Nancy-Université
Université Henri Poincaré

## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

➢ Contact SCD Nancy 1 : theses.sciences@scd.uhp-nancy.fr

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Nancy-Université
Université Henri Poincaré

CRAN

Département de formation doctorale en automatique
UFR STMIA

École doctorale IAEM Lorraine

# Synthèse intégrée du diagnostic de systèmes contrôlés en réseaux avec contraintes de communication

# THÈSE

présentée et soutenue publiquement le 19 Juillet 2011

pour l'obtention du titre de

**Docteur de l'université Henri Poincaré**

**en Automatique, Traitement du Signal, Génie Informatique**

par

Hossein HASHEMI NEJAD

**Composition du jury**

| | | |
|---|---|---|
| *Rapporteurs :* | Mr. Steven X. Ding | Université de Duisburg-Essen, Allemagne |
| | Mr. Daniel SIMON | INRIA Rhône-Alpes, France |
| *Examinateurs :* | Mr. Ye-Qiong Song | LORIA, France |
| | Mr. Jean Marc Thiriet | GIPSA-LAB, France |
| | Mr. Youmin Zhang | Université de Concordia , Canada |
| | Mr. Dominique Sauter | CRAN, Nancy université, France (Directeur de thèse) |
| | Mr. Samir Aberkane | CRAN, Nancy université, France (Co-Directeur de thèse) |

Mis en page avec la classe thloria.

# Abstract

Networked Control systems (NCSs) have been one of research focuses in academia and industrial applications during the last few years. The advantages of NCSs over conventional or hardwired control include reduction of system wiring and increase in maintenance and troubleshooting facilities. Because of these attractive benefits, many industrial companies and institutes have shown interest in applying networks for remote industrial control purposes and factory automation. As a result of extensive research and development, several network protocols for industrial control have been released. However, several problems arise when a control loop is closed via a communication network. No matter what network is used, some network problems (e.g. delay, packet dropout, medium access constraints) can be affected a feedback loop when it is closed via communication network.

Fault diagnosis and fault-tolerant control are important issues for practical control systems, especially in safety-critical systems. The theory and application of classical approaches of fault diagnosis and fault tolerant control should be revised when dealing with NCSs. Objective of this thesis is proposing new approaches to design a fault detection and isolation (FDI) system with considering network-induced effects such as packet dropout and medium access constraints. In addition, proposed algorithms of scheduling and fault diagnosis are implemented in a mini helicopter.

First, Since networked-induces effects such as packet dropout and communication constraints must be considered in FDI design, an extended model for taking into account all these limitations was used. Then a strategy to develop a set of structured residuals was proposed. It guarantees robustness to packet dropout and unknown disturbances.

Regarding network access limitation, sometimes it is necessary to provide a pre-defined sequence (i.e. communication sequence) before designing FDI. It describes the instantaneous medium access status of the sensors and actuators. However, choice of a communication sequence is not trivial and that is depended on the structure of the system. In addition, in practice it is not easy to find a precise mathematical model. Proposed algorithm guarantees the generation of communication sequences which preserve some structural properties of the plant. Furthermore, this algorithm can be implemented on uncertain and large scale systems.

Traditionally, applications of allocation and scheduling techniques are based on offline strategies. But under offline scheduling, performance of diagnostic system may not be guaranteed when the plant is subject to unpredictable disturbances. In addition, online scheduling needs a large computation which may not be always possible in case of embedded system. Thus, a semi-online scheduling which preserve advantages of online scheduling and prevent some limitations of offline scheduling can be considered a compromise solution.

Miniature rotorcraft-based Unmanned Aerial Vehicles (UAVs) are currently subject of a lot of research due to their usefulness in situations that require unmanned or self-piloted operations. They can be classified as fast dynamic systems. So, they may be ideal benchtest for studying effects of the network on performance of closed loop control of the system. The sixth chapter of this dissertation is dedicated to implementation of a fault tolerant control strategy and FDI approach proposed in previous chapters on a quadrotor application.



**Keywords:** Networked control systems, fault detection and isolation,scheduling, quadrotor

# Résumé

Les systèmes contrôlés en réseau (SCR) ont fait l'objet de nombreux travaux de recherche au cours des dernières années, principalement pour ce qui concerne la synthèse de lois de commande. Les systèmes contrôlés en réseau présentent de nombreux avantages, notamment en terme de flexibilité, mais différent problèmes se posent quand une boucle de contrôle est fermée par un réseau de communication. (ex. retards et des pertes, contraintes de communication).
Diagnostic et tolérance aux défauts sont des enjeux importants pour les systèmes de contrôle, particulièrement dans les systèmes de sécurité fondamentaux. La théorie et l'application des approches classiques de diagnostic et tolérance aux défauts doivent être révisités lorsqu'il s'agit de SCR. L'objectif de cette thèse est de proposer de nouvelles approches de diagnostic pour les systèmes contrôlés en réseau en considérant la perte de paquets et la contrainte de communication. De plus, les algorithmes de l'ordonnancement et de diagnostic proposés sont implémentés dans un mini hélicoptère.
Nous considérons d'abord les problèmes de perte de paquets et de contrainte de communication pour ensuite adapter un modèle où la détection des défauts et l'allocation des ressources de communication sont fortement liés. En interprétant ce modèle comme un modèle périodique, nous formalisons et résolvons le problème de détection et localisation de défauts avec un ordonnancement périodique et hors-ligne. L'approche proposée garantit la robustesse des résidus aux perturbations ainsi que perte de paquets sur la commande du système.
Il est parfois nécessaire de fournir une séquence de communication prédéfinie avant de concevoir le système de détection de défauts. IL spécifie l'ordre de l'accès des capteurs et des actionneurs au réseau. Cependant, le choix d'une séquence de communication dépend foncement à la structure du système. Un algorithme graphique proposé dans cette thèse garantit la génération de séquences de communication permettant de préserver certaines propriétés structurelles du système. En outre, cet algorithme peut être utilisé sur les systèmes incertains et assez grands.
Traditionnellement, allocation des ressources et l'ordonnancement sont basés sur les stratégies hors-ligne. Mais la performance du système de diagnostic ne peut pas être garantie sous l'ordonnancement hors-ligne, si le système est objet à des perturbations imprévisibles. En plus, L'ordonnancement en ligne nécessite une grande charge de calcul qui ne peut être toujours possible en cas de système embarqué. Par conséquent, un ordonnancement semi-en ligne qui permet de préserver les avantages de l'ordonnancement en ligne et évite certaines limitations d'ordonnancement hors ligne peut être considéré comme une solution de compromis.
Les drones ou UAVs pour Unmanned Aerial Vehicules font actuellement l'objet de beaucoup de recherches en raison de leurs utilités dans des situations qui nécessitent des opérations autonomes ou autopiloté. Ils peuvent être classés comme des systèmes dynamiques rapides. Par conséquent, ils peuvent être banc d'essai idéal pour étudier les effets du réseau sur les performances de contrôle/diagnostique en boucle fermée. Le dernier chapitre de cette thèse est dédié à implémentation d'une stratégie de commande tolérante aux défauts et les approches de détection et localisation des défauts proposées dans les chapitres précédents sur le drone.



**Mots-clés:** Systèmes contrôlés en réseau, détection et localisation de défauts, ordonnancement ,Quadrotor

# Contents

**Abstract**

**Résumé**

**1 Introduction** **7**

1.1 Why Networked control systems . . . 7

1.2 Safe-Necs(Safe-Networked Control Systems) Project . . . 8

1.3 Objectives and motivations . . . 9

1.4 Thesis outline and contributions . . . 10

1.5 Publications . . . 11

**2 Networked Control Systems** **13**

2.1 A brief history of NCSs . . . 13

2.2 NCSs Categories . . . 15

2.3 Medium access methods in communication networks . . . 16

2.3.1 CSMA/CA and CSMA/CD . . . 16

2.3.2 Token-Bus . . . 17

2.3.3 TDMA . . . 17

2.4 Fundamental issues in NCSs . . . 17

2.4.1 Network-induced Delay . . . 19

2.4.2 Packet dropout . . . 20

2.4.3 Medium access constraints . . . 20

2.4.4 Quantization and feedback control in NCS . . . 21

2.5 Model-based fault detection . . . 21

2.5.1 Fault detection of Networked control systems . . . 23

2.5.1.1 Fault detection in NCSs with network delays . . . 23

2.5.1.2 Fault detection of NCSs with packet losses . . . 24

**3 FDI with limited communication : Fixed scheduling case** **25**

3.1 Communication limitation modeling . . . 25

3.2 Observability and Reachability of extended plant . . . 28
3.3 Fault detection . . . 31
3.3.1 Observer-based approach . . . 33
3.3.2 FD subject to access constraints and Packet dropout . . . 34
3.4 FD in distributed system with communication constraints . . . 35
3.5 Design of structured residual sets in NCSs subject to random packet dropouts . . . 39
3.5.1 Fault isolation . . . 41
3.5.2 Illustrative example . . . 42
3.6 FDI of NCSs subject to random packet dropout and medium access constraints . . . 45
3.6.1 Design of robust residual generators . . . 45
3.6.2 Illustrative example . . . 48
3.7 Conclusion . . . 50

**4 Communication sequence design in NCS with communication constraints : a graphic approach 53**
4.1 Structured system . . . 55
4.2 Graphic representation of linear structured systems . . . 57
4.2.1 Directed graph . . . 57
4.2.1.1 Generic controllability and generic observability . . . 59
4.2.2 Dynamic bipartite graph . . . 61
4.2.3 Generic reachability . . . 63
4.3 Communication sequence design, observability . . . 65
4.4 Communication sequence design, reachability . . . 68
4.5 Conclusion . . . 70

**5 Fault detection in limited communication with dynamic scheduling 71**
5.1 Problem formulation . . . 71
5.2 Parity space-based residual generation and semi-online scheduling . . . 73
5.2.1 Offline communication sequence design . . . 74
5.2.2 Sequence selection . . . 75
5.2.2.1 Multi Sequential Probability Ratio Test(MSPRT) . . . 75
5.2.3 Fault detection . . . 78
5.2.4 Example . . . 78
5.3 Observer-based residual generation and semi-online scheduling . . . 81
5.4 Online scheduling in CAN network by means of hybrid priority . . . 83
5.4.1 CAN network . . . 84
5.4.2 Hybrid priority . . . 85

5.4.3 Hybrid priority with diagnostic objective . . . 86
5.5 Conclusion . . . 88

**6 Drone Application** **89**
6.1 Drone presentation . . . 89
6.1.1 Architecture of prototype of Safe-NECS project . . . 91
6.1.2 Drone movements . . . 92
6.1.3 Attitude representation . . . 93
6.1.3.1 Euler angles . . . 95
6.1.3.2 Quaternion . . . 96
6.1.4 Drone sensors . . . 98
6.1.5 Mechanical Model and attitude estimation . . . 99
6.2 Fault tolerant control . . . 105
6.2.1 Reconfiguration in case of critical failures . . . 106
6.2.2 Actuator and network faults and its effects . . . 107
6.2.3 Fault tolerance control module . . . 109
6.2.3.1 First strategy . . . 109
6.2.3.2 Second strategy . . . 110
6.2.3.3 Third strategy . . . 111
6.2.3.4 Fourth strategy . . . 112
6.3 Semi-online scheduling and fault detection . . . 113
6.3.1 Residual generation . . . 115
6.3.2 residual classification . . . 115
6.3.3 Sensor access scheduling . . . 116
6.3.4 Simulation results . . . 117

**7 Conclusion and future works** **121**

**List of figures** **125**

**List of tables** **129**

**Bibliography** **131**

# Chapter 1

# Introduction

Networked Control systems (NCSs) have been one of research focuses in academia and industrial applications during the last few years. Currently, researcher from varied disciplines are interested in both technological and commercial potentials challenges introduced by NCSs. For this class of systems, a communication network is used as medium in control loop. The role of the network is to ensure data transmission and coordinating manipulation among different system components. The classical definition of NCS which can be found in literature is [WL10] :
*When a traditional feedback control system is closed via a communication channel, which may be shared with other nodes outside the control system, then the control system is called*.

## 1.1 Why Networked control systems

The complexity of the control systems increase notably with their dimensions. Therefore, direct wirings now becomes troublesome and impractical for large scale systems. Connecting control system components (i.e. sensors, actuators and controllers) in these application via a network reduces system wiring. It may be one of the primary advantages of using network comparing with point-to-point connections. Network controllers allow data to be shared efficiently between system components. It enables system controller to fucus on global information and to take intelligent decision over a large physical space.
Secondly, quick development in communication techniques provided flexible system installation, manipulation and expansion with low cost. For example, it is easy to add more sensors or actuators without heavy structural changes in the whole system. Moreover, remote diagnostic and troubleshooting is much easier because of the modular nature of the system. It significantly increase reliability and efficiency of the whole system.
Thirdly, They connect cyber space to physical space making task execution from a distance easily accessible (see [Hsi04, LVL+05]).
Advantages of using a communication network in a control system was detailed in [MM04]. All these benefits had prompted the researchers to put increasing efforts in this field.
However, several problems arise when a control loop is closed via a communication network. No matter what network is used, the following network problems can be affected a feedback loop when it is closed via communication network.

a) Sampling rate constraints ;

b) Medium access constraints ;

c) Disturbances introduced by communication ;

d) Networked-induced delay in control commands and sensor measurements ;

e) Packet dropout ;

More recent researches to handel these problems can be divided into two directions :

a) *Control of network :* Study objective is to make a communication network suitable for real-time transmission. It deals with network itself. To build a communication network the following objectives must be considered [LW08] :
   - Communication network should provide fast response, low data loss ratio and high signal-to-noise ratio(SNR) ;
   - System security, especially for wireless NCS ;
   - System flexibility, i.e. plug-and-play installation/ uninstallation, on-demand bandwidth reservation and smart resources managements ;
   - provide the characterization of channels in forms which are more meaningful for control applications.

b) *Control over network :* It studies the closed loop system performance subject to the available network resources. Research objective is to design controllers or diagnostic systems with considering network-induced effects which provide acceptable performances.

However, recent achievements showed that it is possible to solve communication problems and control problems simultaneously.
Fault diagnosis and fault-tolerant control are important issues for practical control systems, specially in safety-critical systems. The theory and application of classical approaches of fault diagnosis and fault tolerant control can not be used in NCSs. For example, some network protocols can not guarantee instantaneous access to the medium for all systems components. Or some control commands/sensor measurements are received by actuators/controller with unbounded delays. So, the designer should take all these peculiarities and the limitations induced by network in diagnosis system design.

## 1.2 Safe-Necs(Safe-Networked Control Systems) Project

This thesis was supported by Safe-Necs(Safe-Networked Control Systems) research project. This project partially funded by the *National Agency for Research* (ANR) and brings together five teams of five French laboratories :

- CRAN (Research Center for Automatic Nancy) ;
- GIPSA-lab (Grenoble Images Speech Signal Control) Automation Department ;
- INRIA Rhone-Alpes (National Institute for Research in Computer Science and Control) ;
- LAAS (Laboratory for Analysis and Architecture of Systems) ;
- LORIA (Lorraine Laboratory of IT Research and its Applications).

The objectives of this project were twofold. The first was to implement a technical strategy to adapt the network according to the state of the controlled system. This was done by developing scheduling algorithms and proposing new communication protocols. The second objective was to propose the control algorithms and fault detection and isolation system with considering network problems (packet dropout, delay and communication constraints). Compared with earlier works and concurrent projects, Safe-NECS was designed to enhance the integration between control, real-time scheduling and networking. The challenge in this project was the mastering of three levels of regulation : the control closed loop, the supervision closed loop and the feedback scheduling closed loop. To carry out this project, different research

communities worked together. Researchers from automatic control community represented by CRAN and GIPSA-lab as well as real-time scheduling community represented by LORIA, INRIA Rhone-Alpes and LAAS cooperated in this project.
Particularly, the project aimed to develop a *co-design* approach that integrate in a coordinated way several kinds of parameters : the characteristics modeling the Quality of Control (QoC) as given by automatic control specialists, the dependability properties required on a system and the parameters of real-time scheduling (tasks and messages). The advantage of this integrated approach is mainly the minimization of the resources necessary for meeting the required Quality of Control. This minimization is of great significance for autonomous embedded systems. The conventional approaches based on static scheduling normally leads to a less robust NCSs in unpredictable disturbances. Therefore, it is necessary to develop techniques enabling dynamic adaptation of scheduling parameters with respect to the state of the process. The application chosen to illustrate results of our work was a mini UAV (Unmanned Aerial Vehicle) helicopter. Problem of control and observation of this helicopter had been studied in classical close loop (without network) by the Department of Automatic GIPSA-lab. In this application, the embedded nature adds difficulty to the safety analysis. Indeed, given the low application costs and constraints in terms of weight and board size, hardware redundancy can not be considered. In the context of this project a communication network was added between sensors/actuators and embedded controller. Diagnostic module was co-located with the controller. That means the fault detection system will have direct access to the control input signals from the controller, but needs to get the sensor measurements through the network. Transmitting controller commands and sensor measurements via network needs new strategies for diagnosis and fault-tolerant control.
One task of this project focused on the diagnostic aspect of networked systems. The aim was to consider the possibilities of failure of communication network between the system and its diagnostic algorithm. This topic has been studied less than the control of networked systems, if we except a European project (NeCST : Networked Control Systems Tolerant to fault) in which CRAN participated. Thus, classical approaches of diagnostic must be revised when dealing with NCSs. Here, The network is considered as a component which may be faulty as well as other components (e.g. sensors, actuators). It is obvious that the loss of data and delay will influence the diagnostic algorithms in conventional applications where data regularly sampled and synchronized.

## 1.3 Objectives and motivations

Objective of this thesis is to propose new approaches to design a FDI system with considering network-induced effects such as packet dropout and medium access constraints. By taking into account network effects, mathematical model of the plant is changed and a new strategies of fault detection are needed. These strategies can be adopted from classical theories for FDI in periodic systems. Regarding network access limitation, sometimes it is necessary to provide a pre-defined sequence (i.e. communication sequence) before designing FDI. It describes the instantaneous medium access status of the sensors and actuators. But choice of a sequence of communication is not trivial and that is depend on the structure of the system. In addition, in practice it is not easy to find a precise mathematical model. Therefore, it is necessary to find a strategy to find communication sequences which preserve some structural properties of the plant. Also, algorithms should be applicable on uncertain and large scale systems. Offline scheduling is considered usually as conventional scheduling. But under offline scheduling, performance of diagnostic system may not be guaranteed when the plant is subject to unpredictable disturbances. In addition, online scheduling needs a large computation which may not be possible in case of embedded systems. Thus, a semi-online scheduling which preserve advantages of online scheduling and prevent

some limitation of offline scheduling can be considered as a compromise solution. Finally, obtained results in this thesis are implemented in a UAV helicopter.
The following of this thesis is structured as follows.

## 1.4 Thesis outline and contributions

In **Chapter 2,** an introduction of networked control systems (NCS) will be presented. Then the major problems and recent results of designing a NCSs will be presented. Fault detection and isolation (FDI) is main part of thesis. It is reviewed and an overview of its state of art in NCSs is given in this chapter.

**Chapter 3** is dedicated to problem of design of FDI in NCS with considering medium access constraints. Based on a periodic and offline communication sequence, a strategy of fault detection will studied. Then effects of this communication sequence and an extension of previous results in networked distributed systems are studied and Finally an approach for building a FDI system in case of packet dropout and communication constraints will be proposed.

**Chapter 4,** choosing a offline and periodic communication sequence is not trivial. A system may lose its observability/reachability when its sensors/actuators are connected to controller or FDI system via a network with constraint on the communication. In the other hand, an exact model of the system can not be found in practice and many uncertainties exists in model of the system. In this chapter, a graphical algorithm for finding all communication sequences which preserve the structural properties of the NCS is proposed. Thanks to its simple computation and straightforward implementation, it can be used in large scale systems as well as embedded systems.

**Chapter 5** devoted to study a semi-online scheduling strategy. However, from a computer science point of view, offline scheduling algorithms have many advantages. Essentially, they consume few computing resources and do not induce execution overheads. But they can not guarantee good control and diagnostic performance when an unpredictable disturbance occurs in the system. In the other hand, online scheduling has not these limitation. But, its main drawback is that it requires very important computing resources, which make it usually applicable to slow systems. Proposed scheduling algorithm in this chapter can be seen as a compromise between the advantages of the online scheduling (control performance) and those of the offline scheduling (a very limited usage of computing resources). Finally, An implementation of this scheduling in CAN network based on hybrid priority approach is presented.

In **Chapter 6,** project benchtest, mini helicopter, is presented and effects of network faults in its attitude will be studied. A strategy of active fault tolerant control is proposed to compensate physical and network fault in the helicopter. Efficacy of the proposed strategy depends on the rapidity of diagnostic module. Then an efficient diagnostic module and scheduling mechanism based on results of previous chapters will be proposed.

**Chapter 7** constitutes the conclusion of this dissertation.

## 1.5 Publications

The different chapter of this thesis are based on the following publications.

– Hossein Hashemi nejad, Dominique Sauter, Samir Aberkane, *Actuator fault tolerant control in experimental Networked Embedded mini Drone*, 17th IEEE Mediterranean Conference on Control and Automation, June 2009, Thessaloniki, Greece.(Chapter 6).

– Hossein Hashemi nejad, Dominique Sauter, Samir Aberkane, *Design of structured residual sets in networked control systems subject to random packet dropouts*, 20th International Conference on Systems Engineering, September 2009, Coventry, UK.(Chapter 3).

– Hossein Hashemi nejad, Dominique Sauter, Samir Aberkane, *Fault tolerant control in experimental mini helicopter*, 7th workshop on Advanced Control and Diagnosis, 19-20 November 2009, Zielona Góra, Poland.(Chapter 6).

– Hossein Hashemi nejad, Dominique Sauter, Samir Aberkane, *On-line scheduling and Fault detection in NCS with communication constraints in Drone application*, IEEE conference on Control and Fault-Tolerant Systems, SysTol10, November 2010, Nice , France.(Chapter 5,6).

– Sinuhe Martinez-Martinez, Hossein Hashemi nejad, Dominique Sauter, *Communication sequence design in networked control systems with communication constraints : a graphic approach*, 8th European Workshop on Advanced Control and Diagnosis, 2010, Italy.(Chapter 4).

– Sinuhe Martinez-Martinez, Hossein Hashemi nejad, Taha Boukhobza, Dominique Sauter, *Communication sequence selection to preserve reachability/observability in networked control systems with communication constraints : a graphic approach*, 18th IFAC World Congress, 2011, Italy. (Chapter 4).

– Hossein Hashemi nejad, Dominique Sauter, Samir Aberkane, *FDI design in NCS considering Packet dropouts and limited communication constraints*, 19th IEEE Mediterranean Conference on Control and Automation, June 2011, Greece.(Chapter 3).

# Chapter 2

# Networked Control Systems

First of all, in this chapter a brief history of the use of communication networks as part of a control system will be presented and then different classification of NCSs will be studied. Each network has its own rules for governing the communication between its nodes. Different managing accesses concurrency which have a determinate impact on the guarantee of real-time deterministic communication will be presented in 2.3. When sensor measurements and control data are transmitted over the communication network, many network-induced effects will naturally arise. Research works about these problems will be presented in 2.4. Model-based fault detection and fault detection of NCSs with considering network-induced effects will be presented in the last section of this chapter.

## 2.1 A brief history of NCSs

Computer controlled systems started to emerge in the 1950s. At the beginning stage, the computers were too big, high energy consuming and few reliable. This situation changed when direct digital control (DDC) system was developed. In this configuration, the analog control instrumentation was replaced by one computer. Analog sensors and actuators had *point-to-point connections* with the digital computer. The progress in communication, control and real time computation has enabled the development of large scale distributed control systems (DCS). The nuclear science community developed a parallel bus protocol called CAMAC in the early 70s and MIL-STD-1553 standard has been used in military applications since the mid-1970s [Jor95]. For example, It was used in the U.S. Air Force's $F-16$ and the Army's attack helicopter AH-64A Apache [Zha05]. It used *Time Division Multiplexing* (TDM) to allow data transfers between multiple avionics units over a single communication medium. The first DSC *Honeywell's TDC 2000* was developed in 1974 [AW96]. In the beginning, only monitoring information(e.g. alarms) and on/off signals were transmitted on the serial network. A generic setup of DDC and DCS system were illustrated in Figure 2.1 and Figure 2.2. In the 90s, the development of the microprocessor provides capacity of control of entire production plant. Thanks to ASIC chip design and important price drops in silicon, actuators and sensors could be equipped with network interfaces, and as a consequence become independent nodes on a real-time control network. It gives rise to NCS where nodes need to work together closely to preform control tasks. Many network protocols have been developed by groups of companies and professional or organizations. Controller Area Network (CAN) ([Law97, Gmb91] ), Factory Instrumentation Protocol(FIP) [TLB+91], Foundation fieldbus [PYN06], Process Fieldbus (Profibus), DeviceNet [BV96, Zha10] and Ethernet [Spu00] are some examples of standard network protocols

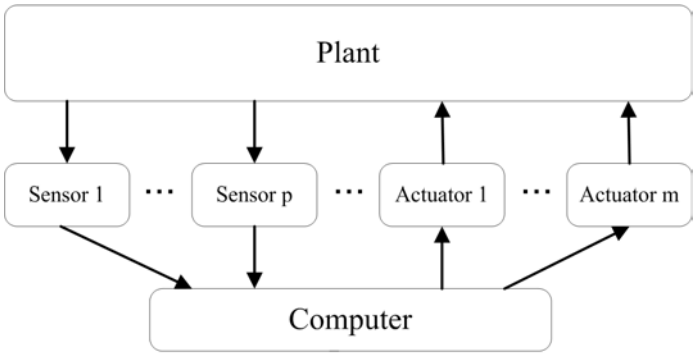



FIGURE 2.1 – Generic setup of DDC system [Zha01]

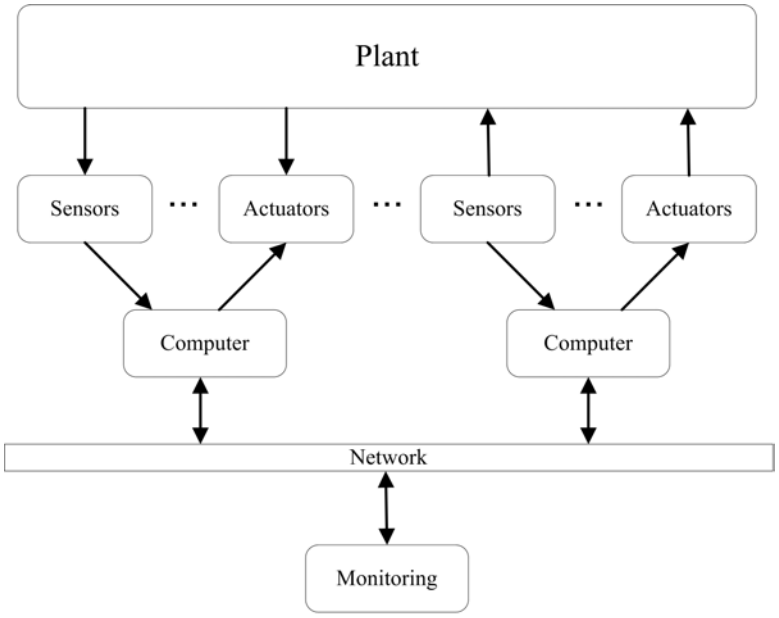



FIGURE 2.2 – A DCS system [Zha01]

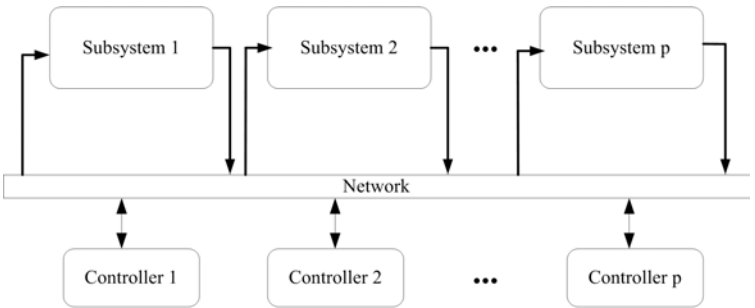



FIGURE 2.3 – shared-network connections

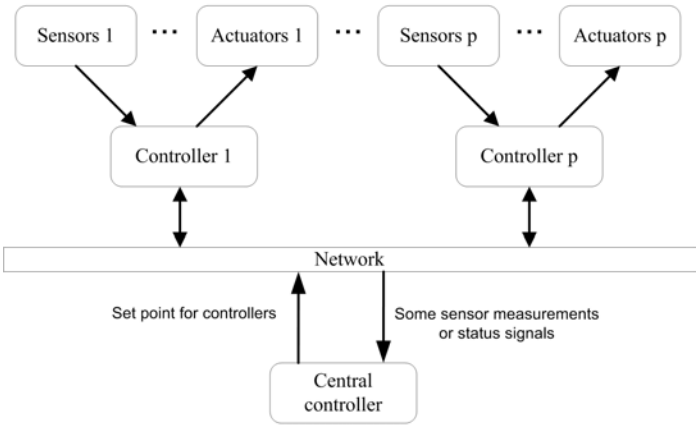



FIGURE 2.4 – Hierarchical structure in NCSs

that were proposed for different industrial applications.
For a more detailed information of these protocols in a control context, refer to [LMT01, Zur05].

## 2.2 NCSs Categories

Generally, control systems that use communication networks can be classified into two major categories [WL10] :

(a) **Shared-network control systems** : In this category, a shared network is used to transfer measurements signals from sensors to controller and monitoring system, and control signals from controller to actuators. Majority of NCSs which having this configuration can be found in automobile and industrial plant applications. As show in Figure 2.3, using network brings more flexibility in installation, troubleshooting and maintenance. In addition, it enables a control loop exchange information with other control loops simply.

(b) **remote control systems** : A communication network can be used for connecting a control system to a controller located far away from it. Remote data acquisition and remote monitoring can be done in this category that sometimes is referred to as *tele-operation* [She92].

Other NCS classification can be done based on the structure of NCS. So, there are two general NCS structure :

(a) **Hierarchical structure** : In this structure design, each subsystem contains its own controller, sensors and actuators as depicted in Figure 2.4. In this configuration, each subsystem's controller receives a set point from cental controller and some sensor measurements or status signal are sent to a central controller via communication network.

(b) **Direct structure** : As shown in Figure 2.5, sensors and actuators are connected directly to communication network. Both direct and hierarchical structure have their own advantages and disadvantages however many NCSs are hybrid of two structure (e.g. [MC05, CR01]).

In addition, It is possible to classify NCSs applications based on sensitivity to time delays as follows :

(a) **Time-critical/time sensitive** : In these applications, time is a critical parameter and long delays can not be tolerated by controlled plant since it can degrade performance of the system.

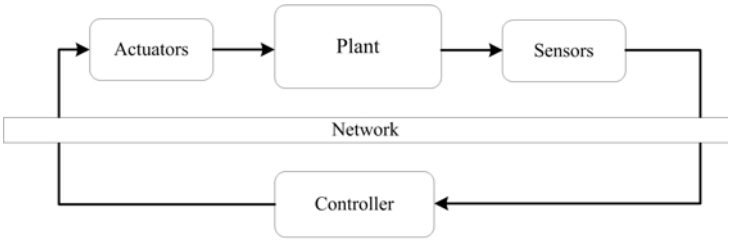



FIGURE 2.5 – Direct structure in NCSs

(b) **Time-delay insensitive** : Can be programs or tasks which run in realtime but whose deadlines are not critical.

## 2.3 Medium access methods in communication networks

Ensuring real-time data exchange between NCSs's components is responsibility of the entire communication stack. Communication mediums usually provide limited number of simultaneous medium access for their users. In modern communication networks, medium access constraints are often resolved via various Medium Access Control (MAC) protocols. They define scheduling and collision arbitration strategies in the network. The choice of a medium access strategy is a crucial step because it determines the cost, complexity and data rate in communication system.
MAC protocols can be divided into two groups : Random MAC protocols and sequential MAC protocols. Under a random MAC protocol, every node is allowed to access to medium whenever it has a packet to transmit. If more than one node want to transmit their packets at the same time, an arbitration strategy is used to resolve the packet collision. In the sequential MAC protocols, each user of network accesses to the medium according to a pre-defined sequence.
ALOHA [Abr70] was the first random MAC protocol that was developed by the university of Hawaii for wireless communication. CSMA/CD (Carrier Sense Multiple Access/Collision Detection) which is used in Ethernet and CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) which is used in IEEE802.11 wireless LAN are examples of this group of MAC protocols. These protocols allow real-time medium access arbitrations and they can be used in a network with burst transmission and relativity large data packet. On the other hand, sequential MAC protocols are well-suited for fixed bit-rate and small packets. They provide low latency and bounded delays. TDMA (Time Division Multiplexing Access) is a example of sequential protocols.
In real-time networks, there are several access protocols. Some important protocols are briefly described as follows :

### 2.3.1 CSMA/CA and CSMA/CD

In the class of CSMA/CA protocols, each message is characterized by a unique priority. Since the shared communication medium can transmit only one message at each instant. Each transmitting node first checks whether the medium is free. If the network is free, then the node can start transmitting. However, it is possible that more than one node detect free communication medium and want to start transmitting at same time. In this situation the transmission of the highest priority message is continued

and other messages with lowest priorities are discarded (Figure 5.12).
In CSMA/CD access protocol, each node that wishes to transmit a message must initially check whether the medium is free. if the medium is free, the node begins the transmission. It can be possible that two nodes start transmitting simultaneously. In this case, they stop message transmitting and then restart transmitting after a random backoff time.

### 2.3.2 Token-Bus

Token bus was standardized by IEEE standard 802.4. It is the basis of many communication protocols such as Profibus, ControlNet [Int99] and MAP [Gro88].
In this protocol, the network nodes construct a virtual ring. The access arbitration is performed by circulation of the token between nodes over the virtual ring. A token is passed around the network nodes and only the node possessing the token may transmit. As a consequence, only one node has the token at each moment and it has access to the medium. By this way, collisions are avoided. If a node that takes possession of the token dose not have any information to transmit, it passes the token to its successor. So, each node must know the address of its neighbor in the ring.
This protocol guarantees an upper bound on the medium access delay and network expansion can be achieved without a significant drop in performance.

### 2.3.3 TDMA

TDMA protocol is the cornerstone of the mobile communication GSM protocol [Gai06]. It allows several users to share the same frequency channel by dividing the signal into different time slots. In this protocol, each node knows exactly the moment of its access to the medium for transmitting. Each node must transmit its information during its *time slot* and as a consequence, collisions are avoided. Usually, time slots are pre-defined, offline and periodic.
This protocol can be implemented in both centralized and distributed structures. Here, bandwidth may be wasted as senders own their time slice, whether they have something to send or not. In centralized structures, A master node is in charge of triggering the communication slaves nodes by transmitting a synchronization signal. But breakdown of master node leads to total network breakdown.

## 2.4 Fundamental issues in NCSs

NCS is a multidisciplinary area closely affiliated with computer networking, signal processing, communication, robotic, information technology and control theory. Data networking technologies are applied widely in industrial and military control applications. These applications include automobile, aeronautic, manufacturing. Some example of networked systems are shown in Figure 2.6, Figure 2.7 and Figure 2.8 The introduction of communication networks in control loops make analysis and synthesis of NCSs complex. There are several important network-induced effects that arise when dealing with an NCS.

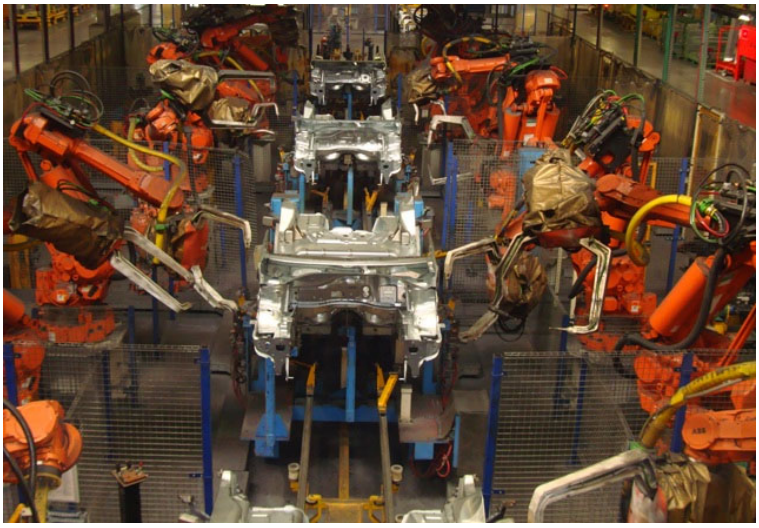


FIGURE 2.6 – Factory automation, PSA Production line

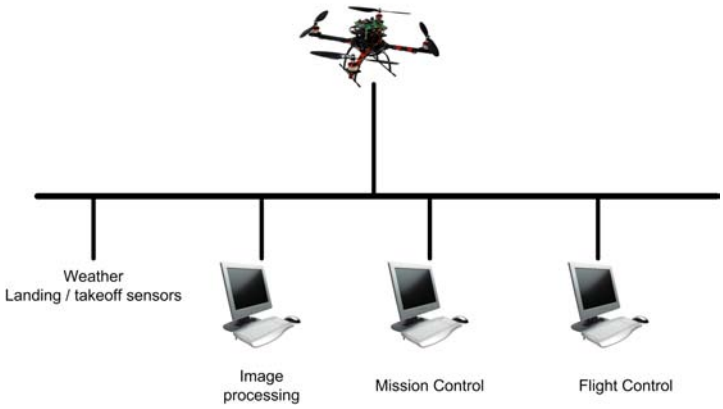



FIGURE 2.7 – Unmanned vehicle navigation

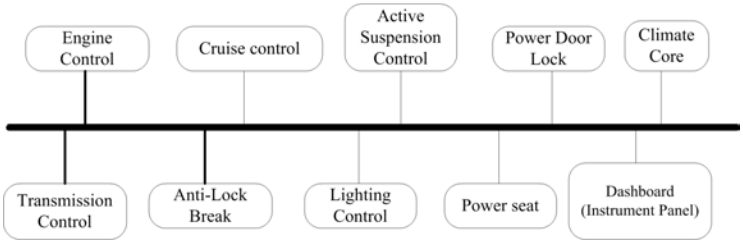



FIGURE 2.8 – Example of network in a automobile

### 2.4.1 Network-induced Delay

One major challenge for NCS design is network-induced delay effects in a control loop. It occurs when the system components exchange data across the network. It can degrades control performance significantly or even destabilize the system.
The delays in a NCS consist of :

- A communication delay between sensors and controllers, $\tau^{sc}$ ;
- A communication delay between controller and actuators, $\tau^{ca}$ ;
- Computational time in controller $\tau^{c}$ witch generally can be included in delay of controller-to-actuator.

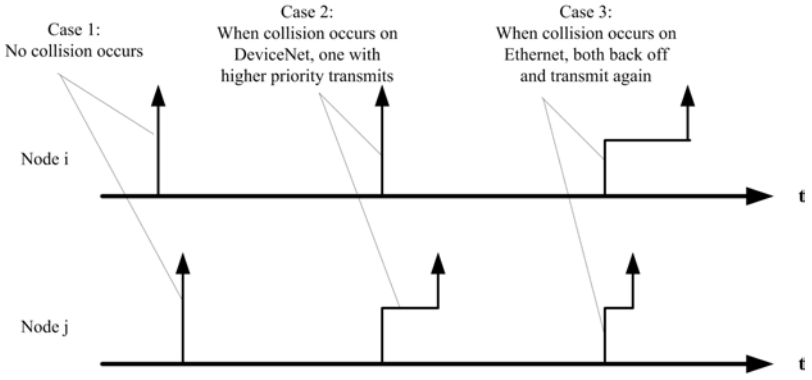



FIGURE 2.9 – Delay on a random access network [Zha01]

Different industrial networks have different communication delay features. They may be constant or bounded. For example delay feature of a token-type field bus is a deterministic bounded delay. Packets on random access networks are affected by random delays, and the worst-case transmission time of a packet is unbounded. These networks are generally considered as non-deterministic networks (e.g. CSMA). Delay in these networks are naturally time-varying or sometimes hard to estimate. For instance, delay in Ethernet is uncertain and stochastic. However, in some networks such as CAN and DeviceNet, network messages are prioritized. So, higher priority message has a better chance for timely transmission. Figure 2.9 showed two type of random access networks in case of the collision.
Stability analysis and controller design of NCSs with time-varying delay have received much attention recently [PYN06, TNT07, YHM06, HN08, NT04].
A simple approach to study stability of a NCS is to find the longest allowable delay for given controller. However, it usually leads to time-consuming procedure. For example, in [KS94, MA04], maximum allowable frequency-domain shift of system's eigenvalues caused by time delay was analyzed. For linear NCS, a strategy for finding Maximum Allowable Transfer Interval (MATI) was proposed in [ZBP01, GW02].
In some works, Lyapunov-Krasovskii (L-K) and Lyapunov-Razumikhin (L-R) functions are used to analyze time-delay independent and time-delay dependent stability of systems with time-varying and invariable bounded delays [BL02, Ric03].
Several methodologies have been studied in the literature to handle the network modeling of NCSs with time delays. By using delayed differential equations, [PT06, PTT08] have modeled NCSs which are effective when network induced delay is bigger than the sampling period. [VGA+09] studied two delay compensation methods, adaptive Smith predictor approach and robust control approach.
In [HWJZ08, SLA09] transmission delays were modeled as a Markov chain, i.e.

$$\lambda_{ij} = P\{\tau(k+1) = j | \tau(k) = i\},$$

where $\tau(k)$ is assumed to take values in finite state space $\{-1,0,\ldots,N\}$, $N$ is a positive integer and stationary transition probability matrix $\Lambda = [\lambda_{ij}]$.

### 2.4.2 Packet dropout

Packet dropout is a typical feature of network communications. It is due to link failure or when packets are purposefully dropped in order to avoid congestion or to guarantee the most recent data to be sent. Although most network protocols are equipped with transmission-retry mechanisms, they can only re-transmit for limited time. After this time has expired, the packets are dropped. Normally feedback controllers can tolerate a certain amount of packet losses. But the consecutive packet losses have an adverse impact on the overall performance.
The simple method to model packet dropout behavior is using the i.i.d. Bernoulli model [Li09]. In this model, a Bernoulli random variable $\alpha_k \in \{0,1\}$ indicates whether the packet at the $k-th$ instance is successfully received or not. In successful receiving case $\alpha_k = 1$, otherwise $\alpha_k = 0$. For any $k$, $\alpha_k$ is i.i.d. distributed with the probability :

$$P\{\alpha_k = 1\} = \lambda$$
$$P\{\alpha_k = 0\} = 1 - \lambda$$

where $\lambda \in [0,1]$. Another method is using Gilbert-Elliott model which considers the network as a discrete-time Markov chain with two possible states : 'good' and 'bad'. In the 'good' state, the packet is successfully received with a higher probability, and in the 'bad' state, the packet is dropped with a higher probability [Wil02]. This model is able to capture the dependence between consecutive packet dropouts, i.e. bursty packet dropping.
Stability analysis of a NCS with packet dropout consideration was studied in many papers such as [HT02, AS03, HY04, WC07, SQ11]. NCS control synthesis witch simultaneously considers network-induced delay and packet dropout was investigated in [YWC04].

### 2.4.3 Medium access constraints

When communication medium can only provide limited number of simultaneous medium access channels for its user. As a consequence, only limited number of sensors and/or actuators are allowed to communicate with the controller at each instant $k$.
In classical control theory a perfect data exchange between system components is assumed. Therefore in NCSs with constraint on communication access, control/FDI design is not only design a classical feedback controller/FDI system. It involves also defining a medium access scheduling strategy. In some works, a Zero order hold (ZOH) at the receiving end (i.e. at the plant's and controller's input stages) was assumed [Hri00]. In this case, if an actuator or a sensor fails to access the medium, the most recently updated values stored in ZOH will be used by the plant or controller. An alternative way is ignoring failed sensor or actuator in controller or plant [ZV06]. In this thesis, this alternative way is considered for sensors or actuators which lose their access to the medium.
One of the simplest scheduling strategy is static scheduling. In this scheduling strategy, sensors and actuators access to the medium based on a predefined sequence, termed *communication sequence* [Bro95]. An LQG design method for NCSs which are subject to medium access constraints was presented in [ZHV05]. The reachability and observability analysis of an NCS with limited communication was studied in [ZV06]. In [ICC06] an algorithm for building the command and communication sequence that

ensure the reachability and observability of a NCS with medium access constraints was taken into account.
For preserving performance of feedback controller or diagnostic module, sometimes a sensor or actuators require immediate attention. So, under static scheduling, NCS may be less robust to unpredictable disturbances. In addition, by using a static scheduling, a global timer is needed to synchronize all sensors and actuators. As a solution to these problems, dynamic medium access scheduling was proposed. By using this scheduling, access to medium is determined *on-line* based on control/FDI needs. The works of [WY01, WYB02] studied a dynamic scheduling. They proposed the MEF-TOD (Maximum Error First, Try once Discard) policy to stabilize a general NCS.

### 2.4.4 Quantization and feedback control in NCS

Quantization has been discussed in the context of digital control and signal processing for over thirty years. An analog signal must be quantized before transmitting via a digital communication medium. A quantizer acts as a functional that maps a real-valued function into a piecewise constant function taking on a finite set of values. Normally, quantizers round off the signals uniformly with a constant step. It can help reduce the size of packets and data rate in communication medium.
It is possible to increase sensitivity of quantizer as the system state approaches to zero [BL00]. It showed that this policy of quantization can stabilize a linear system by a linear time-invariant feedback controller if the system is stabilized by the feedback controller without quantization.
Normally, by increasing control signals's update frequency, disturbance rejection abilities will be improved. Whereas, increasing their quantization precision improves the steady state performance. But in case of limited bandwidth, increasing quantization precision necessities the reduction of update frequency and vice versa. [BGc10] proposed a strategy to choose online the update rate and the quantization precision level.
Some detailed discussions about quantization problem in NCS can be found in [SRAB04, LB04, Sal05, BTX03].
Regardless of the network architecture/medium/protocol the outcome of a NCS is affected to other factors. For details, the interested reader is referred, for instance, to the textbook [IF02] and papers [LW08, RMY09].

## 2.5 Model-based fault detection

Fault free condition in a controlled plant can not be guaranteed and faults may occur in the system anytime. For instance, in a process industry valves and hoses can leak, bearings can jam, sensors can be in error and so on. They may cause the degradation in the performance of the plant which affect the operation cost and quality of final product. For example [ISSC09] showed that industries lose billions of dollars every year due to plant faults. Fault detection and Isolation(FDI) is the primary stage of fault-tolerant control systems. The interested reader is referred to [Ise05, CP99, Din08, Ise11] for some basic terms used in FDI literatures such as faults, failure, disturbances, uncertainties,... . Different approaches for fault detection have been developed. They can be classified into the following categories [Din08] :

- Hardware redundancy based FD
- Plausibility test
- Signal-based FD

– Model-based FD

The model-based approach to fault diagnosis in industrial processes has been receiving remarkable attention since 70s. Model-based fault diagnosis can be defined as detection, isolation and characterization of faults in components of a system from the comparison of the system's available measurements, with a priori information represented by the system mathematical model [CP99].

The basic idea of model based FD system for a process is illustrated by Figure 2.10. The mathematical model of the process is running simultaneously with physical process. It is driven by process's inputs and it estimates process's outputs (without considering disturbances). In fault free condition, the co-called *residual signal* which is the difference between the estimated outputs and measured outputs, should be zero. If there is a fault in the process, the residual will be deviated from zero. The procedure of creating the estimated outputs and building the residual for extracting fault symptoms from the system is called *residual generation*. The process which is used to generate residuals is called *residual generator*. The second step in the procedure of fault detection is *Decision Making*. In this step, a decision rule is applied to determine if any fault is occurred. A decision rule may consist of a simple threshold test on the instantaneous value of residual. This threshold may be a fixed or a time-varying value. No technical

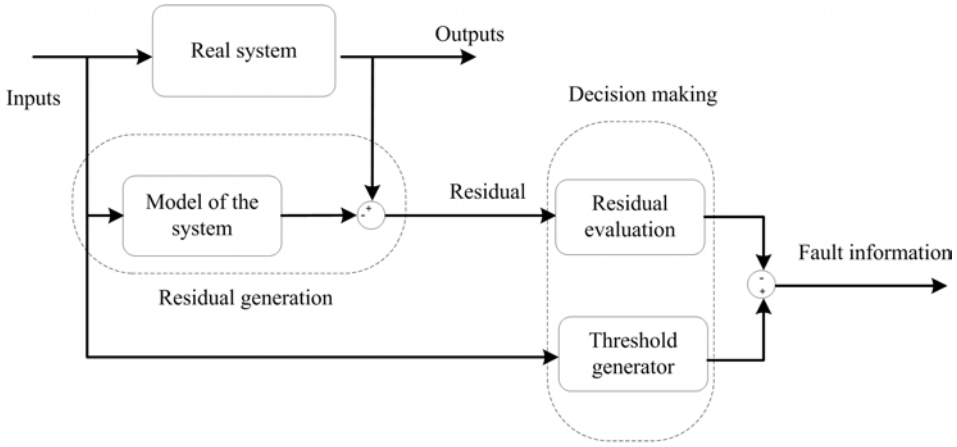



FIGURE 2.10 – The model based fault detection

system can be modeled exactly and there are always uncertainties in mathematical model of the system. As a consequence, the residual dose not carry only fault information. It is also influenced by other parameters. To ensure reliable FD, residual generator should be robust to model uncertainties as well as disturbances and it must provide maximum sensitivity to faults [FDM00]. The generation of residual signals is a central issue in model based FD. A large number of techniques are available for residual generation.

The observer based residual generation can be consider as one of the most important techniques for generating the residual. The observer theories were initially presented to construct the system states with increased robustness against model uncertainties and disturbances in advanced control theory. Uncertain factors in system modeling may be considered as an unknown input. Although the unknown input vector is unknown, its distribution matrix is assumed to be known. Thus, by using unknown input observer (UIO) approach, the unknown input(disturbance) can be decoupled from residual [CPZ96, GF88]. Some techniques studied the problem of residual generation with minimized $H_\infty$ disturbances attenuation (e.g. [KNSS05]). [DF98] proposed a ratio index for optimal design of observer-based residual generator. The residual generators with optimal index $H_\infty/H_\infty$ and $H_/H_\infty$ were studied in [FD94, WYL07]. The parity relation approach was proposed by [LWV86, CW84]. The basic idea of the parity relation method is to provide a proper check of parity of the measurements of the system. For designing an optimal parity

vector, a performance index was proposed. It indicates if a perfect disturbances decoupling is achievable or not.
Although, there are a remarkable number of methods for designing a robust fault detection system by decoupling disturbances and uncertainties but these methods are not optimal in sense of FD. It is possible to distinguish the faults from the disturbances and uncertainties in available residual signals by residual evaluation strategies. As mentioned in [Din08], the efforts for achieving an optimization without considering the evaluation function and the associated threshold computation can result in poor FDI performance. Residual evaluation strategies can be divided into two category of methods :

a) The statistic tests : which are established in statistical methods framework, for instance [ZBB94, Bas88, Bas03].

b) Norm based residual tests : which have been achieved in the norm based framework. A norm based residual evaluation method was studied initially by [NAR88] and in [RN99, DF91, JBN06] residual evaluation problem have been formulated in $H_\infty$ framework.

### 2.5.1 Fault detection of Networked control systems

As discussed previously, in recent years, more and more attention has been paid to the study of stability analysis of networked control systems. However, study on fault detection of NCS has just begun. So far, only a limited number of contributions on FD of NCS can be found. In this section, recent results in fault detection in NCS will be presented.

#### 2.5.1.1 Fault detection in NCSs with network delays

It is clear that the network-induced delay in NCSs could degrade performance of traditional fault diagnosis systems. For instance, [YD04] showed that a residual of a classical observer can not be decoupled from input when network-induced delay is present. As a consequence, the observer can not satisfy essential requirements of the FD system.
Different communication networks have different delay features. For example, Ethernet has an uncertain stochastic delay, whereas, a token bus protocol has a deterministic and bounded delay.
Work concerning FD of NCSs with induced delay can be decomposed into the following groups [FYZ07] :

a) Approaches based on state estimation and observer theory on time-delay system ;

b) Approaches based on low-pass post-filtering ;

c) Approaches based on structure matrix of $\tau_k$(combining $\tau^{sc}$ and $\tau^{ca}$).

[YD04] proposed a residual generator and the low pass filter for fault detection of a NCS with random and short delays. A strategy for designing a FD system based on parity space approach and Stationary Wavelet Transform (SWT) was presented in [YWD04] .
Consider the system model (2.1) with short delays :

$$x(k+1) = \tilde{A}x(k) + \tilde{B}u(k) + g(t) + \tilde{B}_d d(k) + \tilde{B}_f f(k), \tag{2.1}$$

where $x(t)$, $u(t)$, $y(t)$ and $d(t)$ are respectively state vector, the control and output signals, and disturbances. $g(t)$ is a time-varying term. When the total delay $\tau_k$ is random, the variable $g(t)$ is considered as a random disturbance in (2.1). [YRLW06, WYCW06, LCY05] proposed the co-called structure matrix $\tau_k$ to address the fault detection in NCSs by

- decomposing $g(k)$ into known and unknown part. where known part is known information which is extracted from $g(k)$ and unknown part consists unknown information related to $\tau_k$
- use classical robust fault detection methods to achieve robustness to $\tau_k$

[LSA06] assumed that the behavior of delay is random and governed by Makrov chain. But, In all cited works the network induced-delays is assumed to be less than one sampling interval. However, in practice, the delay may be more than one sampling period. [WYW06] studied a FD system with unknown delay which may be greater than sampling period.
We refer the readers to the survey [FYZ07] for more information about the different approaches for designing a FD system of NCS with unknown delays.

#### 2.5.1.2 Fault detection of NCSs with packet losses

Generally speaking, packet dropout can be modeled in deterministic or stochastic sense.[Sei01, SSF+04] assumed that packet losses can be modeled as a bernoulli process. Underlying finite-state Markov chain can be used to model correlated packet dropouts [XHH00, SS03]. Also, stochastic losses can be modeled by the poisson process [Xu06].
[WYD+09] addressed the fault detection for NCSs subject to random packet dropout. It considered that random packet dropout can be occur in both sensor-to-controller and the controller-to actuator links. It studied a observer-based residual generator. In addition, proposed residual generator based on parity space approach was based on a new optimization index to deal with stochastic system parameter.
[SYB09] proposed a fault isolation filter with particular directional properties. Directional residual generations ensure the treatment of multiple faults appearing simultaneously as well as data dropouts. The remaining degrees of freedom in the design of the filter's gain can be used to satisfy $H_\infty$ disturbance attenuation when full isolation cannot be achieved.
Problem of FD in discrete time-varying (LDTV) systems subject to multiple packet dropouts was studied in [LZ09]. FD system was designed based on parity space approach and it assumed that packet dropout is time-stamped. It means that the occurrence of packet data dropouts is on-line available and in order to reduce the online computation, recursive algorithms was introduced.
For more information about recent works of FD in NCS, we refer to [ASY08, FYZ07, HWZ09, ZYC11].

# Chapter 3

# FDI with limited communication : Fixed scheduling case

Model-based fault diagnosis has became an important subject in modern control theory and practice and there are many research works which were dedicated to study this subject (e.g. see [Wil76], [Fra90], [Ger98], [CP99], [ME00],[Ise05] and [Din08]). In classical control systems, information from all sensors is assumed to be instantaneously available for FD system. However in NCSs , no matter what networks are used, several network effects will be introduced into control loop during communication. In this chapter the problem of fault detection and isolation in NCS with communication constraints will be studied. A strategy for extending model of original system to model of NCS with constraints on communication is used. Then, by using suitable periodic communication sequence, the extended system becomes periodic system. Periodic communication sequence is designed separately from FD system and it is not changed during system running. By using lifted technique, periodic system becomes linear time invariant system, or adopting previous approaches concerning to FD design of periodic system FD design will be straightforward. It is possible to consider also other network effects such as packet dropout in design of FD system. Finally, based on used network and information availability about characteristic of the packet dropouts different strategies for designing FDI system will be proposed in 3.5 and 3.6.

## 3.1 Communication limitation modeling

Suppose that the model of the plant, connected to the network with communication constraints shown in Figure 3.1 is described by linear time-invariant(LTI) system (3.1).

$$\begin{array}{rcl} \dot{x_c}(t) & = & A_c x_c(t) + B_c u_c(t) \\ y_c(t) & = & C_c x_c(t) \end{array} \tag{3.1}$$

where $x_c(t) \in \mathbb{R}^n$, $u_c(t) \in \mathbb{R}^m$, $y_c(t) \in \mathbb{R}^p$ represent respectively the state, the command input, and the output of the system. Each of plant's $m$ inputs receives its control signal $u_{c,i}$ from a designed actuator ; each output $y_{c,i}$ is measured by a designed sensor. In figure Figure 3.1, output channels is referred to communication links that enables data transmission from sensors to central station (controller/FDI). And input channels are links between controller and actuators. Here, to focus on the effects of medium access constraints, it is assumed that the transmissions are instantaneous, that the communication channel is

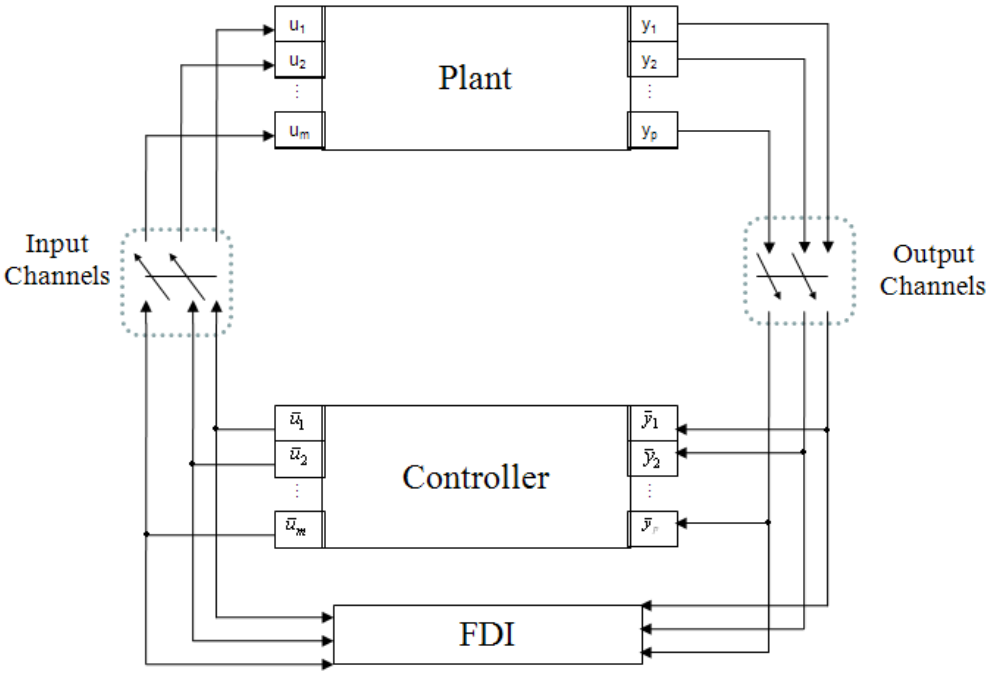



FIGURE 3.1 – NCS with communication constraints

noiseless and that there are not packet dropouts. We will assume, throughout this thesis, that the original system or pairs $(A_c, B_c)$ and $(A_c, C_c)$ are respectively reachable and observable and system sampling period is *nonpathological*. A sampling period is called pathological if it causes the loss of the reachability or observability properties after discretization. [KH63] proved that the set of pathological sampling period can be found and it depends on the eigenvalue of the matrix $A_c$. Therefore, in order to avoid the loss of reachability and observability, it is sufficient to choose a sampling period outside of this set.
Communication constraints exist in both input and output side of the plant. It is assumed that the shared communication medium can simultaneously provide $\omega_\sigma$ outputs channels as well as $\omega_\rho$ input channels where :

$$\begin{aligned} 1 \leq \omega_\rho \leq m \\ 1 \leq \omega_\sigma \leq p \end{aligned} \tag{3.2}$$

Considering this limitation in communication, only $\omega_\sigma$ of $p$ sensors can access to the medium at any one time, while others must wait. Similarly, $\omega_\rho$ from $m$ actuators receive their command from controller at each sampling period. It is important to remark that communication constraints addressed here is lack of access to shared medium for all sensors and actuators simultaneously. This occurs independently of any other communication limitation, such as possible packet losses or transmission delays [ZV06]. For sake of simplicity, both of them are neglected in modeling the plant. However, the presented model will be augmented in next sections to include the effects of packet dropout.
For effective modeling medium access constraints in NCS, it is possible to use a *communication sequence* (see [ZHV05] and [Bro95]), which refer to the order in which the actuators and sensors are allowed to have access to the communication medium. It can be formally described by introducing two binary-valued function $\sigma_i(t) \in \{0,1\}$ and $\rho_i(t) \in \{0,1\}$ for $i-th$ sensor or actuator, respectively.

$$\begin{cases} \sigma_i(t) = 1 & \text{if } y_{c,i} \text{ has access to the network at instant } t \\ \sigma_i(t) = 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

It is similar for input channels :

$$\begin{cases} \rho_i(t) = 1 & \text{if } u_{c,i} \text{ has access to the network at instant } t \\ \rho_i(t) = 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

The limitation that affects the transmission of measures to central station and limitation concerning the sending control commands to the actuators may be modeled by

$$\begin{aligned} \sum_{i=1}^{p} \sigma_i \leq \omega_\sigma \\ \sum_{i=1}^{m} \rho_i \leq \omega_\rho \end{aligned} \tag{3.5}$$

The medium access status for of $m$ actuators and $p$ sensors over time can be represented by the $m-to-\omega_\rho$ and $p-to-\omega_\sigma$ communication sequences.

$$\sigma(t) = \begin{bmatrix} \sigma_1(t) \\ \sigma_2(t) \\ \vdots \\ \sigma_p(t) \end{bmatrix} \quad , \quad \rho(t) = \begin{bmatrix} \rho_1(t) \\ \rho_2(t) \\ \vdots \\ \rho_p(t) \end{bmatrix} \tag{3.6}$$

Hereafter, $\rho$ and $\sigma$ refer to input and output communication sequence, respectively.
Available measurements in controller side of Figure 3.1 can be described as :

$$\bar{y}_{c,i} = \sigma_i \cdot y_{c,i} \ , \ i = 1, \ldots, p. \tag{3.7}$$

Now let $\bar{y}_c$ denotes measurement vector available to controller and FDI module at time $t$. Based on the above communication sequence, it can be found by :

$$\bar{y}_c = M_\sigma(t) \cdot y_c \quad , \quad M_\sigma(t) \stackrel{\Delta}{=} \text{diag}(\sigma_i(t)) \tag{3.8}$$

If one sensor was allowed to access to the medium, its corresponding measurement in controller side is updated at this time and otherwise zero is considered. For designing a FDI module, it is better to ignore the sensor which has not access to the medium. then in this case

$$\bar{y}_c = \bar{M}_\sigma(t) \cdot y_c \tag{3.9}$$

Where $\bar{M}_\sigma$ is obtained by deleting the zero rows of $M_\sigma$.
Similarly, at input side of plant, when actuator $j$ loses its access to the network, the control signal generated by controller for that actuator will be unavailable for that actuator. Instead, the actuator sets $u_{c,j} = 0$ until it obtains medium access again. Let $\bar{u}_c(t)$ denote the control signals generated by controller and $u_c(t)$ denotes the input signal used by actuator. So

$$u_c(t) = M_\rho(t) \cdot \bar{u}_c(t) \quad , \quad M_\rho(t) \stackrel{\Delta}{=} \text{diag}(\rho_i(t)) \tag{3.10}$$

Considering (3.1) , (3.8) , and (3.10), a linear time-varying (LTV) system with $\bar{u}_c$ as its inputs and $\bar{y}_c$ as its outputs can be obtained as follows :

$$\begin{aligned} \dot{x_c}(t) &= A_c x_c(t) + B_c M_\rho(t) \bar{u}_c(t) \\ \bar{y}_c(t) &= M_\sigma(t) C_c x_c(t) \end{aligned} \tag{3.11}$$

equations (3.11) describe the system from FDI/controller 's point of view. It called *extended plant*. It may have the following properties [Zha05] :

a) The extended plant is a LTV with constant matrix $A_c$.

b) It has fewer effective inputs and outputs than original system (3.1).

c) The state of the extended plant (3.11) coincides with that of original system (3.1) with no need for lifting or extensification of the state. And, a fault can be detected in the NCS by designing a FD system for extended plant.

d) As it was illustrated on Figure 3.1, the FDI system is co-located with the controller on a central control and side. So, it has direct access to the system inputs (controller signals) but measurements are received from network.

e) $\rho$ and $\sigma$ can only take on $\binom{m}{\omega_\rho}$ and $\binom{p}{\omega_\sigma}$ possible values, respectively. This make the extended plant (3.11) essentially a switched system [LM99] switching between $\binom{m}{\omega_\rho} \cdot \binom{p}{\omega_\sigma}$ LTI systems.

f) By using a periodic communication sequence with period $T$ (i.e. $\rho(t+T) = \rho(t)$ , $\sigma(t+T) = \sigma(t)$) matrices $M_\sigma(t)$ and $M_\rho(t)$ change periodically and as a consequence, extended plant can be considered as a periodic linear system (PLS).

g) This modeling technique can be easily extended for non-linear systems [MJS09].

Normally, feedback controller and FDI module are implemented in digital computers, hence it is interesting to extend the previous results for discrete-time systems. Suppose therefore, the model of the plant in Figure 3.1 is given by the discrete-time LTI system :

$$\begin{array}{ll} x(k+1) & = Ax(k) + Bu(k) \\ y(k) & = Cx(k) \end{array} \tag{3.12}$$

Similarly to continue systems, it is supposed that data transmission and reception are done via the communication medium with limited capacity. Then the medium access status of the $p$ sensors and $m$ actuators can be represented by the discrete-time $p-to-\omega_\sigma$ and $m-to-\omega_\rho$ communication sequence.

$$\begin{aligned} \sigma(k) &= \begin{bmatrix} \sigma_1(k) & \sigma_2(k) & \dots & \sigma_p(k) \end{bmatrix}^T \\ \rho(k) &= \begin{bmatrix} \rho_1(k) & \rho_2(k) & \dots & \rho_p(k) \end{bmatrix}^T \end{aligned} \tag{3.13}$$

At each sampling instant $k$, $\rho(k)$ is an m-dimensional vector consisting of $\omega_\rho$ ones and $(m-\omega_\rho)$ zeros. Considering communication sequence matrices $M_\rho(k) \triangleq \text{diag}(\rho_i(k))$ and $M_\sigma(k) \triangleq \text{diag}(\sigma_i(k))$ , it is possible to find extended plant of (3.12) by

$$\begin{array}{lll} x(k+1) & = & Ax(k) + BM_\rho(k)\bar{u}(k) \\ \bar{y}(k) & = & M_\sigma(k)Cx(k) \end{array} \tag{3.14}$$

## 3.2 Observability and Reachability of extended plant

Dynamic of the extended plant (3.11) and (3.14) depends on communication policies $\rho$ and $\sigma$. Important properties of system (3.1) and (3.12) , e.g. reachability and observability, may be lost when communication constraints are imposed. So, it is necessary to find the conditions allowing the NCS keep

these properties in case of access limitation. In the other words, under what condition a periodic communication sequence guarantees preservation of observability and reachability of extended plant. Before proceeding with this analysis, some definitions concerning the controllability/reachability and observability are reviewed. They can be found for LTI and LTV systems in most standard linear system theory books (e.g. [Rug96, AW96, Hes09]).

**Definition 3.1.** [AW96] The system (3.12) is controllable if it is possible to find a control sequence such that the origin can be reached from any initial state in finite time.

**Definition 3.2.** [AW96] The system (3.12) is reachable if it is possible to find a control sequence such that an arbitrary state can be reached from any initial state in finite time.

Based on these definitions, controllability dose not imply reachability. However the two concepts are equivalent if $A$ is invertible (here, this assumption was taken).

**Definition 3.3.** [AW96] The system (3.12) is observable if there is a finite $k$ such that knowledge of inputs $u(0),\dots,u(k-1)$ and outputs $y(0),\dots,y(k-1)$ is sufficient to determine the initial state of the system.

In 3.1, it was assumed that original system is observable and reachable, i.e.,

$$\operatorname{rank}\left(\begin{bmatrix} B & AB & \dots & A^{n-1}B \end{bmatrix}\right) = \operatorname{rank}\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right) = n \tag{3.15}$$

The extended plant (3.14) with initial condition $x(0)=0$ evolves from $k=0$ to $k=k_f$. then

$$x(k_f) = R.\begin{bmatrix} \bar{u}(0) & \bar{u}(1) & \dots & \bar{u}(k_f-1) \end{bmatrix}^T$$

where

$$R = \begin{bmatrix} A^{k_f-1}BM_\rho(0) & A^{k_f-1}BM_\rho(1) & \cdots & A^{k_f-1}BM_\rho(k_f-1) \end{bmatrix} \tag{3.16}$$

So the extended plant (3.14) is reachable on $[0,k_f]$ iff $\operatorname{rank}(R)=n$. At each step $k$, $M_\rho(k)$ has effect of selecting $\omega_\rho$ columns from $m$ columns of term $A^{k_f-k-1}B$ on the RHS of (3.16). The matrix $R$ will have full rank if the $k_f\cdot\omega_\rho$ columns that $M_\rho(k)$ selects for $k=0,\dots,k_f-1$ contain $n$ linearly independent columns.

**Definition 3.4.** [MJS09] For given pair of communication sequence $\rho(\cdot)$ and $\sigma(\cdot)$ the extended plant (3.14) is *reachable* on $[k_0,k_f]$, if for any $x_f$, there exists a control signal $\bar{u}(k)$ that steers (3.14) from $x(k_0)=0$ to $x(k_f)=x_f$. (3.14) is $l$-step reachable if it is reachable on $[k,k+l]$ for any $k$, when $l$ is a positive integer.

**Definition 3.5.** [MJS09] For given pair of communication sequence $\rho(\cdot)$ and $\sigma(\cdot)$ the extended plant (3.14) is *observable* on $[k_0,k_f]$, if any initial condition on $k_0$ can uniquely be determined by the corresponding response $\bar{y}(k)$ for $k\in[k_0,k_f]$. It is said that the extended plant (3.14) is $l$-step observable if $l$ is a positive integer and observable on $[k,k+l]$ for any $k$.

If the matrix $A$ is invertible, there always exist a periodic communication sequence that preserves reachability and observability of the original plant. The following theorems were proved in [ZHV05].

**Theorem 3.1.** If A in (3.14) is an invertible matrix and the pair $(A,B)$ is reachable. For any integer $1 \leq \omega_\rho \leq m$, there exists an $m-to-\omega_\rho$ communication sequence $\rho(\cdot)$ and an integer $k_f \leq \left[\frac{n}{\omega_\rho}\right] \cdot n$, such that the extended plant (3.14) is reachable in $[0,k_f]$.

**Theorem 3.2.** If A in (3.14) is an invertible matrix and the pair $(A,B)$ is reachable. For any integer $1 \leq \omega_\rho \leq m$, there exists integers $l$, $T > 0$ and T-periodic $m-to-\omega_\rho$ communication sequence $\rho(\cdot)$ such that the extended plant is *l-step reachable*.

The extended plant (3.14) is observable in $[0,k_f]$ iff

$$\text{Rank}\left(\begin{bmatrix} M_\sigma(0)C \\ M_\sigma(1)CA \\ \vdots \\ M_\sigma(k_f-1)CA^{k_f-1} \end{bmatrix}\right) = n \tag{3.17}$$

At each instant $k$, $M_\sigma$ choose $\omega_\sigma$ rows from $CA^k$ and as it is well known, the observability property of a system is the dual of the reachability property. Therefore, the following theorems can be proven[ZHV05].

**Theorem 3.3.** If A in (3.14) is an invertible matrix and the pair $(A,C)$ is observable. For any integer $1 \leq \omega_\sigma \leq p$, there exists a $p-to-\omega_\sigma$ communication sequence $\sigma(\cdot)$ and an integer $k_f \leq \left[\frac{n}{\omega_\sigma}\right] \cdot n$, such that the extended plant (3.14) is observable in $[0,k_f]$.

**Theorem 3.4.** If A in (3.14) is an invertible matrix and the pair $(A,C)$ is observable. For any integer $1 \leq \omega_\sigma \leq p$, there exists integers $l$, $T > 0$ and T-periodic $p-to-\omega_\sigma$ communication sequence $\sigma(\cdot)$ such that the extended plant is *l-step observable*.

After constructing the periodic communication sequence that preserve observability and reachability of the system, the problem of stabilization and fault detection of becomes straightforward. Under periodic communication, model of extended plant (3.11) and (3.14) becomes a *periodic system* ([Bit86, BC96]). The problem of stabilization and fault detection of periodic systems has been studied extensively in literature. For instance, [BC08] provided a comprehensive view of the theory of periodic systems, analysis models and tools. It studied robust stability problem in this class of systems. Fault detection of periodic systems was studied in [Fad01, LM07, FCN03, ZD08, ZD07a, ZD07b, ZDWZ05]. In [ZV06] the problem of designing a stabilizing feedback controller for a closed-loop NCS was studied. The stabilizing feedback controller consists of state observer and a time varying feedback gain $K$ (Figure 3.2).
The observer that estimates system states for controller can be described by :

$$\hat{x}(k+1) = A\hat{x}(k) + BM_\rho(k)\bar{u}(k) + \mathbf{H}(k)[\bar{y}(k) - M_\rho(k)C\hat{x}(k)] \tag{3.18}$$

and feedback can be computed by

$$\bar{u}(k) = \mathbf{K}(k)\hat{x}(k) \tag{3.19}$$

For given $\alpha > 0$ and $\eta > 0$, the observer gain $\mathbf{H}(k)$ and feedback gain $\mathbf{K}(k)$ are found by :

$$\begin{array}{rcl} \mathbf{K}(k) & = & -\bar{B}^T(k)(A^{-1})^T \mathscr{W}_{\eta\alpha}^{-1}(k,k+l) \\ \mathbf{H}(k) & = & [(A^{-l})^T \mathscr{M}_{\eta\alpha}(k-l+1,k+1)A^{-l}]^{-1} \times (A^{-1})^T \bar{C}(k) \end{array} \tag{3.20}$$

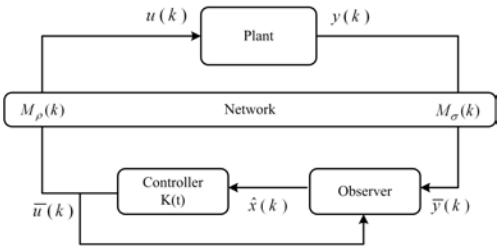



FIGURE 3.2 – Observer-based stabilization of NCS

The close-loop NCS is uniformly exponentially stable with rate $\alpha$, where

$$\begin{aligned} \mathscr{W}_{\eta\alpha}(k_0,k_f) &\triangleq \sum_{j=k_0}^{k_f-1} (\eta\alpha)^{4(k_0-j)} A^{k_0-j-1}\bar{B}(j)(A^{k_0-j-1})^T, \\ \mathscr{M}_{\eta\alpha}(k_0,k_f) &\triangleq \sum_{j=k_0}^{k_f-1} (\eta\alpha)^{4(j-k_f+1)} (A^{j-k_0})^T \bar{C}^T(j)\bar{C}(j)A^{j-k_0} \end{aligned} \tag{3.21}$$

Where $\bar{B}(k) = BM_\rho(k)$ and $\bar{C}(k) = M_\sigma(k)C$.

## 3.3 Fault detection

The Fault detection (FD) system is implemented with controller in central computer(Figure 3.1). For achieving sufficient performance in FD system, it must take into account communication limitation. Let the plant (3.12) with unknown disturbance vector, $d \in \mathbb{R}^{n_d}$ and fault vector, $f \in \mathbb{R}^{n_f}$ which must be detected, is described by :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + E_d d(k) + E_f f(k) \\ y(k) &= Cx(k) + Du(k) + F_d d(k) + F_f f(k) \end{aligned} \tag{3.22}$$

where $E_d$, $E_f$, $f_d$ , and $F_f$ are known and real matrices with appropriate dimensions. Two T-periodic communication sequences are adopted for scheduling input and output channels. Dynamics of the NCS from the FD point of view is characterized by

$$\begin{aligned} x(k+1) &= Ax(k) + BM_\rho(k)\bar{u}(k) + E_d d(k) + E_f f(k) \\ \bar{y}(k) &= \bar{M}_\sigma(k)(Cx(k) + D\bar{u}(k) + F_d d(k) + F_f f(k)) \end{aligned} \tag{3.23}$$

where $\bar{M}_\sigma(k) \in \mathbb{R}^{\omega_\sigma \times p}$ is a T-periodic matrix obtained by selecting $\omega_\sigma$ row of the identity matrix and also $M_\rho(k) \in \mathbb{R}^{m\times m}$ is a T-periodic diagonal matrix. (3.23) under T-periodic communication sequence becomes a T-periodic system. Output of the system (3.23) over a moving finite horizon window $[k-s,k]$ is described by

$$Y(k) = H_{0,k}x(k-s) + H_{u,k}U(k) + H_{d,k}D_s(k) + H_{f,k}F(k) \tag{3.24}$$

where $s$ is an integer which denote the length of the horizon window, and

$$Y(k) = \begin{bmatrix} \bar{y}(k-s) \\ \bar{y}(k-s+1) \\ \vdots \\ \bar{y}(k) \end{bmatrix} \quad , \quad U(k) = \begin{bmatrix} \bar{u}(k-s) \\ \bar{u}(k-s+1) \\ \vdots \\ \bar{u}(k) \end{bmatrix}$$
$$D_s(k) = \begin{bmatrix} d(k-s) \\ d(k-s+1) \\ \vdots \\ d(k) \end{bmatrix} \quad , \quad F(k) = \begin{bmatrix} f(k-s) \\ f(k-s+1) \\ \vdots \\ f(k) \end{bmatrix} \tag{3.25}$$

$$H_{u,k} = \begin{bmatrix} \bar{M}_\sigma(k-s)D & 0 & \dots & 0 \\ \bar{M}_\sigma(k-s+1)CBM_\rho(k-s) & \bar{M}_\sigma(k-s+1)D & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \bar{M}_\sigma(k)CA^{s-1}BM_\rho(k-s) & \dots & \dots & \bar{M}_\sigma(k)D \end{bmatrix} \tag{3.26}$$

$$H_{d,k} = \begin{bmatrix} \bar{M}_\sigma(k-s)F_d & 0 & \dots & 0 \\ \bar{M}_\sigma(k-s+1)CE_d & \bar{M}_\sigma(k-s+1)F_d & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \bar{M}_\sigma(k)CA^{s-1}E_d & \dots & \dots & \bar{M}_\sigma(k)F_d \end{bmatrix} \tag{3.27}$$

$$H_{f,k} = \begin{bmatrix} \bar{M}_\sigma(k-s)F_f & 0 & \dots & 0 \\ \bar{M}_\sigma(k-s+1)CE_f & \bar{M}_\sigma(k-s+1)F_f & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \bar{M}_\sigma(k)CA^{s-1}E_f & \dots & \dots & \bar{M}_\sigma(k)F_f \end{bmatrix} \tag{3.28}$$

$$H_{0,k} = \begin{bmatrix} \bar{M}_\sigma(k-s)C \\ \bar{M}_\sigma(k-s+1)CA \\ \vdots \\ \bar{M}_\sigma(k)CA^s \end{bmatrix} \tag{3.29}$$

Where matrices $H_{0,k}$, $H_{u,k}$, $H_{d,k}$, $H_{f,k}$ are T-periodic with respect to $k$.
Based on parity relation approach of periodic systems (see [ZD08, ZD07a, ZD07b]), in order to satisfy the different requirements on the residual signal, a residual generator can be constructed as follows :

$$r(k) = v_k(Y(k) - H_{u,k}U(k)) \tag{3.30}$$

Where $v_k \in \mathbb{R}^{1\times(s+1)\omega_\sigma}$, called parity vector, is a T-periodic parameter. It must satisfy the condition (3.31) to eliminate the influence of the initial state $x(k-s)$, for any $k$ from the residual.

$$v_k H_{0,k} = 0 \tag{3.31}$$

The residual dynamics is governed by

$$r(k) = v_k(H_{d,k}D_s(k) + H_{f,k}F(k)) \tag{3.32}$$

From (3.32), residual is insensitive to disturbances. if the following rank condition is satisfied :

$$\text{rank}[H_{0,k}\ \ H_{d,k}\ \ H_{f,k}] > \text{rank}[H_{d,k}\ \ H_{f,k}] \tag{3.33}$$

Therefore, perfect decoupling between $r(k)$ and both the disturbances and initial state $x(k-s)$ can be achieved. The parity vector can be found in such way that

$$v_k[H_{0,k}\ \ H_{d,k}] = 0 \quad v_k[H_{f,k}] \neq 0 \tag{3.34}$$

For designing FD, it is sufficient to find values of $H_{0,k}$, $H_{u,k}$, $H_{d,k}$, $H_{f,k}$, and $v_k$ over one period.
In some cases, condition (3.33) cannot be satisfied. In this case, optimal periodic $v_k$ can be found by a suitable compromise between robustness to unknown disturbances and sensitivity to faults. It may be achievable by solving the following optimization problem under constraint (3.31).

$$\min_{v_k} J_k = \min_{v_k} \frac{v_k H_{d,k} H_{d,k}^T v_k^T}{v_k H_{f,k} H_{f,k}^T v_k^T} \tag{3.35}$$

### 3.3.1 Observer-based approach

The residual generation based on periodic observer approach was studied in many papers(e.g. [ZD08]). T-periodic observer for system (3.23) can be constructed as

$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + \bar{B}(k)\bar{u}(k) + L(k)(\bar{y}(k) - \hat{y}(k)), \\ r(k) &= W(k)(\bar{y}(k) - \hat{y}(k)) \\ \hat{y}(k) &= \bar{C}(k)\hat{x}(k) + \bar{D}(k)\bar{u}(k) \end{aligned} \tag{3.36}$$

where

$$\begin{aligned} \bar{B}(k) &= BM_\rho(k) \\ \bar{\xi}(k) &= \bar{M}_\sigma(k)\xi \end{aligned} \tag{3.37}$$

$\xi$ can be replaced in (3.37) by $C$, $D$ , $F_d$ and $F_f$ for finding $\bar{C}$ , $\bar{D}$ , $\bar{F}_d$ and $\bar{F}_f$, respectively.
In (3.36), $L(k)$ and $W(k)$ are T-periodic observer gain matrices and weighting matrices, respectively. According to (3.36) and (3.23), the residual dynamics are given by :

$$\begin{aligned} e(k+1) &= (A - L(k)\bar{C}(k))e(k) + (E_d - L(k)\bar{F}_d(k))\,d(k) + (E_f - L(k)\bar{F}_f(k))\,f(k) \\ r(k) &= W(k)\left(\bar{C}(k)e(k) + \bar{F}_d(k)d(k) + \bar{F}_f(k)f(k)\right) \end{aligned} \tag{3.38}$$

where estimation error $e(k)$ is given by $e(k) = \hat{x}(k) - x(k)$.
To increase robustness of FD to the unknown disturbances without loss of the sensitivity to the faults, the following optimal problems can be used :

$$\sup_{L(k),W(k)} J_{LTP,OBS,\infty/\infty}\big(L(k),W(k)\big) = \sup_{L(k),W(k)} \frac{\sup_{d=0,f\in l_2-\{0\}} \left(\frac{||r||_2}{||f||_2}\right)}{\sup_{f=0,d\in l_2-\{0\}} \left(\frac{||r||_2}{||d||_2}\right)} \tag{3.39}$$

$$\sup_{L(k),W(k)} J_{LTP,OBS,-/\infty}\big(L(k),W(k)\big) = \sup_{L(k),W(k)} \frac{\inf_{d=0,f\in l_2-\{0\}} \left(\frac{||r||_2}{||f||_2}\right)}{\sup_{f=0,d\in l_2-\{0\}} \left(\frac{||r||_2}{||d||_2}\right)} \tag{3.40}$$

The solution of optimization problem (3.39) and (3.40) are derived by solving an equivalent optimization problem for the cyclically lifted LTI system first and then recover the periodic matrices $L(k)$ and $W(k)$ . Solutions of optimization problems (3.39) and (3.40) were given in [ZDWZ05, ZD08].

### 3.3.2 FD subject to access constraints and Packet dropout

Briefly from 3.1, it is supposed that dynamics of the plant after discretization are given by (3.22). The central station gets the system measurements and then transmits the generated control commands to the actuators. Since only one part of actuators and sensors are allowed to communicate with the central station simultaneously, all actuators and sensor measurement values in FD can not be updated at each sampling time $k$. As discussed in 3.1, $'0'$ is fed into an actuator when it fails to access the network. Whereas, if $i-th$ sensor dose not have access to the medium to send its measurements, its value is considered unknown to the central station (3.9).
Furthermore, it is supposed that random packet dropout exists in both input and output channels. It is modeled in the system as a Bernoulli process. i.e.

$$\begin{array}{rcl} u(k) & = & \alpha_k M_p(k)\bar{u}(k) \\ \bar{y}(k) & = & \beta_k \bar{M}_\sigma(k) y(k) \end{array} \tag{3.41}$$

where $\alpha_k \in \{0,1\}$ and $\beta_k \in \{0,1\}$ are two independent Bernoulli processes and they denote the receiving status for actuators and sensors, respectively. $'1'$ means the successful receiving and $'0'$ denotes packet dropout status. these Bernoulli processes satisfy

$$\begin{array}{ll} P\{\alpha_k = 1\} = \alpha, & P\{\alpha_k = 0\} = 1-\alpha \\ P\{\beta_k = 1\} = \beta, & P\{\beta_k = 0\} = 1-\beta \end{array} \tag{3.42}$$

All control commands are sent to actuators by one packet. So, when packet dropout occurs, there are no control commands update at the actuators. It is similar to the case when actuators have not access to the network. As a consequence, $'0'$ is fed to *all* actuators. These functionalities are carried out at plant side and there are no available information concerning this functionalities in the central station. Furthermore, the design of FD system depends strongly on whether the communication protocol supports packet acknowledgement or not. It allows the central station to know whether control commands are received by actuators. Here, it is supposed that network does not send transmission acknowledgement for input commands.
Based on the above analysis, dynamics of the plant from the point of view of the central station can be described as follows :

$$\begin{array}{l} x(k+1) = Ax(k) + BM_p(k)\alpha_k\bar{u}(k) + E_d d(k) + E_f f(k) \\ y(k) = Cx(k) + F_d d(k) + F_f f(k) \\ \bar{y}(k) = \beta_k \bar{M}_\sigma(k) y(k) \end{array} \tag{3.43}$$

Since FD and controller are co-located in central station, packet dropout information of sensor-to-controller link (i.e. $\beta_k$) is available for FD. Whereas, packet dropout information of controller-to-actuator

link (i.e. $\alpha_k$) is not available for FD.
In (3.43), effects of packet dropouts and communication constraints were considered. Therefore, if two T-periodic communication sequences $\sigma(k)$ and $\rho(k)$ are used, (3.43) becomes a periodic system and based on results presented on 3.3.1, a FD system can be constructed.
By considering packet dropout and access constraints, periodic observer equations (3.36) are changed into

$$\begin{aligned}\hat{x}(k+1) &= A\hat{x}(k) + BM_\rho(k)\bar{u}(k) + L(k)\big(\bar{y}(k) - \hat{y}(k)\big)\\ \hat{y}(k) &= \bar{C}(k)\hat{x}(k)\\ r(k) &= W(k)\big(\bar{y}(k) - \hat{y}(k)\big)\end{aligned} \tag{3.44}$$

Observer gain matrix $L(k)$ and weighting matrix $W(k)$ should be designed to guarantee residual sensitivity to faults and robustness to unknown disturbances as well as packet dropout.
The residual dynamics is governed by

$$\begin{aligned}e(k+1) &= \big(A - L(k)\bar{C}(k)\big)e(k) + \big(\bar{E}_d(k) - L(k)\bar{F}_d(k)\big)\bar{d}(k) + \big(E_f - L(k)\bar{F}_f(k)\big)f(k)\\ r(k) &= W(k)\bar{C}(k)e(k) + W(k)\bar{F}_d(k)\bar{d}(k) + W(k)\bar{F}_f(k)f(k)\end{aligned} \tag{3.45}$$

where

$$\begin{aligned}\bar{C}(k) &= \bar{M}_\sigma(k)C \quad , \quad \bar{E}_d(k) = \begin{bmatrix}BM_\rho(k) & 0 & E_d\end{bmatrix}\\ \bar{F}_f(k) &= \bar{M}_\sigma(k)F_f \quad , \quad \bar{F}_d(k) = \begin{bmatrix}0 & \bar{M}_\sigma(k) & \bar{M}_\sigma(k)F_d\end{bmatrix}\end{aligned} \tag{3.46}$$

$$\bar{d}^T(k) = [\Delta\mathfrak{u} \;\; \Delta\mathfrak{y} \;\; d(k)]^T = [(\alpha_k - 1)u(k) \quad (\beta_k - 1)y(k) \quad d(k)]^T \tag{3.47}$$

The optimal $L(k)$ and $W(k)$ in the sense of

$$\min_{W(k),L(k)} \frac{\sup_{f=0,\bar{d}\in l_2-\{0\}} \frac{||r|_2}{||\bar{d}||_2}}{\sup_{\bar{d}=0,f\in l_2-\{0\}} \frac{||r|_2}{||\bar{d}||_2}} \tag{3.48}$$

are given by

$$L_{opt}(k) = -L_{0,k}^T \quad W_{opt}(k) = W_{0,k} \tag{3.49}$$

where $W_{0,k}$ is the left inverse of full-column rank matrix $\mathscr{H}_k$ satisfying

$$\mathscr{H}_k\mathscr{H}_k^T = \bar{C}(k)X_k\bar{C}^T(k) + \bar{F}_d(k)\bar{F}_d^T(k) \tag{3.50}$$

and $(X_k, L_{0,k})$ is the stabilization solution to the difference periodic Riccati system(DPRS)

$$\begin{bmatrix} AX_kA^T - X_{k+1} + \bar{E}_d(k)\bar{E}_d^T(k) & AX_k\bar{C}^T(k) + \bar{E}_d(k)\bar{F}_d^T(k) \\ \bar{C}(k)X_kA^T + \bar{F}_d(k)\bar{E}_d^T(k) & \bar{C}(k)X_k\bar{C}^T(k) + \bar{F}_d(k)\bar{F}_d^T(k)\end{bmatrix} \times \begin{bmatrix} I \\ L_{0,k}\end{bmatrix} = 0 \tag{3.51}$$

This approach and False alarm rate(FAR) computation for the FD scheme (3.44) was studied in [WYDW09].

## 3.4 FD in distributed system with communication constraints

Here, we considered that plant (3.1) is a distributed process with unknown disturbances and fault vectors. The distributed process can be decomposed into $p$ subsystems and each of them is equipped

with a local controller, a number of sensors and actuators. It is described by

$$\begin{aligned} \dot{x_c}(t) &= A_c x_c(t) + B_c u_c(t) + E_{c,d} d_c(t) + E_{c,f} f_c(t) \\ u_c(t) &= \left[u_{c,1}(t), \ldots, u_{c,p}(t)\right]^T \\ B_c &= \left[B_{c,1}, \ldots, B_{c,p}\right] \end{aligned} \tag{3.52}$$

where $u_{c,i}(t) \in \mathbb{R}^{k_{ui}}, i = 1, \ldots, p$ is control input of $i-th$ subsystem.
In addition, the output of the subsystems are updated at different sampling rate $T_{s,i}$ and they can be found by

$$T_{s,i} = l_i T_0 \tag{3.53}$$

where $l_i$ is a positive integer. After discretization (3.52) with sampling time $T_0$, it is described by

$$x(k_0+1) = A_d x(k_0) + B_d u_d(k_0) + E_{dd} d_d(k_0) + E_{df} f_d(k_0) \tag{3.54}$$

where

$$\begin{aligned} &A_d = e^{A_c T_0} \quad , \quad B_d = \left[B_{d,1}, \cdots, B_{d,p}\right] \quad , \quad B_{d,i} = \int_0^{T_0} A_c^{\tau} B_{c,i} d\tau \\ &E_{dd} = \int_0^{T_0} A_c^{\tau} E_{c,d} d\tau \qquad , \qquad E_{df} = \int_0^{T_0} A_c^{\tau} E_{c,f} d\tau \end{aligned} \tag{3.55}$$

It is supposed that between two actuators updates, the actuators keep last received value. i.e.

$$u_{d,i}(k_0) = \begin{cases} u_{c,i}(k_0 T_0) & \text{if } k_0/l_i \text{ is integer} \\ u_{c,i}\big((k_0-1)T_0\big) & \text{Otherwise} \end{cases} \tag{3.56}$$

Due to the different sampling times of subsystems, the process outputs are constructed by

$$y_i(k_i) = C_{d,i} x(k_i) + D_{d,i} u_{d,i}(k_i) + F_{dd,i} d_d(k_i) + F_{df,i} f_d(k_i) \tag{3.57}$$

where $k_i$ denoted the local sampling time instant for $i-th$ subsystem and $F_{dd,i}$ and $F_{df,i}$ are known matrices with appropriate dimensions. Let

$$T = l_{\max} T_0 \ , \ l_{\max} = \max\{l_1, \cdots, l_p\}.$$

Then by using the lifting technique [CF96] the following system with sampling time $T$ for the process is obtained

$$\begin{aligned} \mathtt{x}(k+1) &= \mathscr{A}\mathtt{x}(k) + \mathscr{B}\mathtt{u}(k) + \mathscr{E}_d \mathtt{d}(k) + \mathscr{E}_f \mathtt{f}(k) \\ \mathtt{y}(k) &= \mathscr{C}\mathtt{x}(k) + \mathscr{D}\mathtt{u}(k) + \mathscr{F}_d \mathtt{d}(k) + \mathscr{F}_f \mathtt{f}(k) \end{aligned} \tag{3.58}$$

where

$$\mathtt{u}(k)=\begin{bmatrix}\tilde{\mathtt{u}}_1(k)\\ \vdots\\ \tilde{\mathtt{u}}_p(k)\end{bmatrix}\quad,\quad \tilde{\mathtt{u}}_i(k)=\begin{bmatrix}u_{d,i}\big(kT\big)\\ u_{d,i}\big(kT+l_iT_0\big)\\ \vdots\\ u_{d,i}\big(kT+(\alpha_i-1)l_iT_0\big)\end{bmatrix}$$

$$\mathtt{y}(k)=\begin{bmatrix}\tilde{y}_1(k)\\ \vdots\\ \tilde{y}_p(k)\end{bmatrix}\quad,\quad \tilde{y}_i(k)=\begin{bmatrix}y_i(kT)\\ y_i\big(kT+l_iT_0\big)\\ \vdots\\ y_i\big(kT+(\alpha_i-1)l_iT_0\big)\end{bmatrix} \tag{3.59}$$

$$\alpha_i=\begin{cases}\dfrac{l_{max}}{l_i} & \text{if } \dfrac{l_{max}}{l_i}=\text{integer}\\ \\ \text{int}\big(\dfrac{l_{max}}{l_i}\big) & \text{Otherwise}\end{cases}\qquad i=1,\cdots,p$$

State space matrices in (3.58) are found by

$$\mathscr{A}=A_d^{l_{\max}}=e^{A_cl_{\max}}\quad,\quad \mathscr{B}=\big[\mathscr{B}_1,\cdots,\mathscr{B}_p\big],$$

$$\mathscr{B}_i=\begin{bmatrix}A_d^{l_{\max}-l_i}\sum_{j=0}^{l_i-1}A_d^jB_{d,i} & A_d^{l_{\max}-2l_i}\sum_{j=0}^{l_i-1}A_d^jB_{d,i} & \cdots & \sum_{j=0}^{l_i-1}A_d^jB_{d,i}\end{bmatrix},$$

$$\mathtt{d}(k)=\begin{bmatrix}d_d(kT)\\ \vdots\\ d_d\big(kT+(l_{\max}-1)T_0\big)\end{bmatrix}\quad,\quad \mathtt{f}(k)=\begin{bmatrix}f_d(kT)\\ \vdots\\ f_d\big(kT+(l_{\max}-1)T_0\big)\end{bmatrix} \tag{3.60}$$

$$\mathscr{E}_d(k)=\begin{bmatrix}A_d^{l_{\max}-1}E_{dd} & \cdots & A_dE_{dd} & E_{dd}\end{bmatrix}\quad,\quad \mathscr{E}_f(k)=\begin{bmatrix}A_d^{l_{\max}-1}E_{df} & \cdots & A_dE_{df} & E_{df}\end{bmatrix}$$

And in its output equation

$$\mathscr{C}=\begin{bmatrix}\mathscr{C}_1\\ \vdots\\ \mathscr{C}_p\end{bmatrix}\quad,\quad \mathscr{C}_i=\begin{bmatrix}C_{d,i}\\ C_{d,i}A_d^{l_i}\\ \vdots\\ C_{d,i}A_d^{(\alpha_i-1)l_i}\end{bmatrix}$$

$$\mathscr{D}\triangleq\text{diag}(\mathscr{D}_i)\quad,\quad \mathscr{D}_i=\begin{bmatrix}D_{d,i} & 0 & \cdots & 0\\ C_{d,i}\mathscr{B}_i & D_{d,i} & \cdots & 0\\ \vdots & \ddots & \ddots & \vdots\\ C_{d,i}A_d^{(\alpha_i-2)l_i}\mathscr{B}_i & \cdots & C_d\mathscr{B}_i & D_{d,i}\end{bmatrix}$$

$$\mathscr{F}_d = \begin{bmatrix} \mathscr{F}_{d,1} \\ \vdots \\ \mathscr{F}_{d,p} \end{bmatrix} \quad , \quad \mathscr{F}_{d,i} = \begin{bmatrix} F_{dd,i} & 0 & \cdots & 0 \\ C_{d,i}\bar{E}_{d,i} & F_{dd,i} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ C_{d,i}A_d^{(\alpha_i-2)l_i}\bar{E}_{d,i} & \cdots & C_d\bar{E}_{d,i} & F_{dd,i} \end{bmatrix}$$

$$\mathscr{F}_f = \begin{bmatrix} \mathscr{F}_{f,1} \\ \vdots \\ \mathscr{F}_{f,p} \end{bmatrix} \quad , \quad \mathscr{F}_{f,i} = \begin{bmatrix} F_{df,i} & 0 & \cdots & 0 \\ C_{d,i}\bar{E}_{f,i} & F_{df,i} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ C_{d,i}A_d^{(\alpha_i-2)l_i}\bar{E}_{f,i} & \cdots & C_d\bar{E}_{f,i} & F_{df,i} \end{bmatrix}$$

$$\bar{E}_{d,i} = \begin{bmatrix} A_d^{l_i-1}E_{dd} & \cdots & A_dE_{dd} & E_{dd} \end{bmatrix} \quad , \quad \bar{E}_{f,i} = \begin{bmatrix} A_d^{l_i-1}E_{df} & \cdots & A_dE_{df} & E_{df} \end{bmatrix} \tag{3.61}$$

In case of the perfect communication, $\bar{m}$ measurements are sent via the network in the time interval $[kT,(k+1)T]$. It is supposed that, because of limitation on access to the network, only $\omega_m$ measurements must be selected to sent to FD system. $\bar{m}$ and $\omega_m$ are found by

$$\bar{m} = \sum_{i=1}^{p} m_i\alpha_i \quad , \quad \omega_m = \sum_{i=1}^{p} m_{qi}\alpha_i \tag{3.62}$$

where $m_i$ is number of measurement signals of $i-th$ subsystem and $m_{qi}$ is number of measurements which can send when access to the network is limited. Let $\bar{\mathtt{y}}(k)$ denotes available measurements for FD at time instant $k$. The output equation in (3.58) will be reformulated as

$$\begin{array}{rcl} \bar{\mathtt{y}}(k) & = & \mathscr{M}_q\big(\mathscr{C}\mathtt{x}(k)+\mathscr{D}\mathtt{u}(k)+\mathscr{F}_d\mathtt{d}(k)+\mathscr{F}_f\mathtt{f}(k)\big) \\ & = & \bar{\mathscr{C}}\mathtt{x}(k)+\bar{\mathscr{D}}\mathtt{u}(k)+\bar{\mathscr{F}}_d\mathtt{d}(k)+\bar{\mathscr{F}}_f\mathtt{f}(k) \end{array} \tag{3.63}$$

where

$$\mathscr{M}_q = \begin{bmatrix} \mathscr{M}_{q,1} & 0 & \cdots & 0 \\ 0 & \mathscr{M}_{q,2} & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathscr{M}_{q,p} \end{bmatrix} \quad , \quad \mathscr{M}_{q,i} = \begin{bmatrix} Q_{i,1} \\ Q_{i,2} \\ \vdots \\ Q_{i,\alpha_i} \end{bmatrix} \quad Q_{i,z} \in \mathbb{R}^{m_{qi}\times m_i}, z = 1,\ldots,\alpha_i \tag{3.64}$$

Entries $e_{lj}$ of block $Q_{i,z}$ for $i-th$ subsystem is a binary function that is defined by

$$e_{lj} = \begin{cases} e_{lj} = 1 & \text{if } y_{ij} \text{ has access to the medium} \\ e_{lj} = 0 & \text{Otherwise} \end{cases}$$

Matrix $\mathscr{M}_q$ is a constant matrix and during a period $[kT,(k+1)T]$, it indicates which sensor has access to the network at each instant $k$. As a consequence, the system (3.63) becomes a LTI system and it is possible to design a fault detection system based on FD approaches for this kind of systems that was studied in many papers. According to (3.58) and (3.63), parity relation for the outputs of the system can be written

$$\mathscr{Y}_s(k) = \mathscr{H}_0x(k-s)+\mathscr{H}_u\mathscr{U}_s(k)+\mathscr{H}_d\mathscr{D}_s(k)+\mathscr{H}_f\mathscr{F}_s(k) \tag{3.65}$$

where

$$\mathscr{Y}_s(k) = \begin{bmatrix} \bar{\mathrm{y}}(k-s) \\ \bar{\mathrm{y}}(k-s+1) \\ \vdots \\ \bar{\mathrm{y}}(k-s+1) \end{bmatrix} \quad , \quad \mathscr{U}_s(k) = \begin{bmatrix} \mathrm{u}(k-s) \\ \mathrm{u}(k-s+1) \\ \vdots \\ \mathrm{u}(k-s+1) \end{bmatrix} \tag{3.66}$$

and other matrices can be found simply by using parity space approach for LTI systems. The residuals can be constructed

$$r(k) = V_s\big(\mathscr{Y}_s(k) - \mathscr{H}_u\mathscr{U}_s(k)\big) = V_s\big(\mathscr{H}_d\mathscr{D}_s(k) + \mathscr{H}_f\mathscr{F}_s(k)\big) \tag{3.67}$$

where $V_s \in \mathbb{R}^{\gamma\times(s+1)\omega_m}$ is parity matrix.
In the context of a distributed system which measurements are sent via a network, performance of fault detection can not be guaranteed by considering only FD system. But, also it strongly depend on Quality of Service(QoS) of the used network. Minimizing the number of data transmission has important effect on network load, transmission delay or even packet dropout. [PDGC09] proposed an algorithm for minimization of the number of measurement transmission. By using this algorithm, $\mathscr{M}_q$ can be calculated. It ensures the desired fault detection performance and optimum trade-off between the fault detection performance and the total communication cost in the system described by (3.58) and (3.63).

## 3.5 Design of structured residual sets in NCSs subject to random packet dropouts

In this section, the problem of FDI design with considering random packet dropouts is studied. Then, in next section, a strategy to design a FDI system with considering random packet dropouts and medium access constraints will be proposed.
Here, a networked control system where the controller sends control signals to actuators via a packet dropping network, as illustrated by Figure 3.3, is supposed. Mathematical model of the plant, which may be subject to faults, is described by :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Ff(k) \end{aligned} \tag{3.68}$$

In practice, control commands and system measurements which are sent via network links in NCS may be lost or corrupted by noise during transmission. Packet dropout may be because of network failures or when packets are purposefully dropped in order to avoid congestion or to guarantee the most recent to be sent. Although a single packet loss neither deteriorate the system performance nor destabilizes the system, the consecutive packet losses have an adverse impact on the overall performance.
It is assumed that a UDP-like protocol is used which means that controller does not receive any acknowledgments for successful delivery of control packets. In the other words, there is no feedback in controller from actuator side that shows information was received completely or not. It is assumed that under intermitted communication, when the packet containing the control signal $u_i$ is dropped, the $'0'$ is fed to $i-th$ actuator. Comparing to the system configuration described in 3.3.2, packet dropout occurs in one input commands and there is no possibility to packet dropout in all input channels simultaneously. Here, the solution is proposed based on this assumption but it can be extended for the case when command of more than one actuator is lost.

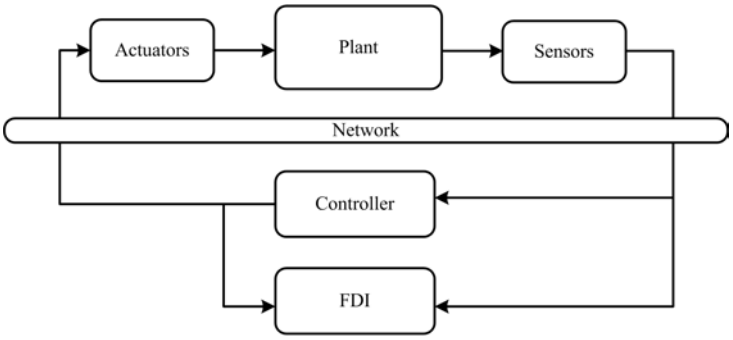



FIGURE 3.3 – feedback control system with random packet dropouts

For packet dropout modeling, introduce $\tilde{u}(k) = \begin{bmatrix}\tilde{u}_1 & \tilde{u}_2 & \dots & \tilde{u}_m\end{bmatrix}^T$ with $\tilde{u}_i(k) = \delta_i(k)u_i(k)$,

$$\begin{cases} \delta_i = -1 & \text{when } u_i \text{ is dropped} \\ \delta_i = 0 & \text{Otherwise} \end{cases}$$

Then model of the system with unreliable data transmission can be expressed as :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B\tilde{u}(k) \\ y(k) &= Cx(k) + Ff(k) \end{aligned} \tag{3.69}$$

For data acquisition, it is supposed that the sensor is time-driven with a constant sampling time $h$.
Here, it is considered that all actuators work in fault-free condition and there is no possibility of fault occurrence in more than one sensor simultaneously. In case of packet dropout in one input, system equation can be expressed as :

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + b_i\tilde{u}_i(k) \\ y^j(k) = C^j x(k) + f^j(k) \\ y_j(k) = c_j x(k) + f_j(k) \end{cases} \text{ for } \begin{matrix} j = 1,2,\dots,p \\ i = 1,2,\dots,m \end{matrix} \tag{3.70}$$

Where $c_j \in \mathbb{R}^{1\times n}$ is $j-th$ row of matrix $C$, $C^j \in \mathbb{R}^{(p-1)\times n}$ is obtained from matrix $C$ by deleting its $j-th$ row $c_j$ , $y_j$ is the $j-th$ component of $y$ and $y^j \in \mathbb{R}^{p-1}$ is obtained from vector $y$ by deleting the $j-th$ component $y_j$. By considering, packet dropout occurrence in one input, matrix of unknown input, $E_i$, will be :

$$E_i = b_i \quad \text{for } i = 1,2,\dots,m$$

$b_i$ is the $i-th$ column of matrix $B$.
Based on the description, $v = p \times m$ UIO-based residual generator can be constructed as follows :

$$\begin{cases} w(k+1) = F^{ij}w(k) + T^{ij}Bu(k) + K^{ij}y(k) \\ \hat{x} = w(k) + H^{ij}y(k) \end{cases} \text{ for } \begin{matrix} j = 1,2,\dots,p \\ i = 1,2,\dots,m \end{matrix} \tag{3.71}$$

The residual signals can be written as :

$$r^{ij}(k) = \left(I - C^j H^{ij}\right)y^j(k) - C^j w(k) \tag{3.72}$$

Where $\hat{x}$ is the estimated state vector and $w$ is the state of these full observers. $F, T, K$ and $H$ are matrices to be designed for achieving unknown input decoupling requirement.

That is to say, the state estimation error vector $(e = \hat{x} - x)$ of the observers goes to zero asymptotically, regardless of packet dropout in one input. By applying the theory of UIOs [CP99, TNP02], the observers is achieved by solving the following equations :

$$\big(H^{ij}C^j - I)\big)E_i = 0 \tag{3.73}$$
$$T^{ij} = I - H^{ij}C^j \tag{3.74}$$
$$F^{ij} = A - H^{ij}C^jA - K_1^{ij}C^j \tag{3.75}$$
$$K_2^{ij} = F^{ij}H^{ij} \tag{3.76}$$
$$K^{ij} = K_1^{ij} + K_2^{ij} \tag{3.77}$$

and $F^{ij}$ must be a stable matrices.
The system defined by (3.71) is a bank of unknown input observers for the system defined by (3.70) if the necessary and sufficient conditions are established :

1. rank $(C^jE_i)$ = rank $(E_i)$
2. $(C^j, A_1^{ij})$ are detectable pair, where $A_1^{ij} = F^{ij} + K_1^{ij}C^j$

If these conditions are fulfilled, UIOs provide the estimation of state vector, used to generate a residual vector $r^{ij}(k)$ independent of corresponding sensor fault and packet dropout in one input of the system. This means that $r^{ij}(k)$ is insensitive to fault in $j-th$ sensor even if there is packet dropout in $i-th$ input of system.

### 3.5.1 Fault isolation

For the fault isolation, a bank of $v = p \times m$ unknown input observer is established, each residual vector $r_h = r^{ij}, \quad h = 1, \ldots, v$, produced by the $h-th$ UIO, may be used to detect a fault according to statistical test. Consequently, it involves the use of statistical tests such as the Page Hinkley test, limit checking test, generalized likehood ration test, and trend analysis test.
An output vector of the statistical test, called the *Coherence vector* $S[r_h(k)]$, can then be built form the bank of $p \times m$ residual generators.

$$S\big(r(k)\big) = \Big[S\big(||r_1(k)||\big) \cdots S\big(||r_{p\times m}(k)||\big)\Big]^T, \tag{3.78}$$

where $S\big(||r_h(k)||\big)$ represents a mathematical indicator associated to norm of the residual vector $r_h$. It is equal to zero in the fault free case and set to one when a fault is detected.
The coherence vector is then compared to the fault signature vector $S_{ref,f_h}$ associated to the $h-th$ fault according to the residual generator built to produce a signal sensitive to all faults except one as represented in Table 3.1.
Decision-making is then realized according to an elementary logic test that can be described as follows : an indicator $I(F_h)$ is equal to '1' if $S(r)$ is equal to $h-th$ column of incidence matrix $S_{ref,f_h}$ and is equal to zero otherwise. The element associated with the indicator equal to one is then declared to be faulty.

| $S(r)$ | Fault Free | $S_{ref,f_1}$ | $S_{ref,f_2}$ | ... | $S_{ref,f_v}$ | Packet dropout in unknown input |
|---|---|---|---|---|---|---|
| $S(\|r_1\|)$ | **0** | **0** | **1** | ... | **1** | **1** |
| $S(\|r_2\|)$ | **0** | **1** | **0** | ... | **1** | **1** |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋮ |
| $S(\|r_v\|)$ | **0** | **1** | **1** | ... | **0** | **1** |

TABLE 3.1 – Inference Matrix

### 3.5.2 Illustrative example

Consider a system with the following state space matrices :

$$A = \begin{bmatrix} -3.6 & 0 & 0 & 0 \\ 0 & -3.6 & 0 & 0.07 \\ 0 & 0 & -36 & 0.25 \\ 0 & 0.63 & 0.77 & -0.141 \end{bmatrix}$$
$$B = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{3.79}$$

By using $h = 0.01s$ as sampling time of the system, it is possible to build $m \times p = 9$ unknown input observers and each observer is insensitive to packet dropout in one input of system as well as fault in one sensor. A table of sensor fault and packet dropout sensitivity of residuals for each observer is illustrated in Table 3.2 (e.g. residuals of $OBS3_2$ are insensitive to fault of second sensor and packet dropout in third input of the system).

In the fault scenario, it is supposed that there is the possibility of packet dropout in the first input of system and a bias fault occurs in second sensor at 2s. Generated residuals by 9 UIOs without packet dropout in the system are shown in Figure 3.5 and residuals in case of packet dropout in first input of the system are illustrated in Figure 3.6.

As it is shown in Figure 3.5 after sensor fault occurrence, residuals of observers that are not insensitive to fault in second sensor deviate from zero. The three observers $OBS1_1, OBS1_2, OBS1_3$ on the Figure 3.6 are insensitive to packet dropout in the first input so residuals of these observers before fault occurrence go to zero asymptotically whereas residuals for other observers, insensitive to packet dropout in second input and third input, are non zero. After fault occurrence, residual in $OBS1_1$ and $OBS1_3$ change to a non-zero value and residual of $OBS1_2$ that is insensitive to fault in second sensor remains zero.

| | | Packet dropout | | | Sensor Fault | | |
|---|---|---|---|---|---|---|---|
| | | U1 | U2 | U3 | S1 | S2 | S3 |
| Observer | OBS 1_1 | **0** | **1** | **1** | **0** | **1** | **1** |
| | OBS 1_2 | **0** | **1** | **1** | **1** | **0** | **1** |
| | OBS 1_3 | **0** | **1** | **1** | **1** | **1** | **0** |
| | OBS 2_1 | **1** | **0** | **1** | **0** | **1** | **1** |
| | OBS 2_2 | **1** | **0** | **1** | **1** | **0** | **1** |
| | OBS 2_3 | **1** | **0** | **1** | **1** | **1** | **0** |
| | OBS 3_1 | **1** | **1** | **0** | **0** | **1** | **1** |
| | OBS 3_2 | **1** | **1** | **0** | **1** | **0** | **1** |
| | OBS 3_3 | **1** | **1** | **0** | **1** | **1** | **0** |

TABLE 3.2 – Sensor fault and packet dropout sensitivity

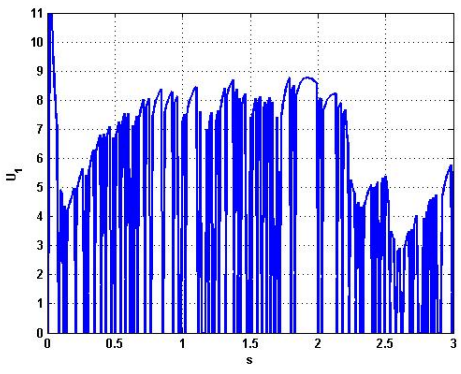



FIGURE 3.4 – Packet dropout in $u_1$

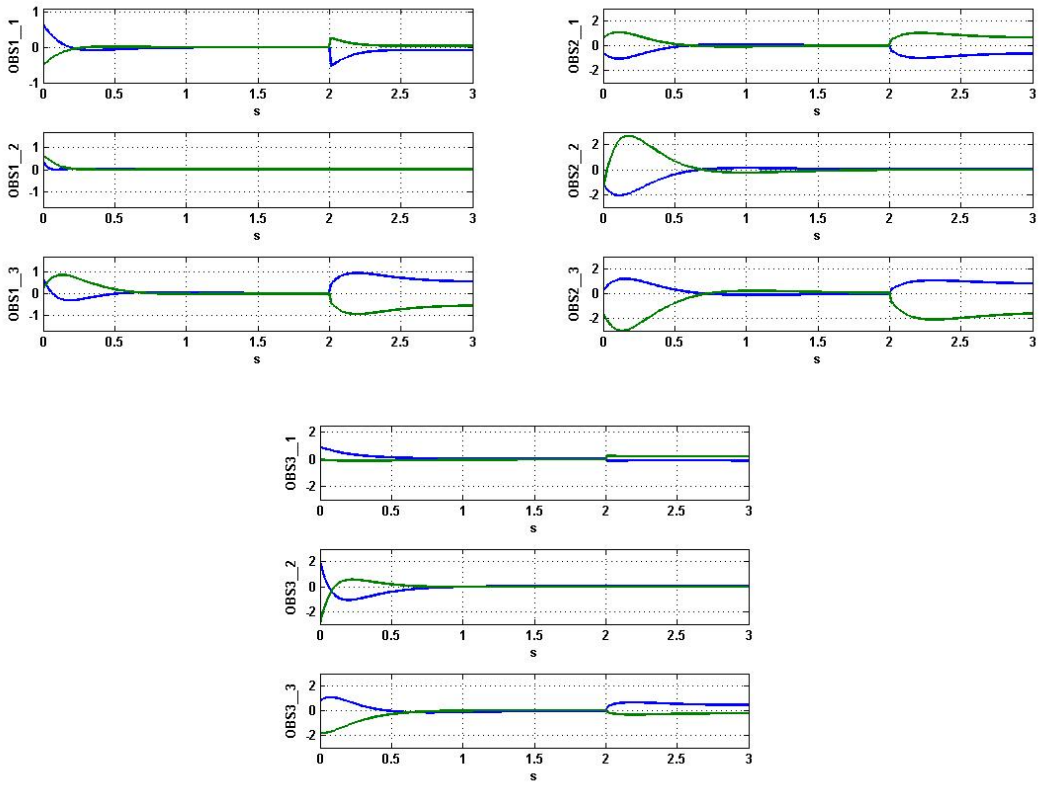



FIGURE 3.5 – Residuals of observers without packet dropout

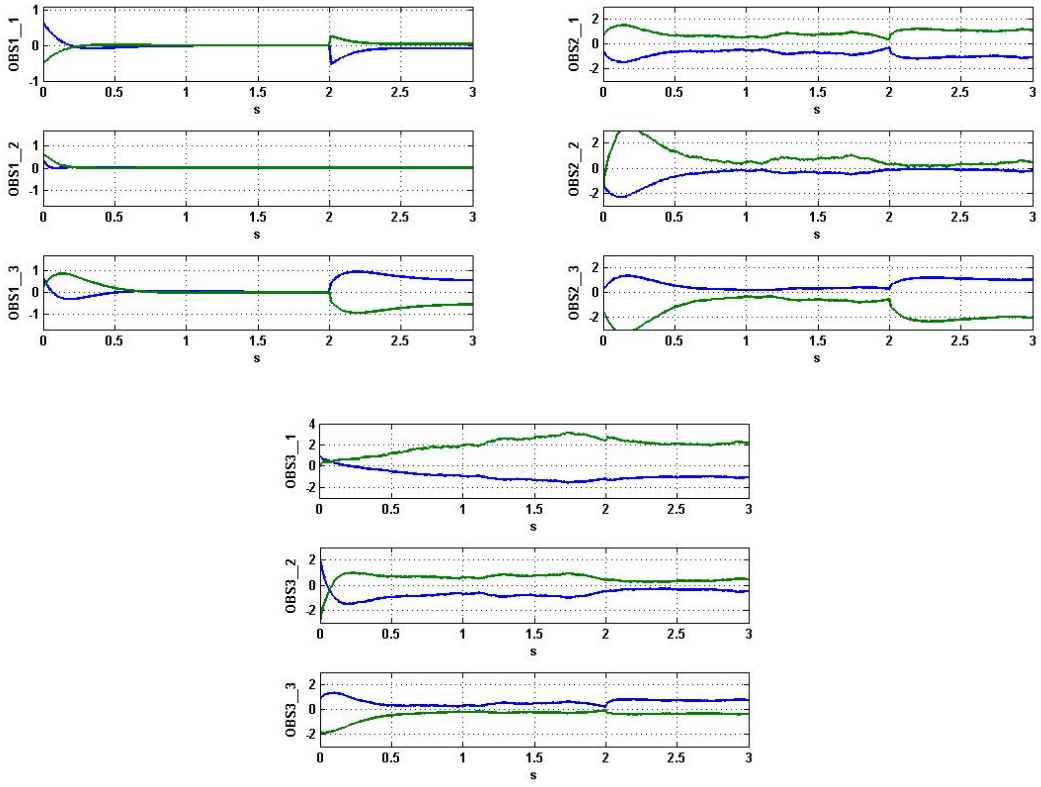



FIGURE 3.6 – Residuals of observers with packet dropout

## 3.6 FDI of NCSs subject to random packet dropout and medium access constraints

Suppose that the controlled plant through the digital communication network, and which may be subject to faults, is describe as :

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k)+Ed(k) \\ y(k) &= Cx(k)+Ff(k) \end{aligned} \tag{3.80}$$

It is assumed that the shared communication network can simultaneously provides $\omega_\sigma$ ($1 \leq \omega_\sigma \leq p$) output channels. Similar to the problem studied in 3.5, it is further assumed that the controller sends control signals to actuators via packet dropping channels as illustrated by Figure 3.7. Thus, variables $\tilde{u}(k) = [\tilde{u}_1(k), \tilde{u}_2(k), \cdots, \tilde{u}_m(k)]$ and $\delta_i(k)$ are introduced as

$$\tilde{u}_i(k) = \delta_i(k)u_i(k) \qquad \text{for any } i \in \{1, \cdots, m\} \tag{3.81}$$

It is considered that for commands sent by controller there is no concept of acknowledgment, retransmission moreover packet dropout probability is unavailable. In addition, it is assumed that packet dropout occurs in one input commands. These assumptions allow us to relax the some assumptions which were considered in the previous works (see 3.3.2).
Binary-value of $\delta_i$ can be defined at each instant $k$ as follow

$$\begin{cases} \delta_i = -1 & \text{when } u_i \text{ is dropped} \\ \delta_i = 0 & \text{Otherwise} \end{cases} \tag{3.82}$$

Then it is possible to find model of the system for FDI module with considering unreliable data transmission and constraints on communication in output channels by :

$$\begin{cases} x(k+1) &= Ax(k)+Bu(k)+B\tilde{u}(k)+Ed(k) \\ \bar{y}(k) &= \bar{C}(k)x(k)+\bar{F}(k)f(k) \end{cases}, \quad \begin{aligned} \bar{C}(k) &= \bar{M}_\sigma(k)C \\ \bar{F}(k) &= \bar{M}_\sigma(k)F \end{aligned} \tag{3.83}$$

In what follows, based on the extended plant model (3.83), a strategy for fault detection and isolation will be presented.

### 3.6.1 Design of robust residual generators

Here, it is assumed that all actuators work in fault free condition. For obtaining a set of structured residuals, robust to packet dropout in one actuator, equation (3.83) is reformulated as follows :

$$\begin{cases} x(k+1) &= Ax(k)+Bu(k)+E_i[\tilde{u}_i(k) \quad d(k)]^T \\ y^j(k) &= \bar{C}^j(k)x(k)+\bar{F}^j(k)f^j(k) \end{cases}, \quad \begin{aligned} &i = 1,2,\ldots,m \\ &j = 1,2,\ldots,p \end{aligned} \tag{3.84}$$

$$\bar{E}_i = [b_i \quad E]$$

Where $\bar{C}^j(k) \in \mathbb{R}^{(\omega_\sigma - 1)\times n}$ is obtained from the matrix $C$ by removing its $j-th$ row then applying communication policy $\sigma(k)$ and $b_i$ is the $i-th$ column of the matrix $B$.
The proposed method for FDI design is based on parity relation of periodic systems that was studied in

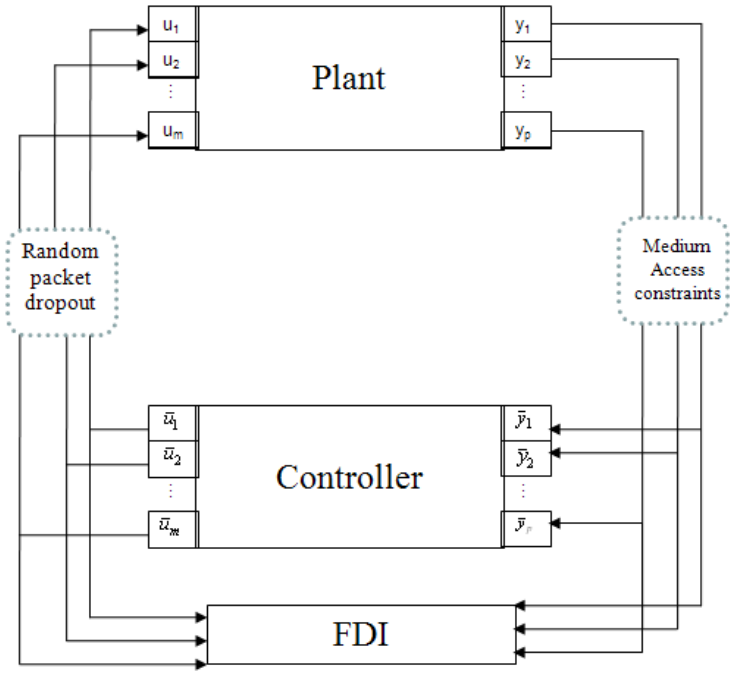



FIGURE 3.7 – Feedback control system

previous sections.
Consider $s$ as an integer denoting the length of a moving time window and input-output parity relation of (3.84) over a horizon $[k-s,k]$ expressed by :

$$Y^j(k) = H^j_{0,k}x(k-s) + H^j_{u,k}U(k) + H^{i,j}_{d,k}D_i(k) + H^j_{f,k}\tilde{F}^j(k) \tag{3.85}$$

where

$$Y^j(k) = \begin{bmatrix} y^j(k-s) \\ y^j(k-s+1) \\ \vdots \\ y^j(k) \end{bmatrix} \quad , \quad U(k) = \begin{bmatrix} u(k-s) \\ u(k-s+1) \\ \vdots \\ u(k) \end{bmatrix} \tag{3.86}$$

$$D_i(k) = \begin{bmatrix} [\,\tilde{u}_i(k-s) & d(k-s)\,]^T \\ [\,\tilde{u}_i(k-s+1) & d(k-s+1)\,]^T \\ \vdots & \\ [\,\tilde{u}_i(k) & d(k)\,]^T \end{bmatrix} \quad , \quad \tilde{F}^j(k) = \begin{bmatrix} f^j(k-s) \\ f^j(k-s+1) \\ \vdots \\ f^j(k) \end{bmatrix} \tag{3.87}$$

Matrices $H^j_{0,k}$, $H^j_{u,k}$, $H^{i,j}_{d,k}$ and $H^j_{f,k}$ are T-periodic matrices, and are computed as follows :

$$H^j_{0,k} = \begin{bmatrix} \bar{C}^j(k-s) \\ \bar{C}^j(k-s+1)A \\ \vdots \\ \bar{C}^j(k)A^s \end{bmatrix} \tag{3.88}$$

$$H_{u,k}^{j}=\begin{bmatrix}0 & 0 & \dots & 0\\ \bar{C}^{j}(k-s+1)B & 0 & \ddots & 0\\ \vdots & \ddots & \ddots & 0\\ \bar{C}^{j}(k)A^{s-1}B & \dots & \dots & 0\end{bmatrix} \tag{3.89}$$

$$H_{d,k}^{i,j}=\begin{bmatrix}0 & 0 & \dots & 0\\ \bar{C}^{j}(k-s+1)\bar{E}_{i} & 0 & \ddots & \vdots\\ \vdots & \ddots & \ddots & 0\\ \bar{C}^{j}(k)A^{s-1}\bar{E}_{i} & \cdots & \cdots & 0\end{bmatrix} \tag{3.90}$$

$$H_{f,k}^{j}=\begin{bmatrix}\bar{F}^{j}(k-s) & 0 & \dots & 0\\ 0 & \bar{F}^{j}(k-s+1) & \dots & 0\\ \ddots & \ddots & \ddots & 0\\ 0 & \dots & \dots & \bar{F}^{j}(k)\end{bmatrix} \tag{3.91}$$

Thus, the idea is to generate a set of $\nu = p \times m$ residual signals of the form :

$$r_{ij}(k)=v_{ij,k}^{T}\left(Y^{j}(k)-H_{u,k}^{j}U(k)\right) \tag{3.92}$$

Where the parity vector $v_{ij,k}$ is a component vector of parity space $P$ defined as follows :

$$P=\left\{v_{ij,k}|v_{ij,k}\begin{bmatrix}H_{0,k}^{j} & H_{d,k}^{i,j}\end{bmatrix}=0\right\} \tag{3.93}$$

Since the purpose of the residual generator is to detect a fault, the following equation must be satisfied

$$v_{ij,k}H_{f,k}^{j}\neq 0 \tag{3.94}$$

Based on condition (3.33) for any $k$, the residual signal can be decoupled from the unknown disturbances and effect of packet dropout in one input if the following condition is satisfied

$$\operatorname{rank}\begin{bmatrix}H_{0,k}^{j} & H_{d,k}^{i,j} & H_{f,k}^{j}\end{bmatrix}>\operatorname{rank}\begin{bmatrix}H_{0,k}^{j} & H_{d,k}^{i,j}\end{bmatrix} \tag{3.95}$$

When full decoupling can not be achieved (condition (3.95) is not satisfied) the parity vector is found by solving an optimization problem of the form (3.35).
Using this scheme, we can see that the dynamics of $r_{ij}(k)$ are not influenced by the initial state $x(k-s)$ nor by disturbance vector $d(t)$. Furthermore, it is independent of fault in $j-th$ sensor and packet dropout in $i-th$ input of system. In the other words, $r_{ij}(k)$ is not sensitive to fault in $j-th$ sensor and packet dropout in $i-th$ command of the system.
For fault isolation, a set of $\nu = p \times m$ structured residuals is established and for each residual a mathematical indicator $S\left(\left\|r_{ij}\right\|\right)$ is defined as :

$$\begin{cases}\left\|r_{ij}\right\|<J_{t,k}\Rightarrow S\left(\left\|r_{ij}\right\|\right)=0\\ \left\|r_{ij}\right\|\geq J_{t,k}\Rightarrow S\left(\left\|r_{ij}\right\|\right)=1\end{cases} \tag{3.96}$$

Then, it is possible to construct a table for fault and packet dropout sensitivity of each residual (Table 3.3).

In fact, in the fault free condition and without packet dropout in inputs, $S(||r_{ij}||)$ for any $i\in\{1,\dots,m\}$ and $j\in\{1,\dots,p\}$ are equal to zero. While, if there exist a random packet dropout in $i-th$ input and fault in $j-th$ sensor of the system, $S(||r_{wx}||)=1$ for any $w\neq i$ , $x\neq j$.

| | | Packet dropout | | | | Sensor faults | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $u_1$ | $u_2$ | ... | $u_m$ | $s_1$ | $s_2$ | ... | $s_p$ |
| Residuals | $S(\|r_{11}\|)$ | **0** | 1 | ... | 1 | **0** | 1 | ... | 1 |
| | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| | $S(\|r_{m1}\|)$ | 1 | 1 | ... | **0** | **0** | 1 | ... | 1 |
| | $S(\|r_{m2}\|)$ | 1 | 1 | ... | **0** | 1 | **0** | ... | 1 |
| | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| | $S(\|r_{mp}\|)$ | 1 | 1 | ... | **0** | 1 | 1 | ... | **0** |

TABLE 3.3 – Sensor and packet dropout sensitivity

### 3.6.2 Illustrative example

Consider a LTI system with the following stat-space representation :

$$A = \begin{bmatrix} 0.22 & 0.25 & 0.1 & -0.1 & 0 & 0.2 \\ 0.1 & 0.1 & 0.5 & 0.01 & -0.42 & 0 \\ 0.5 & -0.2 & 0.2 & 0.25 & 0.22 & 0.27 \\ 0.1 & 0 & 0.25 & 0.1 & 0.07 & 0.001 \\ 0.1 & 0 & 0.1 & 0 & 0.1 & 0.025 \\ 0.25 & 0.33 & 0.33 & 0.51 & 0.63 & 0.03 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 0.19 & 1 \\ 5.67 & 4 & -2 \\ 1.136 & -3.146 & 0.4 \\ 1.136 & -0.9 & 5 \\ 1 & 0.1 & 1 \\ 0.216 & -0.6 & 0.3 \end{bmatrix} \qquad C = I_6 \tag{3.97}$$

A sampling time of $h = 0.01\ s$ is used. There is a random packet dropout in one actuator. Also, because of constraint on medium access, FDI module has access to three measurement at each moment($\omega_\sigma = 3$). A communication sequence with period $T = 0.5\ s$ was chosen. It assigned 70% of medium access time to $y_1$, $y_2$ and $y_3$ while rest of access time was given to transmit measurements of $y_4$, $y_5$ and $y_6$ as follows :

$$\sigma(k) = \begin{cases} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} & 0 \leq t \leq .7T \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} & .7T \leq t \leq .T \end{cases} \tag{3.98}$$

For simulation, it is considered that there is the possibility of 30 percent information loss in third input and FDI module has no information about characteristic, probability or moment of packet losses. A bias fault was considered in second sensor. A set of $p \times m = 18$ residual generators based on parity relation (3.85) is constructed. Each residual is insensitive to packet dropout in one input of the system as well as fault in one sensor. For example, if there is packet dropout in third input, residuals $r_{3j}$ $(j = 1, \ldots, 6)$

remain zero when sensors work in fault free condition.
First simulation was carried out when there was no packet dropout in input channels and sensors worked in fault free condition. After fault occurrence at $t = 1.8s$ in second sensor, residuals that were insensitive to this fault (i.e. $r_{32}$, $r_{22}$ and $r_{12}$) remained zero whereas other residuals were changed to non-zero value. Outputs of some residual generators were shown in Figure 3.8 and Figure 3.9. Note that horizontal axis is time in millisecond

FDI module can detect a sensor fault after updating by new measurement. For instance, the sensor

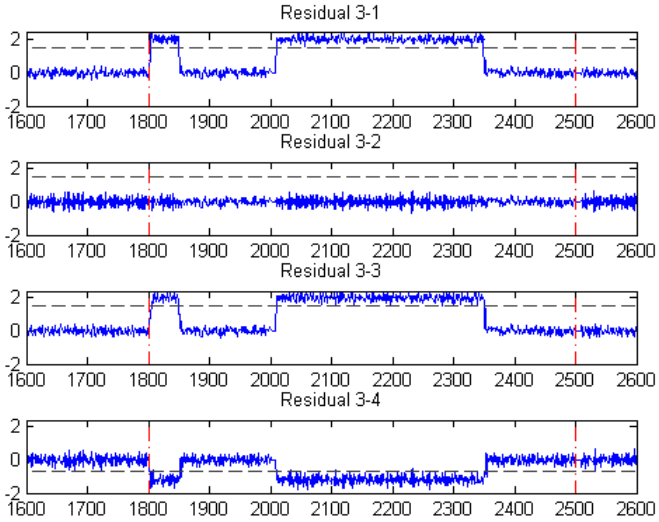



FIGURE 3.8 – Residuals evolution without packet dropout

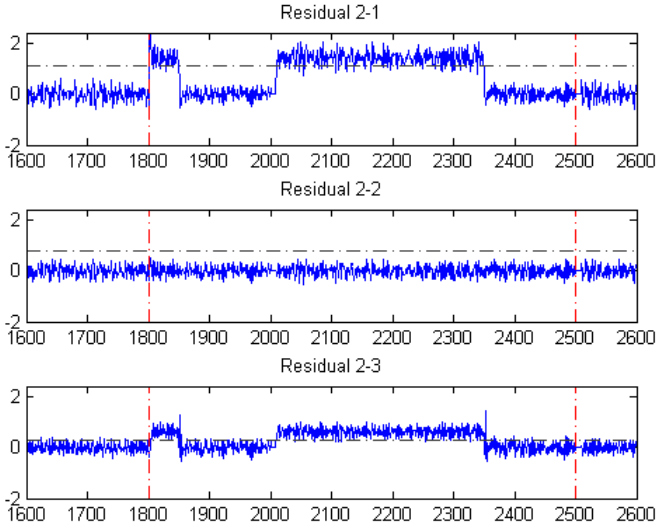



FIGURE 3.9 – Residuals evolution without packet dropout

fault in second sensor can not be detected in $[1.85, 2)$ $s$. It is because of lose of access to the medium

during this time. While, after giving access to second sensor, value of residuals sensitive to fault in this sensor diverted from zero(Figure 3.10). To make a decision, a enough number of data is necessary. By using this communication sequence in the equivalent condition, a hypothesis concerning to faulty case in first three sensors ($y_1$,$y_2$,$y_3$) is accepted faster than others sensors. Therefore it is necessary to choose a communication sequence with a suitable updating rate for each sensor to guarantee global performance in diagnostic module. Other solution for this problem is using dynamic scheduling and allow to access in larger time interval for sensor or subsystem that its fault probability is larger. Dynamic scheduling and fault detection will be studied in chapter five.

The residuals in case of packet dropout in third input of the system were illustrated in Figure 3.11 and

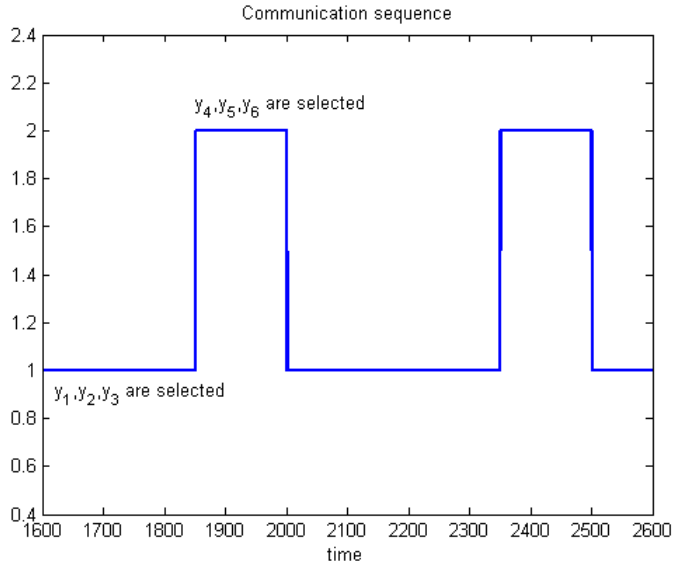



FIGURE 3.10 – Periodic communication sequence

Figure 3.12. As it was shown, value of residuals which are sensitive to packet dropout in third input became non zero. Then, during the fault occurrence ($[1.8, 2.5]$ $s$) in second sensor all residuals except $r_{32}$ changed to non-zero value.

## 3.7 Conclusion

Problem of fault detection and isolation of a linear networked control system (NCS) was studied in this chapter. The considered network can provide limited number of channels for transmitting commands and measurements. A strategy for extending model of the original system to model of NCS with constraints on communication was adopted. By using a periodic communication sequence, extended model became a periodic system. Then, classical approaches for residual generation of periodic system was used. In addition, it is considered that packets containing controller commands and sensors measurements may be dropped and there are no available information (e.g. moment, probability) about packet dropout in central station. Then a strategy to develop a set of structured residuals was proposed. It guarantees robustness to packet dropout and unknown disturbances. Each residual is insensitive to packet dropout of one input also to fault in one sensor. Therefore it is possible to detect and isolate sensor fault in the system even if there are a unknown and random packet dropout in one input of system. However,

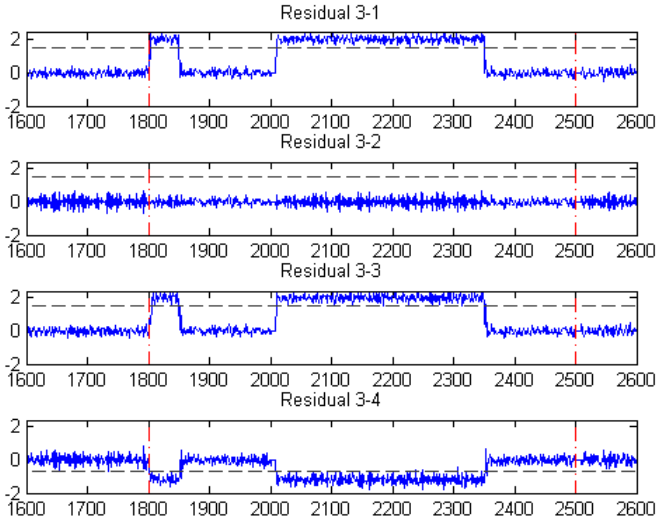



FIGURE 3.11 – Residuals evolution with random packet dropout

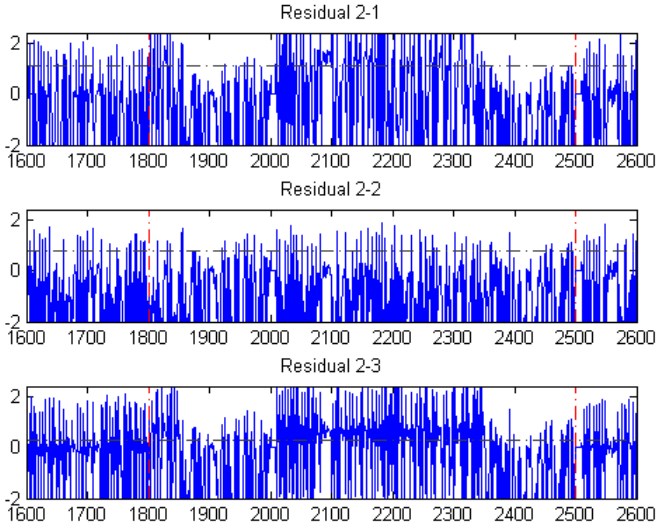



FIGURE 3.12 – Residuals evolution with random packet dropout

some assumptions limit the range of application of this method in practice.
In this chapter, residual generator of NCS with fixed and periodic communication sequence was studied. The residual generation was designed based on parity space-based approach and observer-based approach. However, other model-based approaches such as filter can be used to design residual generator. Communication constraints parameters, $\omega_\rho$ and $\omega_\sigma$, were considered as constant values. An interesting extension is the generalization of this problem to the networks with variable resources, which may be modeled by the time varying parameters $\omega_\rho$ and $\omega_\sigma$.

# Chapter 4

# Communication sequence design in NCS with communication constraints : a graphic approach

In the previous chapter the problem of FDI design with communication constraints was studied and different FDI design strategies based on fixed and periodic communication sequence were presented. The choice of a communication sequence is not obvious and required a specific knowledge of the system. In order to understand these ideas the following examples are analyzed in which reachability/observability of the system may be lost depending on the sequence selection.

**Example 4.1.** The following system is originally observable(when there is no communication constraints).

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad , \quad C = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{4.1}$$

It is easy to verify that the rank of the observability matrix drops when the system loses any of its outputs. Now, let us suppose, for example, a communication constraint fixed to only one channel ($\omega_\sigma = 1$). The

observability matrix (3.17) will have the form

$$\mathscr{O} = \begin{bmatrix} M_\sigma(0)\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \\ M_\sigma(1)\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} A \\ \vdots \\ M_\sigma(k_f-1)\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} A^{k_f-1} \end{bmatrix} \tag{4.2}$$

One may think intuitively a 3-period sequence that guaranties access all outputs to the medium during $[0, k_f - 1]$ may preserve observability of extended system. Then the following communication sequence can be proposed.

$$\{\sigma(0), \sigma(1), \sigma(2), \ldots\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \cdots \right\}$$

By computing observability matrix, it can be found that such sequence dose not allow to preserve the observability of the system for any $k_f$. In fact, with the restriction of one channel there is no periodic communication sequence which guaranties observability of the extended system. Now, if communication limitation is fixed to two channels, the following communication sequence guaranties observability in the extended system.

$$\{\sigma(0), \sigma(1), \sigma(2), \ldots\} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \cdots \right\}$$

In Example 4.7 we will see that any communication sequence which dose not start from $y_1$ can not preserve observability of the extended plant. In the other words, necessary condition for a periodic communication sequence for guaranteing observability of the system is using first output ($y_1$) in first step of communication sequence.

**Example 4.2.** Let the following matrices represent the linear model of a plant connected to a network with communication constraints.

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} b_1 & b_2 \end{bmatrix}, \quad C = I_2 \tag{4.3}$$

Considering (3.16) in the previous chapter, the reachability matrix for the extended plant of (4.3) is given by

$$\mathscr{R}(0, k_f) = \left[ A^{k_f-1}\begin{bmatrix} b_1 & b_2 \end{bmatrix} M_\rho(0), \ \ldots, \ \begin{bmatrix} b_1 & b_2 \end{bmatrix} B M_\rho(k_f-2), \ \begin{bmatrix} b_1 & b_2 \end{bmatrix} B M_\rho(k_f-1) \right] \tag{4.4}$$

Notice that the function $M_\rho$ is to select the inputs which are allowed to access to the medium. The interest here is to find how communication sequence define access to the medium in order to preserve the reachability of the overall system and then full rank of the reachability matrix $\mathscr{R}(0, k_f)$ for a given $k_f$. It can be verified that the original system is reachable when no communication constraints exists. Let

communication constraints is fixed to one channel so the following 2-periodic sequence can be proposed.

$$\{\rho(0),\rho(1),\ldots\} = \left\{ \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix} \right\}. \tag{4.5}$$

then the matrix $R(0,k_f)$ contains either $b_1$ or $b_2$ depending on whether $k_f$ is an odd or even number, and it loses the rank. But, if the 1-periodic sequence

$$\{\rho(0),\rho(1),\ldots\} = \left\{ \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix} \right\} \tag{4.6}$$

is chosen, the reachability matrix $R(0,k_f)$ will remain a full rank matrix and the reachability will be preserved in extended plant.

The interest of these examples is twofold. Firstly, they show that the choice of a sequence is not trivial and that is depends on the structure of the system. Secondly, in Example 4.2 a system with a certain redundancy was presented. This redundancy allows the system to work with only one of its inputs. In such case, reducing the communication channels dose not affect the reachability/obsevability of the extended system. It is worth noting that the latter is not always the case and in this chapter is not considered.
Existence of a communication sequence has already treated in the previous chapter. Here, we concentrate in finding periodic communication sequence that ensure observability and reachability of the NCS. In the general case, a discrete time communication sequence $\eta(k)$ is called T-periodic if $\eta(k) = \eta(k+T)$ for all $k$. If a communication sequence exists for a given value of communication constraints $\omega_\eta$, it is possible that such communication sequence is not unique. In such case two questions arise :

- How can we find all communication sequences with period $T$ such that each of them preserve the observability/reachability of the extended plant (3.14) ?
- What is the minimal size of $\omega_\eta$ which guarantees the preservation of the observability/reachability of the extended plant (3.14).

Some papers presented strategies to find periodic communication sequences(e.g. [ZHV05]). However, in large scale systems these algorithms need high computation burden and in addition, in many applications exact knowledge of the state space matrices is not available and we deal with a system that its model characterized by matrices with zero and free parameters.
By considering these limitations, a graphical algorithm for finding all communication sequences which ensure observability/reachability of the extended plant will be presented in this chapter.

## 4.1 Structured system

In practice, in modeling stage we are faced with the following situation. Firstly, the physical system contains fixed parameters that are due to specific role of certain variables in the system, composition of the subsystems or fixed algebraic relation between variables (e.g. one state variable is derived of another state variable). Secondly, the absence of relation between variables which gives fixed zero entries. Finally, certain parameters are found by empirical relation between variables. Often, all these parameters are subject to the uncertainties and modeling error or in case of nonlinear systems nonzero entries may depend on the operating point [DCvdW03].
In usual linear system approaches, often a full knowledge of the parameters is assumed. furthermore,

they do not allow to take into account available parametric information. Some approaches deal with uncertainty, robustness studies and so on. But the use of this approaches in general causes more complexity in analysis and rather conservative or limited to sufficient condition results.
However, the interesting idea is to consider model characterizations by matrices where the fixed zeros are conserved, while the non-zero entries are replaced by free parameters. Many studies on these kind of systems which called *structured systems* are related to graph theory. State-space representation of a structured system is defined by system matrices(e.g. quadruple matrices $(A,B,C,D)$). But each entry of these matrices can be a fixed zero value or a free parameter. In this class of systems, only knowledge of zero entries is available and when real value of one entry is unknown, this entry is called a free parameter. In this way a structured system represents a large class of linear systems.
Structured models capture most of the available structural information from physical laws and they are helpful to handel uncertain systems and their study needs a low computational burden which allow us to deal with large scale systems.
Consider structured system with $h$ nonzero entries in $A$, $B$, $C$ and $D$. These nonzero entries can be parameterized by scaler real parameters $\lambda_i$ , $i = 1,\ldots,h$ that forming a parameter vector $\Lambda = (\lambda_1,\ldots,\lambda_h)^T \in \mathbb{R}^h$. Structured system $\Sigma_\Lambda$ is given by

$$\Sigma_\Lambda : \begin{cases} x(k+1) &= A^\lambda x(k) + B^\lambda u(k) \\ y(k) &= C^\lambda x(k) + D^\lambda u(k) \end{cases} \tag{4.7}$$

where $A^\lambda$, $B^\lambda$, $C^\lambda$ and $D^\lambda$ have elements from vector $\Lambda$.
Generalization of the results is one of the advantages of property analysis in the structured systems. Because it depends only to structure (zero and free parameters) of the system. In the other words, the validity of such properties (e.g. observability, reachability) is true not only for a given combination of system parameters, but also for almost all values which they can take. However, it is important to mention that the properties depend on the parameter values. They may be true for some values, while for other values not. For some system theoretic properties, once a property is true for one parameter value, it is true for almost all parameter values. we say that a property is true generically [vdW00] if it is true for *almost all* realizations of structured system $\Sigma_\Lambda$. Here "for almost" all realizations can be understood as "for all parameter values($\Lambda \in \mathbb{R}^h$) except for those in some proper algebraic variety in the parameter space".
For example, let us consider a system [Mar08] with the following equations

$$\begin{cases} x(k+1) &= \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} x(k) + \begin{bmatrix} \lambda_5 \\ \lambda_6 \end{bmatrix} u(k) \\ \\ y(k) &= \begin{bmatrix} \lambda_3 & \lambda_4 \end{bmatrix} x(k) \end{cases} \tag{4.8}$$

the observability matrix of the system (4.8) is given by

$$\mathscr{O} = \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} \lambda_3 & \lambda_4 \\ \lambda_1\lambda_3 & \lambda_2\lambda_4 \end{bmatrix} \tag{4.9}$$

The system is observable if rank of observability matrix (4.9) is equal to 2. Determinant of this matrix is found by

$$\begin{aligned} \det(\mathscr{O}) &= \lambda_2\lambda_3\lambda_4 - \lambda_1\lambda_3\lambda_4 \\ &= \lambda_3\lambda_4(\lambda_2 - \lambda_1) \end{aligned} \tag{4.10}$$

The system is observable for all values of parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$ except for $\lambda_3 = 0$ or $\lambda_4 = 0$ or $\lambda_2 - \lambda_1 = 0$. Then, system (4.8) is observable for almost all values of $\Lambda$.

## 4.2 Graphic representation of linear structured systems

A graph can be used to represent structured system $\Sigma_\Lambda$. This graph contains the same information as the structured matrices $A$, $B$ and $C$. Furthermore, this representation has the advantage to clearly visualize the relation between variables in the system.
The properties of systems which depend only on zero and free parameters (i.e. structure of the system) may be studied on this graph and by using graph theory, properties of the system is translated into graph conditions. These graph conditions are generally very intuitive and informative on the system. Interest of using a graph can be summarize as follows [DCvdW03] :

- The associated graph contains all information on the model.
- There is a huge amount of interesting results available in the literature using this model, they concern properties and control of such systems. These results, generally expressed in terms of the associated graph, are often intuitive and easy to interpret physically.

Although structured systems and their graph representation have many advantages in analysis of system properties, they are paid by some drawbacks ([DCvdW03] and [Rei88]) :

- Some important properties particulary stability can not be handled with this model. For instance, stability or instability of structured system $x(k+1) = \lambda x(k)$ can not be found certainly.
- Structured models are based on state space representation with given order. Then all the parametric variations of the system that we accept are limited within this order.

### 4.2.1 Directed graph

A directed graph (digraph) can be used to represent structured linear system $\Sigma_\Lambda$ (4.7). The digraph associated to $(\Sigma_\Lambda)$ is noted $\mathscr{G}(\Sigma_\Lambda)$ and is constituted by a vertex set $\mathscr{V}$ and an edge set $\mathscr{E}$ i.e. $\mathscr{G}(\Sigma_\Lambda) = (\mathscr{V}, \mathscr{E})$. The vertices are associated to the state, the inputs, and measured outputs of $(\Sigma_\Lambda)$ and the edges represent links between these variables. More precisely, $\mathscr{V} = \mathbf{X} \cup \mathbf{U} \cup \mathbf{Y}$. Hence, $\mathscr{V}$ consists of $n+m+p$ vertices. where $\mathbf{U} = \{u_1, u_2, \ldots, u_m\}$ denote the set of input vertices, $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ the set of state vertices and $\mathbf{Y} = \{y_1, y_2, \ldots, y_p\}$ the set of output vertices.
Denoting $(v, v')$ for a directed edge from the vertex $v \in \mathscr{V}$ to the vertex $v' \in \mathscr{V}$. The edge set is $\mathscr{E} = A^\lambda\text{-edges} \cup B^\lambda\text{-edges} \cup C^\lambda\text{-edges} \cup D^\lambda\text{-edges}$, where

$$\begin{aligned} A^\lambda\text{-edges} &= \{(x_j, x_i) | A(i,j) \neq 0\} \\ B^\lambda\text{-edges} &= \{(u_j, x_i) | B(i,j) \neq 0\} \\ C^\lambda\text{-edges} &= \{(y_j, x_i) | C(i,j) \neq 0\} \\ D^\lambda\text{-edges} &= \{(u_j, y_i) | D(i,j) \neq 0\} \end{aligned} \tag{4.11}$$

and $M^\lambda(i,j)$ denotes the $(i,j)$-th element of $M^\lambda$ and $M^\lambda(i,j) \neq 0$ means that the $(i,j)$-th element of $M^\lambda$ is a free parameter (a nonzero).

In order to understand the ideas developed in next sections, some important definitions in the context of graph approach for structured system will be introduced.

- An edge $e = (v_i, v_f) \in \mathscr{E}$, $v_i$ (respectively $v_f$) is the begin (respectively the end) vertex of $e$.
- We denote path $\mathbf{P}$ containing vertices $v_{r0}, v_{r1}, \ldots, v_{ri}$ by $\mathbf{P} = v_{r0} \to v_{r1} \to \ldots, v_{ri}$.
- The pair $(v_{r_j}, v_{r_{j+1}}) \in \mathscr{E}$ for $j = 0, 1, \ldots, i-1$ if there is an integer $l$ and vertices $v_0, v_1, \ldots, v_l \in \mathscr{V}$ such that $(v_{i-1}, v_i) \in \mathscr{E}$ for $i = 1, 2, \ldots, l$. Then, Path $\mathbf{P}$ is of length $l$.
- When $v_0 = v_i$, $\mathbf{P}$ is a *cycle*.

- Some paths are called *disjoint* if they have no common vertex.
- A path **P** is a ***U-rooted*** path if its begin vertex is an element of **U**.
- A path is ***Y-topped*** path if its end vertex is an element of **Y**.
- A **U**-rooted(**Y**-topped) path family consist of disjoint **U**-rooted(**Y**-topped) paths.
- If such family contains a path or a cycle which covers a vertex $v$, it is said to cover such vertex.
- A system $\Sigma_\Lambda$ is *output connected* if in its associated graph $\mathscr{G}(\Sigma_\Lambda)$ for every vertex $x_i$ there exists a direct path to the output set **Y**.
- A system $\Sigma_\Lambda$ is *input connected* if in its associated graph $\mathscr{G}(\Sigma_\Lambda)$ for every vertex $x_i$ there exists a direct path from the input set **U**.
- Let $\mathscr{V}_1$ and $\mathscr{V}_2$ denote two subset of $\mathscr{V}$. The cardinality of $\mathscr{V}_1$ is noted card($\mathscr{V}_1$). A path **P** is said $\mathscr{V}_1-\mathscr{V}_2$ path, if its begin vertex belongs to $\mathscr{V}_1$ and its end vertex belongs to $\mathscr{V}_2$. If only vertices of **P** belonging to $\mathscr{V}_1\cup\mathscr{V}_2$ are its begin and its end vertices, **P** is said a direct $\mathscr{V}_1-\mathscr{V}_2$ path([BHMM07]).
- A set of $l$ disjoint $\mathscr{V}_1-\mathscr{V}_2$ paths is called a $\mathscr{V}_1-\mathscr{V}_2$ linking of size $l$. The linking which consist of a maximal number of disjoint $\mathscr{V}_1-\mathscr{V}_2$ paths are called maximum $\mathscr{V}_1-\mathscr{V}_2$ linking. We define by $p(\mathscr{V}_1-\mathscr{V}_2)$ the size of these maximum $\mathscr{V}_1-\mathscr{V}_2$ linking.
- The length of a $\mathscr{V}_1-\mathscr{V}_2$ linking is defined as the sum of the lengths of all paths.

**Example 4.3.** Consider a structured system described by the following matrices

$$A^\lambda=\begin{bmatrix}0&\lambda_1&0&0&0&0&0&0&0\\ \lambda_2&0&0&0&0&0&0&0&0\\ 0&\lambda_3&0&0&0&0&0&0&0\\ 0&0&\lambda_4&0&0&0&0&0&0\\ 0&0&0&\lambda_5&0&\lambda_6&\lambda_7&0&0\\ 0&0&\lambda_8&\lambda_9&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&\lambda_{10}\end{bmatrix},\quad B^\lambda=\begin{bmatrix}0&0\\ 0&0\\ \lambda_{11}&\lambda_{12}\\ 0&0\\ 0&0\\ 0&0\\ 0&\lambda_{13}\\ 0&\lambda_{14}\\ 0&0\end{bmatrix}$$

$$C^\lambda=\begin{bmatrix}0&0&0&0&\lambda_{15}&0&0&0&0\\ 0&0&0&0&\lambda_{16}&0&0&0&\lambda_{17}\\ 0&0&0&0&0&0&\lambda_{18}&0&0\\ 0&0&0&0&0&0&0&\lambda_{19}&0\end{bmatrix},\quad D^\lambda=\begin{bmatrix}0&0\\ 0&0\\ 0&0\\ 0&\lambda_{20}\end{bmatrix} \tag{4.12}$$

The resulting graph can be depicted as in Figure 4.1. One can notice that the graphic representation is rather intuitive. The vertices are associated to the states, inputs and outputs and edges represent links between them. The graph $\mathscr{G}$ of the system is composed 15 vertices, i.e. $\mathscr{V}=\{u_1,u_2\}\cup\{x_1,x_2,\ldots,x_9\}\cup\{y_1,y_2,y_3,y_4\}$ and 20 edges corresponding to the nonzero entries in the matrices $A^\lambda$, $B^\lambda$, $C^\lambda$ and $D^\lambda$. For example, the edge $(x_2,x_1)$ is associated with parameter $\lambda_1$ in $A^\lambda$ and the edge $(u_2,y_4)$ is associated with parameter $\lambda_{20}$ in $D^\lambda$. Or, two simple paths starting from vertices $x_8$ and $u_2$ and arriving to vertex $y_4$ represent equation $y_4(k)=\lambda_{19}x_8(k)+\lambda_{20}u_2(k)$.

As mentioned above, a lot of properties of the system can be translated into graph condition. For instance, the following matrix is made up of matrices of structured system (4.7).

$$\begin{pmatrix}A^\lambda&B^\lambda\\ C^\lambda&D^\lambda\end{pmatrix}$$

[vdW00] showed that the above matrix *generically* has full row rank, equal to $n+p$, if and only if in the its graph corresponding to the structured system there exists a disjoint union of a size $p$ linking from **U**

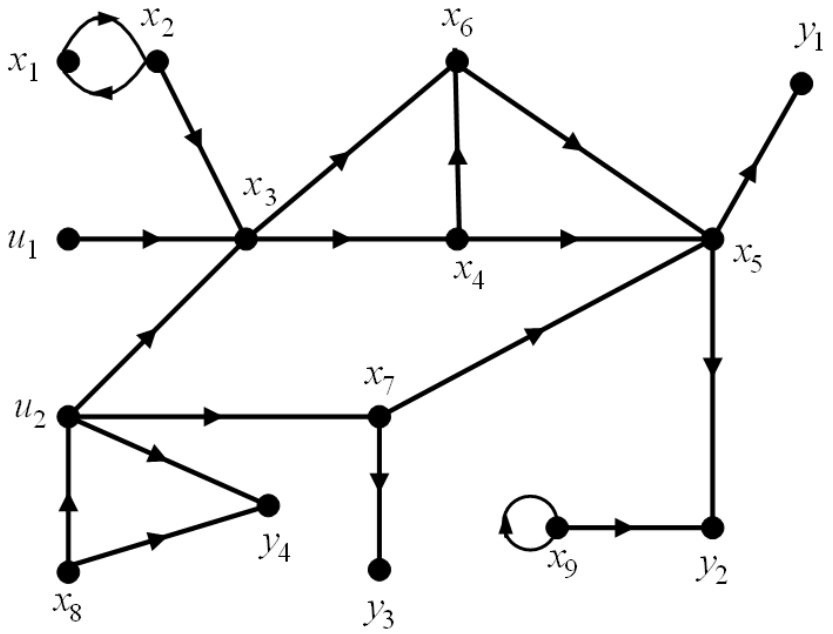



FIGURE 4.1 – Digraph associated to system Example 4.3

to **Y**, a **U**-rooted path family and a cycle family that cover $n$ state vertices.
In literature many papers studied structural properties analysis by using graph-theoretic approach. For example, the Problem of structural controllability and observability of structured system was studied in [WIL86]. [BHMM07] proposed a graph-theoretic approach to analyze state and input observability of a structured linear systems with unknown inputs. Sensor classification for FDI was studied in [CCDYA07]. [DCvdW03] is a survey paper that deals with generic properties of linear systems. A solution to problem of additional sensor location in order to recover state and input observability of structured system was presented in [BF09]. This paper proposed a sensor placement procedure based on classical and well-known graph theory algorithms. In the context of networked control systems (NCSs), [BH08] dealt with the observability analysis of a distributed system.
In the following sections, the directed graph conditions for controllability and observability of a structured system will be studied.

#### 4.2.1.1 Generic controllability and generic observability

**Theorem 4.1.** For a structured system described in (4.7) and its directed graph $\mathscr{G}(\Sigma_\Lambda)$, the following statements are equivalent(see [Rei88]).

1. The structured system $\Sigma_\Lambda$ is generically controllable.
2. In $\mathscr{G}(\Sigma_\Lambda)$ every state vertex is end vertex of U-rooted path and there exists a disjoint union of a **U**-rooted path family and a cycle family that cover all state vertices.

**Example 4.4.** Consider a structured linear system represented by the directed graph(digraph) depicted in Figure 4.2. Clearly, This system is generically controllable since the path $u_1 \to x_2 \to x_1$ and $u_2 \to x_4 \to x_3$ cover all the state vertices.

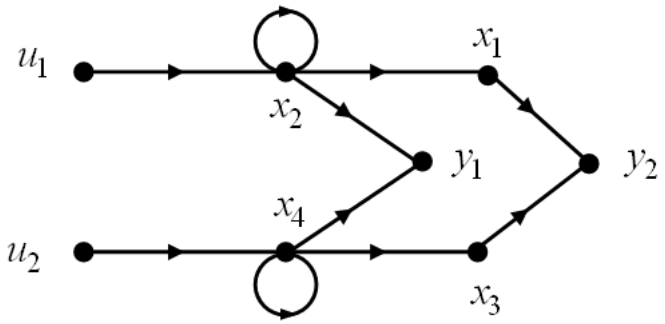



FIGURE 4.2 – Digraph associated to system Example 4.4

**Theorem 4.2.** Let $\Sigma_\Lambda$ be the linear structured system defined by (4.7) with associated graph $\mathscr{G}(\Sigma_\Lambda)$. The system (in fact (A,C)) is structurally observable if and only if :

1. The system $\Sigma_\Lambda$ is output connected(i.e. every state vertex is the begin vertex of an **Y**-topped path).
2. There exists a family of disjoint **Y**-topped paths and a family of cycle covering all the state vertex set in the associated directed graph $\mathscr{G}(\Sigma_\Lambda)$.

**Example 4.5.** Let consider structured system $\Sigma_\Lambda$ with

$$A^\lambda = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad C^\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix}$$

Simply it can be found that the system is not observable by constructing observability matrix $[C, CA]^T$. Directed graph of the system is depicted in Figure 4.3.a. In this graph every vertex is the begin vertex of an **Y**-topped path. However, there dose not exist a disjoint **Y**-topped paths family and a cycle family that cover all state vertices. Indeed, there exists one disjoint path that dose not cover all state vertices.
Now we consider previous structured system with the following matrices

$$A^\lambda = \begin{bmatrix} 0 & 0 \\ 0 & \lambda_3 \end{bmatrix} \quad C^\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix}$$

Its directed graph is illustrated in Figure 4.3.b . Similar to last system, the system is output connected. In addition, there are a disjoint **Y**-topped path family ($x_1 \to y$) and a cycle family ($x_1 \to x_1$) that cover all the state vertices. According to Theorem 4.2, this system is generically observable. This can also be verified by the observability matrix$[C, CA]^T$.
Figure 4.3.c is the associated directed graph for structured system

$$A^\lambda = \begin{bmatrix} 0 & 0 \\ 0 & \lambda_3 \end{bmatrix} \quad C^\lambda = \begin{bmatrix} \lambda_1 & 0 \end{bmatrix}$$

Although, there are a family of **Y**-topped disjoint path family and a cycle that cover all state vertices. But output connectivity condition is not satisfied. So, the system is not observable.

Furthermore, The system presented in the example 4.4 is not generically observable since not all state vertices can be covered by a **Y**-topped disjoint path and a cycle family.

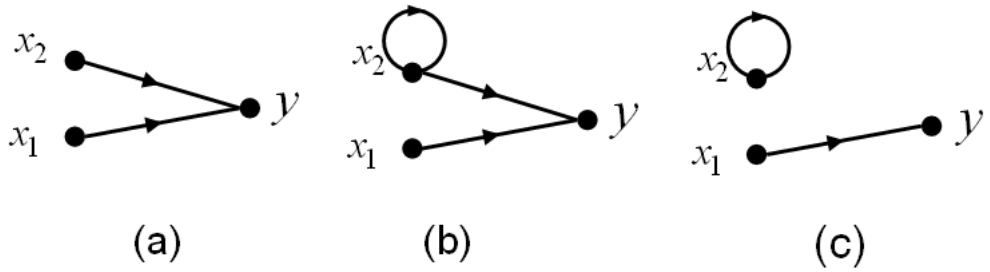



FIGURE 4.3 – Digraph associated to system Example 4.5

### 4.2.2 Dynamic bipartite graph

In this section we introduce the bipartite graph which will be useful in determination of the access sequence.
In order to capture graphically the dynamic of the instantaneous medium access for each step $k$, a particular graph will be associated to the structured system $(\Sigma_\Lambda)$ called dynamic bipartite graph. The dynamic bipartite graph associated to the structured system $(\Sigma_\Lambda)$ is noted $\mathscr{B}_k^\sigma(\Sigma_\Lambda) = (\mathbf{X}, \mathbf{Y}; \mathbf{W}_k)$. The vertex set $\mathbf{X}$ is associated to the states and the vertex set $\mathbf{Y}$ is associated to the outputs. The edge set is defined as follows :

$$\begin{array}{rcl} \mathbf{W}_k^\sigma & = & \{\mathbf{W}_{1,1}^\sigma, \mathbf{W}_{1,2}^\sigma, \ldots, \mathbf{W}_{i,k}^\sigma\} | \\ & & \mathbf{W}_{i,k}^\sigma = (x_j, y_{i,k}), \text{if there exists a path in } \mathscr{G}(\Sigma_\Lambda) \\ & & \text{of lenght } k \text{ between } x_j \text{ and } y_i \end{array} \tag{4.13}$$

The index $k$ must be fixed before constructing the dynamic bipartite graph.

**Example 4.6.** Consider the discrete linear system (4.1) given in Example 4.1, its associated structured system is represented by the following matrices.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2 & 0 \\ 0 & 0 & 0 & 0 & \lambda_3 \\ 0 & 0 & 0 & \lambda_4 & 0 \end{bmatrix} \quad , \quad B = \begin{bmatrix} \lambda_{10} & 0 \\ 0 & \lambda_{10} \\ 0 & 0 \\ 0 & 0 \\ 0 & \lambda_{11} \end{bmatrix} \tag{4.14}$$

$$C = \begin{bmatrix} \lambda_5 & \lambda_6 & 0 & 0 & 0 \\ 0 & \lambda_7 & \lambda_8 & 0 & 0 \\ 0 & 0 & \lambda_9 & 0 & 0 \end{bmatrix}$$

and the directed graph associated to this system is presented in Figure 4.4.

for $k = 5$, the following edge subset can be generated

$$\begin{array}{rcl} \{\mathbf{W}_{1,1}^\sigma; \ldots; \mathbf{W}_{3,1}^\sigma\} & = & \{(x_1, y_{1,1}), (x_2, y_{1,1}), (x_2, y_{2,1}), (x_3, y_{2,1}), (x_3, y_{3,1})\} \\ \{\mathbf{W}_{1,2}^\sigma; \ldots; \mathbf{W}_{3,2}^\sigma\} & = & \{(x_2, y_{1,2}), (x_2, y_{2,2}), (x_4, y_{2,2}), (x_4, y_{3,2})\} \\ \{\mathbf{W}_{1,3}^\sigma; \ldots; \mathbf{W}_{3,3}^\sigma\} & = & \{(x_2, y_{1,3}), (x_2, y_{2,3}), (x_5, y_{2,3}), (x_5, y_{3,3})\} \\ \{\mathbf{W}_{1,4}^\sigma; \ldots; \mathbf{W}_{3,4}^\sigma\} & = & \{(x_2, y_{1,4}), (x_2, y_{2,4}), (x_4, y_{2,4}), (x_4, y_{3,4})\} \\ \{\mathbf{W}_{1,5}^\sigma; \ldots; \mathbf{W}_{3,5}^\sigma\} & = & \{(x_2, y_{1,5}), (x_2, y_{2,5}), (x_5, y_{2,5}), (x_5, y_{3,5})\} \end{array} \tag{4.15}$$

For this kind of bipartite graph attention must be paid to those edges having the same vertices and belonging to the same edge subset $\mathbf{W}_{i,k}^\sigma$. The dynamic bipartite graph generated with the edges subsets calculated above is depicted in Figure 4.5 for $k = 5$.

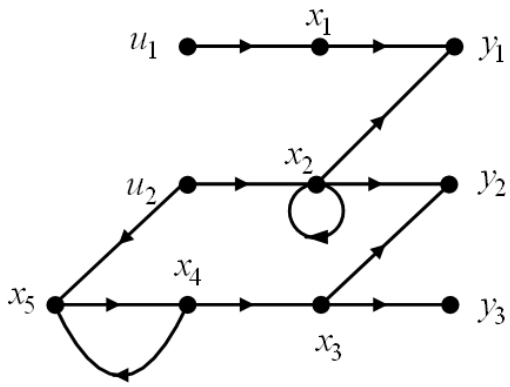



FIGURE 4.4 – Digraph associated to system (4.1)

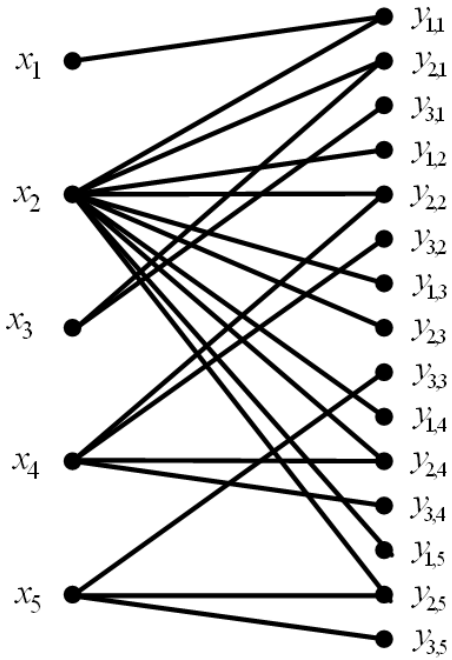



FIGURE 4.5 – Dynamic bipartite graph associated to system (4.1)

For this bipartite graph some definitions must be introduced in order to understand the ideas developed in the following sections.

- A *matching* is an edge set $M \subseteq W_k^\sigma$ such that the edges in $M$ are disjoint.
- A $\eta$-*matching* is a matching taking at most $\eta$ disjoint edges in each edge subset $\mathbf{W}_{i,j}^\sigma$ for $i = 1, \ldots, p$ and $j$ fixed.
- The cardinality of a matching $M$ is the number of edges in it.
- In case $|X| = |Y|$ a $\eta$-matching that covers all vertices is complete $\eta$-matching.
- $\eta$-matching $M$ is maximal, if it has a maximum cardinality.
- A *vertex sequence* is a vertex subset $S \subseteq Y$ such that $S = [s_1, \ldots, s_j]$ where $s_j = \left\{ \begin{bmatrix} y_1 \\ \vdots \\ y_i \end{bmatrix}_j \; | y_i \in \mathbf{W}_{i,j}^\sigma \subseteq M \right\}$

for $j = 1, \ldots, k$. Notice that only one sequence can be generated by maximal $\eta$-matching.
Thus in Figure 4.5 (k=5) for example, neither $w_1 = (x_2, y_{1,1}) \in \mathbf{W}^{\sigma}_{i,1}$ and $w_2 = (x_2, y_{2,1}) \in \mathbf{W}^{\sigma}_{i,1}$ nor $w_1 = (x_2, y_{1,1}) \in \mathbf{W}^{\sigma}_{i,1}$ and $w_3 = (x_2, y_{1,2}) \in \mathbf{W}^{\sigma}_{i,2}$ are disjoint. On the contrary edges $w_1 = (x_1, y_{1,1}) \in \mathbf{W}^{\sigma}_{i,1}$ and $w_4 = (x_2, y_{2,1}) \in \mathbf{W}^{\sigma}_{i,1}$ are disjoint. Moreover, a maximal 3-matching may be constituted of the following edges

$$\begin{array}{l} \{(x_1, y_{1,1}), (x_3, y_{2,1})\} \in \mathbf{W}^{\sigma}_{i,1} \\ \{(x_2, y_{1,2}), (x_4, y_{3,2})\} \in \mathbf{W}^{\sigma}_{i,2} \\ \{(x_5, y_{3,3})\} \in \mathbf{W}^{\sigma}_{i,2} \end{array}$$

It can be verified that is not possible to form a maximal 1-matching in the system. A vertex sequence generated by maximal 2-matching can be given by

$$S = \left\{ \begin{bmatrix} y_1 \\ y_2 \\ 0 \end{bmatrix} , \begin{bmatrix} y_1 \\ 0 \\ y_3 \end{bmatrix} , \begin{bmatrix} 0 \\ 0 \\ y_3 \end{bmatrix} \right\}$$

Moreover, there exists a close relation between maximal $\eta$-matching in a dynamic bipartite graph and the maximal number of disjoint paths in a directed graph as it is enounced in the next Lemma (this relation is also proved for a maximal matching in a normal bipartite graph, see [Mur87]).

**Lemma 4.1.** Let the system $(\Sigma_\Lambda)$ be the linear structured system defined by (4.16) with its associated directed graph $\mathscr{G}(\Sigma_\Lambda)$ and dynamic bipartite graph $\mathscr{B}^{\sigma}_{k}(\Sigma_\Lambda)$ . The following statements are equivalents :

- There exists a family of disjoint **Y**-topped paths and a cycle family covering all the state vertex set in the associated directed graph $\mathscr{G}(\Sigma_\Lambda)$ ,
- There exists a maximal $\eta$-matching of size $n$ with $\eta = p$ in $\mathscr{B}^{\sigma}_{k}(\Sigma_\Lambda)$ for some $k \neq 0$.

$$\Sigma_\Lambda = \left\{ \begin{array}{lcl} x(k+1) & = & A^{\lambda} x(k) + B^{\lambda} u(k) \\ y(k) & = & C^{\lambda} x(k) \end{array} \right. \tag{4.16}$$

**Proof.** Suppose that there exists a complete matching $M$ on $\mathscr{B}^{\sigma}_{k}$ for some $k$. Then the cardinality of the matching $M$ is equal to the number of vertices it covers, $n$. For each $x_j \in \mathbf{X} (1 \leq j \leq n)$, there is a unique sequence of disjoint edges

$$(x_{j_1}, y_{i_1,1}), (x_{j_2}, y_{i_1,2}), \ldots, (x_{j_p}, y_{i_1,l}) \ldots, (x_{j_q}, y_{i_1,k})$$

contained in the matching which form a disjoint path $P = x_{j_q} \to \ldots \to x_{j_1} \to y_{i_1}$, and a cycle family $P_c = x_{j_q} \to \ldots \to x_{j_{p+1}} \to x_{j_q}$ in the associated directed graph $\mathscr{G}$. Thus a complete matching on $\mathscr{B}^{\sigma}_{k}$ determines a family of disjoint **Y**-topped paths $P_j = x_{j_p} \to \ldots c_{j_1} \to y_{i_1}$ and cycle families $P_{c,j} = x_{j_q} \to \ldots \to x_{j_{p+1}} \to x_{j_q}$ for $j = 1, \ldots, n$ and $i = 1, \ldots, p$ on $\mathscr{G}$
Conversely, suppose that there exists a family of disjoint **Y**-topped paths covering all the state vertices on $\mathscr{G}$. Then, by the definitions given above concerning the disjoint edges and the construction of the edge set $\mathbf{W}^{\sigma}$ on $\mathscr{B}^{\sigma}_{k}$, it is possible to construct a sequence of disjoint edges which form a complete matching on $\mathscr{B}^{\sigma}_{k}$ for some $k \neq 0$.

### 4.2.3 Generic reachability

For studying reachability conditions, bipartite graph $\mathscr{B}^{\rho}_{k}(\Sigma_\Lambda)$ associated to the structured system $(\Sigma_\Lambda)$ will be constructed as follows. $\mathscr{B}^{\rho}_{k}(\Sigma_\Lambda) = (\mathbf{U}, \mathbf{X}; \mathbf{W}^{\rho}_{k})$. The vertex set **X** is associated to the states

and the vertex set **U** is associated to the inputs. and

$$\begin{aligned} \mathbf{W}_k^{\rho} &= \{\mathbf{W}_{1,1}^{\rho}, \mathbf{W}_{1,2}^{\rho}, \ldots, \mathbf{W}_{i,k}^{\rho}\} | \\ &\quad \mathbf{W}_{i,k}^{\rho} = (u_{i,k}, x_j), \text{if there exists a path in } \mathscr{G}(\Sigma_\Lambda) \\ &\quad \text{of lenght } k \text{ between } u_i \text{ and } x_j \end{aligned} \tag{4.17}$$

Similar to bipartite graph associated to outputs, $k$ must be fixed before constructing $\mathscr{B}_k^{\rho}(\Sigma_\Lambda)$.
In case of reachability analysis for the structured system described in (4.14), the following edge set for $k = 4$ can be obtained.

$$\begin{aligned} \{\mathbf{W}_{1,1}^{\rho}; \mathbf{W}_{2,1}^{\rho}\} &= \{(u_{1,1}, x_1), (u_{2,1}, x_2), (u_{2,1}, x_5)\}; \\ &\quad \{(u_{2,2}, x_2), (u_{2,2}, x_4)\}; \\ &\quad \{(u_{2,3}, x_2), (u_{2,3}, x_5), (u_{2,3}, x_3)\}; \\ &\quad \{(u_{2,4}, x_2), (u_{2,4}, x_4)\}. \end{aligned} \tag{4.18}$$

and its associated dynamic graph $\mathscr{B}_4^{\rho}$ is depicted in Figure 4.6. Definitions which were introduced in

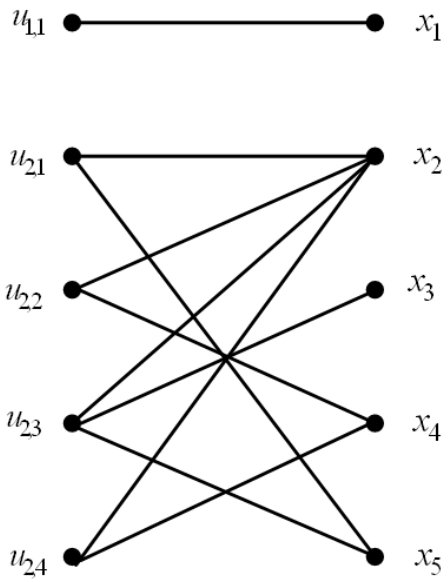



FIGURE 4.6 – $\mathscr{B}_4^{\rho}$ Dynamic bipartite graph w.r.t. input associated to the structured system (4.14)

bipartite graph $\mathscr{B}_k^{\sigma}$ associated to outputs can be revised in $\mathscr{B}_k^{\rho}$, bipartite graph associated to inputs, as follow :

- A *matching* is an edge set $M \subseteq W^{\rho}$ such that the edges in $M$ are disjoint.
- A $\eta$*-matching* is a matching taking at most $\eta$ disjoint edges in each edge subset $\mathbf{W}_{i,j}^{\rho}$ for $i = 1, \ldots, m$ and $j$ fixed.
- The cardinality of a matching $M$ is the number of edges in it.
- In case $|X| = |U|$ a $\eta$-matching that covers all vertices is complete $\eta$-matching.
- $\eta$-matching $M$ is maximal, if it has a maximum cardinality.
- A *vertex sequence* is a vertex subset $S \subseteq U$ such that $S = [s_1, \ldots, s_j]$ where $s_j = \left\{ \begin{bmatrix} u_1 \\ \vdots \\ u_i \end{bmatrix}_j \; | u_i \in \mathbf{W}_{i,j}^{\rho} \subseteq M \right\}$ for $j = 1, \ldots, k$. Notice that only one sequence can be generated by maximal $\eta$-matching.

For example, in Figure 4.6 edges $(u_{2,2},x_2) \in W^{\rho}_{2,2}$ and $(u_{2,3},x_2) \in W^{\rho}_{2,3}$ are not disjoint. A maximal 1-matching may be constituted of the following edges

$$\{(u_{1,1},x_1);(u_{2,1},x_5);(u_{2,2},x_2);(u_{2,3},x_3);(u_{2,4},x_4)\}$$

The vertex sequence generated by this maximal 1-matching is

$$S = \left\{ \begin{bmatrix} u_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix} \right\}$$

Considering relation between maximal $\eta$-matching dynamic bipartite graph and the maximal number of disjoint paths in directed graph ([Mur87]) and lemma 4.1 the following lemma can be defined.

**Lemma 4.2.** Let the system $(\Sigma_\Lambda)$ be the linear structured system defined by (4.16) with its associated directed graph $\mathscr{G}(\Sigma_\Lambda)$ and dynamic bipartite graph $\mathscr{B}^{\rho}_{k}(\Sigma_\Lambda)$ . The following statements are equivalents :

- There exists a family of disjoint **U**-rooted paths and a cycle family covering all the state vertex set in the associated directed graph $\mathscr{G}(\Sigma_\Lambda)$ ,
- There exists a maximal $\eta$-matching of size $n$ with $\eta = m$ in $\mathscr{B}^{\rho}_{k}(\Sigma_\Lambda)$ for some $k \neq 0$.

**Proof.** Similar to proof of lemma 4.1, suppose that there exists a complete matching $M$ on $\mathscr{B}^{\rho}_{k}$ for some $k$. Then the cardinality of the matching $M$ is equal to the number of vertices it covers, $n$. For each $x_j \in \mathbf{X}(1 \leq j \leq n)$, there is a unique sequence of disjoint edges

$$(u_{i_1,1},x_{j_1}),(u_{i_1,2},x_{j_2}),\ldots,(u_{i_1,l},x_{j_p}),\ldots,(u_{i_1,k},x_{j_q})$$

contained in the matching which form a disjoint path $P = u_{i_1} \to x_{j_1} \to \ldots x_{j_p}$, and a cycle family $P_c = x_{j_{p+1}} \to \ldots \to x_{j_q} \to x_{j_{p+1}}$ in the associated directed graph $\mathscr{G}$. Thus a complete matching on $\mathscr{B}^{\rho}_{k}$ determines a family of disjoint **U**-rooted paths $P_j = u_{i_1} \to x_{j_1} \to \ldots x_{j_p}$ and cycle families $P_{c,j} = x_{j_{p+1}} \to \ldots \to x_{j_q} \to x_{j_{p+1}}$ for $j = 1,\ldots,n$ and $i = 1,\ldots,m$ on $\mathscr{G}$
Conversely, suppose that there exists a family of disjoint **U**-rooted paths covering all the state vertices on $\mathscr{G}$. Then, by the definitions given above concerning the disjoint edges and the construction of the edge set $\mathbf{W}^{\rho}$ on $\mathscr{B}^{\rho}_{k}$, it is possible to construct a sequence of disjoint edges which form a complete matching on $\mathscr{B}^{\rho}_{k}$ for some $k \neq 0$.

## 4.3 Communication sequence design, observability

In the previous chapter, it was considered that the original system (4.19) is connected to central station(controller/FDI) via a network with constraints on medium access. By using two periodic communication sequences $\sigma(k)$ and $\rho(k)$ an extended plant (4.20) was modeled.

$$\Sigma_o : \begin{cases} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k) \end{cases} \tag{4.19}$$

$$\Sigma_e : \begin{cases} x(k+1) &= Ax(k)+BM_\rho(k)u(k) \\ y(k) &= \bar{M}_\sigma(k)Cx(k) \end{cases} \tag{4.20}$$

In this section, a graphical method to search the T-periodic communication sequences which preserve observability of extended plant (4.20) will be studied.

In a graphic-theoretic approach, conditions for the observability of the system (4.19) without communication constraints, can be deduced from [CDT05] and [BHMM07], and stated in Theorem 4.2. As mentioned in 3.2 the original system (4.19) is assumed observable and so it is output connected. In the sequel we assume the output connectivity of structured system $(\Sigma_\Lambda)$ , constructed from (4.19), without loss of generality. Then, we concentrate in second condition of Theorem 4.2. It can be expressed in terms of maximal matching in the associated bipartite graph $\mathscr{B}_k^\sigma$ of system $(\Sigma_\Lambda)$ according to Lemma 4.1.
In 3.2, it was shown that extended plant (4.20) is observable on $[0,k_f]$ if rank of observability matrix $\mathscr{O}(0,k_f)$ is equal to $n$ i.e.

$$\mathscr{O}(0,k_f) = \begin{bmatrix} \bar{M}_\sigma(0)C \\ \bar{M}_\sigma(A)CA \\ \vdots \\ \bar{M}_\sigma(k_f-1)CA^{k_f} \end{bmatrix} = n \tag{4.21}$$

Consequently, conditions of Theorem 4.2 are equivalent to equation (4.21). It is clear that even with communication constraints this condition must be satisfied if the interest is to preserve the observability. In the next proposition we extend the graphic conditions given in Theorem 4.2 to the case when communication constraint is taken into account.

**Proposition 4.1.** Let $\Sigma_\Lambda$ be the linear structured system defined by (4.16) with associated dynamic bipartite graph $\mathscr{B}_k^\sigma$ $(\Sigma_\Lambda)$ . The system is structurally observable on $[k_0,k_f]$ if and only if in $\mathscr{B}_k^\sigma$ $(\Sigma_\Lambda)$ there exists a maximal and complete $\omega_\sigma$-matching of size $n$

Proposition 4.1 can be easily proved considering results of Lemma 4.1 and those found in [Zha05]. It is worth noting that in the dynamic bipartite graph a maximal matching and $\omega_\sigma$-matching of size $n$ may not be unique. As a consequence different sequence $S_j$ may be generated for every maximal and complete $\omega_\sigma$-matching.
Then, according to Proposition 4.1 every combination of disjoint edges which form a maximal $\omega_\sigma$-matching of size $n$ preserves the observability of the system. To find all possible communication sequence that preserve observability of the NCS the following algorithm is proposed.

**Algorithm 4.1.** Let $\mathscr{G}(\Sigma_\Lambda)$ be the directed graph associated to the structured system $\Sigma_\Lambda$ defined in (4.16) :

**1.** From the directed graph $\mathscr{G}(\Sigma_\Lambda)$ determine the size $k$ of the maximal **Y**-topped path,

**2.** Build the dynamic bipartite graph $\mathscr{B}_k^\sigma(\Sigma_\Lambda)$ for $k$ ($k$ might be infinity, in such case fix $k=n$),

**3.** Set the constraint $\omega_\sigma$ of the communication medium,

**4.** If there exists in $\mathscr{B}_k^\sigma(\Sigma_\Lambda)$ a maximal $\omega_\sigma$-matching $M$ of size $n$

- **4.1** then the output communication sequence $\sigma(k)$ is given by associated vertex sequence $S$ formed with the maximal $\omega_\sigma$-matching $M$
- **4.2** Else, for given communication constraint $\omega_\sigma$ observability of extended plant can not be preserved. If it is possible put $\omega_\sigma = \omega_\sigma + 1$ and return to step **3**.

In order to illustrate the selection of the communication sequence, the following example is given.

**Example 4.7.** Let consider structured system described in (4.14). Its directed graph and dynamic bipartite graph for $k=5$ were illustrated in Figure 4.4 and Figure 4.5, respectively.

Suppose the communication constraint imposed on the netwotk is $\omega_\sigma = 1$ (i.e. only one available channel at a time). It is clear by Figure 4.5 that a 1-maximal matching of size 5 can not be found. Indeed, it can not be possible to cover all the state vertices using only one output at a time. The reason is because of vertices $x_1$ and $x_3$ are at same distance, $k = 1$ from different outputs, $y_1$ and $y_3$ respectively. It means that these two vertices should be measured simultaneously and in the first sampled time in order to preserve observability of extended plant. From analysis of this system in Example 4.6 a 2-maximal matching can be found. Thus, the minimal communication constraint accepted for this system to be observable is $\omega_\sigma = 2$.
The following (not unique) 2-maximal matching may be cited :

$$\begin{array}{l} \{(x_1, y_{1,1}), (x_3, y_{2,1})\} \in W_{i,1}^{\sigma} \\ \{(x_2, y_{1,2}), (x_4, y_{3,2})\} \in W_{i,2}^{\sigma} \\ \{(x_5, y_{3,3})\} \in W_{i,3}^{\sigma} \end{array} \tag{4.22}$$

and then the vertex sequence is given by

$$S_1 = \left\{ \begin{bmatrix} y_1 \\ y_2 \\ 0 \end{bmatrix} , \begin{bmatrix} y_1 \\ 0 \\ y_3 \end{bmatrix} , \begin{bmatrix} \times \\ \times \\ y_3 \end{bmatrix} \right\} \tag{4.23}$$

which obviously a 3-periodic communication sequence given by

$$\{\sigma_1(0), \sigma_1(1), \sigma_1(2), \ldots\} = \left\{ \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} , \begin{bmatrix} \times \\ \times \\ 1 \end{bmatrix} , \ldots \right\} \tag{4.24}$$

Where "$\times$" indicates that respective output, $y_2$ or $y_3$, may be used as long as output $y_3$ is always present. Indeed output $y_2$ or $y_3$ can be added or not without loss of the rank in the observability. It is possible to verify that the rank of the observability matrix is not lost.

$$\mathscr{O} = \begin{bmatrix} \bar{M}(0)C \\ \bar{M}(1)CA \\ \bar{M}(2)CA^2 \end{bmatrix} = \begin{bmatrix} \lambda_5 & \lambda_6 & 0 & 0 & 0 \\ 0 & \lambda_7 & \lambda_8 & 0 & 0 \\ 0 & \lambda_1 * \lambda_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2 * \lambda_9 & 0 \\ 0 & 0 & 0 & 0 & \lambda_2 * \lambda_3 * \lambda_9 \end{bmatrix} \tag{4.25}$$

It is also possible to find another maximal 2-matching

$$\begin{array}{l} \{(x_1, y_{1,1}), (x_3, y_{3,1})\} \in W_{i,1}^{\sigma} \\ \{(x_2, y_{2,2}), (x_4, y_{3,2})\} \in W_{i,2}^{\sigma} \\ \{(x_5, y_{2,3})\} \in W_{i,3}^{\sigma} \end{array} \tag{4.26}$$

which gives the following associated vertex sequence :

$$S_2 = \left\{ \begin{bmatrix} y_1 \\ 0 \\ y_3 \end{bmatrix} , \begin{bmatrix} 0 \\ y_2 \\ y_3 \end{bmatrix} , \begin{bmatrix} \times \\ \times \\ y_3 \end{bmatrix} \right\} \tag{4.27}$$

with its 3-periodic communication sequence

$$\{\sigma_2(0), \sigma_2(1), \sigma_2(2), \ldots\} = \left\{ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} , \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} , \begin{bmatrix} \times \\ \times \\ 1 \end{bmatrix} , \ldots \right\} \tag{4.28}$$

Similar to the first communication sequence (4.24), $y_2$ or $y_3$ on the third column may be used or not without loss of the observability. More communication sequences can be found. In all the cases it is interesting to point out that some outputs are fixed to an established order. This fact is completely related to the system's structure. In this example, $y_1$ must have access to the medium at the beginning of any chosen sequence.
It is worth noting that the method allows to calculate the communication sequences for different communication constraints. Also, this algorithm allows to find the minimal communication constraint upon which the extended plant preserves observability. Once the communication constraint is fixed, all the admissible communication sequences are generated. It is important to mention that the proposed algorithm does not intend to be the most efficient one. In fact, this algorithm is of exponential complexity because its aim is to calculate all the sequences.
If the interest is to implement a polynomial algorithm, the problem of preserving observability in the extended plant should be formulated differently. The idea here would be to answer to the question of whether or not a predefined communication sequences preserves observability of a given extended plant. In such a case, the communication restriction and the size $T$ of the periodic communication are fixed from the beginning. The algorithm would not explore all the possibilities but only one : that given by the predefined sequence.
With a given communication sequence a normal bipartite graph can be construct. It remains to verify if the given bipartite graph forms a complete matching. It is well known that the test of matching in a bipartite graph is of polynomial complexity.

## 4.4 Communication sequence design, reachability

Based on observability issue presented in the previous section, we can state the conditions of reachability with communication constraints and then present an algorithm for finding all periodic communication sequences which preserve reachability of extended plant.
In extended plant (4.20) supposed that initial state $x(0) = 0$ and (4.20) evolves from $k = 0$ to $k = k_f$. Then

$$x(k_f) = \mathscr{R}(0,k_f) \cdot [\bar{u}(0)\ \bar{u}(1)\ \ldots\ \bar{u}(k_f-1)]^T \tag{4.29}$$

where

$$\mathscr{R}(0,k_f) = \left[A^{k_f-1}BM_\rho(0)\ A^{k_f-2}BM_\rho(1)\ \ldots\ BM_\rho(k_f-1)\right] \tag{4.30}$$

The extended plant (4.20) is reachable on $[0,k_f]$ if

$$\operatorname{rank}\big(\mathscr{R}(0,k_f)\big) = n \tag{4.31}$$

Reachability conditions of a directed graph can be derived from Theorem 4.1. These conditions can be applied in our case since they involve the same condition (4.30). Thus conditions to the reachability for linear time discrete systems with communication constraints can be deduced directly from those found in [Lin74] and [DCvdW03].
Original system (4.19) was considered reachable. Therefore, input connection (first condition of Theorem 4.1) is assumed. It remains second condition which is similar to the observability case but considering inputs instead. Thus, next proposition is easily deduced from Theorem 4.1 and Lemma 4.2, using exactly the same arguments which than those for observability issue. The proposition is presented without its proof.

**Proposition 4.2.** Let $(\Sigma_\Lambda)$ be the linear structured system defined by (4.16) with associated dynamic bipartite graph $\mathscr{B}_k^\rho(\Sigma_\Lambda)$ . The system is structurally reachable on $[k_0, k_f]$ if and only if in $\mathscr{B}_k^\rho(\Sigma_\Lambda)$ there exists a maximal and complete $\omega_\rho$ matching of size $n$

This proposition can be proved by considering results of Lemma 4.2 and those in [Zha05]. Similar to observability case, in dynamic bipartite graph $\mathscr{B}_k^\rho$ a maximal and complete $\omega_\rho$ matching of size $n$ could not be unique. As a consequence, different sequence $S_j$ may be generated.
According to the proposition 4.2, any combination of disjoint edges which form a maximal $\omega_\rho$ matching of size $n$ preserve the reachability of the extended system (4.20). First step for finding all communication sequences which guarantee reachability of extended plant is fixing communication constraints $\omega_\rho$. if a maximal and complete $\omega_\rho$-matching is found, it means that all vertex sequence $S_j$ which were formed with this matching can be consider as desired communication sequence.
The following algorithm can be proposed to find all communication sequences that preserve reachability of extended plant.

**Algorithm 4.2.** Let $\mathscr{G}(\Sigma_\Lambda)$ be the directed graph associated to the structured system $\Sigma_\Lambda$ defined in (4.16) :

**1.** From the directed graph $\mathscr{G}(\Sigma_\Lambda)$ determine the size $k$ of the maximal **U**-rooted path,

**2.** Build the dynamic bipartite graph $\mathscr{B}_k^\rho(\Sigma_\Lambda)$ for $k$ ($k$ might be infinity, in such case fix $k = n$),

**3.** Set the constraint $\omega_\rho$ of the communication medium,

**4.** If there exists in $\mathscr{B}_k^\rho(\Sigma_\Lambda)$ a maximal $\omega_\rho$-matching $M$ of size $n$

- **4.1** then the output communication sequence $\rho(k)$ is given by associated vertex sequence $S$ formed with the maximal $\omega_\rho$-matching $M$
- **4.2** Else, for given communication constraint $\omega_\rho$ reachability of extended plant can not be preserved. If it is possible put $\omega_\rho = \omega_\rho + 1$ and return to step **3**.

In order to illustrate the selection of the communication sequences, the following example is given.

**Example 4.8.** Let consider structured system described in (4.14). Its directed graph $\mathscr{G}(\Sigma_\Lambda)$ and dynamic bipartite graph $\mathscr{B}_4^\rho(\Sigma_\Lambda)$ for $k = 4$ were illustrated in Figure 4.4 and Figure 4.6, respectively.
In the first step, communication constraint is fixed to one channel ($\omega_\rho = 1$). It is clear by Figure 4.6 that a maximal 1-matching of size 5 can be found. It means that with this communication constraint we can find a communication sequence that guarantees reachability of extended plant. The 1-matching of size 5 is given by

$$\left\{(u_{1,1}, x_1), (u_{2,1}, x_5), (u_{2,2}, x_2), (u_{2,3}, x_3), (u_{2,4}, x_4)\right\} \tag{4.32}$$

As it can be seen from Figure 4.6, only one 1-matching of size 5 can be found for communication constraints $\omega_\rho = 1$.The sequence vertex associated to this matching is

$$S = \left\{ \begin{bmatrix} u_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix}, \begin{bmatrix} 0 \\ u_2 \end{bmatrix} \right\} \tag{4.33}$$

to which is associated the following 5-periodic communication sequence

$$\{\rho(0), \rho(1), \ldots\} = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ldots \right\} \tag{4.34}$$

## 4.5 Conclusion

This chapter studied the problem of finding periodic communication sequences that preserve reachability/observability of a linear time invariant plant. The plant communicate with its controller and its FDI system over a shared medium. The medium supports a limited number of simultaneous connection between sensors and controllers. Selecting a group of sensors or actuators at each sampling instant for accessing to the network, has direct effect on observability and reachability of the whole system. An graph-based algorithm for finding all communication sequences which guarantee observability/reachability extended plant was proposed.
Due to using structured system for generating directed graph and bipartite graph, only information about structure of original system is needed. Therefore, the proposed algorithms can be deals with a NCS with uncertain parameters. In addition, comparing with previous works that studied design of communication sequence, this strategy is simpler and complex mathematical computation is not necessary. So it can be applied in systems with large number of sensors and actuators.
Proposed method in this chapter studied design of offline and periodic communication sequence. It is possible to use these algorithms to generate non periodic communication sequences and if there are no constraint on computation in central station, an dynamic scheduling could be achievable.
In the FDI context, the structural approach can give more insight dealing with the detectability of system and the generation of communication sequences that preserve it.

# Chapter 5

# Fault detection in limited communication with dynamic scheduling

The problem of fault detection and isolation with communication constraints was studied in chapter three. The rules that allow the exchange of information, i.e. message scheduling policy, was specified offline. They are fixed during system running and the communication sequences were repeated periodically.
In this chapter the problem of semi-online scheduling and fault detection of a set of subsystems that share a communication network is studied. The proposed scheduling algorithm can be considered as a compromise between offline scheduling (with problem of ignoring dynamic of systems) and online scheduling(with problem of CPU charge for online computation) algorithms. This chapter is organized as follows :
First of all, problem of message scheduling with the objective of fault detection is studied. Then, a strategy to semi-online message scheduling design and fault detection with parity space approach is presented. In next section, a modification of this strategy for observer-based fault detection is proposed. And, last section denotes to un implementation of this strategy on the CAN network by using the hybrid priority approach.

## 5.1 Problem formulation

A distributed process described by

$$\dot{x}_{ps}(t) = A_{ps}x_{ps}(t) + B_{ps}u_{ps}(t) + E_{ps}d_{ps}(t) + E_{ps}f_{ps}(t) \tag{5.1}$$

Where $x_{ps} \in \mathbb{R}^{k_n}, u_{ps} \in \mathbb{R}^{k_u}, d_p \in \mathbb{R}^{k_d}, f_{ps} \in \mathbb{R}^{k_f}$ are process state, input ,unknown disturbance and the fault vectors to be detected. $A_{ps}, B_{ps}, E_{ps}, F_{ps}$ are known matrices of appropriate dimensions.
It is assumed that the process can be decomposed into $p$ sub-systems, and each of them is equipped with a controller, a number of sensors and actuators. By means of discretization with sampling time $h_0$, *i-th* subsystem ($i = 1,^l dots, p$) is then described by :

$$\Sigma_i : \begin{cases} x_i(k+1) = A_i x_i(k) + B_i u_i(k) + E_{d,i} d_i(k) + E_{f,i} f_i(k) \\ y_i(k) = C_i x_i(k) + D_i u_i(k) + F_{d,i} d(k) + F_{f,i} f(k) \end{cases} \tag{5.2}$$

Where $y_i(k) \in \mathbb{R}^{m_i}$ stands for sensors signals of the *i-th* subsystem. Subsystem outputs are transmitted to FDI module via a network. All subsystems used a shared communication medium to send their measurements. If there is no constraint on communication, all FDI modules will be updated and one residual can be generated at each sampling time. Because of constraints on the communication , only one subsystem can have access to network at each sampling time. As a consequence, only FDI module of one subsystem can be updated by new measurements at each moment. In figure 5.1 distributed process (5.1) is illustrated.

Traditionally, research on scheduling techniques focused on static strategies that would guarantee ave-

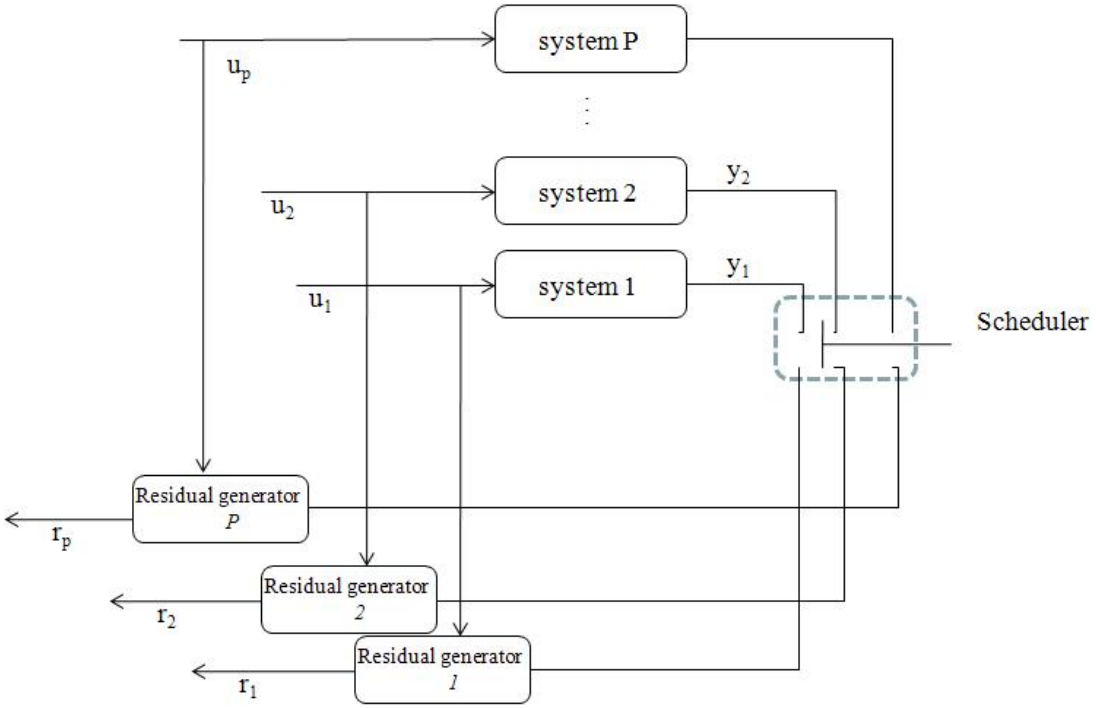



FIGURE 5.1 – communication architecture for distributed process.

rage control performance at the expense of permanently occupying the available bandwidth. In this case, one straightforward solution is to let medium access of sensors follow a pre-defined communication sequence that can be repeated periodically. By using this approach, linear time invariant (LTI) model of subsystems will be changed to a linear periodic model. Previous chapters were dedicated to develop FDI system and periodic communication sequence with offline scheduling of the messages. From the control perspective, the static scheduling method is an "open-loop" solution because once established at system set-up, the static scheduling will not be adjusted during system running. Static techniques may not be efficient when changing conditions occurs at the control-application or network levels. Also it may not be able to satisfy an urgent need of a NCS. For example, A NCS, which is tracking a trajectory at crucial point, may need more bandwidth to send information than typical condition. In addition, when an abnormality or fault occurrence in one part of process there is no difference between faulty subsystem and other subsystems. It may cause a slow fault detection or a false alarm.

An alternative to static communication sequences is to let the medium access of subsystems be determined online which is based on needs of the system. This type of access scheduling is often referred to as *dynamic scheduling*. Problem of stability of a collection of continuous-time feedback LTI systems that rely on an idealized network of limited capacity was considered in [HVK02]. Network allocation is based on most attention to subsystems whose states are further from origin. In a related approach

[GCH06], the problem of the distributed control over deterministic real-time networks is addressed. In this work optimal control and scheduling problem is formulated. Then, an efficient online approach for the solving of this problem is proposed. The Large-Error-First (LEF) scheduling algorithm was studied in [YMF03]. It proposed to assign bandwidth to NCS with different priorities based on realtime errors obtained from system states. Priority assignment can be explained as follows : (i) if some plants are in the steady state, they will not receive actual resources, (ii) if some plants are in the transient state, the first priority will be assigned to this group, and (iii) the plant with the largest error is given the highest priority. Implementation of this idea on the CAN network will be studied in 5.4. [KCM09] proposed a scheduling method which was based on maximum allowable delay bound. It is obtained for the stability of the discrete system and is used as basic parameter for event-based scheduling. Proposed scheduling method can efficiently guarantee realtime transmission of sporadic data.
Here, a semi-online scheduling in the context of fault detection is presented. The scheduling is based on this idea that a subsystem with higher probability of fault must have more access to medium to send its outputs than other subsystems. In the other words, scheduling decisions are based on size of residual deviation of each subsystem from its normal condition. For this purpose, a set of admissible communication sequences is defined in advance. Each communication sequence specifies a slot time for each subsystem to access to the network. Then during system running based on needs of each subsystem, a communication sequence from the ensemble of admissible communication sequences is chosen. Criterion of admissibility of one communication sequence can be possibility of generation of minimum one residual in this communication sequence for all subsystem. It means that the number of measurements updates of each subsystem must be higher than its moving time windows in the context of parity space approach (5.2) or stability of residual must be guaranteed in the context of observer based diagnostic (5.3). Length of access time slot for one subsystem depends on the size of deviation of its corresponding residual (i.e. class of residuals) comparing with class of residual of all subsystems.

## 5.2 Parity space-based residual generation and semi-online scheduling

The first step to successful fault detection is residual generation. For this purpose, a residual generator (RG) based on parity space approach [Ger97] is developed for each subsystem. All residual generators are identical so for simplicity index $i$ is omitted. Let $s$ be an integer denoting the length of a moving time window. The output of system (5.2) over the moving window $[k-s,k]$ can be expressed by the initial state $x(k-s)$, the stacked control input vector $u_{k,s}$, the stacked disturbance vector $d_{k,s}$, and the fault vector $f_{k,s}$ as

$$y_{k,s} = H_{0,s}x(k-s) + H_{u,s}u_{k,s} + H_{d,s}d_{k,s} + H_{f,s}f_{k,s} \tag{5.3}$$

where

$$\xi_{k,s} = \begin{bmatrix} \xi(k-s) \\ \xi(k-s+1) \\ \vdots \\ \xi(k) \end{bmatrix} \tag{5.4}$$

with $\xi$ standing for $y,u,d,f$ and

$$H_{0,s} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^s \end{bmatrix}, \quad H_{u,s} = \begin{bmatrix} D & 0 & \cdots & 0 \\ CB & D & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ CA^{s-1}B & \cdots & CB & D \end{bmatrix} \tag{5.5}$$

$H_{d,s}$ and $H_{f,s}$ are constracted similarly as $H_{u,s}$ and can be acheieved by replacing $B$, $D$, respectively, by $E_d, F_d$ and $E_f, F_f$.
residual for each subsystem can be constructed as

$$r(k) = v_s(y_{k,s} - H_{u,s}u_{k,s}). \tag{5.6}$$

The parity space $P_s$ is defined by $P_s = \{v_s | v_s[H_{0,s}, H_{d,s}] = 0, v_s H_{f,s} \neq 0\}$ and vectors ,$v_s$, belonging to $P_s$ are called parity vectors. Influence of the initial state can be eliminated if condition

$$\text{rank}\ ([H_{0,s}, H_{d,s}, H_{f,s}]) > \text{rank}\ ([H_{0,s}, H_{d,s}]) \tag{5.7}$$

is satisfied. But this condition is strict and usually is not satisfied in some applications. Therefore the FD problem may be formulated as to solve the following optimization problem

$$\begin{aligned} \max_{v_s, v_s H_{0,s}=0} J_{PS}(v_s) &= \max_{v_s, v_s H_{0,s}=0} \frac{\sup\limits_{d_{k,s}=0, f_{k,s}\neq 0} (r^T(k)r(k)/f_{k,s}^T f_{k,s})}{\sup\limits_{f_{k,s}=0, d_{k,s}\neq 0} (r^T(k)r(k)/d_{k,s}^T d_{k,s})} \\ &= \max_{v_s, v_s H_{0,s}=0} \frac{v_s H_{f,s} H_{f,s}^T v_s^T}{v_s H_{d,s} H_{d,s}^T v_s^T} \end{aligned} \tag{5.8}$$

[ZD08] explained a method to solve this problem.

### 5.2.1 Offline communication sequence design

Here, a *communication sequence* specifies length of time slot and order of accessing to medium for each subsystem. Let the binary-value of function $\sigma_i(k)$ for $i = 1, \ldots, p$ denotes the medium access status of outputs of $i-th$ subsystem at time $k$, i.e., $\sigma_i(k) : \mathbb{Z} \to \{0, 1\}$, where $'1'$ means "accessing" and $'0'$ means "not accessing". It is assumed that only one subsystem has access to medium at each moment. Therefore, the medium access status of all $p$ subsystems will be presented by a *p-to-1 communication sequence*

$$\sigma(k) = [\sigma_1(k), \ldots, \sigma_p(k)]$$

Considering (5.6), information of distributed actuators and sensors of $i-th$ subsystem in the time interval $[k-s, k]$ must be available for generating residual at instant $k$. It means that each subsystem must have minimum time access of $h_0 s$ for which its RG can generate one residual signal during a communication sequence. This value can be considered as minimum size of access time slot for this subsystem ($ts_{i,min}$). If in one communication sequence, access time slots of each subsystem ($ts_i, i = \{1, 2, \ldots, p\}$) are greater than or equal to their minimum access slot time ($ts_{i,min}$) , this communication sequence can guarantee generating of minimum one residual signal for each residual generator.

**Definition 5.1.** A communication sequence is *admissible* when it guarantees generation of minimum one residual signal for all subsystems of the process.

It is only admissibility condition that must be considered for finding communication sequences offline. Communication sequences can be classified by different slot time which assigns to each subsystem. After finding all admissible communication sequences, they are stored in scheduler. When the process is in running mode, by selecting one communication sequence form this list the scheduler determines when

a subsystem has ownership of a time slot to transmit its output to specific FDI module.
Computing online communication sequence based on needs of FDI modules may increase computation charge in scheduler in practical implementation. Therefore, computation requirements of scheduler is reduced by finding all admissible communication sequences offline. In addition, scheduler can select one communication sequence based on message urgency of subsystems. It may be considered as an advantage of proposed scheduling comparing to offline scheduling.

### 5.2.2 Sequence selection

It is assumed that faults do not occur simultaneously in more than one subsystem. furthermore , residuals are considered as Gaussian variables and their variances before and after fault occurrence are known and constant. So for increasing performance of diagnostic, sequence selection from ensemble of admissible sequences is based on this principle that it provides more access to the subsystem with more deviation in its residual from normal condition.
Here for abnormality (i.e. residual deviation from normal condion) classification, a Multi Hypothesis Sequential Probability Ratio Test (MSPRT) is used. Based on the class (accepted hypothesis) of residual of all RG modules, a new communication sequence is chosen in the scheduler.

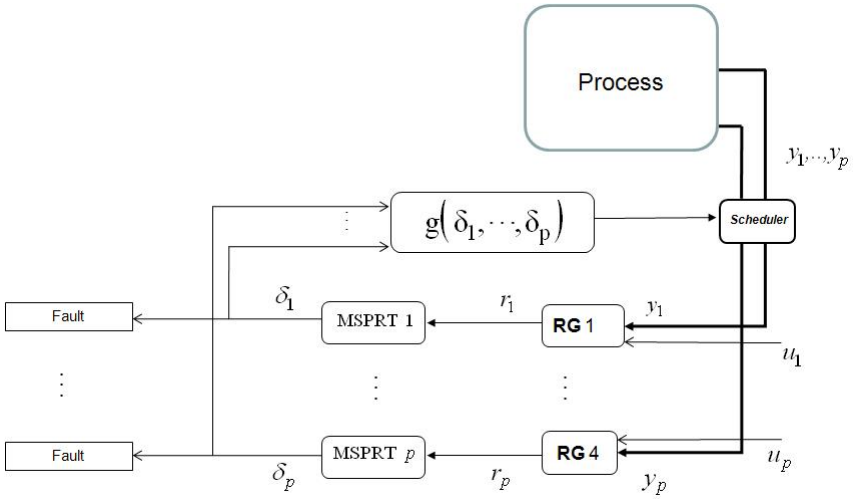



FIGURE 5.2 – message scheduling architecture

#### 5.2.2.1 Multi Sequential Probability Ratio Test(MSPRT)

In engineering applications, a statistical hypothesis testing is a method of making decisions using experimental data. A reliable decision on the underlying hypothesis is made through repeated observations. Given a cost associated to each observation, a well-known tradeoff arises between accuracy and number of iterations.
Used methods for hypothesis testing may be sequential or non-sequential. In a non-sequential method, a fixed size of samples (i.e., a fixed number of measurements) is used and the decision is made based on this whole block of samples altogether. A sequential method uses the samples one by one and the decision may be made at any time when sufficient evidence is gathered. A sequential test consists of

a stopping rule, which determines when the test is done, and a decision rule, which determines which hypothesis to choose.
Various sequential hypothesis tests have been proposed to detect the underlying hypothesis within a given degree of accuracy. The use of sequential tests for binary hypothesis testing has been well studied, and the properties of the Sequential Probability Ratio Test (SPRT) have been thoroughly investigated in many literature such as [Sie85].
In case of change detection, SPRT has been proven that the average number of required samples (or observations) is much less than the fixed sample size test for achieving the desired false alarm and miss probabilities [Wal45]. Other reason for the interest in SPRT is its appropriate threshold for the test statistics that can be determined easily without a knowledge of the distribution of the measurements, while for a non-sequential test the threshold usually depends on the distribution [LW99].
The problem of sequential testing of multiple hypotheses is considerably more difficult than that of testing two hypotheses (e.g. SPRT). Works on this problem has taken two different approaches. One approach has aimed at finding an optimal multi hypothesis sequential test. A recursive solution to the optimization problem in a Bayesian setting has been obtained [Tar88]. The second approach has been to extend and generalize the SPRT to the case of more than two hypotheses without much consideration to optimality.
The Multi-hypothesis Sequential Probability Ratio Test (MSPRT) for testing a group of multi hypothesis was introduced in [BV94], and was further generalized in [DTV99, DTV00].
Let $X_1, X_2, \ldots$ be an infinite sequence of random variable and identically distributed (i.i.d) with probability density function $f$, and $H_l$ be the hypothesis that $f = f_l$ for $l = 0, 1, \ldots, M-1$ . Objective in a MSPRT test is to determine the true hypothesis with a desired accuracy as quickly as possible. The posterior probability after N observation is given by :

$$p_N^l = P(H = H_l | X_1, \ldots, X_N) \tag{5.9}$$

A MSPRT Test can be described as follows. For given $M$ hypothesis with their prior probabilities $\pi_l$ , $l = 0, 1, \ldots, M-1$, the null hypothesis $H_0$ and $H_{M-1}$ correspond to the fault-free case and faulty case, respectively and the alternative hypotheses $H_v, v \in 1, \ldots, M-2$ correspond to abnormality modes. The posterior probability after $N$ generated residual in one RG is computed by :

$$p_N^l = \frac{\pi_l \prod_{k'=1}^{N} f_l(r_{k'})}{\sum_{j=1}^{M-1} \pi_j (\prod_{k'=1}^{N} f_j(r_{k'}))} \tag{5.10}$$

Where $f_l$ is the probability density function of the residual, conditioned on the hypothesis $l$ when it is assumed that $f_j \neq f_l$ for all $j \neq l$ and $r_{k'}$ is generated residual at instant $k'$.
The stopping time $N_A$ and final decision $\delta$ in MSPRT test can be described as follows :

$$\begin{aligned} N_A &= \text{first } N > 1 \text{ such that } p_N^l > \tfrac{1}{1+A_l} \text{ for at least one } l \\ \delta &= \arg\ \max_j \left( \pi_j \prod_{k'=1}^{N} f_j(r_{k'}) \right) \end{aligned} \tag{5.11}$$

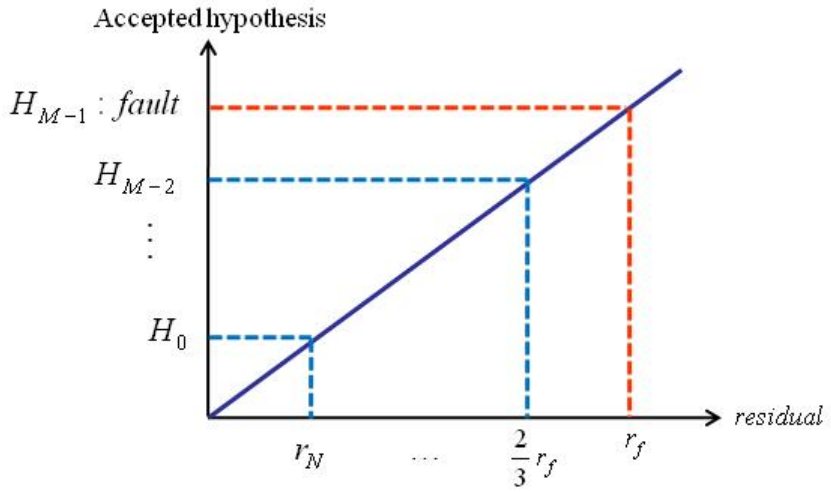



FIGURE 5.3 – mean of residual and its hypothesis in MSPRT test

For given frequentist error probabilities $\alpha_0, \ldots, \alpha_{M-1}$ (accept a given hypothesis wrongly) threshold parameter $A_l$ is found by :

$$A_l = \frac{\alpha_l}{\pi_l \gamma_l} \tag{5.12}$$

Where $\gamma_l$ is a constant function of $f_l$[BV94].
Concerning the dynamic needs associated to the flow of the subsystem outputs, it is important to specify how they can be expressed and how their expression can be translated into access time slot. For each module of RG a module of MSPRT test will be designed. Architecture of message scheduling is shown in Figure 5.2 Accepted hypothesis of each MSPRT test can be used as indicator of class of residual deviation from normal condition. It indicates if there is deviation in residual from normal condition and size of this deviation. Residual classification can be done by using a linear function. Mean of residual and its corresponding hypothesis was illustrated in Figure5.3. Where $r_f$ and $r_N$ denote residual means for accepting hypothesis $H_{M-1}$ and $H_0$, respectively.
Based on class of each residual (accepted hypothesis in each MSPRT test, $\delta_i$) necessary access time slot of subsystems can be computed in scheduler then scheduler selects one communication sequence that have these access time slots. Necessary access time slot for $i-th$ subsystem will be calculated by :

$$ts_i = \begin{cases} ts_{i,min} & \text{if } \dfrac{L_{seq}}{\sum_{j=1}^{p} \delta_j + 1}(\delta_i + 1) < ts_{i,min} \\ ts_{i,max} & \text{if } \dfrac{L_{seq}}{\sum_{j=1}^{p} \delta_j + 1}(\delta_i + 1) > ts_{i,max} \\ \dfrac{L_{seq}}{\sum_{j=1}^{p} \delta_j + 1}(\delta_j + 1) & \text{otherwise} \end{cases} \tag{5.13}$$

Where $ts_i$ is necessary access time slot of $i-th$ subsystem and $L_{seq}$ is length of communication sequence. At the end of one sequence, based on last decision of MSPRT tests a necessary time slot for each subsystem is calculated in scheduler. Then with considering necessary slot time of all subsystems, one sequence from list of admissible sequences is chosen as next communication sequence.

### 5.2.3 Fault detection

If decision of each MSPRT test is accepting hypothesis $H_{M-1}$, it means that residual of corresponding MSPRT test exceeded the threshold of fault free condition. Depend on fault tolerant control strategy, different strategies can be used for assigning access time slot to faulty subsystem. For example, if fault tolerant control system needs fault information after fault occurrence, faulty subsystem must have maximum time slot to send its measurements to FD module.
In next chapter, 6.3, implementation of this strategy on Drone application will be discussed.

### 5.2.4 Example

A process with two independent subsystems with the following state-space is considered.

$$A = \begin{pmatrix} 0.84 & 0 \\ 0 & 0.6 \end{pmatrix}, B = \begin{pmatrix} 0.39 & 0 \\ 0 & 1.79 \end{pmatrix}, C = I_2 \tag{5.14}$$

Both subsystems used a sampling time of $h = 1ms$. For sake of simplicity, we assume that only one subsystem can have access to the network at each instant $k$. As illustrated in Figure 5.2, for each subsystem one residual generator based on parity space approach was developed. The outputs of RGs were connected to two modules of MSPRT test. The MSPRT tests classify residuals into four different classes. First class, $H_0$, corresponds to normal condition and last one is faulty class. A set of admissible communication sequences with length of $12ms$ was specified offline. These communication sequences were stored in scheduler and during of system running one of them based on output (decision) of all MSPRT tests is selected online.
The first simulation was realized in fault free condition when residuals are near to zero. Residual of RGs were shown in Figure 5.4. Accepted hypotheses (decisions) of MSPRT tests were illustrated in Figure 5.5. In this figure value $'-1'$ indicates *no decision* in corresponding MSPRT test that means the test needs more data to make a decision or accept one hypothesis. Other values denote to accepted hypothesis.
In this simulation hypothesis $H_0$ was accepted in all MSPRT tests then necessary access time slot for each subsystem was computed in scheduler. All residual are in normal condition, so each subsystem can have half of time length of communication sequence to send its measurements. If there is not problem in the process , this sequence will be repeated periodically. Therefore, it is similar to works that used static periodic communication sequence.
In the next simulation, an abnormality was introduced to first subsystem at $t = 69ms$. As a consequence, in first MSPRT test a higher hypothesis ($H_2$) comparing to pervious simulation was accepted whereas second subsystem remained in its normal condition ,$H_0$ was accepted.(Figure 5.7). For next communication sequence last decision of each test was considered in scheduler for computing length of time slot for accessing each subsystem to medium. Second hypothesis was accepted in first MSPRT test that means first subsystem must have more access to medium than other subsystem in next communication sequence. Thus, computed time slot for the first subsystem will be $ts_1 = 9ms$ whereas in normal condition it is $ts_1 = 6ms$. A new communication sequence with respecting necessary time slots for each subsystems was selected from admissible sequences each $12ms$. New communication sequence that was started at $t = 85ms$ assigned 75% of sequence length to first subsystem and 25% to second one(Fig.5.8 ).
When $t = 110ms$ an other abnormality was occurred in second subsystem and its residual diverted from normal condition. (Figure 5.6). Then, first hypothesis was accepted at $t = 117s$. As a consequence, in next sequence selection at $t = 120s$ a high access time was assigned to this subsystem. In new communication sequence, the second subsystem had 66.6% of sequence length an other one had 33.4% to send their measurements to RG modules. (Figure 5.9 and Figure 5.8)

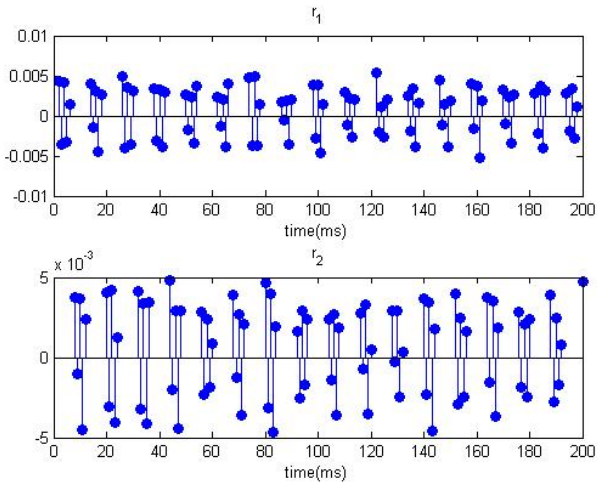



FIGURE 5.4 – Generated residual in normal condition

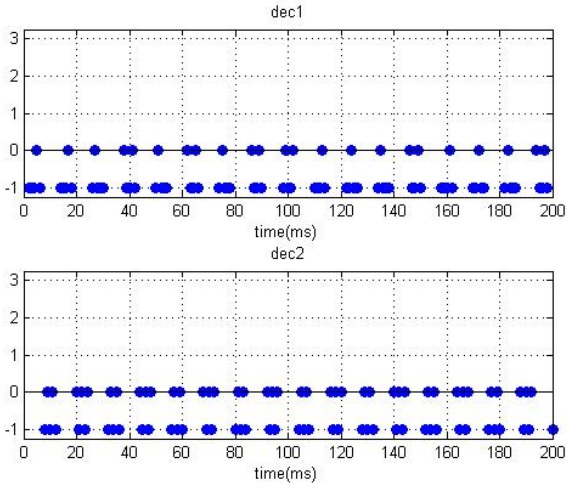



FIGURE 5.5 – Accepted hypothesis of each MSPRT test in normal condition

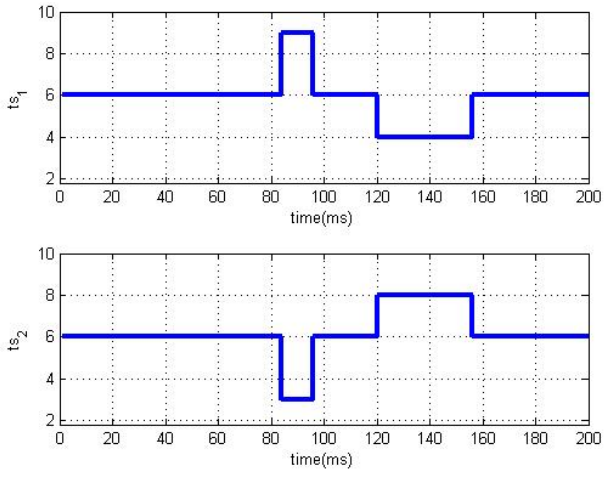



FIGURE 5.9 – Necessary time slot for each subsystem

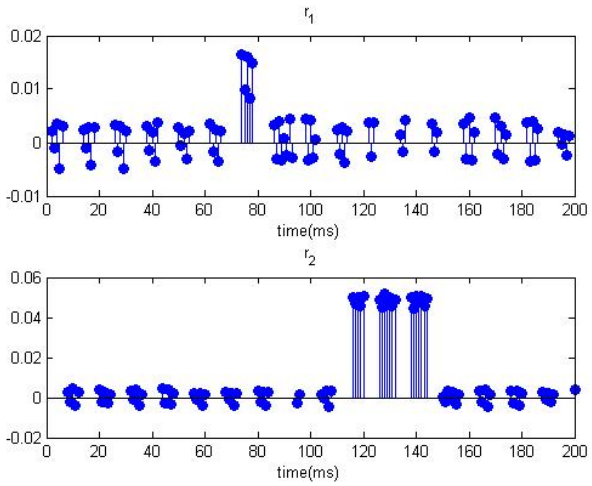



FIGURE 5.6 – Generated residual

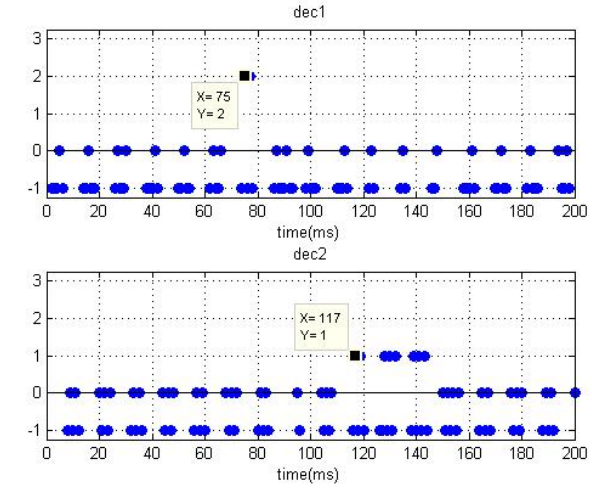



FIGURE 5.7 – Accepted hypothesis of each MSPRT test

In third simulation, a fault was introduced in the first subsystem at $t = 67ms$. By using offline periodic scheduling, when both subsystem have half of communication sequence length, fault was detected at $t = 114ms$. But by using semi-online scheduling which proposed here, fault will be detected at $t = 104ms$. Necessary time slot of the first subsystem and accepted hypotheses correspond to first MSPRT test in case of semi-online scheduling was illustrated in Figure 5.10.

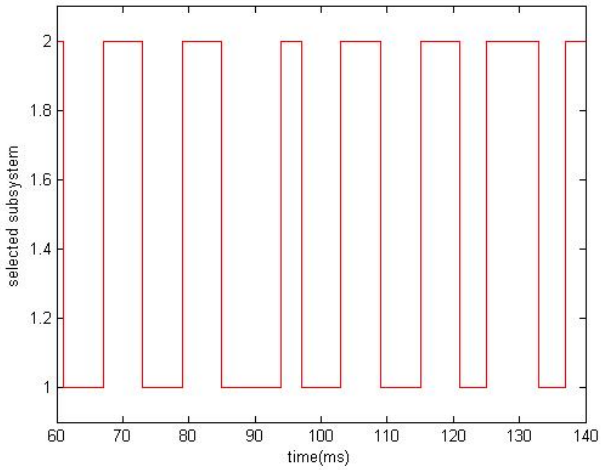



FIGURE 5.8 – Communication sequence

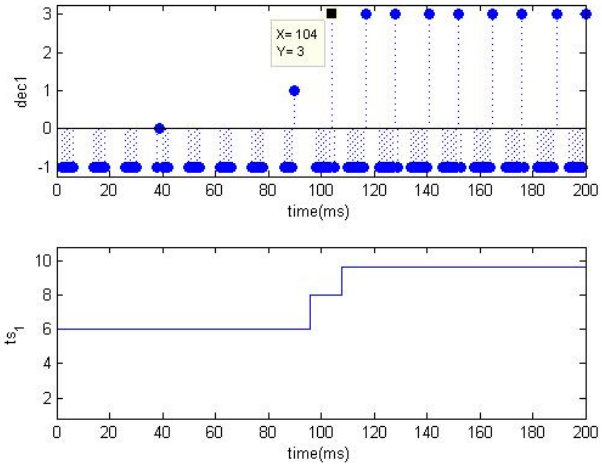



FIGURE 5.10 – Accepted hypotheses and necessary time slot of first subsystem

## 5.3 Observer-based residual generation and semi-online scheduling

In section 5.2 problem of fault detection with dynamic scheduling was studied. For fault detection, a module of residual generator based on parity space approach was proposed and one communication sequence was called *admissible* if it guaranteed generation of minimum one residual signal for each RG during the communication sequence. In this section for process described in (5.2) observer-based RG approach is proposed. Then, condition of admissibility of one communication sequence based on stability of residuals in the absence of fault will be studied.
Without loss of generality linear model of $i-th$ subsystem defined by :

$$\sum_i : \begin{cases} \dot{x}_i(t) = A_i x_i(t) + B_i u_i(t) + E_{f,i} f_i(t) \\ y_i(t) = C_i x_i(t) \end{cases} \tag{5.15}$$

Where $f_i \in \mathbb{R}^{q_i}$represents fault vectors to be identified. Similar to 5.2 a residual generator is developed for each subsystem as follows ( for simplicity index $i$ is omitted).

$$O_i : \begin{cases} \dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Bu(t) + Ly(t) \\ \hat{y}(t) = C\hat{x}(t) \\ r(t) = T(y(t) - \hat{y}(t)) \end{cases} \tag{5.16}$$

Where $L \in \mathbb{R}^{n,m}$ and $T \in \mathbb{R}^{q,m}$ are matrices that are designed in order to fulfill fault detection and isolation requirement. There are various approaches (e.g. [CP99, Din08]) to determine the gain matrices $L$ and $T$. One observer is updated when scheduler allows its subsystem to access to the network. Because of constraints on communication all observer can not be updated in each sampling time $h$. So, first step is finding ratio of observer update $\xi$ during one communication sequence. All observers are identical therefore, it is possible to solve the problem of residual stability for one observer and extend it to all observers. It is assumed that observer update times are $t_k$ where $t_k - t_{k-1} = h$ and control input is kept constant during period $t \in [t_{k-1}, t_k^-)$. Also it is considered that $\hat{x}(t_k^-) = \lim_{\delta t \to 0} x(t_k - \delta t)$.
From viewpoint of the FDI module the system behavior is described by :

$$x(t) = \Phi_\Sigma(t_{k-1}, t)x(t_{k-1}) + \Gamma(t_{k-1}, t)u(t_{k-1}) + \int_{t_{k-1}}^{t} \Phi_\Sigma(t, \tau)E_f f(\tau) \tag{5.17}$$

Where $\Phi_\Sigma(t_{k-1}, t) = e^{A(t-t_{k-1})}$ and $\Gamma(t_{k-1}, t) = \int_{t_{k-1}}^{t} \Phi_\Sigma(t, \tau)Bu(\tau)d\tau$. Between two update observer runs in open loop condition according to :

$$O : \begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) & t \in [t_{k-1}, t_k^-) \\ \hat{y}(t) = C\hat{x}(t) \end{cases} \tag{5.18}$$

Estimated state can be described by

$$\hat{x}(t) = \Phi_\Sigma(t_{k-1}, t)\hat{x}(t_{k-1}) + \Gamma(t_{k-1}, t)u(t_{k-1}) \quad t \in [t_{k-1}, t_k^-) \tag{5.19}$$

and after updating at instate $t_k$

$$\begin{array}{rcl} \hat{x}(t_k) & = & \hat{x}(t_k^-) + L(y(t_k) - C\hat{x}(t_k^-)) \\ & = & (I - LC)\hat{x}(t_k^-) + LCx(t_k) \end{array} \tag{5.20}$$

From 5.17 and 5.19, on the interval $t \in [t_{k-1}, t_k^-)$ the estimation error $\varepsilon(t) = x(t) - \hat{x}_(t)$ propagate as :

$$\varepsilon(t_k^-) = \Phi_\Sigma(t_{k-1}, t_k)\varepsilon(t_{k-1}) + \int_{t_{k-1}}^{t_k} \Phi_\Sigma(t_{k-1}, \tau)Ef(\tau)d\tau \tag{5.21}$$

then, with 5.20, after output updating, the state estimation error satisfy to :

$$\begin{array}{rcl} \varepsilon(t_k) & = & x(t_k) - \hat{x}(t_k) \\ & = & (I - LC)\varepsilon(t_k^-) \\ & = & (I - LC)\left[\Phi_\Sigma(t_{k-1}, t_k)\varepsilon(t_{k-1}) + \int_{t_{k-1}}^{t_k} \Phi_\Sigma(t_{k-1}, \tau)Ef(\tau)d(\tau)\right] \end{array} \tag{5.22}$$

The residual generator must be stable when system is operating under no fault condition. if fault free condition is assumed $f(t) \stackrel{\triangle}{=} 0$ $t \in [t_0, t_k]$ and $\varepsilon(t_0)$ is initial estimation error then

$$\begin{array}{rcl} \varepsilon(t_k) & = & (I - LC)e^{A(t_k - t_{k-1})}\varepsilon(t_{k-1}) \\ & = & (I - LC)e^{A(t_k - t_{k-1})}(I - LC)e^{A(t_{k-1} - t_{k-2})}\varepsilon(t_{k-2}) \end{array}$$

It is suppose that in the interval $t \in [t_0, t_k]$ the number of observer updates is $l$, therefore

$$\begin{aligned} \varepsilon(t_k) &= (I-LC)^l e^{Akh}\varepsilon(t_0) \\ &= \left((I-LC)e^{Ah}\right)^l e^{A(k-l)h}\varepsilon(t_0) \\ &= \phi_0(t_k,t_0)\varepsilon(t_0) \end{aligned} \tag{5.23}$$

and on the interval $t \in [t_0, t_{k+1}^-]$ :

$$\varepsilon(t) = \Phi_\Sigma(t,t_k)\phi_0(t_k,t_0)\varepsilon(t_0) \tag{5.24}$$

If system described by equation 5.17 is stable, that is to say eigenvalues of $\Phi_\Sigma$ are inside the unit circle, the state observer given by 5.18 and 5.20 is exponentially stable if and only if the eigenvalues of $(I-LC)e^{Ah}$ are inside the unit circle. Now, the case when system 5.17 is not stable will be studied. A system described by equation $\varepsilon(t) = \Lambda(t,t_0)\varepsilon(t_0)$ is stable if there exist positive constants $c_1, c_2$ such that :

$$\|\Lambda(t,t_0)\| \leq c_1 e^{-c_2(t-t_0)} \quad \text{for all } t > t_0$$

Taking the norm of equation described in 5.24 :

$$\|\varepsilon(t)\| \leq \|\Phi_\Sigma(t,t_k)\| \cdot \left\|((I-LC)e^{Ah})^l\right\| \cdot \left\|e^{A(k-l)h}\right\| \cdot \|\varepsilon(t_0)\| \tag{5.25}$$

Now lets analyze the first term on the right hand side of (5.25).

$$\|\Phi_\Sigma(t,t_k)\| = \left\|e^{A(t-t_k)}\right\| \leq 1 + (t-t_k)\bar{\sigma}(A) + \tfrac{(t-t_k)^2}{2!}\bar{\sigma}(A)^2 \cdots \leq e^{\bar{\sigma}(A)(t-t_k) \leq e^{\bar{\sigma}(A)h}} \tag{5.26}$$

where $\bar{\sigma}(A)$ is largest singular value of $A$. In general this term can always be bounded since the time difference $t-t_k$ is always smaller than $h$. In the other words even when $A$ has eigenvalues with positive real part ,$\left\|e^{A(t-t_k)}\right\|$ can only grow a certain amount. This growth is completely independent of $k$.
Therefore, for a given system dynamic, the stability condition depend on gain $L$ of observer and its update ratio $\xi$ over $k$ number of samples.

$$\xi = \frac{l}{k} \tag{5.27}$$

For $i-th$ subsystem observer gain $L_i, T_i$ and its update ratio $\xi_i$ can be found offline then a communication sequence is admissible if it guarantees minimum ratio $\xi_i$ of observer update for $i-th$ subsystem when it is used in scheduler.

## 5.4 Online scheduling in CAN network by means of hybrid priority

Hybrid priority was presented in [JM07] and [JMC08] and it was used in [CBL09] for message scheduling in a mini helicopter application. It was inspired by Mixed Traffic Scheduling scheme [ZS97] and [KK00]. In the following a brief introduction of static priority in CAN network and Hybrid priority will be presented then hybrid priority scheme for message scheduling with diagnostic objective will be studied.

### 5.4.1 CAN network

Controller Area Network (CAN) [Gmb91] was initially created by German automotive system supplier *Robert Bosch* in the mid 1980s for automotive applications as a method for enabling robust serial communication. This serial communication protocol efficiently supports distributed realtime control with a very high level of security. The CAN protocol is optimized for short messages [LMT01].
CAN protocol is a message-based protocol, not an address based protocol. All the sent messages are broadcasted to all the other network nodes. In a CAN-based network, data are transmitted and received using Message Frames that carry data from a transmitting node to one or more receiving nodes. Transmitted data do not necessarily contain addresses of either the source or the destination of the message. Instead, each message is labeled by an identifier that is unique throughout the network. All other nodes on the network receive the message which can be accepted or rejected.
CAN is a deterministic protocol. Consequently, it is possible to compute an upper bound over messages response times. The computation of the worst-case response time of fixed priority messages have been studied in [TB94] and [TBW95]. The evaluation of a Rate-monotonic(RM) priority assignment policy was also addressed in these references. The use of Earliest deadline first(EDF) scheduling in CAN networks was studied in [Nat00]. However, slow data rate (maximum of 1 *Mbits/s*) is major disadvantages of CAN comparing to the other networks. In addition it is not suitable for transmission of messages of large data sizes, although it does support fragmentation of data that is more than 8 bytes.
CAN protocol defines four different types of messages (or Frames). The first and most common type of frame is a Data Frame. This is used when a node transmits information to any or all other nodes in the system. Second is a Remote Frame, which is basically a Data Frame with the RTR bit set to signify it is a Remote Transmit Request. There are two other frame types for handling errors. Error Frames are generated by nodes that detect any one of the many protocol errors defined by CAN. Overload errors are generated by nodes that require more time to process messages already received. A data frame is composed of seven different bit fields :

- Start of frame
- Arbitration field
- Control field
- Data field
- CRC field
- ACK field
- End of frame

The Arbitration field consists of the Identifier field (ID field) that its standard length is 11 bits.
CAN is a local area network and it uses CSMA/CA scheme for the scheduling of the messages. The CSMA/CA stands for Carrier Sense Multiple Access with collision avoidance. In this access model, each message is characterized by a unique priority. Since the shared communication medium can transmit only one message simultaneously, each node that wishes to transmit a message must initially check whether the network is free (by sensing the network to find whether or not a carrier signal is being transmitted). If the network is free, then the node can start transmitting. However, it is possible that other nodes start transmitting at the same time, because they have detected at the same time that the communication medium has become free. In this situation, the transmission of the highest priority message is continued ; the other messages with lowest priorities are discarded. By this way, collisions are avoided.
Concerning the frame scheduling in the MAC layer of CAN, it is based on priorities which appears in ID filed of the frames. It is done by the comparison bit by bit of the field ID (it starts from the Most Significant Bit, MSB). In CAN the bit $'0'$ is a dominant bit and the bit $'1'$ is a recessive bit. The lower the numerical value of the CAN identifier ID will be resulted the higher the priority. CAN bus behaves like a AND gate. Identifier field is located at the beginning of the frame. When a collision occurs, different

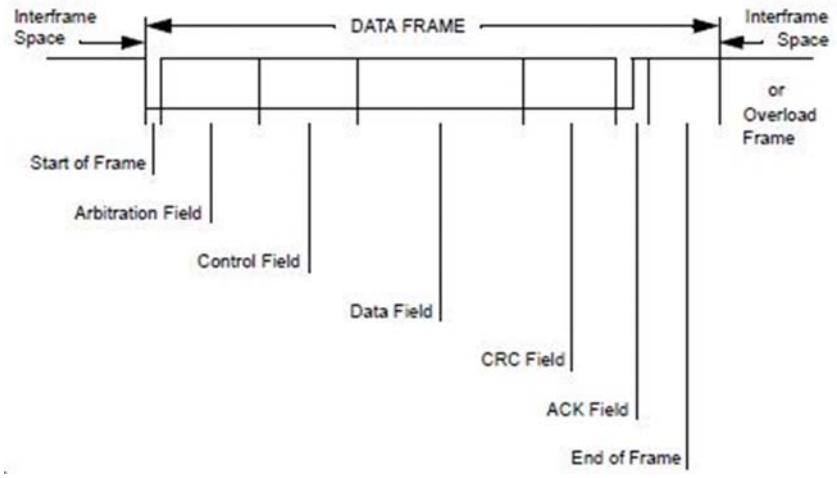



FIGURE 5.11 – CAN data frame [Gmb91]

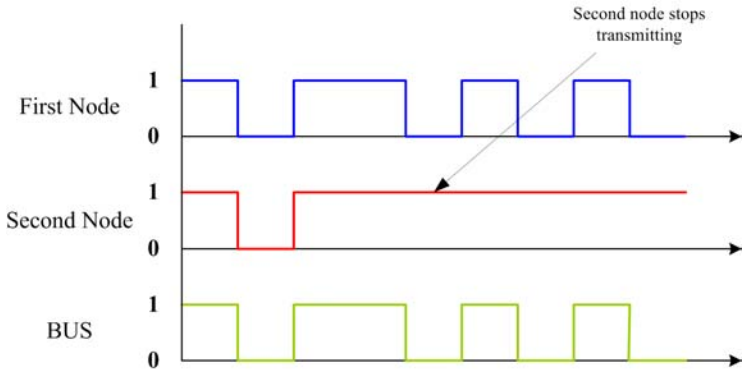



FIGURE 5.12 – Illustration of the bus arbitration in CAN networks

nodes directly compare the identifiers of their messages with logical level at the bus. A node that detects a dominant level $'0'$ on the bus whereas it has sent a $'1'$, it knows that it sent a message with lower priority than another message. Therefore, it stops transmitting the message. Fig. 5.12 illustrates bus arbitration for two messages which were sent at same time by two nodes. Second node stopped transmitting after detecting that it has a lower priority than first node.

### 5.4.2 Hybrid priority

There are some general observations on the CAN arbitration. Firstly, a message with a smaller ID field value is a higher priority message. Secondly, if network is shared by some applications the highest priority message transmits without disturbance (since all other stations must wait until bus is ideal).
Usually a static priority is used in CAN. It means that ID field of each message is specified in advance. When network is dedicated to one control application (i.e. dedicated network), the worst-case delay can be calculated easily. Then, by using an enough good controller, control performance is approximately same as without network case. In the other words, the network has no influence on the performance of the control [JMCP]. When the network is shared between more than one application or an external flow

pass through the network, priority value of each flow plays an important role in control performance of the process. In a control loop that contains controller flow, measurement flow and external flow, with shared network the following important results were shown in [JMCP] :

- Control performance is not influenced by external flow if the priority of external flow is lower than the priority associated to measurement flow and controller flow.
- If the priority associated to controller flow is lower than priority associated to external flow and measurement flow, the external flow may influence the performance of the process control and in a particular case when network load induced by external flow is very high, the controller may not stabilize any more the system.

If there is a high load in the communication and control application has not the highest priority , an acceptable control performance is not achievable. However it is not always possible to give the highest priority to control application. Furthermore, in case of distributed system when more than one process control shared the network, one of them will not have obviously the highest priority.
To overcome these difficulties, a possible way is to take into account urgency or needs of each message at each moment. In this way, scheduling of messages is based on transmission urgency of the messages. In hybrid priority the identifier field of a frame is dividing into two levels : bit combination of first level (Least significant bits) represent static priority of flow which is constant value. Second level (Most significant bits) specifies dynamic priority which depends on transmission urgency of the message. Second level may be constant or variable during system running. By this way, this is possible to specify a static priority offline and in function of urgency of the message a dynamic priority can be assigned to messages. Fig. 5.13 shows structure of ID field in hybrid priority. The scheduling is executed by : first, comparing the second level of identifier ID and if all flows have identical urgency, comparing the first level will be done. [JM07] showed that when a NCS contains a feedback controller and a external flow, hybrid prio-

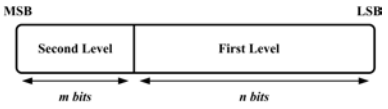



FIGURE 5.13 – Frame identifier structure in hybrid priority [JM07]

rity can guarantee performance of controller whatever the network load. However, hybrid priority brings some limitation in number of flow which must be scheduled. When a static priority is used priorities, $2^{m+n}$ different priorities can be specified. Whereas, in hybrid priority a hierarchy structure is considered and consequently, number of scheduled flows is smaller than static priority.

### 5.4.3 Hybrid priority with diagnostic objective

Here, the process described in 5.1 with $p$ subsystems is considered. Subsystems transmit their measurements to RGs via a CAN network. IF a static priority is used, when a collision is occurred, a subsystem with lower identifier ID (higher priority) can transmit its measurements and other subsystems must wait for ideal bus condition. Problem will be occurred when a low priority subsystem is in faulty condition and for fault detecting an enough number of measurements is needed. Other problem that may be occurred is delay in messages with low priority which may cause a false alarm [JMS+10]. As a solution of these problems, the structure described by Figure 5.1 with hybrid priority will be proposed.
In the previous cases, a slot time for each subsystem was assigned and during this slot time subsystems were allowed to send their measurements to RGs. Length of slot time depended on the size of derivation in residual of each subsystem comparing to other subsystems. In CAN network, message of subsystems

which their residuals are far from normal condition have higher priority. Therefore messages contain measurements of these subsystems will be transmitted faster to residual generator than other subsystems which work in their normal condition.
For implementing hybrid priority, as explained before, identifier field of each message divided into two level : First level of each measurement flow is specifies offline. This is constant need of each flow that can be specified based on characterization of each subsystem. Then, the computation of dynamic priority (second level) is done in scheduler when it receives the results (accepted hypotheses) from all MSPRT tests. And, it will be re-evaluated after each update in output of MSPRT tests. Then, new dynamic priorities will be used in next measurement transmission. Unlike hybrid priority that presented in the pervious work, value of dynamic priority is a discrete variable and number of dynamic priority is equal to number of defined hypothesis in MSPRT test(i.e. $M$). As a consequence, number of static priority will be increased and more flows can be scheduled comparing to pervious methods.
In normal condition when all residuals are near to zero (or values which result accepting null hypothesis $H_0$ in MSPRT tests) dynamic priority will have minimum value ($P_{min}$, all bits equal to $'1'$). So, scheduling is based on static priority of each message. But if residual of one subsystem deviates from normal condition and higher hypothesis is accepted, dynamic priority of its flow is increased to allow faster transmission.
Maximum dynamic priority ($P_{max}$) is assigned to flow of subsystem that its residual is close to threshold of fault free condition$(H_{M-2})$ or faulty condition $(H_{M-1})$ depend on the FTC strategy of the system. Relation between class of residual and dynamic priority was shown in fig.Figure 5.14. Relation between

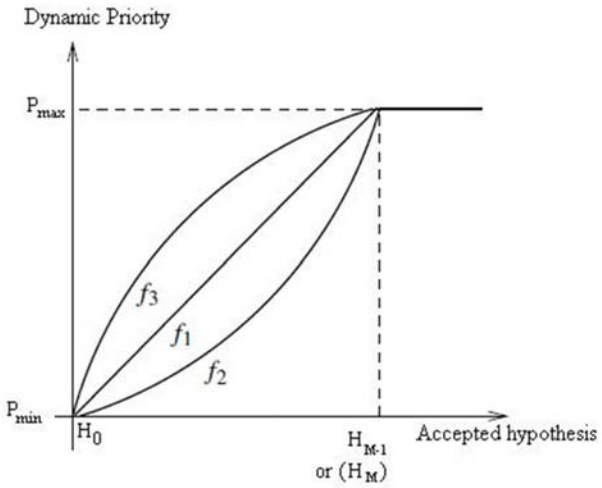



FIGURE 5.14 – Relation of Accepted hypothesis in MSPRT test and Dynamic priority

dynamic priority of flow of a subsystem and class of its residual can be expressed with three different function. one linear function $f_1$ and two non-linear function $f_2$ and $f_3$, choice of one of these function is based on characterization of the process. Fast reaction in case of residual deviation from normal condition can be obtained by using function $f_3$. In this function, by increasing class of residual from normal condition ($H_0$) dynamic priority will be increased very fast. Whereas, for function $f_2$ dynamic priority will be increased when class of residual is close to faulty condition.

## 5.5 Conclusion

In this Chapter problem of semi-online scheduling and fault detection was studied. First of all, this strategy with parity space-based residual generation was proposed. Then based on stability analysis of residual, an extension of this strategy in case of observer-based approach was studied. Finally, an adaptation of this strategy in CAN network by using hybrid priority was proposed. This scheduling algorithm, can be seen as a compromise between the advantages of the online scheduling (fast fault detection) and those of the offline scheduling (a very limited usage of computing resources). In addition, this strategy can be considered as a modular solution. Because any used part such as RG module, residual classification module, scheduler can be designed separately from others. For example, it can be used for observer-based residual generators as well as parity space residual generators. Residuals are assumed as Gaussian variables and their variances before and after fault occurrence are considered to be known and constant. By using these assumptions, a classical multi sequential hypothesis test was proposed. However, it is possible to use a appropriate multi hypothesis test when residual signals have other characteristic.

# Chapter 6

# Drone Application

In this chapter an implementation of the proposed methods in the previous chapters on a mini helicopter(drone) application will be presented. Drone can be classified as a fast dynamic system. So, it is ideal benchtest for studying effects of the network on performance of closed loop control of the system. For instance, intermitted packet dropouts, even with short duration, in command of one rotor can destabilize whole of the system (see 6.2.2). Also, constraint on the the communication has direct effects on performance of diagnostic and fault tolerant control strategy (see 6.3).
First of all, drone application will be described. Then, some useful attitude representation which are used in the system will be presented. Mechanical model of drone will be studied in 6.1.5. In 6.2 problem of packet dropout on command of the motors will be studied. Then, different strategies for compensating effects of the fault in output of the system will be proposed. These strategies can be more efficient if a fast fault detector is adopted. In the last section, An approach of semi-online message scheduling and fault detection will be studied in 6.3.

## 6.1 Drone presentation

Unmanned Aerial Vehicles (UAVs) are currently subject of a lot of research due to their usefulness in situations that require unmanned or self-piloted operations. They are used to perform intelligence, surveillance, and reconnaissance missions and have capacity to enter to environments that are dangerous to human life. UAVs have several basic advantages over manned systems including increased manoeuvrability, reduced cost, reduced radar signatures, longer endurance. They have the capacity to do a complete mission autonomously with a predefined program. They provide a wide point-of-view comparing to unmanned ground vehicles.[Aus10] and [RS08] provide good information about UAVs and their control problems.
Constraints such as weight, size and power consumption plays an important role in design of small UAVs that must be light and low cost. Then, the cooperation of many of these vehicles is the most suitable approach for many applications. A team of aerial vehicles can simultaneously collect information from multiple locations and exploit the information derived from multiple disparate points to build models that can be used to take decisions [MKBO10].
In case of fault occurrence in any subsystems or components, the UAV may be designed automatically to take corrective action and/or alert its operator. For some systems, attempts are being made to implement on-board decision-making capability.

| | Category | Maximum take off weight (Kg) | Maximum Flight Altitude (m) | Endurance (hours) | Data like rang (km) |
|---|---|---|---|---|---|
| Micro/Mini UAVs | Mico(MAV) | 0.10 | 250 | 1 | > 10 |
| | Mini | < 30 | 150-300 | < 2 | > 10 |
| Tactical UAVs | Close Rang (CR) | 150 | 3,000 | 2-4 | 10-30 |
| | Short Rang (SR) | 200 | 3,000 | 3-6 | 30-70 |
| | Medium Rang (MR) | 150 - 500 | 3,000 - 5,000 | 6-10 | 70-200 |
| | Long Rang (LR) | - | 5,000 | 6-13 | 200-500 |
| | Medium Altitude Long Endurance (MALE) | 1,000 - 1,500 | 5,000 - 8,000 | 24-48 | > 500 |
| Strategic UAVs | high altitude,long endurance (HALE) | 2,500 - 12,500 | 15,000 - 20,000 | 24 - 48 | > 2,000 |

TABLE 6.1 – UAV classification

UAV vehicles may fall into different categories based on their size or application. The European Association of Unmanned Vehicles Systems (EUROUVS) has drawn up a classification based on such parameters as flight attitude, endurance, speed and maximum take off weight (MTOW). [M.B08] presented a good state of art and current EUROUVS classification. Brief classes of UAV vehicles are presented as follows :

**Micro and mini UAVs :** It comprise the category of the smallest platforms that also fly at lower altitude(under 300 meters) with weight of under 30 kilograms. Designs for this class of device have focused on creating UAVs that can operate in urban canyons or even inside building, flying along hallways, carrying some devices, transmitters, or cameras. The development of mico-electromechanical systems (MEMS) in recent years helped overcome some constraints on performance of small and highly functional navigation robots. Rotary wing mini UAVs with capacities of Vertical take off landing (VTOL) are more predominant.

**Tactical UAVs (TUAV) :** This category includes heavier platforms at higher altitude (from 3,000 to 8,000 meter). Tactical UAVs primarily support military applications. Medium Altitude and long endurance UAVs (MALE) with endurance of 24-48 hours are important group of UAVs in this category.

**Strategic UAVs :** High attitude and long endurance(HALE) UAVs can be classified into this category. They are highly automated, with takeoffs and landings being performed automatically and their maximum attitude can reach about 20,000 meters. More details of UAV classification can be found in Table 6.1. Although standard helicopter is one of most important vehicles that are capable of rising and descending vertically from and to the ground, thus requiring no runway, However rotorcraft vehicles are considered to be more preferable for surveillance, precise delivery and some other missions requiring agility and accuracy. Quadrotor is small rotorcraft vehicle that is lifted and propelled by four rotors. It has attracted considerable interest in industrial and academic research.

The particular interest of the research community in the quadrotor design can be linked to two main advantages over comparable vertical take off and landing UAVs, such as helicopters. First, quadrotors do not require complex mechanical control linkages for rotor actuation, relying instead on fixed pitch rotors and using variation in motor speed for vehicle control. This simplifies both the design and maintenance of the vehicle. Second, by enclosing the rotors within a frame, the rotors can be protected from breaking during collisions which permit to fly in indoor environments. These added safety benefits greatly accelerate the design and test flight process by allowing testing by inexperienced pilots. Furthermore, It benefits from having very few constraints on motion and an ability to carry a "high" payload compared

to its own weight.



FIGURE 6.1 – Quadrotor benchtest

### 6.1.1 Architecture of prototype of Safe-NECS project

A quadrotor (hereafter drone) is used as prototype of Safe-NECs (safe-Networked control sytstem) project. This ANR (Agence National de la recherche) project was designed to enhance the integration between control, real-time and networking. Objective of this platform is to provide a beach test for implementation new algorithms proposed from project participants on the following points :

- Modeling the effects of scheduling and networking protocols on Quality of control(QoC) and dependability.
- Fault tolerance and online reconfiguration.
- Distributed diagnostic and decision making mechanisms.
- Network effects such as packet dropout, delay, communication constraints on performance of control and diagnostic.

To achieve these objectives, drone with several electronic module was developed in GIPSA laboratory. Each module communicates with other via a CAN network. In this beach test five main modules can be identified(Figure 6.2).

- **Measurement Module :** This module consists of an inertial measurement unit(IMU) for system positioning made of three rate gyros, three accelerometers and three magnetometers.
  This module is controlled by a PIC18F [MIC04] Microchip and it provides some measurements that are used for estimation of the attitude.
- **Controller module :** It consists of a *DsPIC33F* Micro-controller which is in charge to compute drone commands. Nonlinear observers for attitude estimating, FDI and FTC modules were implemented here.
- **Actuator module :** Control loop for commanding four motor of drone can be found here. For computing necessary voltage of motors two different strategies can be used, first : Based on commands of controller, actuator module computes necessary voltage of each motor. Second : necessary voltage of each motor is computed in controller module and is sent to actuator module.
- **Communication :** To transmit all data from different modules a CAN network is used.
- **Power supply :** This card supplies electrical energy to all other modules.

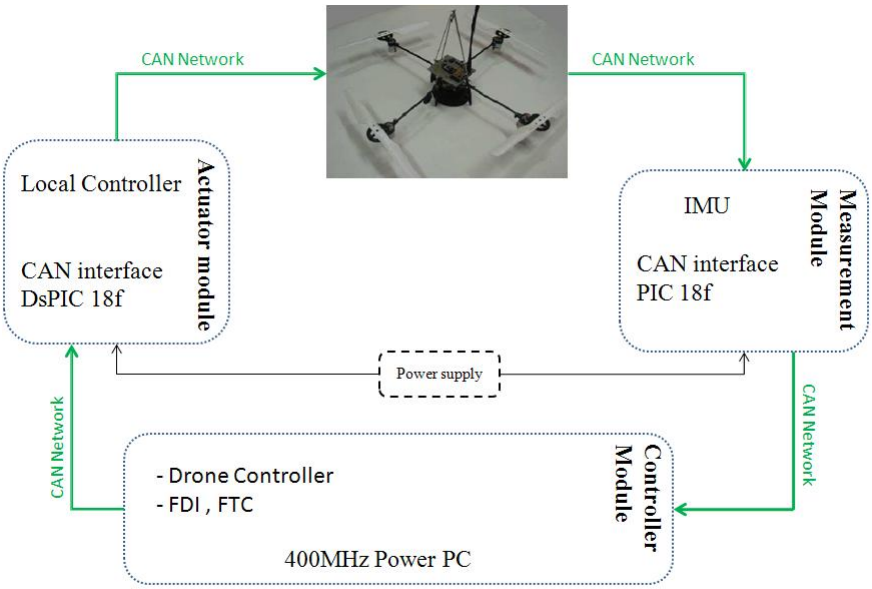



FIGURE 6.2 – Electronic architecture of drone

### 6.1.2 Drone movements

The drone is controlled by independently varying the rotation speed $\omega_i, i = 1:4$, of each electric motor which are in cross configuration. Each propeller is connected to the motor through the reduction gears. All the propellers axes of rotation are fixed and parallel. Furthermore, they have fixed-pitch blades and their air flows points downwards (to get an upward lift). Here, it is considered that the movements of drone are directly related just to propellers velocities and neither the motors nor the reduction gears are fundamental. The the force $f_i$ and relative torque $Q_i$ produced by motor $i$ are proportional to the square of the rotational speed. The total thrust, $F_T$, generated by the four motors is given by (6.1) where $b > 0$ and $k > 0$ are two parameters depending on the air density, the radius, and other factors that will be studied in 6.1.5.

$$\begin{aligned} F_T &= \sum_{i=1}^{4} f_i = b\sum_{i=1}^{4} \omega_i^2, \\ Q_i &= k\omega_i^2 \end{aligned} \tag{6.1}$$

The two pairs of actuators $(M_1, M_3)$ and $(M_2, M_4)$ as described in Figure 6.3 turn in opposite directions. First pair turn clockwise, while $M_2$ and $M_4$ rotate counter-clockwise. By varying the rotor speed, one can change the lift force and create motion(6.1). Thus, increasing or decreasing the four propeller's speeds together generates vertical motion. Changing the $M_2$ and $M_4$ propeller's speed conversely produces roll rotation coupled with lateral motion. Pitch rotation and the corresponding lateral motion, result from $M_1$ and $M_3$ propeller's speed conversely modified. Yaw rotation is more subtle, as it results from the difference in the counter-torque between each pair of propellers. In spite of the four actuators, the drone still an under-actuated and dynamically unstable system.
It is possible to classify three different flight mode for the drone :

- **Stationary flight Mode :** when all the propellers rotate at the same (hovering) speed to counterbalance the acceleration due to gravity. Thus, the quadrotor performs stationary flight and no forces or torques move it from its position($F_T = mg$).
- **Vertical flight mode :** In the absence of disturbances, the total thrust $F_T$ is always vertical. When $F_T$ is higher than weight of drone($F_T > mg$), it goes up while smaller total thrust than drone weight

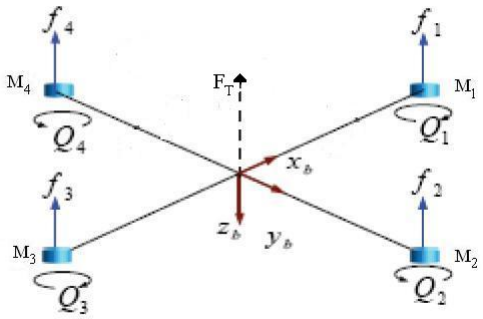



FIGURE 6.3 – produced force and reactive torque in drone

results going down movement in drone.

– **Horizontal flight Mode :** when total thrust is constant and equal to weight of drone and it moves in direction of $x_n$

The different movements of drone is illustrated in Figure 6.4. In this figure the arrow width is proportional to propeller rotational speed.

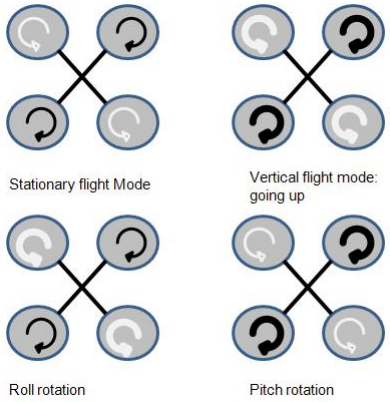



FIGURE 6.4 – drone concept motion

### 6.1.3 Attitude representation

When analyzing the motion of one object in 6 Degree of Freedom(DOF) it is convenient to define two coordinate frames as indicated in Figure 6.5. The moving coordinate frame $B(x_b, y_b, z_b)$ is conveniently fixed to the vehicle and is called the body-fixed reference frame. The origin $O$ of the body frame is usually chosen to coincide with center of gravity when center of gravity is in the principal plane of symmetry or at any other convenient point if this is not the case. The motion of body frame is described relative to inertial reference frame. Normally, an Earth-fixed reference frame $R(x_n, y_n, z_n)$ with origin of $o_n$ considered as inertial frame. The North-East-Down (NED) coordinate system is the most common inertial frame in aeronautical applications. It is local geographic coordinate system fixed to the Earth. This is usually determined by a tangent plane attached to the geodetic reference ellipsoid (WGS84) at

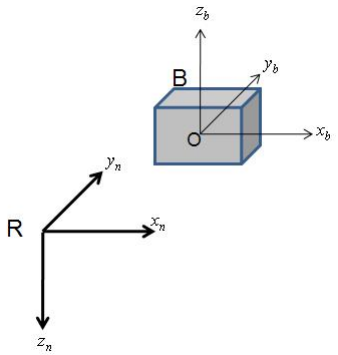



FIGURE 6.5 – Body-fixed and inertial frame

a point of interest, which determines the origin of the system on $o_n$. The positive unit vector $\vec{x_n}$ points towards the true North, $\vec{y_n}$ towards the East, and $\vec{z_n}$ points towards the interior of the earth perpendicular to the reference ellipsoid completing the dextral orthogonal system [PF07a].
The position and orientation of object are described relative to the inertial reference frame while the linear and angular velocities of object are expressed in body-fixed coordinate system.

**Definition 6.1.** A motion of a rigid body or reference frame $B$ relative to a rigid body or reference frame R is called *simple rotation* of $B$ in $R$ if there exists a line $L$, called an axis of rotation, whose orientation relative to both $R$ and $B$ remains unaltered throughout the movement [Fos94].

Let $\vec{b}$ and $\vec{r}$ be two vectors fixed in $B$ and $R$. Hence, the vector $\vec{b}$ can be expressed in term of vector $\vec{r}$ , the unit vector $\vec{e} = [e_1\ e_2\ e_3]^T$ parallel to the axis of rotation $L$ which $B$ is rotates about and $\beta$ the angle frame $B$ is rotated. The rotation is described by

$$\vec{b} = \cos\beta\vec{r} + (1-\cos\beta)\vec{e}\vec{e}^T\vec{r} - \sin\beta\vec{e}\times\vec{r} \tag{6.2}$$

Consequently, the rotation sequence from $\vec{b}$ to $\vec{r}$ can be expressed as :

$$\vec{b} = C\vec{r} \tag{6.3}$$

matrix $C$ is an operator taking a fixed vector $\vec{r}$ and rotating to a new vector $\vec{b}$ and it is called rotation matrix. Following expression for $C$ can be obtained From 6.2 :

$$C = \cos\beta I_3 + (1-\cos\beta)\vec{e}\vec{e}^T - \sin\beta[\vec{e}^\times] \tag{6.4}$$

where $I_3$ is the $3\times 3$ identity matrix and $[\vec{\xi}^\times]$ is a skew-symmetric matrix such that

$$[\vec{\xi}^\times] = \begin{pmatrix}\xi_1\\ \xi_2\\ \xi_3\end{pmatrix} = \begin{pmatrix}0 & \xi_3 & -\xi_2\\ -\xi_3 & 0 & \xi_1\\ \xi_2 & -\xi_1 & 0\end{pmatrix} \tag{6.5}$$

The set of all $3\times 3$ skew-symmetric matrices is denoted by $SS(3)$ while the set of all $3\times 3$ rotation matrix is usually referred to by the symbol $SO(3)$.

Matrix $C \in SO(3)$ can be interpreted as a *coordinate transformation matrix* that gives the orientation of a transformed coordinate frame with respect to inertial coordinate frame. This interpretation is particulary used in guidance and control application where we are concerned with motion variables in inertial and body reference frames.
Expanding (6.4) yield the following expressions for the matric elements $C_{ij}$

$$\begin{aligned} C_{11} &= (1-\cos\beta)e_1^2 + \cos\beta \\ C_{22} &= (1-\cos\beta)e_2^2 + \cos\beta \\ C_{33} &= (1-\cos\beta)e_3^2 + \cos\beta \\ C_{12} &= (1-\cos\beta)e_1e_2 + e_3\sin\beta \\ C_{21} &= (1-\cos\beta)e_2e_1 - e_3\sin\beta \\ C_{23} &= (1-\cos\beta)e_2e_3 + e_1\sin\beta \\ C_{32} &= (1-\cos\beta)e_3e_2 - e_1\sin\beta \\ C_{31} &= (1-\cos\beta)e_3e_1 + e_2\sin\beta \\ C_{13} &= (1-\cos\beta)e_1e_3 - e_2\sin\beta \end{aligned} \tag{6.6}$$

A coordinate transformation matrix $C$ satisfies :

$$CC^T = C^TC = I \tag{6.7}$$

which implies that $C$ is orthogonal. As a consequence, the inverse coordinate transformation matrix (rotation matrix) can be computed as : $C^{-1} = C^T$.
Based on one application different attitude representation may be used and each of them has its own advantages and disadvantages. [Shu93] gave a survey of the attitude representation and the relations between the various representation of attitude. Two attitude representation in our application were used that will be detailed briefly in the following sections.

#### 6.1.3.1 Euler angles

One of the simplest way to describe one rotation in 3-dimensional is using Euler angles. They are a specification of a rotation obtained by applying three consecutive principal rotations. For this, the rotation of a coordinate system $B$ relative to $R$ can be expressed by three consecutive rotations about the main axes that take $R$ into $B$.
It is possible to specify the Euler Angles and the axes of sequential rotation using rotation such as $\psi, \theta, \phi$, which denotes a rotation of $(OXYZ)$ by angles $\psi$ about $OZ$, resulting in the intermediate orientation,$(OX'Y'Z')$, followed by a rotation of $\theta$ about $OY'$, resulting in $(OX''Y''Z'')$, and then a final rotation by angle $\phi$ about $OX''$, to produce new orientation $(OX'''Y'''Z''')$(Figure 6.6). There are twelve distinct ways to select a sequence of three principal axes and apply the principal rotations and each triplet of rotated angles is called a set of *Euler angles*. The set most commonly used in navigation is that of *roll*, *pitch* and *yaw*(Figure6.7), which corresponds to the rotations performed in the following order [PF07b] :

- Rotation $\psi$ about the $z$-axis of $R$ (yaw angle)
- Rotation $\theta$ about the $y$-axis of $R$ (pitch angle)
- Rotation $\phi$ about the $x$-axis of $R$ (roll angle)

The positive angle convention corresponds to a right-handed screw advancing in the positive direction of the axis of rotation. The rotation matrix is written as :

$$C = C_{z,\psi}C_{y,\theta}C_{x,\phi} \tag{6.8}$$

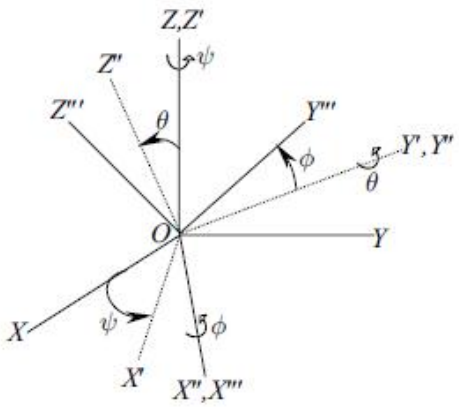



FIGURE 6.6 – The Euler Angles [Fos94]

where

$$C_{x,\phi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \sin\phi \end{pmatrix}$$

$$C_{y,\theta} = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{6.9}$$

$$C_{z,\psi} = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

After multiplication

$$C = \begin{pmatrix} c\psi c\theta & -s\psi c\theta + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{pmatrix} \tag{6.10}$$

where $c = \cos(.)$ and $s = \sin(.)$.
The Euler Angles are not unique. In addition, there are certain orientation for which the Euler angles cannot be determined at all from rotation matrix ,$C$. In such case, the Euler angle representation is said to be *singular*, and become useless (e.g. $\theta = \pm\frac{\pi}{2}$. Thus, a single Euler angle representation cannot be utilized when an arbitrary orientation is possible. For avoiding problem of singularities, that appears in classical angular representation and efficacy in computational, Quaternion representation is used[Tew07].

#### 6.1.3.2 Quaternion

An alternative to the Euler angles representation is a four-parameter method based on unit quaternion(or Euler symmetric parameters). A quaternion is a special set composed of four mutually dependent

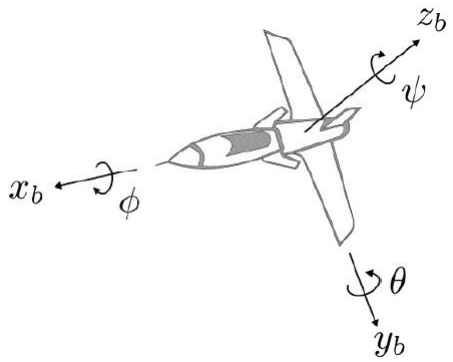



FIGURE 6.7 – Roll, Pitch and yaw angles[GC08]

scalar parameters, $q_1, q_2, q_3, q_4$, such that the first three form a vector, called *vector quaternion* ,

$$\vec{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{6.11}$$

and the fourth, $q_4$, represents the *scaler part*. Consequently, a quaternion $q$ may be viewed as a linear combination of scalar $q_4$ and a vector $\vec{q}$,(6.12). By applying quaternion, it is possible to describe the motion of the body-fixed reference frame relative to the inertial frame. It can be derived from Euler axis, $\vec{e}$ and principal rotation angle, $\beta$ as follows (NOTE : In the literature, the order is sometimes reversed $q = (q_4, \vec{q})^T$) :

$$q = \begin{pmatrix} \vec{q} \\ q_4 \end{pmatrix} = \begin{pmatrix} \vec{e} \sin \frac{\beta}{2} \\ \cos \frac{\beta}{2} \end{pmatrix} \in \mathbb{H} \tag{6.12}$$

where

$$\mathbb{H} = \{q | q_4^2 + \vec{q}^T \vec{q} = 1,\ q_4 \in \mathbb{R},\ \vec{q} \in \mathbb{R}^3\} \tag{6.13}$$

From (6.4) and (6.12), it is possible to obtain the following coordinate transformation matrix for the Euler parameters :

$$C = C(q) = (q_4^2 - \vec{q}^T \vec{q}) I_3 + 2(\vec{q}\vec{q}^T + q_4 [\vec{q}^\times]) \tag{6.14}$$

Equation (6.14) can be written in terms of the individual quaternion elements as follows :

$$C(q) = \begin{pmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1 q_2 + q_3 q_4) & 2(q_1 q_3 - q_2 q_4) \\ 2(q_1 q_2 + q_3 q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2 q_3 + q_1 q_4) \\ 2(q_1 q_3 + q_2 q_4) & 2(q_2 q_3 - q_1 q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{pmatrix} \tag{6.15}$$

which yields the following expression for calculating the quaternion elements from the elements of rotation matrix, $C_{ij}$ :

$$\begin{aligned} q_1 &= \frac{C_{23} - C_{32}}{4 q_4} \\ q_2 &= \frac{C_{31} - C_{13}}{4 q_4} \\ q_3 &= \frac{C_{12} - C_{21}}{4 q_4} \end{aligned} \tag{6.16}$$

where

$$q_4 = \pm\frac{1}{2}\sqrt{1+C_{11}+C_{22}+C_{33}} = \frac{1}{2}\sqrt{1+trace\,C} \tag{6.17}$$

Since scalers and spatial vectors are used in quaternion, the rules in scalar and vector algebra also apply to quaternion[Cho92]. Let consider $q = q_0 + \vec{q}$ (or in Hamiltion form $q = b\vec{i} + c\vec{j} + d\vec{k}$) and $p = p_0 + \vec{p}$. Quaternion multiplication, designed by $\otimes$, is defined as

$$\begin{aligned} p \otimes q &= (q_0 + \vec{q}) \otimes (p_0 + \vec{p}) \\ &= p_0 \otimes q_0 + p_0 \otimes \vec{q} + q_0 \otimes \vec{p} + \vec{p} \otimes \vec{q} \end{aligned} \tag{6.18}$$

where $p_0 \otimes q_0 = p_0 q_0$ ,$p_0 \otimes \vec{q} = p_0\vec{q}$ and $q_0 \otimes \vec{p} = q_0\vec{p}$. The *vector-vector* quaternion product is defined as

$$\vec{p} \otimes \vec{q} = -\vec{p} \cdot \vec{q} + \vec{p} \times \vec{q} \tag{6.19}$$

where operation '$\cdot$' and '$\times$' denote the dot and the cross product in the space of spatial vectors. It is possible to separate the scaler and the vector parts of (6.18) as

$$\vec{p} \otimes \vec{q} = \begin{bmatrix} p_0 q_0 - \vec{p}^T \vec{q} \\ \vec{p} q_0 + (p_0 I_3 + \vec{p})\vec{q} \end{bmatrix} = \begin{bmatrix} q_0 p_0 - \vec{q}^T \vec{p} \\ \vec{q} p_0 + (q_0 I_3 + \vec{q})\vec{p} \end{bmatrix} \tag{6.20}$$

The selection of an attitude representation to implement a rotation or orientation operation is highly dependent on memory, storage and required performance of application, as well as the hardware and operating system environment.
The quaternion representation is compact also it requires only four scalars for storage. In addition, the computation cost of using quaternion is less than using direction cosines or Euler Angles when quaternion multiplication is applied [Rob58]. The Euler angle is compact, requiring only three scalars for storage, but is computationally inefficient conversion to and from another type of representation is generally required for rotation and composition operations. An additional disadvantage is the presence of singular points. The primary importance of the Euler angle conventions are that measurements of many physical systems or controller command in some systems is presented by Euler angles.

### 6.1.4 Drone sensors

In order to establish a certain control mechanism for drone, the changing states and properties of its internal and external environment must be known to the controller. In order to establish this requirement some sensors must be used that are specific for different observation environments. In our prototype sensors with MEMS (Micro Electronic and Mechanical Systems) technology are used.
Recent advancements and rapid growth of micro/nanotechnology have opened doors for potential applications of MEMS devices in various fields. MEMS refer to devices that have characteristic length of less than 1$mm$ but more than 1 micron, that combine electrical and mechanical components, and that are fabricated using integrated circuit batch-processing technologies. Current manufacturing techniques for MEMS include surface silicon micromachining ; bulk silicon micromachining ; lithography, electrodeposition, and plastic molding ; and electrodischarge machining [eH01] and [eH06]. The resulting microsystems have shown, for a variety of applications, unprecedented levels of miniaturization, reliability, and new capabilities. Using MEMS sensors in drone results reducing in the weight, dimension and cost of the application.
Drone sensors consists of a tri-axes accelerometer, a tri-axes magnetometer, a tri-axes rate gyro mounted at right angle. Furthermore, there are four sensor to measure angular velocity of each motor.

**Rate gyro :**Based on the principles of conservation of angular momentum, they are used to measure the rate of turn with respect to the three body-axes. The gyroscopes uses multiple vibrating elements. when sensor is in rest mode it produces identical and constant amplitude sinusoidal signals. and in rotation mode, the Coriolis forces induce a change amplitude of these signals.
Mathematical model of rate gyro(gyroscope) can be shown as follows

$$\omega_g = \omega_d + \beta + \eta_1, \quad \dot{\beta} = -T^{-1}\beta + \eta_2 \tag{6.21}$$

Where $\omega_d$ is real rotation speed in drone, $\beta$ is the bias inherent in rate gyro measurements, $T = \tau I_3$ and $\tau = 100s$, $\eta_i, i = 1,2$ are assumed to be Gaussian white noises of appropriate dimension.
**Accelerometer :** An accelerometer is an electromechanical device that measure acceleration of one object. Mechanical accelerometers consist of a spring-mass system, with a seismic mass carried by elastic tether beams. Acceleration along the sensitive axis leads to a deflection of the seismic mass, with elastic forces from the tether beams balancing the external forces. With $m$ and $k$ the seismic mass and the spring constant of the spring-mass system, the resonance frequency of the spring mass system $f_0$ is :

$$f_0 = \frac{1}{2\pi}\sqrt{\frac{k}{m}} \tag{6.22}$$

and the quasistatic sensitivity $s$, i.e., the deflection per acceleration unit at low frequencies, is :

$$s = \frac{1}{\omega_0^2} = \frac{1}{4\pi^2 f_0^2} \tag{6.23}$$

In our application, The 3-axis accelerometer senses the inertial forces and gravity in body frame **B**. The transformation of accelerometer measurements from inertial frame **R** to body frame **B** is computed as follows

$$b_{acc} = C(q)(\dot{v} - g) + \eta_{acc} \tag{6.24}$$

where $b_{acc}$ corresponds to the measurements in **B** and $\eta_{acc}$ is Gaussian zero-mean white noise. If motion is supposed quasi-static so that linear accelerations $\dot{v}$ are neglected (i.e. $\dot{v} = 0$). Note that this assumption is fully valid because when the quadrotor is controlled to obtain hover conditions ($\varphi = \theta = \psi = 0$). Moreover, the Coriolis effect is not taken into account. In this way, accelerometers are only sensitive to the gravitational field $g$[BSGL09].
**Magnetometers :** A magnetometer measures the three components of the magnetic field and provides a measurement of the attitude relative to inertial coordinates [ODC05]. The information provided by the three magnetometers mounted orthogonally is added to the inertial measurements. The magnetic field is sensed in body frame **B** and it is defined by

$$b_{mag} = C(q)h_m + \eta_{mag} \tag{6.25}$$

where $h_m = [h_{mx}\, 0\, h_{mz}]^T$ and $b_{mag}$ are the three components of the magnetic field in **R** and **B**, respectively. In our application, *MAG 3* [MEM] provides all the sensors required for inertial measurement in a single SMT package measuring $0.70 \times 0.70 \times 0.40$ inches. This analog inertial measurement unit is capable of sensing rotation, acceleration and magnetic field about three orthogonal axes. It consists of triaxial analog outputs of acceleration, rate of turn (gyrometer) and magnetic field data(Magnetometer). MAG3 Orientation Diagram is illustrated in Figure 6.8.

### 6.1.5 Mechanical Model and attitude estimation

The drone dose not have a swashplate. In fact it does not need any blade pitch control. The throttle input is the sum of the thrusts of each motor. A quadrotor reduced to a PVTOL when the pitch and

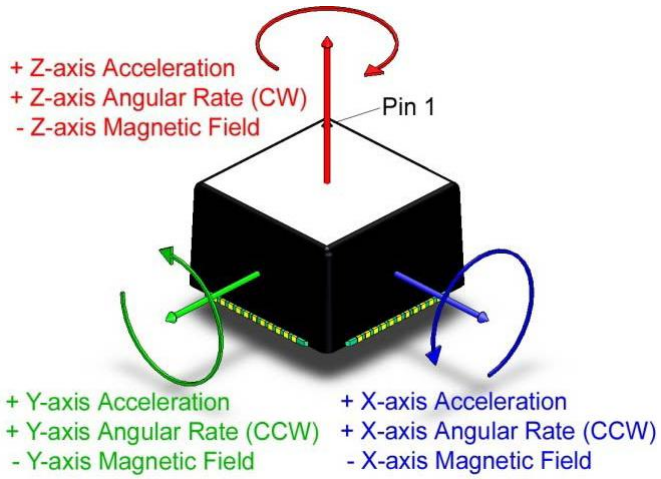



FIGURE 6.8 – MAG3 Orientation diagram [MEM]

yaw angles are set to zero and in a way it can be seen as two PVTOLs connected such that their axes are orthogonal [CLD]. For developing movement equations of the drone, the following assumptions are taken into account.

- Drone structure in cross form is considered as a rigid object.
- Structure of the drone is assumed completely symmetric. This assumption allows us to consider a orthogonal inertia matrix.
- The force $f_i$ and the relative torque $Q_i$ produced by $i-th$ motor are proportional to $\omega^2$.
- Introduced distributions by wind were ignored.

To develop the dynamic equation of the drone two frames are considered. The Earth inertial frame $R(e_x, e_y, e_z)$ and the body frame $B(e_1, e_2, e_3)$ attached to the aircraft with its origin at the center of mass of the vehicle(Figure 6.9). The orientation of the drone is parameterized by by three rotation angles with respect to frame $B$ :yaw ($\psi$), pitch ($\theta$) and roll ($\phi$). There are different approaches to describe dynami-

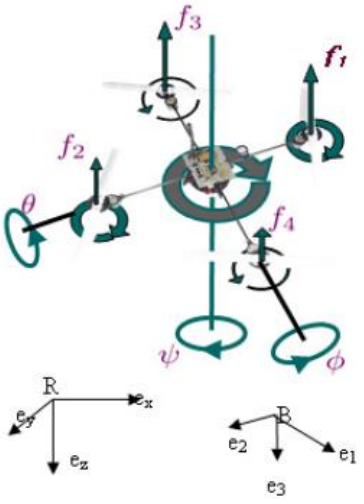



FIGURE 6.9 – inertial and body frame of drone

cal model of one system. In Newton-Euler approach forces and torques are used while, in Lagrangian

approach kinematic and potential energy are principal parameters. Because of using Quaternion representation in this application Lagrangian approach cannot be used [GC08]. Dynamic of the system can be describe as follow :

$$\begin{aligned}
\dot{\vec{p}} &= \vec{v} \\
\dot{\vec{v}} &= \vec{e}_z g - \tfrac{1}{m} C^T(q) F_T \\
\dot{q} &= \tfrac{1}{2} q \otimes \Omega = \tfrac{1}{2} \Omega(\overline{\omega}) q \\
I_f \dot{\overline{\omega}} &= -[\overline{\omega}^{\times}] I_f \overline{\omega} - G_a + \tau_a
\end{aligned} \tag{6.26}$$

where $m$ indicates the mass of the drone. $g$ is the acceleration due the gravity. $\vec{e}_z = (0,0,1)^T$ is a unit vector in reference frame $R$. The vector $\vec{p} = (x,y,z)^T$ denotes the position of the center of mass of the drone in the frame $B$ relative to $R$. The vector $\vec{v} = (v_x, v_y, v_z)^T$ is the linear velocity of the origin of $B$ measured in the reference $R$. $I_f \in \mathbb{R}^{3\times3}$ is symmetric positive definite constant inertia matrix of drone with respect to frame $B$ and $\overline{\omega} \in \mathbb{R}^{3\times3}$ is angular velocity of the drone measured by rate gyros in $B$ frame. $F_T$ is the total thrust generated by the four rotors that is calculated from

$$F_T = \sum_{i=1}^{4} f_i = b \sum_{i=1}^{4} \omega_i^2 \tag{6.27}$$

Each rotor may be thought of as a rigid disk rotating around the axis $e_3$ in $B$ frame with angular velocity of $\omega_i$ $(i = 1,\ldots,4)$ and $b$ is proportionality constant depending on the density of air, the cube of the radius of the rotor blades, the number of blades, the chord length of the blades, the lift constant (linking angle of attack of the blade airfoil to the lift generated), the drag constant (associated with the airframe) and the geometry of the wake[HMLO02, MH02]. The mechanical equation for each motor is expressed as :

$$I_r \dot{\omega}_i = \tau_i - Q_i \quad , i \in 1,2,3,4 \tag{6.28}$$

where $\tau_i$ denotes the electromotive torque of rotor $i$ and $I_r$ the moment inertia of rotor. The reactive torque generated by rotor $i$ in (6.28) is given by :

$$Q_i = k\omega_i^2 \tag{6.29}$$

The gyroscopic torque,$G_a$, in (6.26), due to the combination of the rotation of the drone and four rotor, are modeled as :

$$G_a = \sum_{i=1}^{4} I_r(\overline{\omega} \times e_z)(-1)^{i+1} \omega_i \tag{6.30}$$

The torque generated by the rotors $\tau_a = [\tau_1, \tau_2, \tau_3]$ in (6.26) can be found by :

$$\begin{aligned}
\tau_1 &= db(\omega_2^2 - \omega_4^2) && Roll \\
\tau_2 &= db(\omega_1^2 - \omega_3^2) && Pitch \\
\tau_3 &= k(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) && Yaw
\end{aligned} \tag{6.31}$$

where $d$ is the distance from the rotor to the center of mass the drone. In (6.26), the attitude dynamic are represented by $q = (q_0, \vec{q}^T)^T$ which is a rotational quaternion and $\Omega = (0, \overline{\omega})^T$ is also a quaternion[TGLT07]. A unit quaternion represents the axis of rotation $\vec{k}$ and the angle of rotation $\xi$ around this axis under form $q = [\cos\frac{\xi}{2} \;\; \vec{k}^T \sin\frac{\xi}{2}]^T$. In (6.26) $q$ corresponds to the change of coordinates between the two frame $R$ and $B$ that gives the orientation of the drone. If $\vec{r}$ is a vector expressed in $R$, then its coordinates in $B$ are given by :

$$b = q^{-1} \otimes r \otimes q \tag{6.32}$$

where $r = (0, \vec{r}^T)^T$ , $b = (0, \vec{b}^T)^T$ and $q^{-1}$ is the complementary rotation of quaternion $q$ defined by :

$$q^{-1} = [q_0 \;\; -\vec{q}^T]^T = \bar{q} \tag{6.33}$$

(6.32) also can be expressed as

$$\vec{b} = C(q)\vec{r} \tag{6.34}$$

where $C(q)$ is the Rodrigues matrix defined as :

$$C(q) = (q_0^2 - \vec{q}^T\vec{q})I + 2(\vec{q}\vec{q}^T - q_0[q^\times]) \tag{6.35}$$

This lead to the following definition of $\Omega(\overline{\omega})$ in (6.26)

$$\Omega(\overline{\omega}) = \begin{bmatrix} 0 & -\overline{\omega}^T \\ \overline{\omega} & -[\overline{\omega}^\times] \end{bmatrix} \tag{6.36}$$

where self cross product $[\overline{\omega}^\times]$ can be found by

$$[\overline{\omega}^\times] = \begin{bmatrix} 0 & -\overline{\omega_3} & -\overline{\omega_2} \\ \overline{\omega_3} & 0 & -\overline{\omega_1} \\ -\overline{\omega_2} & \overline{\omega_1} & 0 \end{bmatrix} \tag{6.37}$$

by combining (6.37) and (6.27), total thrust and generated torques can be shown in vectorial form :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ F_T \end{bmatrix} = \begin{bmatrix} 0 & db & 0 & -db \\ db & 0 & -db & 0 \\ k & -k & k & -k \\ b & b & b & b \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = M \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{6.38}$$

It is easily verified that the matrix $M \in \mathbb{R}^{4\times4}$ is full rank for $b, k, d > 0$.

As explained earlier, nine sensor are used in IMU. So, vector of measurements can be constructed by

- Three measurements from tri-axe accelerometers $\overrightarrow{b_{acc}} = [b_{acc_x} \;\; b_{acc_y} \;\; b_{acc_z}]$
- Three measurements from tri-axe magnetometers $\overrightarrow{b_{mag}} = [b_{mag_x} \;\; b_{mag_y} \;\; b_{mag_z}]$
- Three measurements from tri-axe rate gyros $\overrightarrow{\omega_g} = [\omega_x \;\; \omega_y \;\; \omega_z]$

| Parameter | Description | Value | Unit |
|---|---|---|---|
| $d$ | distance from the rotor to the center of mass the drone | 0.225 | $m$ |
| $m$ | mass of drone | 0.520 | $Kg$ |
| $J_a$ | motor inertia | $3.4\times10^{-5}$ | $Jg\cdot m^2$ |
| $R$ | motor resistance | 0.67 | $\Omega$ |
| $K_m$ | motor constant | $4.3\times10^{-3}$ | $N\cdot m/A$ |
| $K_g$ | motor constant | 5.6 | |
| $I_r$ | moment inertia of rotor | $3.357\times10^{-5}$ | $Jg\cdot m^2$ |
| $b$ | constant | $29.1\times10^{-5}$ | |
| $I_f$ | symmetric positive definite inertia matrix | $diag\{8.28,8.28,5.7\}\times10^{-3}$ | $kg\cdot m^2$ |
| $\eta_{acc}$ | white noise in accelerometers | 0.002 | $m/s^2$ |
| $\eta_m$ | white noise in magnetometer | 0.0007 | $mgauss$ |
| $\eta_G$ | white noise in rate gyros | 0.01 | $rad/s$ |
| $K_Q$ | constant | $1.1\times10^{-6}$ | |

TABLE 6.2 – notation and their values used in model of drone

therefore measurement vector which passes through the network is

$$Y_m=\begin{bmatrix}\vec{b_{acc}}\\ \vec{b_{mag}}\\ \vec{\omega_g}\\ \vec{\omega_i}\end{bmatrix}=\begin{bmatrix}acc_x\\ acc_y\\ acc_z\\ mag_x\\ mag_y\\ mag_z\\ \omega_x\\ \omega_y\\ \omega_z\\ \vec{\omega_i}\end{bmatrix}\qquad i=1,\dots,4 \tag{6.39}$$

Measurements $(b_{acc},b_{mag},\omega_g)$ are used in controller side to feed a non linear observer to estimated quaternion of drone, $\hat{q}$. Block diagram of this observer was illustrated in Figure 6.10. In this figure $\otimes$ represents quaternion multiplication and $\oplus$ denotes to matrix multiplication. The non linear observer can be described as follows. A "pseudo-measurement" quaternion $q_{ps}$ is computed from measurements of $b_{acc},b_{mag}$ based on non linear static measurement equation

$$q_{ps}=\underset{|q|^2=1}{\arg\min}\left[\frac{1}{2}\|[b_{acc}^T\ \ b_{mag}]^T-h(q)\|_2^2\right] \tag{6.40}$$

where $h(q)$ is derived from (6.24) and (6.25). A sequential Quadratic Programming(SQP) algorithm is used at this step. $\hat{q}$ is then obtained by propagating the kinematic equation (6.26) using $\omega_g$ in (6.21), and the discrepancy between $\hat{q}$ and $q_{ps}$ is computed as follows

$$q_e=\hat{q}\times q_{ps}^{-1}=\begin{bmatrix}q_{e0}&\vec{q}_e^T\end{bmatrix}^T \tag{6.41}$$

The main controller provides necessary torque $\tau_a$ about each axes of drone and by using inverse of matrix $M$ in (6.38) reference of angular velocity of each motor $\omega_{r,i}$ is computed. These values are send to local controllers that are in charge to calculate necessary voltage of motors, $u_i$. In Figure 6.11 closed

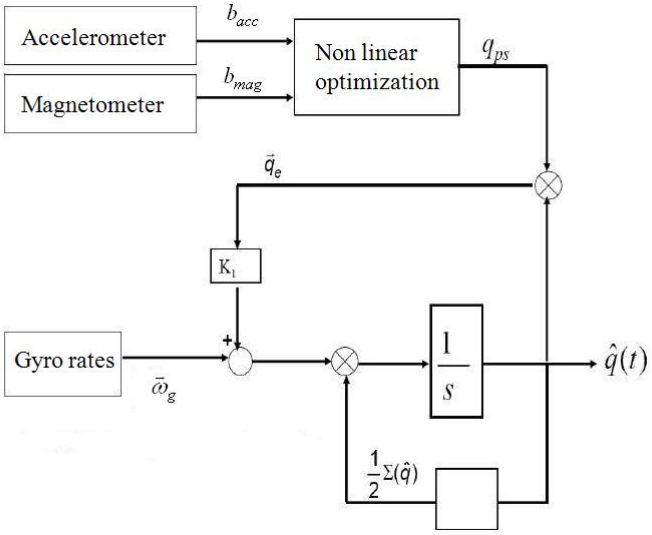



FIGURE 6.10 – Attitude estimation

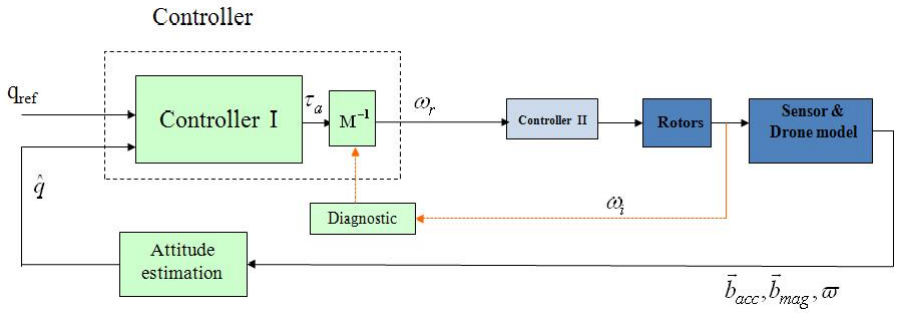



FIGURE 6.11 – Drone close loop

loop control of drone was illustrated.
Four DC motors of drone can be modeled by

$$\dot{\omega}_i = \frac{K_m u_i}{J_a R} - \frac{K_m^2}{J_a R}\omega_i - \frac{K_Q|\omega_i|}{J_a K_g^3} \tag{6.42}$$

where $K_m$, $J_a$, $R$ and $K_Q$ are motor constants. In this application maximum voltage which can be applied to motors is 9 $V$ and when this voltage is applied to motors (it is considered that all motors are identical) a maximum angular velocity of 260 $rad/s$ can be obtained. Then, based on value of $b$ and $k$ in (6.38) maximum generated torque in each axes will be :

$$\tau_1 = 0.40\ Nm \quad \tau_2 = 0.40\ Nm \quad \tau_3 = 0.15\ Nm \tag{6.43}$$

Table 6.2 summarizes the notation and parameters of model of drone.

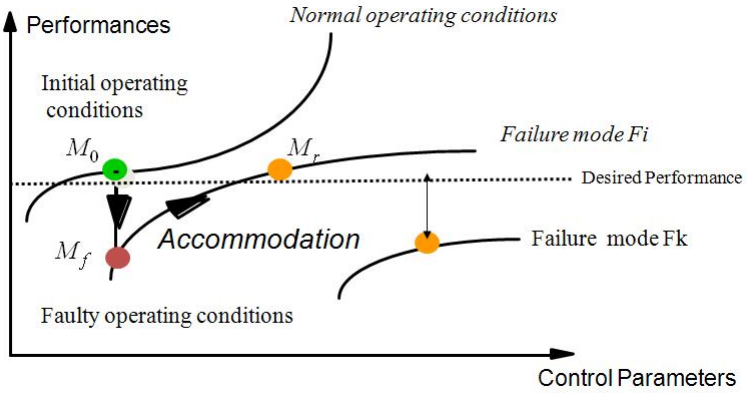



FIGURE 6.12 – The reconfiguration problem

## 6.2 Fault tolerant control

Existing works in Fault tolerant control (FTC) systems design can be classified into two main approaches : passive and active. In passive approaches, the conceivable system components are assumed to be known a priori, and control system takes into account of all these failure modes in design stage. Once the control system is designed, it will remain fixed during the entire system operation [Jia05]. So even in case of component failures, the control system should be able to maintain the necessary performance. In the other words, in a passive FTC, the control system must works under all possible system operating scenarios that were considered in design stage. In passive FTC, various component failures must be taken into account. As a consequence, the designed controller has to be conservative.
In contrast, an active FTC reacts to the component failures by properly reconfigurating its control actions so that the performance stability of the entire system can still be acceptable [ZJ08]. Control reconfiguration is triggered by fault detector so to achieve a successful control system reconfiguration, this approach relies strongly on real-time fault detection/diagnostic scheme. In the other words, performance and rapidity of fault detector has important influence on performance of active FTC. The critical issue facing any active FTC, comparing to passive strategy, is that there is only limited amount of reaction time available to perform fault detection and diagnostic and control system reconfiguration. The speed, accuracy, and the robustness of these schemes are the factors to the success of any active FTC. In 6.3 importance of fast fast fault detection in drone application will studied.
Network is one component in drone application and similar to other components such as sensors, actuators may be effected by some faults such as packet dropout. Depending on attitude of drone, a packet dropout in messages corresponding to commands of drone may degrade performance of the system or even destabilize it (see 6.2.2). In the following, an approach for fault tolerant control in case of freezing fault in motors of drone because of packet dropout or physical problem in drone is studied.
The principle of our active approaches, illustrated by Figure 6.12, is very simple. After the fault occurrence, the system deviates from its nominal operating point defined by its input/output variables $M_0$ to a faulty one $M_f$. The goal of fault-tolerant control is to determine a new control law that takes the degraded system parameters into account and drives the system to a new operating point $M_r$ such that the main performances (stability, accuracy, ...) are preserved (i.e. are as close as possible to the initial performances). It is therefore important to define precisely the degraded modes that are acceptable with regard to the required performances, since after the occurrence of faults, conventional feedback control design may result in unsatisfactory performance such as tracking error, instability, and so on. The classical way to achieve the fault tolerant control consists of the following task that is illustrated in Figure 6.13 :

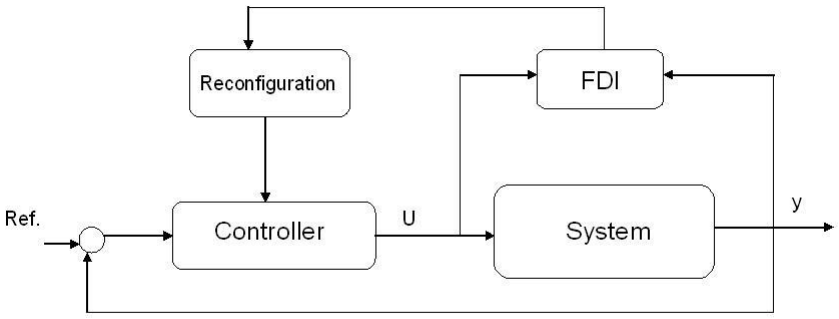



FIGURE 6.13 – Fault reconfiguration in drone

– Fault detection and isolation (FDI)
– System output evaluation and choice of right degraded system
– Fault compensation via control reconfiguration.

### 6.2.1 Reconfiguration in case of critical failures

In case of critical failures which cannot be compensated via feedback control, the nominal performances cannot be preserved and degraded performances should be tolerated by the process operator. From a practical point of view, many industrial processes offer the possibility to deal with the output performances by reducing the objectives.
We show that under steady state operating conditions, a solution to the above mentioned problem can be easily proposed. Let us consider a dynamical system (S) described by the following discrete state equation :

$$\begin{cases} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k) \end{cases} \tag{6.44}$$

Where the different variables $u(k)$, $x(k)$, $y(k)$ designate variations around the nominal operating conditions according to :

$$\begin{cases} X(k) = X_0 + x(k) \\ U(k) = U_0 + u(k) \\ Y(k) = Y_0 + y(k) \end{cases} \tag{6.45}$$

And under steady state conditions, it is supposed that the nominal operating point $M_0 : (U_0, Y_0)$ satisfy to the physical constraints :

$$S_0 : (U_0, Y_0) = 0 \tag{6.46}$$

With the output controlled variables $Y_0 = CX_0$.
But, if a critical failure occurs, and affects the system, then the solution is no longer valid and the output of the system move to a new operating point $M_f : (U_f, Y_f)$ where subscript $f$ designate the faulty conditions. Consequently, the initial performances are not reachable and the performance index must accommodate to new operating conditions closer to the initial ones.
The reconfiguration strategy which is proposed here relies on the acceptation of degraded performance for the reconfigured operating condition $M_r : (U_r, Y_r)$.
Let us decompose the output vector into $Y_r = [Y_r^p \ , \ Y_r^s]$ in order to exhibit priority and secondary output variables. The former are considered of a prime importance for the system and should be kept constant

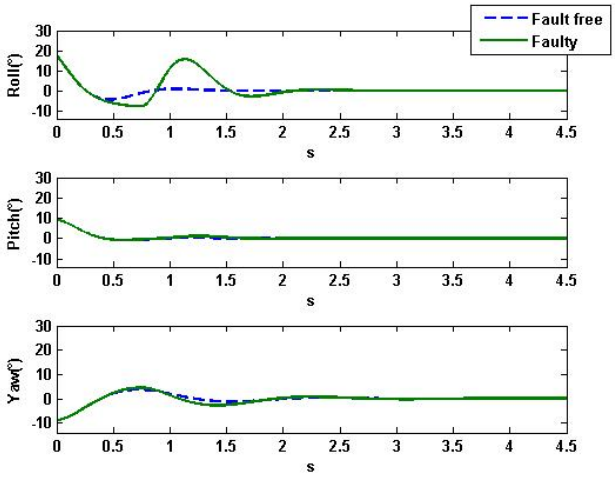



FIGURE 6.14 – drone attitude in faulty and fault free condition

at the nominal set-point values, thus, leading to the condition $Y_r^p = Y_0^p$ while the latter are free to evolve inside a region of the state space corresponding to acceptable degraded performances. This can be reflected via the optimization of the performance index $\Psi[(Y_r^s - Y_0^s), U]$ which is to be to define according to operator requirements.
Under the given constraints and with respect to the criteria which to be optimized the new operating point leading to the control reconfiguration, minimize the Lagrangian function :

$$L = \Psi(Y_r^s - Y_0^s, U_r) + \mu^T (Y_r^p - Y_0^p) + \lambda^T S(U_r, Y_r) \tag{6.47}$$

where components of vector $\lambda$ and $\mu$ are the lagrange multipliers.

### 6.2.2 Actuator and network faults and its effects

In what follow, we consider an actuator freezing fault. Due to the assumption of zero order holder (ZOH) at the receiving side of the communication medium, i.e., when an actuator fails to access the medium, the value stored in a ZOH is fed into the system. Stored value in ZOH corresponds to last command before fault occurrence in actuator. There is the possibility to consider a temporary freezing fault in an actuator due to some faults in network, such as packet loss or long delay, which causes to not update motor command during a period. Also occurrence of this fault may be because of some physical problems in rotor. But in both cases angular velocity of a motor remains constant in its last value during the fault occurrence.
For example, a network fault occurs at $0.25s$ in the command message of second motor and lasts $0.5s$, angular velocity of second rotor remains constant in its value before fault occurrence. Figure 6.14, Figure 6.15, and Figure 6.16 illustrate drone attitude (expressed in Euler angles), the motors angular velocities and total thrust when the initial attitude was $(25;10;-10)(deg)$ and the reference was $(0;0;0)(deg)$. To develop reconfiguration module, it was supposed that there is no more than one faulty rotor in the system at each moment and system sensors are faultless. Based on (6.31), it is possible to say that effects of the fault occurrence in second and fourth motors can be seen in roll and pitch angles. Whereas pitch and yaw angles will be effected in case of fault occurrence in first and third motor. Effects of freezing fault in hover flight without disturbance can be neglected. However, in real experiments and in presence of

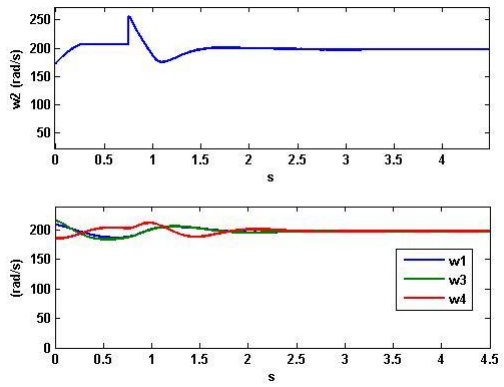



FIGURE 6.15 – Angular velocities of motors

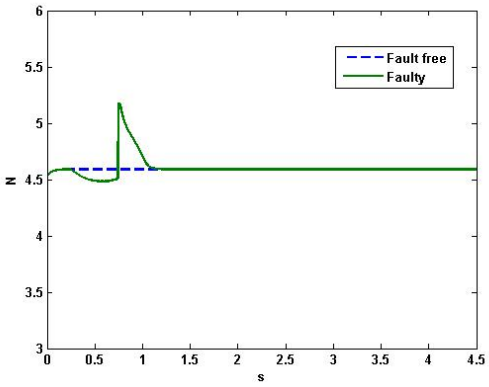



FIGURE 6.16 – Total thrust generated by four motors

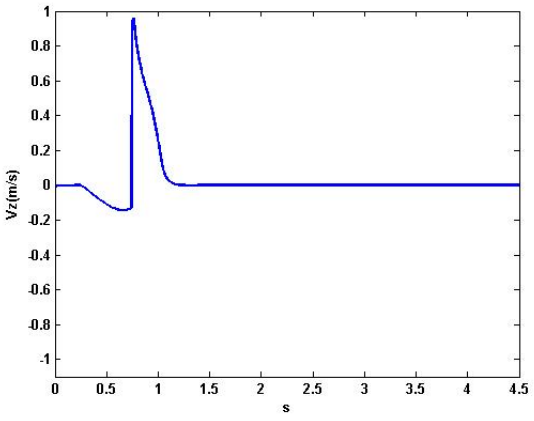



FIGURE 6.17 – linear velocity of drone due to network fault

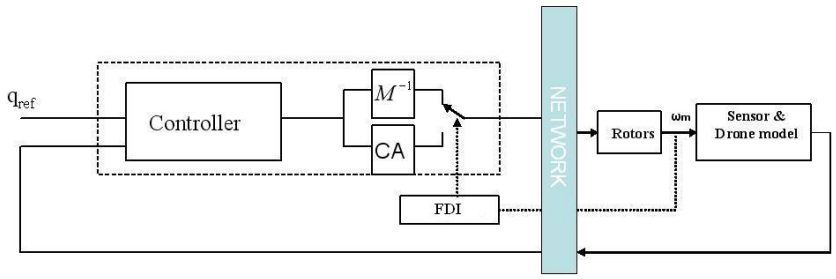



FIGURE 6.18 – Fault reconfiguration in drone

disturbance in the system or during transitory phases even a short duration fault destabilizes the drone. One straightforward for solving this problem is developing a robust controller or use some FTC methods that can be found in literature. By considering constraints on memory and limitation in micro-controller computation, our proposed approach can be implemented simply in micro-controller that was shown in Figure 6.2.

### 6.2.3 Fault tolerance control module

As it is illustrated by Figure 6.14, after fault occurrence controller could not compensate fault effects on the system and Euler angles deviate from their nominal conditions. Also the fault is introduced negative or positive vertical acceleration in drone (Figure 6.17). The developed FTC algorithm is based on the reallocation of the total control amount by taking into account the actuator limitations and the information provided by the FDI module. By using block $M^{-}1$ in Figure 6.18 the controller provides references of motor velocity. In the case of fault occurrence, by means of a special block of fault detection and isolation, the fault is detected and the control allocator (CA) is triggered. Then CA module will be in charge to compensate fault effects on the drone by using some reconfiguration strategies.
Here, we will consider critical faults type that cannot be compensated via feedback control. In this case, the nominal performances cannot be preserved and degraded performances should be tolerated by the process operator. Different possible choices of prior and secondary objectives leads to the following FTC strategies. When a fault was detected, based on flight mode of drone or its attitude, primary and secondary outputs are selected and its corresponding strategy for compensating effects of faults on the drone was chosen in CA. Output classification and proposed strategies are given as follows.

#### 6.2.3.1 First strategy

In this strategy, the prior objective is to maintain pitch and roll angles while the secondary one is to optimizing total thrust and yaw angle. The Lagrangian is then given by :

$$L = \psi_1(\tau_{3a} - \tau_{3c})^2 + \psi_2(F_{T,a} - F_{T,c})^2 + \lambda_1(\tau_{1a} - \tau_{1c}) + \lambda_2(\tau_{2a} - \tau_{2c}) \tag{6.48}$$

Where $\tau_{jc}$ , $j = 1,2,3$ is calculated torque by the controller at each sampling period and $tau_{ja}$ is necessary torque to reach drone reference in degraded mode. By applying this strategy to faulty drone, the total thrust and torque corresponding to yaw angle of the drone are low priority outputs whereas keeping two other torque relating to roll and pitch angles are included to high priority system constraints. After

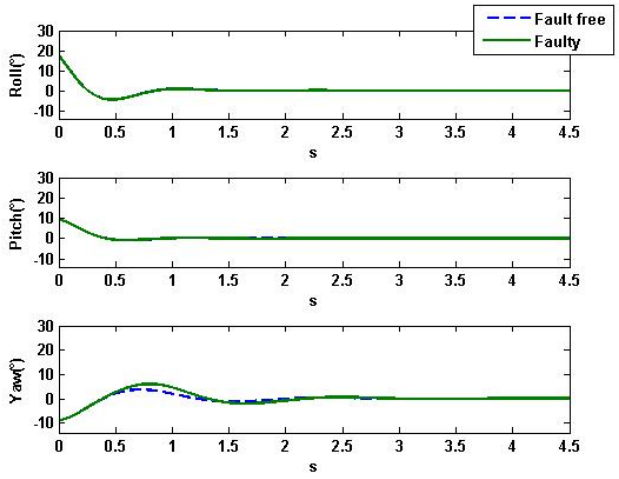



FIGURE 6.19 – drone attitude (First Strategy)

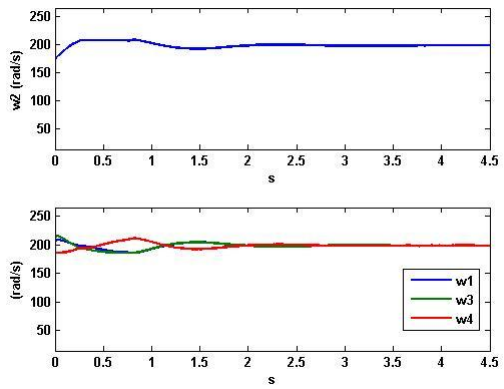



FIGURE 6.20 – Angular velocities of motors (First Strategy)

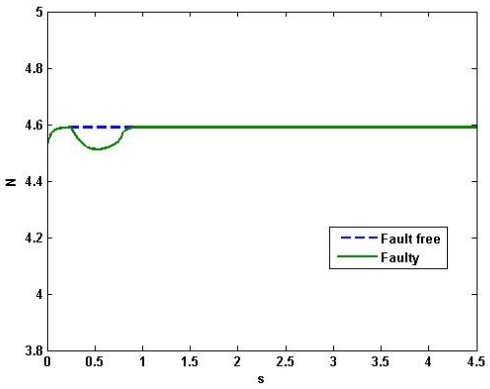



FIGURE 6.21 – Total thrust generated by four motors (First Strategy)

reconfiguration, the roll and pitch angles are close to their references. Results of applying this reconfiguration strategy in case of freezing fault in second motor at $t = 0.25s$ were compared with fault free condition in Figure 6.19, Figure 6.20, and Figure 6.21.

#### 6.2.3.2 Second strategy

If Drone is in stable condition and total thrust is close to its reference, after fault occurrence there are not critical problem in vertical movement and Euler angles have high priority to be stabilized. So, (6.48) will be changed to :

$$L = \psi_1(F_{T,a} - F_{T,c})^2 + \lambda_1(\tau_{1a} - \tau_{1c}) + \lambda_2(\tau_{2a} - \tau_{2c}) + \lambda_3(\tau_{3a} - \tau_{3c}) \tag{6.49}$$

There are not optimization problem in this case. This strategy keeps three angles as output constraints and it considers $F_T$ as free factor of the system outputs. Drone responses after reconfiguration based on

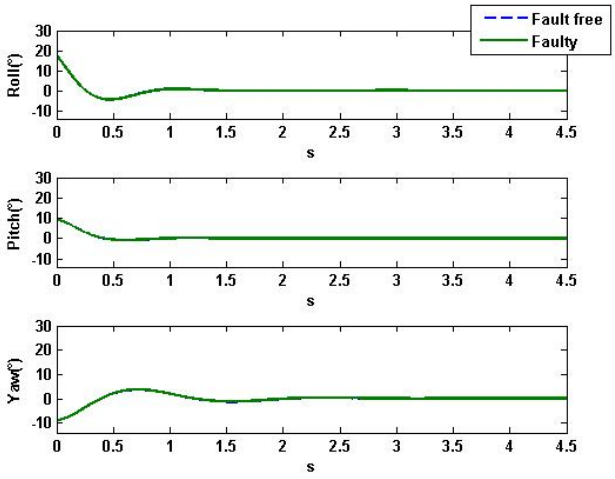



FIGURE 6.22 – drone attitude( Second Strategy)

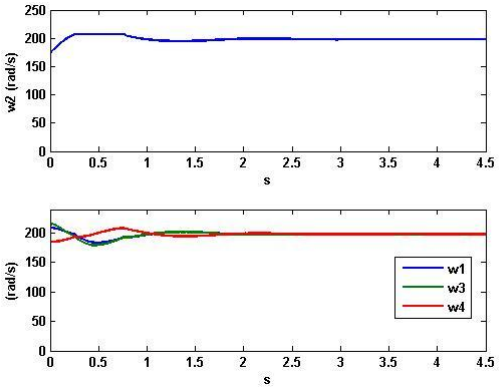



FIGURE 6.23 – Angular velocities of motors (Second Strategy)

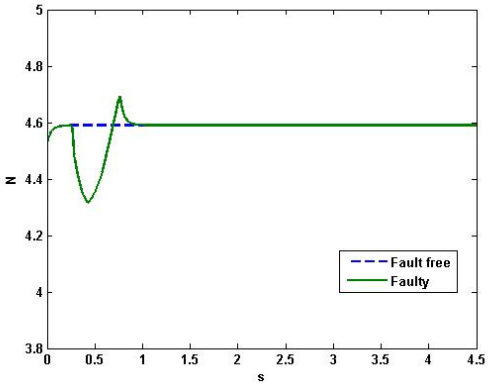



FIGURE 6.24 – Evolution of the total thrust in second strategy (Second Strategy)

second strategy for a networked fault which occurred at $0.25s$ were described in Figure 6.22, Figure 6.23, and Figure 6.24. After detecting the fault by diagnostic module, reconfiguration strategy was triggered and it could eliminate fault effects from Euler angles but $F_T$ was diverted from nominal condition. As a consequence, a negative or positive vertical acceleration is possible in the drone.

#### 6.2.3.3 Third strategy

Here the torques relating to roll and pitch angles also the total thrust $F_T$ are more important than yaw angle. So keeping values of $\tau_1$, $\tau_2$ and $F_T$ in drone outputs are hight priority in reconfiguration and $\psi_1(\tau_{3a} - \tau_{3c}) = 0$ must be minimized. Therefore, (6.46) is rewritten :

$$L = \lambda_1(F_{T,a} - F_{T,c}) + \lambda_2(\tau_{1a} - \tau_{1c}) + \lambda_3(\tau_{2a} - \tau_{2c}) + \psi_1(\tau_{3a} - \tau_{3c})^2 \tag{6.50}$$

In this strategy objective is to apply exact values of necessary torques related to roll and pitch angles as well as the total force to faulty drone while, the torque correspond to yaw angle is free. In Figure

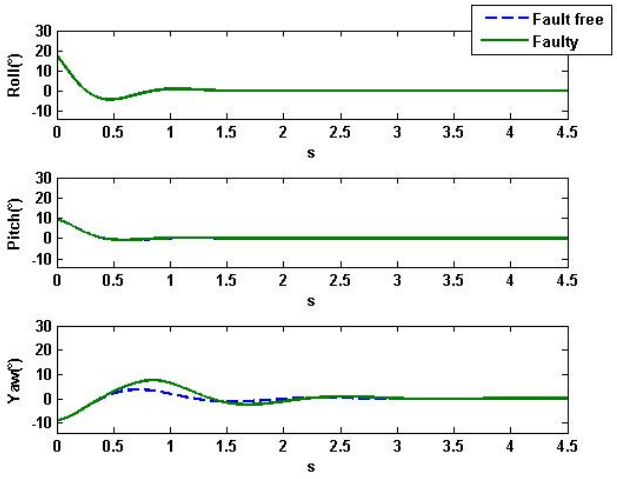



FIGURE 6.25 – drone attitude(Third Strategy)

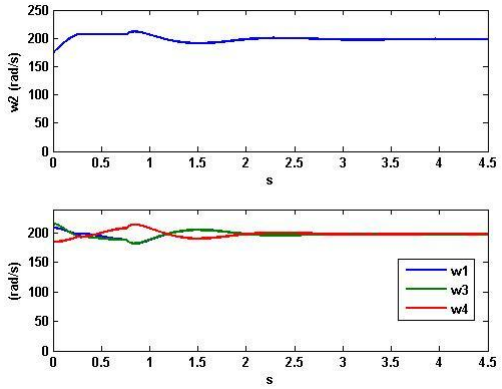



FIGURE 6.26 – Angular velocities of motors (Third Strategy)

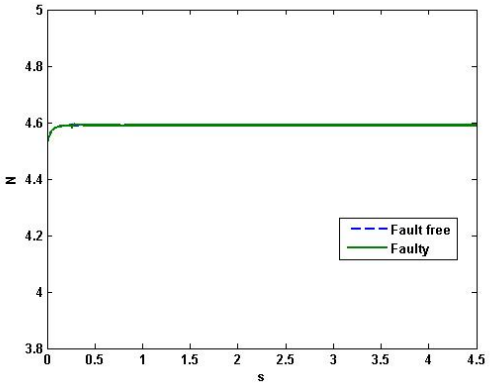



FIGURE 6.27 – Evolution of the total thrust in second strategy (Third Strategy)

6.25, Figure 6.26 and Figure 6.27 results of applying this strategy on the drone were shown. When fault was occurred in the system at $0.25s$, angular velocity of second motor remained constant at last received value. After fault detection by the diagnostic module, reconfiguration strategy distributed total command. Then, the vertical force and the roll and pitch angles were kept controllable by controller but yaw angle was changed to free parameter. As it was shown in Figure 6.25 there is deviation in yaw angle after fault occurrence.

#### 6.2.3.4 Fourth strategy

If the drone is in steady state position and network faults such as long delay or packet dropout is happen in short duration, it is possible to prevent completely vertical force deviation and minimize deviation in Euler angles in the system outputs. So reconfiguration strategy may be written as

$$L = \lambda_1(F_{T,a} - F_{T,c}) + \psi_1(\tau_{1a} - \tau_{1c})^2 + \psi_2(\tau_{2a} - \tau_{2c})^2 + \psi_3(\tau_{3a} - \tau_{3c})^2 \tag{6.51}$$

## 6.3 Semi-online scheduling and fault detection

In drone application fast fault detection is no less important than fault tolerance control strategy. In case of fault occurrence, a fast fault detector activates faster reconfiguration module and as a consequence error in the system will be corrected more quickly. Here, two cases are studied. When a freezing fault is occurred in second motor at $0.75s$. In the first simulation reconfiguration module is activated at $0.9s$ and in second case it is activated at $1.1s$. Attitude of drone in Euler angles representation were illustrated in Figure 6.28 and Figure 6.29. This simulation was done when drone initial attitude was $(25,-10,10)$ *deg* and controller reference was $(0,0,0)$ *deg*

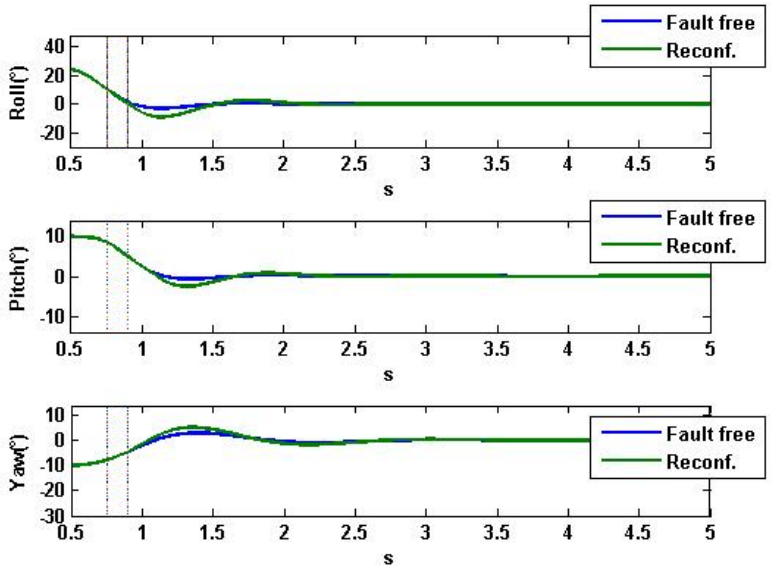



FIGURE 6.28 – Fast fault detection in case of freezing fault

It was seen in Figure 6.29 that even after trigging reconfiguration block by a slow fault detector, there is a large deviation in roll angle ($\approx 20^{\circ}$). There are enough works in literature that present efficient ways for designing a fast fault detector. In classical theories a perfect data exchange is assumed that it is not our case in drone application where a network is used for data transferring. Objective of this section is proposing a fast fault detection with considering communication constraints in drone network.
Measurement of IMU sensors, the four flows from the main control unit, the four flows from the sensors of motors and an external flow pass through network. IMU data acquisition is synchronized and sampling time is $10ms$. The scheduling of the messages through the network is an essential mechanism which strongly influences the stability of drone closed loop [BGL09].
When there are no constraints on communication, a periodic access to the network is considered for all sensors. But if there is a constraint on access to communication network, all sensors cannot send their measurements simultaneously. So, it is considered that all IMU sensors and just one angular velocity sensor have access to network at each sampling moment. Angular velocity sensors access to network consecutively with an equal time slot. We will refer order of accessing velocity sensors to network as *communication sequence*(see 3.1). In case of detection of an abnormality in one of residual, based on the size of abnormality, corresponding sensor must have access to medium with larger time slot. In follo-

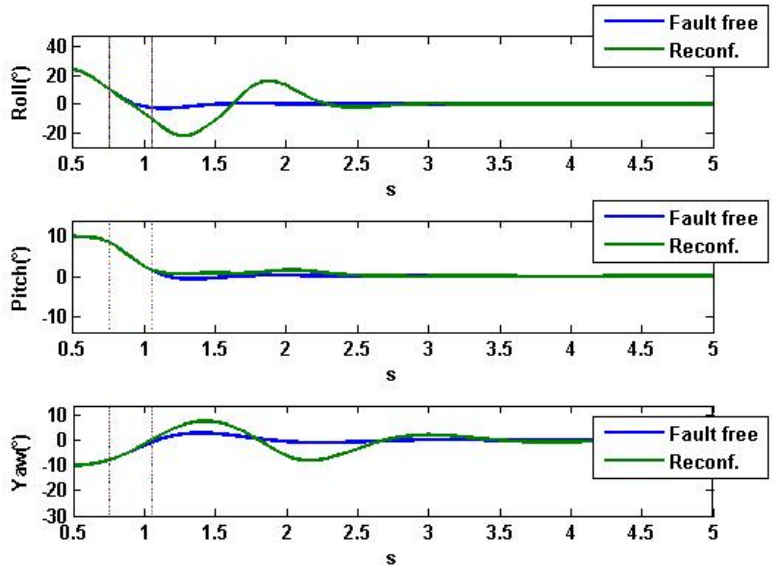



FIGURE 6.29 – Slow fault detection in case of freezing fault

wing, the problem of fault detection and a strategy to calculate access time slots for each velocity sensor is studied.

In case of the perfect communication when there are no constraints on medium access, a residual is generated in each residual generator at each sampling instant. But if limited number of channels is available for sensors, all sensors have not access to the network simultaneously so simultaneous residual generation for all sensors is not possible. One straightforward solution for this problem is to let medium access of sensors follow a pre-defined communication sequence(offline scheduling). But inconvenient of these approaches is that in case fault occurrence number of measurement updates for faulty part is equal to other parts.

Here an semi-online access scheduling is proposed. A set of admissible communication sequences is found offline. They must guarantee generation of minimum one residual for each module of RG during the sequence. Then during system running based of needs of each subsystem a communication sequence form this admissible sequences is chosen. Comparing to classical online scheduling, online computation in central CPU (controller side) is reduced. In addition dynamic of system is taken into account that is advantage of this approach compering to offline scheduling.

Similar to 5.2, It is assumed that faults do not occur simultaneously. So for increasing performance of diagnostic, sequence selection from set of admissible sequences is based on this principle that it provides more access to the sensor with more deviation in its residual from normal condition.

Sampling time of velocity sensors is $0.01s$. Length of communication sequence for accessing four sensors to the network is considered $0.2s$. In normal condition, when all residuals are near to zero, a so-called periodic communication sequence is employed for the allocation of network resource. Each velocity sensor has access to network one quarter of length of sequence that means $0.05s$ for each sensor.

### 6.3.1 Residual generation

First step of developing our strategy of fault detection and semi-online scheduling is residual generation. The motors are identical, so four identical residual generator (RG) were developed and each residual generator will be in charge to compute residual signal for each motor (Figure 6.30). After linearization

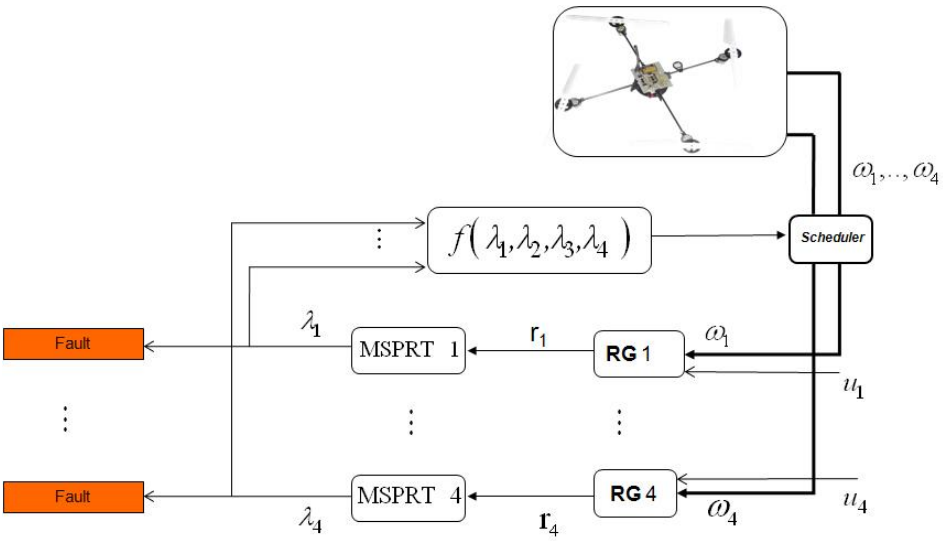



FIGURE 6.30 – Message scheduling and fault detection architecture

of the nonlinear model of motors described in (6.42) in the operating point $\omega_{i,0} = 198.15\ rad/s$, output of i-th ($i = 1,2,3,4$) residual generator will be described as follows

$$r_i(k) = v_{i,s}^T (Y_i(k) - H_{u,i} U_i(k)) \tag{6.52}$$

where

$$Y_i(k) = \begin{bmatrix} \omega_i(k-s) \\ \omega_i(k-s+1) \\ \vdots \\ \omega_i(k) \end{bmatrix}, \quad U_i(k) = \begin{bmatrix} u_i(k-s) \\ u_i(k-s+1) \\ \vdots \\ u_i(k) \end{bmatrix} \tag{6.53}$$

Matrices $H_{0,i}$ and $H_{u,i}$ are calculated based on classical parity relation approach [Ger97] that were presented in 5.2.

### 6.3.2 residual classification

Here for abnormality classification, a Multi Sequential Probability Ratio Test (MSPRT) is used. Based on the class (accepted hypothesis) of residual of all RG modules, a new sequence is chosen from set of admissible sequences. MSPRT is described as follows. Let X1, X2 ,... be an infinite sequence of random variable and identically distributed (i.i.d) with probability density function $f$ and $H_l$ be the hypothesis that $f = f_l$ for $l = 0,1,2,3$. Objective of MSPRT test is to determine the true hypothesis with a desired accuracy as quickly as possible. The posterior probability after $N$ observation is given by :

$$p_N^l = P(H = H_l | X_1, \ldots, X_N) \tag{6.54}$$

For given $M$ hypothesis with their prior probabilities $\pi_l$ , the null hypothesis $H_0$ and $H_{M-1}$ correspond to the fault-free case and faulty case, respectively and the alternative hypotheses $H_v$ , $v \in 1,\ldots,M-2$ correspond to abnormality modes. The posterior probability after $N$ generated residual is computed by :

$$p_N^l = \frac{\pi_l \prod_{k=1}^{N} f_l(r_k)}{\sum_{M-1}^{j=1} \pi_j \prod_{k=1}^{N} f_j(r_k)} \tag{6.55}$$

where $f_l$ is is the probability density function of the residual, conditioned on the hypothesis $l$ and $r_k$ is generated residual at instant $k$.
The stoping time $N_A$ and final decision $\lambda$ in MSPRT test can be described as follows :

$$\begin{gathered} N_A = \text{first } N > 1 \text{ such that } p_N^l > \tfrac{1}{1+A_l} \text{ for at least one } l \\ \lambda = \arg\max_j \left(\pi_j \prod_{k'=1}^{N} f_j(r_{k'})\right) \end{gathered} \tag{6.56}$$

For given frequentist error probabilities $\alpha_0,\ldots,\alpha_M - 1$ (accept a given hypothesis wrongly) threshold parameter $A_l$ is found by (5.12).
In this application, an equal prior probability was assumed for all hypotheses. Furthermore, residuals were considered as Gaussian variables and their variance before and after fault occurrence are known and constant quantities. Accepted hypothesis of MSPRT indicates if there is deviation in residual from normal condition and size of this deviation. For each module of RG a module of MSPRT test is designed (Figure 6.30) and based on accepted hypothesis of all MSPRT test a new communication sequence is chosen. For residual classification four hypotheses is considered as illustrated in Figure 6.31. In this Figure $r_N$ and $r_f$ denote residual mean for accepting hypothesis $H_0$ and $H_3$ respectively. Similar to 5.2.3, if output of any MSPRT test is accepting hypothesis $H_3$, it means that residual of corresponding MSPRT test exceeded the threshold of fault free condition.

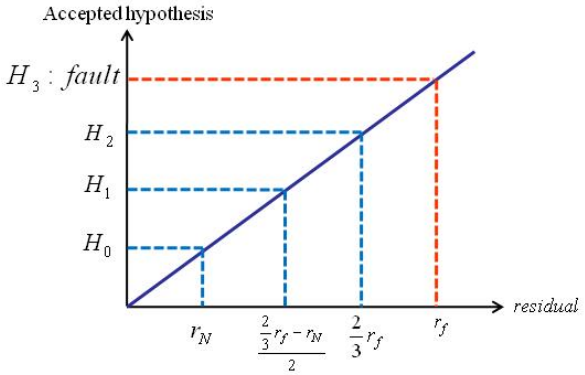



FIGURE 6.31 – Mean of residual and its hypothesis in MSPRT test

### 6.3.3 Sensor access scheduling

Order of parity relation for each RG is fixed to two ; therefore access time slot for one velocity sensor for generating one residual in its RG is set to $0.02s$. That is minimum time slot for each sensor in one communication sequence that guarantee one residual is generated in this sequence. So a sequence is admissible if it assigns minimum $0.02s$ time access to one sensor. Necessary time slot for each sensor

can be calculated based on value of accepted hypothesis $\lambda_i$ in i-th MSPRT test. It is computed at the end of each sequence by :

$$ts_i = \frac{T}{\sum_{j=1}^{4}(\lambda_j + 1)}(\lambda_i + 1) \qquad \lambda_i \in \{0, 1, \ldots, 3\} \tag{6.57}$$

Where $ts_i$ is accessing time slot of i-th sensor in next communication sequence and T is length of sequence. Scheduling can be described as follows :at the end of sequence, based on last decision of MSPRT tests a time slot for each sensor is calculated and scheduler selects one sequence with these time slots for next communication sequence.

### 6.3.4 Simulation results

The first simulation of the system was realized in fault free condition when residuals are close to zero. Residuals of RGs are shown in Figure 6.32. Accepted hypotheses of MSPRT tests were illustrated in Figure 6.33. In this figure value $-1$ indicates *no decision* in corresponding MSPRT test and other values denote to accepted hypothesis. In this simulation, hypothesis $H_0$ was accepted in all MSPRT tests then access time slot for each sensor is computed in scheduler. All residuals are in normal condition, so each sensor can have one quarter of time length of communication sequence to send its measurements. If there is not problem in the system, the sequence will be repeated periodically. It is similar to works which used static periodic communication sequence.

In the next simulation, an abnormality was introduced to residual of third motor at 3.34$s$. As a conse-

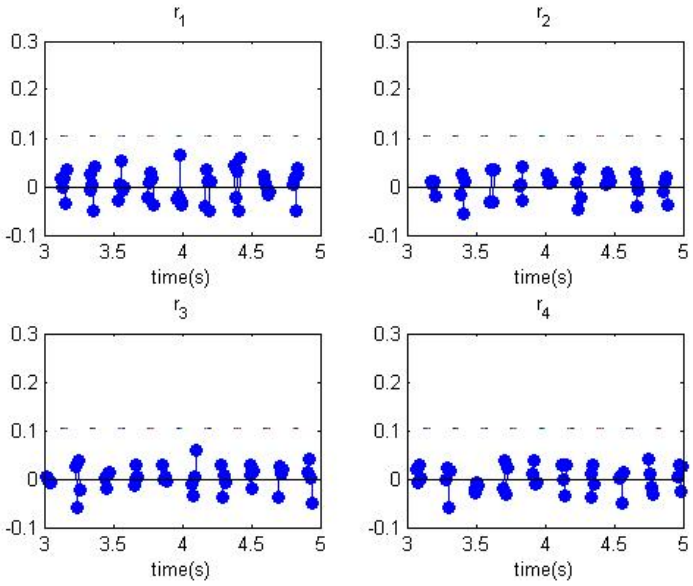



FIGURE 6.32 – generated residuals in normal condition

quence, in third MSPRT test a higher hypothesis comparing to pervious simulation was accepted (Figure 6.35). It means that third sensor must have more access to the medium than other sensors in next communication sequence. A new communication sequence from set of admissible sequences was selected in each 0.02$s$. For next communication sequence selection, last decision of each test was considered in scheduler for calculating length of time slot for accessing each sensor to medium. The scheduler assigned 40% of sequence length to third sensor and 20% to other sensors in next communication sequence that

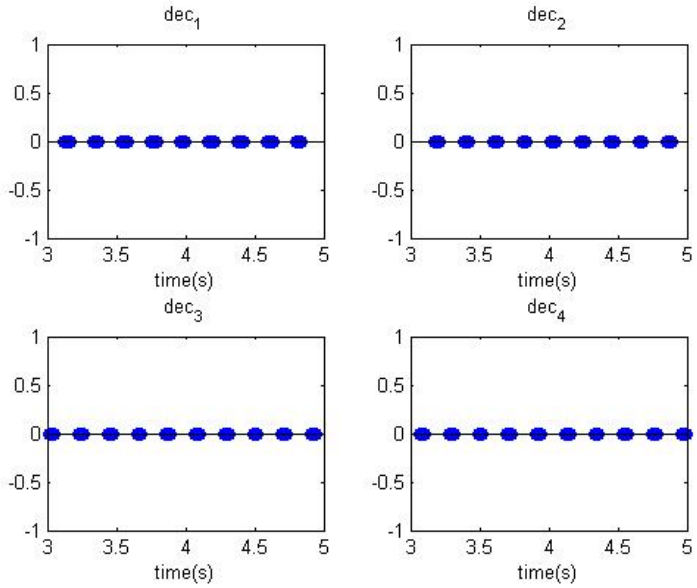



FIGURE 6.33 – accepted hypothesis of each MSPRT test

was started at 3.53$s$ (Figure 6.36). At 4.11$s$ an abnormality was occurred in residual of first motor and its residual diverted from normal condition (Figure 6.34) and second hypothesis was accepted at 4.19$s$. Therefore for next sequence selection at 4.37$s$, a high access time was assigned to this sensor. In new communication sequence first sensor had 50% of sequence time and other sensors 16.7% to send their measurements to RG modules (Figure 6.36 and Figure 6.37).

For comparing fault detection with offline and semi-online scheduling, a fault was introduced in first

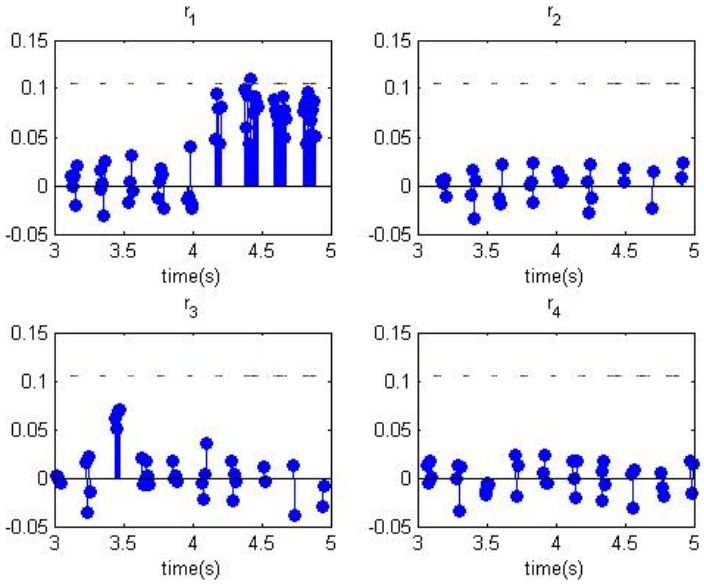



FIGURE 6.34 – generated residuals

motor at 3.1$s$. By using offline periodic scheduling, it was detected at 3.57$s$ but if proposed schedu-

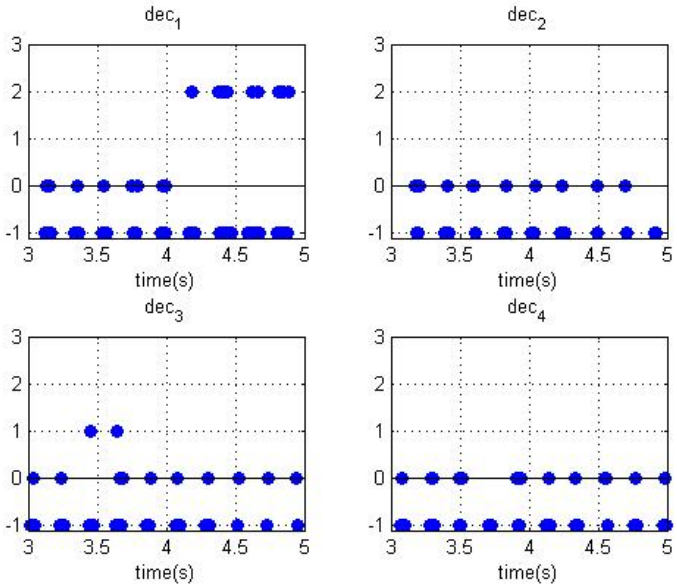



FIGURE 6.35 – accepted hypothesis of each MSPRT test

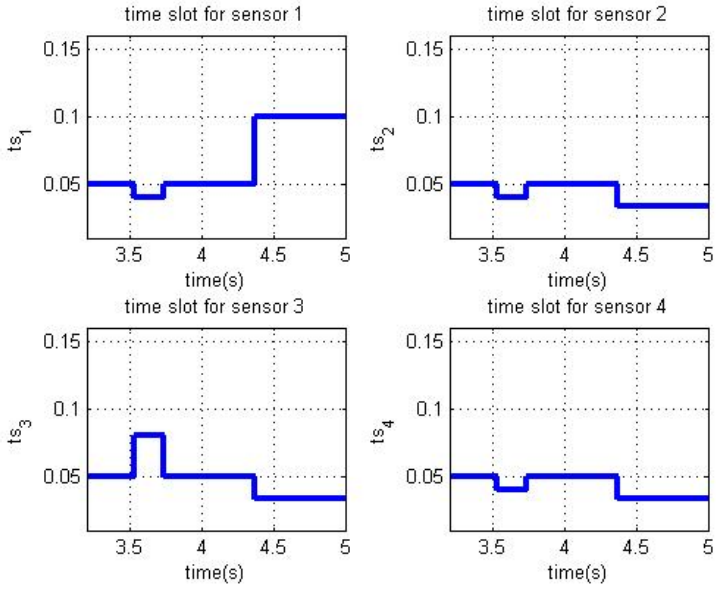



FIGURE 6.36 – Calculated time slot for each sensor

ling is used, fault will be detected at 3.35*s*. Time slot of first sensor and accepted hypotheses in case of semi-online scheduling was illustrated in Figure 6.38.

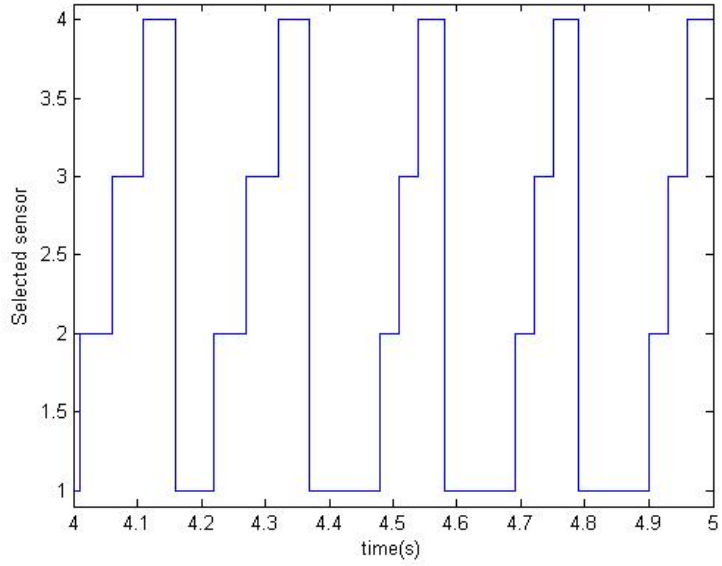



FIGURE 6.37 – Communication sequences

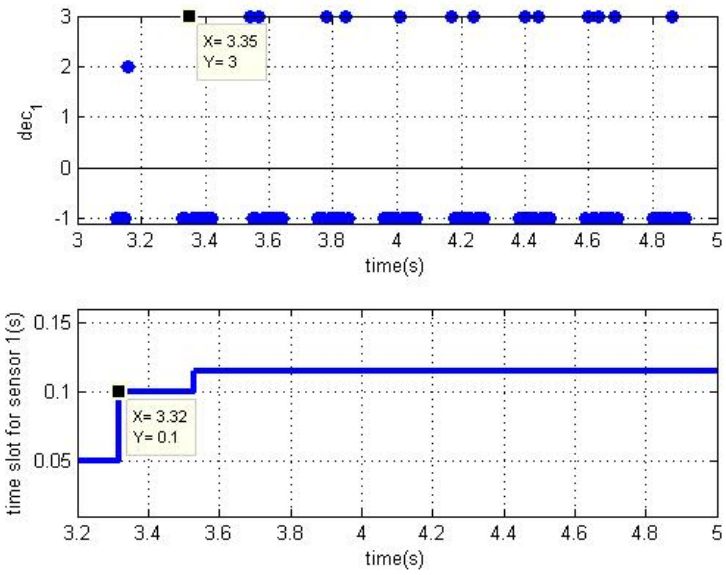



FIGURE 6.38 – Accepted hypotheses and time slot of first sensor in faulty condition

# Chapter 7

# Conclusion and future works

The main concern of the presented work has been the investigation of fault detection and scheduling problems with considering network-induced effects such as medium access constraints and packet dropouts.

First of all, the problem of fault detection and isolation in a networked control system with access limitation was studied. The considered network can provide limited number of channels for transmitting commands and measurements. As a consequence, FDI has not access to all measurements at each sampling instant. Furthermore, due to network limitation packets containing command messages may be lost during message transmission between nodes. Moreover, it is assumed that, there are no available information about characteristic and probability of packet dropout in FDI module. Since networked-induces effects must be considered in FDI design, an extended model for taking into account all these limitations was used. Then, a strategy to develop a set of structured residuals was proposed. Each residual was insensitive to packet dropout of one input also to fault in one sensor. Therefore, it is possible to detect and isolate sensor fault in the system even if there are a unknown and random packet dropouts in one input of the system. However, some assumptions limit the range of application of this method in practice.
By means of static and periodic communication sequence, observer-based and parity space-based residual generator was designed. However, dynamic communication sequence can bring more advantages in fault detection. Other model-based approaches for residual generation of time variant systems can be considered as next step. In this thesis, communication resources were considered as constant parameters. An interesting extension is the generalization of FDI problem of NCS with a communication network with variable resources.

The problems of finding periodic communication sequences that preserve reachability and observability of a linear time invariant system was studied in fourth chapter. Selecting a group of sensors or actuators at each sampling instant for accessing to the medium, has direct effect on observability and reachability of the whole system. A graph-based algorithm for finding all communication sequences which guarantee observability/reachability of extended plant was proposed. Due to using structured system, only information about structure of original system is needed. Therefore, the proposed algorithms can be deals with a NCS with uncertain parameters. In addition, comparing with previous works that studied design of communication sequences, this strategy is simpler and complex mathematical computation is not necessary. As a consequence, it can be applied in systems with large number of sensors and actuators.
By using the proposed algorithm, all periodic communication sequences which preserve reachability/observability

the original system can be found. Objective is to find all communication sequences with minimum necessary resources. However, this algorithm can be extended for finding an optimal communication sequence with predefined communication resource, $\omega$, and sequence length, $T$. Another extension of this algorithm can be proposing a real-time scheduling algorithm. Furthermore, In the FDI context, the structural approach can give more insight dealing with the detectability of system and the generation of communication sequences that preserve it.

From a computer science point of view, offline scheduling algorithms have many advantages. Essentially, they consume few computing resources and do not induce execution overheads. But they cannot guarantee good control and diagnostic performance when unpredictable disturbances occur in the system. Although , online scheduling has not these limitations, its main drawback is that it requires very important computing resources, which make it normally applicable to slow systems. In chapter five, the problem of semi-online scheduling and fault detection was studied. Firstly, a strategy with parity space-based residual generation was proposed. Then based on stability analysis of residual, an extension of this strategy in case of observer-based approach was studied. Communication sequences were found offline then based on the urgency of messages, one communication sequence was chosen from set of stored sequences in scheduler. This scheduling algorithm can be seen as a compromise between the advantages of the online scheduling and those of the offline scheduling. Finally, an adaptation of this strategy in CAN network by using hybrid priority was presented.
The proposed strategy can be considered as a modular solution. Because any part such as RG module, residual classification module, scheduler can be designed separately from others. For example, it can be used for observer-based residual generators as well as parity space residual generators for linear or non-linear systems. Residuals are assumed as Gaussian variables and their variances before and after fault occurrence are considered to be known and constant. Due to these assumptions, a classical multi sequential hypothesis test was proposed. However, it is possible to use an appropriate multi hypothesis test when residual signals have other characteristic.

Drone can be classified as a fast dynamic system. So, it is ideal benchtest for studying effects of the network on performance of closed loop control of the system. In the sixth chapter, the effects of freezing fault in one actuator because of some physical or network problems were studied. After fault occurrence velocity of faulty motor remained constant in its last value before fault occurrence. There was not a reconfiguration method that satisfied all system constraints. So we had to select some high priority constraints from other constraints. This selection was done in function of drone attitude and its position in the air. Then, a fault tolerant algorithm that was included different strategies was proposed. These strategies were satisfied different constraints of the system and guarantee safety conditions of application during of fault occurrence.
In the drone application fast fault detection is no less important than fault tolerance control strategy. In case of fault occurrence, a fast fault detector activates faster reconfiguration module and as a consequence error in the system will be corrected more quickly. Fault diagnosis and scheduling approach which proposed in chapter five was implemented in drone. Some simulation results showed that the semi-online scheduling approach provides faster fault detection in this application.
Research works in our application limit to delay and packet dropouts. Adding a external flow affects performance of the control loop. In order to reduce the network load and thus avoid the uncertainty caused by transmission delays and packet loss, a periodic communication sequence can be employed for the allocation of network resource. Thus, an appreciated controller and fault detection and isolation system should be proposed. In addition, implementation of proposed algorithms for finding communication sequences can be considered as future work in this chapter.

Fault diagnosis and fault-tolerant control of networked control systems are current, active, and fertile

research topics offering interesting challenges in both theory and applications. However, research in this domain is still in progress and the co-design method aiming at integrating the control/diagnostic and scheduling for NCS is a promising topic of research.

# List of figures


2.1 Generic setup of DDC system [Zha01] . . . 14
2.2 A DCS system [Zha01] . . . 14
2.3 shared-network connections . . . 14
2.4 Hierarchical structure in NCSs . . . 15
2.5 Direct structure in NCSs . . . 16
2.6 Factory automation, PSA Production line . . . 18
2.7 Unmanned vehicle navigation . . . 18
2.8 Example of network in a automobile . . . 18
2.9 Delay on a random access network [Zha01] . . . 19
2.10 The model based fault detection . . . 22

3.1 NCS with communication constraints . . . 26
3.2 Observer-based stabilization of NCS . . . 31
3.3 feedback control system with random packet dropouts . . . 40
3.4 Packet dropout in $u_1$ . . . 43
3.5 Residuals of observers without packet dropout . . . 44
3.6 Residuals of observers with packet dropout . . . 44
3.7 Feedback control system . . . 46
3.8 Residuals evolution without packet dropout . . . 49
3.9 Residuals evolution without packet dropout . . . 49
3.10 Periodic communication sequence . . . 50
3.11 Residuals evolution with random packet dropout . . . 51

3.12 Residuals evolution with random packet dropout . . . . . 51

4.1 Digraph associated to system Example 4.3 . . . . . 59
4.2 Digraph associated to system Example 4.4 . . . . . 60
4.3 Digraph associated to system Example 4.5 . . . . . 61
4.4 Digraph associated to system (4.1) . . . . . 62
4.5 Dynamic bipartite graph associated to system (4.1) . . . . . 62
4.6 $\mathscr{B}_4^o$ Dynamic bipartite graph w.r.t. input associated to the structured system (4.14) . . . 64

5.1 communication architecture for distributed process. . . . . 72
5.2 message scheduling architecture . . . . . 75
5.3 mean of residual and its hypothesis in MSPRT test . . . . . 77
5.4 Generated residual in normal condition . . . . . 79
5.5 Accepted hypothesis of each MSPRT test in normal condition . . . . . 79
5.9 Necessary time slot for each subsystem . . . . . 79
5.6 Generated residual . . . . . 80
5.7 Accepted hypothesis of each MSPRT test . . . . . 80
5.8 Communication sequence . . . . . 81
5.10 Accepted hypotheses and necessary time slot of first subsystem . . . . . 81
5.11 CAN data frame [Gmb91] . . . . . 85
5.12 Illustration of the bus arbitration in CAN networks . . . . . 85
5.13 Frame identifier structure in hybrid priority [JM07] . . . . . 86
5.14 Relation of Accepted hypothesis in MSPRT test and Dynamic priority . . . . . 87

6.1 Quadrotor benchtest . . . . . 91
6.2 Electronic architecture of drone . . . . . 92
6.3 produced force and reactive torque in drone . . . . . 93
6.4 drone concept motion . . . . . 93
6.5 Body-fixed and inertial frame . . . . . 94
6.6 The Euler Angles [Fos94] . . . . . 96

6.7 Roll, Pitch and yaw angles[GC08] . . . 97
6.8 MAG3 Orientation diagram [MEM] . . . 100
6.9 inertial and body frame of drone . . . 100
6.10 Attitude estimation . . . 104
6.11 Drone close loop . . . 104
6.12 The reconfiguration problem . . . 105
6.13 Fault reconfiguration in drone . . . 106
6.14 drone attitude in faulty and fault free condition . . . 107
6.15 Angular velocities of motors . . . 108
6.16 Total thrust generated by four motors . . . 108
6.17 linear velocity of drone due to network fault . . . 108
6.18 Fault reconfiguration in drone . . . 109
6.19 drone attitude (First Strategy) . . . 110
6.20 Angular velocities of motors (First Strategy) . . . 110
6.21 Total thrust generated by four motors (First Strategy) . . . 110
6.22 drone attitude( Second Strategy) . . . 111
6.23 Angular velocities of motors (Second Strategy) . . . 111
6.24 Evolution of the total thrust in second strategy (Second Strategy) . . . 111
6.25 drone attitude(Third Strategy) . . . 112
6.26 Angular velocities of motors (Third Strategy) . . . 112
6.27 Evolution of the total thrust in second strategy (Third Strategy) . . . 112
6.28 Fast fault detection in case of freezing fault . . . 113
6.29 Slow fault detection in case of freezing fault . . . 114
6.30 Message scheduling and fault detection architecture . . . 115
6.31 Mean of residual and its hypothesis in MSPRT test . . . 116
6.32 generated residuals in normal condition . . . 117
6.33 accepted hypothesis of each MSPRT test . . . 118
6.34 generated residuals . . . 118

6.35 accepted hypothesis of each MSPRT test . . . . . 119

6.36 Calculated time slot for each sensor . . . . . 119

6.37 Communication sequences . . . . . 120

6.38 Accepted hypotheses and time slot of first sensor in faulty condition . . . . . 120

# List of tables

3.1 Inference Matrix . . . . . 42
3.2 Sensor fault and packet dropout sensitivity . . . . . 43
3.3 Sensor and packet dropout sensitivity . . . . . 48

6.1 UAV classification . . . . . 90
6.2 notation and their values used in model of drone . . . . . 103

# Bibliography


[Abr70] N. Abramson, *The Aloha system-another alternative for computer communications*, Proceedings of Fall Joint Computer Conference, AFIPS Conference (1970).

[AS03] B. Azimi-Sadjadi, *Stability of networked control systems in the presence of packet losses*, In Proceedings of the IEEE Conference on Decision and Control **1** (2003), 676–681.

[ASY08] Christophe Aubrun, Dominique Sauter, and Joseph Yame, *Fault diagnosis of networked control systems*, International Journal of Applied Mathematics and Computer Science **18** (2008), no. 4, 525–537.

[Aus10] Reg Austin, *Unmanned aircraft systems*, Wiley, 2010.

[AW96] Karl Johan Astrom and Bjorn Wittenmark, *Computer-controlled systems : Theory and design*, Prentice Hall ; 3 edition, 1996.

[Bas88] M. Basseville, *Detecting changes in signals and systems - a survey*, Automatica **3** (1988), no. 3, 309–326.

[Bas03] Michele Basseville, *Model based statistical signal processing and desicison theoretical approaches to monitoring*, In Procceding of 1st IFAC Symposium on SAFEPROCESS (2003).

[BC96] Sergio Bittanti and Patrizio Colaneri, *Analysis of discrete-time linear periodic systems*, Digital Control and Signal Processing Systems and Techniques (Cornelius T. Leondes, ed.), Control and Dynamic Systems, vol. 78, Academic Press, 1996, pp. 313 – 339.

[BC08] S. Bittanti and P. Colaneri, *Periodic systems : Filtering and control*, Springer ; 1 edition, 2008.

[BF09] T. Boukhobza and F.Hamelin, *State and input observability recovering by additional sensor implementation : A graph-theoretic approach*, Automatica **45** (2009), no. 7, 1737 – 1742.

[BGc10] Mohamed El Mongi Ben Gaid and Arben Çela, *Trading quantization precision for update rates for systems with limited communication in the uplink channel*, Automatica **46** (2010), no. 7, 1210–1214.

[BGL09] C. Berbra, S. Gentil, and S. Lesecq, *Hybrid priority scheme for networked control quadrotor*, 17th Mediterranean Conference on Control and Automation, Greece (2009), 516–521.

[BH08] T. Boukhobza and F. Hamelin, *Observability analysis for networked control systems : a*

*graph theoretic approach*, 17th IFAC World Congress, Seoul : Korea, Republic of (2008) (2008).

[BHMM07] T. Boukhobza, F. Hamelin, and S. Martinez-Martinez, *State and input observability for structured linear systems : a graph-theoretic approach*, Automatica **43** (2007), no. 7, 1204–1210.

[Bit86] Sergio Bittanti, *Deterministic and stochastic linear periodic systems*, Time Series and Linear Systems (Sergio Bittanti, ed.), Lecture Notes in Control and Information Sciences, vol. 86, Springer Berlin / Heidelberg, 1986, pp. 141–182.

[BL00] R.W. Brockett and D. Liberzon, *Quantized feedback stabilization of linear systems*, IEEE Transactions on Automatic Control **45** (2000), no. 7, 1279–1289.

[BL02] E. K. Boukas and Z. K. Liu, *Deterministic and stochastic time delay systems*, Springer, New York, 2002.

[Bro95] R. W. Brockett, *Stabilization of motor networks*, 34th IEEE Conference on Decision and Control, 1995, pp. 1484–1488.

[BSGL09] Cedric Berbra, Daniel Simon, Sylviane Gentil, and Suzanne Lesecq, *Hardware in the loop networked control and diagnosis of a quadrotor drone*, 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes,Safeprocess, Barcelone :, 2009.

[BTX03] J.S. Baras, Xiaobo Tan, and Wei Xi, *Jointly optimal quantization, estimation, and control of hidden markov chains*, In Proceedings of the IEEE Conference on Decision and Control (2003), 1098–1103.

[BV94] C.W. Baum and V.V. Veeravalli, *Sequential testing of many simple hypotheses with independent observations*, IEEE Transactions on Information Theory **40** (1994), no. 6, 1994–2007.

[BV96] Steve Biegacki and Dave VanGompel, *The application of DeviceNet in process control*, ISA Transactions **35** (1996), no. 2, 169 – 176.

[CBL09] S. Gentil C. Berbra and S. Lesecq, *Hybrid priority scheme for networked control quadrotor*, IEEE 17th Mediterranean Conference on Control and Automation Makedonia Palace, Thessaloniki, Greece, 2009, pp. 516–521.

[CCDYA07] Commault Christian, Jean-Michel Dion, and Sameh Yacoub Agha, *Sensor classification for the Fault Detection and Isolation problem*, Proceedings of the 3rd IFAC Symposium on System, Structure and Control, SSSC 07 3rd IFAC Symposium on System, Structure and Control, SSSC 07, IFAC, 2007.

[CDT05] C. Commault, J.M. Dion, and D.H. Trinh, *Observability recovering by additional sensor implementation in linear structured systems*, IEEE Conference on Decision and Control, and the European Control Conference, Spain, 2005.

[CF96] T.W. Chen and B. Francis, *Optimal sampled-data control systems*, Springer, 1996.

[Cho92] J. C.K. Chou, *Quaternion kinematic and dynamic differential equations*, IEEE Transactions on robotics and automation **8** (1992), no. 1, 53–64.

[CLD] Pedro Castillo, Rogelio Lozano, and Alejandro E. Dzul, *Modelling and control of mini-flying machines*, Springer ; 1 edition (June 1, 2005).

[CP99] J. Chen and R.J. Patton, *Robust model-based fault diagnosis for dynamic systems*, Kluwer Academic Publishers, 1999.

[CPZ96] J. Chen, R.J. Patton, and H.Y. Zhang, *Design of unkown input observers and robust fault-detection filters*, International Journal of Control **63** (1996), no. 1, 85–105.

[CR01] Peter Corke and Peter Ridley, *Steering kinematics for a center-articulated mobile robot*, IEEE transactions on Robotics and Automations **17** (2001), no. 2, 215–218.

[CW84] E. Y. Chow and A. S. Willsky, *Analytical redundancy and the design of robust failure detection systems*, IEEE Trans. Contr. **AC-29** (1984), 603–617.

[DCvdW03] Jean-Michel Dion, Christian Commault, and Jacob van der Woude, *Generic properties and control of linear structured systems : a survey*, Automatica **39** (2003), no. 7, 1125 – 1144.

[DF91] X. Ding and P.M. Frank, *Frequency domain approach and threshold selector for robust model-based fault detection and isolation*, In Procceding of 1st IFAC Symposium on SAFEPROCESS (1991), 307–312.

[DF98] S.X. Ding and P.M. Frank, *Fault detection via optimally robust detection filters*, In Procceding of 28st IEEE Conference on Decision and Control **2** (1998), 1767–1772.

[Din08] Steven X. Ding, *Model based fault diagnosis technique,design schemes, algorithms, and tools*, Springer, 2008.

[DTV99] V.P. Dragalin, A.G. Tartakovsky, and V.V. Veeravalli, *Multihypothesis sequential probability ratio tests. i. asymptotic optimality*, IEEE Transactions on Information Theory **47** (1999), no. 7, 2448–2461.

[DTV00] Vladimir P. Dragalin, Alexander G. Tartakovsk, and Venugopal V. Veeravalli, *Multihypothesis sequential probability ratio tests. II. accurate asymptotic expansions for the expected sample size*, IEEE Transactions on Information Theory **46** (2000), no. 4, 1366–1383.

[eH01] Mohamed Gad el Hak, *The mems handbook*, CRC Press, 2001.

[eH06] Mohamed Gad el Hak, *Mems introduction and fundamentals*, Taylor and Francis Group, 2006.

[Fad01] M. Sami Fadali, *Robust observer-based fault detection for periodic systems*, Proceeding of the American Control Conference, 2001, pp. 25–27.

[FCN03] M.S. Fadali, P. Colaneri, and M. Nel, *$H_2$ robust fault estimation for periodic systems*, Proceeding of the American Control Conference, vol. 4, 2003, pp. 2973–2978.

[FD94] Paul M. Frank and Xianchun Ding, *Frequency domain approach to optimally robust residual generation and evaluation for model-based fault diagnosis*, Automatica **30** (1994), no. 5, 789 – 804.

[FDM00] P.M. Frank, S.X. Ding, and T. Marcu, *Model-based fault diganosis in technical processes*, Transactions of the Institute of Measurement and Control **22** (2000), 57–110.

[Fos94] F. Fossen, *Guidance and control of ocean vehicles*, Wiley, August 1994.

[Fra90] P.M. Frank, *Fault diagnosis in dynamic systems using analytical and knowledge based redundancy, a survey and some new results*, Automatica **26** (1990), no. 3, 459–474.

[FYZ07] Huajing Fang, Hao Ye, and Maiying Zhong, *Fault diagnosis of networked control systems*, Annual Reviews in Control **31** (2007), no. 1, 55 – 68.

[Gai06] M. Ben Gaid, *Optimal scheduling and control for distributed real-time systems*, Ph.D. thesis, Université d'Evry Val d'Essonne, 2006.

[GC08] J.F. Guerrero-Castellanos, *Estimation de l'attitude et commande bornée en attitude d'un corps rigide : Application à un mini helicoptère à quatre rotors*, Ph.D. thesis, Joseph Fourier-Grenoble University, 2008.

[GCH06] M.M. Ben Gaid, A. Cela, and Y. Hamam, *Optimal integrated control and scheduling of networked control systems with communication constraints : application to a car suspension system*, IEEE Transactions on Control Systems Technology **14** (2006), no. 4, 776 – 787.

[Ger97] J. Gertler, *Fault detection and isolation using parity relations*, Control engineering practice **5** (1997), no. 5, 653–661.

[Ger98] Janos Gertler, *Fault dectection and diagnosis in engineering systems*, Marcel Dekker, New York, NY, 1998.

[GF88] W. Ge and C. Fang, *Detection of faulty components via robust observation*, International Journal of Control **47** (1988), no. 2, 581–599.

[Gmb91] Bosch GmbH, *CAN specification 2.0(a)*, Tech. report, [online] available : www.semiconductors.bosch.de/pdf/can2spec.pdf, 1991.

[Gro88] MAP/TOP Users Group, *Manufacturing automation protocol specification - version 3.0*.

[GW02] L. Bushenell G. Walsh, H. Ye, *Stability analysis of networked control systems*, IEEE Transactions on Control Systems Technology **10** (2002), no. 3, 438–446.

[Hes09] Joao P. Hespanha, *Linear systems theory*, Princeton University Press ; annotated edition, August 2009.

[HMLO02] T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski, *Dynamic modelling and configuration stabilisation for an x4-flyer*, IFAC World Congress, Barcelona, Spain (2002).

[HN08] D. Huang and S.K. Nguang, *State feedback control of uncertain networked control systems with random time delays*, IEEE Transactions on Automatic Control **53** (2008), no. 3, 829–834.

[Hri00] D. Hristu, *Stabilization of lti systems with communication constraints*, In Proceedings of the American Control Conference (2000), 2342–2346.

[Hsi04] TT Hsieh, *Using sensor networks for highway and traffic applications*, IEEE Potentials **23** (2004), no. 2, 13–16.

[HT02] C. N. Hadjicostis and R. Touri, *Feedback control utilizing packet dropping network links*, In Proceedings of the IEEE Conference on Decision and Control **2** (2002), 1205–1210.

[HVK02] Hristu-Varsakelis and P.R. Kumar, *Interrupt-based feedback control over a shared communication medium*, In Proceedings of the 41st IEEE conference on decision and control (2002), 3223–3228.

[HWJZ08] X. He, Z. Wang, Y.D. Ji, and D.H. Zhou, *Network-based fault detection for discrete time state-delay systems : A new measurement model*, International Journal of Adaptive Control and Signal Processing **22** (2008), no. 5, 510–528.

[HWZ09] Xiao He, Zidong Wang, and D.H. Zhou, *Robust fault detection for networked systems with communication delay and data missing*, Automatica **45** (2009), no. 11, 2634 – 2639.

[HY04] S. Hu and W.Y. Yan, *Stability of networked control systems : Analysis of packet dropping*, In Proceedings of Control, Automation, Robotics and Vision Conference **1** (2004), 304–309.

[ICC06] C. Ionete, C. CelaIonete, and Cela, *Structural properties and stabilization of NCS with medium access constraints*, In Proceedings of the IEEE Conference on Decision and Control (2006), 1141–1146.

[IF02] Hideaki Ishii and Bruce A. Francis, *Limited data rate in control systems with networks limited data rate in control systems with networks*, Springer, 2002.

[Int99] ControlNet International, *ControlNet specification,release 2.0, including errata 2*.

[Ise05] R. Isermann, *Model-based fault-detection and diagnosis - status and applications*, Annual Reviews in Control **29** (2005), no. 1, 71 – 85.

[Ise11] Rolf Isermann, *Fault-diagnosis applications*, Springer Berlin Heidelberg, 2011.

[ISSC09] Iman Izadi, Sirish L. Shah, David S. Shook, and Tongwen Chen, *An introduction to alarm analysis and design*, In Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (2009).

[JBN06] A. Johansson, M. Bask, and T. Norlander, *Dynamic threshold generators for robust fault detection in linear system with parameter uncertainty*, Automatica **42** (2006), no. 7, 1095–1106.

[Jia05] Jin Jiang, *Fault-tolerant control systems-an introductory overview*, ACTA automatica **31** (2005), no. 1, 161–174.

[JM07] Guy Juanole and Gérard Mouney, *Networked control systems :definition and analysis of a hybrid priority scheme for the message scheduling*, 13th IEEE international Conference on Embedded and Real-time Computing Systems and applications, 2007, pp. 267 – 274.

[JMC08] Guy Juanole, Gerard Mouney, and Christophe Calmettes, *On different priority schemes for the message scheduling in networked control systems*, 16th Mediterranean Conference on Control and Automation, 2008, pp. 1106–1111.

[JMCP] Guy Juanole, Gerard Mouney, Christophe Calmettes, and Marek Peca, *Fundamental considerations for implementing control systems on a CAN network*, 1, vol. 6, 6th IFAC international Conference on Fieldbus Systems and their Applications.

[JMS+10] Guy Juanole, Gerard Mouney, Dominique Sauter, Christophe Aubrun, and Christophe Calmettes, *Decision making improvement for diagnosis in networked control systems based on*

*dynamic message scheduling*, IEEE 18th Mediterranean Conference on Control and Automation (MED), 2010, pp. 280 – 285.

[Jor95] J. R. Jordan, *Serial networked field instrumentation*, Wiley, New York, 1995.

[KCM09] Dong-Sung Kima, Dong-Hyuk Choi, and Prasant Mohapatra, *Real-time scheduling method for networked discrete control systems*, Control engineering practice **17** (2009), no. 5, 564–570.

[KH63] R. E. Kalman and B. Ho, *Controllability of linear dynamical systems*, Contributions to Differential Equations **1** (1963), 188–213.

[KK00] K.Zuberi and K.Shin, *Design and implementation of efficient message scheduling for controller area network*, IEEE Transactions On Computer **49** (2000), no. 2, 182–188.

[KNSS05] M.J. Khosrowjerdi, R. Nikoukhah, and N. Safari-Shad, *Fault detection in a mixed $H_2/H_\infty$ setting*, IEEE Transactions on Automatic Control **50** (2005), no. 7, 1063–1068.

[KS94] H. Kim and K. G. Shin, *On the maximum feedback delay in a linear/nonlinear control system with input disturbances caused by controller-computer failures*, IEEE Transactions on Control Systems Technology **2** (1994), no. 2, 110–122.

[Law97] W. Lawrenz, *CAN system engineering,from theory to practical applications*, Springer, New York, 1997.

[LB04] Keyong Li and J. Baillieul, *Robust quantization for digital finite communication bandwidth (DFCB) control*, IEEE Transactions on Automatic Control **49** (2004), no. 9, 1573–1584.

[LCY05] H. Liu, Y. Cheng, and H. Ye, *A combinative method for fault detection of networked control systems*, In Proceedings of the 20th IAR/ACD Annual Meeting (2005), 59–63.

[Li09] Wei Li, *Observer-based fault detection of technical systems over networks*, Ph.D. thesis, University of Duisburg-Essen, 2009.

[Lin74] C.T Lin, *Structural controllability*, IEEE Transaction on Automatic Control **19** (1974), no. 3, 201–208.

[LM99] D. Liberzon and A. S. Morse, *Basic problems in stability and design of switched systems*, IEEE Control Systems Magazine **19** (1999), no. 5, 59–70.

[LM07] S. Longhi and A. Monteriu, *A geometric approach to fault detection of periodic systems*, 46th IEEE conference on decision and control, 2007, pp. 6376–6382.

[LMT01] Feng-Li Lian, James R. Moyne, and Dawn M. Tilbury, *Performance evaluation of control networks : Ethernet, ControlNet, and DeviceNet*, IEEE Control systems magazine **21** (2001), no. 1, 66–83.

[LSA06] S. Li, D. Sauter, and C. Aubrun, *Robust fault isolation filter design for networked control systems*, In Proceedings of the 11th IEEE International Conference on Emerging Technologies and Factory Automation (2006), 681–688.

[LVL+05] Wai-Lun Danny Leung, Rangsarit Vanijjirattikhan, Zheng Li, Le Xu, Tyler Richards, and Bulent Ayhan, *Intelligent space with time-sensitive application*, Proceedings of the

2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (2005), 1413–1418.

[LW99] X.Rong Li and Lu Wang, *Fault detection using sequential probability ratio test*, Power Engineering Society 1999 Winter Meeting, IEEE **2** (1999), 938 – 943.

[LW08] Li Li and Fei-Yue Wang, *Control and communication synthesis in networked control systems*, The International Journal of intelligent control and systems **13** (2008), no. 2, 81–88.

[LWV86] Xi-Cheng Lou, Alan S. Willsky, and George C. Verghese, *Optimally robust redundancy relations for failure detection in uncertain systems*, Automatica **22** (1986), no. 3, 333 – 344.

[LZ09] Yueyang Li and Maiying Zhong, *Parity space-based fault detection for linear time-varying systems with multiple packet dropouts*, Fourth International Conference on Innovative Computing, Information and Control (2009), 236–239.

[MA04] L. A. Montestruque and P. J. Antsaklis, *Stability of model-based networked control systems with time-varying transmission times*, IEEE Transactions on Automatic Control **49** (2004), no. 9, 1562–1572.

[Mar08] Sinuhe Martinez Martinez, *Analyse des propriétés structurelles d'observabilité de l'état et de l'entrée inconnue des systèmes linéaires par approche graphique(in french)*, Ph.D. thesis, Nancy univeersity, 2008.

[M.B08] M.Bento, *Unmanned aerial vehicles : An overview*, InsideGNSS (2008), 54–61.

[MC05] Y. Tipsuwan M.Y. Chow, *Time sensitive network-based control systems and applications*, IEEE IES Network Based Control Newsletter **5** (2005), no. 2, 13–18.

[ME00] R. S. Mangoubi and A.M. Edelmayer, *Model-based fault detection : The optimal past, the robust present and a few thoughts on the future*, SAFEPROCESS, 2000, pp. 64–75.

[MEM] MEMSense, *Triaxial magnetometer, accelerometer and gyroscope analog inertial sensor*.

[MH02] R. Mahony and T. Hamel, *Adaptive compensation of aerodynamic effects*, European Journal of Control EJC (2002).

[MIC04] MICOCHIP, *Pic18f2455/2550/4455/4550 :datasheet*, Micochip Technology Inc. (2004).

[MJS09] Zehui Mao, Bin Jiang, and Peng Shi, *Protocol and fault detection design for nonlinear networked control systems*, IEEE Transactions on Circuits and Systems II : Express Briefs **56** (2009), no. 3, 255–259.

[MKBO10] I. Maza, K. Kondak, M. Bernard, and A. Ollero, *Multi-uav cooperation and control for load transportation and deployment*, Journal of Intelligent and Robotic Systems **57** (2010), 417–449.

[MM04] Michael M. Mirabito and Barbara L. Morgenstern, *New communications technologies : Applications, policy, and impact*, Focal Press, 2004.

[Mur87] K. Murota, *System analysis by graphs and matroids,*, Springer-Verlag, New York, U.S.A, 1987.

[NAR88] A. Emami Naeini, M. Akhter, , and S. Rock, *Effect of model uncertainty on failure detection : The threshold selector*, IEEE Transactions on Automatic Control **33** (1988), 1106–1115.

[Nat00] M. Di Natale, *Scheduling the CANbus with earliest deadline techniques*, In Proceedings of the 21st IEEE Real-Time Systems Symposium, 2000, pp. 259 – 268.

[NT04] D. Nesic and A.R. Teel, *Input-to-state stability of networked control systems*, Automatica **40** (2004), 2121–2128.

[ODC05] Robert Osiander, M. Ann Garrison Darrin, and John L. Champion, *Mems and microstructures in aerospace applications*, CRC Press, October2005.

[PDGC09] Tao Peng, S.X. Ding, Weihua Gui, and Jie Chen, *A parity space approach to fault detection for networked control system via optimal measurement selection*, 48th IEEE Conference on Decision and Control, 2009, pp. 6994–6999.

[PF07a] Tristan Perez and Thor I. Fossen, *Kinematic models for manoeuvring and seakeeping of marine vessels*, Modeling, Identification and Control **28** (2007), no. 1, 19–30.

[PF07b] Tristan Perez and Thor Inge Fossen, *Kinematic models for manoeuvring and seakeeping of marine vessels*, Modeling, Identification and Control **28** (2007), no. 1, 19–30.

[PT06] C. Peng and Y.C. Tian, *Robust $H_\infty$ control of networked control systems with parameter uncertainty and state-delay*, European Journal of Control **12** (2006), no. 5, 471–480.

[PTT08] Chen Peng, Yu-Chu Tian, and Moses O. Tade, *State feedback controller design of networked control systems with interval time-varying delay and nonlinearity*, International Journal of Robust and Nonlinear Control **18** (2008), no. 12, 1285–1301.

[PYN06] Yanbin Pang, Shuang-Hua Yang, and Hirokazu Nishitanic, *Analysis of control interval for foundation fieldbus-based control systems*, ISA Transactions **45** (2006), no. 3, 447 – 458.

[Rei88] K.J. Reinschke, *Multivariable control, a graph-theoretic approach*, Berlin : Springer, 1988.

[Ric03] Jean-Pierre Richard, *Time-delay systems : an overview of some recent advances and open problems*, Automatica **39** (2003), no. 10, 1667 – 1694.

[RMY09] A. Gupta Rachana and Chow Mo-Yuen, *Networked control system : Overview and research trends*, IEEE Transactions on Industrial Electronics **57** (2009), no. 7, 2527–2535.

[RN99] M.L. Rank and H. Niemann, *Norm based design of fault detectors*, International Journal of Control **72** (1999), no. 9, 773–783.

[Rob58] Alfred C. Robinson, *on the use of quaternion in simulation of rigid-body motion*, Aeronautical Research Laboratory (1958).

[RS08] Steven Rasmussen and Tal Shima, *Uav cooperative decision and control*, Society for Industrial Mathematics, first edition (December 16, 2008).

[Rug96] Wilson J. Rugh, *Linear system theory*, Prentice Hall, 2nd edition, 1996.

[Sal05] Antonio Sala, *Computer control under time-varying sampling period : An LMI gridding approach*, Automatica **41** (2005), no. 12, 2077 – 2082.

[Sei01] P.J. Seiler, *Coordinated control of unmanned aerial vehicles*, Ph.D. thesis, niversity of California at Berkeley, 2001.

[She92] TB Sheridan, *Telerobotics, automation, and human supervisory control*, The MIT Press, Cambridge, MA (1992).

[Shu93] M.D. Shulster, *Survey of attitude representations*, Journal of the Astronautical Sciences **41** (October 1993), 439–517.

[Sie85] D. Siegmund, *Sequential analysis : Test and confidence intervals*, Springer-Verlag, 1985.

[SLA09] Dominique Sauter, Shanbin Li, and Christophe Aubrun, *Robust fault diagnosis of networked control systems*, International Journal of Adaptive Control and Signal Processing **23** (2009), no. 8, 722–736.

[Spu00] C. Spurgeon, *Ethernet : The definitive guide*, O'Reilly and Associates, Inc. (Sebastopol,CA, 2000).

[SQ11] Ye-Guo SUN and Shi-Yin QIN, *Stability and stabilization of networked control systems with bounded packet dropout*, Acta Automatica Sinica **37** (2011), no. 1, 113 – 118.

[SRAB04] R. Sandoval-Rodriguez, C.T. Abdallah, and R.H. Byrne, *Effects of quantization, saturation, and sampling time in multi output systems*, In Proceedings of the IEEE Conference on Decision and Control **4** (2004), 4509–4514.

[SS03] S.C. Smith and P. Seiler, *Estimation with lossy measurements :jump estimator for jump systems*, IEEE Transactions on Automatic Control **48** (2003), no. 12, 2163–2171.

[SSF+04] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, *Kalman filtering with intermittent observations*, IEEE Transactions on Automatic Control **49** (2004), no. 9, 1453–1464.

[SYB09] Dominique Sauter, Joseph Yame, and Taha Boukhobza, *Design of fault isolation filters under intermittent communication*, 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (2009).

[Tar88] A.G. Tartakovskii, *Sequential testing of many simple hypotheses with independent observations*, Probl. Inform. Transm. **4** (1988), no. 4, 299–309.

[TB94] K. Tindell and A. Burns, *Guaranteeing message latencies on control area network (CAN)*, In Proceedings of the 1st International CAN Conference (1994).

[TBW95] K. Tindell, A. Burns, and A.J. Wellings, *Calculating controller area network (CAN) message response times*, Control Engineering Practice **3** (1995), no. 8, 1163–1169.

[Tew07] Ashish Tewari, *Atmospheric and space flight dynamics*, Modeling and Simulation in Science, Engineering and Technology, Birkhäuser Boston, 2007.

[TGLT07] A. Tanwani, S. Gentil, S. Lesecq, and J. M. Thiriet, *Experimental networked embedded mini drone - part ii. distributed fdi*, European Control Conference,Kos, Greece, 2007.

[TLB+91] J.P. Thomesse, P. Lorenz, J.P. Bardinet, P. Leterrier, and T. Valentin, *Factory instrumentation protocol : Model, products, and tools*, Control Engineering **38** (1991), no. 12, 65–67.

[TNP02] D. Theilliol, H. Noura, and J.P. Ponsart, *Fault diagnosis and accommodation of three-tank system based on analytical redundancy*, ISA Transaction **41** (2002), 365–382.

[TNT07] M. Tabbara, D. Nesic, and A.R. Teel, *Stability of wireless and wireline networked control systems*, IEEE Transactions on Automatic Control **52** (2007), no. 9, 1615–1630.

[vdW00] J.W. van der Woude, *The generic number of invariant zeros of a structured linear system*, SIAM Journal of Control and Optimization **38** (2000), no. 1, 1–21.

[VGA+09] Nikolai Vatanski, Jean-Philippe Georges, Christophe Aubrun, Eric Rondeau, and Sirkka-Liisa Jämsä-Jounela, *Networked control with delay measurement and estimation*, Control Engineering Practice **17** (2009), no. 2, 231 – 244.

[Wal45] A. Wald, *Sequential tests of statistical hypotheses*, The Annals of Mathematical Statistics **16** (1945), no. 2, 117–186.

[WC07] J. Wu and T. Chen, *Design of networked control systems with packet dropouts*, IEEE Transactions on Automatic Control **52** (2007), no. 7, 1314–1319.

[Wil76] A. S. Willsky, *A survey of design methods for failure detection in dynamic systems*, Automatica **12** (1976), no. 6, 601–611.

[WIL86] J.L. WILLEMS, *Structural controllability and observability*, Systems and control letters **8** (1986), no. 1, 5–12.

[Wil02] A. Willig, *A new class of packet and bit-level models for wireless channels*, 13th IEEE Intl. Symp. on Personal, Indoor and Mobile Radio Communications, 2002.

[WL10] Fei-Yue Wang and Derong Liu, *Networked control systems : Theory and applications*, Springer London Ltd, octobre 2010.

[WY01] Gregory Walsh and Hong Ye, *Scheduling of networked control systems*, IEEE Control Systems Magazine (2001), 57–65.

[WYB02] Gregory Wash, Hong Ye, and Linda Bushnell, *Stability analysis of networked control systems*, IEEE Transactions on Control Systems Technology **10** (2002), no. 3, 438–446.

[WYCW06] Y.Q. Wang, H. Ye, Y. Cheng, and G. Z. Wang, *Fault detection of NCS based on eigendecomposition and pade approximation*, In Proceedings of the IFAC Symposium SAFEPROCESS (2006), 937–941.

[WYD+09] Yongqiang Wang, Hao Ye, Steven X. Ding, Guizeng Wang, and Donghua Zhou, *Residual generation and evaluation of networked control systems subject to random packet dropout*, Automatica **45** (2009), no. 10, 2427 – 2434.

[WYDW09] Yong-Qiang WANG, Hao YE, Steven DING, and Gui-Zeng WANG, *Fault detection of networked control systems subject to access constraints and random packet dropout*, Acta Automatica Sinica **35** (2009), no. 9, 1235–1239.

[WYL07] Jian Liang Wang, Guang-Hong Yang, and Jian Liu, *An LMI approach to $H_$ index and mixed $H_/H_\infty$ fault detection observer design*, Automatica **43** (2007), no. 9, 1656 – 1665.

[WYW06] Y.Q. Wang, H. Ye, and G.Z. Wang, *A new method for fault detection of networked control*

*systems*, In Proceedings of the 1st IEEE Conference on Industrial Electronics and Applications, Singapore, Republic of Singapore (2006).

[XHH00] L. Xiao, A. Hassibi, and J. How, *Control with random communication delays via a discrete-time jump system approach*, In Proceedings of the American Control Conference **3** (2000), 2199–2204.

[Xu06] Y. Xu, *Communication scheduling methods for estimation over networks*, Ph.D. thesis, University of California, 2006.

[YD04] H. Ye and S.X. Ding, *Fault detection of networked control systems with network-induced delay*, In Proceedings of the 8th International Conference on Control Automation,Robotics and Vision (2004), 294–297.

[YHM06] H.C. Yan, X.H. Huang, and M.Wang, *Delay-dependent stability and stabilization criteria of networked control systems with multiple time-delays*, Journal of Control Theory and Applications **4** (2006), no. 4, 321–326.

[YMF03] J. Yepez, P. Marti, and J.M. Fuertes, *Control loop scheduling paradigm in distributed control systems.*, vol. 2, IEEE Annual Conference of the Industrial Electronics Society, 2003. IECON '03. The 29th, 2003, pp. 1441 – 1446.

[YRLW06] H. Ye, R.He, H. Liu, and G.Z. Wang, *A new approach for fault detection of networked control systems*, In Proceedings of the IFAC 14th Symposium on System Identification (2006), 654–659.

[YWC04] M. Yu, L. Wang, and T. Chu, *An LMI approach to networked control systems with data packet dropout and transmission delays*, In Proceedings of the IEEE Conference on Decision and Control **4** (2004), 3545–3550.

[YWD04] H. Ye, G.Z. Wang, and S.X. Ding, *A new parity space approach for fault detection based on stationary wavelet transform*, IEEE Transactions on Automatic Control **49** (2004), no. 2, 281–287.

[ZBB94] Q. Zhang, M. Basseville, and A. Benveniste, *Early warning of slight changes in systems*, Automatica **30** (1994), no. 1, 95 – 113.

[ZBP01] W. Zhang, M. S. Branicky, and S. M. Phillips, *Stability of networked control systems*, IEEE Control Systems Magazine **21** (2001), no. 1, 84–99.

[ZD07a] P. Zhang and S.X. Ding, *Disturbance decoupling in fault detection of linear periodic systems*, Automatica **43** (2007), no. 8, 1410–1417.

[ZD07b] Ping Zhang and Steven X. Ding, *Fault detection of networked control systems with limited communication*, Fault Detection, Supervision and Safety of Technical Processes 2006, Elsevier Science Ltd, 2007, pp. 1074 – 1079.

[ZD08] Ping Zhang and S.X. Ding, *On fault detection in linear discrete-time, periodic, and sampled-data systems*, Journal of Control Science and Engineering (2008).

[ZDWZ05] P. Zhang, S.X. Ding, G.Z. Wang, and D.H. Zhou, *Fault detection of linear discrete-time periodic systems*, IEEE transactions on Automatic Control **50** (2005), no. 2, 239–244.

[Zha01] Wei Zhang, *Stability analysis of networked control systems*, Ph.D. thesis, Departement of electrical Engineering and computer science,Case Western Reseve University, 2001.

[Zha05] Lei Zhang, *Access scheduling and controller design in networked control systems*, Ph.D. thesis, University of Maryland, 2005.

[Zha10] Peng Zhang, *Industrial control networks*, Advanced Industrial Control Technology, William Andrew Publishing, Oxford, 2010, pp. 363 – 427.

[ZHV05] L. Zhang and D. Hristu-Varsakelis, *Stabilization of networked control systems : Designing effective communication sequences*, 16th IFAC World Congress, 2005.

[ZJ08] Youmin Zhang and Jin Jiang, *Bibliographical review on reconfigurable fault-tolerant control systems*, Annual Reviews in Control **32** (2008), 229–252.

[ZS97] K. Zuberi and K. Shin, *Scheduling messages on controller area network for real-time cim applications*, IEEE Transactions On Robotics And Automation **13** (1997), no. 2, 310–314.

[Zur05] Richard Zurawski, *The industrial communication technology handbook*, CRC Press, 1 edition (February 23, 2005).

[ZV06] L. Zhang and D. Hristu Varsakelis, *Communication and control co-design for networked control systems*, Automatica **42** (2006), no. 6, 953–958.

[ZYC11] Huo Zhihong, Zheng Yuan, and Xu Chang, *A robust fault-tolerant control strategy for networked control systems*, Journal of Network and Computer Applications **34** (2011), no. 2, 708 – 714.