Model–Based Fault Detection and Isolation: Linear Polynomial Methods for FDI with Applications

Silvio Simani

Dipartimento di Ingegneria, Università di Ferrara
Via Saragat 1, 44122 Ferrara - ITALY
E–mail: silvio.simani@unife.it
URL: http://www.ing.unife.it/simani

General Overview

- **Growing demand for higher reliability**
- **Actuators & sensors are the most important components**

  ⇒ hardware & software (analytical) redundancy

- **Hardware redundancy methods use multiple lanes of sensors, computers & software to measure and/or control a particular variable**

  ⇒ multiple hardware redundancy is harder to achieve

FDI Overview

- **Analytical redundancy makes use of a mathematical model of the monitored process**
  - ⇒ *model–based* approach to Fault Detection & Isolation (FDI)
  - ⇒ The model–based FDI is normally implemented in software form as a computer algorithm
- **Model–based methods use a model of the monitored system to produce residuals**
  - ⇒ The system cannot be described accurately by a mathematical model

Model–Based FDI

- **In real complex systems modelling uncertainty arises inevitably for example process noise, parameter variations & modelling errors**
  - ⇒ The detection of incipient faults presents a challenge to model–based FDI techniques
  - ⇒ unseparatable mixture between fault effects & modelling uncertainty

Robust Model–Based FDI

- **Optimisation to minimise the effect of modelling uncertainty, whilst maximising some fault effects**
- **Intelligent techniques, adaptive methods**
- **Robust FDI is still an open problem for further research**

Presentation Outline

- **Fault Detection & Isolation**

  $\Rightarrow$ Linear multivariable systems with additive faults & disturbances
- **Input–output polynomial forms**
- **Residual generator subspace basis $\hookrightarrow$ dynamic filter computation**

  $\Rightarrow$ Analytical Derivation, Fault & noise sensitivity analysis
- **Application: sensor & actuator faults of dynamic processes**

  $\Rightarrow$ Linear & linearised model

Introduction: System Modelling

- **Fault diagnosis for dynamic processes**

  $\Rightarrow$ disturbance de–coupling techniques

- **Input–output descriptions of the monitored system**

  $\Rightarrow$ model disturbance term takes into account system unknown inputs

Introduction: Fault Diagnosis

- **Set of parity relations insensitive to the disturbance term**

  $\Rightarrow$ residual or symptom signals

- **Simulated process: power plant & small aircraft**

  $\Rightarrow$ sensor & actuator faults

Analytical Redundancy & Model–based Approach

- **Main problem: modelling uncertainties**

  $\Rightarrow$ unavoidable in real industrial systems

- **System description $\hookrightarrow$ input–output linear model**

  $\Rightarrow$ by modelling or identification procedures

  $\Rightarrow$ the disturbance term describes unknown (or non–measurable) inputs of the real process

Model–based Approach

- **Parity relations design for residual generation**
- **Detection of faults affecting input & output process measurements**
- **Residual generator: insensitive to disturbance signals**

Presentation Topics (1)

- **The system under diagnosis is modelled in terms of input–output polynomial description**
- **The design of disturbance de–coupled residual generators is reduced to the determination of the null–space of a specific polynomial matrix**

  $\Rightarrow$ The use of input–output forms allows to design the analytical description for the disturbance de–coupled residual generators

Presentation Topics (2)

- **These dynamic fault detection filters, organised into bank structures, are able to achieve *Fault Isolation* properties**

  $\Rightarrow$ An appropriate choice of their parameters allows to maximise robustness with respect to both measurement noise & modelling errors, while optimising fault sensitivity characteristics

- **The proposed FDI approach has been applied to 2 dynamic process models:**

  $\Rightarrow$ *(i)* Power Plant of "Pont sur Sambre"

  $\Rightarrow$ *(ii)* Piper PA30

Presentation Topics (3)

- **The residual generators have been designed on the basis of the linear & linearised models**
  - ⇒ Experiments with data from linear & non–linear simulators implemented in Matlab/Simulink® environment
- **An important aspect of the approach to FDI suggested is the simplicity of structure of the technique used to generate the residual functions for FDI**
  - ⇒ In comparison with traditional schemes *e.g.* based on banks of unknown input observers (UIO) & Kalman filters

Talk Structure (1)

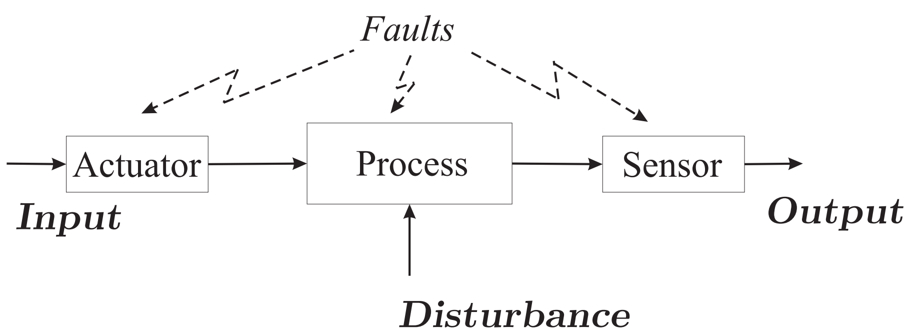
- **Mathematical description of the monitored system is outlined**
- **The approach exploited for the design of residual generators is described**
- **Structural characteristics of such filters are also explained**
  - ⇒ How to achieve disturbance de–coupling, sensitivity optimisation of the residual functions & robustness with respect to measurement noise & modelling errors

Talk Structure (2)

- **Problem of the design of banks of residual generators for the Isolation of faults affecting the input & the output sensors**
- **Application to 2 dynamic process models with some numerical results**

General Scenario

- **Monitored system**



$\Rightarrow$ additive fault & disturbance occurrence

$\Rightarrow$ fault–free & faulty data sequences

Mathematical Description

$$P(s)\,y(t) = Q(s)\,u(t)$$

- s is the derivative operator
- $P(s)$ & $Q(s)$ are polynomial matrices with dimension $(m \times m)$ & $(m \times \ell)$ respectively, with $P(s)$ nonsingular.
- $u(t) \in \Re^{\ell}$ & $y(t) \in \Re^{m}$, the input & output vectors of the considered multivariable system
- $u(t) \equiv \mathcal{L}\big[u(t)\big](s)$ & $y(t) \equiv \mathcal{L}\big[y(t)\big](s)$

Model Properties

- **Models of type $P(s)\,y(t) = Q(s)\,u(t)$ can be frequently found in practice**
  - $\Rightarrow$ applying well–known physical laws
  - $\Rightarrow$ for describing the input–output dynamical links
- **powerful tool when the knowledge of the system state does not play a direct role**

Model Properties

$$P(s)\,y(t) = Q(s)\,u(t)$$

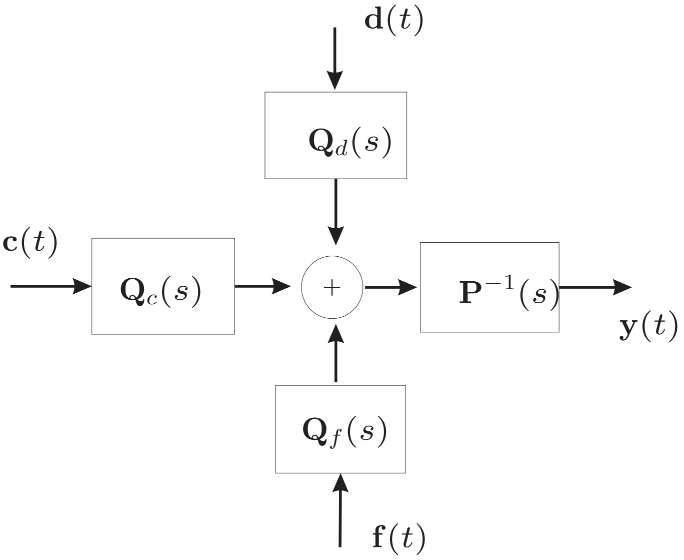
- **Powerful tool when the knowledge of the system state does not play a direct role**
  - $\Rightarrow$ see, *e.g.* residual generator design, identification, de–coupling, output controllability,...
- **Algorithms to transform state–space models to equivalent input–output polynomial representations & vice versa are available [Beghelli, Guidorzi, Castaldi, Soverini]**

Mathematical Description

$$P(s)\,y(t) = \begin{bmatrix} Q_c(s) & Q_d(s) & Q_f(s) \end{bmatrix} \begin{bmatrix} c(t) \\ d(t) \\ f(t) \end{bmatrix},$$

- The equivalent representation $\left\{\tilde{P}(s),\, \tilde{Q}(s)\right\}$ is a $canonical$ input–output form
- Link between $\left\{P(s),\, Q(s)\right\}$ & $\left\{\tilde{P}(s),\, \tilde{Q}(s)\right\}$ via an unimodular matrix $\mathbf{M}(s)$
- $c(t) \equiv \mathcal{L}\big[c(t)\big](s)$, $d(t) \equiv \mathcal{L}\big[d(t)\big](s)$ & $f(t) \equiv \mathcal{L}\big[f(t)\big](s)$

Model Description for the Monitored System



Input–Output Canonical Form Properties

$$\left\{\begin{array}{lcll} \deg \tilde{p}_{ii}(s) & > & \deg \tilde{p}_{ji}(s) & i \neq j \\ \deg \tilde{p}_{ii}(s) & > & \deg \tilde{p}_{ij}(s) & j > i \\ \deg \tilde{p}_{ii}(s) & \geq & \deg \tilde{p}_{ij}(s) & j < i \\ \deg \tilde{p}_{ii}(s) & \geq & \deg \tilde{q}_{ij}(s) & \end{array}\right.$$

- with the polynomials $\tilde{p}_{ii}(s)$ monic.
- Integers $\nu_i = \deg \tilde{p}_{ii}$ $(i = 1, \ldots, m)$ equal the corresponding row–degrees.
- Integers ν_i are the ordered set of Kronecker invariants associated to the pair $\{\tilde{A}, \tilde{C}\}$ of every observable realization of $\{\tilde{P}(s), \tilde{Q}(s)\}$.

State–Space *Canonical* Form Properties

- **Canonical state–space $(\tilde{\mathrm{A}}, \tilde{\mathrm{C}})$ models** :

$$\tilde{\mathbf{A}} = [\tilde{\mathbf{A}}_{ij}]\ , \text{ with } \tilde{\mathbf{A}}_{ii} = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ \times_{ii1} & \times_{ii2} & \dots & \times_{ii\nu_i} \end{bmatrix}_{(\nu_i \times \nu_i)},$$

$$\tilde{\mathbf{A}}_{ij} = \begin{bmatrix} 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & & & & & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 \\ \times_{ij1} & \dots & \times_{ij\nu_{ij}} & 0 & \dots & 0 \end{bmatrix}_{(\nu_i \times \nu_j)}$$

State–Space *Canonical* Form Properties

$$\tilde{\mathbf{C}} = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \dots & 0 & 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ \vdots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & 0 & \dots & 0 \end{bmatrix}$$

$\Rightarrow$ where the 1's entries in the matrix $\tilde{\mathbf{C}}$ are in the column 1, $(\nu_i + 1)$, $\dots$, $(\nu_1 + \dots + \nu_{m-1} + 1)$

Filter Design for Fault Diagnosis

- **A general linear residual generator is a filter of type:**

$$R(s)\,r(t) = S_y(s)\,y(t) + S_c(s)\,c(t)$$

⇒ $r(t)$ signal is a scalar

⇒ Faults are neglected here!

Residual Generator Computation Problem

⇒ Design the residual generator ($r(t)$ scalar)

$$R(s)\,r(t) = S_y(s)\,y(t) + S_c(s)\,c(t)\ .$$

⇒ for the input–output model (fault–free conditions)

$$P(s)\,y(t) - Q_c(s)\,c(t) = Q_d(s)\,d(t)$$

Residual Generator Computation Problem

⇒ $L(s)$ is a polynomial row belonging to the left null–space of $Q_d(s)$

⇒ Left null–space of $Q_d(s)$ is $\mathcal{N}_\ell(Q_d(s))$

$$L(s) \in \mathcal{N}_\ell(Q_d(s))$$

⇒ *i.e.* $L(s)\, Q_d(s) = 0$

⇒ *e.g.* 2 outputs & 1 disturbance

Residual Generator Formulation (with faults)

$$P(s)\, y(t) - Q_c(s)\, c(t) - Q_f(s)\, f(t) = Q_d(s)\, d(t) \hookrightarrow$$

$$L(s)\, P(s)\, y(t) - L(s)\, Q_c(s)\, c(t) = L(s)\, Q_f(s)\, f(t)$$

$$\left\{ \begin{array}{lcl} S_y(s) & = & L(s)\, P(s) \\ S_c(s) & = & -L(s)\, Q_c(s) \\ R(s) & = & (1 + \tau_1\, s)(1 + \tau_2\, s) \ldots (1 + \tau_{n_f}\, s) = \\ & = & a_1\, s^{n_f} + a_2\, s^{n_f - 1} + \cdots + a_{n_f}\, s + 1 \end{array} \right.$$

Residual Generator Formulation

- n_f is the maximal row–degree of the pair $\{L(s)\,P(s),\, L(s)\,Q_c(s)\}$.
- Without faults: $R(s)\,r(t) = L(s)\,P(s)\,y(t) - L(s)\,Q_c(s)\,c(t) = 0$
- With faults: $R(s)\,r(t) = L(s)\,P(s)\,y(t) - L(s)\,Q_c(s)\,c(t) =$
  $= L(s)\,Q_f(s)\,f(t)$

Residual Generator Properties

- ***Bounds* for the order n_f of the residual generator**

  ⇒ n_f is the maximal row–degree of the pair $\{L(s)\,P(s),\, L(s)\,Q_c(s)\}$

  ⇒ n_f is the degree of $R(s)$ (filter causality)

$$\nu_{\min} \;\le\; n_f \;\le\; \left(\ell_d + 1\right)\nu_{\max}$$

  ⇒ $\nu_{\min}$ & $\nu_{\max}$ are the minimal & the greatest Kronecker invariant, respectively.

  ⇒ [Frisk & Niberg, Bonfè, *et al.*]

Fault Description

- **Fault modelling**

$$\begin{cases} \mathbf{c}^*(t) & = & \mathbf{c}(t) + \mathbf{f}_c(t) \\ \mathbf{y}^*(t) & = & \mathbf{y}(t) + \mathbf{f}_y(t) \end{cases}$$

$\Rightarrow$ $\mathbf{f}_c(t)$ & $\mathbf{f}_y(t)$: actuator & sensor additive faults

$\Rightarrow$ *e.g.* step, ramp, intermittent signals

- **$\mathbf{c}(t)$, $\mathbf{y}(t)$: fault–free signals**

- **$\mathbf{c}^*(t)$, $\mathbf{y}^*(t)$: input & output measurements**

Fault Modelling

$$\begin{cases} \mathbf{c}^*(t) & = & \mathbf{c}(t) + \mathbf{f}_c(t) \\ \mathbf{y}^*(t) & = & \mathbf{y}(t) + \mathbf{f}_y(t) \end{cases}$$

$$\begin{aligned} R(s)\, r(t) &= L(s)\, P(s)\, y^*(t) - L(s)\, Q_c(s)\, c^*(t) \\ &= L(s)\, Q_c(s)\, f_c(t) - L(s)\, P(s)\, f_o(t) \\ &= \left[L(s)\, Q_c(s) | - L(s)\, P(s)\right] \left[\begin{array}{c} f_c(t) \\ f_o(t) \end{array} \right] \\ &= L(s) \left[Q_c(s) | - P(s)\right] \left[\begin{array}{c} f_c(t) \\ f_o(t) \end{array} \right] = L(s)\, Q_f(s)\, f(t) \end{aligned}$$

Fault Modelling

$$\begin{cases} \mathbf{c}^*(t) & = & \mathbf{c}(t) + \mathbf{f}_c(t) \\ \mathbf{y}^*(t) & = & \mathbf{y}(t) + \mathbf{f}_y(t) \end{cases}$$

$R(s)\,r(t) = L(s)\,Q_f(s)\,f(t) \hookrightarrow$ ideal conditions!

with $f(t) = \begin{bmatrix} f_c(t) \\ f_o(t) \end{bmatrix}$

$Q_f(s) = \big[Q_c(s) | - P(s) \big]$

Residual function $r(t)$ for *Fault Detection*

- **Fault–free & faulty situations**
- **Residual function $r(t)$ comparison**

  $\Rightarrow$ fixed threshold ε

  $\Rightarrow$ *threshold logic*:

$$\begin{cases} |r(t)| & \leq & \varepsilon & \text{for fault–free case,} \\ |r(t)| & > & \varepsilon & \text{for faulty cases.} \end{cases}$$

- **Analysis of different residual functions $r_i(t)$**

Residual Generator Design

$$R(s)\,r(t) = \underbrace{L(s)\,P(s)\,y(t) - L(s)\,Q_c(s)\,c(t)}_{=0} + \underbrace{L(s)\,Q_f(s)\,f(t)}_{\text{faulty case}}$$

$$R(s)\,r(t) = \underbrace{L(s)\,Q_f(s)\,f(t)}_{\text{faulty case}}$$

- **Ideal conditions**

  $\Rightarrow$ Easy solution

Residual Generator Design

- **Optimisation Approach (real conditions):**

$$R(s)\,r(t) = \underbrace{L(s)\,P(s)\,y(t) - L(s)\,Q_c(s)\,c(t)}_{\text{modelling error} \neq 0} + \underbrace{L(s)\,Q_f(s)\,f(t)}_{\text{faulty case}}$$

$\Rightarrow$ $R(s)$ & $L(s)$ $\hookrightarrow$ optimal selection

$$R(s) = (1 + \tau_1\,s)(1 + \tau_2\,s)\ \ldots\ (1 + \tau_{n_f})$$

- **Fault sensitivity maximisation: max of** $R^{-1}(s)\,L(s)\,Q_f(s)\,f(t)$

Residual Generator Parameter Optimisation: $L(s)$

- **Parameters of $L(s)$**

  $\Rightarrow$ $b_i(s)(i = 1, 2, \ldots, m - \ell_d)$ rows of a basis $B_{(m-\ell_d)\times m}(s)$ of the $\mathcal{N}_\ell(Q_d(s))$

  $\Rightarrow$ Assumption: $m - \ell_d > 1$

  $\Rightarrow$ then, $L(s) = \sum_{i=1}^{m-\ell_d} k_i b_i(s)$ & k_i maximising:

Residual Generator Parameter Optimisation: $L(s)$

$$\lim_{s\to 0} \frac{1}{R(s)} [\sum_{i=1}^{m-\ell_d} k_i b_i(s)] \, Q_f(s) = [\sum_{i=1}^{m-\ell_d} k_i b_i(0)] \, Q_f(0) \text{ with } \sum_{i=1}^{m-\ell_d} k_i^2 = 1$$

- **Fault $f(t)$ step–function of magnitude F**

$$\lim_{t\to\infty} r(t) = \lim_{s\to 0} s \frac{L(s)Q_f(s)}{R(s)} \frac{F}{s} = [\sum_{i=1}^{m-\ell_d} k_i b_i(0)] \, Q_f(0) \, F.$$

Residual Generator Parameter Optimisation: $R(s)$

- **Location of the roots of the polynomial $R(s)$**
  - $\Rightarrow$ Influences the transient characteristics (maximum overshoot, delay time, rise time, settling time, etc.) of the fault detection filter
  - $\Rightarrow$ Optimisation of fault detection time, false alarm & missed fault rates

Residual Generator Parameter Optimisation: $R(s)$

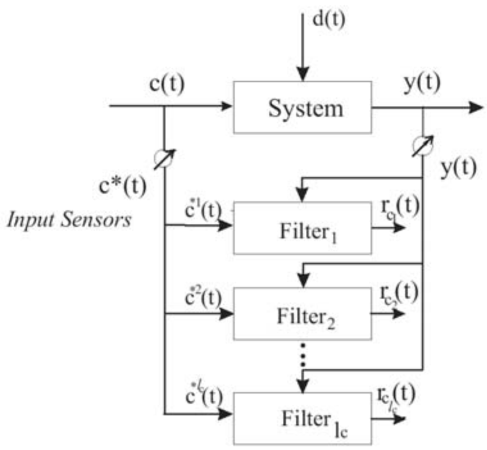
$$\frac{|G_f(j\omega)|^2}{|G_r(j\omega)|^2} = 1, (\text{where } \omega \text{ belongs to a given frequency range})$$

- $\Rightarrow$ reference transfer function $G_r(s)$
- $\Rightarrow$ $G_f(s) = L(s)Q_f(s)/R(s)$

Fault Isolation Introduction

- **After fault detection...**
- **Problem of the design of banks of residual generators**
  - ⇒ Isolation of faults affecting input & output sensors
  - ⇒ The disturbance de–coupling method suggested previously is exploited
  - ⇒ It is assumed that $m > \ell_d + 1$

Input Sensor Fault Isolation



Scheme for input sensor fault isolation.

Input Sensor Fault Isolation

- **The number of these generators is equal to the number ℓ_c of system control inputs**
- **The i–th device ($i = 1, \ldots, \ell_c$) is driven by all but the i–th input & by all the outputs of the system**
- **A fault on the i–th input sensor affects all but the i–th residual generator**

  ⇒ $c^{*i}(t)$ represents the $\ell_c - 1$ dimensional vector obtained by deleting from $c^*(t)$ the i–th component

Silvio Simani Bologna, Italy. November, 2011

Input Sensor Fault Isolation

- $c^*(t) = c(t) + f_{c_i}(t)$

  ⇒ with $f_{c_i}(t) = \begin{bmatrix} 0 & \ldots & 0 & h_{c_i}(t) & 0 & \ldots & 0 \end{bmatrix}^T$

  ⇒ $c^{*i}(t) = c^i(t)$ when the fault on the i–th input sensor $h_{c_i}(t)$ is considered

Input Sensor Fault Isolation

- **In these conditions:**

$$P(s)\,y(t) = Q_c(s)\,c(t) + Q_d(s)\,d(t) + q_{c_i}(s)\,h_{c_i}(t)\,,$$

$\Rightarrow$ $q_{c_i}(s)$ represents the i–th column of the matrix $Q_c(s)$

- **By multiplying by the matrix $L_{c_i}(s)$**

  $\Rightarrow$ $L_{c_i}(s)$ is a row vector belonging to the basis for the left null space of the matrix $\left[Q_d(s)\,|\,q_{c_i}(s)\right]$

Input Sensor Fault Isolation

- **If $L_{c_i}(s)$ is a row vector belonging to the basis for the left null space of the matrix $\left[Q_d(s)\,|\,q_{c_i}(s)\right]$**

  $\Rightarrow$ $Q_c^i(s)$ is the matrix obtained by deleting from $Q_c(s)$ the i–th column:

- **The i–th filter becomes:**

$$R_{c_i}(s)\,r_{c_i}(t) = L_{c_i}(s)\,P(s)\,y(t) - L_{c_i}(s)\,Q_c^i(s)\,c^{*i}(t) = 0,$$

$\Rightarrow$ while, for the j–th filter, with $j \neq i$:

$$R_{c_j}(s)\,r_{c_j}(t) = L_{c_j}(s)\,P(s)\,y(t) - L_{c_j}(s)\,Q_c^j(s)\,c^{*j}(t) = L_{c_j}(s)\,q_{c_i}(s)\,h_{c_i}(t)$$

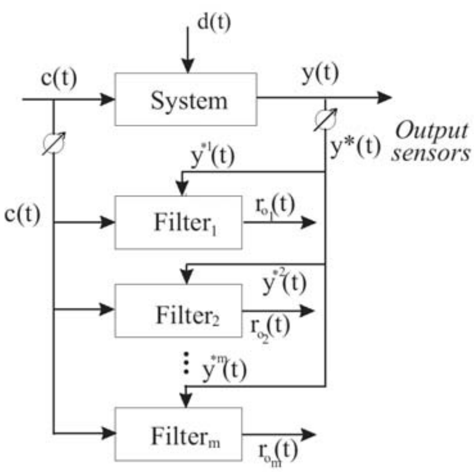
Residual Generation for Fault Isolation

⇒ $R_{c_i}(s)$ & $R_{c_j}(s)$ are arbitrary polynomials with all the roots with negative real part

- **In a similar way, *output sensor* isolation**
- **All the input sensors & the remaining output sensors are fault–free**

⇒ a bank of residual generator filters is used

Output Sensor Fault Isolation



Bank of residual generators for output sensor fault isolation

Output Sensor Fault Isolation

- **The number of these generators is equal to the number m of system outputs**
- **The i–th device ($i = 1, \dots, m$) is driven by all but the i–th output & by all the inputs of the system**
  - $\Rightarrow$ A fault on the i–th output sensor affects all but the i–th residual generator
  - $\Rightarrow$ $y^{*i}(t)$ represents the $m-1$ dimensional vector obtained by deleting from $y^*(t)$ the i–th component

Output Sensor Fault Isolation

- $y^*(t) = y(t) + f_{o_i}(t)$
  - $\Rightarrow$ with $f_{o_i}(t) = \left[\begin{array}{ccccccc} 0 & \dots & 0 & h_{o_i}(t) & 0 & \dots & 0 \end{array} \right]^T$
- $P(s)\,y(t) \;= Q_c(s)\,c(t) \,+\, Q_d(s)\,d(t) - p_i(s)\,h_{o_i}(t)$
  - $\Rightarrow$ where $p_i(s)$ represents the i–th column of the matrix $P(s)$
  - $\Rightarrow$ $y^{*i}(t) \,=\, y^i(t)$ when a fault on the i–th output sensor $h_{o_i}(t)$ is considered

Output Sensor Fault Isolation

⇒ By multiplying by the matrix $L_{o_i}(s)$

⇒ $L_{o_i}(s)$ is a row vector belonging to the basis for the left null space of the matrix $\left[Q_d(s)\,|\,p_i(s)\right]$

⇒ $P^i(s)$ the matrix obtained by deleting from $P(s)$ the i–th column

Output Sensor Fault Isolation

- **The equation of the i–th filter becomes:**

$$R_{o_i}(s)\,r_{o_i}(t) = L_{o_i}(s)\,P^i(s)\,y^{*i}(t) - L_{o_i}(s)\,Q_c(s)\,c(t) \;=\; 0,$$

- **while, for the j–th filter, with $j \neq i$:**

$$R_{o_j}(s)\,r_{o_j}(t) = L_{o_j}(s)\,P^j(s)\,y^{*j}(t) - L_{o_j}(s)\,Q_c(s)\,c(t) = -L_{o_j}(s)\,p_i(s)\,h_{o_i}(t).$$

⇒ $R_{o_i}(s)$ & $R_{o_j}(s)$ are arbitrary polynomials whose roots have negative real part

Sensor Fault Isolation Summary

- **Summary of the FDI capabilities of the presented schemes**
- **"*Fault Signatures*" in case of a single fault in each input & output sensor**
- **The residuals which are affected by faults are marked with the presence of '1' in the correspondent table entry**
- **An entry '0' means that the fault does not affect the correspondent residual**

Sensor Fault Isolation Table

Fault signatures

Residual / Fault	f_{c_1}	f_{c_2}	$\dots$	$f_{c_{\ell_c}}$	f_{o_1}	f_{o_2}	$\dots$	f_{o_m}
r_{c_1}	0	1	$\dots$	1	1	1	$\dots$	1
r_{c_2}	1	0	$\dots$	1	1	1	$\dots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$r_{c_{\ell_c}}$	1	1	$\dots$	0	1	1	$\dots$	1
r_{o_1}	1	1	$\dots$	1	0	1	$\dots$	1
r_{o_2}	1	1	$\dots$	1	1	0	$\dots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{o_m}	1	1	$\dots$	1	1	1	$\dots$	0

Sensor Fault Isolation Conditions

- **When not all the elements out of the main diagonal of the table are '1's**

  $\Rightarrow$ the fault isolation is still feasible if the columns of the fault signature table are all different from each other

- **When $m - (\ell_d + 1) > 1$**

  $\Rightarrow$ all the bases of the left null space of the matrices $\left[Q_d(s)\,|\,q_{c_i}(s)\right]$ & $\left[Q_d(s)\,|\,p_i(s)\right]$ have dimension bigger than 1

  $\Rightarrow$ the degrees of freedom in the choice of the vectors $L_{c_i}(s)$ & $L_{o_i}(s)$ belonging to the left null space can be exploited

Sensor Fault Isolation Conditions

$\Rightarrow$ All the elements out of the main diagonal on the Table are '1's when:

- For $i = 1, \ldots, \ell_c$, the column vectors of the matrix $Q_c^i(s)$ & the column vectors of the matrix $P(s)$ are not orthogonal with the row vector $L_{c_i}(s)$.

- For $j = 1, \ldots, m$, the column vectors of the matrix $P^j(s)$ & the column vectors of the matrix $Q_c(s)$ are not orthogonal with the row vector $L_{o_j}(s)$.

Application Examples

- **1. $P(z)$ & $Q(z)$ matrices are known**

  $\Rightarrow R(z)\,r(t) = S_y(z)\,y(t) + S_c(z)\,c(t)$

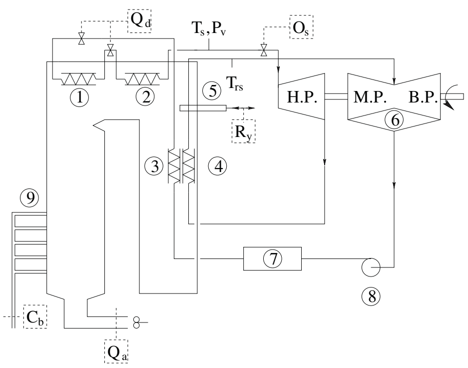
- **2. $P(z)$ & $Q(z)$ matrices are known + measurement noise**

  $\Rightarrow R(z)\,r(t) = S_y(z)\,y(t) + S_c(z)\,c(t)$ computation & sensitivity analysis

- **3. $\{u(t), y(t)\}$ multivariable non–linear process $\hookrightarrow$ $P(s)$ & $Q(s)$ from linearisation**

  $\Rightarrow R(s)\,r(t) = S_y(s)\,y(t) + S_c(s)\,c(t)$ computation & $R(s)$ optimisation

Fault Detection in an Industrial Process

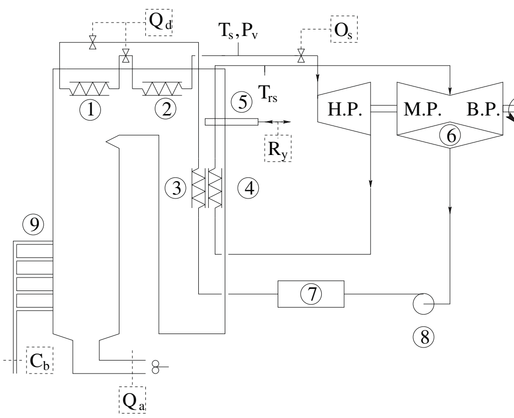


1. super heater (radiation);
2. super heater (convection);
3. super heater;
4. reheater;
5. dampers;
6. condenser;
7. drum;
8. water pump;
9. burner.

Process Description

- **120MW power plant of Pont–sur–Sambre**
  - ⇒ Double–shaft industrial gas turbine
  - ⇒ working in parallel with the electrical mains
- **3 major components:**
  - ⇒ the *reactor*, *turbine*, & *condenser*

Process Description



$u_1(t)$:	C_b	gas flow
$u_2(t)$:	O_s	turbine valves opening
$u_3(t)$:	Q_d	super heater spray flow
$u_4(t)$:	R_y	gas dampers
$u_5(t)$:	Q_a	air flow
$y_1(t)$:	P_v	steam pressure
$y_2(t)$:	T_s	main steam temperature
$y_3(t)$:	T_{rs}	reheat steam temperature

Process Matrices: $\tilde{\mathrm{P}}(z)$ & $\tilde{\mathrm{Q}}(z)$

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} z^4 - 1.0774z^3 - 0.1846z^2 + 0.1004z + 0.1884 & -0.1662z^2 + 0.1932z - 0.0250 & -0.1239z + 0.1086 \\ -0.1021z^3 + 0.2958z^2 - 0.3528z + 0.1710 & z^3 - 1.6750z^2 + 0.8292z - 0.0852 & -0.0180z + 0.0156 \\ 0.1096z^2 - 0.0812z + 0.0142 & -0.2411z^2 + 0.5470z - 0.3544 & z^2 - 1.5875z + 0.6624 \end{bmatrix},$$

$$\tilde{\mathbf{Q}}(z) = \begin{bmatrix} 0.02 + 0.06z + 0.08z^2 - 0.1z^3 - 0.1z^4 & 0.02 + 0.06z - 0.1z^2 + 0.009z^3 & 0.05 - 0.05z - 0.03z^2 \\ 0.01 - 0.02z - 0.1z^2 - 0.1z^3 + 0.2z^4 & 0.1 - 0.1z + 0.1z^2 + 0.01z^3 & -0.1 + 0.1z + 0.04z^2 \\ 0.02 - 0.005z + 0.003z^2 - 0.003z^3 - 0.02z^4 & -0.02 - 0.09z + 0.04z^2 + 0.01z^3 & 0.09 + 0.01z - 0.01z^2 \\ -0.09 + 0.01z + 0.02z^2 + 0.08z^3 - 0.02z^4 & -0.01 + 0.01z - 0.02z^2 - 0.001z^3 & 0.01 - 0.1z + 0.01z^2 \\ 0.01 - 0.2z - 0.1z^2 + 0.9z^3 + 0.1z^4 & -0.1 - 0.3z + 0.03z^2 + 0.1z^3 & 0.04 - 0.1z - 0.2z^2 \end{bmatrix}^T,$$

$\Rightarrow$ Kronecker invariants: $\nu_1 = 4$, $\nu_2 = 3$, $\nu_3 = 2$

Power Plant Description: *Discrete–Time Model*

$$P(z)\, y(t) = Q(z)\, u(t)$$

- z is the unitary advance operator
- $P(z)$ & $Q(z)$ are polynomial matrices with dimension $(m \times m)$ & $(m \times \ell)$ respectively, with $P(s)$ nonsingular.
- $u(t) \in \Re^\ell$ & $y(t) \in \Re^m$, the input & output vectors of the considered discrete–time multivariable system $(t = 1, 2, \ldots, N)$
- $u(t) \equiv \mathcal{Z}\big[u(t)\big](z)$ & $y(t) \equiv \mathcal{Z}\big[y(t)\big](z)$

Residual Function & 1 Disturbance De–coupling

- $u_2(t)$ **represents the disturbance signal** $d_1(t)$

  $\Rightarrow \; \tilde{\mathbf{Q}}_d(z) = \left[\; \tilde{\mathbf{Q}}_2(z) \;\right]$

- **De–coupling of the input** $u_2(t)$

  $\Rightarrow \; \tilde{\mathbf{Q}}_c(z) = \left[\; \tilde{\mathbf{Q}}_1(z)\, \tilde{\mathbf{Q}}_3(z)\, \tilde{\mathbf{Q}}_4(z)\, \tilde{\mathbf{Q}}_5(z) \;\right]$

  $\Rightarrow$ $\hookrightarrow$ sensitive to $u_1(t)$, $u_3(t)$, $u_4(t)$ & $u_5(t)$

Residual Function & 1 Disturbance De–coupling

$\Rightarrow \; n_{f_1} = 4 \;\&\; n_{f_2} = 5$

$\Rightarrow$ Computation of the coefficients of the polynomials of the matrices $\mathbf{S}_c(z)$ & $\mathbf{S}_y(z)$

$\Rightarrow$ Null space of $\tilde{\mathbf{Q}}_d(z) \hookrightarrow$

$$\Rightarrow \; \mathbf{L}(z) = \begin{bmatrix} -0.029z - 0.19z^2 & -0.048z^2 \\ 0.46 + 0.62z + 0.49z^2 & -0.21z + 0.44z^2 \\ -0.11z + 0.35z^2 & 0.42 - 0.69z + 0.32z^2 \end{bmatrix}^T$$

Residual Function & 1 Disturbance De–coupling: $\mathbf{S}_c(z)$

$$\mathbf{S}_c(z) = \left[\ \mathbf{S}_{c_1}(z),\ \mathbf{S}_{c_3}(z),\ \mathbf{S}_{c_4}(z),\ \mathbf{S}_{c_5}(z)\ \right]$$

$$\mathbf{S}_{c_1}(z) = \left[\begin{array}{c} -0.0046z^4 + 0.0142z^3 - 0.0086z^2 - 0.0138z + 0.0113 \\ 0.0100z^5 - 0.0243z^4 - 0.0386z^3 - 1.8 \times 10^{-6}z^2 + 0.0292z + 0.0041 \end{array} \right],$$

$$\mathbf{S}_{c_3}(z) = \left[\begin{array}{c} -0.0023z^4 + 0.0030z^3 + 0.0047z^2 + 0.0016z - 0.0039 \\ 0.0087z^5 + 0.0321z^4 + 0.0209z^3 - 0.0067z^2 - 0.0115z - 0.0049 \end{array} \right],$$

Residual Function & 1 Disturbance De–coupling: $\mathbf{S}_c(z)$

$$\mathbf{S}_c(z) = \left[\ \mathbf{S}_{c_1}(z),\ \mathbf{S}_{c_3}(z),\ \mathbf{S}_{c_4}(z),\ \mathbf{S}_{c_5}(z)\ \right]$$

$$\mathbf{S}_{c_4}(z) = \left[\begin{array}{c} 0.0004z^4 - 0.0049z^3 + 0.0078z^2 - 0.0042z + 0.0008 \\ -0.0006z^5 - 0.0004z^4 + 0.0062z^3 - 0.0059z^2 + 0.0002z + 0.0009 \end{array} \right],$$

$$\mathbf{S}_{c_5}(z) = \left[\begin{array}{c} -0.0060z^4 + 0.0002z^3 + 0.0124z^2 - 0.0062z - 0.0015 \\ 0.0025z^5 + 0.0096z^4 + 0.0025z^3 - 0.0169z^2 - 0.0038z - 0.0019 \end{array} \right],$$

Residual Function & 1 Disturbance De–coupling: $\mathbf{S}_y(z)$

$$\mathbf{S}_y(z) = \left[\ \mathbf{S}_{y_1}(z),\ \mathbf{S}_{y_2}(z),\ \mathbf{S}_{y_3}(z)\ \right],$$

$$\mathbf{S}_{y_1}(z) = \begin{bmatrix} 0.0053z^4 - 0.0545z^3 + 0.1073z^2 - 0.0843z + 0.0269 \\ -0.0601z^5 - 0.0575z^4 + 0.1645z^3 - 0.0760z^2 + 0.0419z - 0.0045 \end{bmatrix},$$

$$\mathbf{S}_{y_2}(z) = \begin{bmatrix} -0.1173z^4 + 0.4365z^3 - 0.5235z^2 + 0.2431z - 0.0326 \\ 0.4615z^5 - 0.1553z^4 - 0.5843z^3 + 0.3770z^2 - 0.0025z - 0.0154 \end{bmatrix},$$

Residual Function & 1 Disturbance De–coupling: $\mathbf{S}_y(z)$

$$\mathbf{S}_y(z) = \left[\ \mathbf{S}_{y_1}(z),\ \mathbf{S}_{y_2}(z),\ \mathbf{S}_{y_3}(z)\ \right],$$

$$\mathbf{S}_{y_3}(z) = \begin{bmatrix} 0.1417z^4 - 0.4147z^3 + 0.4482z^2 - 0.2097z - 0.0349 \\ -0.1692z^4 + 0.3667z^3 - 0.2578z^2 + 0.0529z + 0.0040 \end{bmatrix}.$$

Residual Function & 1 Disturbance De–coupling: $\mathrm{L}(z)$

$$\mathbf{L}(z) = \begin{bmatrix} -0.029z - 0.19z^2 & -0.048z^2 \\ 0.46 + 0.62z + 0.49z^2 & -0.21z + 0.44z^2 \\ -0.11z + 0.35z^2 & 0.42 - 0.69z + 0.32z^2 \end{bmatrix}^T$$

- **It can be evaluated by means of the command `null` (Matlab Polynomial Toolbox) of the matrix $\tilde{\mathrm{Q}}_d(z)$**

Residual Function & 2 Disturbance De–coupling

- $u_2(t)$ **&** $u_4(t) \hookrightarrow$ **disturbance signals** $d_1(t)$ **&** $d_2(t)$

  $\Rightarrow \; \tilde{\mathbf{Q}}_d(z) = [\; \tilde{\mathbf{Q}}_2(z),\; \tilde{\mathbf{Q}}_4(z) \;] \hookrightarrow$

- **input** $u_2(t)$ **&** $u_4(t)$ **de–coupling**

  $\Rightarrow \; \tilde{\mathbf{Q}}_c(z) = [\; \tilde{\mathbf{Q}}_1(z), \tilde{\mathbf{Q}}_3(z), \tilde{\mathbf{Q}}_5(z) \;]$

  $\Rightarrow \; \hookrightarrow$ sensitive to $u_1(t)$, $u_3(t)$ & $u_5(t)$

Residual Function & 2 Disturbance De–coupling

⇒ $n_{f_1} = 9$

⇒ Computation of the coefficients of the polynomials of the matrices $\mathbf{S}_c(z)$ & $\mathbf{S}_y(z)$

⇒ Null space of $\tilde{\mathbf{Q}}_d(z) \hookrightarrow$

$$\mathbf{L}(z) = \left[\begin{array}{c} -0.029 + 0.17z - 0.34z^2 + 0.2z^3 + 0.0016z^4 - 0.005z^5 \\ -0.024 + 0.036z - 0.092z^2 + 0.18z^3 + 0.43z^4 - 0.62z^5 + 0.042z^6 \\ -0.018 + 0.068z - 0.11z^2 + 0.11z^3 + 0.17z^4 - 0.37z^5 + 0.13z^6 + 0.016z^7 \end{array} \right]^T$$

Residual Function & 2 Disturbance De–coupling: $\mathbf{S}_c(z)$

$$\mathbf{S}_c(z) = \left[\ \mathbf{S}_{c_1}(z), \ \mathbf{S}_{c_2}(z), \ \mathbf{S}_{c_3}(z) \ \right]$$

$$\mathbf{S}_{c_1}(z) = \big[0.0002z^9 - 0.0077z^8 + 0.0315z^7 - 0.0257z^6 - 0.0056z^5 + \\ +0.0067z^4 + 0.0015z^3 - 0.0022z^2 + 0.0027z - 0.0011\big],$$

$$\mathbf{S}_{c_3}(z) = \big[0.0003z^9 - 0.0044z^8 - 0.0092z^7 + 0.0118z^6 + 0.0074z^5 + \\ -0.0063z^4 + 0.0016z^3 - 0.0034z^2 + 0.0015z - 0.0001\big],$$

Residual Function & 2 Disturbance De–coupling: $S_c(z)$

$$\mathbf{S}_c(z) = \left[\ \mathbf{S}_{c_1}(z),\ \mathbf{S}_{c_2}(z),\ \mathbf{S}_{c_3}(z)\ \right]$$

$$\mathbf{S}_{c_5}(z) = \big[-0.0004z^9 - 0.0016z^8 + 0.0028z^7 + 0.0015z^6 - 0.0026z^5 + \\ +0.0001z^4 + 0.0036z^3 - 0.0048z^2 + 0.0016z - 3\times 10^{-5}\big],$$

Residual Function & 2 Disturbance De–coupling: $S_y(z)$

$$\mathbf{S}_y(z) = \left[\ \mathbf{S}_{y_1}(z),\ \mathbf{S}_{y_2}(z),\ \mathbf{S}_{y_3}(z)\ \right]$$

$$\mathbf{S}_{y_1}(z) = \big[-0.0031z^9 + 0.0360z^8 - 0.0172z^7 - 0.0988z^6 + \\ +0.1273z^5 - 0.0550z^4 + 0.0262z^3 - 0.0307z^2 + 0.0194z - 0.0044\big],$$

$$\mathbf{S}_{y_2}(z) = \big[0.0167z^9 - 0.3144z^8 + 0.7071z^7 - 0.5489z^6 + \\ +0.0923z^5 + 0.1046z^4 - 0.1078z^3 + 0.0702z^2 - 0.0241z + 0.0027\big],$$

Residual Function & 2 Disturbance De–coupling: $\mathrm{S}_y(z)$

$$\mathbf{S}_y(z) = \left[\ \mathbf{S}_{y_1}(z), \ \mathbf{S}_{y_2}(z), \ \mathbf{S}_{y3}(z) \ \right]$$

$$\mathbf{S}_{y3}(z) = \big[0.0124z^9 - 0.0380z^8 + 0.0272z^7 + 0.0422z^6 + \\ -0.0740z^5 - 0.0074z^4 + 0.0979z^3 - 0.0886z^2 + 0.0329z - 0.0044\big],$$

Residual Function & 2 Disturbance De–coupling: $\mathrm{L}(z)$

- $n_f = 9$ **&** $\mathbf{L}(z)$ **given by:**

$$\mathbf{L}(z) = \begin{bmatrix} -0.029 + 0.17z - 0.34z^2 + 0.2z^3 + 0.0016z^4 - 0.005z^5 \\ -0.024 + 0.036z - 0.092z^2 + 0.18z^3 + 0.43z^4 - 0.62z^5 + 0.042z^6 \\ -0.018 + 0.068z - 0.11z^2 + 0.11z^3 + 0.17z^4 - 0.37z^5 + 0.13z^6 + 0.016z^7 \end{bmatrix}^T,$$

Residual Function & 2 Disturbance De–coupling: $\mathrm{L}(z)$

- **Computed by the Matlab null space [Forney, 1975; Chen, 1984] of the matrix $\tilde{\mathrm{Q}}_d(z)$**
  - ⇒ The Matlab function `null` computes a polynomial basis for the right null space of a polynomial matrix
- **This basis is *minimal* in the sense of Forney [Forney, 1975; Kailath, 1980], *i.e.*:**
  - ⇒ *(i)* it is column reduced & has full column rank (irreducible)
  - ⇒ *(ii)* it has minimal order

2. Residual Function $r(t)$ with Measurement Noise

- **$P(z)$ & $Q(z)$ is the linear model of Pont–sur–Sambre**
  - ⇒ $\{u(t), y(t)\}$ simulated from the linear model + additive noise
  - ⇒ $$\begin{cases} \mathbf{c}^*(t) = \mathbf{c}(t) + \tilde{\sigma}_c(t) \\ \mathbf{y}^*(t) = \mathbf{y}(t) + \tilde{\sigma}_y(t) \end{cases}$$
  - ⇒ Residual generator $R(z)\, r(t) = S_y(z)\, y^*(t) + S_c(z)\, c^*(t)$
  - ⇒ from the linear model

2. Residual Function $r(t)$ with Measurement Noise

$$R(s)\,r(t) = \underbrace{L(s)\,P(s)\,y^*(t) - L(s)\,Q_c(s)\,c^*(t)}_{\text{measurement noise} \neq 0} + \underbrace{L(s)\,Q_f(s)\,f(t)}_{\text{faulty case}}$$

⇒ $\{\tilde{\sigma}_c(t), \tilde{\sigma}_y(t)\} \leqq 10\%$

⇒ $R(z) \hookrightarrow$ fault sensitivity maximisation

⇒ 1 or 2 de–coupled disturbances + fault isolation $\hookrightarrow n_f = 4, 5, \&9$

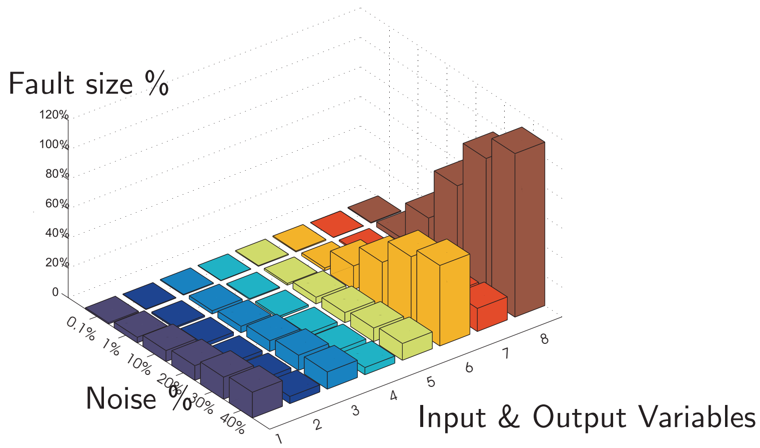
2. Fault Detection Capabilities with Noise

Noise	$f_{c_1}\%$	$f_{c_2}\%$	$f_{c_3}\%$	$f_{c_4}\%$	$f_{c_5}\%$	$f_{y_1}\%$	$f_{y_2}\%$	$f_{y_3}\%$
0.1%	0.8226	0.32	0.48	0.19	0.47	0.61	0.01	0.50
1 %	3.72	0.90	2.16	0.60	1.12	1.83	0.91	3.31
10 %	7.44	2.32	4.36	1.20	4.53	15.86	5.47	28.75
20 %	9.92	2.54	7.27	2.40	6.80	31.12	9.13	63.24
30 %	14.88	3.51	10.18	3.60	9.07	47.60	12.78	94.55
40 %	17.36	4.68	11.63	4.80	11.29	54.31	15.52	110.52

⇒ Minimal detectable fault

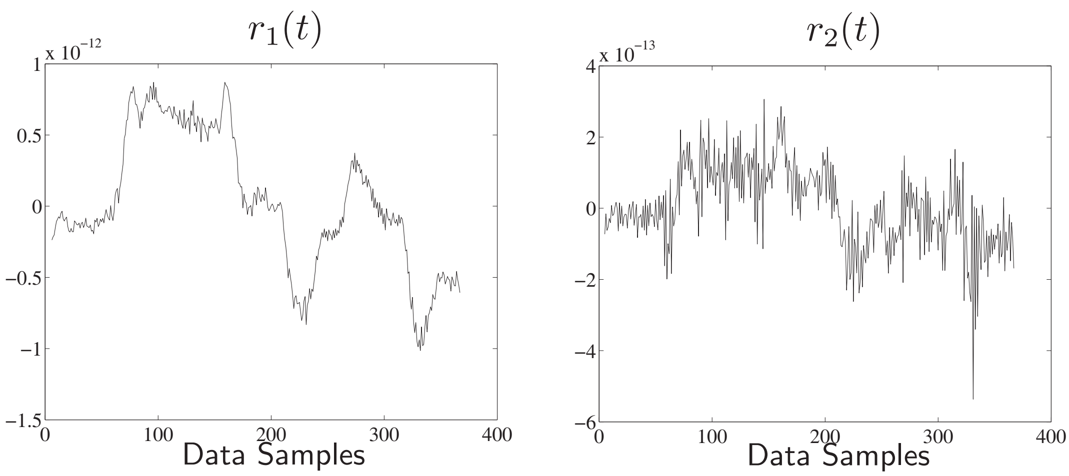
⇒ Increasing additive noise amplitude

2. Fault Sensitivity with Measurement Noise



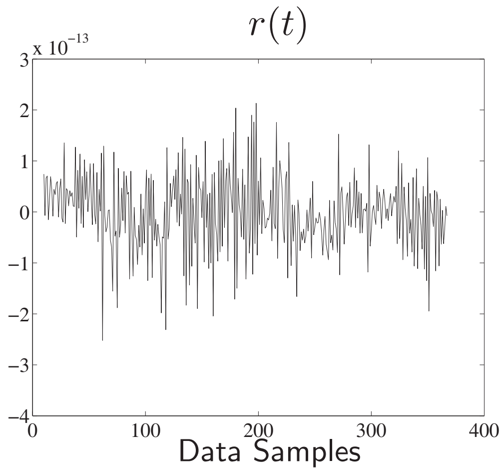
Variable Measurement Noise

2. Fault–Free Residual Signals (1 disturbance)



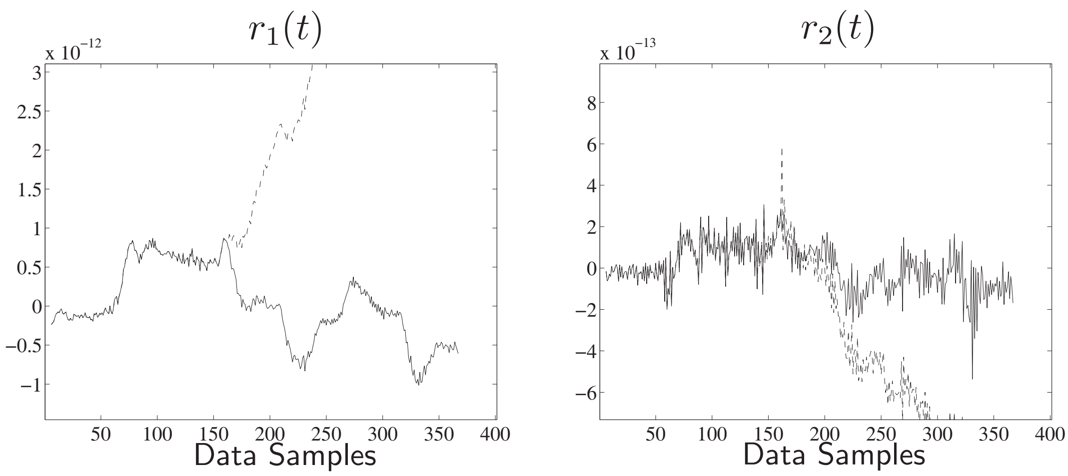
First & second monitored fault–free residuals for the case of one disturbance signal ($\ell_d = 1$).

2. Fault–Free Residual Signals (2 disturbance signals)



Monitored fault–free residual for the case of 2 disturbance signals ($\ell_d = 2$).

2. Fault–Free & Faulty Residual Signals

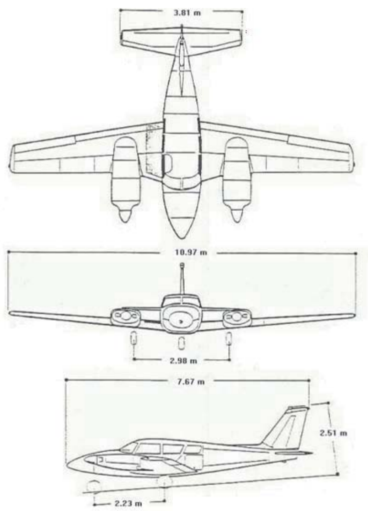


Faults on the first input & first output when 1 disturbance ($l_d = 1$) has been de–coupled.

3. Residual Function $r(t)$ with Model Uncertainty

- **$\{P(s)\,,\,Q(s)\}$ model linearisation of a Piper PA–30 aircraft non–linear system**
  - $\Rightarrow$ $\{u(t), y(t)\}$ & faults simulated from the non–linear model simulator
  - $\Rightarrow$ Residual generator $R(s)\,r(t) = S_y(s)\,y(t) + S_c(s)\,c(t)$ from the *linearised model*
  - $\Rightarrow$ $R(s)$ poles by trial & error procedure $\hookrightarrow$ fault sensitivity maximisation
  - $\Rightarrow$ $3 + 1$ de–coupled disturbances $\hookrightarrow$ wind gust $+$ fault isolation; $n_f = 3, \ldots, 5$

Application Example 2



Piper PA–30 Model *Nomenclature*

V	True Air Speed (TAS)	δ_e	elevator deflection angle
α	angle of attack	δ_a	aileron deflection angle
β	angle of sideslip	δ_r	rudder deflection angle
P	roll rate	δ_{th}	throttle aperture percentage
Q	pitch rate	X,Y	horizontal coordinates
R	yaw rate		(inertial reference system)
ϕ	bank angle	H	altitude
θ	elevation angle		(inertial reference system)
ψ	heading angle	γ	flight path angle
n	engine r.p.m.	m	airplane mass
$\begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}$		airplane inertia moments matrix	
F_x, F_y, F_z		total force components along body axes	
M_x, M_y, M_z		total moment components along body axes	

Piper PA–30 Model *(1)*

$$
\begin{aligned}
\dot{V} &= F_x \frac{\cos\alpha\cos\beta}{m} + F_y \frac{\sin\beta}{m} + F_z \frac{\sin\alpha\cos\beta}{m} \\
\dot{\alpha} &= \frac{-F_x \sin\alpha + F_z \cos\alpha}{mV\cos\beta} + Q - (P\cos\alpha + R\sin\alpha)\tan\beta \\
\dot{\beta} &= \frac{-F_x\cos\alpha\sin\beta + F_y\cos\beta - F_z\sin\alpha\sin\beta}{mV} + P\sin\alpha + \\
& \quad -R\cos\alpha \\
\dot{P} &= \frac{M_x I_z + M_z I_{xz} + PQ\, I_{xz}\,(I_x - I_y + I_z)}{I_x I_z - I_{xz}^2} + \frac{QR\left(I_y I_z - I_{xz}^2 - I_z^2\right)}{I_x I_z - I_{xz}^2}
\end{aligned}
$$

Piper PA–30 Model *(2)*

$$\begin{aligned}
\dot{Q} &= \frac{M_y + PR\,(I_z - I_x) - P^2 I_{xz} + R^2 I_{xz}}{I_y} \\
\dot{R} &= \frac{M_x I_{xz} + M_z I_x + PQ\left(I_x^2 - I_x I_y + I_{xz}^2\right)}{I_x I_z - I_{xz}^2} + \\
&\quad + \frac{QR\,I_{xz}\left(-I_x + I_y - I_z\right)}{I_x I_z - I_{xz}^2} \\
\dot{\phi} &= P + Q\sin\phi\tan\theta + R\cos\phi\tan\theta \\
\dot{\theta} &= Q\cos\phi - R\sin\phi
\end{aligned}$$

Input–Output *Canonical Form* Computation

- $\tilde{P}(s)$ **&** $\tilde{Q}(s)$ **from model linearisation:**

$$\begin{aligned}
\dot{x}(t) &= \mathrm{A}x(t) + \mathrm{B}c(t) + \mathrm{E}d(t) \quad \text{with} \\
x(t) &= \left[\Delta V\ \Delta\alpha\ \Delta\beta\ \Delta P\ \Delta Q\ \Delta R\ \Delta\phi\ \Delta\theta\ \Delta\psi\ \Delta H\ \Delta n\right]^T \\
c(t) &= \left[\Delta\delta_e\ \Delta\delta_a\ \Delta\delta_r\ \Delta\delta_{th}\right]^T \\
d(t) &= \left[w_u\ w_v\ w_w\right]^T \\
y(t) &= \left[\Delta V\ \Delta P\ \Delta Q\ \Delta R\ \Delta\phi\ \Delta\theta\ \Delta\psi\ \Delta H\ \Delta n\right]^T
\end{aligned}$$

Residual Function $r(t)$ with Model Uncertainty

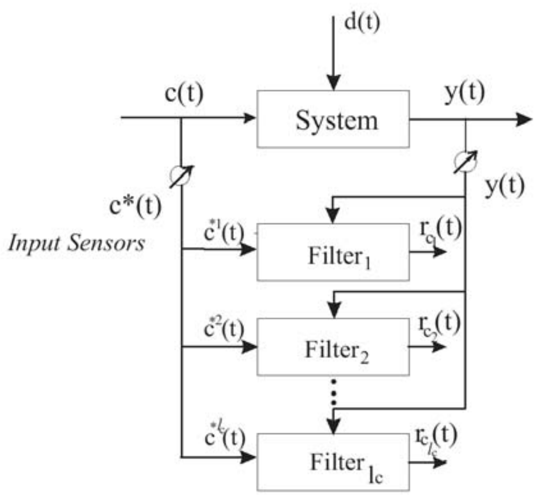
- $u(t) \in \Re^4$, $y(t) \in \Re^9$ **obtained from a non–linear Simulink model of Piper PA30**
- **With** $\tilde{P}(s)$ **&** $\tilde{Q}(s)$ **derived from model linearisation**

  $\Rightarrow$ $R(s)\,r(t) = L(s)\,\tilde{P}(s)\,y(t) - L(s)\,\tilde{Q}_c(s)\,c(t) + L(s)\,\tilde{Q}_f(s)\,f(t)$

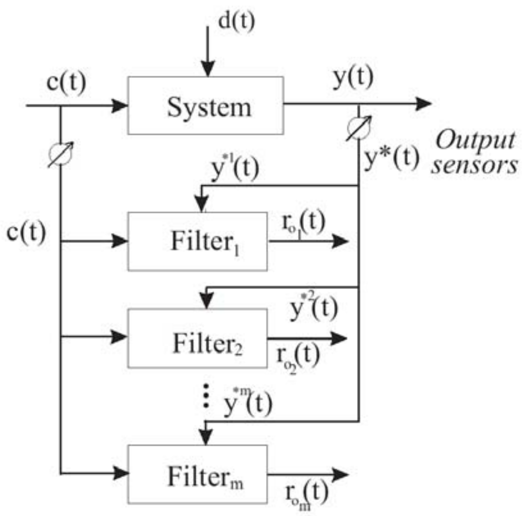
  $\Rightarrow$ $L(s)\,\tilde{P}(s)\,y(t) - L(s)\,\tilde{Q}_c(s)\,c(t)$ is model mismatch error
- $R(s)$ **&** $L(s)$**: polynomial matrices optimisation** $\hookrightarrow$ ***GAOT*** **for Matlab**

  $\Rightarrow$ $\ell_d = 3$ (wind gusts) $+ 1$ (input or output)

Scheme for Input Sensor FDI



Scheme for Output Sensor FDI



Università di Ferrara, Dipartimento di Ingegneria
Via Saragat, 1. I-44100, Ferrara, Italy

Model Mismatch & Measurement Errors

- **Faulty $\{u(t), y(t)\}$ simulated by the non–linear model**

$\Rightarrow$ Wind gusts: $(w_u,\ w_v,\ w_w)$. Correlation times & variances: $\tau_u = 2.326[s]$, $\tau_v = 7.143[s]$, $\tau_w = 0.943[s]$, $E[w_u^2] = E[w_v^2] = E[w_w^2] = 0.7[(m/s)^2]$

$\Rightarrow$ Trajectory:

- radius of curvature $1000[m]$
- speed $V = 50[\frac{m}{s}]$
- altitude $H = 330[m]$
- flap $= 0^o$.

$\Rightarrow$ Detailed model of **Inertial Measurement Unit** (MEMS technology), **Air Data Computer**, **Heading Reference System**

Università di Ferrara, Dipartimento di Ingegneria
Via Saragat, 1. I-44100, Ferrara, Italy

Simulator Block Brief Description (1)

Command Surfaces Deflection Measurements

⇒ δ_e, δ_a, δ_r, δ_{th} acquired with a sample rate of 100Hz by means of potentiometers

⇒ Errors modelled by bias & white noise:

Input sensor	Bias	White Noise Std
Elevator deflection angle	0.0052 rad	0.0053 rad
Aileron deflection angle	0.0052 rad	0.0053 rad
Rudder deflection angle	0.0052 rad	0.0053 rad
Throttle aperture	1%	1%

Simulator Block Brief Description (2)

Angular Rate Measurement

⇒ Angular rate measures by 3 gyroscopes of (IMU) with sample rate of 100Hz

⇒ Measurement errors:

- Non unitary scale factor: multiplicative factor $\in [0.99, 1.01]$.
- Alignment error of spin axes with respect to body (reference) axes: six error angles up to 1 deg (uniform random variables)
- Limited bandwidth of the considered gyro (10 Hz).
- g–sensitivity ($72\ \frac{deg}{h\ g}$).
- Additive white noise (216 deg/h).
- Gyro drift, coloured stochastic process, 1080 deg/h std. dev. & a decay time of 20 min.

Simulator Block Brief Description (3)

Attitude Angle Measurement

⇒ angles generated by a digital filtering system based on a DSP that processes both the angular rate & the accelerations provided by the IMU with a sample rate of 100Hz

⇒ 2 Measurement errors correlated by a first order filter system with time constant equal to 60 sec

- A systematic error generated by the apparent vertical.
- A white noise modelling the imperfection of both the system & the environment influences.

⇒ the resulting attitude angle measurements are affected by an additive coloured noise with std.dev. of 1 deg

Simulator Block Brief Description (4a)

Air Data System (ADS)

⇒ the ADS unit consists of an Air Data Computer (ADC) providing measures with a sample rate of 1 Hz

⇒ Errors affecting the TAS:

- Calibration error affecting the differential pressure sensor. This error leads to a TAS computation systematic error, performed the ADC, fulfilling the ARINC (Aeronautical Radio Inc.) accuracy requirements (2 m/sec).
- Additive coloured noise due to wind gusts (std. dev. 1 & correlation time 2.3 sec).
- Additive white noise (std. dev. 0.5 m/sec) modelling the imperfection of the system & the environment influences.

Simulator Block Brief Description (4b)

Air Data System (ADS)

⇒ Altitude errors:

- Calibration error affecting the static pressure sensor. This error leads to an altitude computation systematic error, performed the ADC, fulfilling the ARINC accuracy requirements (5 m).
- Additive White noise (std. dev. 1 m) modelling the imperfection of the system & the environment influences.

Simulator Block Brief Description (5)

Heading Reference System (HRS)

⇒ Magnetic compass coupled to a directional gyro

⇒ Measurement errors correlated by a first order filter with time constant equal to 60 sec :

- a systematic error generated by a bias of the magnetic compass (1 deg),
- a white noise modelling the imperfection of the system & the environment influences.

⇒ The resulting heading measurement is affected by an additive coloured noise std.dev. 1 deg

Simulator Block Brief Description (6)

- **Engine Shaft Rate Measurement**
  - ⇒ Incremental encoder white noise. Quantisation error resolution of 10000 pulse/rev
- **Servo Actuator Models**
  - ⇒ second order linear models with saturations
- **Dryden Atmosphere Model**
  - ⇒ Model block of the Aerospace Blockset of Matlab® 6.5
  - ⇒ Turbulence obtained by forming filter excited by a band–limited white noise (U.S. Military Specification MIL–F–8785C)

FDI Filter Design

- Filters fed by 4 component $c(t)$ & by 9 component $y(t)$
- The input & output sequences are affected by the measurement errors
- 4 residual generator filter bank used to detect input sensor for the 4 control variables $c(t) = [\Delta\delta_e(t), \Delta\delta_a(t), \Delta\delta_r(t), \Delta\delta_{th}(t)]^T$
- Fault *isolation* properties if each residual generator is fed by all but one the 4 inputs & by the 9 outputs $y(t) = [\Delta V(t), \Delta P(t), \Delta Q(t), \Delta R(t), \Delta\phi(t), \Delta\theta(t), \Delta\psi(t), \Delta H(t), \Delta n(t)]^T$
- The output variables $\Delta\alpha(t)$ & $\Delta\beta(t)$ not considered as critical to obtain

FDI Filter Design: De–coupling Issues

- Each filter bank is independent of one of the 4 inputs
- insensitive to the corresponding fault signals
- Residual generator bank is be de–coupled from 3 wind gusts $d(t) = [w_u(t), w_v(t), w_w(t)]^T$
- FDI capability related to the properties of the residual generators with measurement errors, modelling approximations & un–decoupled disturbance signals
- Filter robustness properties in terms of fault sensitivity & disturbance insensitivity

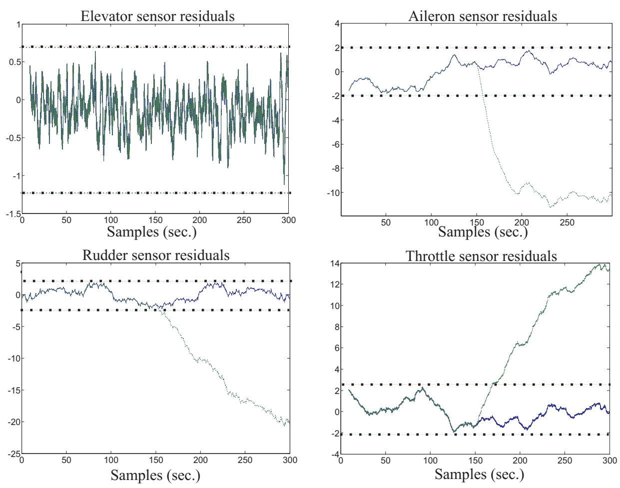
FDI Filter Design: Optimisation

- Filter synthesis performed by choosing residual generator linear combination of residual generators maximising the steady–state gain of the transfer functions between input sensor fault signals $f_{c_i}(t)$ & residual functions $r_{c_j}(t)$ $(i, j = 1, \ldots, 4,\ j \neq i)$
- Polynomial roots $R_{c_j}(s)$ optimised numerically [GAOT, 1995] for obtaining suitable transient dynamics.
- Aircraft operating conditions with different faults were simulated
- Faults in single input–output sensors are generated by in the input–output signals $c(t)$ & $y(t)$.

FDI Filter Remarks: Fault Detection

- The residual signals indicate fault occurrence if their values are lower or higher than the fault–free thresholds
- The thresholds depend on the residual errors due to measurement errors, linearised model approximations & un–decoupled disturbance signals
- Positive & negative threshold 10% margins on the maximum & minimum values of the fault–free residual signals

Examples of Faulty & Fault–Free Residuals



Bank residuals for the 1^{st} input sensor fault $f_{c_1}(t)$ isolation

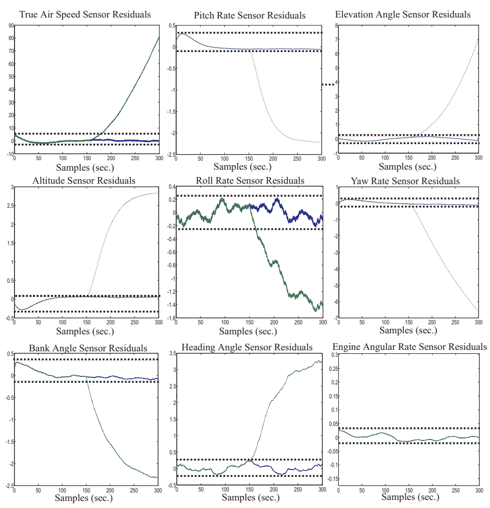
FDI Filter Design Remarks

- The first residual for the input sensor $f_{c_1}(t)$ independent of a fault on the input sensor itself: $r_{c_1}(t)$ filter designed to be sensitive to the input signal $c^{*1}(t)$

- Filter parameter optimisation, *i.e.* the the roots $-1/\tau_i (i = 1, 2, \ldots, n_f)$ & of real constants k_i, obtained by means of the *Genetic Algorithm Optimisation Toolbox* (GAOT) [GAOT, 1995] for Matlab® (local minima problems)

- $R_{c_i}(s)$ polynomial matrix roots optimised & placed between -1 & -10^{-2} for maximising fault detection promptness & for minimising false alarm rate

Examples of Faulty & Fault–Free Residuals



Residuals of the bank for the isolation of the 9^{th} output sensor fault $f_{o_9}(t)$.

FDI Filter Design Remarks

- Minimal fault detection capabilities obtained by optimising maximum & minimum values of $r_{c_i}(t)$ & $r_{o_j}(t)$ in fault–free conditions (acceptable false–alarms rate)
- Minimal detectable step faults on the various sensors are collected in the Tables
- First Table collects the minimal detectable *step* fault simulated on the *input* sensors: $r_{c_j}(t)$ monitored for FDI of the considered input fault case $f_{c_i}(t)$ $(i, j = 1, \ldots, 4, i \neq j)$
- Second Table collects minimal detectable *step* fault amplitudes on *output* sensors: $r_{o_j}(t)$ monitored for FDI of output sensor fault case $f_{o_i}(t)$ $(i, j = 1, \ldots, 9, i \neq j)$

Minimal Detectable *Input* Faults

Minimal detectable step input sensor faults.

Input Sensor Variable $c_i(t)$	Fault Size	Detection Delay
Elevator deflection angle	2^o	18 sec
Aileron deflection angle	3^o	6 sec
Rudder deflection angle	4^o	8 sec
Throttle aperture %	2%	15 sec

Minimal Detectable *Output* Faults

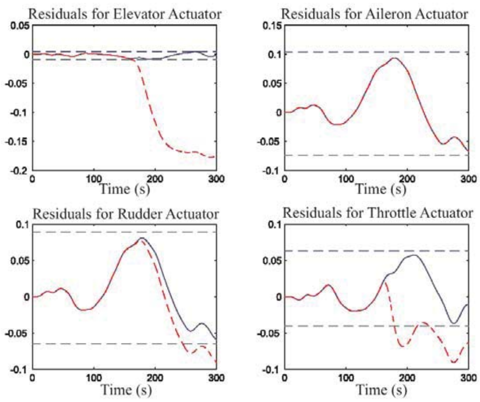
Minimal detectable step output sensor faults.

Output Sensor Variable $y_i(t)$	Fault Size	Detection Delay
True Air Speed	8 m/sec	27 sec
Pitch Rate	3 deg/sec	22 sec
Elevation Angle	5 deg	28 sec
Altitude	8 m	12 sec
Roll Rate	2 deg/sec	24 sec
Yaw Rate	3 deg/sec	29 sec
Bank Angle	5 deg	5 sec
Heading Angle	6 deg	25 sec
Engine Angular Rate	20 RPM	30 sec

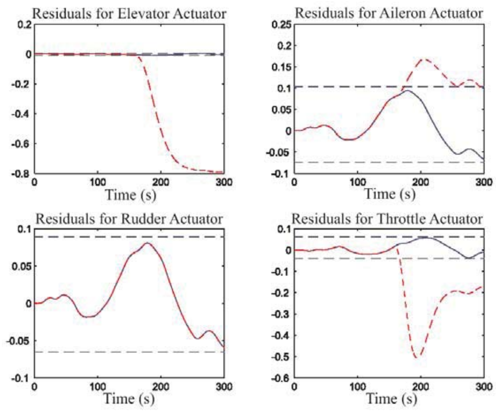
Minimal Detectable Fault Remarks

- The minimal detectable fault values in Tables expressed in the unit of measure of sensor signals & s.t. any fault occurrence must be detected & isolated asap
- Detection delay time evaluated & based on the slowest residual crossing time w.r.t. the settled thresholds
- Residual generator performance seems to assess the diagnostic capabilities of the technique
- FDI strategy appears promising for diagnostic application to general aviation aircrafts
- Similar results obtained by dynamic observers, UIO or Kalman filters: but the corresponding realisations require a more complex design & an higher cost implementation

Residual Function Examples: faulty & fault–free residuals



Aileron actuator fault δ_a



Rudder actuator fault δ_r.

Final Remarks

- Some results are shown in FDI of sensor faults of dynamic system by using a model–based approach
- Different types of fault having a barely detectable effect on anyone measurement, can be detected easily using a bank of residual generator in the form of dynamic filters
- An important aspect of the approach suggested here to FDI is the simplicity of structure of the technique used to generate the residual functions for FDI, when compared with traditional schemes *e.g.* based on banks of Unknown Input Observers (UIO) & Kalman filters.

Final Remarks

- The method outlined focuses to some extent on input–output or state–space concepts; the actual algorithm is based only on input–output processing of all measurable signals ($c(t)$ & $y(t)$)
- Algorithmic simplicity important when verification & validation of a demonstrable scheme for air–worthiness certification needed
- Complex computations & schemes require high cost & complexity w.r.t. certification
- Modelling uncertainty & measurement noise well tackled: this method used in real applications
- Further studies for evaluating effectiveness on real aircraft system data

References: *General FDI*

- J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.
- R. J. Patton, P. M. Frank, and R. N. Clark, eds., *Fault Diagnosis in Dynamic Systems, Theory and Application*. Control Engineering Series, London: Prentice Hall, 1989.
- R.J.Patton,P.M.Frank,and R.N.Clark, eds., *Issues of Fault Diagnosis for Dynamic Systems*. London Limited:Springer–Verlag, 2000.
- J. Chen and R. J. Patton, *Robust Model–Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- R.Isermann and P.Ballé, "Trends in the application of model–based fault detection and diagnosis of technical processes",*Control Engineering Practice*, vol.5, no.5, pp.709–719, 1997.

References: *Same Author*

- S. Simani, C. Fantuzzi, and R. J. Patton, *Model–based fault diagnosis in dynamic systems using identification techniques*. Advances in Industrial Control, London, UK: Springer–Verlag,first ed., November 2002. ISBN 1852336854.
- 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*, Vol. 20, Issue: 8, October 2006, pp. 381–408.
- S. Simani, C. Fantuzzi, and S. Beghelli, Diagnosis techniques for sensor faults of industrial processes, *IEEE Transactions on Control Systems Technology*, vol.8, pp.848–855, September 2000.
- S. Simani, R. Diversi, U. Soverini, Identification of residual generators for fault detection of linear dynamic models. *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05*. Pp. 7651–7655, 2005.

References: *Related Works*

- E. Frisk, "Order of residual generators: bound and algorithms", in *SAFEPROCESS'2000: Proc. of IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, vol. 1, (Budapest, Hungary), pp. 599–604, 2000.
- E. Frisk and M. Nyberg, "A minimal polynomial basis solution to residual generation for fault diagnosis in linear systems", *Automatica*, vol. 37, no. 9, pp. 1417–1424, 2001.

References: *Canonical Forms*

- S. Beghelli and R. Guidorzi, “A new input–output canonical form for multivariable systems”, *IEEE Trans. Automat. Contr.*, vol. AC–21, pp. 692–696, 1976.
- R. P. Guidorzi, “Canonical Structures in the Identification”, *Automatica*, vol. 11, pp. 361–374, 1975.
- R. P. Guidorzi and R. Rossi, “Identification of a power plant from normal operating records”, *Automatic Control Theory and Applications*, vol. 2, pp. 63–67, September 1974.

References: *Algorithms*

- T. Kailath, *Linear systems*. Englewood Cliffs, New Jersey. Prentice Hall, 1980.
- G. D. Forney Jr., “Minimal bases of rational vector spaces with applications to multivariable linear systems”, *SIAM Journal on Control*, vol. 13, no. 3, pp. 493–520, 1975.
- C. T. Chen, *Linear Systems Theory and Design*. New York: Holt, Rinehart and Winston, 1984.
- C. Houck, J. Joines, and M. Kay, “A Genetic Algorithm for Function Optimization: A Matlab Implementation”, *Tech. Rep.* NCSU–IE TR–95–09, North Carolina State University, Raleigh, NC, USA, 1995. (Available at `ftp://ftp.eos.ncsu.edu/pub/simul/GAOT/`).