

Fault Detection and Isolation

Application to an Unmanned Air Vehicle (drone)

Christophe Farges

MASTER 2 MAINTENANCE AÉRONAUTIQUE
spécialité Ingénierie et Maintenance Aéronautique Avionique

4TNV902U

3E ANNÉE ENSPIMA
spécialité Systèmes Aéronautiques

AP9SYCDA

Introduction

- Automatic control at Evering up to now
 - Modeling
 - From physical equations: helicopter model, state space model of a plane...
 - Using identification
 - Control
 - Proportional Integral Derivative control
 - Linear Quadratic control
 - Numerical control
 - Robust multi-input multi-output control (M2)
- ⇒ The topic of this course is **diagnosis**
- Overall concept of diagnosis involves two essential tasks
 - ① **detect** an abnormal behavior affecting a system (**fault**)
 - ② **isolate** (localize) the faulty component (sensor, actuator...)
- ⇒ *introductory examples of diagnosis in aerospace applications*

Introduction

- Example 1: failures affecting a flight control system



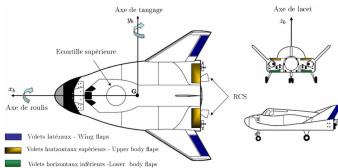
- Symptom: undesired oscillations of flight control surfaces
 - Consequences: flight performance and handling qualities degradation, actuator wear...
 - Cause: malfunction of electrical components of the control servo-loop
- ⇒ How to detect this fault in flight in order to switch to another actuator?
(each flight control surface has two redundant actuators)

Introduction

● Example 2: failures affecting the HL-20 re-entry vehicle



- concept designed by NASA for earth-to-orbit manned orbital missions: transfer a crew to the ISS, satellites servicing...
- objectives: supplement to the USS shuttle orbiter with low operational costs, improved flight safety, possibility of landing on conventional runways

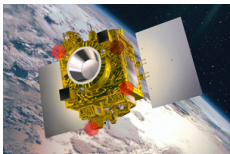


- 7 control surfaces
- 2 Inertial Measurement Units (accelerometers and gyroscopes), 1 air data computer (attitude, dynamic pressure, speed), 1 GPS

- ⇒ actuator faults on wing flaps (*lock-in-place due to control servo-loop, swerve due to hydraulic drive system malfunction*)
- ⇒ sensor fault on IMU (*damaging of sensor during the hypersonic phase, bias, drift*)

Introduction

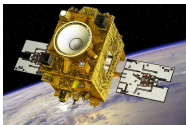
- Example 3: MICROSCOPE (MICRO-Satellite à traînée Compensée pour l'Observation du Principe d'Equivalence) by CNES



- 300 kg minisatellite launched on April 25 2016
- objective: test the equivalence principle with a better precision than on Earth (100 times)
- instrument: constituted of two identical differential accelerometers, each accelerometer contains 2 concentric cylindrical proof masses controlled in the center of a cage by electrostatic levitation
 - the reference accelerometer has test masses of same material (Pt)
 - the test accelerometer has test masses of different materials (Pt/Ti)
- \neq control applied to the two masses \Rightarrow violation of equivalence principle
- experiment very sensitive to trajectory deviation \Rightarrow necessity to detect:
 - actuator faults: lock-in of some of the 12 nozzles diaphragms
 - sensor faults: bias on the position and speed data returned by IMU

Introduction

● Example 3: MICROSCOPE (MICRO-Satellite à traînée Compensée pour l'Observation du Principe d'Equivalence) by CNES



- The last results of the MICROSCOPE mission were published on September 14, 2022 and confirm the equivalence principle with an unprecedented precision of 10^{-15} . This result confirms once again the theory of General Relativity proposed by Albert Einstein. [IHES]
- Launched in 2016, a century after the publication of Einstein's theory, the MICROSCOPE mission aims to test the equivalence principle between inertia and gravitation, a fundamental pillar of General Relativity, postulating that all bodies fall in the same way in the vacuum. The violation of the equivalence principle is predicted by some unification theories between gravitation and quantum physics. In particular, a weak but non-zero violation of the equivalence principle by the dilaton in string theory has been predicted by recent work studying the mechanism of cosmological attraction – hence the importance of testing the equivalence principle with high precision. The mission was designed by the Office National d'Etudes et de Recherches Aéropatiales (ONERA), in collaboration with the Observatoire de la Côte d'Azur (OCA), the CNES (Centre National d'Etudes Spatiales) and the ZARM (Bremen, Germany) [...]
- MICROSCOPE uses very advanced technologies to compare the free fall acceleration of two bodies of different compositions, one made of platinum, the other of titanium. The whole thing takes place aboard a satellite that orbited the Earth from April 2016 until October 2018, making 1,642 revolutions and thus traveling 73 million km, equivalent to half the Earth-Sun distance. Preliminary results were published in 2017 – this was already a confirmation of the equivalence principle with unparalleled accuracy that allowed the mission's co-investigators and project leaders to win the 2019 Grand Prix Servant of the French Academy of Sciences.
- The analysis of the data collected during two and a half years by the ONERA and OCA scientific teams, with the contribution of CNES and the collaboration of European laboratories, was published on September 14, 2022 in two prestigious journals: Physical Review Letters, and Classical and Quantum Gravity. These latest results push the limits of the accuracy of the equivalence principle test even further. With its results, MICROSCOPE provides important constraints on new theories of gravitation at a level of precision that researchers expect to remain unmatched for a long time.

Introduction

- Objectives of the course
 - Overview of fault detection and isolation methods used in aerospace applications
 - Focus on model-based approaches and particularly on the *parity space approach*
 - Case study: in-flight fault detection and isolation for a quadcopter
- Organization of the sessions
 - 8 sessions with both courses and exercises
 - 1 session dedicated to a past exam
 - 4-5 practical work sessions for the case study, including control part (part of the avionics project for ENSPIMA students)
- Required background
 - basic knowledge on state space modeling and numerical control
 - working knowledge on MATLAB/SIMULINK



Outline of the course

- 1 Fault detection and isolation methods overview
 - Fault diagnosis tasks (definitions)
 - Signal processing approaches
 - Hardware redundancy
 - Analytical redundancy
- 2 Parity space approach
 - Reminder on sampled data systems
 - Static parity space approach
 - Dynamic parity space approach

Outline of the course

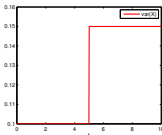
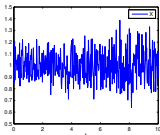
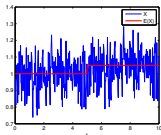
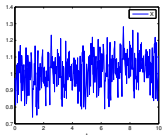
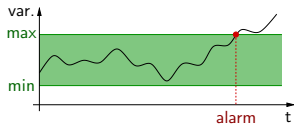
- 1 Fault detection and isolation methods overview
 - Fault diagnosis tasks (definitions)
 - Signal processing approaches
 - Hardware redundancy
 - Analytical redundancy
- 2 Parity space approach
 - Reminder on sampled data systems
 - Static parity space approach
 - Dynamic parity space approach

Fault diagnosis tasks (definitions)

- Fault **detection** task
 - Objective: spot the occurrence of events that may lead to abnormal behavior of the system
 - *faults* must be distinguished from *disturbances* that divert the system from the desired behavior but occur in normal functioning
- Fault **isolation** task (or *localization task*)
 - Objective: delimit the fault to a component or a set of components (actuators, sensors)
- Fault diagnosis techniques
 - Model-free techniques
 - signal processing approaches
 - hardware redundancy
 - **Model-based techniques**
 - analytical redundancy

Signal processing approaches

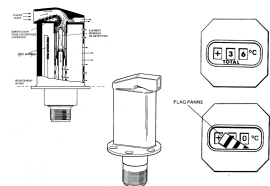
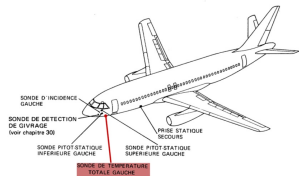
- Hypothesis
 - Some measurable system variables carry information about the faults
- Principle
 - Use signal processing to monitor if the variables behave normally
- Time domain analysis
 - Magnitudes (limit-value checking)
 - if variables leave a range corresponding to a normal functioning, an alarm is triggered
 - Mean, standard deviation



- Frequency domain analysis
 - Spectral power densities

Signal processing approaches

- Example: total air temperature probe of Dassault Mercure



- Temperature indication results from a resistance measurement whose value follows a known law depending on temperature
- Measuring range: $[-99^{\circ}\text{C}, +50^{\circ}\text{C}]$
- Monitored faults
 - loss of power supply, short circuit
 - error affecting measurement processing
 - for instance when temperature leaves the range
 - detected using an electrical circuit
- use of AMM to localize the fault

- Advantage: easy to implement

- Drawbacks

- not efficient for wide operating range systems
- extra cost for the implementation of additional measurement chains

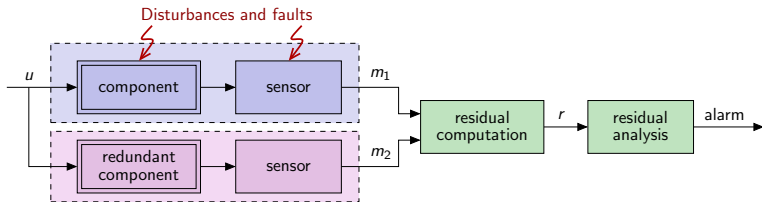
Hardware redundancy

- Principle

- introduce additional identical (redundant) hardware components
- fault detected if output of the original component differs from the redundant ones

- Dual modular redundancy

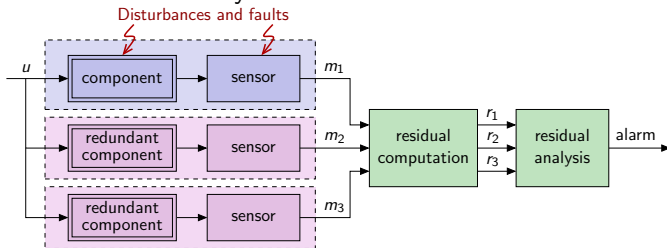
- Critical component is duplicated



- $r = m_1 - m_2$ is called **residual** signal
 - residual r is compared to a threshold depending on the measure quality
- ⇒ Faulty component not isolated

Hardware redundancy

- Triple modular redundancy



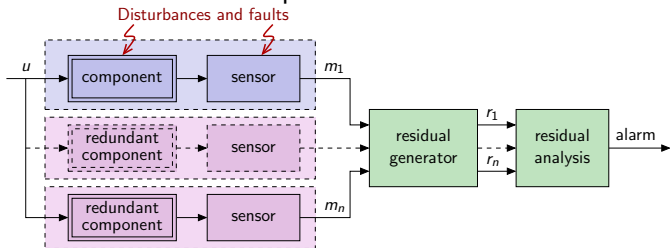
- Three residual signals: $r_1 = m_1 - m_2$, $r_2 = m_1 - m_3$, $r_3 = m_2 - m_3$
- Voter isolates which component is faulty

Component 1	Component 2	Component 3	r_1	r_2	r_3
✓	✓	✓	0	0	0
X	✓	✓	$\neq 0$	$\neq 0$	0
✓	X	✓	$\neq 0$	0	$\neq 0$
✓	✓	X	0	$\neq 0$	$\neq 0$
X	X	X	$\neq 0$	$\neq 0$	$\neq 0$
X	✓	X	$\neq 0$	$\neq 0$	$\neq 0$
X	X	✓	$\neq 0$	$\neq 0$	$\neq 0$
X	X	X	$\neq 0$	$\neq 0$	$\neq 0$

⇒ isolates a **single** fault only

Hardware redundancy

- General case: n identical components

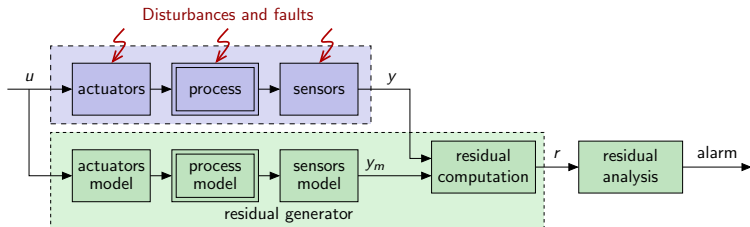


- faulty component isolated if a max. of $\lfloor \frac{n-1}{2} \rfloor$ faults occur simultaneously
- Example: detection and localization of sensor faults on Airbus A380
 - angle of attack, pitch/roll/yaw rates...
- Advantages: easy to design and implement
- Drawbacks
 - faults affecting all components simultaneously cannot be detected (*loss of power supply, electrical ground problem...*)
 - high costs (thus restricted to a number of key components)

Analytical redundancy

- General idea

- replace hardware redundancy by a plant model implemented in a computer
 - involves the measurement and control signals of the plant
 - requires a dynamical model of the plant (*actuators + process + sensors*)



- plant behavior is compared in real time to its model behavior
 - a difference can be interpreted as a symptom of a fault

- Advantages:

- no extra cost
- easy to implement (in the computer hosting the control law)
- allows to discriminate the disturbances and faults effects ⇒ better results than using signal processing based approaches

Analytical redundancy

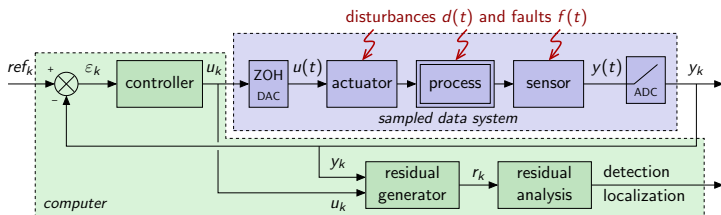
- Back to example 1: failures affecting a flight control system



- Symptom: undesired oscillations of flight control surfaces
 - Consequences: flight performance and handling qualities degradation, actuator wear...
 - Cause: malfunction of electrical components of the control servo-loop
- ⇒ Before A380: signal processing based (*without model*)
- ⇒ A380 program: analytical redundancy since oscillations belong to the control law bandwidth (*model-based*)

Analytical redundancy

- Principle
 - Compare plant behavior (affected by faults and disturbances) to its model
 - comparison result: **residual** (fault indicating signal)
- Implementation in the case of digital control



→ residual generator designed according to sampled data system model

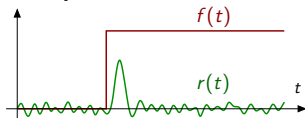
- Objective (ideal case)
 - fault-free case: $r_k = 0 \forall d_k$
 - in faulty case: $r_k \neq 0$
- Objective (realistic case)
 - r_k must be the least sensitive to d_k and the most sensitive to f_k
 - *residual analysis* generates alarm (threshold...) and isolates the fault

Analytical redundancy

- Types of detectability

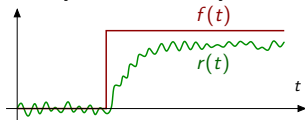
- Weak detectability

→ residual is affected by fault in transient state only



- Strong detectability

→ residual is affected by fault in steady state



⇒ Otherwise, the fault is said to be undetectable by the residual

- Diagnosis problem

- Given a sampled data system model, how to design the residual generator and how to perform the residual analysis?

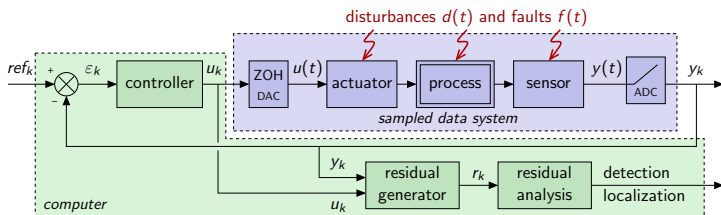
⇒ Methodology presented in this course: the parity space approach

Outline of the course

- 1 Fault detection and isolation methods overview
 - Fault diagnosis tasks (definitions)
 - Signal processing approaches
 - Hardware redundancy
 - Analytical redundancy
- 2 Parity space approach
 - Reminder on sampled data systems
 - Static parity space approach
 - Dynamic parity space approach

Reminder on sampled data systems

- Bloc diagram involving D/A converters for control and FDI purposes



- actuators + plant + sensors $\equiv \Sigma$ modeled by state space representation

$$\rightarrow \Sigma: \begin{cases} \dot{x}(t) &= \tilde{A}x(t) + \tilde{B}u(t) \\ y(t) &= \tilde{C}x(t) + \tilde{D}u(t) \end{cases}$$

- actuators + process + sensors + DAC + ADC $\equiv \Sigma_k$

$$\rightarrow \Sigma_k: \begin{cases} x(k+1) &= Ax(k) + Bu(k) + B_d d(k) + B_f f(k) \\ y(k) &= Cx(k) + Du(k) + D_d d(k) + D_f f(k) \end{cases}$$

$$\text{with } A = e^{\tilde{A}T_e}, \quad B = \int_0^{T_e} e^{\tilde{A}(T_e-\alpha)} \tilde{B} d\alpha, \quad C = \tilde{C}, \quad D = \tilde{D}$$

- \rightarrow *disturbances and fault are assumed to affect the sampled data model*

Parity space approach - principle

- Model

$$\Sigma_k \begin{cases} x(k+1) &= Ax(k) + Bu(k) + B_d d(k) + B_f f(k) \\ y(k) &= Cx(k) + Du(k) + D_d d(k) + D_f f(k) \end{cases}$$

- $x(k) \in \mathbb{R}^n$: state, $u(k) \in \mathbb{R}^m$: control, $y(k) \in \mathbb{R}^p$: measure
- $d(k) \in \mathbb{R}^{m_d}$: disturbance, $f(k) \in \mathbb{R}^{m_f}$: fault

- Objective

- Compute $r(k)$ using signals $u(k)$, $y(k)$ and model Σ_k
- Residual must be sensitive to $f(k)$, robust w.r.t. disturbances $d(k)$

- Principle of parity space approach

- use direct redundancy between measured signals
(when a measured variable can be derived from the other)
 - at time k , $r(k)$ is generated from $y(k)$ et $u(k)$ only
 - ⇒ **static parity space**
- use time redundancy linking measures and inputs at different time instants
 - at time k , $r(k)$ is generated from gathered values of measures and controls
 - ⇒ **dynamic parity space**

Static parity space - introductive examples

- Model

$$\sum_k \begin{cases} x(k+1) = Ax(k) + Bu(k) + B_d d(k) + B_f f(k) \\ y(k) = Cx(k) + Du(k) + D_d d(k) + D_f f(k) \end{cases}$$

- $x(k) \in \mathbb{R}^n$: state, $u(k) \in \mathbb{R}^m$: control, $y(k) \in \mathbb{R}^p$: measure
- $d(k) \in \mathbb{R}^{m_d}$: disturbance, $f(k) \in \mathbb{R}^{m_f}$: fault

- Objective: find $r(k) = f(y(k))$ sensitive to faults only

- Example 1: hardware redundancy

- $y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x(k) + \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix}$

→ $f_i(k)$: fault on sensor i

- $r(k) = \underbrace{y_1(k) - y_2(k)}_{\text{computation form}} = \cancel{x(k)} + f_1(k) - \cancel{x(k)} - f_2(k) = \underbrace{f_1(k) - f_2(k)}_{\text{evaluation form}}$

⇒ $r(k)$ can detect both $f_1(k)$ and $f_2(k)$

Static parity space - introductive examples

- Model

$$\Sigma_k : y(k) = Cx(k) + D_d d(k) + D_f f(k)$$

- $x(k) \in \mathbb{R}^n$: state, $u(k) \in \mathbb{R}^m$: control, $y(k) \in \mathbb{R}^p$: measure
- $d(k) \in \mathbb{R}^{m_d}$: disturbance, $f(k) \in \mathbb{R}^{m_f}$: fault

- Objective: find $r(k) = f(y(k))$ sensitive to faults only

- Example 2
$$y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix}$$

→ $f_1(k)$ affects sensor 3 and $f_2(k)$ affects sensors 2 et 3

- Computation form :
$$\begin{cases} r_1(k) = 2y_1(k) - y_3(k) \\ r_2(k) = y_1(k) + y_2(k) - y_5(k) \end{cases}$$

→ Exercise: determine the evaluation form and verify that $r_1(k)$ et $r_2(k)$ are independent from $x(k)$ but sensitive to $f_1(k)$ and $f_2(k)$

→ is it possible to isolate the fault using these residuals ?

⇒ method to derive $r_1(k)$ and $r_2(k)$ expressions ⇒ static parity space

Static parity space - detection

- Model

$$\Sigma_k : y(k) = Cx(k) + D_d d(k) + D_f f(k)$$

- $x(k) \in \mathbb{R}^n$: state, $u(k) \in \mathbb{R}^m$: control, $y(k) \in \mathbb{R}^p$: measure
- $d(k) \in \mathbb{R}^{m_d}$: disturbance, $f(k) \in \mathbb{R}^{m_f}$: fault

- Residual obtained as a linear combination of measures

- $r(k) = Wy(k)$ (computation form)

→ W parity matrix

→ determine W in previous exercise

⇒ how to choose W s.t. $r(k)$ is sensitive to faults only?

- Evaluation form of the residual

- $r(k) = Wy(k) = WCx(k) + WD_d d(k) + WD_f f(k)$

→ ideal case (disturbance-free case): $r(k) = WCx(k) + WD_f f(k)$

Static parity space - detection

- Computation of parity matrix W

- Evaluation form (disturbance-free case): $r(k) = WCx(k) + WD_f f(k)$
- Robustness constraint
 - $f(k) = 0 \Rightarrow r(k) = 0$ (for any $x(k)$)
- Sensitivity to faults constant
 - $f(k) \neq 0 \Rightarrow r(k) \neq 0$
- Solution?

Static parity space - detection

- Computation of parity matrix W

- Evaluation form (disturbance-free case): $r(k) = WCx(k) + WD_f f(k)$

- Robustness constraint

→ $f(k) = 0 \Rightarrow r(k) = 0$ (for any $x(k)$)

- Sensitivity to faults constant

→ $f(k) \neq 0 \Rightarrow r(k) \neq 0$

- Solution?

⇒ choose W s.t. $WC = 0 \Rightarrow r(k) = WD_f f(k)$

- W is orthogonal to C

- W exists if $p > \text{rank}(C)$ (measures redundancy), $W \in \mathbb{R}^{(p-\text{rank}(C)) \times p}$

Remark: if all the columns of C are independant $\Rightarrow \text{rank}(C) = n$

\Rightarrow condition becomes $p > n$ (more measures than state variables)

Remark

The term **parity** comes from the parity bits used in computers science. These bits introduce redundancy in order to detect a transmission error affecting digital data.

Static parity space - detection

- Detectability and parity space

- Evaluation form: $r(k) = WCx(k) + WD_f f(k)$ avec W tq $WC = 0$

- reminder: the fault is detectable if $\forall f(k) \neq 0 \Rightarrow r(k) \neq 0$

- ⇒ sensitivity to faults is not guaranteed

- All the faults are detectable if WD_f has no null column

- ⇒ a posteriori verification

- remark: if a fault is detectable, it is strongly detectable

Static parity space - detection

- A method to determine a parity space matrix

⇒ Objective: find W s.t. $WC = 0$ with $C \in \mathbb{R}^{p \times n}$, $p > n$, $\text{rank}(C) = n$

① partition $C = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}$, $C_1 \in \mathbb{R}^{n \times n}$, $C_2 \in \mathbb{R}^{(p-n) \times n}$

→ if C_1 is full rank, choose $W = [C_2 C_1^{-1} \mid -I_{p-n}]$ (hence $WC = 0$)

→ else go to step 2

② permute lines of C s.t. the n first lines constitute a full rank matrix and determine the corresponding transition matrix $T \in \mathbb{R}^{n \times n}$ s.t. $\tilde{C} = TC$ (matrix T is constituted of zeros and ones)

③ partition $\tilde{C} = \begin{bmatrix} \tilde{C}_1 \\ \tilde{C}_2 \end{bmatrix}$, $\tilde{C}_1 \in \mathbb{R}^{n \times n}$, $\tilde{C}_2 \in \mathbb{R}^{(p-n) \times n}$ (\tilde{C}_1 is invertible)

④ compute $\tilde{W} = [\tilde{C}_2 \tilde{C}_1^{-1} \mid -I_{p-n}]$ (hence $\tilde{W}\tilde{C} = 0$)

⑤ retrieve W according to $W = \tilde{W}T$ (hence $WC = \tilde{W}TC = \tilde{W}\tilde{C} = 0$)

⇒ W is not unique (depends for instance on the choice of the linearly independent lines)

→ this method guarantees the independence of the $p - n$ parity equations

Remark: if $\text{rank}(C) < n$, remove columns of C in order to retain only $\text{rank}(C)$ independent columns

Static parity space - detection

- Computation of a parity matrix with MATLAB
 - Objective: find W s.t. $WC = 0$ with $C \in \mathbb{R}^{p \times n}$, $p > n$, $\text{rank}(C) = n$
 - W is not unique
 - In order to get a unique solution, a constraint is added
 - lines of W must constitute an orthonormal basis
 - Problem writes
 - Find W s.t.
$$\begin{cases} WC & = 0 \\ WW^T & = I_{p-n} \end{cases}$$
 - Solution is obtained using $W = \text{null}(C)'$
 - Remark: result differs from the one found using previous method

Static parity space - detection

• Exercise

$$\bullet y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix}$$

- 1 Determine a parity matrix W
- 2 Find the evaluation form of the residual
- 3 Does this residual generator allow to detect all faults?
- 4 Same question with an additional fault $f_3(k)$ affecting measure $y_4(k)$:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix}$$

→ same exercise using MATLAB

Static parity space - detection

- Robustness to disturbances
 - Model: $y(k) = Cx(k) + D_d d(k) + D_f f(k)$
 - Residual $r(k) = WCx(k) + WD_d d(k) + WD_f f(k) = WD_d d(k) + WD_f f(k)$
with W s.t. $WC = 0$
 - A posteriori verification of robustness
 - if $WD_d = 0$: insensitivity to disturbances
 - How to take into account robustness a priori?

Static parity space - detection

- Robustness to disturbances

- Model: $y(k) = Cx(k) + D_d d(k) + D_f f(k)$
- Residual $r(k) = WCx(k) + WD_d d(k) + WD_f f(k) = WD_d d(k) + WD_f f(k)$
with W s.t. $WC = 0$
- A posteriori verification of robustness
 - if $WD_d = 0$: insensitivity to disturbances
- How to take into account robustness a priori?
 - ⇒ choose W s.t. $W [C \mid D_d] = 0$
if such a W exists...
 - existence condition: $p > \text{rank}([C \mid D_d])$
 - simplified existence condition: $p > (n + m_d)$ (if $[C \mid D_d]$ is full rank)
 - If such a W matrix does not exist
 - a scalar residual \bar{r} is computed as a linear combination of coordinates of r
 - ⇒ \bar{r} should be the most sensitive to faults and the least sensitive to disturbances

Static parity space - detection

- Robustness to disturbances

- Model: $y(k) = Cx(k) + D_d d(k) + D_f f(k)$
- Residual $r(k) = Wy(k) = WD_d d(k) + WD_f f(k)$ avec W tq $WC = 0$
- Objective

→ generate a residual scalar \bar{r} from r coordinates:

$$\bar{r}(k) = v^T r(k) = v^T WD_d d(k) + v^T WD_f f(k) \text{ avec } W \text{ tq } WC = 0$$

$v \in \mathbb{R}^{p-n}$ is called the residual selector

→ v chosen s.t. \bar{r} is the most sensitive to f and the least sensitive to d

- Criterion to be minimized

→ v is chosen in order to minimize criterion

$$J = \frac{\|v^T WD_d\|_2^2}{\|v^T WD_f\|_2^2} = \frac{v^T WD_d D_d^T W^T v}{v^T WD_f D_f^T W^T v}$$

- Optimal selector w.r.t. criterion J

$$\bullet v^* = \arg \min_v \frac{v^T WD_d D_d^T W^T v}{v^T WD_f D_f^T W^T v}$$

→ how to compute v^* ? \Rightarrow Gantmacher theorem

Static parity space - detection

- Robustness to disturbances

Gantmacher theorem

[Theory of matrices, 1961]

Vector $v^* = \arg \min_v \frac{v^T M v}{v^T N v}$ is the eigenvector associated to the smallest eigenvalue λ_{min} of matrix pencil (M, N) and $\min_v \frac{v^T M v}{v^T N v} = \lambda_{min}$

- Reminder on real matrices pencil
 - Matrix pencil associated to square matrices $M \in \mathbb{R}^{n \times n}$ and $N \in \mathbb{R}^{n \times n}$ is the set of matrices $P(\alpha) = M + \alpha N = (M, N)$, $\alpha \in \mathbb{R}$
 - Eigenvalues of matrix pencil (M, N) :
 - (M, N) has n eigenvalues
 - let q the number of null eigenvalues of N , then (M, N) has q infinite eigenvalues (equal to $+\infty$) and $n - q$ finite eigenvalues
 - the $n - q$ finite eigenvalues of (M, N) are $\lambda \in \mathbb{C} : \det(M - \lambda N) = 0$
 - Eigenvector V_i associated to eigenvalue λ_i is $V_i \in \mathbb{C}^n : M V_i = \lambda_i N V_i$
 - if $M = M^T$ and $N = N^T \Rightarrow \lambda \in \mathbb{R}$, $V_i \in \mathbb{R}^n$

Static parity space - detection

- Robustness to disturbances

- Residual that is the most sensitive to f and the least sensitive to d :

$$\rightarrow \bar{r}(k) = v^{*T} r(k) = v^T W D_d d(k) + v^T W D_f f(k) \text{ with } W \text{ tq } WC = 0$$

$$\rightarrow v^* = \arg \min_v \frac{v^T W D_d D_d^T W^T v}{v^T W D_f D_f^T W^T v}$$

- Method to determine optimal selector

① Find W s.t. $WC = 0$

② Determine eigenvalues λ of pencil $(W D_d D_d^T W^T, W D_f D_f^T W^T)$

③ Determine eigenvector v^* associated to the smallest eigenvalue λ_{min}

- Criterion optimal value is λ_{min} :

$$\rightarrow \min_v \frac{v^T W D_d D_d^T W^T v}{v^T W D_f D_f^T W^T v} = \lambda_{min}$$

Static parity space - detection

• Exercise

$$\bullet y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix}$$

→ $d_1(k)$ is the noise affecting measures 1, 2 and 3

→ $d_2(k)$ is the noise affecting measures 4 and 5

- 1 Remind the computation for of the residual $r(k) = Wy(k)$ that is insensitive to $x(k)$
 - 2 Give the evaluation form of $r(k)$ depending on $d(k)$ and $f(k)$.
 - 3 Is $r(k)$ sensitive to $d(k)$? Is it possible to derive a residual that is insensitive to $x(k)$ and $d(k)$?
 - 4 Determine $\bar{r}(k)$ the most sensitive to $f(k)$ and the least sensitive to $d(k)$
 - 5 Give evaluation form of $\bar{r}(k)$ depending on $d(k)$ and $f(k)$
 - 6 Compute criterion J value for $r_1(k)$, $r_2(k)$ and $\bar{r}(k)$. Conclude on the achieved improvement.
- Verify the results using MATLAB

Static parity space - detection

- Decoupling w.r.t. given faults

- Model: $y(k) = Cx(k) + D_f^+ f^+(k) + D_f^- f^-(k)$

- residual should be the most sensitive to faults $f^+(k)$

- residual should be the least sensitive to faults $f^-(k)$

- Residual obtained using a selector (*same method as previously*)

- $r_+(k) = v_+^{*T} r(k) = v_+^T W D_f^+ f^+(k) + v_+^T W D_f^- f^-(k)$ avec W tq $WC = 0$

- $v_+^* = \arg \min_{v_+} \frac{v_+^T W D_f^- (D_f^-)^T W^T v_+}{v_+^T W D_f^+ (D_f^+)^T W^T v_+}$

- Method to determine the optimal selector

- Determine W s.t. $WC = 0$

- Determine eigenvalues λ of pencil $(W D_f^- (D_f^-)^T W^T, W D_f^+ (D_f^+)^T W^T)$

- Determine eigenvector v_+^* associated to the smallest eigenvalue λ_{min}

- Interest

- if a single fault $f^+(k)$ is chosen, residual is the most sensitive to this fault

- ⇒ using a bundle of such residual generators is a first approach to address the isolation problem

Static parity space - detection

• Exercice (new matrix C)

$$\bullet y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 0 & 2 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 0 & 0 \\ 2 & 5 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_1^-(k) \\ f_2^-(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 3 \\ 1 \\ 1 \end{bmatrix} f^+(k)$$

→ residual should be the most sensitive to $f^+(k)$

→ residual should be the least sensitive to $f_1^-(k)$ and $f_2^-(k)$

- ① Find W and give the computation form of the residual $r(k) = Wy(k)$ that is insensitive to $x(k)$
- ② Is it possible to determine a residual that is insensitive to $x(k)$ and $f^-(k)$?
- ③ Find the residual $r_+(k)$ that is the most sensitive to $f^+(k)$ and the least sensitive to $f^-(k)$
 - in order to obtain a single solution, set the second coordinate of v_+^* to 1
- ④ Give the evaluation form of the residual $r_+(k)$ depending on $f^+(k)$ and $f^-(k)$
- ⑤ Compute the criterion J value for $r_1(k)$, $r_2(k)$ and $r_+(k)$. Conclude on the quality of the obtained residual.

Static parity space - isolation

- Residual expression in disturbance-free case
 - $r(k) = Wy(k) = WCx(k) + WD_f f(k)$
 - $x(k) \in \mathbb{R}^n$, $f(k) \in \mathbb{R}^{m_f}$, $r(k) \in \mathbb{R}^{p-\text{rank}(C)}$, $W \in \mathbb{R}^{(p-\text{rank}(C)) \times p}$ s.t. $WC = 0$
- Isolation problem
 - After the detection of the fault (here, fault is detected when $r(k) \neq 0$)
 - from the knowledge of $r(k)$, how to determine which fault occurred?
 - ⇒ which coordinate, among the m_f coordinates of $f(k)$, is not null?
- Solution
 - Residuals $r(k)$ move inside a space of dimension $p - \text{rank}(C)$
 - direction of $r(k)$ is a signature of a given fault
 - Columns of WD_f constitute the m_f directions to which the residual $r(k)$ point in presence of a fault
 - in order to isolate the fault, direction of residual $r(k)$ is computed and compared to WD_f

Static parity space - isolation

- Residual expression in disturbance-free case

- $$r(k) = Wy(k) = WCx(k) + WD_f f(k), \quad x(k) \in \mathbb{R}^n, \quad f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_{m_f}(k) \end{bmatrix} \in \mathbb{R}^{m_f}, \quad r(k) \in \mathbb{R}^{p-\text{rank}(C)}$$
- $$\rightarrow W \in \mathbb{R}^{(p-\text{rank}(C)) \times p} \quad \text{s.t.} \quad WC = 0,$$
- $$W_{rf} = WD_f \in \mathbb{R}^{(p-\text{rank}(C)) \times m_f}, \quad W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & \dots & W_{rf}^{[m_f]} \end{bmatrix}$$

- Isolation method (disturbance-free case)

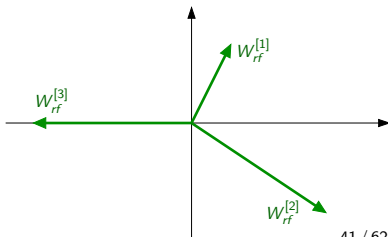
- orientation of $r(k)$ w.r.t. directions given by WD_f
- \Rightarrow collinearity of $r(k)$ and m_f vectors $W_{rf}^{[i]} \in \mathbb{R}^{p-\text{rank}(C)}$ is studied

if $r(k)$ is collinear to $W_{rf}^{[i]} \Rightarrow$ fault f_i is isolated

- Example: $p - \text{rank}(C) = 2$ and $m_f = 3$

- objective: isolate 1 fault among 3
- 3 faults directions

$$\rightarrow W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & W_{rf}^{[2]} & W_{rf}^{[3]} \end{bmatrix}$$



Static parity space - isolation

- Residual expression in disturbance-free case

- $$r(k) = Wy(k) = WCx(k) + WD_f f(k), \quad x(k) \in \mathbb{R}^n, \quad f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_{m_f}(k) \end{bmatrix} \in \mathbb{R}^{m_f}, \quad r(k) \in \mathbb{R}^{p-\text{rank}(C)}$$
- $$\rightarrow W \in \mathbb{R}^{(p-\text{rank}(C)) \times p} \quad \text{s.t.} \quad WC = 0,$$
- $$W_{rf} = WD_f \in \mathbb{R}^{(p-\text{rank}(C)) \times m_f}, \quad W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & \dots & W_{rf}^{[m_f]} \end{bmatrix}$$

- Isolation method (disturbance-free case)

- orientation of $r(k)$ w.r.t. directions given by WD_f
- \Rightarrow collinearity of $r(k)$ and m_f vectors $W_{rf}^{[i]} \in \mathbb{R}^{p-\text{rank}(C)}$ is studied

if $r(k)$ is collinear to $W_{rf}^{[i]} \Rightarrow$ fault f_i is isolated

- Example: $p - \text{rank}(C) = 2$ and $m_f = 3$

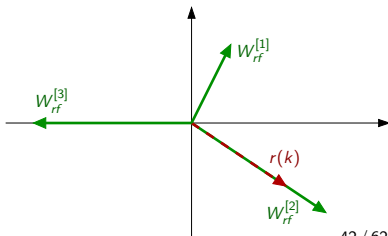
- objective: isolate 1 fault among 3

- 3 faults directions

$$\rightarrow W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & W_{rf}^{[2]} & W_{rf}^{[3]} \end{bmatrix}$$

- here r is colinear to $W_{rf}^{[2]}$

\Rightarrow *fault f_2 is isolated*



Static parity space - isolation

- Residual expression in disturbance-free case

- $$r(k) = Wy(k) = WCx(k) + WD_f f(k), \quad x(k) \in \mathbb{R}^n, f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_{m_f}(k) \end{bmatrix} \in \mathbb{R}^{m_f}, r(k) \in \mathbb{R}^{p-\text{rank}(C)}$$

$$\rightarrow W \in \mathbb{R}^{(p-\text{rank}(C)) \times n} \text{ s.t. } WC = 0, W = [W^{[1]} \ \dots \ W^{[m_f]}]$$

- Isolation method (disturbance-free case)

- Special case: detection of sensors faults

- $$f(k) = \begin{bmatrix} f_1(k) \\ \vdots \\ f_p(k) \end{bmatrix} \in \mathbb{R}^p \Rightarrow 1 \text{ fault by sensor}$$

$\rightarrow D_f = I_p \Rightarrow W_{rf} = W$

\Rightarrow the p columns of W define the p directions of sensors faults

\Rightarrow if r is collinear to $W^{[i]} \Rightarrow$ sensor fault on i^{th} measure

Static parity space - isolation

- Residual expression in presence of disturbances

- $r(k) = Wy(k) = WCx(k) + WD_f f(k) + WD_d d(k)$

- $\rightarrow r(k) \in \mathbb{R}^{p-\text{rank}(C)}$, $x(k) \in \mathbb{R}^n$, $f(k) \in \mathbb{R}^{m_f}$, $d(k) \in \mathbb{R}^{m_d}$, $W \in \mathbb{R}^{(p-\text{rank}(C)) \times n}$ s.t. $WC = 0$

- $\rightarrow W_{rf} = WD_f = [W_{rf}^{[1]} \dots W_{rf}^{[m_f]}] \in \mathbb{R}^{(p-\text{rank}(C)) \times m_f}$, $W_{rd} = WD_d = [W_{rd}^{[1]} \dots W_{rd}^{[m_d]}] \in \mathbb{R}^{(p-\text{rank}(C)) \times m_d}$

- Isolation method (in presence of disturbances)

- orientation of $r(k)$ depends on

- \rightarrow the m_f directions of faults given by columns of W_{rf}

- \rightarrow des m_d directions of disturbances given by columns of W_{rd}

if $r(k)$ is the most collinear to $W_{rf}^{[i]} \Rightarrow$ fault f_i is isolated

- Example: $p - n = 2$, $m_f = 3$, $m_d = 2$

- objective: isolate 1 fault among 3

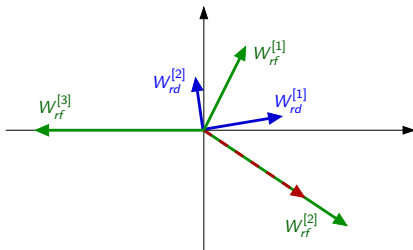
- 3 directions for faults

- $\rightarrow W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & W_{rf}^{[2]} & W_{rf}^{[3]} \end{bmatrix}$

- 3 directions for disturbances

- $\rightarrow W_{rd} = \begin{bmatrix} W_{rd}^{[1]} & W_{rd}^{[2]} \end{bmatrix}$

- here r is the most collinear to $W_{rf}^{[2]}$



Static parity space - isolation

- Residual expression in presence of disturbances

- $r(k) = Wy(k) = WCx(k) + WD_f f(k) + WD_d d(k)$

- $\rightarrow r(k) \in \mathbb{R}^{p-\text{rank}(C)}$, $x(k) \in \mathbb{R}^n$, $f(k) \in \mathbb{R}^{m_f}$, $d(k) \in \mathbb{R}^{m_d}$, $W \in \mathbb{R}^{(p-\text{rank}(C)) \times n}$ s.t. $WC = 0$

- $\rightarrow W_{rf} = WD_f = [W_{rf}^{[1]} \dots W_{rf}^{[m_f]}] \in \mathbb{R}^{(p-\text{rank}(C)) \times m_f}$, $W_{rd} = WD_d = [W_{rd}^{[1]} \dots W_{rd}^{[m_d]}] \in \mathbb{R}^{(p-\text{rank}(C)) \times m_d}$

- Isolation method (in presence of disturbances)

- orientation of $r(k)$ depends on

- \rightarrow the m_f directions of faults given by columns of W_{rf}

- \rightarrow des m_d directions of disturbances given by columns of W_{rd}

if $r(k)$ is the most collinear to $W_{rf}^{[i]} \Rightarrow$ fault f_i is isolated

- Example: $p - n = 2$, $m_f = 3$, $m_d = 2$

- objective: isolate 1 fault among 3

- 3 directions for faults

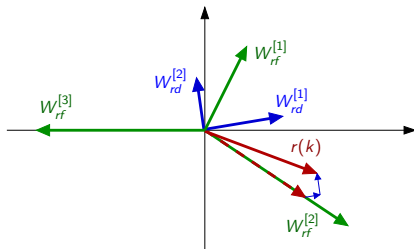
- $\rightarrow W_{rf} = \begin{bmatrix} W_{rf}^{[1]} & W_{rf}^{[2]} & W_{rf}^{[3]} \end{bmatrix}$

- 3 directions for disturbances

- $\rightarrow W_{rd} = \begin{bmatrix} W_{rd}^{[1]} & W_{rd}^{[2]} \end{bmatrix}$

- here r is the most collinear to $W_{rf}^{[2]}$

- \Rightarrow fault f_2 is isolated \Rightarrow how to evaluate collinearity using MATLAB?

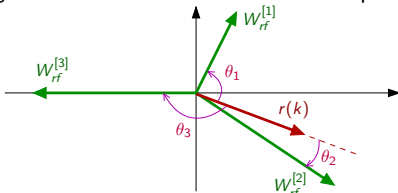


Static parity space - isolation

- Residual expression in presence of disturbances

if $r(k)$ is **the most collinear** to $W_{rf}^{[i]} \Rightarrow$ fault f_i isolated

- How to evaluate collinearity of two vectors?
 - the angle between the vectors can be computed



\rightarrow angle θ_i retrieved from scalar product $r^T \cdot W_{rf}^{[i]} = \|r\| \cdot \|W_{rf}^{[i]}\| \cdot \cos(\theta_i)$

$$\bullet \theta_i = \arccos \frac{r^T W_{rf}^{[i]}}{\|r\| \cdot \|W_{rf}^{[i]}\|} \Rightarrow \theta_i = \arccos \frac{r^T W_{rf}^{[i]}}{\sqrt{r^T r} \sqrt{W_{rf}^{[i]T} W_{rf}^{[i]}}}$$

\rightarrow the smallest θ_i is, the most r and $W_{rf}^{[i]}$ are collinear

\rightarrow fault f_i is isolated where $i = \arg \min_i \theta_i$

\rightarrow considering above example: $\min \theta_i = \theta_2 \Rightarrow$ fault f_2 isolated

Static parity space - isolation

• Exercise

$$\bullet \quad y(k) = \begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \\ y_4(k) \\ y_5(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 2 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 2 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix}$$

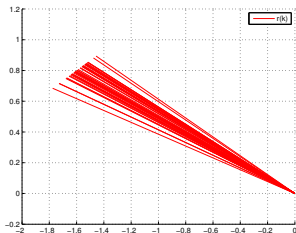
→ $d_1(k)$ is the noise affecting measures 1, 2 and 3

→ $d_2(k)$ is the noise affecting measures 4 and 5

- ① Remind the computation for of the residual $r(k) = Wy(k)$ that is insensitive to $x(k)$.
 - ② Give the evaluation form of $r(k)$ depending on $d(k)$ and $f(k)$ and note matrices $W_{rd} = WD_d$ and $W_{rf} = WD_f$ values.
 - ③ After fault occurrence at time $k = 50$, $r(60) = \begin{bmatrix} -2.09 \\ 1.44 \end{bmatrix}$. Isolate the fault using a graphical approach.
 - ④ Confirm the validity of the conclusion by computing angles θ_i
- *Verify the results using MATLAB*

Static parity space - isolation

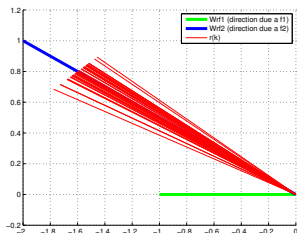
- Exercise
 - Taking into account the evolution of $r(k)$
 - another fault leads to the $r(k)$ vectors below



→ Which fault can be evaluated?

Static parity space - isolation

- Exercise
 - Taking into account the evolution of $r(k)$
 - another fault leads to the $r(k)$ vectors below



- Which fault can be evaluated?
- ⇒ fault f_2 is isolated (since $r(k)$ is oriented along $W_{rf}^{[2]}$)

Static parity space - recap

- Model: $y(k) = Cx(k) + D_d d(k) + D_f f(k)$, $y \in \mathbb{R}^p, x \in \mathbb{R}^n, d \in \mathbb{R}^{m_d}, f \in \mathbb{R}^{m_f}$
 - Residual : $r(k) = Wy(k) = WCx(k) + WD_d d(k) + WD_f f(k)$
 - Insensitivity to state: find W s.t. $WC = 0$ ($W \in \mathbb{R}^{(p-\text{rank}(C)) \times n}$)
 - Sensitivity to faults verified a posteriori:
if column i of WD_f non null \Rightarrow fault f_i detectable with $r(k)$
 - Perfect robustness to disturbances: find W s.t. $W [C \ D_d] = 0$
 - Residual the most sensitive to faults and the least sensitive to disturbances: $\bar{r}(k) = v^T Wy(k)$ (v obtained using Gantmacher theorem)
 - Faults isolation: r the most collinear to column i of $WD_f \Rightarrow f_i$ isolated
 - Static parity space approach limitations
 - \rightarrow only sensor faults can be detected and isolated
 - \rightarrow what if W does not exist? (no direct redundancy between measures)
- \Rightarrow A solution: use time redundancy between inputs and outputs at different time instants \Rightarrow dynamic parity space

Dynamic parity space - principle

- Model

- $\Sigma_k \begin{cases} x(k+1) = Ax(k) + Bu(k) + B_d d(k) + B_f f(k) \\ y(k) = Cx(k) + Du(k) + D_d d(k) + D_f f(k) \end{cases}$

→ $x \in \mathbb{R}^n, y \in \mathbb{R}^p, u \in \mathbb{R}^m, d \in \mathbb{R}^{m_d}, f \in \mathbb{R}^{m_f}$

- Idea

- use time redundancy between inputs and outputs at different time instants
- measures $y(k)$ et controls $u(k)$ collected on a time-window

- Model on time-window $[k-s, k]$

$$Y(k-s, k) = \Phi_U(s)U(k-s, k) + Q_o(s)x(k-s) + \Phi_D(s)D(k-s, k) + \Phi_F(s)F(k-s, k)$$

→ s : size of time window

→ $Y(k-s, k) = \begin{bmatrix} y(k-s) \\ y(k-s+1) \\ \vdots \\ y(k) \end{bmatrix}, U(k-s, k) = \begin{bmatrix} u(k-s) \\ u(k-s+1) \\ \vdots \\ u(k) \end{bmatrix}, D(k, s) = \dots$

→ exercise: find expressions of $\Phi_U(s), Q_o(s), \Phi_D(s)$ et $\Phi_F(s)$
(use recursive approach to determine $y(k-s)$, then $y(k-s+1)$...)

Dynamic parity space - principle

- Model

- $$\Sigma_k \begin{cases} x(k+1) = Ax(k) + Bu(k) + B_d d(k) + B_f f(k) \\ y(k) = Cx(k) + Du(k) + D_d d(k) + D_f f(k) \end{cases}$$

- Model on time-window $[k-s, k]$

$$Y(k-s, k) = \Phi_U(s)U(k-s, k) + Q_o(s)x(k-s) + \Phi_D(s)D(k-s, k) + \Phi_F(s)F(k-s, k)$$

$$\underbrace{\begin{bmatrix} y(k-s) \\ y(k-s+1) \\ y(k-s+2) \\ \vdots \\ y(k) \end{bmatrix}}_{Y(k-s, k)} = \underbrace{\begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^s \end{bmatrix}}_{Q_o(s)} x(k-s) + \underbrace{\begin{bmatrix} D & 0 & \dots & 0 & 0 \\ CB & D & \dots & 0 & 0 \\ CAB & CB & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & D & 0 \\ CA^{s-1}B & CA^{s-2}B & \dots & CB & D \end{bmatrix}}_{\Phi_U(s)} \underbrace{\begin{bmatrix} u(k-s) \\ u(k-s+1) \\ u(k-s+2) \\ \vdots \\ u(k) \end{bmatrix}}_{U(k-s, k)}$$

$$+ \underbrace{\begin{bmatrix} D_d & 0 & \dots & 0 & 0 \\ CB_d & D_d & \dots & 0 & 0 \\ CAB_d & CB_d & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & D_d & 0 \\ CA^{s-1}B_d & CA^{s-2}B_d & \dots & CB_d & D_d \end{bmatrix}}_{\Phi_D(s)} \underbrace{\begin{bmatrix} d(k-s) \\ d(k-s+1) \\ d(k-s+2) \\ \vdots \\ d(k) \end{bmatrix}}_{D(k-s, k)} + \underbrace{\begin{bmatrix} D_f & 0 & \dots & 0 & 0 \\ CB_f & D_f & \dots & 0 & 0 \\ CAB_f & CB_f & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & D_f & 0 \\ CA^{s-1}B_f & CA^{s-2}B_f & \dots & CB_f & D_f \end{bmatrix}}_{\Phi_F(s)} \underbrace{\begin{bmatrix} f(k-s) \\ f(k-s+1) \\ f(k-s+2) \\ \vdots \\ f(k) \end{bmatrix}}_{F(k-s, k)}$$

→ objective: find $r(k) = f(Y(k-s, k), U(k-s, k))$ insensitive to $x(k)$

Dynamic parity space - principle

- Model on time-window $[k - s, k]$

$$Y(k-s, k) = \Phi_U(s)U(k-s, k) + Q_o(s)x(k-s) + \Phi_D(s)D(k-s, k) + \Phi_F(s)F(k-s, k)$$

- Residual obtained by collected measures and controls linear combination

- $r(k) = W(Y(k-s, k) - \Phi_U(s)U(k-s, k))$ (computation form)

→ takes advantage of redundancy between u and y at different times

⇒ inter-redundancy

Remark

- Scalar residuals $\tilde{r}_j(k)$ can be generated using time redundancy of a single measure $y_j(k)$ ⇒ auto-redundancy

$$\rightarrow \tilde{r}_j(k) = \tilde{W}_j \left(\tilde{Y}_j(k-s, k) - \tilde{\Phi}_U^j(s)U(k-s, k) \right)$$

$$\tilde{Y}_j(k-s, k) = \tilde{\Phi}_U^j(s)U(k-s, k) + \tilde{Q}_o^j(s)x(k-s) + \tilde{\Phi}_D^j(s)D(k-s, k) + \tilde{\Phi}_F^j(s)F(k-s, k)$$

→ $\tilde{\Phi}_U^j(s), \tilde{Q}_o^j(s), \tilde{\Phi}_D^j(s), \tilde{\Phi}_F^j(s)$ obtained when replacing C, D, D_d, D_f by their j^{th} line

⇒ simplifies sensor faults isolation

Dynamic parity space - detection

- Model on time-window $[k - s, k]$
 - $Y(k-s, k) = \Phi_U(s)U(k-s, k) + Q_o(s)x(k-s) + \Phi_D(s)D(k-s, k) + \Phi_F(s)F(k-s, k)$
 - Computation form
 - $r(k) = W(Y(k-s, k) - \Phi_U(s)U(k-s, k))$
 - Evaluation form
 - $r(k) = WQ_o(s)x(k-s) + W\Phi_D(s)D(k-s, k) + W\Phi_F(s)F(k-s, k)$
 - Insensitivity to state
 - residual is insensitive to state if W s.t. $WQ_o(s) = 0$
 - existence condition: W exists if $p(s+1) > \text{rank}(Q_o(s))$
- $W \in \mathbb{R}^{(p(s+1) - \text{rank}(Q_o(s))) \times p(s+1)}$
- W determined using the method presented in *static parity space* approach

Dynamic parity space - detection

• Evaluation form

$$r(k) = WQ_o(s)x(k-s) + W\Phi_D(s)D(k-s,k) + W\Phi_F(s)F(k-s,k)$$

→ with W s.t. $WQ_o(s) = 0$

• Sensitivity to faults: a posteriori verification

- faults effect on residual evaluated using matrix $W_{rF} = W\Phi_F(s)$
- in disturbance-free case, residual writes

$$\rightarrow r(k) = \left[W_{rF}^{[1]} \dots W_{rF}^{[m_f]} \mid W_{rF}^{[m_f+1]} \dots W_{rF}^{[2m_f]} \mid \dots \mid W_{rF}^{[sm_f+1]} \dots W_{rF}^{[(s+1)m_f]} \right] \begin{bmatrix} f_1(k-s) \\ \dots \\ f_{m_f}(k-s) \\ \hline f_1(k-s+1) \\ \dots \\ f_{m_f}(k-s+1) \\ \hline \dots \\ \hline f_1(k) \\ \dots \\ f_{m_f}(k) \end{bmatrix}$$

- if a single constant fault $f_i(k)$ occurs, after s samples residual writes:

$$\rightarrow r(k) = W_{rF}^{[i]} f_i(k-s) + W_{rF}^{[m_f+i]} f_i(k-s) + \dots + W_{rF}^{[sm_f+i]} f_i(k-s)$$

$$= (W_{rF}^{[i]} + W_{rF}^{[m_f+i]} + \dots + W_{rF}^{[sm_f+i]}) f_i(k-s)$$

$$f_i \text{ strongly detectable} \Leftrightarrow \sum_{j=0}^s W_{rF}^{[jm_f+i]} \neq 0$$

$$f_i \text{ weakly detectable} \Leftrightarrow \exists q \leq s \text{ s.t. } \sum_{j=s-q}^s W_{rF}^{[jm_f+i]} \neq 0$$

Dynamic parity space - detection

- Evaluation form

- $r(k) = WQ_o(s)x(k-s) + W\Phi_D(s)D(k-s,k) + W\Phi_F(s)F(k-s,k)$

- with W s.t. $WQ_o(s) = 0$

- Robustness to disturbances: verified a posteriori

- disturbances effect on residual evaluated using $W_{rD} = W\Phi_D(s)$

residual insensitive to d_i if all the columns of $W_{rD}^{[i]}, W_{rD}^{[m_d+i]}, \dots, W_{rD}^{[sm_d+i]}$ are null

- Perfect robustness to disturbances: a priori requirement

- choose W s.t. $W\Phi_D(s) = 0 \Rightarrow$ choisir W s.t. $W [Q_o(s) \quad \Phi_D(s)] = 0$

- such a matrix W exists if $p(s+1) > \text{rank}([Q_o(s) \quad \Phi_D(s)])$

- condition rarely met

- ⇒ seek for a residual the most sensitive to faults and the least sensitive to disturbances

Dynamic parity space - detection

- Evaluation form

- $$r(k) = WQ_o(s)x(k-s) + W\Phi_D(s)D(k-s,k) + W\Phi_F(s)F(k-s,k)$$

→ with W s.t. $WQ_o(s) = 0$

- Residual the most sensitive to faults and the least sensitive to disturbances

- scalar residual obtained from coordinates of $r(k)$

→
$$\bar{r}(k) = v^T r(k) = v^T W\Phi_D(s)D(k-s,k) + v^T W\Phi_F(s)F(k-s,k)$$

- choice of the residual selector

→
$$v^* = \arg \min_v \frac{\|v^T W\Phi_D(s)\|_2^2}{\|v^T W\Phi_F(s)\|_2^2} = \arg \min_v \frac{v^T W\Phi_D(s)\Phi_D^T(s)W^T v}{v^T W\Phi_F(s)\Phi_F^T(s)W^T v}$$

→ method to determine v^*

- 1 Determine W s.t. $WQ_o(s) = 0$
- 2 Compute eigenvalues λ of pencil $(W\Phi_D(s)\Phi_D^T(s)W^T, W\Phi_F(s)\Phi_F^T(s)W^T)$
- 3 Compute the eigenvector v^* associated to the smallest eigenvalue λ_{min}

Dynamic parity space - Exercise

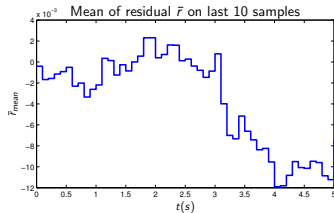
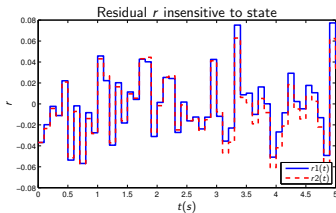
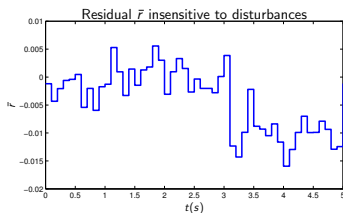
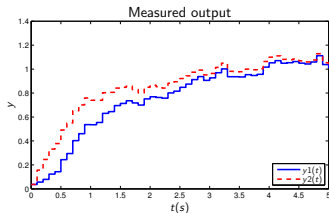
$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0 & 0.9 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(k) + \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0.1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix}$$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_1(k) \\ d_2(k) \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1(k) \\ f_2(k) \\ f_3(k) \end{bmatrix}$$

- $f_1(k)$ is an actuator fault, $f_2(k)$ and $f_3(k)$ are sensor faults
- $d_1(k)$ is a state disturbance and $d_2(k)$ is a measurement noise
- ① Is it possible to apply static parity space approach?
- ② What is the minimal window size s_{min} s.t. a residual insensitive to x exists?
- ③ Find W and give the computation form of the residual depending on collected measures and controls.
- ④ Give the evaluation form depending on collected faults and disturbances. Retrieve the sensors and actuator faults directions.
- ⑤ 3 samples after a single fault, $r(k) = \begin{bmatrix} 0.02 \\ -0.2 \end{bmatrix}$. Which fault occurred?
- ⑥ Give expression of scalar residual $\bar{r}(k)$ that is the least sensitive to disturbances (for $s = s_{min}$) and evaluate its quality w.r.t. $r_1(k)$ and $r_2(k)$
- ⇒ Verify the results using MATLAB

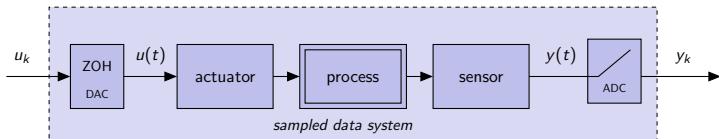
Dynamic parity space - Exercise

- Simulation results in presence of white noise and a constant fault f_1



Reminder on sampled data systems

- Bloc diagram involving D/A converters for control and FDI purposes



- actuators + plant + sensors $\equiv \Sigma$ modeled by state space representation

$$\rightarrow \Sigma: \begin{cases} \dot{x}(t) &= \tilde{A}x(t) + \tilde{B}u(t) \\ y(t) &= \tilde{C}x(t) + \tilde{D}u(t) \end{cases}$$

- actuators + process + sensors + DAC + ADC $\equiv \Sigma_k$

$$\rightarrow \Sigma_k: \begin{cases} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{cases}$$

$$\text{with } A = e^{\tilde{A}T_e}, \quad B = \int_0^{T_e} e^{\tilde{A}(T_e-\alpha)} \tilde{B} d\alpha, \quad C = \tilde{C}, \quad D = \tilde{D}$$

Reminder on sampled data systems

- Solution of state space equations of the continuous-time model Σ

$$x(t) = e^{\tilde{A}t}x(0) + \int_0^t e^{\tilde{A}(t-\tau)}\tilde{B}u(\tau)d\tau$$

- State value at time $t = kT_e$ and $t = (k+1)T_e$ is

$$x(kT_e) = e^{\tilde{A}kT_e}x(0) + \int_0^{kT_e} e^{\tilde{A}(kT_e-\tau)}\tilde{B}u(\tau)d\tau$$

$$x((k+1)T_e) = e^{\tilde{A}(k+1)T_e}x(0) + \int_0^{(k+1)T_e} e^{\tilde{A}((k+1)T_e-\tau)}\tilde{B}u(\tau)d\tau$$

$$= e^{\tilde{A}T_e} \left[e^{\tilde{A}kT_e}x(0) + \int_0^{(k+1)T_e} e^{\tilde{A}(kT_e-\tau)}\tilde{B}u(\tau)d\tau \right]$$

$$= e^{\tilde{A}T_e} \left[\underbrace{e^{\tilde{A}kT_e}x(0) + \int_0^{kT_e} e^{\tilde{A}(kT_e-\tau)}\tilde{B}u(\tau)d\tau}_{x(kT_e)} + \int_{kT_e}^{(k+1)T_e} e^{\tilde{A}(kT_e-\tau)}\tilde{B}u(\tau)d\tau \right]$$

Reminder on sampled data systems

- State value at time $t = (k + 1)T_e$ is

$$x((k + 1)T_e) = e^{\tilde{A}T_e} \left[x(kT_e) + \int_{kT_e}^{(k+1)T_e} e^{\tilde{A}(kT_e - \tau)} \tilde{B}u(\tau) d\tau \right]$$

- With change of variables $\alpha = \tau - kT_e$

$$x((k + 1)T_e) = e^{\tilde{A}T_e} \left[x(kT_e) + \int_0^{T_e} e^{-\tilde{A}\alpha} \tilde{B}u(\alpha + kT_e) d\alpha \right]$$

- Because of the zero-order hold, $u(kT_e + \alpha) = u(kT_e)$ if $0 < \alpha < T_e$

$$x((k + 1)T_e) = e^{\tilde{A}T_e} \left[x(kT_e) + \int_0^{T_e} e^{-\tilde{A}\alpha} \tilde{B}u(kT_e) d\alpha \right]$$

- As $u(kT_e)$ does not depend on α

$$\underbrace{x((k + 1)T_e)}_{x_{k+1}} = \underbrace{e^{\tilde{A}T_e}}_A \underbrace{x(kT_e)}_{x_k} + \underbrace{\int_0^{T_e} e^{\tilde{A}(T_e - \alpha)} \tilde{B} d\alpha}_B \underbrace{u(kT_e)}_{u_k}$$