t) + e(t) \end{cases}$$

Identified Model

(A, B, C, H) — *Observer Design*

$$\begin{cases} x(t+1) = A\,x(t) + B\,u(t) + H\;\; y(t) - C\,x(t) \\ y(t) = C\,x(t) \\ e(t) = y(t) - C\,x(t) \\ r(t) = e(t) \end{cases}$$

Diagnostic Observer/Filter

r(t) — *Residual Generation*

System Model

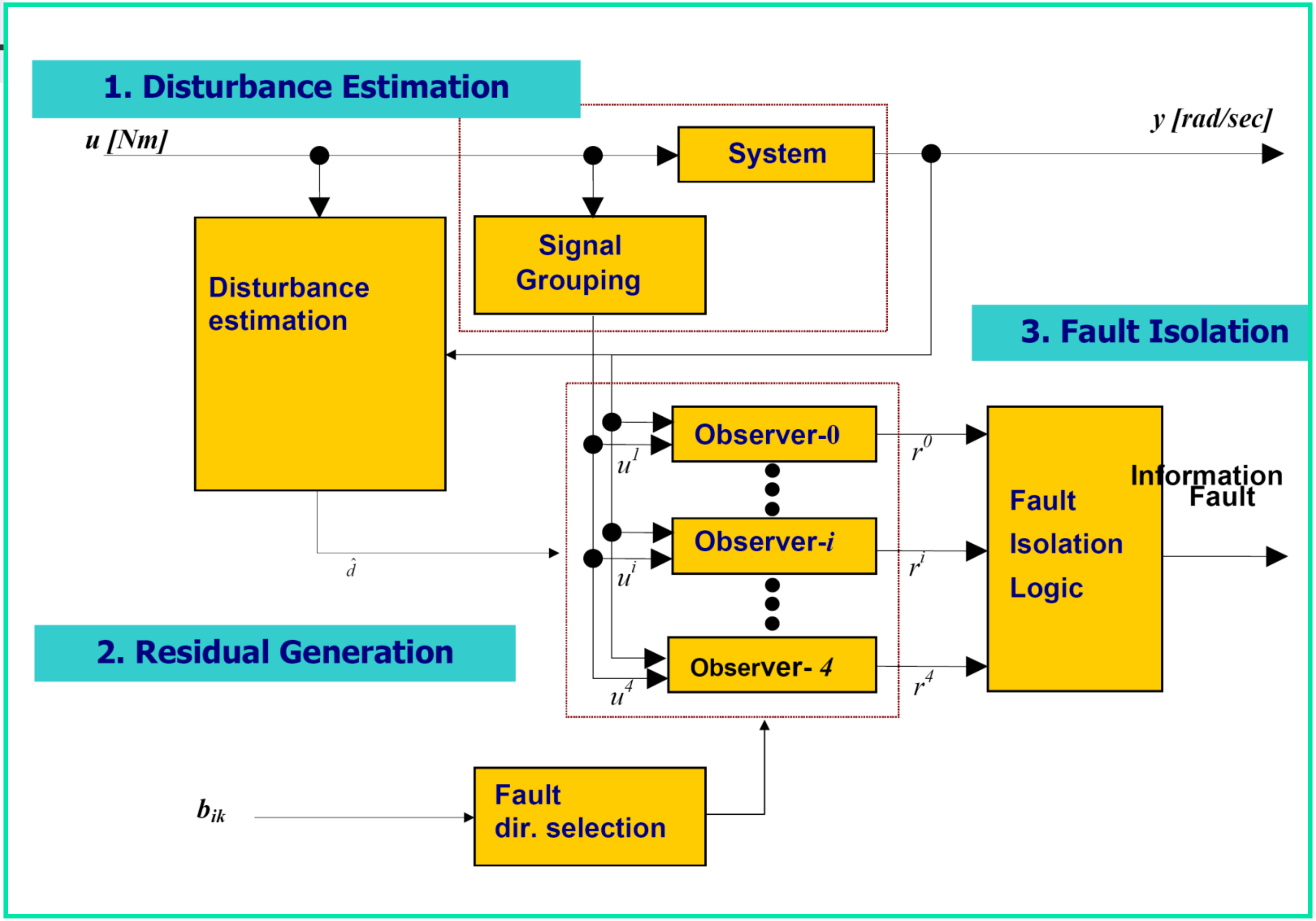
+

Residual Design

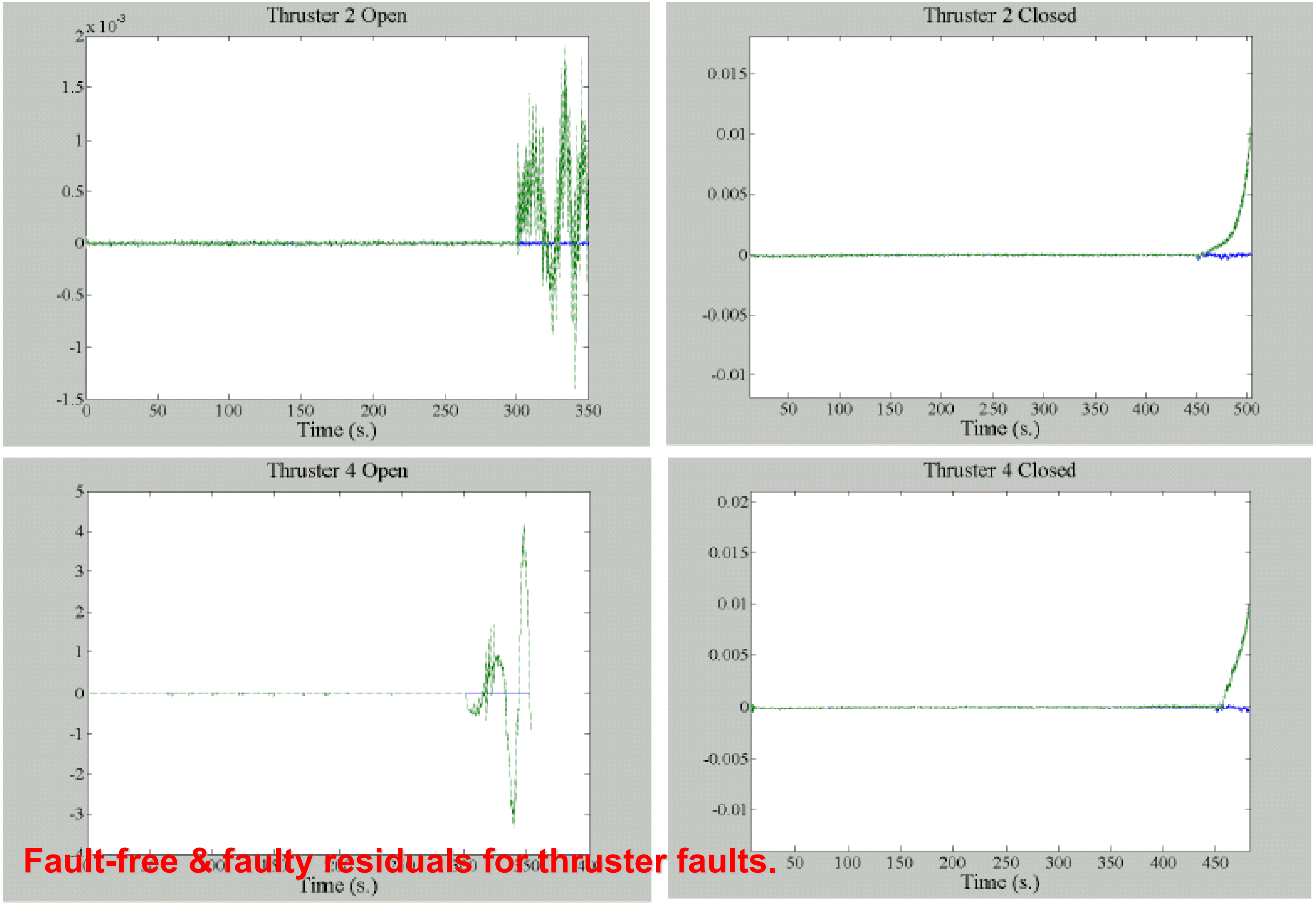
+

Residual Generation

FDI Design Overview: UIO



Residual Signal Examples…



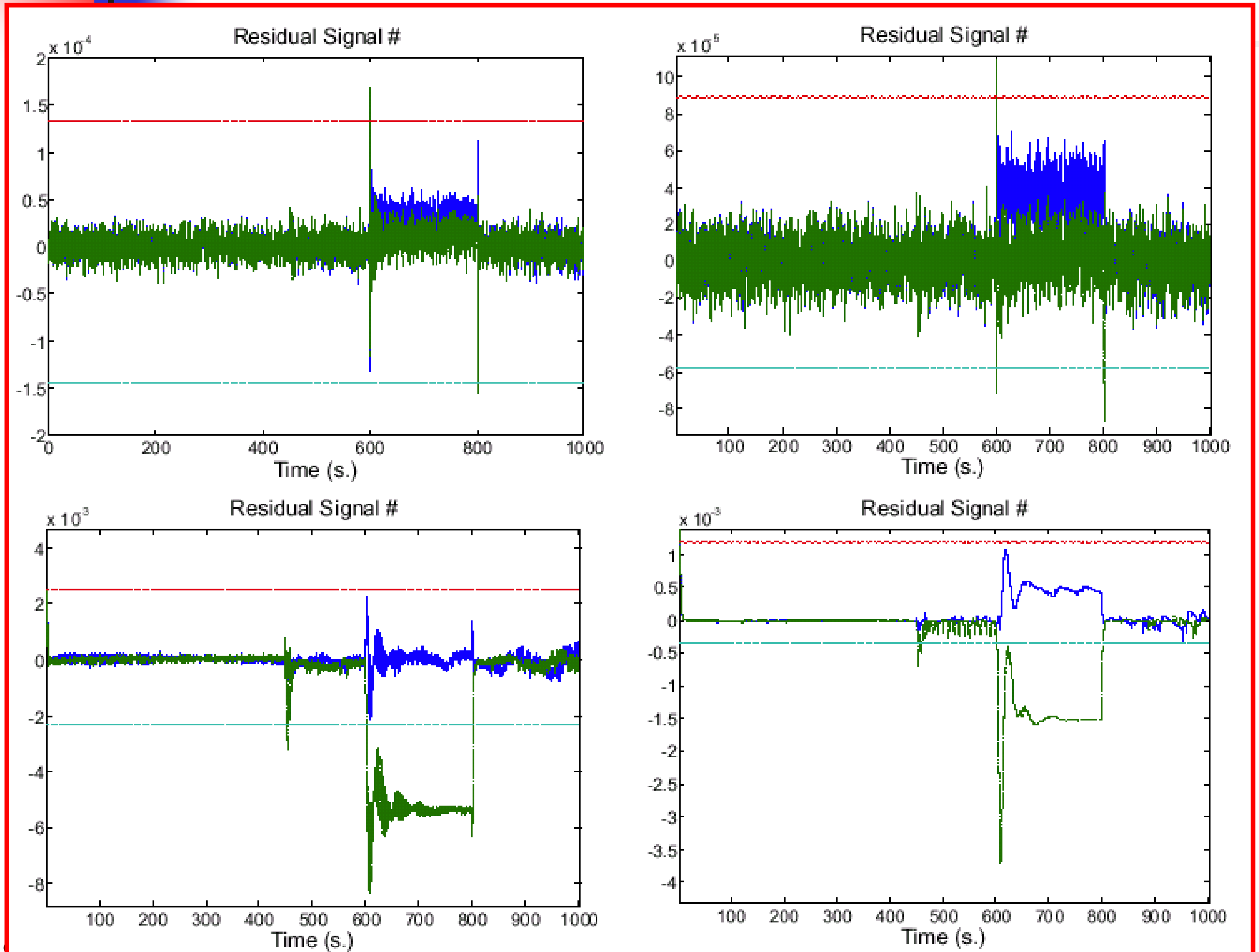
Fault-free & faulty residuals for thruster faults.

Performance Analysis Example

FAULT CASE	FAULT TIME	FALSE ALARM RATE	CORRECT ISOLATION RATE	WRONG ISOLATION RATE	MISSED FAULT RATE	MEAN DETECTION TIME	MEAN ISOLATION TIME	FAULT ALARM PROBABILITY	MONTE CARLO RUNS #	MEAN CPU TIME (Sec.)
Thruster1Closed	100	0.000	0.994	0.006	0.000	4.454	133.874	1.000	500	0.349
Thruster1Closed	300	0.000	0.996	0.004	0.000	7.975	120.400	1.000	500	0.371
Thruster1Closed	500	0.000	0.999	0.001	0.000	1.315	61.315	1.000	500	0.448
Thruster1Closed	650	0.000	0.994	0.006	0.000	0.100	46.914	1.000	500	0.520
Thruster1Closed	750	0.000	0.992	0.008	0.000	0.102	29.158	1.000	500	0.559
Thruster2Closed	100	0.000	0.999	0.001	0.000	3.022	116.371	1.000	500	0.350
Thruster2Closed	300	0.000	0.999	0.001	0.000	7.733	124.602	1.000	500	0.373
Thruster2Closed	500	0.000	0.992	0.008	0.000	1.301	58.577	1.000	500	0.437
Thruster2Closed	650	0.000	0.994	0.006	0.000	0.302	51.548	1.000	500	0.523
Thruster2Closed	750	0.000	0.993	0.007	0.000	0.114	28.654	1.000	500	0.554
Thruster3Closed	100	0.000	0.994	0.006	0.000	3.200	126.882	1.000	500	0.355
Thruster3Closed	300	0.000	0.999	0.001	0.000	8.389	126.433	1.000	500	0.375
Thruster3Closed	500	0.000	0.992	0.008	0.000	0.892	64.824	1.000	500	0.443
Thruster3Closed	650	0.000	0.998	0.002	0.000	0.109	48.933	1.000	500	0.521
Thruster3Closed	750	0.000	0.996	0.004	0.000	0.104	32.067	1.000	500	0.559
Thruster4Closed	100	0.000	0.999	0.001	0.000	2.497	113.289	1.000	500	0.348
Thruster4Closed	300	0.000	0.996	0.004	0.000	8.603	123.277	1.000	500	0.371
Thruster4Closed	500	0.000	0.999	0.001	0.000	1.116	58.326	1.000	500	0.441
Thruster4Closed	650	0.000	0.992	0.008	0.000	0.042	44.282	1.000	500	0.515
Thruster4Closed	750	0.000	0.998	0.002	0.000	0.101	26.398	1.000	500	0.550

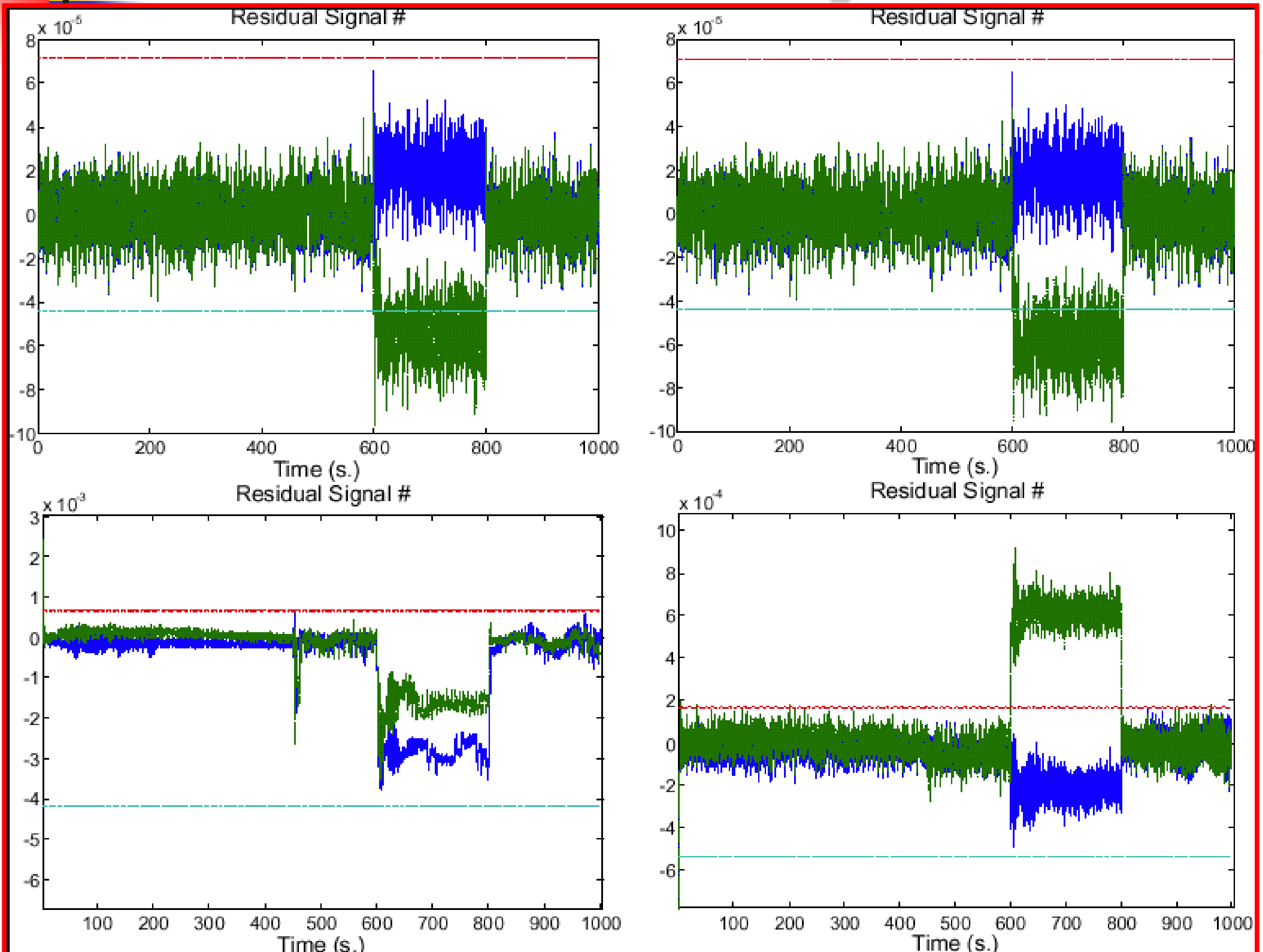
Performance evaluation html file for the thruster closed fault cases with 500 Monte-Carlo runs. The probabilistic thresholds were fixed with $\sigma = 3.0000$.

Performance Analysis Example



False alarm situation

Performance Analysis Example



Wrong Isolation Case

MEX FDI Method Results

Performance Index	Classical FDI	Method 1	Method 2
Mean false alarm rate	0.00	0.00	0.00
Detection possible	Yes	Yes	Yes
Isolation possible	No	Yes	Yes
Mean true isolation rate (typical value)	N/A	0.994 (Thruster closed) 0.996 (Thruster open)	More than 0.999 (1152 MC runs)
Max pointing error before detection (typical value)	1.08	0.10 (Thruster closed) 0.10 (Thruster open)	0.10 (Combined)
Max pointing error before isolation	N/A	N/A	0.11

Conclusions

- ✓ Model-Based FDI
- ✓ Analytical Redundancy
- ✓ State-Space Models
- ✓ Residual Generation
  - ✓ Unknown Input Observers UIO
  - ✓ Dynamic Observers / Kalman Filters
  - ✓ Neural Networks
- ✓ Residual Evaluation/Change Detection

FDI Issues

- From FDI to FTC
  - Fault Tolerant Control
- Model Uncertainty and FDI
  - Model-reality mismatch
  - Sensitivity problem: incipient faults!
- Robustness in FDI
  - Disturbance, modelling errors, uncertainty
  - UIO and Kalman filter: robust residual generation
  - Knowledge-based approaches (NN and FL)

References

- [Babuska, 1998] Babuska, R. (1998). Fuzzy Modelling for Control. Kluwer Academic Publishers.
- [Babuska, 2000] Babuska, R. (2000). Fuzzy Modelling and Identification Toolbox. Control Engineering Laboratory, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands, version 3.1 edition. (Available at http://lcewww.et.tudelft.nl/~babuska).
- [Basseville and Nikiforov, 1993] Basseville, M. and Nikiforov, I. V. (1993). Detection of Abrupt Changes: Theory and Application. Prentice-Hall Inc.
- [Beard, 1971] Beard, R. V. (1971). Failure accomodation in linear systems through self-reorganisation. Technical Report MVT-71-1, Man Vehicle Lab., Cambridge, Mass.
- [Bettocchi et al., 1996] Bettocchi, R., Spina, P. R., and Fabbri, F. (1996). Dynamic Modelling of Single-Shaft Gas Turbine. In ASME Paper 96-GT-332, pages 1-9.
- [Brown and Harris, 1994a] Brown, M. and Harris, C. (1994a). Neurofuzzy adaptive modelling and control. Prentice Hall.
- [Calado et al., 2001] Calado, J., Korbicz, J., Patan, K., Patton, R., and Sa da Costa, J. (2001). Soft computing approaches to fault diagnosis for dynamic systems. European Journal of Control, 7(2-3):248-286.
- [Chen and Patton, 1999] Chen, J. and Patton, R. J. (1999). Robust Model-Based Fault Diagnosis for Dynamic Systems. Kluwer Academic.
- [Chow and Willsky, 1980] Chow, E. Y. and Willsky, A. S. (1980). Issue in the development of a general algorithm for reliable failure detection. In Proc. of the 19th Conf. on Decision & Control, Albuquerque, NM.

References (Cont'd)

- [De Persis and Isidori, 2001] De Persis, C. and Isidori, A. (2001). A geometric approach to non-linear fault detection and isolation. IEEE Transactions on Automatic Control, 45(6):853-865.
- [Edwards et al., 2000] Edwards, C., Spurgeon, S. K., and Patton, R. J. (2000). Sliding mode observers for fault detection and isolation. Automatica, 36(1):541-553.
- [Emami-Naeini et al., 1988] Emami-Naeini, A., Akhter, M., and Rock, M. (1988). Effect of model uncertainty on failure detection: the threshold selector. IEEE Trans. on Automatic Control, 33(2).
- [Fantuzzi and Rovatti, 1996] Fantuzzi, C. and Rovatti, R. (1996). On the approximation capabilities of the homogeneous Takagi{Sugeno model. Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, pages 1067-1072.
- [Fantuzzi et al., 2002] Fantuzzi, C., Simani, S., Beghelli, S., and Rovatti, R. (2002). Identication of piecewise affine models in noisy environment. International Journal of Control
- [Frank, 1990] Frank, P. M. (1990). Fault diagnosis in dynamic systems using analytical and knowledge based redundancy: A survey of some new results. Automatica, 26(3):459-474.
- [Gertler, 1988] Gertler, J. (1988). Survey of model-based failure detection and isolation in complex plants. IEEE Control System Magazine, pages 3-11.
- [Gertler, 1998] Gertler, J. (1998). Fault Detection and Diagnosis in Engineering Systems. Marcel Dekker, New York.

References (Cont'd)

- [Gustafson and Kessel, 1979] Gustafson, D. E. and Kessel, W. C. (1979). Fuzzy clustering with a variance covariance matrix. In Proc. IEEE CDC'79, pages 161-166, San Diego, CA, USA.
- [Himmelblau, 1978] Himmelblau, D. M. (1978). Fault Diagnosis in Chemical and Petrochemical Processes. Elsevier. Amsterdam.
- [Isermann, 1984] Isermann, R. (1984). Process fault detection based on modelling and estimation methods: A survey. Automatica, 20(4):387-404.
- [Isermann, 1997] Isermann, R. (1997). Supervision, fault detection and fault diagnosis methods: an introduction. Control Engineering Practice, 5(5):639-652.
- [Isermann, 1998] Isermann, R. (1998). On fuzzy logic applications for automatic control, supervision and fault diagnosis. IEEE Trans. on Sys. Man. and Cyber. Part A: Sys. & Humans, 28(2):221-235.
- [Isermann and Balle, 1997] Isermann, R. and Balle, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. Control Engineering Practice, 5(5):709-719.
- [Jazwinski, 1970] Jazwinski, A. H. (1970). Stochastic processes and filtering theory. Academic Press, New York.
- [Leontaritis and Billings, 1985b] Leontaritis, I. and Billings, S. A. (1985b). Input-output parametric models for non{linear systems part II: stochastic non-linear systems. Int. J. Control, 41(2):329-344.

References (Cont'd)

- [Leontaritis and Billings, 1985a] Leontaritis, I. and Billings, S. A. (1985a). Input-output parametric models for non-linear systems part I: deterministic non-linear systems. Int. J. Control, 41(2):303-328.
- [Liu and Patton, 1998] Liu, G. P. and Patton, R. J. (1998). Eigenstructure Assignment for Control System Design. John Wiley & Sons, England.
- [Ljung, 1999] Ljung, L. (1999). System Identication: Theory for the User. Prentice Hall, Englewood Clis, N.J., second edition.
- [Lou et al., 1986] Lou, X., Willsky, A., and Verghese, G. (1986). Optimal robust redudancy relations for failure detection in uncertainty systems. Automatica, 22(3):333-344.
- [Luenberger, 1971] Luenberger, D. G. (1971). An introduction to observers. IEEE Transactions on Automatic Control, AC-16(6):596-602.
- [Luenberger, 1979] Luenberger, D. G. (1979). Introduction to Dynamic System: Theory, Models and Application. John Wiley and Son, New York.
- [Mamdani, 1976] Mamdani, E. (1976). Advances in the linguistic synthesis of fuzzy controllers. Int. J. Man-Machine Studies, 8:669-678.
- [Massoumnia et al., 1989] Massoumnia, M., Verghese, G. C., and Willsky, A. S. (1989). Failure detection and identication. IEEE Trans. Automat. Contr., 34:316-321.
- [Massoumnia, 1986] Massoumnia, M. A. (1986). A geometric appoach to failure detection and identication in linear systems. PhD thesis, Massachusetts Institute of Technology, Massachusetts, USA.

References (Cont'd)

- [Napolitano et al., 1998] Napolitano, M. R., Widon, D. A., Casanova, J. L., Innocenti, M., and Silvestri, G. (1998). Kalman lters and neural-networks schemes for sensor validation in flight control system. IEEE Trans. on Control System Technology, 6(5):596-611.
- [Nelles, 2001] Nelles, O. (2001). Nonlinear System Identication. Springer-Verlag Berlin Heidelberg, Germany.
- [Patton and Chen, 1994a] Patton, R. and Chen, J. (1994a). A review of parity space approaches to fault diagnosis for aerospace systems. AIAA Journal of Guidance, Control & Dynamics, 17(2):278-285.
- [Patton et al., 1989] Patton, R. J., Frank, P. M., and Clark, R. N., editors (1989). Fault Diagnosis in Dynamic Systems, Theory and Application. Control Engineering Series. Prentice Hall, London.
- [Patton et al., 2000] Patton, R. J., Frank, P. M., and Clark, R. N., editors (2000). Issues of Fault Diagnosis for Dynamic Systems. Springer-Verlag, London Limited.
- [Rovatti et al., 2000] Rovatti, R., Fantuzzi, C., and Simani, S. (2000). High-speed DSP-based implementation of piecewise affine and piecewise-quadratic fuzzy systems. The Signal Processing Journal, 80(6):951-963.
- [Simani and Fantuzzi, 2000] Simani, S. and Fantuzzi, C. (2000). Fault diagnosis in power plant using neural networks. International Journal of Information Sciences, 127(3-4):125-136. Special Issue: Applications to Intelligent Manufacturing and Fault Diagnosis: PART 1 - Fault Diagnosis.

References (Cont’d)

- [Simani et al., 2000a] Simani, S., Fantuzzi, C., and Beghelli, S. (2000a). Diagnosis techniques for sensor faults of industrial processes. IEEE Transactions on Control Systems Technology, 8(5):848-855.
- [Simani et al., 2002] Simani, S., Fantuzzi, C., and R. J. Patton (2000a). Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques. Springer Verlag; 1st edition (July 24, 2006) ISBN-10: 1852336854 ISBN-13: 978-1852336851
- [Simani et al., 1999c] Simani, S., Fantuzzi, C., Rovatti, R., and Beghelli, S. (1999c). Parameter identication for piecewise linear fuzzy models in noisy environment. International Journal of Approximate Reasoning, 1-2(22):149-167.
- [Takagi and Sugeno, 1985] Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its application to modeling and control. IEEE Transaction on System, Man and Cybernetics, SMC-15(1):116-132.
- [Willsky, 1976] Willsky, A. S. (1976). A survey of design methods for failure detection in dynamic systems. Automatica, 12(6):601-611.
- [Wunnenberg, 1990] Wunnenberg, J. (1990). Observer-based fault detection in dynamic systems. PhD thesis, University of Duisburg, Duisburg, Germany.
- [Wunnenberg and Frank, 1987] Wunnenberg, J. and Frank, P. M. (1987). Sensor fault detection via robust observer. In System Fault Diagnosis, Reliability, and Related Knowledge-Based Approaches, volume 1, pages 147-160. S. Tzafestas et al edition.
- [Xie and Soh, 1994] Xie, L. and Soh, Y. C. (1994). Robust Kalman ltering for uncertain systems. Systems and Control Letters, 22:123-129.

References (Cont'd)

- [Xie et al., 1994] Xie, L., Soh, Y. C., and de Souza, C. E. (1994). Robust Kalman filtering for uncertain discrete-time systems. IEEE Transaction on Automatic Control, 39:1310-1314.
- [Zhang and Morris, 1996] Zhang, J. and Morris, J. (1996). Process modelling and fault diagnosis using fuzzy neural networks. Fuzzy Sets and Systems, 79(1):127-140.
- [Zhou et al., 1996] Zhou, K., Doyle, J. C., and Glover, K. (1996). Robust and Optimal Control. Prentice Hall, New Jersey.
- [Simani and Fantuzzi, 2006] Simani, Silvio; Fantuzzi, Cesare. Dynamic system identification and model-based fault diagnosis of an industrial gas turbine prototype. Mechatronics Volume: 16, Issue: 6, July, 2006, pp. 341-363
- [Simani, 2007] S. Simani. Fault diagnosis of a simulated industrial gas turbine via identification approach. International Journal of Adaptive Control and Signal Processing. Volume 21, Issue 4, Date: May 2007, Pages: 326-353.
- [Bonfè, et al., 2006] M. Bonfè, P. Castaldi, W. Geri, S. Simani Fault detection and isolation for on-board sensors of a general aviation aircraft. International Journal of Adaptive Control and Signal Processing. Volume 20, Issue 8, Date: October 2006, Pages: 381-408
- [Simani, 2005] Simani, S.; Identification and fault diagnosis of a simulated model of an industrial gas turbine. IEEE Transactions on Industrial Informatics, Volume 1, Issue 3, Aug. 2005 Page(s):202 – 216.