Commande tolérante aux défauts : une approche basée sur la platitude

César Martínez Torres

Thèse en cotutelle présentée

pour obtenir le grade de

Docteur de

L'Université de Bordeaux

et de l'Universidad autónoma de Nuevo León

École doctoral de sciences physiques et de l'ingénieur

École doctorale de génie électrique

Spécialité: Automatique

Par César Martínez Torres

Fault Tolerant Control by flatness approach

Sous la direction de Franck Cazaurang et Efraín Alcorta García

et la co-direction de Loïc Lavigne et David A. Díaz Romero

Soutenue le 25 Mars 2014

Membres du jury:

M. SIRA RAMÍREZ Hebertt	Professeur	CINVESTAV IPN	Président
M. THEILLIOL Didier	Professeur	Université de Lorraine	Rapporteur

RÉSUMÉ

Titre: Commande tolérante aux défauts : une approche basée sur la platitude.

L'objectif de ce manuscrit est de fournir une technique de commande tolérante aux défauts basée sur la platitude différentielle. Pour ce type de systèmes, il est possible de trouver un ensemble de variables, nommées sorties plates, tel que, les états et les entrées de commande du système puissent s'exprimer en fonction de ces sorties et d'un nombre fini de ses dérivées temporelles. Le bloc de détection et d'isolation doit assurer la détection du défaut le plus rapidement possible. Cette action est effectuée en exploitant la propriété de non-unicité des sorties plates. En effet, si un deuxième jeu de sorties plates peux être trouvé et si ce deuxième jeu n'est couplé avec le premier que par une équation différentielle, le nombre des résidus permettant la détection de défauts pourra être augmenté. La condition pour cela est que les deux jeux soient différentiellement couplés ce qui signifie qu'il existe une équation qui contienne des dérivées temporelles et qui couple un élément du premier jeu avec un élément du deuxième jeu de sorties plates. En conséquence le nombre de résidus disponibles pour la détection est supérieur au nombre que l'on aurait si on avait seulement un jeu des sorties plates.

En ce qui concerne la reconfiguration, si le système plat satisfait les propriétés énumérées ci-dessus, nous obtiendrons autant de valeurs des états et des entrées que le nombre de jeux de sorties plates trouvés. En effet chaque entrée de commande et chaque état du système peuvent être recalculés en fonction des sorties plates. L'approche proposée fournit de cette manière un résidu prenant en compte une mesure calculée avec le vecteur plat contenant le défaut et une autre avec le vecteur plat libre de défaut. Les signaux redondants libres de défauts seront ainsi utilisés comme références du contrôleur de manière à ce que les effets du défaut soient masqués et ne rentrent pas la boucle de commande. Ceci sera utile pour fournir une stratégie de commande entièrement basée sur les systèmes plats.

Les travaux présentés dans ce mémoire sont donnés sous l'hypothèse suivante:

- Les sorties plates sont fonctions de l'état du système, néanmoins dans ce manuscrit elles seront limitées à être directement une partie de l'état du système ou une combinaison linéaire d'entre eux.
- La boucle de commande est fermée avec un correcteur par retour d'état.
- Enfin pour les travaux réalisés en fin de manuscrit les sorties plates doivent pouvoir être mesurées ou reconstruites.
- Les défauts affectant les actionneurs sont considérés rejetés par le contrôleur, par conséquent la reconfiguration est seulement effectuée après la détection d'un défaut capteur.

La faisabilité de l'approche proposée est analysée sur deux systèmes non linéaires, un drone quadrirotor et un système de trois cuves.

Mots clés: Commande tolérante aux défauts, platitude différentielle, systèmes non linéaires.

ABSTRACT

Title: Fault Tolerant Control by flatness approach

The objective of this Ph.D. work is to provide a flatness based active fault-tolerant control technique. For such systems, it is possible to find a set of variables, named flat outputs, such that states and control inputs can be expressed as functions of flat outputs and their time derivatives. The fault detection and isolation block has to provide a fast and accurate fault isolation. This action is carried out by exploiting the non-uniqueness property of the flat outputs. In fact, if a second set of flat outputs which are coupled by a differential equation of the first is calculated, the number of residues augments. Differentially coupled means that it exists an equation with time derivatives inside, that couple one element of the first set with one of the second. As a consequence of augmenting the number of residual signal more faults than in the one set case may be isolated.

Regarding reconfiguration, if the flat system complies with the properties listed above, we will obtain versions of states and control inputs as much of flat output vectors, are found, because each control input and state is a function of the flat output. The proposed approach provides in this manner one measure related to a faulty flat output vector and one or more computed by using an unfaulty one.

The redundant state signals could be used as reference of the controller in order to hide the fault effects. This will be helpful to provide an entirely flatness-based fault-tolerant control strategy.

The works presented in this manuscript are under the following hypothesis:

- The flat outputs are functions of the state of the system, however in this work the flat outputs are constrained to be states of the system or a linear combination of them.
- The control loop is closed with a state feedback controller.
- For purposes of this work flat outputs need to be measured.
- Faults affecting the actuators are considered rejected by the controller; by consequence reconfiguration is only carried out after a sensor fault occurs.

Feasibility of the proposed approach is analyzed in two nonlinear plants, an unmanned quadrotor and a three tank system.

Keywords: Fault tolerant control, differential flatness, nonlinear systems.

Unite de recherche

Laboratoire d'Intégration du Matériau au Système (IMS) UMR CNRS 5218.
351 Cours de la Libération, Talence.

Acknowledgements

I would like to express mi gratitude to mi supervisors Franck Cazaurang and Efraín Alcorta García, as well as to mi co-supervisors Loïc Lavigne and David A. Díaz Romero for the time that they consecrated to my PhD project. I appreciate also their continuously support and guidance throughout this three years, not only in the academical field but in the personal too.

Besides my advisors, I would like to thank the rest of my thesis committee: Hebertt Sira Ramírez and Didier Theilliol for their insightful comments that help to improve the quality of the manuscript. In an special manner i would like to thank the members of the committee that make the effort to come to Monterrey despite their busy schedules.

I thank my fellow labmates of DIE in the UANL and the IMS in Bordeaux for the enriching discussions, and for all the fun we have had during this three years.

Por último pero no menos importante agradezco a mis padres Luisa Torres y Jaime Martínez por que han sido una guía durante todos estos años, sin ellos no habría podido llegar hasta este punto. De igual manera agradezco a mis hermanos Jaime y Luis Javier por su apoyo incondicional.

Agradezco también a mi esposa Nadia por su comprensión y apoyo en los momentos difíciles de esta tesis. Confío en poder acompañarte en tus proyectos como tu lo hiciste conmigo.

Son muchas las personas que de manera directa o indirecta estuvieron presentes durante todo este proceso a las que me encantaría agradecerles su amistad, consejos, apoyo, ánimo y compañía.

¡Gracias!

In memoriam of my Granny

Paula Casillas López.

The greater our knowledge increases

the more our ignorance unfolds.

John F. Kennedy

CONTENTS

INTRODUCTION GÉNÉRALE

Dans les années précédentes, l'explosion démographique et la globalisation, ont initié la nécessité de dessiner et d'exploiter des processus de production rentables et donc des systèmes fiables. La présence d'un défaut dans un processus de production peut générer de grandes pertes, non seulement dans le produit manufacturé, mais aussi dans l'équipement de fabrication lui-même. Pour certains systèmes, l'apparition d'un défaut est plus critique, par exemple si un avion en vol de croisière (35000 ft.) est affecté par un défaut, les conséquences peuvent le conduire à la destruction et donc à la perte de vies humaines. Actuellement les avions possèdent un système de surveillance qui assure la sécurité globale du véhicule. Afin d'accomplir cette tâche, les constructeurs utilisent la redondance matérielle, ce qui signifie que deux ou plusieurs sous-systèmes (Ordinateur de commande du vol, capteurs, actionneurs. . .) travaillent ensemble et peuvent prendre le relai en cas de défaillance de l'un d'entre eux. Cette action assure une protection robuste contre les défauts et garanti un taux de panne inférieur à 10^{-9} fautes par heure de vol. Cette solution est simple à implémenter mais représente un cout élevé en raison de la nécessité de tripler ou quadrupler chaque élément.

Pour les systèmes complexes (par exemple; avions, processus pétrochimiques ou nucléaires), chaque composant a été désignée pour accomplir un tâche particulière de maniéré à permettre le fonctionnement global du système. Ainsi, un défaut affectant les actionneurs, capteurs ou le système lui-même peut affecter la performance nominale du système. Les techniques de commande classiques assurent la stabilité du système en boucle fermée ainsi que des performances spécifiques quand le système est soumis à un défaut. Cependant si un défaut affecte le système un correcteur classique ne peut pas maintenir la performance nominale et même la stabilité du système peut ne pas être assurée, par conséquence le système peut être endommagé.

Afin d'éviter la perte du système, les chercheurs ont développé des systèmes de commande capables de s'auto-réparer, cela signifie que le correcteur assure au moins la stabilité du système et au mieux le comportement nominal malgré l'occurrence d'un défaut. Les systèmes qui pressentent cette caractéristique sont connus comme des systèmes tolérantes aux défauts (FTC pour son sigle en anglais). Actuellement la reconfiguration de la commande après la détection d'un défaut est une part essentielle pour quasiment tous les systèmes automatiques. Si le système est affecté par un défaut, le système FTC doit être capable de le détecter, de l'isoler et de le rejeter le plus vite possible en ayant toujours comme objectif de maintenir le comportement nominal ou au moins la stabilité du système.

Tout au long ce manuscrit un défaut sera défini comme dans "Fault diagnosis Systems" publié par Rolf Isermann, [37]. Cette publication définit un défaut comme *" une déviation non permise d'au moins une propriété du système à partir de son état acceptable ou en condition standard"* Deux types différents de défauts sont considérés, additif et multiplicatif. Le premier est représentée par l'addition d'un terme sur la mesure ou sur l'entrée de commande, selon le cas (voir equation (1a)). Les défauts multiplicatifs sont représentés par un terme qui multiplie la mesure ou l'entrée de commande en fonction de la variable concernée (voir équation (1b)).

$$Y(t) = U(t) + f(t) \tag{1a}$$

$$Y(t) = (A + f(t))(U(t)) \tag{1b}$$

Où $Y(t)$ représente la sortie du capteur ou de l'actionneur, $U(t)$ est le signal d'entrée du capteur ou de l'actionneur. $f(t)$ représente le défaut. A indique un facteur multiplicatif lequel est habituellement égal à 1. La figure Fig. 1 montre le schéma bloc correspondant à chaque type de défaut.

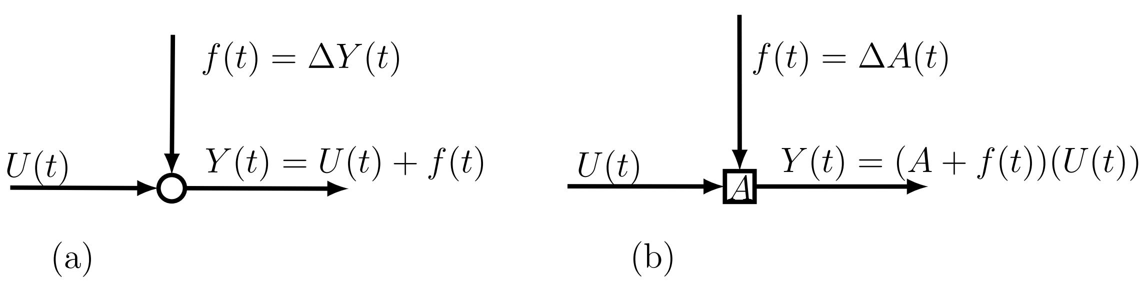

Figure 1: (a) Défaut additive; (b) Défaut multiplicative

Afin de contrer l'effet du défaut, deux types de stratégies peuvent être utilisées, les stratégies passives et les stratégies actives. Le premier type fait partie des méthodes connues en commande robuste. [4]. Les stratégies actives sont caractérisées par la présence d'un module de détection et d'isolement des défauts, (FDI pour son sigle en anglais), lequel après la présence d'un défaut envoie l'information au bloc de reconfiguration pour pouvoir adapter le system afin de contrer les effets du défaut. Voir section 1.4.1 pour plus de détails.

Les travaux présentés dans ce manuscrit font partie de la famille des méthodes actives. L'approche proposée utilise les propriétés des systèmes plats pour générer une redondance analytique, qui sera utilisée pour générer des indicateurs de défauts (résidus). La caractéristique principale de l'approche proposée est basée sur le fait que le module FDI est couplé avec le bloc de reconfiguration. Cette action permet de réduire la charge du calculateur en minimisant le temps de réaction pour contrer les effets du défaut. Comme cette approche est basée sur les propriétés des systèmes plats, elle peut être appliquée aux systèmes linéaires et non linéaires indistinctement. Ce travail est consacré à l'étude de faisabilité de la méthode pour les systèmes non linéaires.

Le manuscrit est divisé en trois chapitres:

Le chapitre 1 est dédié à la présentation des propriétés et des définitions des systèmes plats. La planification de trajectoires basée sur la platitude est présentée également. Un état de l'art sur la détection et l'isolation de défaut ainsi que la reconfiguration est aussi développée.

Le chapitre 2 présente l'approche proposée. Tout d'abord, une brève description sur les approches FTC basées sur la platitude est présentée. L'algorithme pour trouver les sorties plates d'un système est aussi présenté. De manière à faciliter la compréhension et à souligner les avantages de la technique, l'approche de commande tolérante aux défauts est présentée en deux parties, la première dédiée à la détection et l'isolation des défauts et dans un deuxième temps la reconfiguration est prise en charge. Ces deux parties sont également développées sur deux cas d'étude, de manière à montrer que si le système plat rempli certaines hypothèses la méthode FTC proposée sera améliorée.

Enfin, le chapitre 3 est dédié à démontrer l'applicabilité de la méthode FTC sur des cas concrets. Deux systèmes sont considérés. Le premier est un hélicoptère à quatre rotors sans pilote (Drone). Ce système a seulement un jeu de sorties plates, cependant une reconfiguration partielle peut être appliquée. Dans le deuxième cas, l'approche proposée est appliquée sur un système de trois cuves classique. Contrairement au drone, il est possible de déterminer deux jeux de sorties plates sur cette application. Cette caractéristique est donc exploitée pour reconfigurer entièrement le système après l'apparition d'un défaut.

General introduction

In the last decades, demographic explosion and globalization, unchained the necessity to design and operate profitable production process and reliable transport systems. The presence of a fault in a production process can lead to substantial loss, not only in the manufactured product, but in the production equipment itself. In some systems, the fault occurrence is even more critical, for example if an airplane flying at cruising attitude (35000 ft.) is affected by a fault, the consequences of it can lead to destruction of the airplane and by consequence the lost of human lives. Nowadays airplanes count with a surveillance stage, which is in charge of monitoring the entire system to assure the safety of the vehicle. In order to accomplish such task, manufacturers use physical redundancy, meaning that two or more subsystems (e.g. flight-control computers, sensors) work together. This provides robust protection against a simple fault and guarantees a failure rate lower than 10^{-9} failures per flight hour. This solution is easy to implement but represents a high cost due to the necessity to triple or quadruple the elements.

In complex systems (e.g. nuclear plants, petrochemical process, airplanes), every single component has been designed to accomplish a particular task in order to permit the global operation of the system. Thus, a failure in actuators, sensors or the system itself may affect the nominal performance. The classical control techniques assure system stability in closed loop and the nominal performance desired when no-fault is present. However in a faulty case a classic closed loop may result in a low performance or system instability and the possibility of system destruction.

In order to avoid the loss of the system, researchers have developed control systems capable of self-repair, meaning that the controller assures at least system stability and at best nominal behavior despite the occurrence of a fault. Systems presenting this capability are known as Fault Tolerant Control Systems (FTC). Nowadays fault reconfiguration is an essential part of almost every controlled system. FTC systems are designed to behave as a classical control system until a fault affects the controlled plant. If the plant is affected by a fault, the FTC system has to be capable of detecting, identifying and rejecting it as soon as possible. Such actions have as final objective preservation of nominal behavior or at least stability.

Throughout this manuscript a fault is defined as in the book “Fault-diagnosis Systems” published by Rolf Isermann, [37]. That publication defines the fault as *“an unpermitted deviation of at least one characteristic property of the system from the acceptable, usual, standard condition”*. Two different types of faults are considered, additive and multiplicative. The first one is represented by the addition of a term in the measure or in the control input according to the current case, see equation (2a). Multiplicative faults are represented by a term which multiplies the measure or the input according to the affected variable. See equation (2b).

$$Y(t) = U(t) + f(t) \tag{2a}$$

$$Y(t) = (A + f(t))(U(t)) \tag{2b}$$

Where $Y(t)$ represents the output of the sensor or the actuator, $U(t)$ stands for the sensor or actuator input signal. $f(t)$ represents the fault. A denotes a multiplicative factor which is usually equal to one. Fig. 2 shows the block diagrams corresponding to each kind of fault.

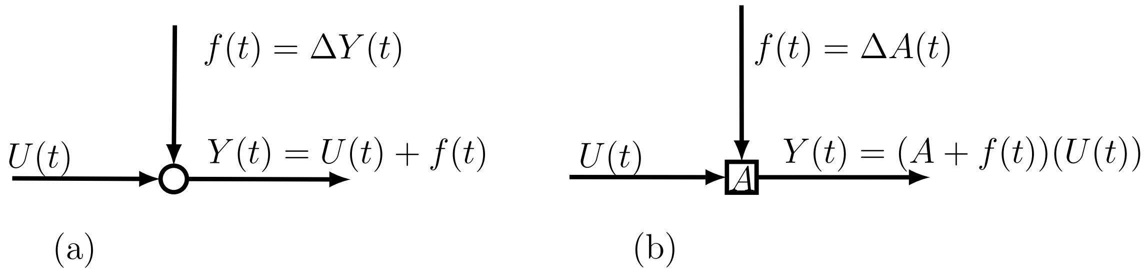

Figure 2: (a) Additive fault; (b) Multiplicative fault

In order to counteract the fault, two different strategies could be used, passive and active. The first strategy comes from the group of robust control. [4]. The active approaches are characterized by the presence of a Fault Detection and Isolation (FDI) module, which on occasion of the fault sends information to the reconfiguration block in order to adapt the system to counteract the fault effect. See section 1.4.1 for more details.

The work presented in this manuscript fits into the framework of active methods. The proposed approach uses the properties of the differentially flat systems to generate analytical redundancy, such redundancy can be used to generate residual signals. The main characteristic of the proposed approach is the fact that the FDI module is coupled with the reconfiguration block. This action could reduce the computational load by minimizing the reaction time to counteract the fault. Because the approach is based on the properties of the differentially flat systems it could be applied to linear and nonlinear systems indistinctly. This work is devoted to investigate feasibility on nonlinear systems.

The manuscript is divided into three chapters:

Chapter 1 is devoted to presenting the properties and the definition of flat systems. Flatness-based motion planning is presented as well. A state of the art of the technique of Fault detection isolation and reconfiguration is developed too.

Chapter 2 presents the proposed approach. First, a brief discussion on the literature of FTC's based on flatness approach is presented. The algorithm to compute flat outputs is presented. In order to facilitate comprehension and highlight the advantages of the technique, the fault tolerant control approach is divided into the fault detection and isolation task and the reconfiguration block. Both activities are divided into two cases, in order to show that if a flat system has at least two different set of flat outputs the FTC method is improved.

Chapter 3 is dedicated to demonstrating the applicability of the FTC method. Two systems are taken into account. The first one is a Unmanned Aerial Vehicle quadrotor. This system exhibits only one set of flat outputs, however partial reconfiguration could be applied. In the second case, the technique proposed in chapter 2 is applied to a classical three tank system. Such plant in contrast to the UAV, presents two sets of flat outputs, this feature is exploited to fully-reconfigure the system after fault.

CHAPTER 1

FUNDAMENTALS

Abstract:

The goal of this chapter is to present the basic concepts of the main parts of this research work. The properties of the so-called differential flatness systems are presented. Flatness-based motion planning is also presented. The definition of the Fault Tolerant Control systems is developed in section 1.4. Sections 1.5 and 1.6 are devoted to presenting the model-based Fault detection and Isolation techniques and the existent fault reconfiguration techniques.

1.1 INTRODUCTION

The appearance of a fault in a system directly affects its performance. As a consequence, this will impact the final objective of the system e.g., final position of a control surface in a plane, the water level in a tank, etc. Classical control laws are designed to ensure stability and nominal performance of the system. However a classic controller does not take into account the appearance of faults affecting sensors, actuators or the system itself. Such appearance will affect the nominal performance in the best of the cases, and in the worst one the system will lose not only performance but even stability. Such behavior should be avoided, especially in critical systems, for instance, nuclear plants or airplanes.

Control systems that take into account such scenarios are known as Fault Tolerant Control Systems (FTC). The purpose of these systems can globally be divided in two main tasks: Fault Detection and Isolation (FDI) and Control Reconfiguration.

This chapter is devoted to presenting such control systems and some techniques of fault detection and fault recovery. Regarding FDI, special attention is dedicated to model-based approaches [23] whose fault detection principle is based on the comparison between sensor measures and the measure estimation coming from a mathematical model describing the physical process. For fault recovery, the research is focused in control reconfiguration [52].

The main contribution of this thesis is based on the properties of the so-called differentially flat systems [27]. The next section presents the definition of those particular systems, as well as the flatness-based motion planning approach, which, is facilitated thanks to the inherent properties of the flat systems. Section 1.4 presents the different approaches found in the literature for FTC systems. Main attention is focused on active fault detection systems, which reacts after a fault occurrence, in order to prevent system loss.

Section 1.5 is devoted to FDI systems and particularly to quantitative model-based methods. A general outlook of these methods is also presented. A non-exhaustive list of fault recovery methods is presented in section 1.6.

1.2 DIFFERENTIALLY FLAT SYSTEMS

The flatness theory search to determine if a system of differential equations could be parametrized by arbitrary functions. The first works where carried out in [12], aiming at aeronautical applications. The development of the theory continued in the Ph.D. dissertation of P. Martin [55]. This work has led to the formal concept of flatness presented by M. Fliess et al. in [27].

The differential flatness of non-linear and linear systems could be described by using mathematical formalisms, and specifically differential algebra or differential geometry.

1.2.1 FLATNESS CONCEPT

A non-linear system is flat if there exists a set of variables, differentially independent, called flat outputs, whose cardinality is equal to the number of control inputs. For instance, the vector state and the control inputs can be expressed as functions of the flat outputs and a finite number of their time derivatives. As a consequence, state and control input trajectories can be obtained by planning only the flat output trajectories. This property can be particularly exploited in trajectory planning, see [49,50,62,82] and trajectory tracking [2,80]. Flatness could be used to design robust controllers; see for instance [9,43].

Definition 1.1 *Flat system: Let us consider the nonlinear system* $\dot{x} = f(x,u)$*,* $x \in \Re^n$ *the state vector,* $u \in \Re^m$ *the control vector and* f *a* C^∞ *function of* x *and* u*. The system is differentially flat if, and only if, there exists a flat output vector* $z \in \Re^m$ *such as:*

- *The flat output vector is expressed as a function of the state x and the control input u and a finite number of their time derivatives.*

$$z = \phi_z\left(x, u_1, u_2, ..., u_m, \dot{u}_1, \dot{u}_2, ..., \dot{u}_m, u_1^{(\gamma_1)}, u_2^{(\gamma_2)}, ...u_m^{(\gamma_m)}\right) \tag{1.1}$$

- *The state x and the control input u are expressed as functions of the vector z and a finite number of their time derivatives.*

$$\begin{aligned} x &= \phi_x\left(z, \dot{z}, ..., z^{(a)}\right) \\ u &= \phi_u\left(z, \dot{z}, ..., z^{(b)}\right) \end{aligned} \tag{1.2}$$

Where a and b are a vector of integers containing in each entry the order of derivation of the corresponding component of the flat output vector z.

Every flat system is equivalent to a linear controllable one by means of a diffeomorphism and endogenous dynamic feedback [45], by consequence every controllable linear system is flat, and conversely. Moreover regarding observability, a flat system is always observable from the flat outputs.

1.2.2 Flat systems examples

This section presents various examples of flat systems. Additional examples can be found in [45].

Example 1.2 *Planar ducted fan [82]:*

The system is mounted on a rotating arm that moves in as the fan moves up. [39], see Fig. 1.1 neglecting some dynamics the nonlinear model obtained is:

$$\begin{bmatrix} m_x\ddot{x} \\ m_y\ddot{y} \\ J\ddot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \\ r & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 + m_g g \end{bmatrix} + \begin{bmatrix} 0 \\ -m_g g \\ 0 \end{bmatrix} \tag{1.3}$$

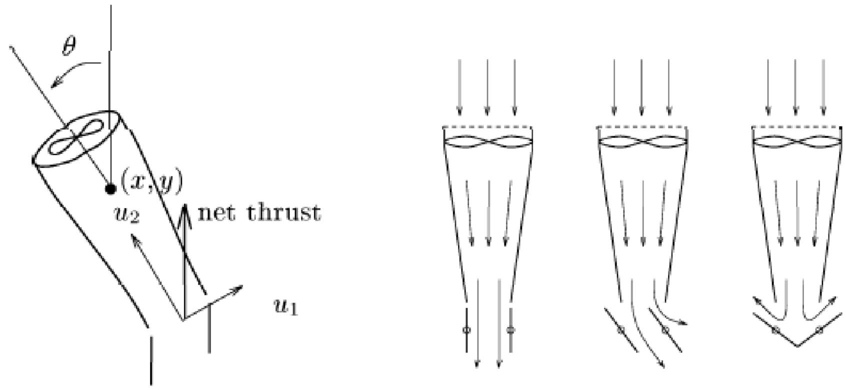

Figure 1.1: Planar ducted fan [82]

Where (x and y) are the coordinates of the center of mass, θ is the angle with the vertical axis, u_1 is the force perpendicular to the fan shroud, r is the distance between the center of mass and the point where the force is applied, g is the gravitational constant, m_x and m_y are the inertial mass of the fan in the (x, y) direction respectively, $m_g g$ is the weight of the fan, and J is the moment of inertia. The tracking outputs are the (x, y) coordinates of the center of mass.

The flat outputs are:

$$\begin{aligned} z_1 &= x - \frac{J}{m_x r} sin\theta \\ z_2 &= y - \frac{J}{m_y r} cos\theta \end{aligned} \tag{1.4}$$

The angle θ can be expressed in function of the flat outputs:

$$\theta = \frac{-m_x \ddot{z}_1}{m_y \ddot{z}_2 + m_g g} \tag{1.5}$$

In an straight manner the states x and y can be obtained directly from the equations 1.4 and equation 1.5.

$$x = z_1 + \frac{J}{m_x r} sin\left(\frac{-m_x \ddot{z}_1}{m_y \ddot{z}_2 + m_g g}\right) \tag{1.6}$$

$$y = z_2 + \frac{J}{m_y r} cos\left(\frac{-m_x \ddot{z}_1}{m_y \ddot{z}_2 + m_g g}\right) \tag{1.7}$$

The interested reader can find the calculation details in [22] and [56]. In order to obtain the expressions of the control inputs it is necessary to compute the time derivative of θ, which will be in function of the flat outputs, because, θ itself, is a function of z. After obtaining the necessary time derivatives, each control input can be expressed as function of the flat outputs.

$$u_1 = \frac{J\ddot{\theta}}{r} \tag{1.8}$$

$$u_2 = \frac{cos\theta u_1 - m_x\ddot{x} - sin\theta m_g g}{sin\theta} \tag{1.9}$$

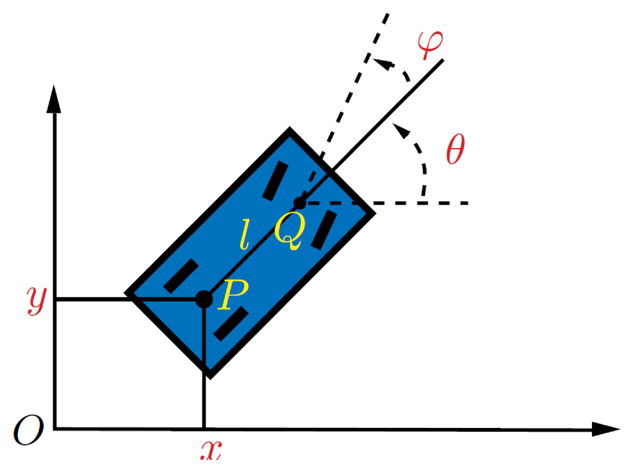

Figure 1.2: Non holonomic car [45]

Example 1.3 *Non holonomic car [45]:*

Consider a vehicle of four wheels rolling without slipping on the horizontal plane. We denote by (x, y) the coordinates of the point P, the middle of the rear axle, Q the middle point of the front axis, θ the angle between the longitudinal axis of the vehicle and the Ox axis. The system has two control inputs namely u and φ the angle of the front wheels with respect to the longitudinal axis. See Fig. 1.2.

The kinematic model of the vehicle can be expressed as follows:

$$\begin{aligned}\dot{x} &= ucos\theta \\ \dot{y} &= usin\theta \\ \dot{\theta} &= \frac{u}{l}tan\varphi\end{aligned} \tag{1.10}$$

This system has two control inputs. By definition, the flat output vector is constituted by two elements. Let us prove that $z = [x,\ y]^T = [z_1,\ z_2]^T$.

Combining the two first equations of (1.10) we can obtain:

$$\begin{aligned}\theta &= tan^{-1}\left(\frac{\dot{z}_2}{\dot{z}_1}\right) \\ u &= \sqrt{\dot{z_1}^2 + \dot{z_2}^2}\end{aligned} \tag{1.11}$$

The expression of $\dot{\theta}$ can be found by computing the time derivative of the first equation in (1.11). Such computation leads to:

$$\dot{\theta} = \frac{\ddot{z}_2\dot{z}_1 - \dot{z}_2\ddot{z}_1}{\dot{z_1}^2 + \dot{z_2}^2} \tag{1.12}$$

From the third equation of (1.10) we can compute

$$\varphi = tan^{-1}\left(\frac{l\dot{\theta}}{u}\right) = tan^{-1}\left(\frac{(l(\ddot{z}_2\dot{z}_1 - \dot{z}_2\ddot{z}_1)}{(\dot{z_1}^2 + \dot{z_2}^2)^{\frac{3}{2}}}\right) \tag{1.13}$$

Equations (1.10,1.12,1.13), demonstrate that the kinematic model of the non holonomic car, described by the equations (1.10), is flat.

1.3 MOTION PLANNING

The goal of motion planning is to provide the system with desired, feasible trajectories for which there exist corresponding control input that moves the concerned system from a start state to a goal condition, while respecting constraints and avoiding collision.

Motion planning for manipulator robots attracted research interest in the early 90's, see [51] for instance. More recently special attention has been devoted to vehicle motion planning. See for instance [33] and references therein for an extensive trajectory planning algorithms survey for UAV's.

This section is focused in the flatness-based motion planning approach.

1.3.1 TERMINOLOGY

The terminology [42] used in this work is the next:

- **Path planning:** A geometric representation to move from an initial to a final condition. The main goal is to find a collision-free path among a collection of static and dynamic obstacles.
- **Trajectory planning:** also known as trajectory generation. It includes velocities, accelerations, and jerks along the path. Normally the main task is to find trajectories for a priori specified paths. Those trajectories could be obliged to fulfill a certain criterion (eg., minimum execution time, minimum energy consumption).
- **Motion planning:** Is the union of path and trajectory planning.

1.3.2 FLATNESS-BASED MOTION PLANNING

As defined in the subsection 1.3.1 the motion planning goal is to compute a trajectory that satisfies certain path constraints.

Let us define a non linear system $\dot{x} = f(x, u)$. The motion planning consists in fulfilling the initial and final conditions presented below, [45]:

$$\begin{gathered} x(t_i) = x_i,\ u(t_i) = u_i \\ x(t_f) = x_f,\ u(t_f) = u_f \end{gathered} \tag{1.14}$$

Once the path is defined, the trajectory generation problem consists in finding a trajectory $t \mapsto (x(t), u(t))$ for $t \in [t_i, t_f]$ that satisfies the system constraints and the initial and final conditions (1.14). Trajectory constraints of type $(x(t), u(t)) \in A(t)$, where $A(t)$ is a submanifold of $X \times U$ could be added to the motion planning initial problem. This results in a growing complexity that requires an iterative solution by numerical methods to find the control input u that satisfies the initial and final conditions (1.14). This iterative process can be solved by using optimal control techniques, however for nonlinear systems some problems are still unsolved. Besides this solution needs to integrate the system equations in order to evaluate the solution proposed.

Motion planning by flatness, does not need to integrate the system equations and for a flat output trajectory, command inputs can be computed directly. The resulting u vector always respects the system dynamics. See equation (1.1). As a consequence the solutions of the set of differential equations are found. See [49, 64].

Definition 1.1 implies that every system variable can be expressed in terms of the flat outputs and a finite number of its time derivatives. As a consequence if we want to compute a trajectory whose initial and final conditions are specified, it suffices to construct a flat output trajectory to obtain the open loop control inputs satisfying the desired state and input trajectories.

In order to compute all the system's variables, the flat output trajectory created needs to be at least, r times differentiable, where r, is the maximal time derivative of the flat output appearing in the equation 1.1. Additionally this trajectory is not required to satisfy any differential equation. In this work the flat outputs trajectories are created by using a simple polynomial approach. See Appendix A for further details.

If the trajectories need to be optimal, in some sense a more advanced trajectory generation technique has to be used. Some application examples can be found in [10, 49, 50, 82].

Let us revisit the example 1.3. Flat output nominal trajectories ($z = [x, y]^T = [z_1, z_2]^T$) are obtained by using fifth order polynomials. Such order is used because if we see in detail the equations (1.11), (1.12) and (1.13) the maximum time derivative inside the equations is two, as a consequence the polynomial used to generate the flat output trajectories needs to be at least equal to three, in order to create sufficiently derivable trajectories. The desired value for the x and y position is the same and it is equal to five, see Fig. 1.3. After computing the time derivatives of the flat output trajectories, it is straightforward to obtain the nominal trajectories for the remaining control inputs and states by using the equations (1.11), (1.12) and (1.13). See Figs. 1.3 and 1.4. If the system is naturally stable, the nominal control inputs obtained can ideally control the system in open loop.

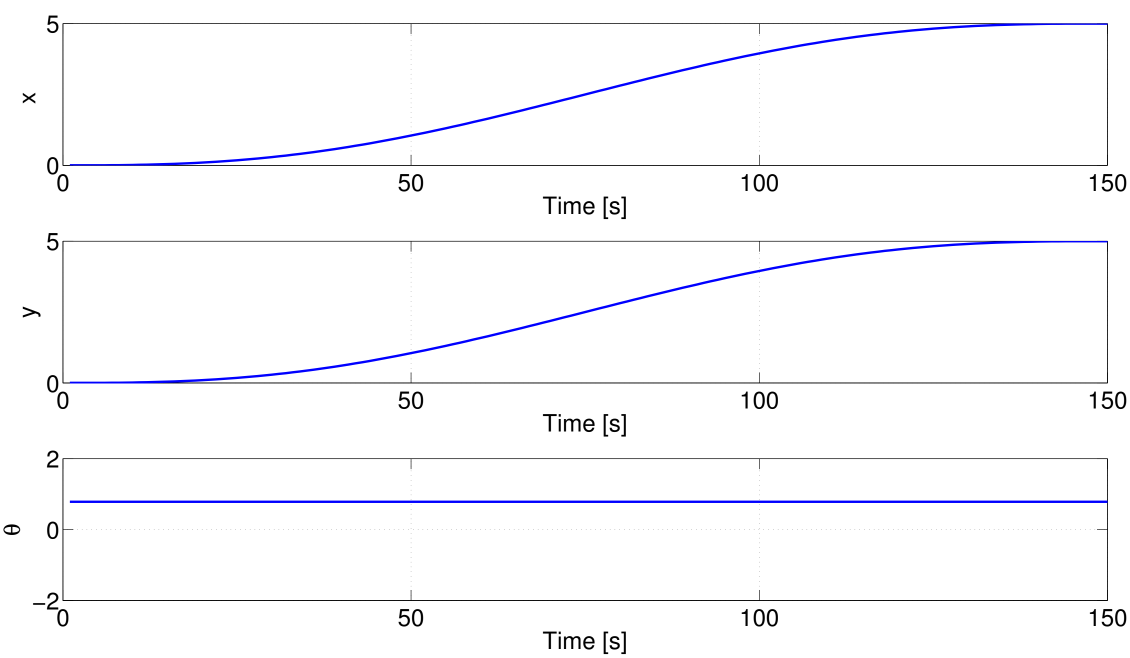

Figure 1.3: Flat outputs and states

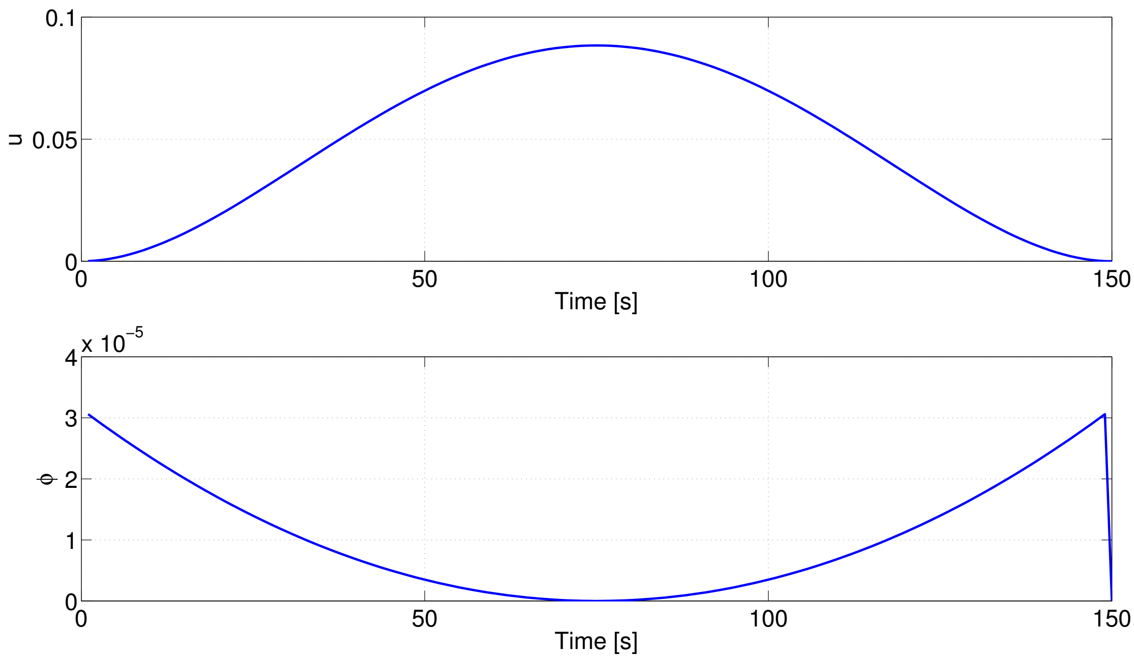

Figure 1.4: Control inputs

1.4 FAULT TOLERANT CONTROL

Industrial and transport systems have become a complex network composed of processors, interfaces, actuators and sensors which may suffer malfunctions. This phenomena could compromise the performance of the entire system if a fault occurs. A fault tolerant control (FTC) is designed keeping in mind such potential system component failures, and avoids system loss which can affect productivity or safety as a result.

FTC is divided in two different approaches:

- Passive: Known as robust control. Here, the control law is designed to be insensitive to some faults. This approach has limited fault-tolerant capabilities and is beyond the scope of this work. Interested readers are referred to [93] and references therein.

- Active: In this approach, the control system is reconfigured using the information coming from the detection block, having the goal of maintain, at least, system stability and, at the best, the nominal behavior. See [63, 70].

This research work is focused on Active Fault Tolerant Control approach.

1.4.1 ACTIVE FAULT TOLERANT CONTROL SYSTEMS (AFTCS)

Active approaches consist of adjusting the controller on-line, according to the detected fault, having as the goal preservation of the faulty system performance close to the nominal one.

For critical failures such as an actuator lost, for instance, the nominal behavior cannot be maintained, thus, system performance is reduced as shown in Fig. 1.5. The FTC objective is to reconfigure the controller as fast as possible in order to maintain nominal performance. Moreover, controller reconfiguration is not efficient for some kinds of faults, thus is it impossible to keep the system operating even in a degraded mode. In this case FTC function is to shut down the systems safely.

In this way three activities must be covered by the FTC system, [70]:

- Deal with various kinds of faults (sensor, actuators and the system itself).
- Provide information about the fault and the achievable performance.
- Decide if the system can still operate or not.

The active fault tolerant control algorithm regards four stages, [92]. see Fig. 1.6.

- A reconfigurable feedforward/feedback controller, which can react to the failure by changing some controller parameters or the entire closed loop.

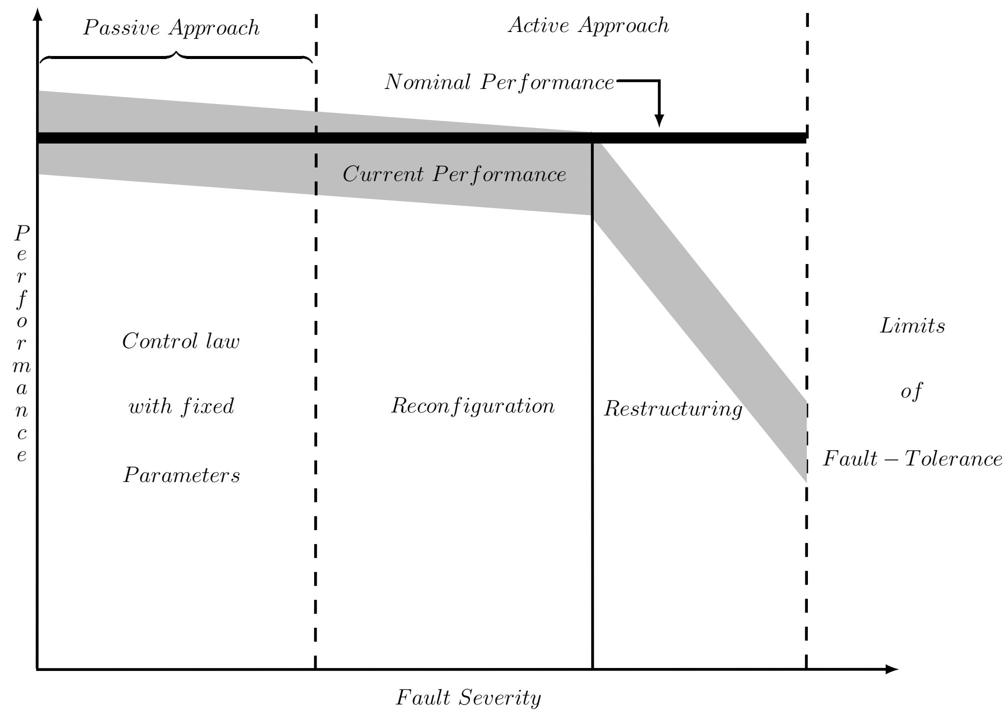

Figure 1.5: FTC Strategies [70]

- A Fault detection and identification (FDI) block. This block has to perform fast and accurate failure recognition.
- A controller reconfiguration mechanism, which is in charge of linking the fault identification mechanism and the reconfigurable controller.
- A trajectory planner/re-planner designed to avoid actuator saturation and adjust the reference trajectory after failure.

Thanks to its versatility, differential flatness can be used to create a completely FTC structure. In fact since its formulation in 1992, flatness has been widely used to design controllers. See [35,56,65,74,90] for some examples. Regarding fault detection and identification some work has been presented, [48, 59, 66]. The reference [59] is a topic of this dissertation. Motion planning/replanning can be achieved using flatness. See [10, 11, 50] for some examples. Control reconfiguration has been studied as well. See [54, 58, 60, 81]. More details of the FDI and FTC approaches will be presented in the next chapter.

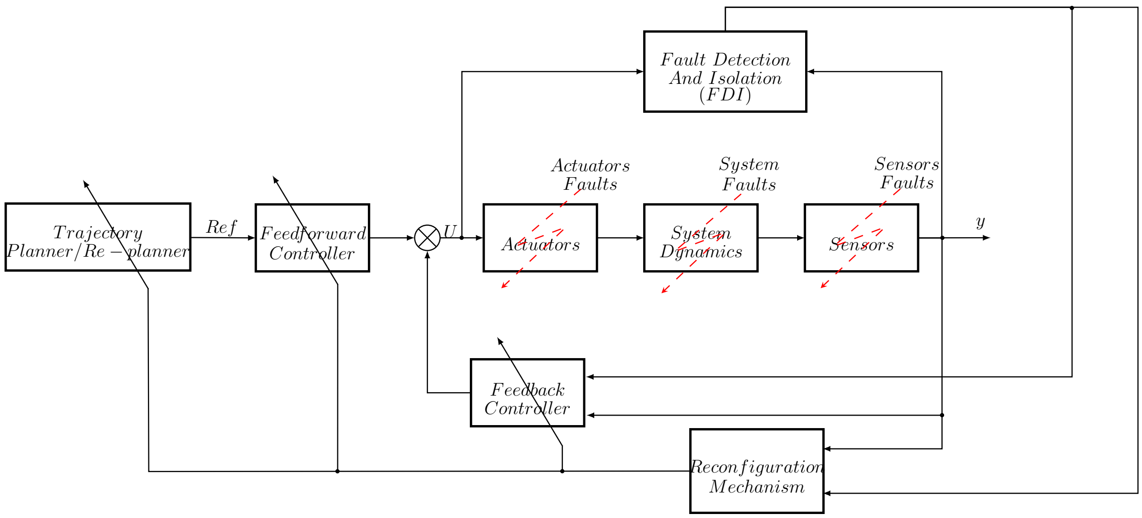

Figure 1.6: General structure of AFTCS [92]

1.5 Fault Detection and Isolation (FDI)

This stage can be divided in two essential tasks [23]:

- Fault detection: in charge of detecting the non-expected behavior of the system.
- Fault isolation: localizing the faulty element.

A third activity can be added such consists in determining the amplitude of the fault. FDI block plays an important role inside AFTCS, because if fault detection is accurate the control reconfiguration will be more effective. According to [86], FDI methods can be divided into two different groups, see Fig. 1.7:

- Model-based Methods:
  - Quantitative methods: Based on mathematical functional expression of relationships between inputs and outputs of the system, [86].
  - Qualitative methods: Based on qualitative functional expression of relationships between inputs and outputs, [84].

- Process history based: Based on the availability of large amount of process history, [85].

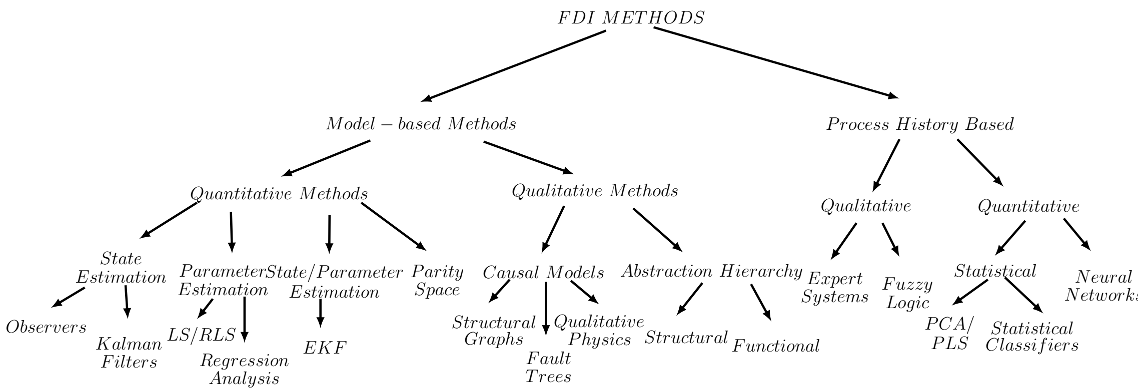

Figure 1.7: FDI Methods [92]

In order to isolate the fault, the three approaches presented below require some a priori knowledge; that is, the set of faults and their relationship with the residues. This information is normally sorted in, the form of a table. Qualitative model-based and process histories are outside of the boundaries of this thesis. Interested readers can find a review of them in [84] and [85] respectively.

1.5.1 Quantitative model-based FDI approach

Quantitative methods require a mathematical model of the system in order to compute residual signals, which reflect the faults affecting the system. Then, this information is introduced into a decision rule. The union of these tasks helps to obtain information about the fault affecting the plant.

In order to compute the residual signals, redundancy is needed, This could be obtained through two different approaches:

- Hardware Redundancy
- Analytical Redundancy

Figure 1.8 shows the differences between these two approaches.

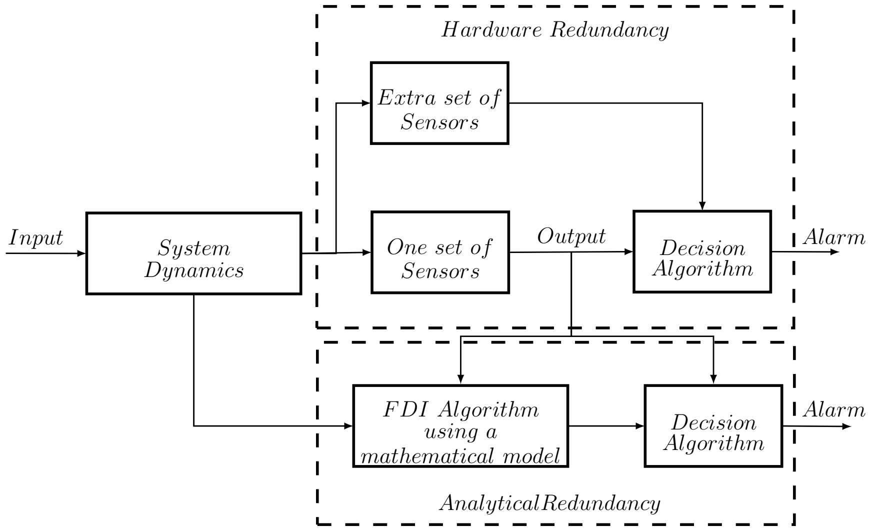

Figure 1.8: Hardware redundancy and analytical redundancy diagram

Hardware redundancy Also known as physical redundancy, it is widely used in chemical industries, aeronautics and industrial processes where persons lives are in danger. The main idea consists in multiplying the number of sensors dedicated to perform the same activity. Those sensors usually are based on different technologies.

For instance, if the system is equipped with three sensors, three residues will be computed as follows:

$$\begin{aligned} r_1 &= m_1 - m_2 \\ r_2 &= m_1 - m_3 \\ r_3 &= m_2 - m_3 \end{aligned} \tag{1.15}$$

A voting mechanism points to the faulty sensor by using the table 1.1 in which the symbol $\boxed{\checkmark}$ means that the sensor is fault-free and $\boxed{\times}$ means that the sensor is faulty.

This approach has as advantage an easy design and efficiency. On the other hand, since sensors are multiplied, the construction and operation costs are elevated.

Sensor 1	Sensor 2	Sensor 3	r_1	r_2	r_3
$\checkmark$	$\checkmark$	$\checkmark$	0	0	0
$\times$	$\checkmark$	$\checkmark$	$\neq 0$	$\neq 0$	0
$\checkmark$	$\times$	$\checkmark$	$\neq 0$	0	$\neq 0$
$\checkmark$	$\checkmark$	$\times$	0	$\neq 0$	$\neq 0$
$\checkmark$	$\times$	$\times$	$\neq 0$	$\neq 0$	$\neq 0$
$\times$	$\checkmark$	$\times$	$\neq 0$	$\neq 0$	$\neq 0$
$\times$	$\times$	$\checkmark$	$\neq 0$	$\neq 0$	$\neq 0$
$\times$	$\times$	$\times$	$\neq 0$	$\neq 0$	$\neq 0$

Table 1.1: Hardware redundancy FDI logic

Analytical redundancy In contrast to physical redundancy, here, the redundant signals are computed via mathematical equations describing the plant. Theorems and lemmas presented in this section will be given without proof, the interested reader can find them in [15, 17, 78].

Residual signals are constructed by making a comparison between the real process and the mathematical model, it is straightforward to think that if the system is unfaulty, the residual signal will be equals to zero, however if a fault affects the plant, the resultant residue will be different to zero. This approach is correct if the mathematical model describes perfectly the process and disturbances are not present. In practice this behavior is not possible, since model uncertainties are always present. Moreover residue post-processing is necessary to distinguish the effects of different faults. After fault generation the principal concern is to obtain the maximum quantity of fault information from them. Such problem is known as FPRG (Fundamental Problem in Residual Generation), the main concern is that the FDI process is require to discern between the fault and external disturbances.

The FPRG problem consists of generate analytical redundancy relations (ARR) fulfilling the next conditions:

- The residuals are not affected by the external disturbances.
- the residuals are affected by the faults.

Let us define an ARR in a formal manner.

Consider a system of the form:

$$\begin{aligned} \dot{x} &= f(x, u, \upsilon, f) \\ y &= h(x, u, \upsilon, f) \end{aligned} \tag{1.16}$$

Where $x \in \Re^n$ is the state vector, $y \in \Re^p$ is the output vector, $u \in \Re^{m_u}$ is the vector containing the control inputs, $\upsilon \in \Re^{m_\upsilon}$ is the vector containing the unknown inputs and $f \in \Re^{m_f}$ stands for the fault vector. In order to simplify the notation the known (u) and unknown inputs (υ, f) will be compacted in one variable denoted by ω. Signals y and ω are smooth signals. $\bar{\omega}^{(s)}$ denotes the time derivatives of ω until the order s. An order of derivation (s_i) is attributed to each output $y_i, i \in [1, 2...p]$.

Lemma 1.4 *[17] The time derivative of order s_i of the output y_i is a smooth function and is written as follows:*

$$y_i^{(s_i)} = g_{i,s_i}\left(x, \bar{\omega}^{(s_i)}\right) \tag{1.17}$$

All the outputs with their subsequent time derivatives are grouped to obtain:

$$\begin{aligned} \begin{bmatrix} y_1 \\ \vdots \\ {y_1}^{(s_i)} \\ \vdots \\ y_p \\ \vdots \\ y_p^{(s_p)} \end{bmatrix} &= \begin{bmatrix} g_{1,0}(x,\omega) \\ \vdots \\ g_{1,s_1}(x, \bar{\omega}^{(s_1)}) \\ \vdots \\ g_{p,0}(x,\omega) \\ \vdots \\ g_{1,s_p}(x, \bar{\omega}^{(s_p)}) \end{bmatrix} \\ \bar{y}^{(s)} &= G_s\left(x, \bar{\omega}^{(s)}\right) \end{aligned} \tag{1.18}$$

Definition 1.5 *[17] An analytical redundancy relation (ARR) is a relation of the form:*

$$\omega_i\left(\bar{y}^{(s)}, \bar{\omega}^{(s)}\right) = 0 \tag{1.19}$$

This definition implies that the set of ARR is not unique, in fact an infinitely number of them can be computed by multiplying them by a not null function in $\bar{y}$,| $\bar{\omega}$. In the same manner a third ARR can be obtained from two other ARR, however those relations cannot be used for fault detection tasks, because if one relation is related to another in an algebraic manner the same fault will affect them and this could prevent the fault isolation. For this reason the ARR has to be algebraically independent, this, is defined in [40]as follows:

Definition 1.6 *[17] Consider η elements ω_1, ω_2, ..., ω_η belonging to a family of functions Ω, such elements are algebraically independent over a ring k if there exists a not null polynomial function $P(\omega_1, \omega_2, ...\omega_\eta) = 0$. If such Polynomial function does not exist the elements $\omega_1, \omega_2, ...\omega_\eta$ are algebraically independents.*

The analytical redundancy relations defined in the paragraphs above can be used as residual generators if such relations are decomposed into two parts, one that depends on the known variables (y and u) and a second which depends of at least one of the unknown variables (υ and f), [78].

$$\omega_i\left(\bar{y}^{(s)}, \bar{\omega}^{(s)}\right) = \omega_i\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, \bar{f}^{(s)}\right) = \omega_{i,c}\left(\bar{y}^{(s)}, \bar{u}^{(s)}\right) - \omega_{i,e}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, \bar{f}^{(s)}\right) = 0 \tag{1.20}$$

Where $\omega_{i,c}\left(\bar{y}^{(s)}, \bar{u}^{(s)}\right)$ is the residual computation form. It includes only known variables, as a consequence it can be computed online. In the absence of faults f it is identically zero and verifies that:

$$\omega_{i,c}\left(\bar{y}^{(s)}\bar{u}^{(s)}\right) = \omega_i\left(\bar{y}^{(s)}, \bar{u}^{(s)}, 0, 0\right) \forall \bar{y}^{(s)}, \bar{u}^{(s)} \tag{1.21}$$

and $\omega_{i,e}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, \bar{f}^{(s)}\right)$ is the residual evaluation form and verifies that:

$$\omega_{i,e}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, 0, 0\right) = 0 \, \forall \bar{y}^{(s)}, \bar{u}^{(s)} \tag{1.22}$$

If the system is not affected by an unknown input ($\upsilon = 0$), the residual evaluation form is equal to zero only if no fault is present. Under such conditions the obtained ARR can be used as residual generators. However as explained above in real applications unknown inputs are always present. In order to solve the FPRG problem it becomes necessary to compute ARR which will be robust to the unknown inputs. Such framework is overcome by splitting the residual evaluation form into two parts, one that will not depend of the faults and other that will depend of them.

$$\omega_i\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, \bar{f}^{(s)}\right) = \omega_{i,e,r\bar{o}b}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}\right) + \omega_{i,e,rob}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, \bar{f}^{(s)}\right) \quad (1.23)$$

Where $\omega_{i,e,r\bar{o}b}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}\right) = \omega_{i,e}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, 0\right)$, in order to exploit the computation form the ARR has to be designed in such a way that the expression $\omega_{i,e,r\bar{o}b}\left(\bar{y}^{(s)}, \bar{u}^{(s)}, \bar{\upsilon}^{(s)}, 0\right)$ will be always equal to zero.

The last stage to use the ARR for the FDI purposes is the rearrangement of them in order to be able to isolate the faults. To this, the analytical redundacy relations has to be insensitive to the unknown inputs and the residual evaluation form needs to be constructed in such a manner that the resultant fault signature matrix have a structure that allows the isolation of each fault. The next is an example of a fault matrix structure.

A fault is considered detectable if and only if it has a non-zero signature in at least one residual, it means that at least one of the residual signals is impacted by the fault effect. The fault is considered isolable if and only if it has a unique fault signature, table 1.2 clarifies both concepts.

Fault	r_1	r_2	r_3
1	0	0	0
2	1	0	0
3	0	1	0
4	0	1	0

Table 1.2: Isolability and detectability

Fault 1 is not detectable because any residue is affected, faults 2 to 4 are detectable, however only fault number 2 can be isolated because is the only of them that has a unique fault signature.

As discussed above it is impossible to detect and isolate all the faults it some ARR is not algebraically independent of the other. For this it becomes necessary to present the maximal number of algebraically independent ARR that can be obtained for a certain system model. This, is resumed in the next theorem.

Theorem 1.7 *[17] For some fixed multi-index ($s_1, s_2, ...s_p \in \mathbf{N}^p$) it is associate an integer r_s defined as the rank of the Jacobian matrix in x of G, where G is defined as the right side of equation (1.18).*

$$r_s = rank\left[\frac{\partial G_s}{\partial_x}\right] = \begin{bmatrix} \frac{\partial g_{1,0}}{\partial x_1} & \dots & \dots & \frac{\partial g_{1,0}}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{1,s_1}}{\partial x_1} & \dots & \dots & \frac{\partial g_{1,s_1}}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{p,0}}{\partial x_1} & \dots & \dots & \frac{\partial g_{p,0}}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_{p,s_1}}{\partial x_1} & \dots & \dots & \frac{\partial g_{p,s_p}}{\partial x_n} \end{bmatrix} \tag{1.24}$$

Thus, it exists exactly

$$\eta = \sum_{i=1}^{p} s_i + p - r_s \tag{1.25}$$

analitycal redundancy relations algebraically independents. The interested reader can find the proof in [17]. By definition the rank of the matrix $\frac{\partial G_s}{\partial x}$ will be at maximum n, where n is the maximal number of the number of lines (lin) and the number of columns (col) of the matrix, for instance if $lin > col$, $n = lin$, if $lin = col = n$. Let us suppose that the number of time derivatives are chosen as $\eta = \sum_{i=1}^{p} s_i + p \leqq n$, as consequence η is always equals to zero. Thus, in order to ensure that the ARR exists it suffice to chose the orders of the time derivatives wisely.

Different quantitative model-based FDI techniques for linear and nonlinear systems are presented below.

State estimation

The main idea of those techniques is to provide an estimate of the system state $\hat{x}$, from measurements of the control inputs and system measurements. In this way residual signals for sensor faults can be computed by simply comparing the actual output and the estimated one. $r = y - C\hat{x}$ for linear systems and $r = y - H(\hat{x}, u)$ for nonlinear models.

In order to compute the estimated state, different approaches can be used. The following is a non-exhaustive list of techniques which can be utilized:

Observer based Observer-based approaches are the mostly often applied model-based residual generation techniques. This technique is based in the reconstruction of the outputs of the system by means of the mathematical model of the plant operating in nominal mode. It should be noted that there is a difference between observers used for control purposes and fault detection. Observers needed for control are state observers, meaning that they estimate states which are not directly measured, with the goal to use such estimations to control the concerned plant. On the other hand, observers needed for fault detection generate estimation of the measurements. Both of them are then compared in order to compute a residual signal. Any deviation of residual signal from zero will trigger a fault alarm. However, the presence of modeling uncertainties and disturbances is inevitable. Therefore, the aim is to design observers such that the effect of the disturbances and uncertainties on the residual signal is reduced while the affect of faults is considerably increased.

Consider the nonlinear system described by the equations

$$
\begin{aligned}
\dot{x}(t) &= f(x(t), u(t), \theta_f, \theta_d) \; x(0) = x_0 \\
y(t) &= h(x(t), u(t), \theta_{fs})
\end{aligned}
\tag{1.26}
$$

Where $x(t) \in \Re^n$ is the state vector, $u(t) \in \Re^m$ is the control input, $y(t) \in \Re^p$ is the output of the system. $\theta_f \in \Re^l$ represents the system parameters, $\theta_f = \theta_{f0}$ when no fault is present in the system. $\theta_{fs} \in \Re^{ls}$ represents the parameter in the output equations, $\theta_{fs} = \theta_{fs0}$ represents the nominal output parameters. $\theta_d \in \Re^{ld}$ represents modelling mismatches. If the model of the system is perfectly known $\theta_d = 0$.

The observer-based fault diagnosis problem consists in finding a residual generator $r(t)$ of the form:

$$\begin{aligned} \dot{\xi}(t) &= g(\xi(t), y(t), u(t)\theta_{f0}),\ \xi(0) = \xi_0 \\ r(t) &= R(\xi(t), y(t), u(t), \theta_{fs0}) \end{aligned} \tag{1.27}$$

In the past, several observer-based approaches have been proposed, see for example [13, 23, 28] for a survey.

Nonlinear identity observer approach (NIO) Proposed for the first time in [34], this observer is developed under the assumption that the model is perfectly known and the system is unfaulty. The observer structure is the following:

$$\begin{aligned} \dot{\xi}(t) &= f(\xi, u, \theta_{f0}, 0) + K_{obs}(\xi, u)[y - \hat{y}] \\ r(t) &= y - h(\xi, u, \theta_{fs0}) \end{aligned} \tag{1.28}$$

The observer is designed under the assumptions that no faults and no modelling mismatches are present in the system. The error estimation can be defined by: $e(t) = x(t) - \xi(t)$,thus, its dynamics can be expressed as follows:

$$\begin{aligned} \dot{e} &= F(\xi, u, \theta_{f0}, 0)e - K_{obs}(\xi, u)H(\xi, u, \theta_{fs0,0})e + HOT \\ r(t) &= H(\xi, u, \theta_{fs0,0})e + HOT \end{aligned} \tag{1.29}$$

Where $F(\xi, u, \theta_{f0}, 0) = \frac{\delta h((x,u,\theta_{f0},0))}{\delta x}|_{x=\xi}$ and the matrix $H(\xi, u, \theta_{fs0}) = \frac{\delta f((x,u))}{\delta x}|_{x=\xi}$. HOT stands for the High Order Terms, such expressions are neglected.

The observer gain K_{obs} is determined in such a way that the error dynamics are asymptotically stable. A solution to this problem was first proposed in [1] by assuming that the measurements are linear.

Extended Luenberger Observer The first application of a Luenberger observer was devoted to linear systems [16]. This approach can be directly applied to nonlinear systems. However, if the system is operating far away from the linearizing point, the linearized system could deviate largely from the nonlinear model.

The main idea of the extended version of the Luenberger observer is to linearize the model around current states estimation ($\hat{x}$), instead of a fix point. Once a sufficiently accurate linearization is computed the observer can be applied.

Detailed information of and FDI application can be found in [1, 89]. The practical application of this approach is not optimal, since the observer gain has to be computed repetitively, which means an important computational charge.

Parameter estimation

Parameter estimation approaches are based on the assumption that the faults are reflected in the system parameters. The detection task is accomplished by comparing the nominal parameters versus on-line estimations. The main advantage of this approach is that it yields the size of the parameter deviation which is important to fault analysis. Parameter estimation is useful for component fault detection since it verifies directly the discrepancy between internal parameters. A disadvantage is that an input signal could be needed in order to excite the system and create signals to estimate the parameters. This action may result in problems if the system is operating in the stationary mode. Such problem could be avoided using algebraic parameter estimation.

Most of the parameter estimation techniques are based on least squares (LS), recursive least squares (RLS), extended least squares (ELS), etc.

Simultaneous State/Parameter estimation

Extended Kalman filter The extended Kalman filter (EKF) has been largely applied to estimate states and system parameters of discrete systems. Let us consider the discrete nonlinear system described by:

$$\begin{aligned} x_k &= f(x_{k-1}, u_k, v_k, \theta_k) \\ y_k &= h(x_k, w_k, \theta_k) \end{aligned} \tag{1.30}$$

where $x \in \Re^n$ the state vector, $u \in \Re^m$ the control input vector, $y \in \Re^p$ the output measures, $v \in \Re^n$ and $w \in \Re^p$ are the state and measure noise respectively and $\theta \in \Re^q$ is the vector of parameters.

The main idea of the EKF is to linearize the nonlinear functions f and h around the current state estimation $\hat{x_k}$, and then the Kalman filter is applied.

The Kalman filter is compound by a group of recurrent equations, which are relatively easy to solve from a numerical point of view. The filter provides the optimal estimation of the states and the variance of the estimation error.

$$\begin{aligned} x_k &= A_k x_{k-1} + B_k u_k + G_k v_k \\ y_k &= C_k x_k + E_k w_k \end{aligned} \tag{1.31}$$

where v and w are non-correlated white noises with zero mean. A_k, B_k,C_k, G_k and E_k are the system matrix, linearized and evaluated in the instant k.

$$\begin{aligned} E[v_k v_j^T] &= Q_k \delta kj \\ E[w_k w_j^T] &= R_k \delta kj \\ E[w_k v_j^T] &= 0 \quad \forall k, j \end{aligned} \tag{1.32}$$

where Q and R are the variance matrices of the noise, $E[.]$ is the expectation value of the alleatory variable $[.]$ and $\delta_{kj} = 1, k = j$ and $0, k \neq j$.

The state x and the measure y are deducted from the white noises v and w and the initial condition x_0, with $E[x_0] = 0$. From the initial conditions the covariance matrix $P_0 = E[x_0 x_0^T]$. The goal is provide an estimation of the state vector $\hat{x}_k$, by minimizing the variance of the error estimation.

$$\hat{x}_k = argmin \left\{ [(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T \mid y_{1:k}] \right\} \tag{1.33}$$

For further details see, for example [19, 69].

Parity space

The parity space approach, first presented in [14], makes use of the parity check on the consistency of parity equation by using system measures. In this way the inconsistency in the parity relations indicates the presence of a fault. Chow and Willsky, [14], derived the parity relations based on state-space model of the system. An approach based in transfer functions was developed in [32]. The main idea of this approach is to eliminate the unknown state and then to obtain relations where all components are known. Such proposition is intended to linear systems, however the same idea can be extended to nonlinear systems.

For nonlinear systems an approach based on the inverse model of input-output is presented in [41,67] generalized the parity space approach for linear systems to nonlinear systems described by Takagi-Sugeno fuzzy models. Leuschen et al. present in [44] an extension of the linear parity space approach to nonlinear systems by preserving the original structure of the polynomial parity vector approach. Staroswiecki and Comtet-Varga presented an approach based on elimination theory and Gröbner bases, [17]. Such approach is intended to state affine systems [77] and systems modeled by differential polynomial equations [78]. More recents works are based on Bond Graphs, see for instance [88] and [87].

1.6 FAULT RECOVERY

After the fault isolation stage, the next step in a FTC system consist in re-adjusting the control chain of the system, aiming to achieve nominal behavior or at least stability. This task can be accomplished in two different ways [52, 73].

- Fault accommodation.
- Fault reconfiguration.

Figure 1.9 and the next subsections present a general overview of both groups of methods.

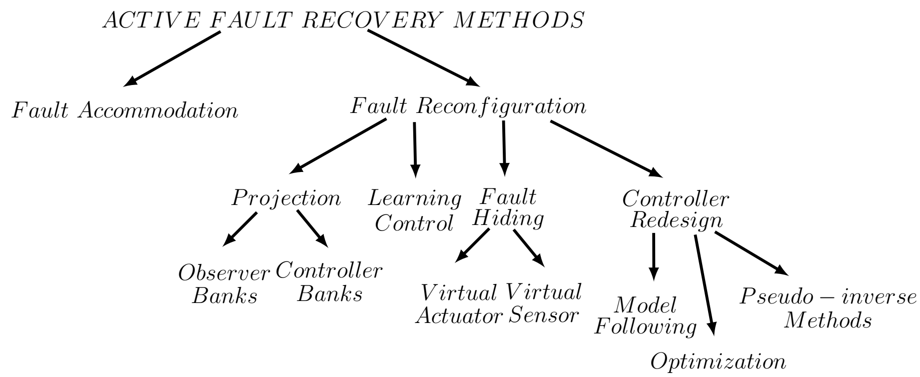

Figure 1.9: Fault recovery methods [52]

1.6.1 Fault Accommodation

Here, the measurements and the signals going into the controller remain unchanged, and fault recovery is carried out by changing the controller structure (dynamic order, parameters, gain values etc.), [63]. One example of this is the adaptive controller technique, where the controller is tuned to minimize the distance between nominal closed loop and the actual behavior, [5].

The adaptive controller technique can be divided into two approaches: direct and indirect. In the first one the controller parameters are directly tuned. The second is performed in two steps. First, the mathematical model of the plant is estimated and then a controller for this plant is computed. This approach presents some limitations. For example, if an abrupt fault affects the system, the needed time to compute all the controllers parameters could be important. By consequence the system could become unstable before finishing the computation. The same case is worse with the second approach, since more time is needed to carry out both steps. Besides, the fault can lead the system outside the linearization zone and by consequence linearizing the system becomes impossible.

1.6.2 Fault reconfiguration

In fault reconfiguration techniques controller parameters and input-output signals are manipulated. In this way those techniques carry out fault recovery not only by reconfiguring the controller but also by including dynamic signal re-routing of measures.

Reconfiguration methods are divided in four different groups, [52]:

- Projection.
- Controller redesign.
- Fault hiding.
- Learning control.

Projection

Methods included in this classification are always based on the off-line design of certain components. Those elements are arranged in banks. Depending on the fault that needs to be reconfigured, two banks can be used, see Fig. 1.10:

Bank of observers Observer banks can only handle sensor faults. Each observer uses the information of all sensors but one. This measure is considered hypothetically faulty. Using all inputs and all outputs except the one considered faulty, each observer can compute an estimation of every system state, and thus estimate the plant output $\hat{y}$. Residues are obtained by computing the difference between the measure y and the estimation $\hat{y}$. The residue with the smallest error represents the current fault case, [29]. Reconfiguration is achieved by feeding the nominal controller with the current fault case. The main advantage of this technique is that it handles the reconfiguration in an integrated manner, which as a consequence can reduce the computing time.

Bank of controllers Sensors, actuators and system faults can be covered with this technique. The FDI is carried out by a diagnostic algorithm. The fault information coming from this block is then used to select the most appropriate of the a priori designed controllers. Since, the number of controllers designed must be equal to the number of managed failures, the off-line effort can be important.

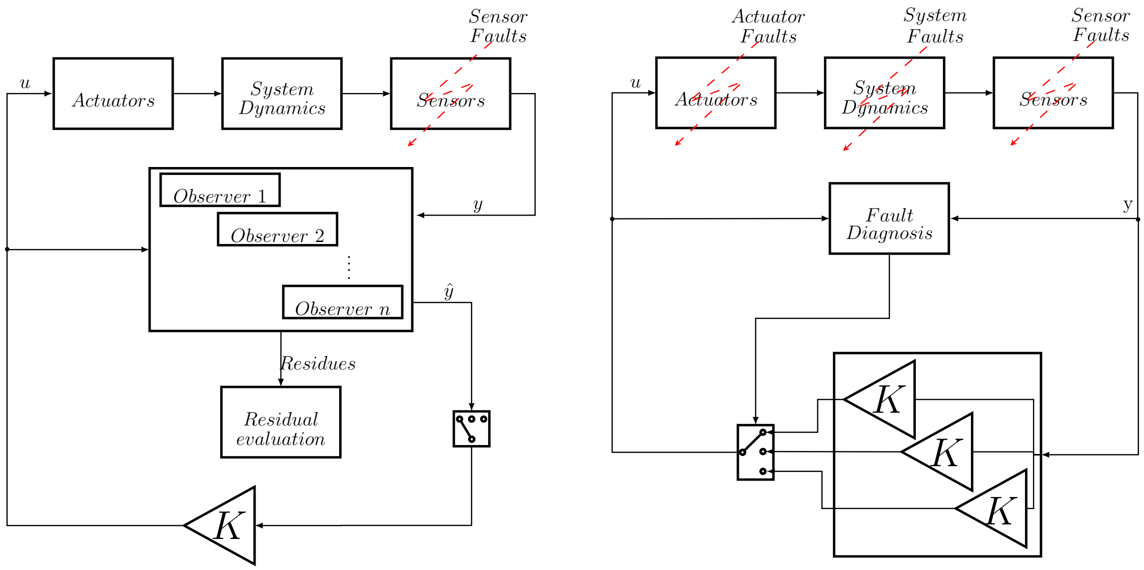

Figure 1.10: Banks of observers and controllers [52]

Controller redesign

This approach can perform, in real time, a complete redesign of the controller in order to recover the faulty system. This action is carried out in explicit or implicit ways. In the first way, the difference between the outputs of the reconfigured plant and the reference model is minimized. In the implicit way, quadratic functions of the actual and modeled states are minimized. Computational cost varies from one method to another. The goal of the control reconfiguration is to minimize the distance between a nominal unfaulty model and the faulty system. This problem is known as model matching. Mathematically, this idea can be expressed as follows:

Define a linear system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + B_d d_x(t) \\ y(t) &= Cx(t) + d_y \end{aligned} \tag{1.34}$$

Where $x(t) \in \Re^n$ is the system state, $y(t) \in \Re^p$ is the output, $u(t) \in \Re^m$ the control input, B_d the disturbance distribution matrix, d_x and d_y are the state and measurements disturbance respectively.

In the presence of faults the system (1.34) becomes:

$$\begin{aligned}\dot{x}_f(t) &= A_f x_f(t) + B_f u_f(t) + B_d d_x(t)\\ y(t) &= C_f x_f(t) + d_y\end{aligned} \tag{1.35}$$

The idea of model matching is to define a reference model constituted by the system (1.35) and the state feedback controller:

$$u(t) = Kx(t) + Gw(t) \tag{1.36}$$

Where K is the static controller feedback matrix, G is the reference pre-filter, $w(t)$ the reference input. The reference model is expressed as follows:

$$\begin{aligned}\dot{x}(t) &= Mx(t) + Nw(t)\\ y &= P^* x\end{aligned} \tag{1.37}$$

In transfer function form, the reference model is:

$$T(s) = P^*\left(sI - M\right)^{-1} N \tag{1.38}$$

Where $M = A - BK$ and $N = BG$. M, N and P^* are selected by the designer. Thus the model-matching problem consists of determining a new feedback controller:

$$u(t) = K_f x(t) + G_f w(t) \tag{1.39}$$

such that:

$$\begin{aligned}A_f - B_f K_f - M &= 0\\ B_f G_f - N &= 0\end{aligned} \tag{1.40}$$

Various approaches have been developed to solve this problem. [52]

Pseudo-inverse methods This method was the first one to treat the model-matching problem [6]. It is addressed to actuators and fault systems. Here the model matching problem presented above is solved by minimizing the distance between the closed loop matrices according to the 2-norm $\parallel . \parallel_2$[1]. Two criteria are minimized:

$$\begin{aligned}J_1 &= \parallel M - (A_f - B_f K_f) \parallel_2\\ J_2 &= \parallel N - B_f G_f \parallel_2\end{aligned} \tag{1.41}$$

[1] 2-norm is defined as $\parallel A \parallel_2 = \sqrt{\lambda_{max}\{A^T A\}}$ and $\parallel A \parallel_2 = \sqrt{\lambda_{max}\{A^* A\}}$ if A is complex

The optimal solution can be computed by using:

$$\begin{aligned} K_f^* &= argmin J_1 = B_f^+(A_f - M) \\ G_f^* &= argmin J_2 = B_f^+ N \end{aligned} \tag{1.42}$$

Where B_f^+ denotes the pseudo-inverse of B_f. The optimal solution obtained $\left(K_f^*, G_f^*\right)$ is plugged into the loop instead of the nominal controller, see Fig1.11. This method does not guarantee the stability of the reconfigured system because the optimisation problem is unconstrained. In order to ensure system stability Gao and Antsaklis presented a modified pseudo-inverse method (MPIM) [30]. Here the unconstrained stability problem become a constrained one. It is formulated in terms of the stability robustness of linear systems with structured uncertainty. The stability problem is solved, however the high computational charge prevents its application in real time. In [76] Staroswiecki presented a computationally simpler approach based on a set of admissible models. The admissible model is chosen in such a way that robust stability of the system is assured. This technique is known as Admissible pseudo-inverse method (APIM).

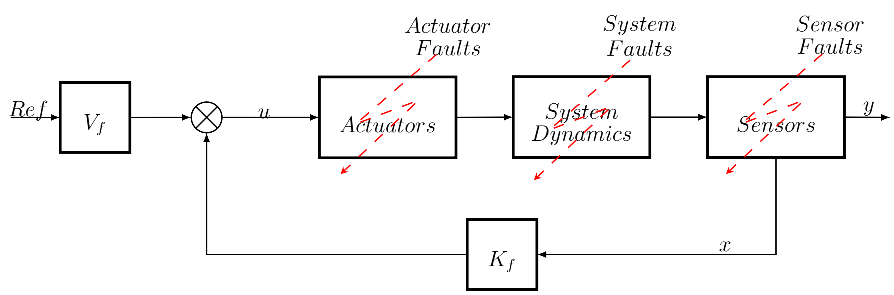

Figure 1.11: Pseudo-inverse Method [52]

Model following

Perfect model following This idea was presented in [31]. Here the model matching problem is solved by combining the use of a stabilizing feedback and a dynamic compensator in order to match exactly the dynamic behavior.

A closed loop linear system satisfies perfect model following with respect to the reference model (1.37) if and only if:

$$
\begin{aligned}
A + BK &= M \\
G &= N
\end{aligned}
\tag{1.43}
$$

Figure 1.12 shows the typical structure. The reference model is running in parallel with the plant and it is implemented in the controller. In this way the control input is:

$$
u(t) = K_e e(t) + (K_m x_m(t) + K_w w(t)) \tag{1.44}
$$

With K_e the stabilizing gain and K_m, K_w the model matching gains. $e(t)$ is defined as the difference between the state variables of the plant and the reference model, to achieve perfect model matching this error has to be equal to zero for all $t > 0$.

The model matching gains are determined to minimize:

$$
\| (M - A_f)x(t) + Nw(t) - B_f u(t) \|_2 = \| \dot{e}(t) - (A_f - B_f K_e)e(t) \|_2 \tag{1.45}
$$

Where the error dynamics is expressed by:

$$
\dot{e}(t) A_f e(t) + (M - A_f)x_m(t) + Nw(t) - B_f u(t) \tag{1.46}
$$

The solution of the equation (1.45) is given by:

$$
\begin{aligned}
K_m &= B_f^{+}(M - A_f) \\
K_w &= B_f^{+} N
\end{aligned}
\tag{1.47}
$$

This technique guarantees closed loop stability if the terms (A_f,B_f) are stabilizable.

Optimization

LQ Redesign This technique was presented in [47]. The main idea of this technique is to design a LQ-optimal nominal controller. After FDI a new LQ controller is designed online, using the faulty plant model. If the faulty plant is still controllable the LQ algorithm will find a new LQ-optimal controller.

Model predictive control (MPC) The main idea of the MPC technique is divided into three main steps. First a prediction of the future behavior of the process state/output is accomplished. Second, the future input signals are computed online at each step by minimizing a cost function under inequality constraints on the manipulated (control) and/or controlled variables. Finally, only the first of the vector control variables is applied on the controlled plant, and the previous step is repeated with new measured input/state/output variables.

To achieve control reconfiguration, it is necessary to update the internal plant model of the MPC controller. The MPC controller will find the optimal sequence using the update plant model. Since the computational charge is important, this technique is applied principally in slow dynamics systems.

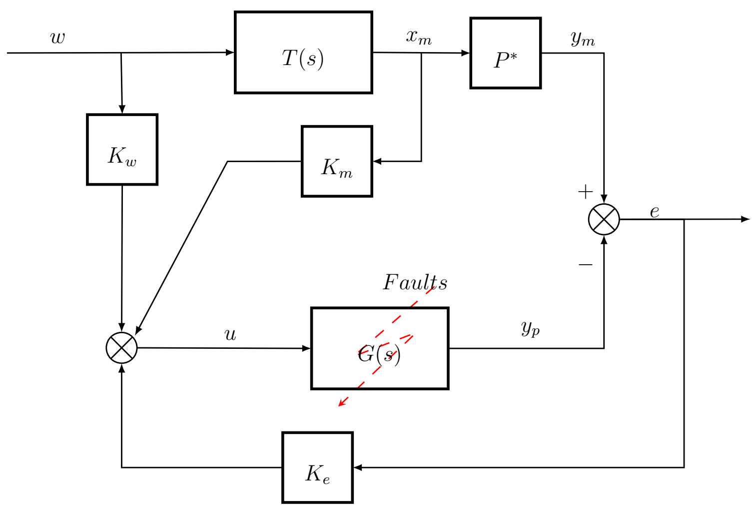

Figure 1.12: Model Following schema [52]

Fault-hiding

The main idea of these approaches is to keep the nominal controller during a fault occurrence. Fault recovery is then carried out by adding a reconfiguration block between the controller and the faulty plant. This block is designed in a manner that the faulty plant mimics the behavior of the unfaulty system. This behavior is obtained by means of two blocks, a virtual sensor, and a virtual actuator Fig. 1.13. A virtual sensor consists of a model of the faulty plant and a gain L. This block is in charge of providing estimates of the system states $\hat{x}$. The virtual actuator is a compound of a reference model, as well as feedback of the difference between the state of the reference model and two matrix M and N. These have to be chosen in such a way that the virtual actuator state is stable and the difference between the nominal output and the real one equals to 0.

In this way sensor and actuator faults can be handled, this approach can be applied to linear [52] and nonlinear [72] systems.

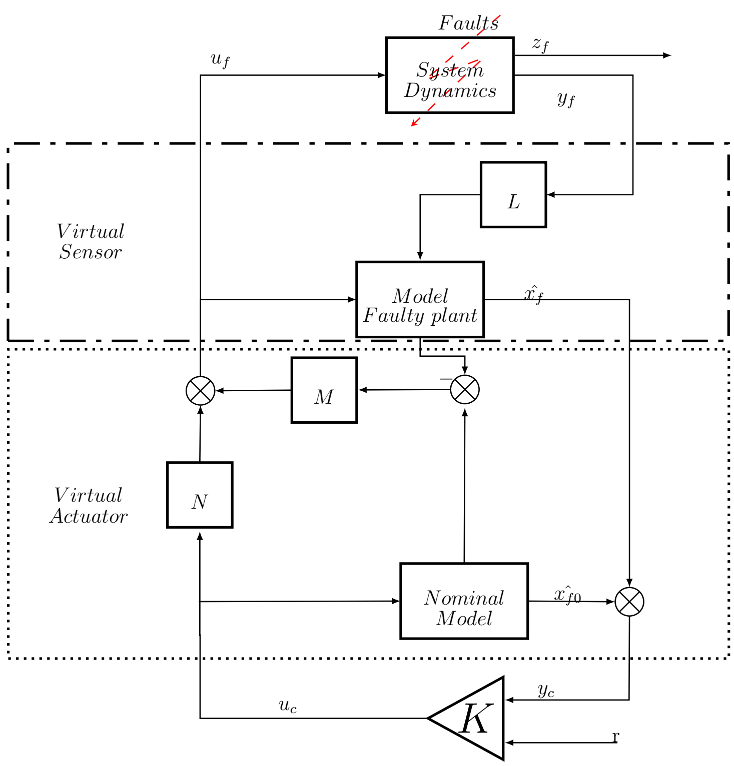

Figure 1.13: Fault-hiding approach [52]

Learning control

This approach is based in the idea of a mix of classical control techniquesand a learning control methods (Neural networks, expert systems, etc.). In this way a database of performance measures is constructed and a decision-making unit decides how to face the fault, [79].

1.7 CONCLUSION

This chapter presented an overview of some model-based FDI techniques and fault reconfiguration approaches for linear and nonlinear systems. The concept of differentially flat systems is presented as well. Flat systems possess an inherent capability to generate analytical redundancy. This property can be exploited to create FDI schemes. The two final sections were devoted to present the FDI and FTC techniques presented currently in the literature.

In the next chapter, the FTC flatness-based proposed approach is detailed. This approach differs from those presented here in that both FDI and fault recovery are carried out by exploiting the inherent properties of the flat systems.

CHAPTER 2

FAULT TOLERANT CONTROL: A FLATNESS-BASED APPROACH

Abstract:

In this chapter the proposed FTC approach is presented. It is based on the fact that the set of flat outputs is not unique. In fact if a second set of flat outputs which are not simply coupled by an algebraic equation but by a differential equation of the first is found, this will provide redundancy, which will increase the number of residuals, facilitating in this manner the fault detection. Additionally the redundant signals will be used to reconfigure the system after fault.

2.1 INTRODUCTION

This chapter presents the proposed approach of this research work.

Chapter 2 is divided in two main parts. In the first, a state of the art of FDI/FTC flatness-based techniques is presented, together with the mathematical theory that helps to found the flat outputs. The necessary conditions to use the are defined as well. In the second, the FTC flatness-based approach is presented. Firstly the attention is only to focus on the FDI technique, and then using the inherent characteristics of the flat systems, the FDI technique is extended to reconfigure the faulty system. In this way the fault detection and fault reconfiguration tasks are done in an integrated manner.

In order to verify the behavior of the system and decide if a fault is present or not, it becomes necessary to generate residual signals. Thanks to its inherent properties differential flatness can be used to generate redundancy in a natural way. This property will be exploited to generate parity equations. Those equations are comparisons between the behavior of the system and the behavior of a flat model of the fault-free case. The resultant will be the residual signal, which will be different from zero in presence of a fault and close to zero in the fault-free case.

The differential flatness property is already proven for many systems [45, 57, 75], the development of algorithms to automatize the computation of flat outputs is an active research area, Quadrat and Robertz present in [71], a method based on module theory. Jean Lévine in [46], presents a method based on the Smith decomposition of polynomial functions. The last reference inspires the FTC based approach presented in this work. This method will be presented in section 2.4. The next two sections are consecrated to present the sate of the art of FDI/FTC flatnes-based approaches.

2.2 Fault detection and isolation by flatness

Since the first publication of flat systems theory in the early 90's, they have attracted much attention in different automatic control areas, such as controller design [35, 56, 65, 74, 90] and motion planning [10, 11]. However, regarding FDI/FTC, not many works are published. See for instance [38, 48, 54, 66, 81, 91]. Differential flatness could be used to create Unknown Input Observers, see for instance [3, 20]. Such observers are specially helpful for FDI process.

The main property of flat systems provides analytical redundancy, since, every control input and system state can be expressed as a function of the flat outputs, see definition 1.1. In this way parity equations could be computed by simply comparing measures versus estimations. The results of such parity equations are known as residual signals. If their amplitude is close to zero the system is working normally; if not, the system is considered faulty. Almost all the approaches presented in the literature take advantage of this property.

In [66] differential flatness is coupled with a nonlinear observer in order to construct the residual signals. See Fig. 2.1. The nonlinear observer is in charge of generating an estimation of the control inputs. Such estimation can now be directly compared to the estimation of the same variables, but this time obtained using the differentially flat equations. The main problem of this technique lies in the fact that the residuals are obtained by comparing two different approaches. Since both of them could differ in some aspects, for instance dynamic speed, this could create some false alarms because of the phase difference of both signals.

The flatness-based FDI approach presented in [53, 54], use an algebraic approach [25,26,61] to estimate actuator faults. Such estimations help to identify the fault. This work takes into account only additive faults. The fact that the fault is estimated will be specially useful to reject the fault.

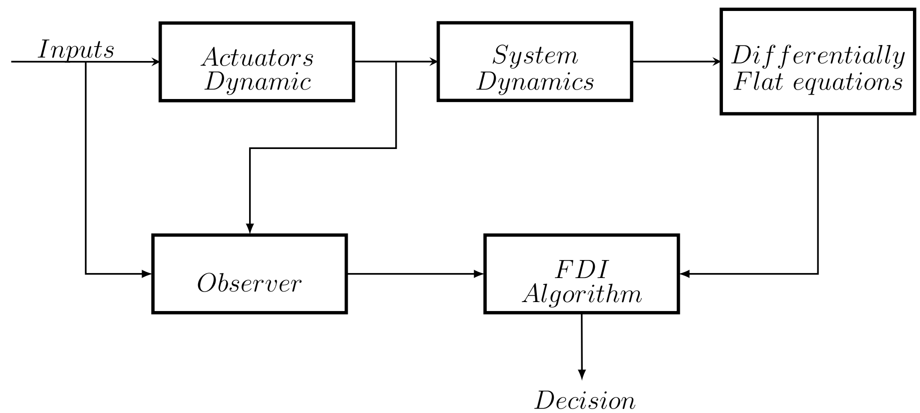

Figure 2.1: FDI flatness-based schema with observer, [66]

The analytical redundancy obtained thanks to the main property of the differentially flat systems is exploited to generate residual signals in [81]. The time derivatives of the flat output are computed by using B-splines. For this work the fault amplitude is estimated and this information is used to compensate for the fault. This approach is applied to linear systems.

In [48] and [91] the residuals are computed by comparing the state estimation to the state measurement. In order to overcome noise and modeling errors and improve the effectiveness of the threshold-based fault detection scheme, a probabilistic distribution is generated. This approach is applied to discrete nonlinear flat systems. The main problem of this approach is the fact that creating an online probabilistic distribution is difficult. To solve this issue authors coupled a simplified pre-computed distribution with neuro-fuzzy logic, which reduces the computational charge but increase the designing work. Those approaches are applied to discrete nonlinear flat systems.

In [38] the residual signals are computed by using the estimation of derivatives obtained with the algebraic approach presented in [25, 26, 61]. Fault indicators are robust with respect to uncertain parameters in the controlled plant.

Table 2.1 shows a summary of the FDI techniques found on the literature.

Reference	Applied in	Handling faults	Fault type	Advantages/Disadvantages
[66]	Nonlinear	Sensors	Multiplicative	Relatively easy design. Difference between dynamics could create false alarms.
[53, 54]	Linear Nonlinear	Actuators	Additive	Estimate the fault amplitude. Only additive faults are taken into account.
[81]	Linear	Sensors	Additive	Estimate the fault amplitude. Only additive faults are taken into account.
[48, 91]	Discrete nonlinear	System	Additive	Could facilitate the real time application. Hard design work.
[38]	Nonlinear	Actuators	Additive	Estimate the fault amplitude. Could be computationally expensive.

Table 2.1: FDI by flatness

2.3 FAULT TOLERANT CONTROL BY FLATNESS

This section is devoted to presenting some FTC flatness-based approaches. Suryawan et al. present in [81] an FTC approach, such approach is carried out by using the estimation of the faults. Such estimations are obtained using the differentially flat equations to compute a fault-free version of the states and then compare them versus the faulty sensor. Such operation will provide an estimate of the fault.See Fig. 2.2. Then the signal is conditioned using B-splines. The obtained trajectory is subtracted from the measure of the faulty sensor. Only faults affecting the sensors are taken into account. These approach is applied to linear systems with a focus on sensor faults.

As in the reference above in [54], the fault is estimated and then such information helps to recover the system from a faulty position. The main difference lies in the fact that this time the fault is estimated by using the algebraic approach. The approach is intended for actuator faults. According to the authors additive and multiplicative faults could be treated indistinctly.

The main disadvantage of both techniques is the fact that estimation plus signal conditioning could take some time to be accomplished. Such time delay could lead the system to instability.

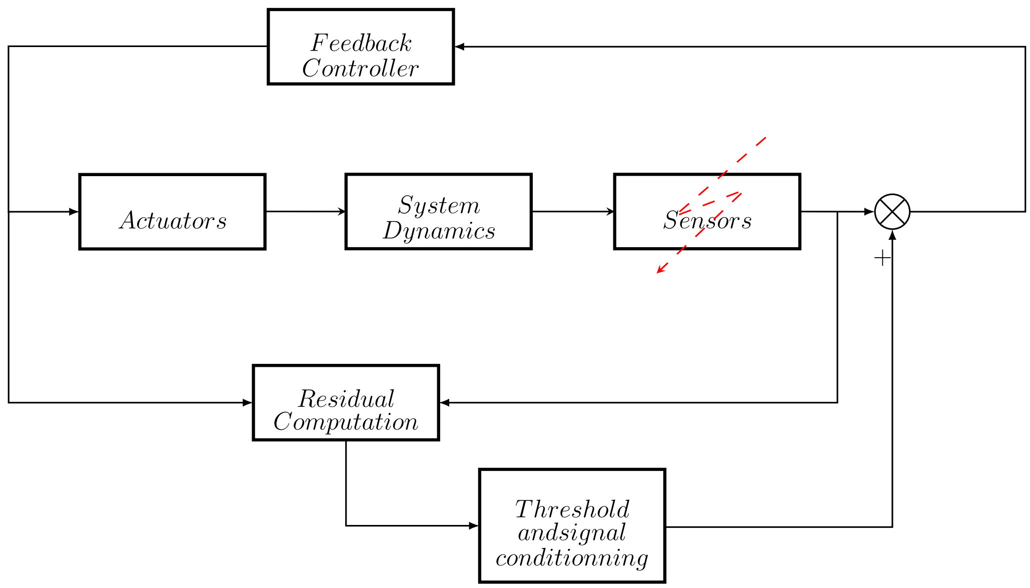

Figure 2.2: FDI flatness-based FTC schema, [81]

2.4 NECESSARY CONDITIONS FOR THE FTC PROPOSED APPROACH

This section is dedicated to presenting the mathematical theory that inspired the proposed FTC approach. The determination of the flat outputs could be carried out intuitively for some classical examples, however in order to check all the possibilities it could be more efficient to use formal calculations algorithms. In fact, the computation of flat outputs is based on a decomposition which is not unique. As consequence the set of flat outputs are neither. This property could be exploited for FDI, since this will provide redundancy and will increase the number of residual signals. Additionally this redundancy could be used to reconfigure the system after fault. More details of the theory presented in this section can be found in [46].

Let us consider a nonlinear system in its implicit form (where the input variables are eliminated).

$$F(x, \dot{x}) = 0 \tag{2.1}$$

An implicit system is defined as $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$, with $\mathfrak{X} = X \times \mathbb{R}^n_\infty$, $dimX = n$, $\tau_{\mathfrak{X}}$ the trivial Cartan field on $\mathfrak{X}$, and rank $\frac{\delta F}{\delta \dot{x}} = n - m$, this system will be Lie-Bäcklund equivalent to the implicit system $(\mathfrak{Y}, \tau_{\mathfrak{Y}}, G)$, with $\mathfrak{Y} = Y \times \mathbb{R}^p_\infty$, $dimY = p$, $\tau_{\mathfrak{Y}}$ the trivial Cartan field on $\mathfrak{Y}$, and rank $\frac{\delta G}{\delta \dot{y}} = p - q$ if and only if it exists a locally C^∞ mapping $\Phi : \mathfrak{Y} \mapsto \mathfrak{X}$, with locally C^∞ inverse Ψ. This implies that:

- $\Phi_* \tau_{\mathfrak{Y}} = \tau_{\mathfrak{X}}$ and $\Psi_* \tau_{\mathfrak{X}} = \tau \mathfrak{Y}$.
- For every $\bar{y}$ such that $L^k_{\tau_{\mathfrak{Y}}} G(\bar{y}) = 0, \forall k \geq 0$, then $\bar{x} = \Phi(\bar{y})$ satisfies $L^k_{\tau_{\mathfrak{X}}} F(\bar{x}) = 0, \forall k \geq 0$ and conversely.

Where L_{τ_n} is the lie derivative along the Cartan field τ_n.

Definition 2.1 *The implicit system $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$ is flat if and only if it is Lie-Bäcklund equivalent to the trivial system $(\mathbb{R}, \tau_m, 0)$.*

Theorem 2.2 *The system $(\mathfrak{X}, \tau_{\mathfrak{X}}, F)$ is flat, if, and only if there exists a locally C^∞ and invertible mapping $\Phi : \mathbb{R}^m_\infty \mapsto \mathfrak{X}$ such that:*

$$\Phi^* dF = 0 \tag{2.2}$$

Defining the polynomial matrices as follows:

$$dF = \frac{\partial F}{\partial x} dx + \frac{\partial F}{\partial \dot{x}} d\dot{x} = \left(\frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}} \frac{d}{dt} \right) \triangleq P(F) dx \tag{2.3}$$

$$P(\Phi_0) \triangleq \sum_{j \geq 0} \frac{\partial \Phi_0}{\partial y^{(j)}} \frac{d^j}{dt^j} \tag{2.4}$$

We thus can write

$$\Phi^* dF = P(F) P(\Phi_0) dy \tag{2.5}$$

By consequence, we have to find a polynomial matrix $P(\Phi_0)$ solution to

$$P(F) P(\Phi_0) = 0 \tag{2.6}$$

If F is restricted to be a meromorphic function [1], $P(\Phi_0)$ may be obtained via the Smith decomposition of $P(F)$.

[1] A meromorphic function on an open subset D of the complex plane is a function that is infinitely differentiable and equal to its own Taylor series on all D except a set of isolated points, which are poles for the function.

The variational system $P(F)$ could be decomposed using the Smith decomposition in:

$$VP(F)U = (I_{n-m}, 0_{n-m,m}) \tag{2.7}$$

Let us define $\mathfrak{K}$ as a field of meromorphic functions from $\mathfrak{X}$ to $\mathbb{R}$, $\mathfrak{K}\left[\frac{d}{dt}\right]$ as the principal ring of $\mathfrak{K}$-polynomials of $\frac{d}{dt} = L_{\tau_{\mathfrak{X}}}$, $\mathcal{M}_{p,q}\left[\frac{d}{dt}\right]$ the module of the $p \times q$ matrices over $\mathfrak{K}\left[\frac{d}{dt}\right]$, with p and q arbitrary integers, and, $\mathcal{U}\left[\frac{d}{dt}\right]$ is the group of unimodular matrices of $\mathcal{M}_{p,p}\left[\frac{d}{dt}\right]$.

By using this notation the set of hyper-regular matrices $P(\Phi) \in \mathcal{M}_{n,m}\left(\frac{d}{dt}\right)$ satisfying (2.2) is given by:

$$P(\Phi) = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} W \tag{2.8}$$

Where $U \in R - Smith(P(F))$ and $W \in \mathcal{U}_m\left(\frac{d}{dt}\right)$ is an arbitrary unimodular matrix.

Let us define:

$$\hat{U} = U \begin{pmatrix} 0_{n-m,m} \\ I_m \end{pmatrix} \tag{2.9}$$

Lemma 2.3 *For every matrix $Q \in L - Smith(\hat{U})$, it exists a matrix $Z \in \mathcal{U}_m\left(\frac{d}{dt}\right)$ such that:*

$$QP(\Phi) = \begin{pmatrix} I_m \\ 0_{n-m,m} \end{pmatrix} Z \tag{2.10}$$

Moreover, for every Q, the sub-matrix $\hat{Q} = (0_{n-m,m},\ I_{n-m})\, Q$ is equivalent to $P(F)$.

A flat output of the variational system is given by:

$$w(\bar{x}) = \begin{pmatrix} w_1(\bar{x}) \\ \vdots \\ w_m(\bar{x}) \end{pmatrix} = (I_m,\ )0_{m-n-m})\, Q\bar{x}dx_{|\mathcal{X}0} \tag{2.11}$$

if $dw = 0$, a flat output of the nonlinear implicit system (2.1) can be obtained by integrating the equation $dy = w$. Otherwise, it is necessary to find an integral base, if such base exists. This means that we have to find an integral factor $M \in \mathcal{U}_m\left(\frac{d}{dt}\right)$ verifying $d(Mw) = 0$.

Definition 2.4 *Strongly closed:*

The $\mathfrak{K}\left(\frac{d}{dt}\right)$-ideal Ω, finitely generated by the 1-forms $(w_1, ...w_m)$ defined by (2.11), is strongly closed in $\mathcal{X}_0$, (or equivalently, the system $(\mathfrak{X},\ \tau_{mathfrakX}, F)$ is flat) if and only if it exists an operator $\mu \in \mathcal{L}_1((\Lambda(\mathfrak{X}))^m)$, and a matrix $M \in \mathcal{U}_m\left(\frac{d}{dt}\right)$ such that:

$$dw = \mu w, \qquad \mathfrak{d}(\mu) = \mu^2, \qquad \mathfrak{d}(M) = -M\mu \tag{2.12}$$

Where $\mathcal{L}_1((\Lambda(\mathfrak{X}))^m)$ is the space of linear operators which maps the p-forms of dimension m in $\mathfrak{X}$ in (P+1)-forms of dimension m in $\mathfrak{X}$, $\mathfrak{d}$ represents the extension of the exterior derivative d, where the coefficients have their value in $\mathfrak{K}\left(\frac{d}{dt}\right)$.

Additionally if the relation (2.12) is satisfied, a flat output z can be obtained by integrating the system of equations $dy = Mw$.

This can be resumed in the next algorithm:

- Compute the variational system $P(F) = \frac{\delta F}{\delta x} + \frac{\delta F}{\delta \dot{x}}\frac{d}{dt}$ of (2.1), if $P(F)$ is not hyper-regular the system is not flat.
- Compute the smith decomposition of $P(F)$.
- Compute $\hat{U} = U \begin{bmatrix} 0_{n-m,m} \\ I_m \end{bmatrix}$.
- Obtain the Smith decomposition of $\hat{U}$.
- Compute the vector of 1-form ω defined in (2.11)
- Obtain the operator μ, such that $d\omega = \mu\omega$ by identification term by term, if possible.
- If not, among the possible operators μ, keep only the operators who verifies that $\mathfrak{d}(\mu) = \mu^2$.
- Determine by identification term by term, a matrix M, which validates $\mathfrak{d}(M) = -M\mu$.
- Between all the options of matrix M, keep only the unimodular matrices, if such matrix does not exist, the system is not flat. On the contrary, a flat output can be obtained by integrating the system of equations $dy = M\omega$.

Example 2.5 *Non holonomic car, see example 1.3*

The system equations of the non holonomic car in its implicit representation are:

$$F(x, y, \theta, \dot{x}, \dot{y}, \dot{\theta}) = \dot{x}sin\theta - \dot{y}cos\theta = 0 \tag{2.13}$$

The first step is to compute the variational system:

$$\begin{aligned} P(F) &= \left(\frac{\partial F}{\partial x} + \frac{\partial F}{\partial \dot{x}}\frac{d}{dt},\ \frac{\partial F}{\partial y} + \frac{\partial F}{\partial \dot{y}}\frac{d}{dt},\ \frac{\partial F}{\partial \theta} + \frac{\partial F}{\partial \dot{\theta}}\frac{d}{dt}\right) \\ &= \left(sin\theta\frac{d}{dt},\ -cos\theta\frac{d}{dt}\ \dot{x}cos\theta + \dot{y}sin\theta\right) \end{aligned} \tag{2.14}$$

Defining $E = \dot{x}cos\theta + \dot{y}sin\theta$, and permuting columns we can write the variational system as follows:

$$P(F) = \left(E,\ -cos\theta\frac{d}{dt},\ sin\theta\frac{d}{dt}\right) \tag{2.15}$$

After applying the Smith decomposition algorithm, we obtain the unimodular matrix U:

$$U = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ \frac{1}{E} & \frac{cos\theta}{E}\frac{d}{dt} & -\frac{sin\theta}{E}\frac{d}{dt} \end{bmatrix} \tag{2.16}$$

Defining $\hat{U}$ as:

$$\hat{U} = U\,(0_{1,2},\ I_2)^T = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ \frac{cos\theta}{E}\frac{d}{dt} & -\frac{sin\theta}{E}\frac{d}{dt} \end{bmatrix} \tag{2.17}$$

After computing the Smith decomposition of $\hat{U}$ we obtain:

$$Q = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \frac{sin\theta}{E}\frac{d}{dt} & -\frac{cos\theta}{E}\frac{d}{dt} & 1 \end{bmatrix} \tag{2.18}$$

Multiply the matrix Q by $(dx,\ dy,\ d\theta)^T$. The last line is equal to $\frac{1}{E}(sin\theta d\dot{x} - cos\theta d\dot{y} + (\dot{x}cos\theta + \dot{y}sin\theta)d\theta) = \frac{1}{E}d(\dot{x}sin\theta - \dot{y}cos\theta)$, which is equal to zero, see (2.14). The remaining part of the system:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} dx \\ dy \\ d\theta \end{bmatrix} = \begin{bmatrix} w1 \\ w2 \end{bmatrix} \tag{2.19}$$

is trivially strongly closed with $M = I_2$, which finally gives the same set of flat outputs $z = [y,\ x]^T$ presented in example 1.3.

By this way we can define the first set of flat outputs z_α as follows:

$$\begin{bmatrix} z_{\alpha 1} \\ z_{\alpha 2} \end{bmatrix} = \begin{bmatrix} y \\ x \end{bmatrix} \tag{2.20}$$

The set of flat outputs is not unique, since it depends on the decomposition of $P(F)$ and, as explained in [46], it is not unique neither. Let us illustrate this with the next example.

Example 2.6 *Non holonomic car (2)*

By right multiplying the variational system depicted in (2.15) by:

$$Q = \begin{bmatrix} cos\theta & 0 & 0 \\ sin\theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.21}$$

and using $sin\theta\frac{d}{dt}(cos\theta) - cos\theta\frac{d}{dt}(sin\theta) = -\dot{\theta}$. The Smith decomposition of $P(F)$ is given by:

$$U = \begin{bmatrix} cos\theta & -\frac{1}{\dot\theta}cos^2\theta\frac{d}{dt} & \frac{1}{\dot\theta}(\dot{x}cos\theta + \dot{y}sin\theta)cos\theta \\ sin\theta & 1 - \frac{1}{\dot\theta}sin\theta cos\theta\frac{d}{dt} & \frac{1}{\dot\theta}(\dot{x}cos\theta + \dot{y}sin\theta)sin\theta \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

The matrix $\hat{U}$ is equal to:

$$\hat{U} = \begin{bmatrix} -\frac{1}{\dot\theta}cos^2\theta\frac{d}{dt} & \frac{1}{\dot\theta}(\dot{x}cos\theta + \dot{y}sin\theta)cos\theta \\ 1 - \frac{1}{\dot\theta}sin\theta cos\theta\frac{d}{dt} & \frac{1}{\dot\theta}(\dot{x}cos\theta + \dot{y}sin\theta)sin\theta \\ 0 & 1 \end{bmatrix} \tag{2.23}$$

The Smith decomposition of $\hat{U}$ gives:

$$Q = \begin{bmatrix} -tan\theta & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{\dot\theta}sin\theta cos\theta\frac{d}{dt} & -\frac{1}{\dot\theta}cos^2\theta\frac{d}{dt} & -\frac{1}{\dot\theta}(\dot{x}cos\theta + \dot{y}sin\theta)cos\theta \end{bmatrix} \tag{2.24}$$

The vector of 1-forms w is given by:

$$w = [w_1\ w_2]^T = Q[dx,\ dy,\ d\theta]^T = [-tan\theta dx + dy,\ d\theta]^T \tag{2.25}$$

We have:

$$dw = [dw_1,\ dw_2]^T = \left[-\frac{1}{cos^2\theta}d\theta \wedge dx,\ 0\right]^T \tag{2.26}$$

Which proves that w is not closed.

We introduce the μ operator:

$$\mu = \begin{bmatrix} 0 & d\left(\frac{x}{cos^2\theta}\right)\wedge \\ 0 & 0 \end{bmatrix} \tag{2.27}$$

Such operator μ verifies $\mu^2 = 0$. Additionally, we have:

$$\mathfrak{d} = \begin{bmatrix} 0 & d\left(\frac{1}{cos^2}dx + 2\frac{xsin\theta}{cos^3\theta}d\theta\right)\wedge \\ 0 & 0 \end{bmatrix} \tag{2.28}$$

By componentwise identification:

$$M = \begin{bmatrix} 1 & -\frac{x}{cos^2\theta} \\ 0 & 1 \end{bmatrix} \tag{2.29}$$

Now, we compute the 1-form as follows:

$$Mw = \begin{bmatrix} \tan\theta dx + dy - \frac{x}{cos^2\theta}d\theta \\ d\theta \end{bmatrix} \tag{2.30}$$

This 1-form is closed. Let us define now the new set of flat outputs as z_β, such vector is compound as follows:

$$\begin{bmatrix} z_{\beta 1} \\ z_{\beta 2} \end{bmatrix} = \begin{bmatrix} y - xtan\theta \\ \theta \end{bmatrix} \tag{2.31}$$

Equations (2.20) and (2.31) presents two possible set of flat outputs in which at least one element inside the z_β vector is coupled by a differential equation. Analytical redundancy can now be computed as a straight consequence. We can verify that it exists at least one element inside the z_α vector which is coupled by a differential equation of one of the elements of z_β by analyzing the equations (1.10) and the expressions of the control inputs (1.11). From the kinematic equations of the car we can rewrite θ which is an element of the z_β vector could be written as follows:

$$\theta = cos^{-1}\left(\frac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}}\right) \tag{2.32}$$

Equation 2.32 proof that it exists a differential equation fulfilling the necessary conditions for the FTC proposed approach.

2.5 Analytical redundancy by flatness-based approach

Analytical redundancy can be achieved thanks to the main property of the flat systems, which dictates that any input or state can be written as functions of the flat outputs. This provides in a direct manner the redundancy needed to compute the residuals, which will indicate the presence of a fault. It will be close to zero if no fault is present and different from it, if a fault affects the system.

Residuals are computed by simply comparing the measured variable versus the estimated using the differentially flat equations. This approach is depicted in the figure 2.3, the main advantage of this method is the fact that the estimations of the states and the inputs are only functions of the flat outputs. This phenomenon helps to determine the size of a fault. See for example [54] and [81].

2.5.1 Flatness-based Fault Detection and Isolation

The main idea of the proposed approach is based in the principle that the set of flat outputs is not unique. In fact, one can find an infinite number of them. (Linked with the matrix M in the algorithm). The idea is to find two or more sets of flat outputs in which at least one element inside one of the vectors is not simply coupled by an algebraic equation but by a differential equation of the first. This will increase the number of residuals, in addition the residuals will be decoupled between them. As a result, this could increase the possibilities of isolating every single fault. Furthermore we consider that the flat outputs are states of the system or a linear combination of them, and they are considered measured or at least estimated.

Let us consider a nonlinear flat model of dimension n, and m control inputs, with z_α as first set of flat outputs, which corresponds to m components of the state vector; also suppose that the full state is measured, ($y = x$). It is always possible to compute n residuals:

- n - m state residuals, because the full state is supposed to be measured.
- m control inputs residuals.

The residual signals are computed by using

$$r^i_{jx} = x_{mk} - \hat{x}_k \tag{2.33}$$

$$r^i_{ju} = u_{ml} - \hat{u}_l \tag{2.34}$$

where x_{mk} and u_{ml} are the k_{th} and l_{th} measured states and control inputs respectively and $\hat{x}_k$ and $\hat{u}_l$ are the k_{th} and l_{th} states and control inputs calculated using the differentially flat equations, i still being the identifier of the set of flat outputs. In order to clarify the proposed approach, suppose that we have a nonlinear system composed by four states, $[x_1\ x_2\ x_3\ x_4]^T \in \Re^n$ and two control inputs $[u_1\ u_2]^T \in \Re^m$, as depicted in definition 1.1, the number of control inputs is equal to the number of flat outputs. As a result $[z] \in \Re^m$, suppose also that the nonlinear system is flat and that, additionnally, we can find not one but two set of flat outputs, for instance $z_\alpha = [z_{\alpha 1}\ z_{\alpha 2}]^T = [x_1\ x_2]^T \in \Re^m$ and $z_\beta = [z_{\beta 1}\ z_{\beta 2}]^T = [x_3\ x_4]^T \in \Re^m$.

In order to show the advantage of computing two sets of flat outputs covering the characteristics presented before, the proposed approach is divided into two cases.

2.5.2 CASE A: n RESIDUALS

Assume now, that only z_α vector exists, this hypothesis implies that:

- The maximal number of residuals is four.
- Sensor faults not affecting flat outputs can be isolated depending on the system.
- Flat output sensor faults can be detected but cannot be isolated.

The n residuals are obtained as follows:

$$\begin{bmatrix} r^\alpha_{1x} \\ r^\alpha_{2x} \\ r^\alpha_{1u} \\ r^\alpha_{2u} \end{bmatrix} = \begin{bmatrix} x_{m3} \\ x_{m4} \\ u_{m1} \\ u_{m2} \end{bmatrix} - \begin{bmatrix} \phi_{\alpha x}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(a)}\right)(e_3)^T \\ \phi_{\alpha x}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(a)}\right)(e_4)^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right)(c_1)^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right)(c_2)^T \end{bmatrix} \tag{2.35}$$

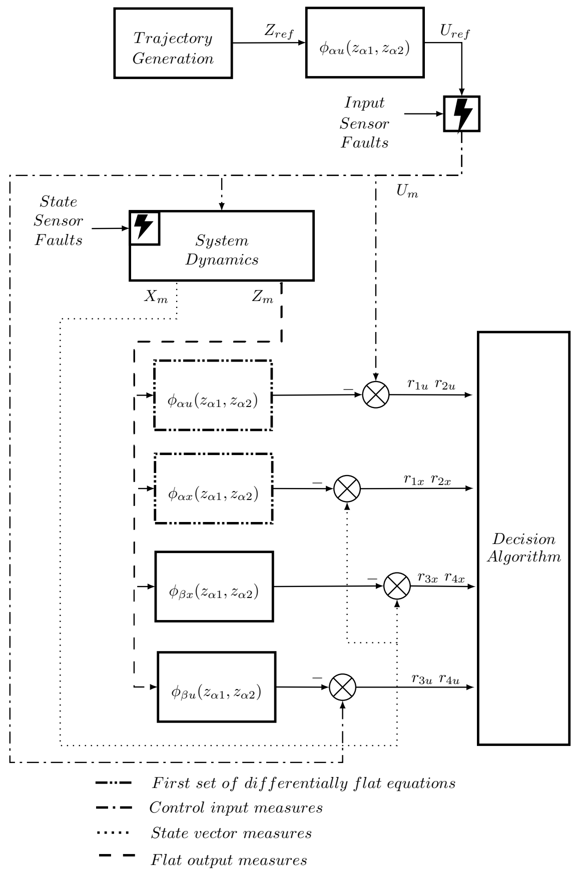

Figure 2.3: Detection diagram

Fault	r^{α}_{1x}	r^{α}_{2x}	r^{α}_{1u}	r^{α}_{2u}
F_{x1}	1	1	1	1
F_{x2}	1	1	1	1
F_{x3}	1	0	0	0
F_{x4}	0	1	0	0
F_{u1}	0	0	1	0
F_{u2}	0	0	0	1

Table 2.2: Faults signature for n residuals

Where $e_a \in \Re^n$, $\forall a \neq k \ e_a = 0$, $e_a = 1 \Leftrightarrow a = k$, where $a = [1, 2, ..n]$ and $c_b \in \Re^m$, $\forall b \neq l \ c_b = 0$, $c_b = 1 \Leftrightarrow b = l$, where $b = [1, 2, ..m]$. Observing in detail equation 2.35, it is straightforward to see that if a fault affects the state measure of (x_{m3}), the residual r^{α}_{1x} will be affected. The rest of residuals are independent of this measure, so they will not be affected by the fault. A fault affecting the other state measure or the actuators can be treated in the same manner.

When a fault affects one of the flat outputs, all the residuals will be affected. As a result the fault can be detected but it cannot be isolated. The fault signatures are presented in Table 2.2.

2.5.3 CASE B: $n + n$ RESIDUALS

Suppose now, that two sets of flat outputs are found, (z_α and z_β), this hypothesis denotes that:

- The maximal number of residuals is eight.
- Sensor faults not affecting flat outputs can be detected and isolated.
- Unfaulty versions of the flat outputs coupled by a differential equation can be computed. This property is specially useful to reconfigure the system after fault. This method will be developed in section 2.6.

Fault	r_{1x}^{α}	r_{2x}^{α}	r_{1u}^{α}	r_{2u}^{α}	r_{1x}^{β}	r_{2x}^{β}	r_{1u}^{β}	r_{2u}^{β}
F_{x1}	1	1	1	1	1	0	0	0
F_{x2}	1	1	1	1	0	1	0	0
F_{x3}	1	0	0	0	1	1	1	1
F_{x4}	0	1	0	0	1	1	1	1
F_{u1}	0	0	1	0	0	0	1	0
F_{u2}	0	0	0	1	0	0	0	1

Table 2.3: Faults signatures for $n+n$ residuals

Using the two sets of flat outputs, eight residuals are computed, those are the next:

$$\begin{bmatrix} r_{1x}^{\alpha} \\ r_{2x}^{\alpha} \\ r_{1u}^{\alpha} \\ r_{2u}^{\alpha} \\ r_{1x}^{\beta} \\ r_{2x}^{\beta} \\ r_{1u}^{\beta} \\ r_{2u}^{\beta} \end{bmatrix} = \begin{bmatrix} x_{m3} \\ x_{m4} \\ u_{m1} \\ u_{m2} \\ x_{m1} \\ x_{m2} \\ u_{m1} \\ u_{m2} \end{bmatrix} - \begin{bmatrix} \phi_{\alpha x}\left(z_{\alpha}, \dot{z}_{\alpha}, ..., z_{\alpha}^{(a)}\right)(e_3)^T \\ \phi_{\alpha x}\left(z_{\alpha}, \dot{z}_{\alpha}, ..., z_{\alpha}^{(a)}\right)(e_4)^T \\ \phi_{\alpha u}\left(z_{\alpha}, \dot{z}_{\alpha}, ..., z_{\alpha}^{(b)}\right)(c_1)^T \\ \phi_{\alpha u}\left(z_{\alpha}, \dot{z}_{\alpha}, ..., z_{\alpha}^{(b)}\right)(c_2)^T \\ \phi_{\beta x}\left(z_{\beta}, \dot{z}_{\beta}, ..., z_{\beta}^{(a)}\right)(e_1)^T \\ \phi_{\beta x}\left(z_{\beta}, \dot{z}_{\beta}, ..., z_{\beta}^{(a)}\right)(e_2)^T \\ \phi_{\beta u}\left(z_{\beta}, \dot{z}_{\beta}, ..., z_{\beta}^{(b)}\right)(c_1)^T \\ \phi_{\beta u}\left(z_{\beta}, \dot{z}_{\beta}, ..., z_{\beta}^{(b)}\right)(c_2)^T \end{bmatrix} \tag{2.36}$$

Where $e_a \in \Re^n$, $\forall a \neq k \; e_a = 0$, $e_a = 1 \Leftrightarrow a = k$, where $a = [1, 2, ..n]$ and $c_b \in \Re^m$, $\forall b \neq l \; c_b = 0$, $c_b = 1 \Leftrightarrow b = l$, where $b = [1, 2, ..m]$.

This time if a fault affects x_1, all the z_α residuals and r_{1x}^{β} will be triggered; the fault is detected and isolated. The same principle can be now applied to every single fault affecting the system, either sensor or actuator faults. Table 2.3 presents the fault signature belonging to each fault.

2.5.4 DETECTION ROBUSTNESS

For this work, the fault detection is achieved by simply comparing the residual amplitude versus a fixed detection threshold.

The amplitude of the detection threshold is determined by running series of fault-free simulations of the system. Three different simulations are run, the first one by changing each parameter individually in the same percentage upwards and downwards. The two final simulations are run by varying all the parameters plus and minus the same percentage used in the previous simulation.

Finally the amplitude of the detection threshold is calculated by selecting the worst case among all the results of the simulations, plus a security margin. Such margin is added in order to avoid false alarms caused by the measure noise or modeling errors.

2.5.5 DERIVATIVES ESTIMATION

In order to compute the system states and the control inputs of the system, and consequently the residual signals, the time derivatives of the flat outputs of the system have to be estimated.

In this work a high-gain observer [83] is used to evaluate the time derivative of noisy signals.

In order to improve the performance of the high-gain observer, a low-pass filter is synthesized. The filter order is fixed regarding the maximal derivative inside the differentially flat equations, hence better noise filtering is obtained. Let us define the equation of the high-gain observer:

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u \tag{2.37}$$

Where:

$$\hat{A} = \begin{bmatrix} -\zeta_1/\epsilon & 1 & ... & ... & 0 \\ -\zeta_2/\epsilon^2 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -\zeta_{n-1}/\epsilon^{n-1} & ... & ... & 0 & 1 \\ -\zeta_n/\epsilon^n & ... & ... & ... & 0 \end{bmatrix} \tag{2.38}$$

And:

$$\hat{B} = \left[\begin{array}{ccccc} -\zeta_1/\epsilon & -\zeta_2/\epsilon^2 & \dots & -\zeta_{n-1}/\epsilon^{n-1} & -\zeta_n/\epsilon^n \end{array} \right]^T \tag{2.39}$$

The polynomial $S^n + \zeta_1 S^{n-1} + ... + \zeta_{n-1} S + \zeta_n$ is Hurwitz and $\epsilon << 0$. The transfer function from u to $\hat{x}$ when $\epsilon \Rightarrow 0$ is $T(s) = [1 \; S \; ... \; S^{n-2} \; S^{n-1}]^T$, the system acts as a differentiator under the consideration that the input u is continuous and derivable. In this case the $n-1$ derivatives are obtained directly from the state vector.

A possible selection of the coefficients $\zeta_i (i = 1, \cdots n)$ is in such a way that the frequency bandwidth of the signal to be derivated is in the frequency bandwidth of the filter $1/(S^n + \zeta_1 S^{n-1} + \cdots + \zeta_{n-1} S + \zeta_n)$ and the ϵ small enough.

2.6 CONTROL RECONFIGURATION FOR DIFFERENTIALLY FLAT SYSTEMS

In order to complement the FTC strategy, control accommodation or control reconfiguration is needed. This work is focused on control reconfiguration. These techniques have as a principal characteristic that they keep the nominal controller synthesized during the design phase. This property allows for reduction of the response time to a fault. This methodology has as a principal characteristic the fact that both stages FDI and reconfiguration are merged in the same block. This characteristic will reduce the time response after fault and could reduce the computational load of the processor. The goal here is to hide the fault to the controller by changing the faulty reference to an unfaulty one. Fig. 2.4 shows the reconfiguration diagram.

Let us review the example of a nonlinear flat model of dimension n with m control inputs, in order to show how basing two or more sets of flat outputs coupled by a differential equation will help to improve the reconfiguration technique. The method will be divided into two different cases.

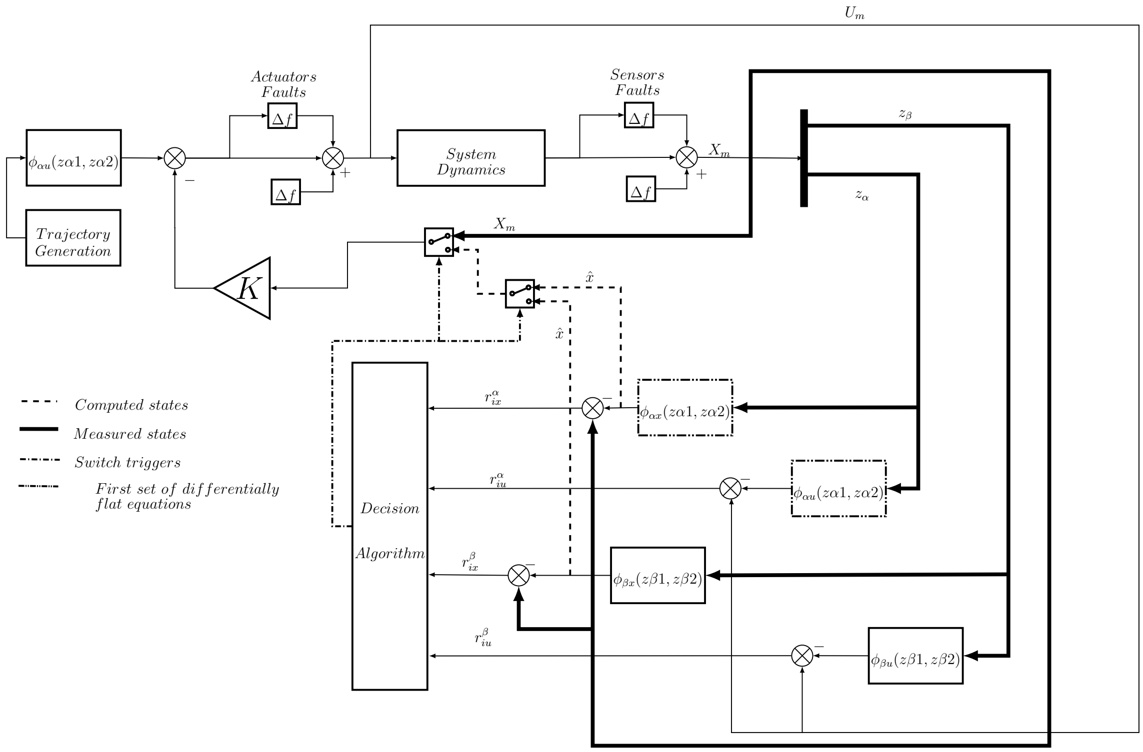

Figure 2.4: Reconfiguration diagram

2.6.1 CASE A: PARTIAL RECONFIGURATION

Partial reconfiguration is achieved if only one set of flat outputs is found. For example $z\alpha$, as explained in section 2.5.2, faults affecting flat outputs cannot be isolated, see table 2.2, however faults affecting measuring sensors can be detected and isolated. Thanks to the properties of the flat system and the fact that the flat outputs are considered fault-free at any time, we can compute the rest of system states. Those signals can then be used to reconfigure the system.

Empirically the number of reconfigurable faults can be obtained by using the next formula:

$$N_{FLAR} = (FOS)(n - m) \tag{2.40}$$

Where FOS is the number of sets of flat outputs found, n is the state dimension and m is the number of control inputs. For instance for our example the number of redundant signals is $(1)(4-2) = 2$, which are in fact the two states that are not flat outputs, x_3 and x_4.

2.6.2 CASE B: FULL RECONFIGURATION

Suppose now that two sets of flat outputs are found, z_α and z_β. Using the $n + n$ residuals every single fault can be detected and isolated, and additionally, the unfaulty set of flat outputs can be used to estimate the faulty state. Then, this new version can be used to feed the controller and reconfigure the system. See Fig. 2.4.

Let us analyze a fault affecting x_1. As is seen in the detail of equation (2.36), it is straightforward to see that all the equations containing $z\alpha 1$ in the right hand will modify their shape. Since the fault affects the measure of the first state, the first residue r_{1x}^{α} will be affected as well. However a faulty-free version of x_1 is computed using the measures of the z_β set. This signal could be used to hide the fault to the controller. Each fault could be treated in the same manner, which proves that the system is fully reconfigurable.

2.7 CONCLUSION

This chapter presents the proposed approach for FTC. Additive and multiplicative faults affecting sensors and actuators can be treated in the same manner for detection and isolation. Active reconfiguration is only carried out for sensor faults; actuators faults are rejected by the controller, it means that it is robust against actuators faults.

The main advantage of the proposed approach is that it merge the FDI process together with the reconfiguration. This adds simplicity during design and could reduce the time response to a fault and the computational load as well.

The next chapter is devoted to investigation of the feasibility of the proposed approach. For this, two nonlinear systems will be considered: An unmanned quadrotor and the three tank system. Both of them belong to the group of flat systems. The technique proposed can be partially applied in the unmanned quadrotor, and will be fully applied in the three tank system.

CHAPTER 3

FAULT TOLERANT CONTROL: APPLICATIONS

Abstract:

In this chapter, the feasibility of the proposed approach is investigated in two nonlinear systems. First, it is applied in a partial way to an unmanned quadrotor and, second, the full technique is applied to a three tank system.

3.1 INTRODUCTION

This chapter is devoted to investigate the feasibility of the proposed FTC approach. It is divided into two main parts. The first part is devoted to presenting the nonlinear model of an unmanned quadrotor. For this nonlinear system the FTC technique is not fully applied, since this system does not meet all the necessary conditions. However, as explained in sections 2.5.2 and 2.6.1, the technique can be applied in a partial manner.

The second part is devoted to apply the technique to a three tank system. This system in contrast to the UAV meets all the necessary conditions to fully exploit the proposed approach.

3.2 UNMANNED QUADROTOR

The American Institute of Aeronautics and Astronautics (AIAA) defines an unmanned aerial vehicle (UAV) as “an aircraft which is designed or modified, not to carry a human pilot and is operated through electronic input initiated by the flight controller or by an on-board autonomous flight management control system that does not require flight controller intervention” [68]. (See Fig. 3.1) The most important characteristic of this kind of vehicle is that they can be recovered at the end of the mission. This property excludes rockets, missiles, shells, etc. UAV’s have been serving the army since the 90’s, however, thanks to their versatility, and the progress in electronics manufacturing, they are nowadays being used in civil applications, [24] for example:

- Remote sensing and earth science research, [36].
- Search and rescue in human hostile zones (e.g., radiation zones, unstable zones after an earthquake).
- Weather monitoring.
- Crop spraying and dusting.
- Fire fighting, [7].

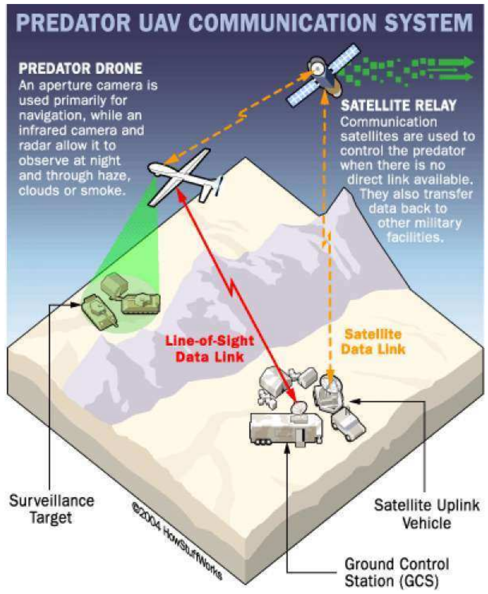

Figure 3.1: UAV communication system

- Communications networks, [21].

The European Association of Unmanned Vehicles Systems (EUROUVS) has drawn up a clasiffication of UAV systems based on such parameters as flight altitude, endurance, speed, maximum take off weight, size, etc. This classification is shown in table 3.1.

	Category (Acronym)	Maximum take-off Weight (kg)	Maximum flight altitude (m)	Endurance (hours)	Data link range (Km)	Example	
						Missions	Systems
Micro/ mini UAV's	Micro (MAV)	0.10	250	1	<10	Scouting, surveillance inside buildings.	Black widow, Microstar, Fancopter, Mosquito.
	Mini	<30	150-300	<2	<10	Film and broadcast ind., agriculture, pollution measurements, communications relay.	Mikado, Aladin, Tracker, Dragon eye, Raven, Skorpio, Robocopter, Pointer II, YH-300SL.
Tactical UAV's	Close Range (CR)	150	3,000	2-4	10-30	Mine detection, search and rescue.	Observer I, Phantom, Copter 4 Robocopter 300, Camcopter.
	Short Range (SR)	200	3,000	3-6	30-70	mine detection.	Luna, SilverFox, EyeView, Hornet.
	Long Range (LR)	-	5,000	6-13	200-500	Communications relay.	Hunter, Vigilante 502.
	Endurance (EN)	500-1500	5,000-8,000	12-24	>500	Battle damage assessment.	Aerosonde, Shadow 600.
	Medium Altitude, Long Endurance (MALE)	1,000-1,500	5,000-8,000	24-48	>500	Weapons delivery, Communications relay.	Skyforce, Heron TP, MQ-1 Predator, Darkstar. Eagle 1 and 2,
Strategic UAV's	High Altitude, Long Endurance (HALE)	2,500-12,500	15,000-20,000	24-48	>2000	boost phase intercept launch vehicle, airport security.	Global Hawk, Raptor, Condor, Theseus, Helios, Libellule, EuroHawk.
Special Task UAV's	Lethal (LET)	250	3,000-4,000	3-4	300	Anti-radar,anti-aircraft.	MALI, Harpy, Lark, Marula.
	Decoys (DEC)	250	50-5,000	<4	0-500	Aerial and naval deception.	Flyrt, MALD, Nulka, ITALD, Chukar.
	Stratospheric (Strato)	-	20,000-30,000	>48	>2,000	-	Pegasus.
	Exo-stratospheric (EXO)	-	>30,000	-	-	-	MarsFlyer, MAC-1.

Table 3.1: UAV's Classification

3.2.1 NONLINEAR MODEL

The operation of the quadrotor is fairly simple. The position ($\xi = x, y, z$) and the orientation ($\eta = \psi, \theta, \phi$) desired are achieved by independently varying the speed and torque of the four rotors. See figure 3.2.

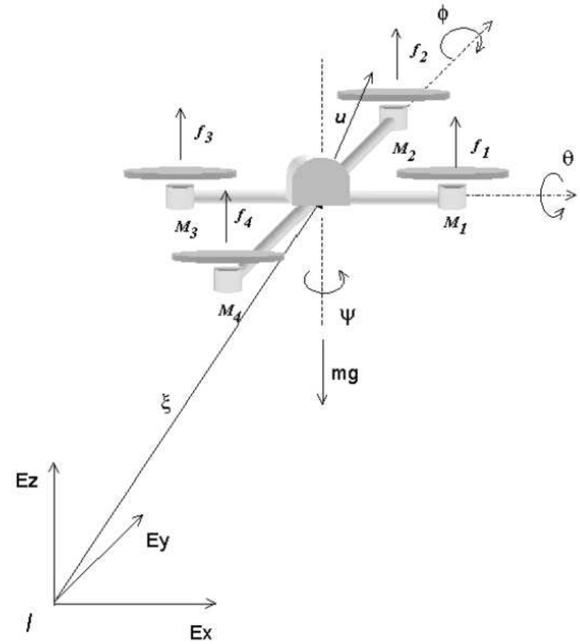

Figure 3.2: Quadrotor schema

Vertical movement is accomplished by adding the lift forces generated for each rotor, in order to avoid the helicopter turning over the z axis, two rotors turn in the clockwise direction (rotors 2 and 4) and the two others turn in the counterclockwise direction, this configuration cancel the horizontal moment of the helicopter, which is specially helpful during the hover position. The pitch moment (θ) is achieved by varying the rotation speeds of the rotors 1 and 3. The roll (ϕ) is obtained by varying the rotation speeds of the rotors 2 and 4. Finally, the yaw moment (ψ) is obtained from the torque resulting from substracting the clockwise (rotors 2 and 4) and from that of counterclockwise (rotors 1 and 3).

The nonlinear model can be obtained by using the motion equations of Euler-Lagrange. The Lagrangian (L) is defined as the addition of the kinetics (T) and the potential (U) energies.

$$L = T_T + T_R + U \tag{3.1}$$

Where T_T stands for the translational kinetic energy and T_R describes the rotational kinetic energy.

The Lagrangian is defined as follows, see [8] for further details:

$$L = \frac{m}{2}\dot{\xi}^T\dot{\xi} + \frac{1}{2}\dot{\eta}^T\mathbb{J}\dot{\eta} - mgz \tag{3.2}$$

Which satisfies the Euler-Lagrange equation.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \mathbf{F}_L \tag{3.3}$$

Where $\mathbf{F}_L$ stands for the forces ($f = R\mathbf{F}_L$) and moments (τ) applied to the body frame of the aircraft. Because the Lagrangian does not contain cross terms combining the position and the orientation, the Euler-Lagrange equation can be divided in the dynamics of the ξ (Translational) and η (Rotational) coordinates individually.

Where R stands for the rotational matrix which represents the orientation of the aircraft relative to a fixed inertial frame. $\mathbf{F}_L$ is defined as follows:

$$\mathbf{F}_L = \begin{bmatrix} 0 \\ 0 \\ u \end{bmatrix} \tag{3.4}$$

$$u = f_1 + f_2 + f_3 + f_4 \tag{3.5}$$

Where $f_i, i = 1, 2, 3, 4$ is the force produced for each one of the rotors, $f_i = k_i w_i^2$. The pitch, yaw and roll moments are written as follows:

$$\tau = \begin{bmatrix} \tau_\psi \\ \tau_\theta \\ \tau_\phi \end{bmatrix} = \begin{bmatrix} \Sigma_{i=1}^4 \tau M_i \\ (f_2 - f_4)l \\ (f_3 - f_1)l \end{bmatrix} \tag{3.6}$$

Where l is the distance between rotors and the center of gravity, and τM_i is the moment produced by the motor i.

Finally, the nonlinear model can be obtained by solving the Euler-Lagrange equations for position and orientation. For position:

$$\frac{d}{dt}\left(\frac{\partial L_T}{\partial \dot{\xi}}\right) - \frac{\partial L_T}{\partial \xi} = f \tag{3.7}$$

Substituting the value of L_T and adding the potential energy because it cause a movement in the z axis, we obtain:

$$\frac{d}{dt}\left(\frac{\partial \frac{m}{2}(\dot{x}^2+\dot{y}^2+\dot{z}^2)+mgz}{\partial \dot{\xi}}\right) - \frac{\partial \frac{m}{2}(\dot{x}^2+\dot{y}^2+\dot{z}^2)+mgz}{\partial \xi} = f \tag{3.8}$$

By computing the derivative we obtain:

$$\frac{d}{dt}\left(\frac{m}{2}(2\dot{x}+2\dot{y}+2\dot{z})\right) + 0 - 0 - 0 - mg = f \tag{3.9}$$

Finally computing the time derivative and rearranging in vector form, we obtain the equations related to the position coordinates.

$$f = \begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z} - mg \end{bmatrix} \tag{3.10}$$

For the orientation coordinates:

$$\frac{d}{dt}\left(\frac{\partial L_R}{\partial \dot{\eta}}\right) - \frac{\partial L_R}{\partial \eta} \tag{3.11}$$

Substituting we obtain:

$$\frac{d}{dt}\left(\frac{\partial(\frac{1}{2}\dot{\eta}^T\mathbb{J}\dot{\eta})}{\partial\dot{\eta}}\right)-\frac{\partial(\frac{1}{2}\dot{\eta}^T\mathbb{J}\dot{\eta})}{\partial\eta} \tag{3.12}$$

Computing the derivatives:

$$\frac{d}{dt}\left(\frac{1}{2}\left(\frac{\partial\dot{\eta}^T}{\partial\dot{\eta}}+0+\dot{\eta}^T\mathbb{J}\frac{\partial\dot{\eta}}{\partial\dot{\eta}}\right)\right)-\frac{1}{2}\left(0+\frac{\partial}{\partial\eta}(\dot{\eta}^T\mathbb{J}\dot{\eta}+0\right)=\tau \tag{3.13}$$

Computing the time derivative:

$$\mathbb{J}\ddot{\eta}+\dot{\mathbb{J}}\dot{\eta}-\frac{1}{2}\left(\frac{\partial}{\partial\eta}\left(\dot{\eta}^T\mathbb{J}\dot{\eta}\right)\right)=\tau \tag{3.14}$$

In order to write the equation above in the general form $M(\eta)\ddot{\eta}+C(\eta,\dot{\eta})\dot{\eta}=\tau$ we factorize $\dot{\eta}$ to the right as follows:

$$\mathbb{J}\dot{\eta}+\left(\dot{\mathbb{J}}-\frac{1}{2}\frac{\partial}{\partial\eta}(\dot{\eta}^T\mathbb{J})\right)\dot{\eta}=\tau \tag{3.15}$$

In this way we can define the coriolis matrix ($C(\eta,\dot{\eta})$) and the inertial matrix as follows:

$$C(\eta,\dot{\eta})=\dot{\mathbb{J}}\dot{\eta}-\frac{1}{2}\left(\frac{\partial}{\partial\eta}\left(\dot{\eta}^T\mathbb{J}\dot{\eta}\right)\right) \tag{3.16}$$

$$M(\eta)=\mathbb{J}(\eta)=W_\eta^T\mathbf{I}W_\eta \tag{3.17}$$

Finally the nonlinear dynamical model of the quadrotor is:

$$f=\begin{bmatrix} m\ddot{x} \\ m\ddot{y} \\ m\ddot{z}-mg \end{bmatrix} \tag{3.18}$$

$$\tau = M(\eta)\ddot{\eta} + C(\eta, \dot{\eta})\dot{\eta} \tag{3.19}$$

In order to simplify the model let us introduce a change of input variables proposed in [8].

$$\tilde{\tau} = \begin{bmatrix} \tilde{\tau_\psi} \\ \tilde{\tau_\theta} \\ \tilde{\tau_\phi} \end{bmatrix} = M(\eta)^{-1}\left(\tau - C(\eta, \dot{\eta})\dot{\eta}\right) \tag{3.20}$$

Where $\tilde{\tau} = \ddot{\eta}$, are the new inputs, after this transformation the nonlinear model becomes:

$$\begin{array}{rclrcl} m\ddot{x} &=& -u_1 sin\theta & \ddot{\psi} &=& u_2 \\ m\ddot{y} &=& u_1 cos\theta sin\phi & \ddot{\theta} &=& u_3 \\ m\ddot{z} &=& u_1 cos\theta cos\phi - mg & \ddot{\phi} &=& u_4 \end{array} \tag{3.21}$$

In this way the nonlinear model is compound by twelve states, $X = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ \psi\ \theta\ \phi\ \dot{\psi}\ \dot{\theta}\ \dot{\phi}]^T = [x_1\ x_2 x_3\ x_4\ x_5\ x_6\ x_7\ x_8\ x_9\ x_{10}\ x_{11}\ x_{12}]^T$ and the control inputs are $U = [u_1\ \ddot{\psi}\ \ddot{\theta}\ \ddot{\phi}]^T = [u_1\ u_2\ u_3\ u_4]^T$.

3.2.2 FLATNESS OF THE MODEL

The goal in the flatness approach is to explicitly express all the states and all the control inputs as functions of the flat outputs and a finite number of its time derivatives. From the equations 3.21, and defining the flat outputs as $z\alpha = [x\, y\, z\, \psi]^T$ [18], because we have four control inputs. We can write all the system states as function of the flat outputs $z\alpha$ and its time derivatives as follows:

$$
\begin{aligned}
x &= z_1 & \dot{x} &= \dot{z}_1 \\
y &= z_2 & \dot{y} &= \dot{z}_2 \\
z &= z_3 & \dot{z} &= \dot{z}_3 \\
\psi &= z_4 & \dot{\psi} &= \dot{z}_4 \\
\theta &= asin\left(\frac{m\ddot{z}_1}{-u_1}\right) & \dot{\theta} &= -m\left(\frac{z_1^{(3)}u_1 - u_1^2\ddot{z}_1}{u_1^2\sqrt{\alpha}}\right) \\
\phi &= atan\left(\frac{\ddot{z}_2}{\ddot{z}_3+g}\right) & \dot{\phi} &= \frac{z_2^{(3)}(\ddot{z}_3+g) - z_3^{(3)}\ddot{z}_2}{(\ddot{z}_3+g)^2+\ddot{z}_2{}^2}
\end{aligned}
\tag{3.22}
$$

In a similar way the control inputs are expressed as function of the flat outputs and its time derivatives.

$$
\begin{aligned}
u_1 &= m\sqrt{((\ddot{z}_1)^2+(\ddot{z}_2)^2+(\ddot{z}_3+g)^2)} \\
u_2 &= \ddot{z}_4 \\
u_3 &= -m\left(\frac{\left((z_1^{(4)}u_1-\ddot{u}_1\ddot{z}_1)u_1^2\sqrt{A}\right)-(z_1^{(3)}u_1-\ddot{u}_1\ddot{z}_1)(C)+2u_1\dot{u}_1\sqrt{A}}{u_1^4A}\right) \\
u_4 &= \frac{(z_2^{(4)}(\ddot{z}_3+g)+z_2^{(3)}z_3^{(3)}-z_3^{(4)}\ddot{z}_2-z_3^{(3)}z_2^{(3)})(z_2^{(3)}(\ddot{z}_3+g)-z^{(3)}\ddot{z}_2)-B}{(\ddot{z}_3+g)^4+(\ddot{z}_2)^4}
\end{aligned}
\tag{3.23}
$$

where

$$A = 1-\left(\frac{m\ddot{z}_1}{u_1}\right)^2 \tag{3.24}$$

$$B = (2(\ddot{z}_3+g)z_3^{(3)}+2\ddot{z}_2z_2^{(3)})(z_2^{(3)}(\ddot{z}_3+g)-z_3^{(3)}\ddot{z}_2) \tag{3.25}$$

$$C = (\frac{-2m\ddot{z}_1(mz_1^{(3)}u_1-\dot{u}_1m\ddot{z}_1)}{u_1\sqrt{\alpha}} \tag{3.26}$$

3.2.3 Flatness-based fault tolerant control of a Quadrotor UAV

Additive faults affecting sensors and control inputs (combination of actuators, see (3.6)) are considered. For sensors measuring x_{m5} (θ) and x_{m6} (ϕ) different fault amplitudes are considered. See Table 3.2. Such amplitude defines the FTC strategy used to counteract the fault. For sensors measuring flat outputs z_1, z_2 and z_3 an additive fault of one meter is considered. For the flat output z_4, $1°$ extra is applied. Only single faults are considered. Reconfiguration after fault is taken into account only for measuring sensors. Once the fault appears (50 s) it is recurrent until the end of the simulation, since the FTC strategy needs to know the amplitude of the fault in order to decide which strategy will be used. For simplicity sake in this work the fault amplitude is supposed perfectly known. By consequence the strategy choice is straightforward.

Fault	Amplitude	Strategy	Amplitude	Strategy	Amplitude	Strategy
F_{x5}	$< 1.8°$	P^1	$> 1.8° < 4.6°$	Rf^1	$>4.6°$	$Re^{1,2}$
F_{x6}	$< 1.8°$	P	$> 1.8° < 3.9°$	Rf	$>3.9°$	Re

Table 3.2: Additives faults for the UAV

[1] P =Passive, Rf =Reconfiguration, Re =Restructuring. [2] This approach is out of the bounds of this work.

Section 2.5.4 describes how the detection threshold is fixed. The parameter that changes is the mass (m) of the helicopter. The nominal value is 0.52Kg. The controller in charge of closing the loop is an LQR. The matrices Q and R are chosen in order to respect the power bounds of the actuators. The nominal trajectories are created using 5th order polynomials. White noise is added to the signal in order to simulate real operation. High gain observers are used to compute the time derivatives. Low-pass filters are coupled with the observers. A trade-off between the time delay caused by the filter and the cut-off frequency needs to be studied in detail. A very high cut-off frequency will not properly reduce the amplitude of the noise. On the other hand, the higher the frequency of the filter, the more important the induced time delay will be, this delay could prevent the use of the reconfiguration method, because, if the estimated signal is not in phase with the measured signal the fact of a change between references could drive the system to instability.

Fault detection and isolation

For this particular system only one set of flat outputs is found, by consequence $n = 12$ residuals are found, which is in fact the number of states, however, for simplicity sake the time derivatives of the three position states and the three orientation states, x_7 to x_{12} are consider unfaulty at any time, such supposition produces only six residuals, which are presented in equation (3.27).

$$\begin{bmatrix} r_{1x} \\ r_{2x} \\ r_{1u} \\ r_{2u} \\ r_{3u} \\ r_{4u} \end{bmatrix} = \begin{bmatrix} x_{m5} \\ u_{m6} \\ u_{m1} \\ u_{m2} \\ u_{m3} \\ u_{m4} \end{bmatrix} - \begin{bmatrix} \phi_x\left(z, \dot{z}, ..., z^{(a)}\right) \; [0_{(4)}\, 1\, 0_{(7)}]^T \\ \phi_x\left(z, \dot{z}, ..., z^{(a)}\right) \; [0_{(5)}\, 1\, 0_{(6)}]^T \\ \phi_u\left(z, \dot{z}, ..., z^{(b)}\right) \; [1\,0\,0\,0]^T \\ \phi_u\left(z, \dot{z}, ..., z^{(b)}\right) \; [0\,1\,0\,0]^T \\ \phi_u\left(z, \dot{z}, ..., z^{(b)}\right) \; [0\,0\,1\,0]^T \\ \phi_u\left(z, \dot{z}, ..., z^{(b)}\right) \; [0\,0\,0\,1]^T \end{bmatrix} \tag{3.27}$$

For faults affecting measuring sensors, two different frameworks are considered. See Table 3.2. However for FDI the fault amplitude is not a key parameter, since the fault signature is the same regardless of the amplitude fault. All residuals are normalized between -1 and +1, those boundaries represent the minimal and maximal amplitude of the fault-free threshold.

Let us analyze each individual fault. A fault affecting measure of x displacement should affect each one of the six residuals because it is a flat output. However, the system is naturally decoupled. As a consequence only the residuals depending on the x_{m1} measure are affected. Those are r_{1x}, r_{1u} and r_{3u}. See equations (3.22) and (3.23). See Fig. 3.3. This behavior is due to the operation of the UAV. Since the axis of the four rotors are fixed to the main frame (cannot tilt) horizontal displacement can only be obtained by tilting the entire frame in order to move the airplane. As a consequence the residuals which depends directly of θ are impacted. The residual r_{1u} is affected because it depends on the time derivative of x.

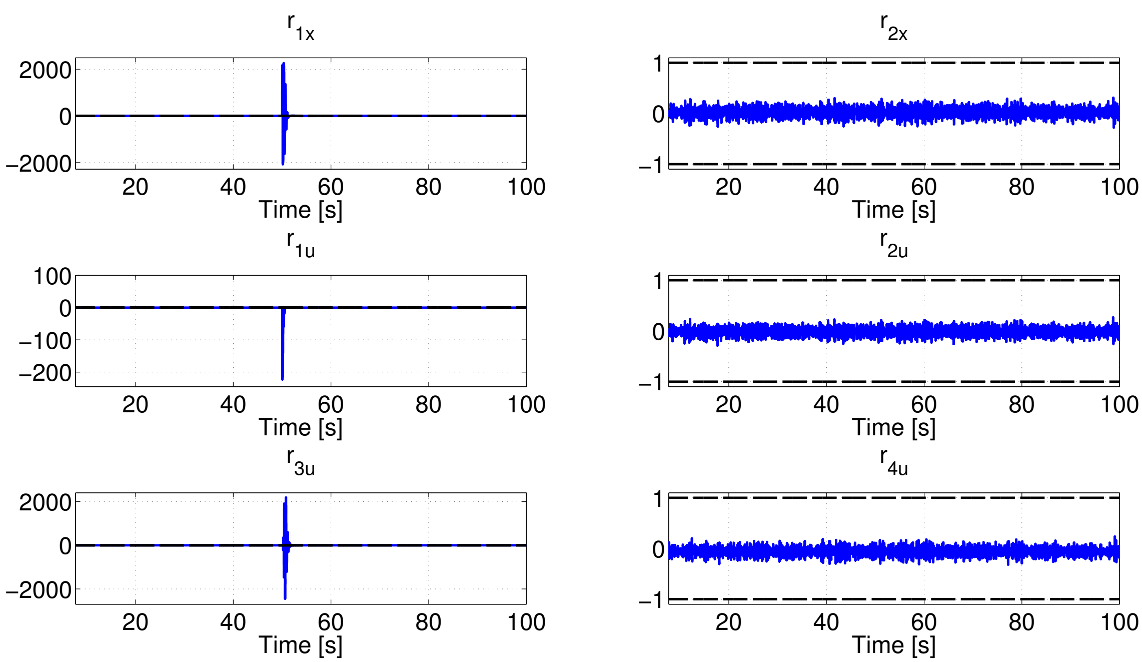

Figure 3.3: Additive fault measure x_1 residuals normalized

Figure 3.4 shows the residuals obtained after a fault of sensor y. This time the fault affects the residuals related to ϕ (r_{2x} and r_{4u}). This is due to the same phenomenon presented in the x axis. Once again the residue r_{1u} is affected because it depends on the faulty measure.

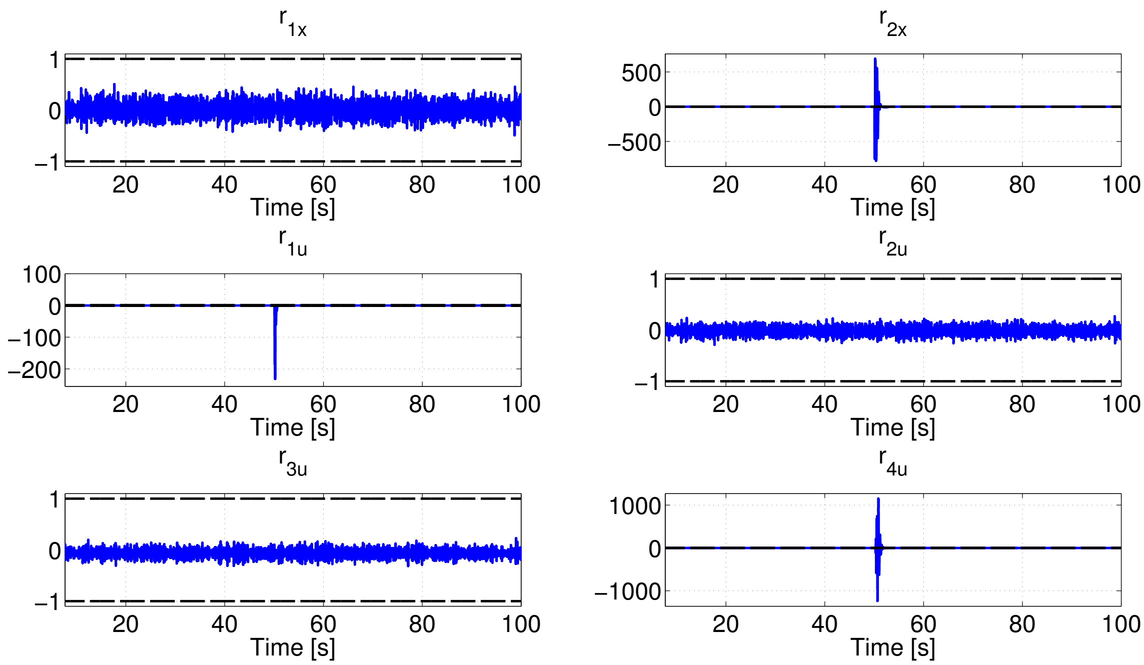

Figure 3.4: Additive fault measure x_2 residuals normalized

A fault affecting the high measure (z) will impact five of six residuals, because it is present directly or indirectly in the equations used to estimate the states and the control inputs. See equations (3.22) and (3.23). The residue not affected depends only on the yaw (ψ) movement of the airplane. As a result the residue r_{4u} is not affected by the fault effect.

Finally, Fig. 3.6 presents the residuals for a fault affecting the sensor measuring the yaw angle. Residual r_{4u} is directly related to this measure, and so it is triggered.

Fault affecting the pitch angle, x_{m5} will trigger all the residuals depending on θ, such residuals are r_{1x} and r_{2u}. However, even if the residue r_{1u} is affected indirectly (via the x displacement) the amplitude change is not enough to exceeds the threshold, see Fig. 3.7.

Roll angle x_{m6} is quite similar. It differs in the fact that in this case the residue r_{1u} is affected by the y displacement. Fig. 3.8.

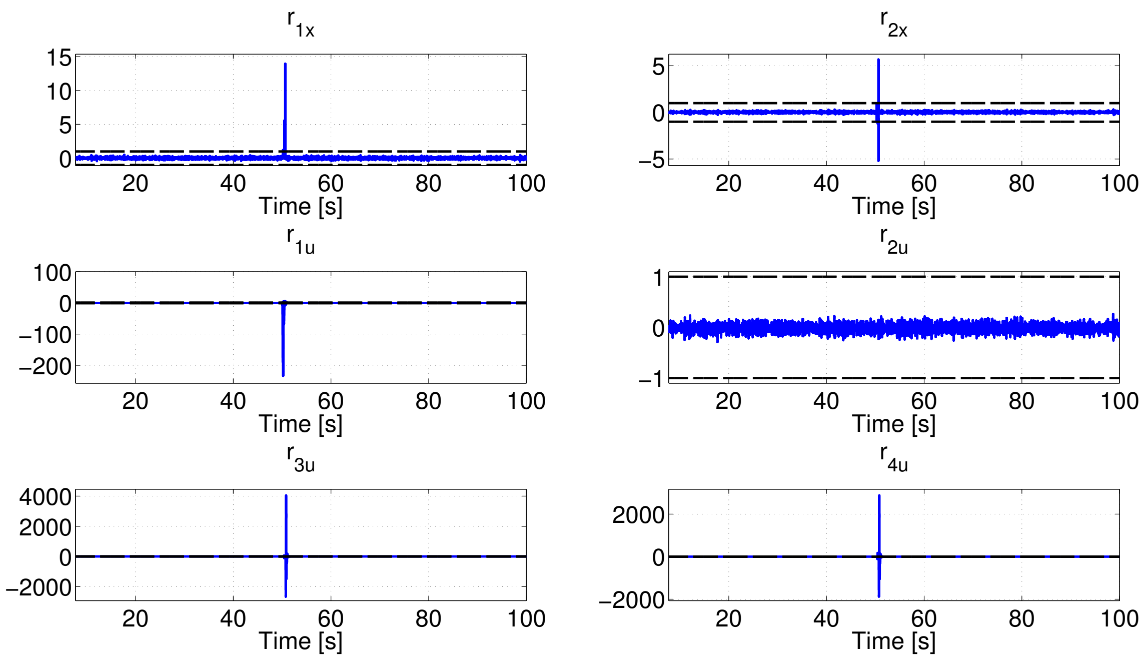

Figure 3.5: Additive fault measure x_3 residuals normalized

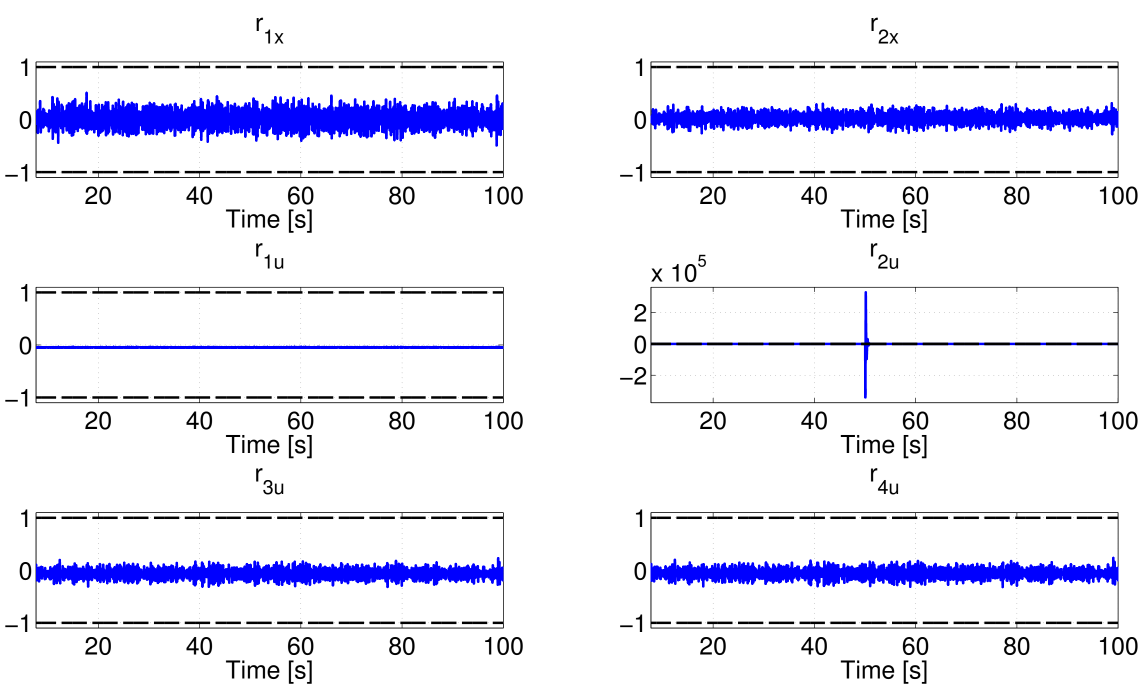

Figure 3.6: Additive fault measure x_4 residuals normalized

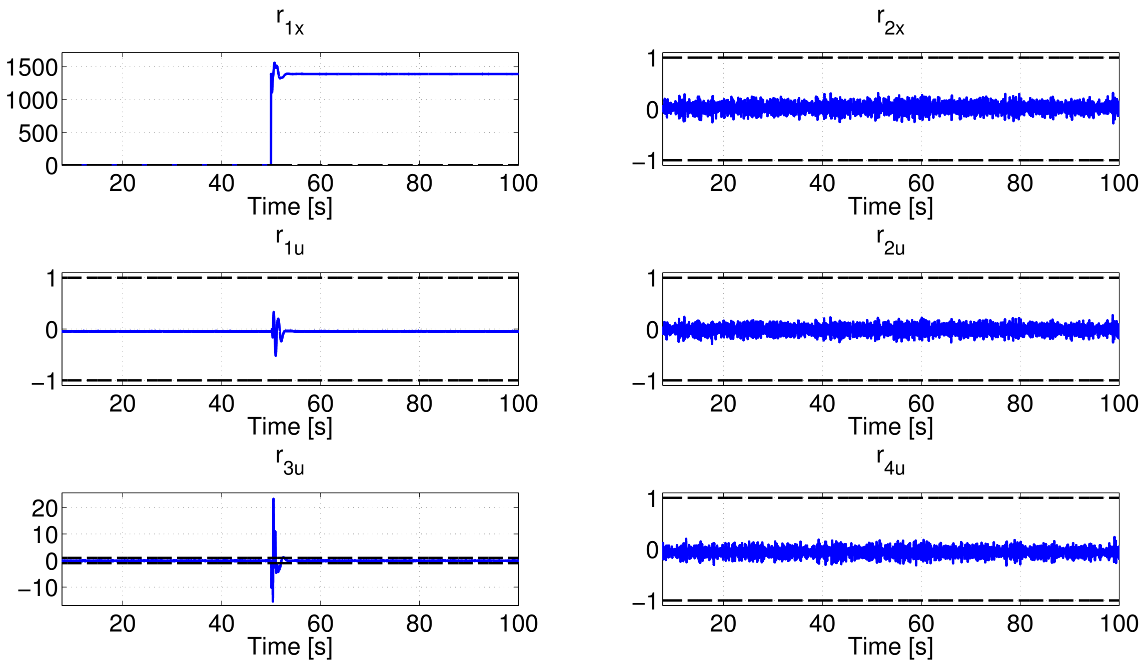

Figure 3.7: Additive fault measure x_5 residuals normalized

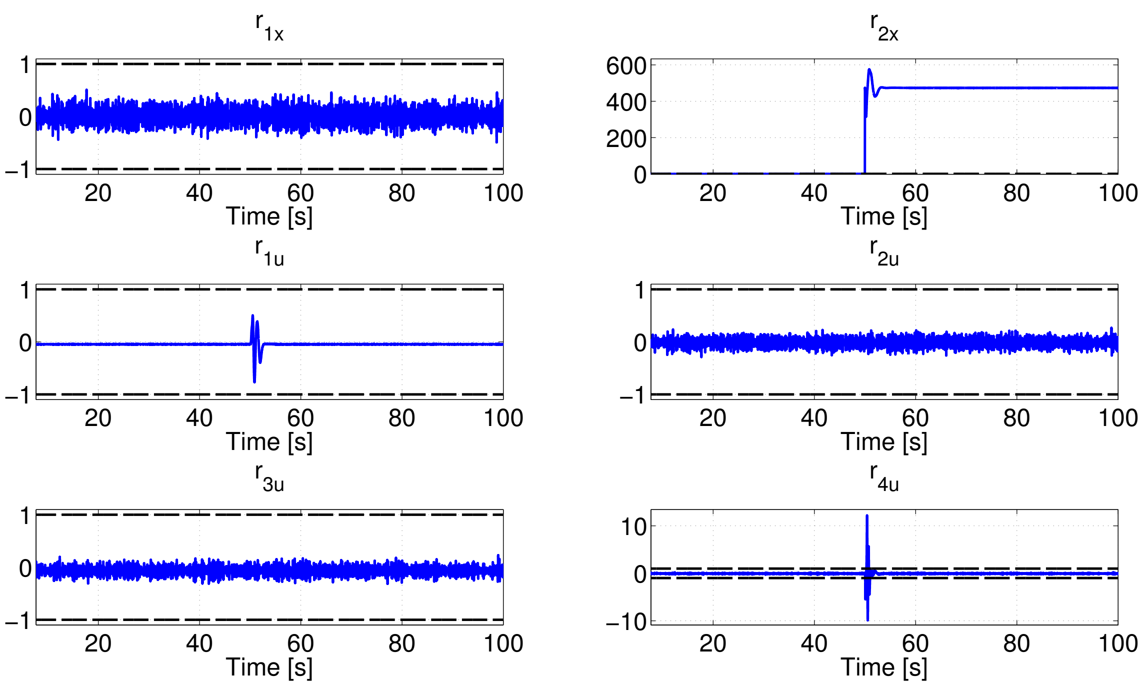

Figure 3.8: Additive fault measure x_6 residuals normalized

For control inputs faults, the fault amplitude is equal to 20 % of the maximal value of the nominal trajectory. See Table 3.2. Faults affecting control input u_1 could be detected and isolated, Fig. 3.9. However faults in the next three control inputs are hidden by the noise. Such faults becomes detectable and isolable if the amplitude is augmented. However even if the movement of the aircraft is completely excessive (displacements of more than 100 meters) the control inputs have, as maximum, an amplitude of 0.02. As a result an enormous fault, for instance equal to one, is completely unrealistic. Such faults are not considered.

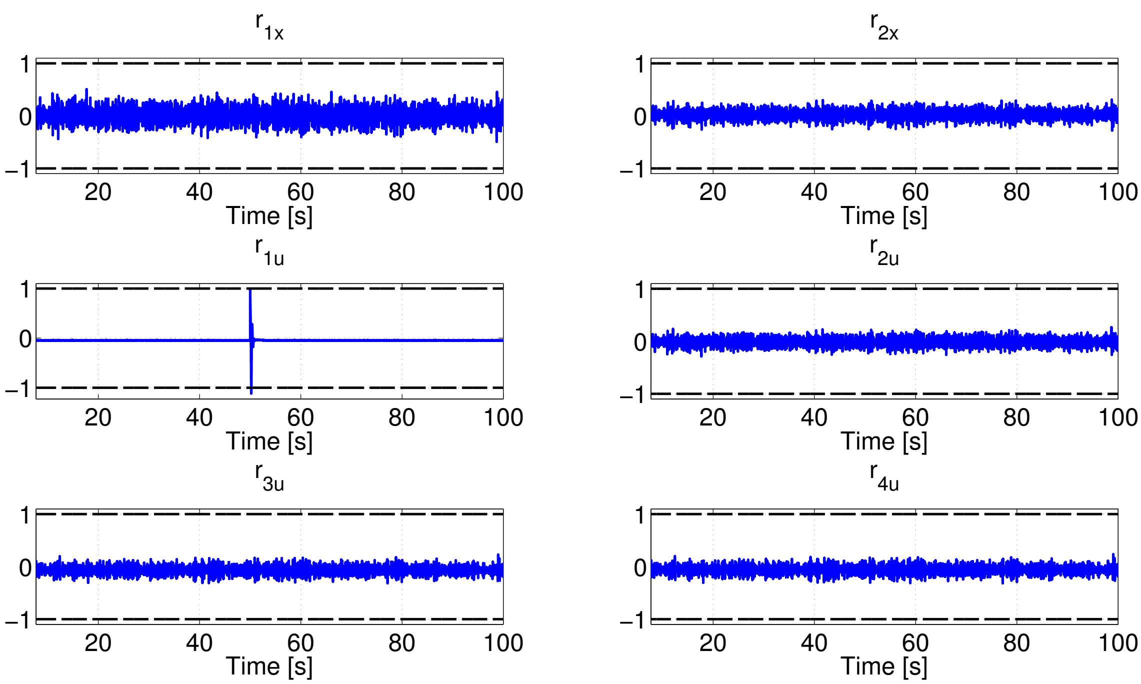

Figure 3.9: Additive fault control input u_1 residuals normalized

Control reconfiguration

For this section only faults of sensors x_{m5} and x_{m6} are considered. The number of redundant available signals is (1)*(12-4)=8. See 2.40. This number is reduced to two, because as in the FDI part, the states x_7 to x_{12} are considered fault-free at any time, so, reconfiguration is not needed.

Fault	r_{1x}	r_{2x}	r_{1u}	r_{2u}	r_{3u}	r_{4u}
F_{x1}	1	0	1	0	1	0
F_{x2}	0	1	1	0	0	1
F_{x3}	1	1	1	0	1	1
F_{x4}	0	0	0	1	0	0
F_{x5}	1	0	0[3]	0	1	0
F_{x6}	0	1	0[3]	0	0	1
F_{u1}	0	0	1	0	0	0

Table 3.3: Residuals matrix Quadrotor UAV

[3]This residue is affected but the amplitude is not enough to exceed the threshold.

The goal of the reconfiguration method is to hide the fault from the controller. This is achieved by computing a fault-free reference using the differentially flat equations 3.22. The strategy of changing the controller reference is simply to switch between the signal coming from the sensor and the signal computed with equation (3.22). Possible instabilities due to the switch effect are not considered in this work.

Figures 3.10 and 3.11 shows the comparison between FTC passive approach (—), the FTC active proposed approach (–· –) and nominal behavior (– – –) for faults affecting θ and ϕ measurement sensors. In the passive case the switch is not activated. The signal coming from sensor remains the same, and the fault is rejected by the controller. On the other hand if the amplitude fault exceeds the limits of the passive approach, the switch is triggered in order to change the signal coming from the measuring sensor according to the estimate computed with the differentially flat system equations. This action has as consequence the reconfiguration of the control loop. See Figs. 3.10 and 3.11.

The effectiveness of the proposed approach presented in the figures 3.10 and 3.11 can be compared versus figures 3.12 and 3.13. It is straightforward to see that if the control is not reconfigured the system becomes quickly unstable. In both figures yaw (ψ) is not touched. This phenomenon is explained by the physical decoupling between this angle and the pitch and roll angles.

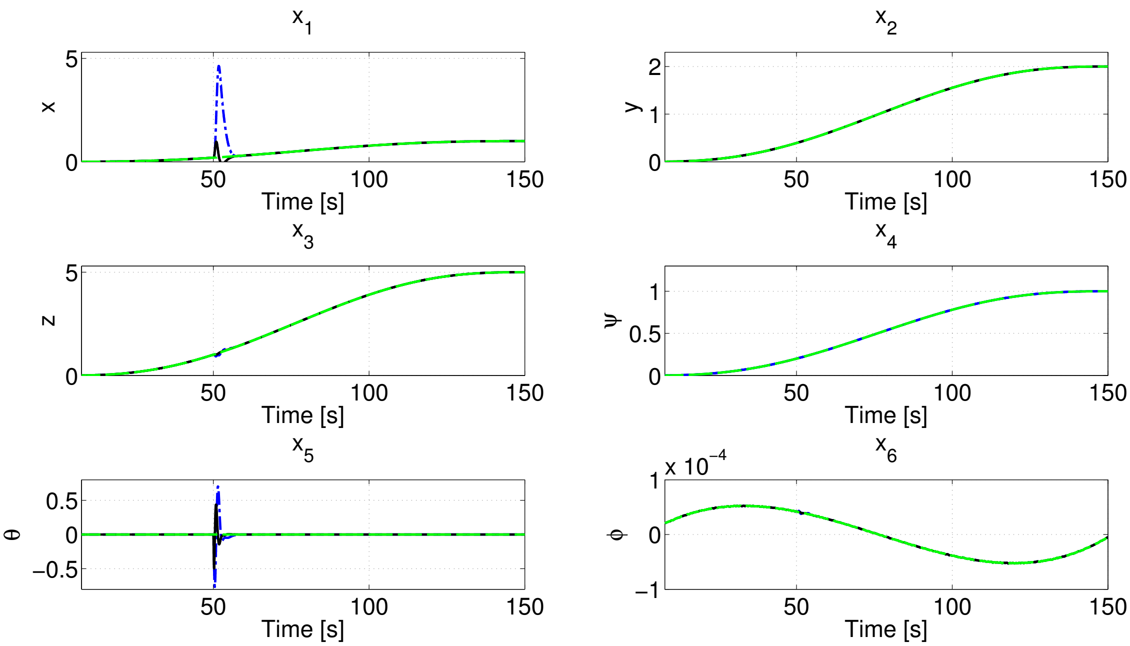

Figure 3.10: Reconfiguration after fault x_5. Passive (—). Proposed approach (–· –). Nominal (– – –).

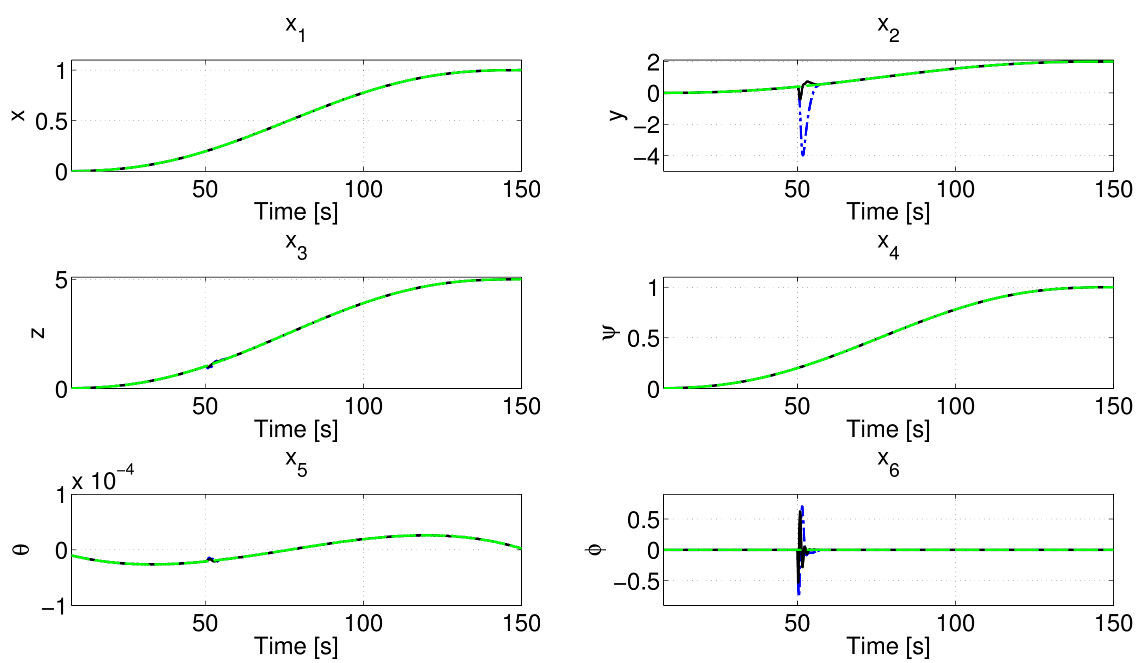

Figure 3.11: Reconfiguration after fault x_6. Passive (—). Proposed approach (–· –). Nominal (– – –).

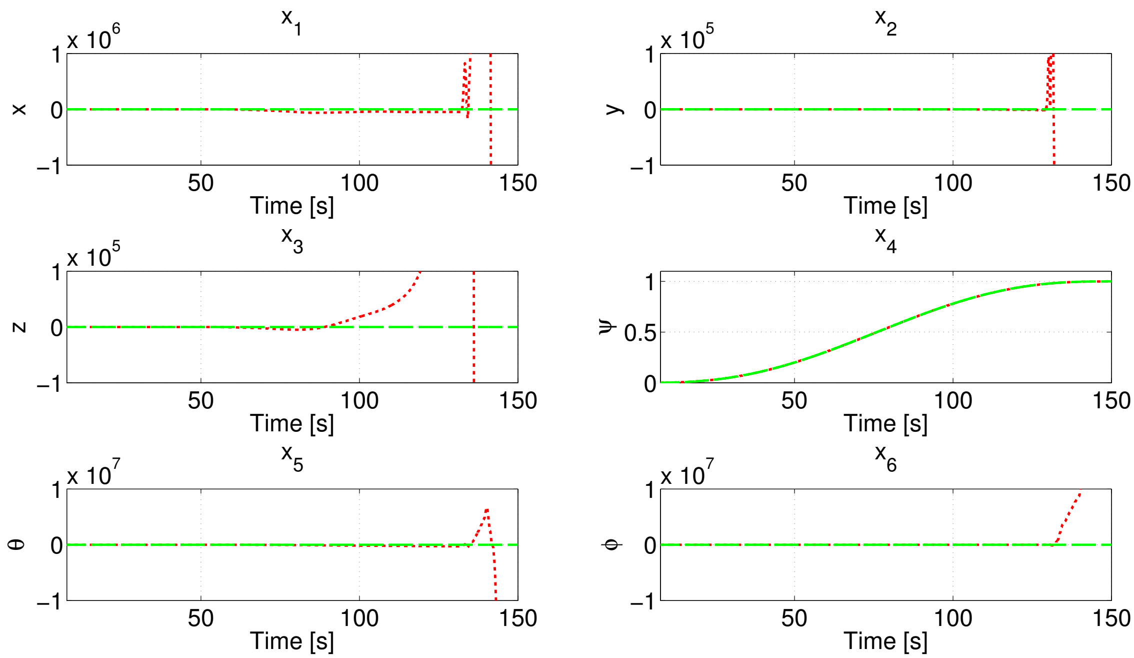

Figure 3.12: Fault affecting x_5. No-reconfiguration ($\cdots$). Nominal ($- - -$).

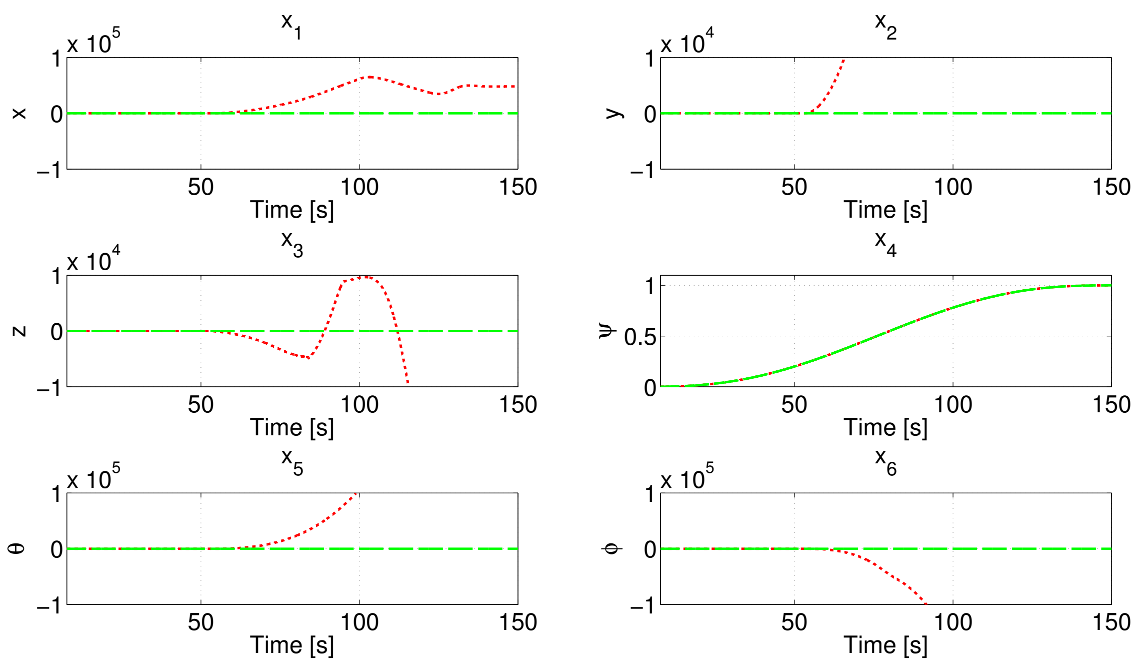

Figure 3.13: Fault x_6. No-reconfiguration ($\cdots$). Nominal ($- - -$).

3.3 THREE TANK SYSTEM

The system is composed of three tanks connected one next to the other. The three of them have the same surface section S, a central reservoir and two inflow pumps. Each tank is linked to the central reservoir by means of a pipe, in which the flow is adjustable manually. The tanks are related with pipes of section S_n. See Fig. 3.14.

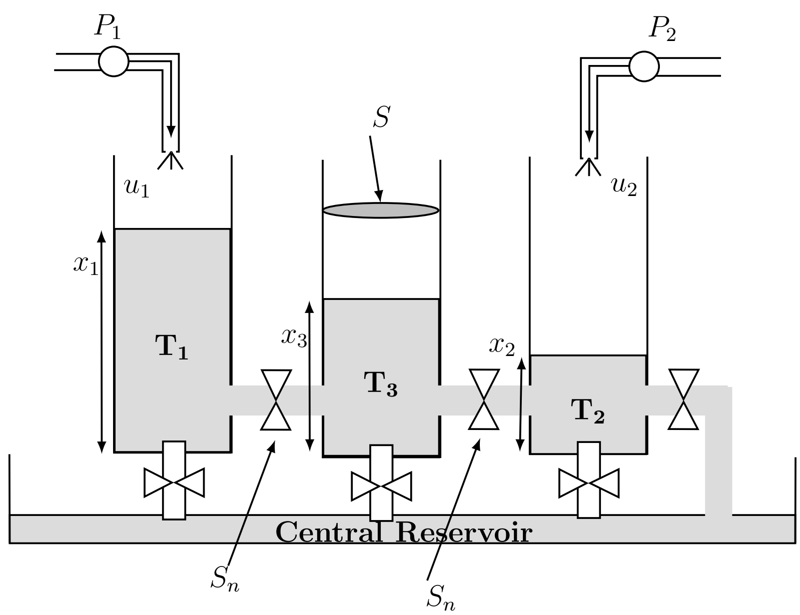

Figure 3.14: Three tank system

3.3.1 NONLINEAR MODEL

The water level inside each tank is proportional to the integral of the flows inside the pipes, by consequence we can write the next equations:

$$
\begin{aligned}
S\dot{x}_1 &= -Q_{10}(x_1) - Q_{13}(x_1, x_3) + u_1 \\
S\dot{x}_2 &= -Q_{20}(x_2) + Q_{32}(x_2, x_3) + u_2 \\
S\dot{x}_3 &= Q_{13}(x_1, x_3) - Q_{32}(x_2, x_3) - Q_{30}(x_3)
\end{aligned}
\tag{3.28}
$$

Where S is the transverse section of the tanks, x_i, $i = 1, 2, 3$ water level of each tank, Q_{i0}, $i = 1, 2, 3$ the outflow between each tank and the central reservoir, Q_{13} and Q_{32} are the outflow between tank 1 and tank 3 and the outflow between tanks 3 and 2 respectively, u_1 and u_2 are the incoming flows of each pump.

The valves connecting tanks one and three with the central reservoir are considered closed, so Q_{10} and Q_{30} are always equal to zero. The flows Q_{13}, Q_{32} and Q_{20} can be expressed as follows:

$$\begin{aligned} Q_{13}(x_1, x_3) &= a_{z1} S_n \sqrt{2g(x_1 - x_3)} \\ Q_{20}(x_2) &= a_{z2} S_n \sqrt{2g(x_2)} \\ Q_{32}(x_2, x_3) &= a_{z3} S_n \sqrt{2g(x_3 - x_2)} \end{aligned} \tag{3.29}$$

Where S_n represents the transverse section of the pipes connecting the tanks and a_{zr}, $r = 1, 2, 3$ represents the flow coefficients.

3.3.2 FLATNESS OF THE MODEL

The flat model is computed by defining x_1 and x_3 as flat outputs, $z_\alpha = [x_1 \; x_3]^T$, so the differentially flat equations can be writen as follows:

$$\begin{aligned} x_1^\alpha &= z_{\alpha 1} \\ x_2^\alpha &= z_{\alpha 2} - \frac{1}{2g}\left(\frac{a_{z1} S_n \sqrt{2g(z_{\alpha 1} - z_{\alpha 2})} - S\dot{z}_{\alpha 2}}{a_{z3} S_n}\right)^2 \\ x_3^\alpha &= z_{\alpha 2} \\ u_1^\alpha &= S\dot{z}_{\alpha 1} + a_{z1} S_n \sqrt{2g(z_{\alpha 1} - z_{\alpha 2})} \\ u_2^\alpha &= S\dot{x}_2^\alpha - a_{z3} S_n \sqrt{2g(z_{\alpha 2} - x_2^\alpha)} + a_{z2} S_n \sqrt{2g x_2^\alpha} \end{aligned} \tag{3.30}$$

$$\phi_{\alpha x}(z_{\alpha 1}, z_{\alpha 2}) = \left[\; x_1^\alpha \; x_2^\alpha \; x_3^\alpha \;\right]^T \tag{3.31}$$

$$\phi_{\alpha u}(z_{\alpha 1}, z_{\alpha 2}) = \left[\; u_1^\alpha \; u_2^\alpha \;\right]^T \tag{3.32}$$

As mentioned above the flat vector for this system, is not unique, so, it is possible to use $z_\beta = [x_2\ x_3]^T$ in order to compute another set of differentially flat equations.

$$\begin{aligned}
x_1^\beta &= z_{\beta 2} + \frac{1}{2g}\left(\frac{a_{z3}S_n\sqrt{2g(z_{\beta 2}-z_{\beta 1})} + S\dot{z_{\beta 2}}}{a_{z1}S_n}\right)^2 \\
x_2^\beta &= z_{\beta 1} \\
x_3^\beta &= z_{\beta 2} \\
u_1^\beta &= S\dot{x}_1^\beta + a_{z1}S_n\sqrt{2g(x_1^\beta - z_{\beta 2})} \\
u_2^\beta &= S\dot{z}_{\beta 1} - a_{z3}S_n\sqrt{2g(z_{\beta 2}-z_{\beta 1})} + a_{z2}S_n\sqrt{2gz_{\beta 1}}
\end{aligned} \tag{3.33}$$

$$\phi_{\beta x}(z_{\beta 1}, z_{\beta 2}) = \left[\ x_1^\beta\ x_2^\beta\ x_3^\beta\ \right]^T \tag{3.34}$$

$$\phi_{\beta u}(z_{\beta 1}, z_{\beta 2}) = \left[\ u_1^\beta\ u_2^\beta\ \right]^T \tag{3.35}$$

3.3.3 FLATNESS-BASED FAULT TOLERANT CONTROL OF A THREE TANK SYSTEM

In this scenario additive and multiplicative faults are considered. Such faults can affect sensors and actuators. Faults affecting the actuators are considered rejected by the controller. So, in this case reconfiguration is not needed.

For additive faults, a $+8cm$ fault is considered for sensors and for flow actuators an extra flow of $0.8 * 10^{-5}m^3/s$ is added. Concerning multiplicative faults a $20\,\%$ failure is considered for sensors and actuators. Only one single fault is considered at any time. Once the fault appears (at 250 s) it is recurrent until the end of the simulation.

The detection threshold is fixed as explained in section 2.5.4. If it is exceeded the fault is considered detected. The varying parameters for this system are the flow coefficients, a_{z1} and a_{z3}. Nominal values are equal to 0.75 and 0.76 respectively. The transverse section of the tanks S and the transverse section of the connecting pipes S_n are $15.4{*}10^{-3}$ and $5 * 10^{-5}$ respectively. Both sections remain without changes during the process to fix the threshold and the simulations.

The control loop is closed with a state feedback LQR controller. The matrices Q and R are chosen in order to respect the mechanical limits of the pumps and avoid outflow peaks. Additionally, saturation functions are connected to both pumps. An integral action on level measures in tanks 1 and 2 is added in order to eliminate the steady-state error.

The nominal trajectories are computed as in Appendix A, the polynomial degree is again fifth, in order to create sufficiently differentiable curves. White noise is added to the measured outputs with a level relevant to the real process measure level. Derivatives are estimated by using a high-gain observer. See 2.5.5 coupled to a low-pass filter to reduce the amplitude of the noise and improve the derivative estimation. Once again a trade-off between time delay and noise filtering is taken into account.

Let us develop the FTC approach dividing it into two different cases.

Case A

Fault detection and isolation Now we consider the analysis of the case when one set of flat outputs is found, in this system $z_\alpha = [x_1,\ x_3]^T$. For FDI this supposition implies the three hypothesis presented in section 2.5.2. By this, three residuals can be computed as in 3.36.

$$\begin{bmatrix} r_{1x}^\alpha \\ r_{1u}^\alpha \\ r_{2u}^\alpha \end{bmatrix} = \begin{bmatrix} x_{m2} \\ u_{m1} \\ u_{m2} \end{bmatrix} - \begin{bmatrix} \phi_{\alpha x}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(a)}\right) \ [0\,1\,0]^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right) \ [1\,0]^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right) \ [0\,1]^T \end{bmatrix} \tag{3.36}$$

Let us analyze each individual fault. Examining equation (3.36) it is straightforward to see that all the right hand, is in function of the α set of flat outputs. Thus, if a fault is present in the measure x_{m1} or x_{m3}, all the residuals will be impacted. This effect will indicate the presence of a fault but prevent the isolation because both faults will have the same fault signature. The phenomenon is the same with additive and multiplicative faults. See Figs. 3.15, 3.16, 3.17 and 3.18.

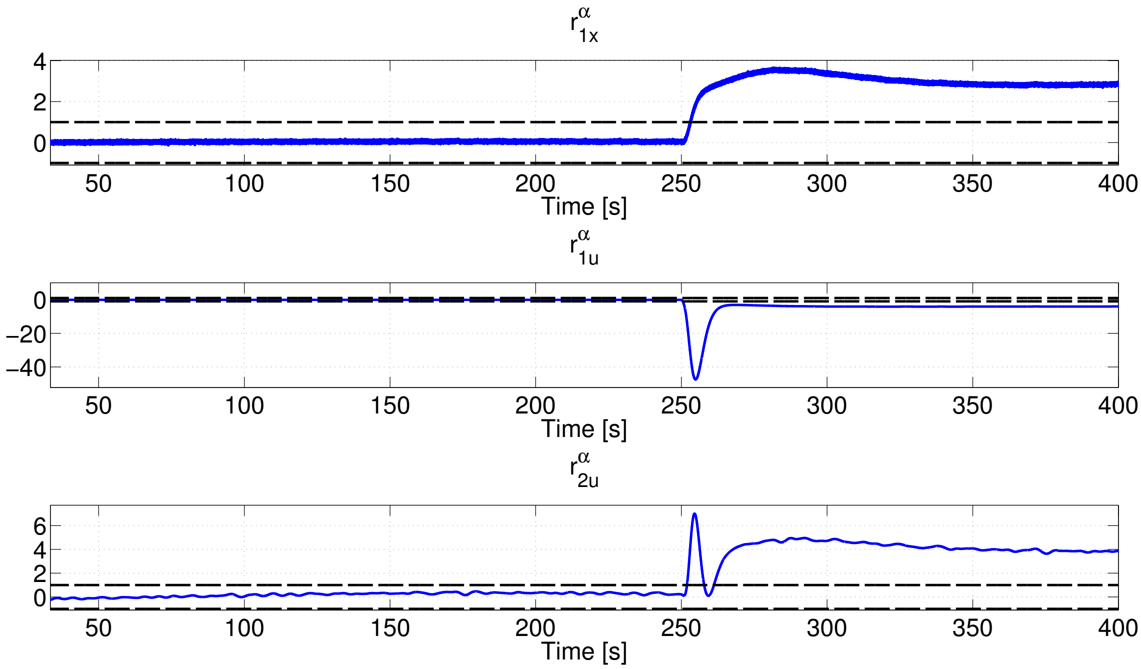

Figure 3.15: Additive fault measure x_1 normalized (z_α set)

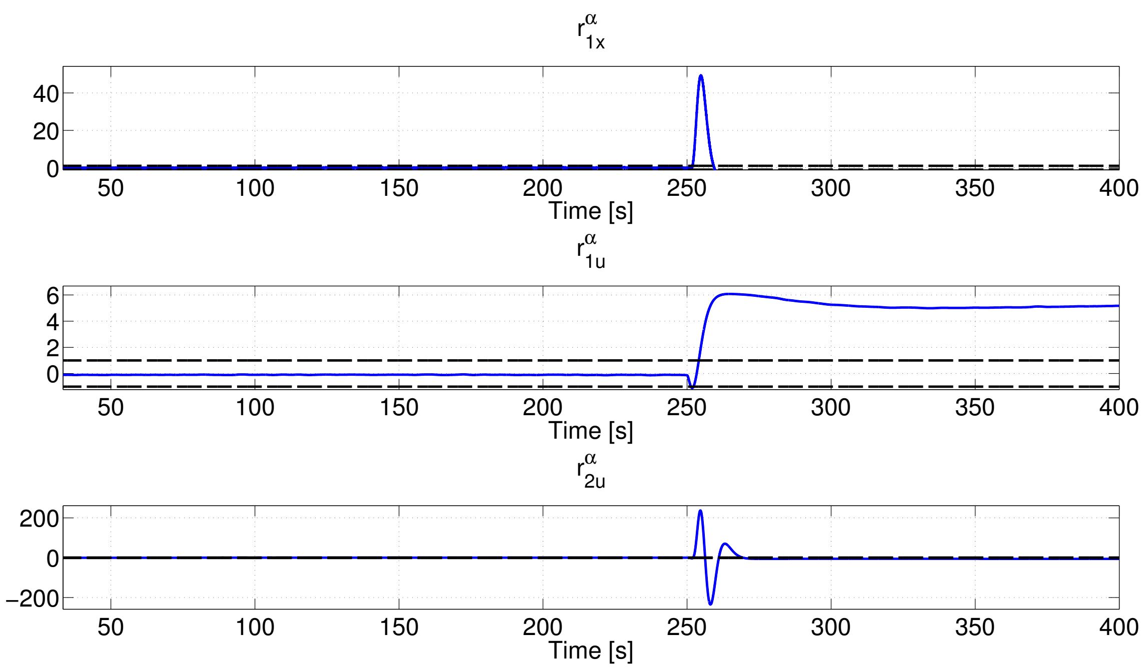

Figure 3.16: Additive fault measure x_3 normalized (z_α set)

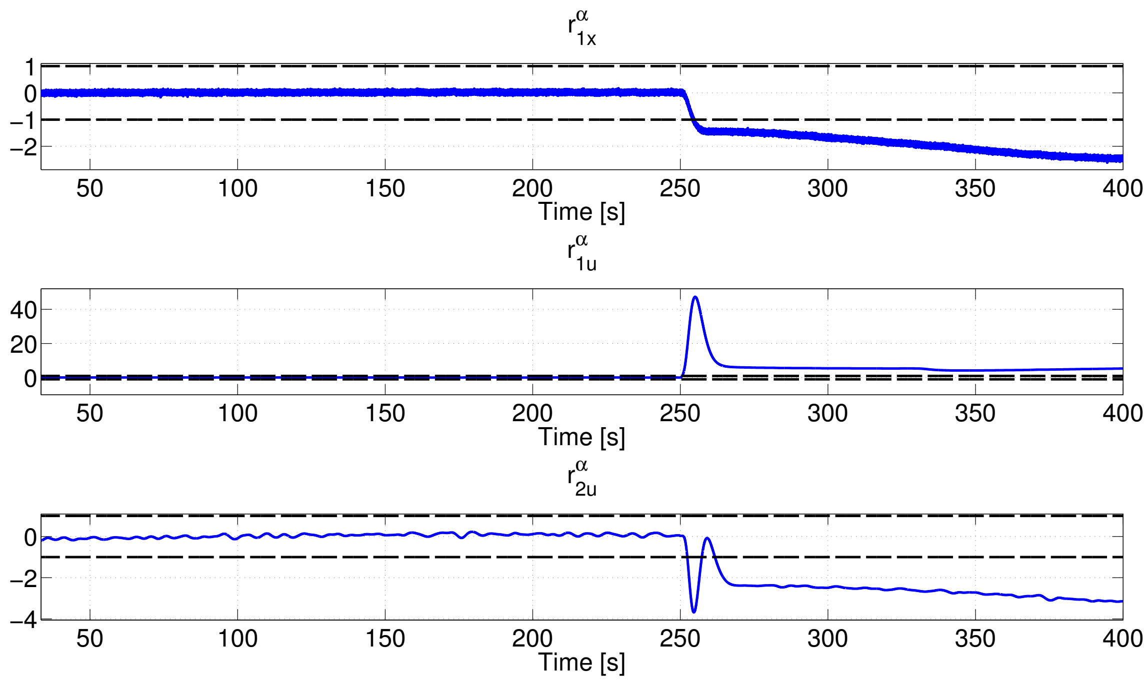

Figure 3.17: Multiplicative fault measure x_1 normalized (z_α set)

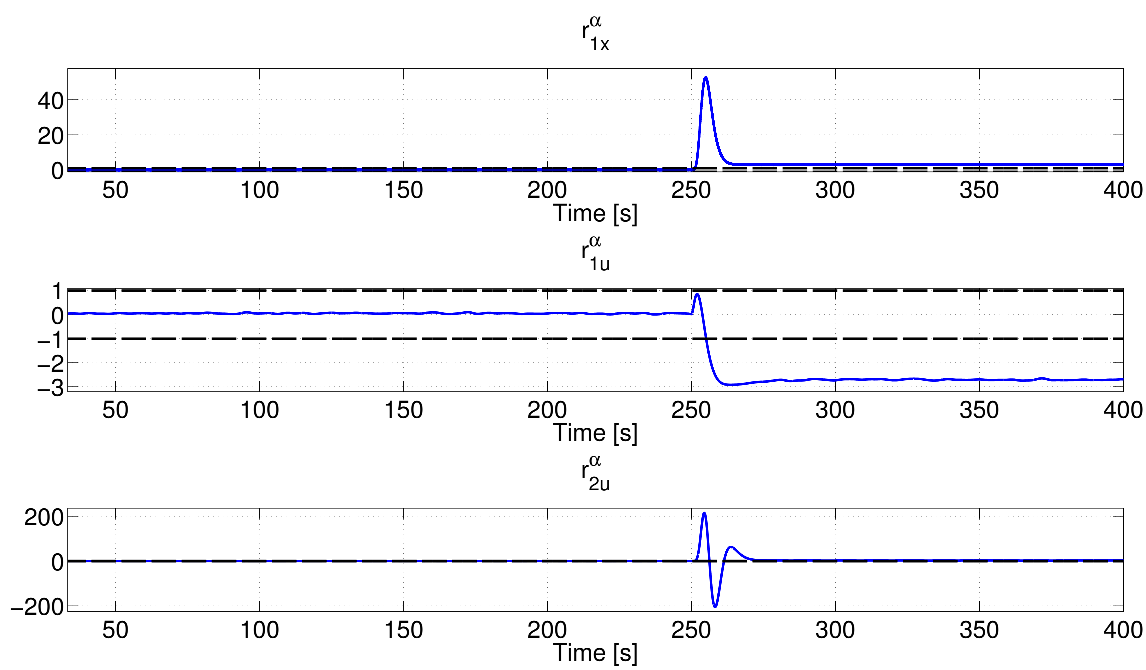

Figure 3.18: Multiplicative fault measure x_3 normalized (z_α set)

Faults affecting actuators will impact r_{1u}^{α}. If the fault is present in pump u_1 and if the fault affects pump u_2 the residual signal that will change it's amplitude will be r_{2u}^{α}. Fault affecting actuator u_1 is detected and isolated by simply comparing the amplitude of the residual signal versus the threshold amplitude, Figs.3.19 and 3.21. However this strategy is not effective for a fault affecting the actuator u_2. Even if the residual signal changes the amplitude, it is not sufficient to exceed the threshold. Figs. 3.20 and 3.22. Such small changes could be detected with another type of decision algorithm. On the other hand a sensor fault in the high measure of tank number 2 can be detected and isolated, since only r_{1x}^{α} is in function of x_{m2} and thus only this residue is affected, providing in this way a particular fault signature. Table 3.4 summarizes the results. The fault signatures are the same for additive and multiplicative faults.

Fault	r_{1x}^{α}	r_{1u}^{α}	r_{2u}^{α}
F_{x1}	1	1	1
F_{x2}	1	0[4]	1
F_{x3}	1	1	1
F_{u1}	0	1	0
F_{u2}	0	0	0[4]

Table 3.4: Residuals matrix Three tanks Case A

[4]This residue is affected but the amplitude is not enough to exceed the threshold.

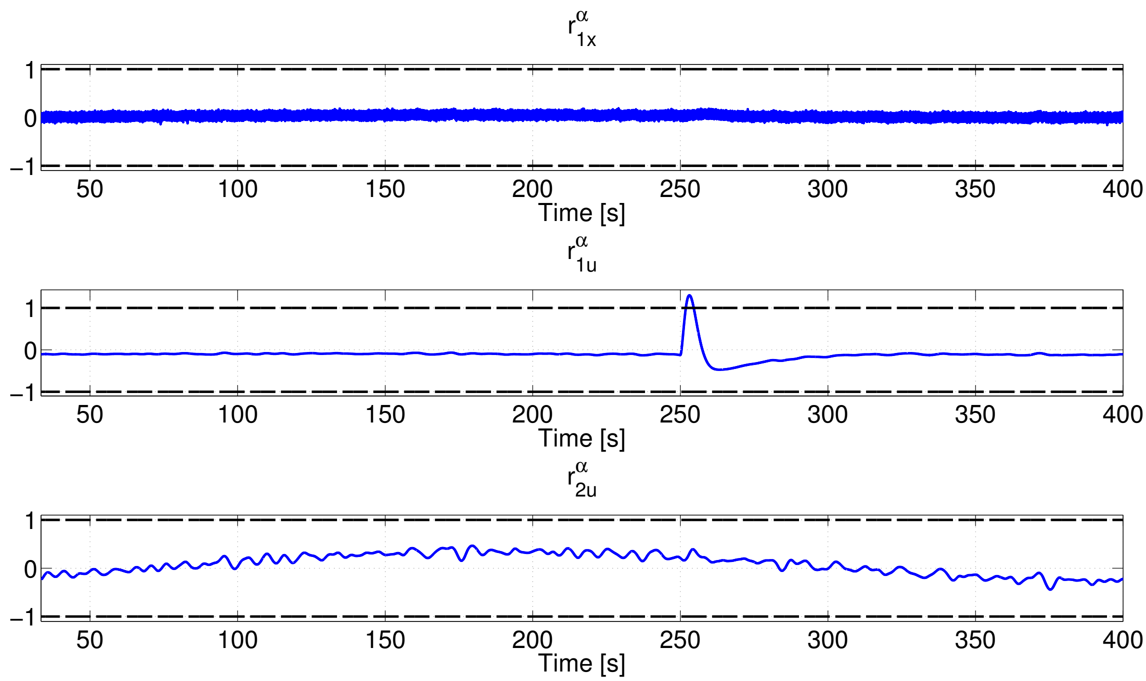

Figure 3.19: Additive fault in flow pump u_1 normalized (z_α set)

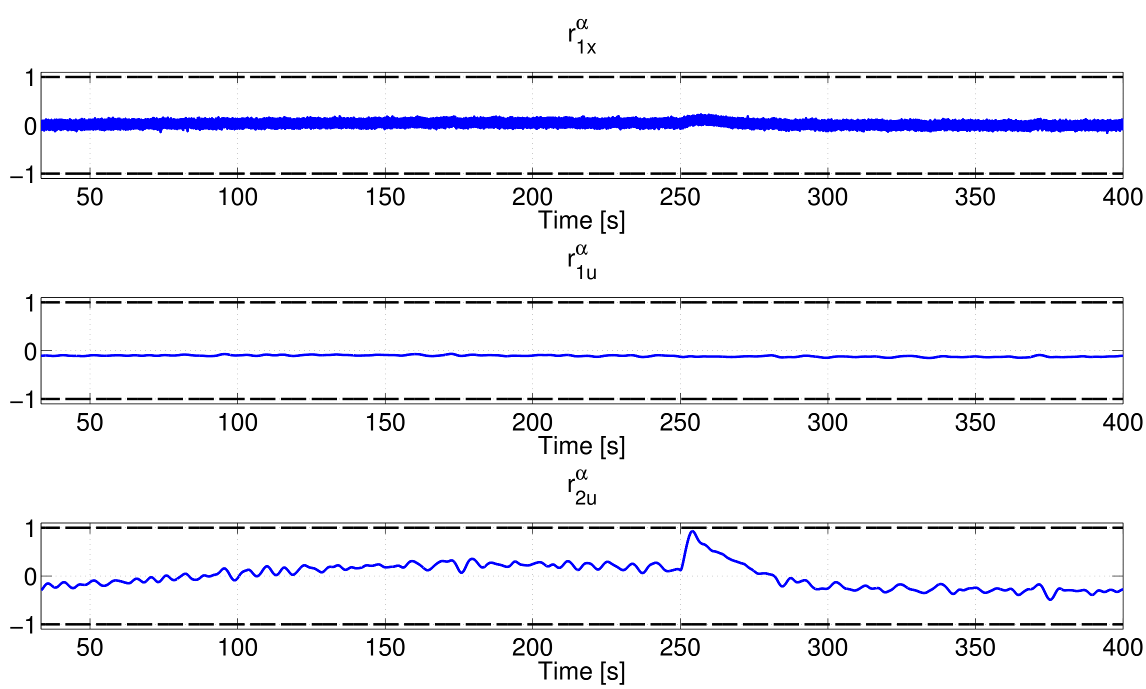

Figure 3.20: Additive fault in flow pump u_2 normalized (z_α set)

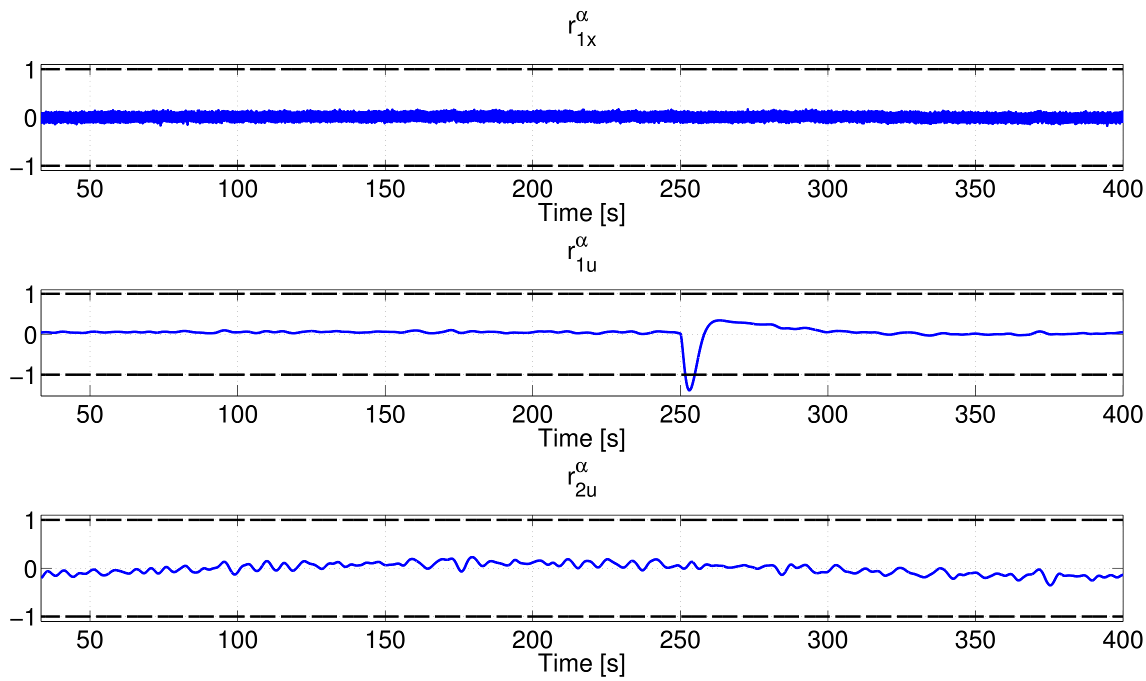

Figure 3.21: Multiplicative fault in flow pump u_1 normalized (z_α set)

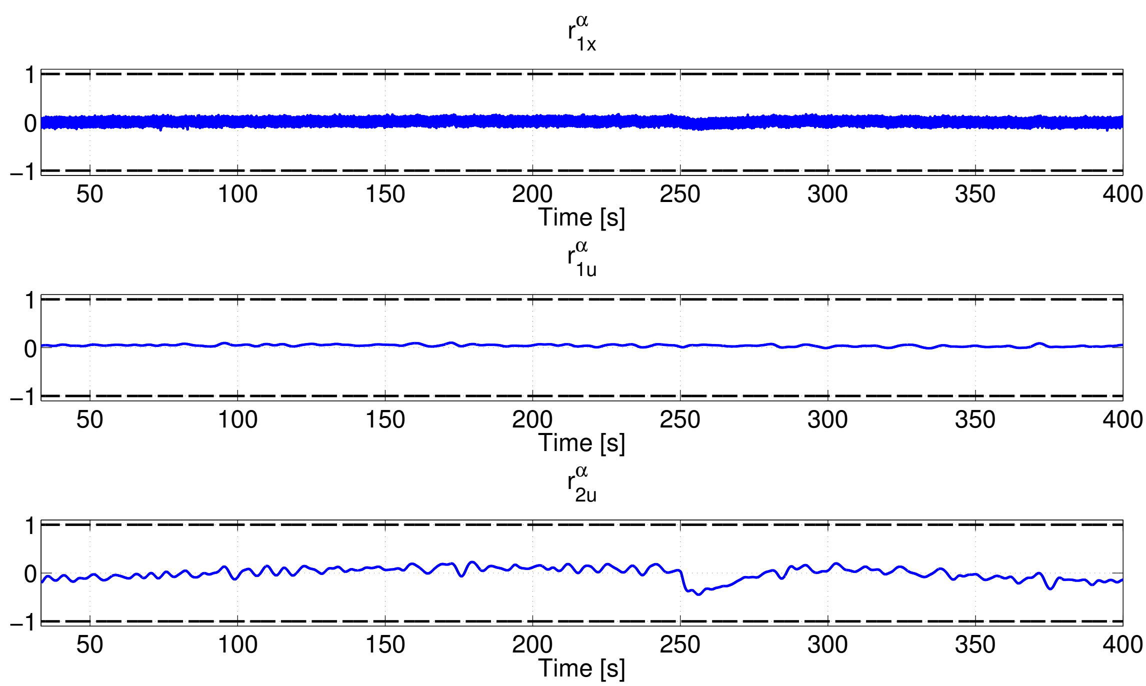

Figure 3.22: Multiplicative fault in flow pump u_2 normalized (z_α set)

Figures 3.23 and 3.24 shows the three residual signals obtained when a fault affects the high measure of tank number two. Such behavior is explained by the directly relation of the r^{α}_{1x} and the measure coming from sensor of the tank number two. Since this tank is directly related to pump number two, the controller tries to compensate the fault. Such reaction impacts the residual which depends on pump two. Residual r^{α}_{1u} is affected because pump number one tries to compensate the fault, however this is not directly related. As a result the amplitude is not enough to exceed the threshold and the fault can be detected but cannot be isolated. Such effect could be avoided by using a more sophisticated FDI decision algorithm.

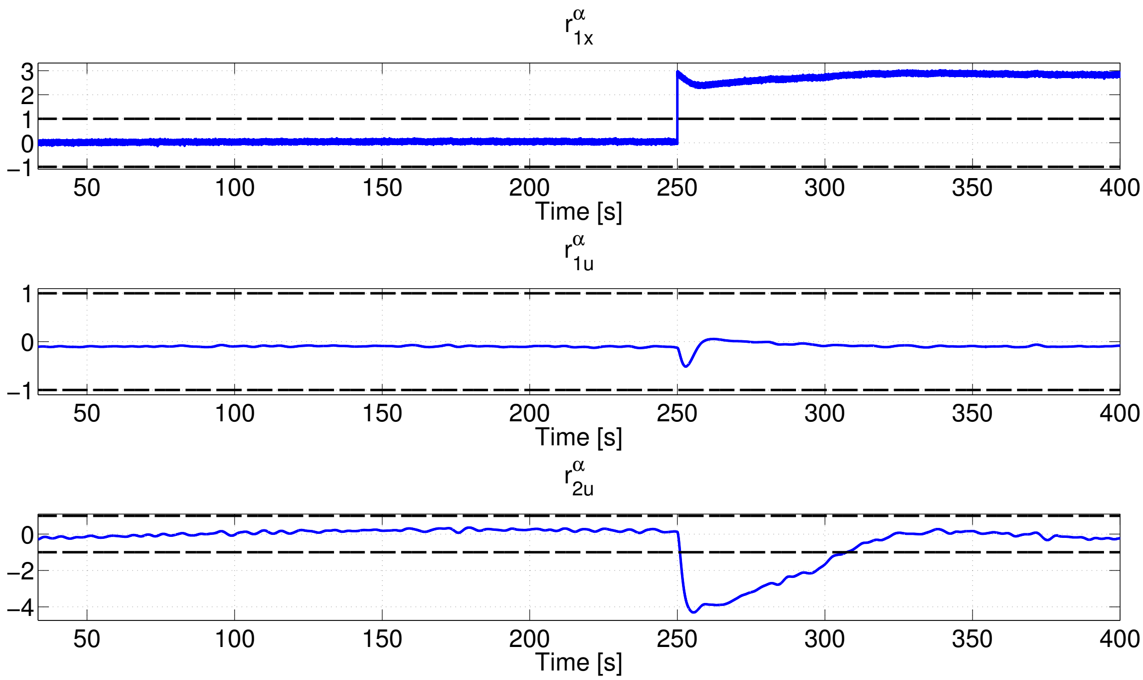

Figure 3.23: Additive fault measure x_2 normalized (z_α set)

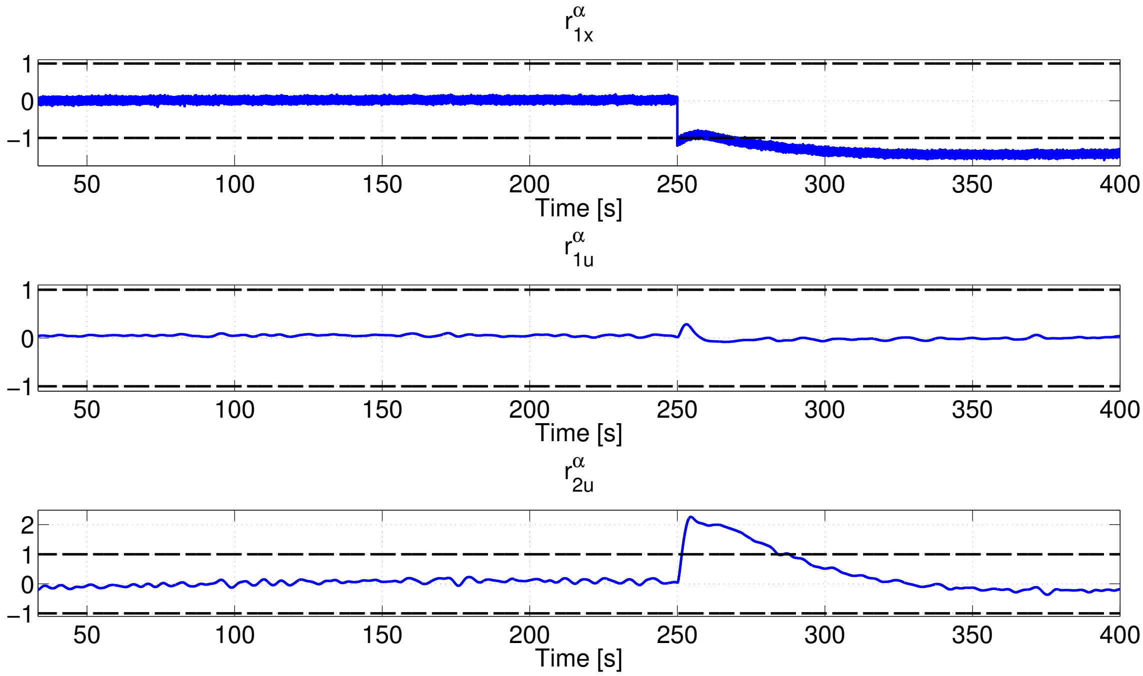

Figure 3.24: Multiplicative fault measure x_2 normalized (z_α set)

Control reconfiguration The number of redundant signals and thus the number of reconfigurable faults can be obtained using the equation (2.40). So, the number of redundant signals available is $(1) * (3 - 2) = 1$. The only signal that could be estimated is the one representing the high measure of tank number 2. The estimated signal is obtained using the expression x_2^α in the equation (3.30). That expression only depends on the z_α vector the estimated signal $\hat{x_2}$ is fault-free, hence $\hat{x_2}$ substitutes the faulty signal x_{m2} in the state feedback. Figures 3.25 and 3.26 presents the comparison of final positions with and without reconfiguration. It is clearly to see that if the signal is not reconfigured the system does not reach the desired final value.

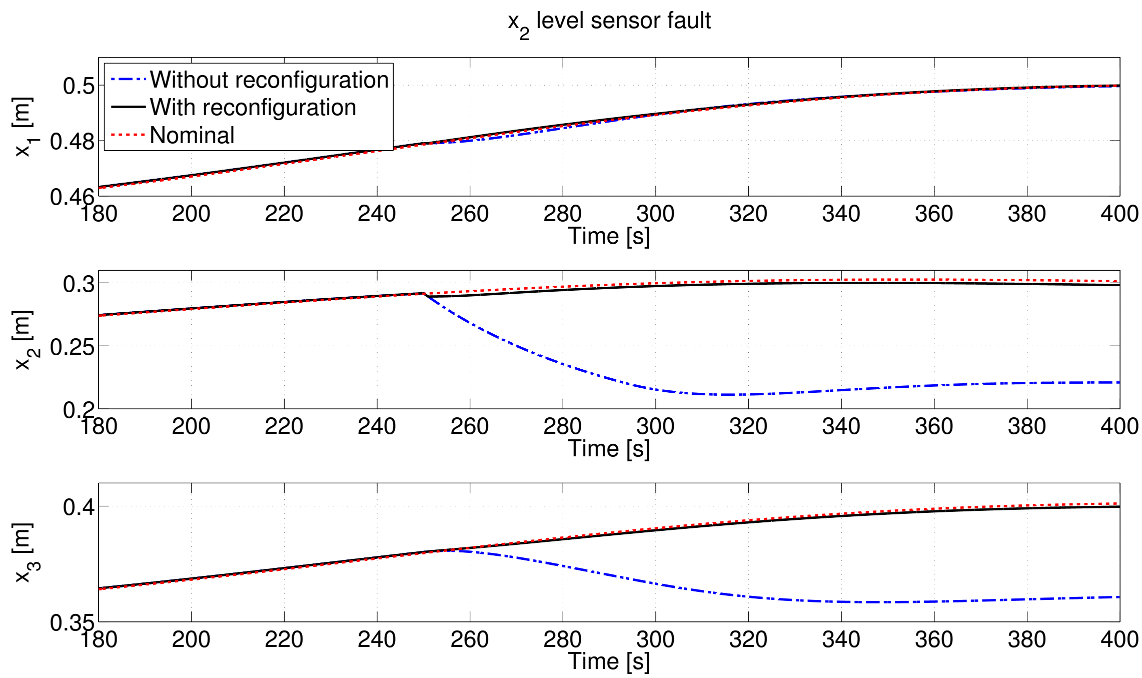

Figure 3.25: Reconfiguration after additive fault in x_2

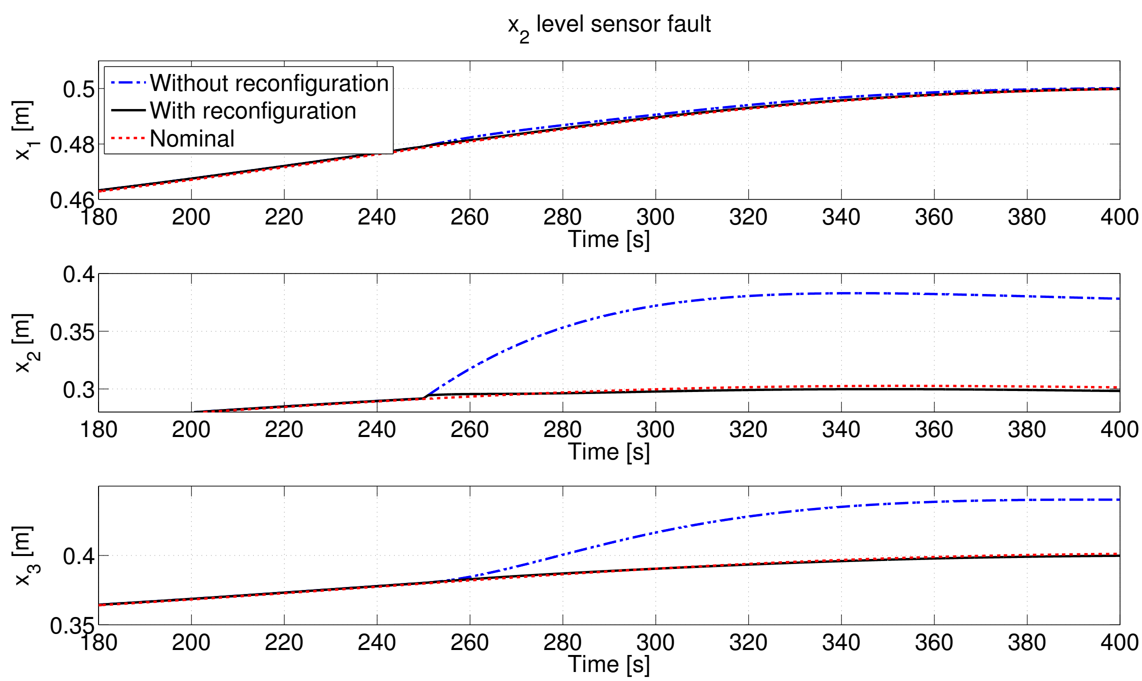

Figure 3.26: Reconfiguration after multiplicative fault in x_2

Case B

Fault detection and isolation Using the two available sets of flat outputs z_α and z_β, the number of residuals is duplicated. For this system the number of residuals is increased to 6 ($n+n$). The residual signals are computed as in the previous case and are presented in equation (3.37).

$$\begin{bmatrix} r_{1x}^{\alpha} \\ r_{1u}^{\alpha} \\ r_{2u}^{\alpha} \\ r_{1x}^{\beta} \\ r_{1u}^{\beta} \\ r_{2u}^{\beta} \end{bmatrix} = \begin{bmatrix} x_{m2} \\ u_{m1} \\ u_{m2} \\ x_{m1} \\ u_{m1} \\ u_{m2} \end{bmatrix} - \begin{bmatrix} \phi_{\alpha x}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(a)}\right) & [0\,1\,0]^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right) & [1\,0]^T \\ \phi_{\alpha u}\left(z_\alpha, \dot{z}_\alpha, ..., z_\alpha^{(b)}\right) & [0\,1]^T \\ \phi_{\beta x}\left(z_\beta, \dot{z}_\beta, ..., z_\beta^{(a)}\right) & [1\,0\,0]^T \\ \phi_{\beta u}\left(z_\beta, \dot{z}_\beta, ..., z_\beta^{(b)}\right) & [1\,0]^T \\ \phi_{\beta u}\left(z_\beta, \dot{z}_\beta, ..., z_\beta^{(b)}\right) & [0\,1]^T \end{bmatrix} \tag{3.37}$$

Increasing the number of residual signals helps to improve the FDI stage. Let us present the framework of each fault. Actuator fault u_1 is as in case A detected and isolated by simply comparing the residuals amplitude versus the threshold. See Figs. 3.27 and 3.29. Once again u_2 fault presents one residue which depends on u_{m2}, but this one does not exceed the threshold. However thanks to the second vector z_β, an additional residue r_{2u}^{β} is present. Such vector exceeds the threshold. This behavior provides an individual fault signature for the u_2 fault. See Table 3.5 and Figs. 3.28 and 3.30.

Fault	r_{1x}^{α}	r_{1u}^{α}	r_{2u}^{α}	r_{1x}^{β}	r_{1u}^{β}	r_{2u}^{β}
F_{x1}	1	1	1	1	1	0[5]
F_{x2}	1	0[5]	1	1	1	1
F_{x3}	1	1	1	1	1	1
F_{u1}	0	1	0	0	1	0
F_{u2}	0	0	0[5]	0	0	1

Table 3.5: Residuals matrix Three tank Case B

[5]This residue is affected but the amplitude is not enough to exceed the threshold.

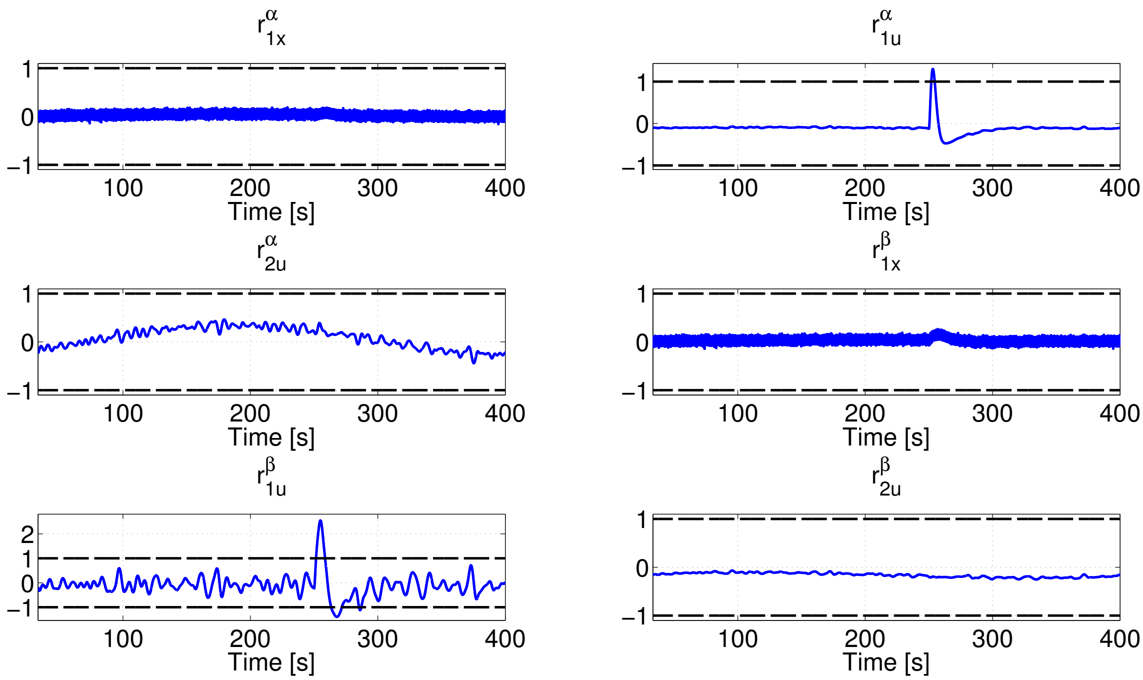

Figure 3.27: Additive fault in flow pump u_1 normalized (z_α and z_β set)

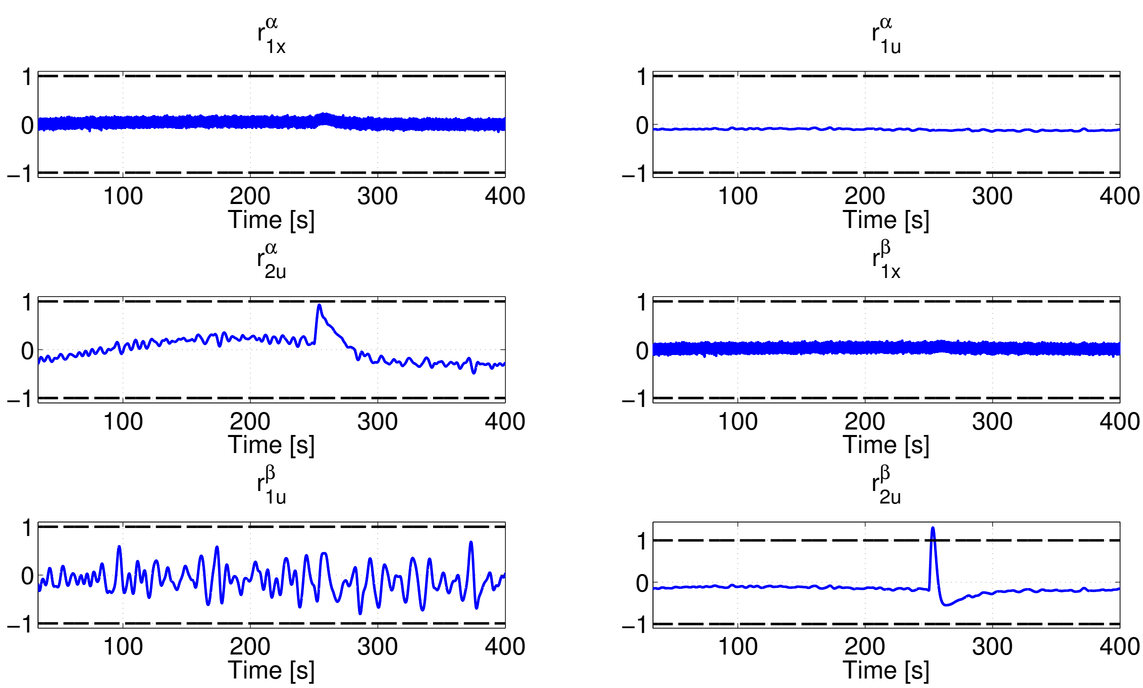

Figure 3.28: Additive fault in flow pump u_2 normalized (z_α and z_β set)

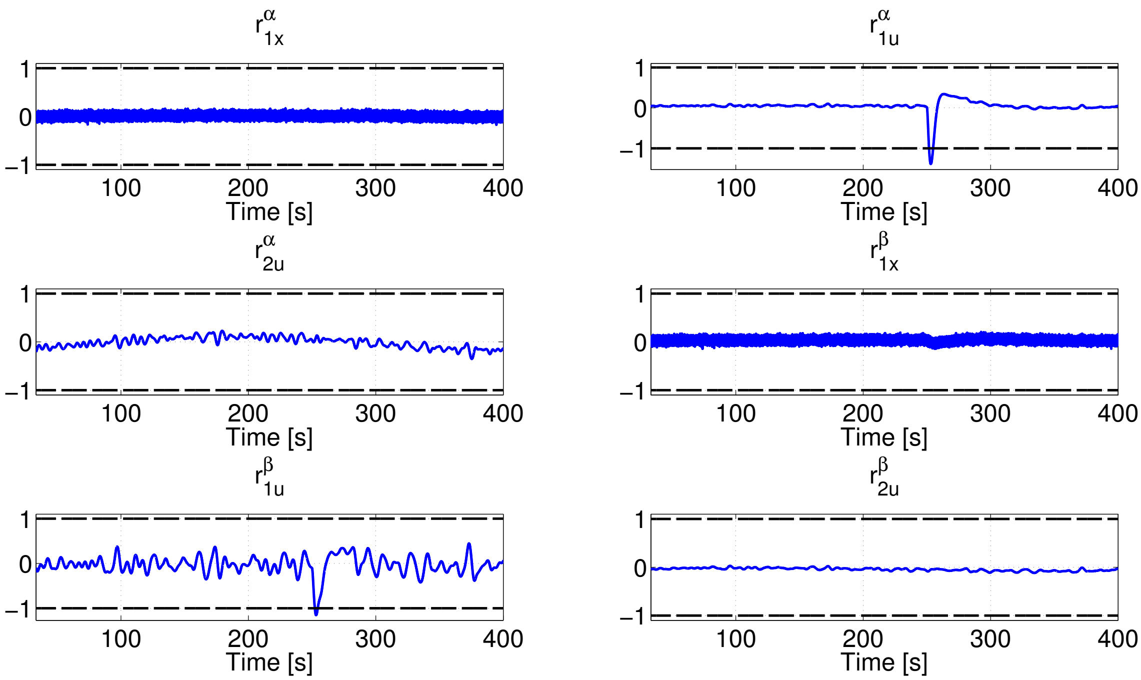

Figure 3.29: Multiplicative fault in flow pump u_1 normalized (z_α and z_β set)

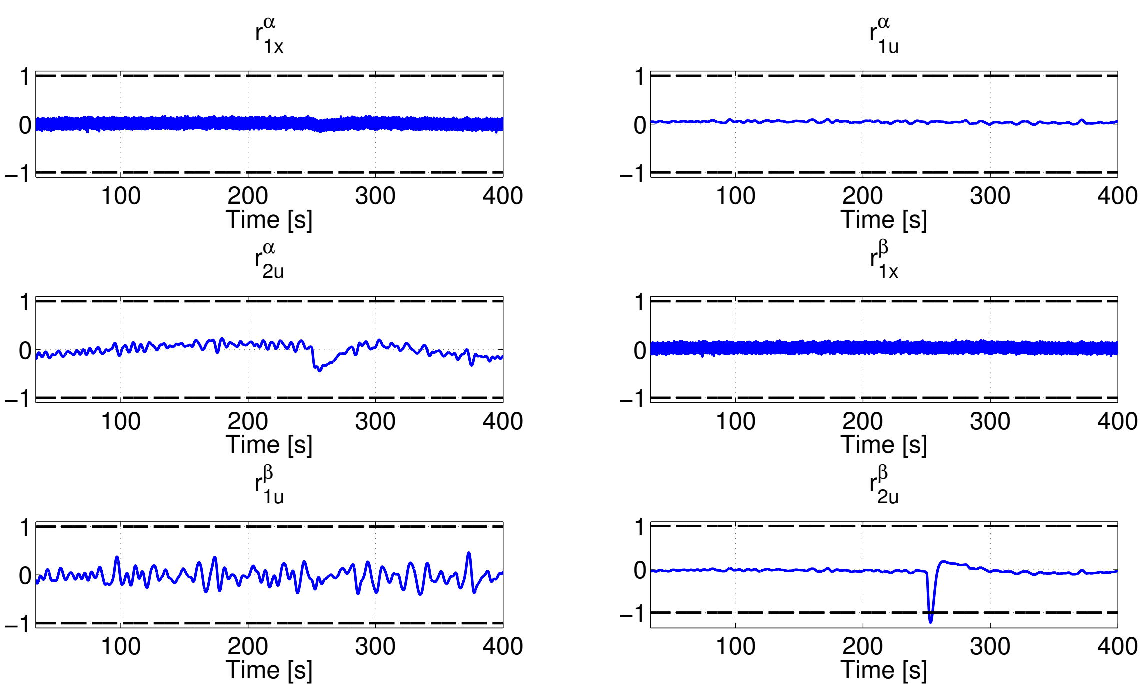

Figure 3.30: Multiplicative fault in flow pump u_2 normalized (z_α and z_β set)

For sensor faults, for instance, a fault affecting the high measure of tank one will affect the three residuals in the upper part of the right side in the equation (3.37) because for this residual signal x_1 is a flat output. Besides the residual signal r_{1x}^{β} depends on the measure x_{m1}. Thus it will be affected as well. See Figs. 3.31 and 3.32. Residual signal r_{2u}^{β} is affected because pump number two reacts to the fault as a reflection of the actuator to counteract the fault. However the amplitude is not enough to exceed the threshold.

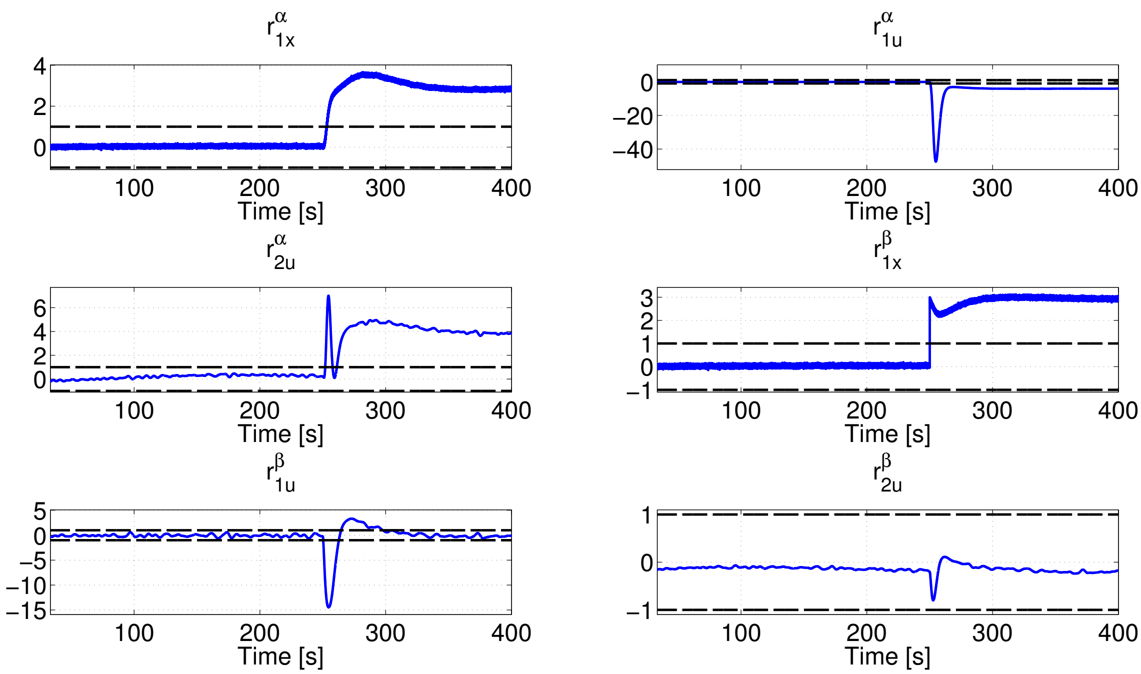

Figure 3.31: Additive fault measure x_1 normalized (z_α and z_β set)

If the fault affects the high measure of tank number two x_{m2} the affected residuals depend on the z_β vector, by consequence the residuals r_{1x}^{β}, r_{1u}^{β} and r_{2u}^{β} are affected, the residual r_{1x}^{α} is affected too because the presence of x_{m2}. This time the residual signal is affected as consequence of the closed loop is r_{1u}^{α}. See Figs. 3.33 and 3.34.

Fault in x_{m3} is an special case, because the state x_3 is part of both flat output vectors. As a result the six residuals will be affected. However this is the only framework in which every residue change its behavior, so, the fault can be detected and isolated. See Figs. 3.35 and 3.36. Table 3.5 summarizes the different fault signatures.

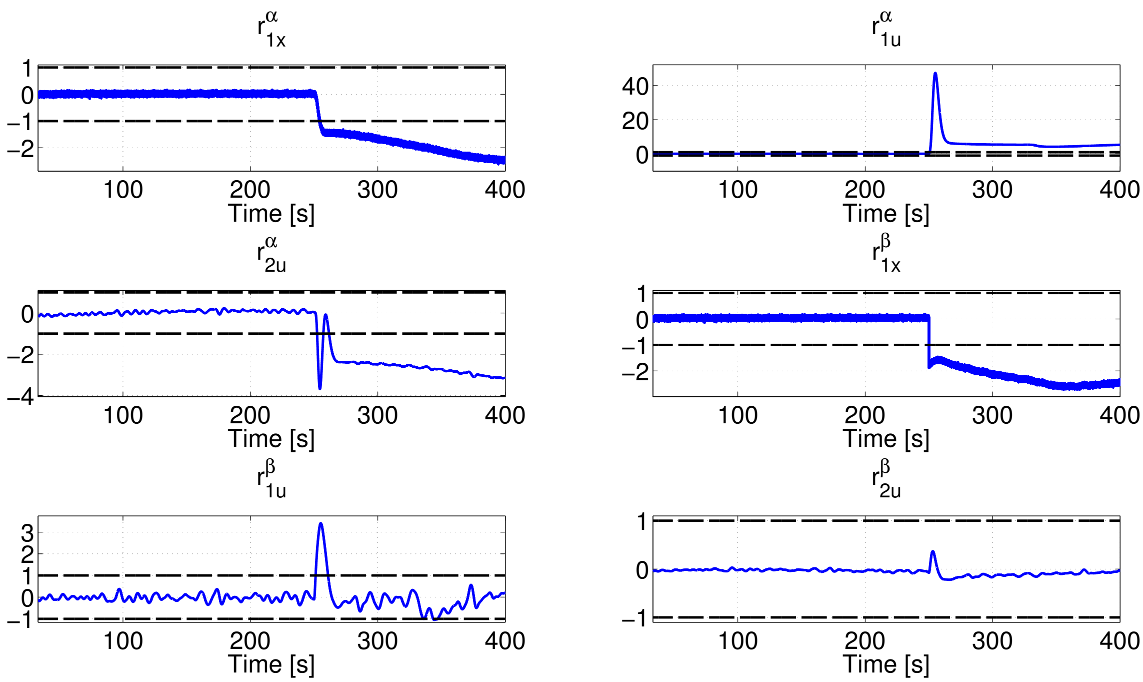

Figure 3.32: Multiplicative fault measure x_1 normalized (z_α and z_β set)

Control reconfiguration As a direct consequence of obtaining a second set of flat outputs, not only the number of residual signals is increased, but the number of redundant signals is augmented as well. This time the equation (2.40) becomes (2)*(3-2)=2, since the system has 3 states, full reconfiguration is not possible. Observing in detail the results of the FDI stage and the flat output vectors it can clearly be seen that the state x_3 triggers all the residual signals because it is part of both flat output vectors. Consequently it is impossible to compute a redundant fault-free version of it. Such effect prevents the reconfiguration after a fault on x_3. This fault is not considered.

The two redundant signals available to accomplish the FTC approach are as in case A: the state x_2, an additional set of flat outputs z_β provide a fault-free version of x_1. The reconfiguration is obtained in the same manner as in case A. Figs. 3.37 and 3.38 depict the trajectories of the outputs with and without reconfiguration. Once again, a remarkable difference exists between the trajectories with and without reconfiguration. Such results prove the efficiency of the proposed approach.

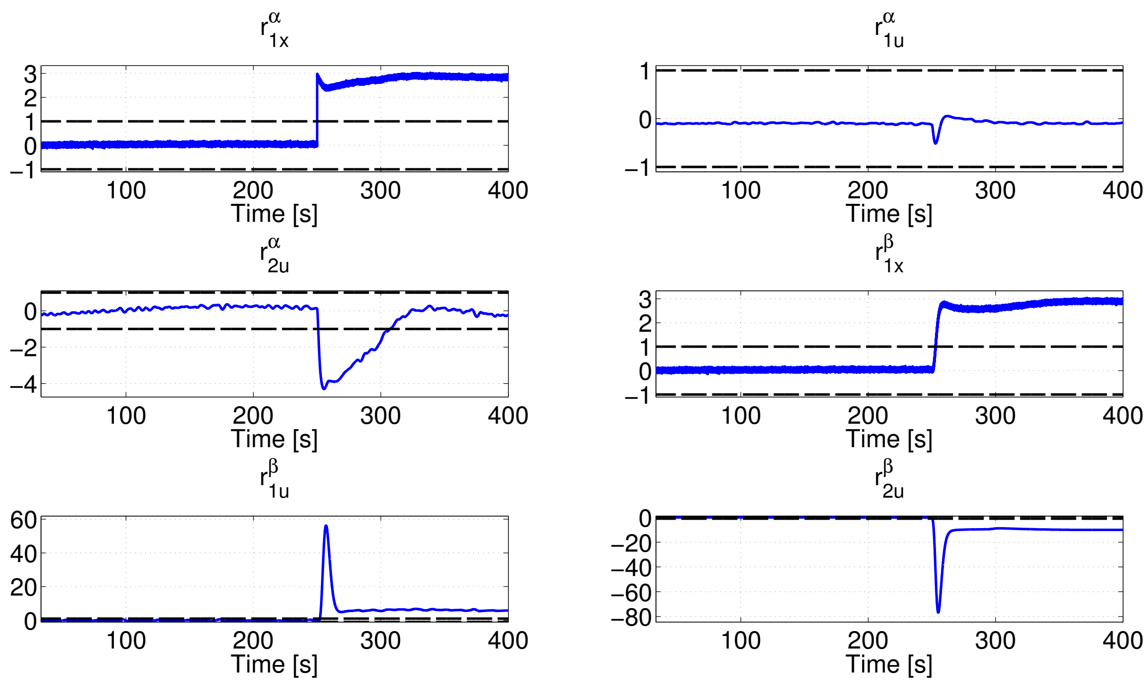

Figure 3.33: Additive fault measure x_2 normalized (z_α and z_β set)

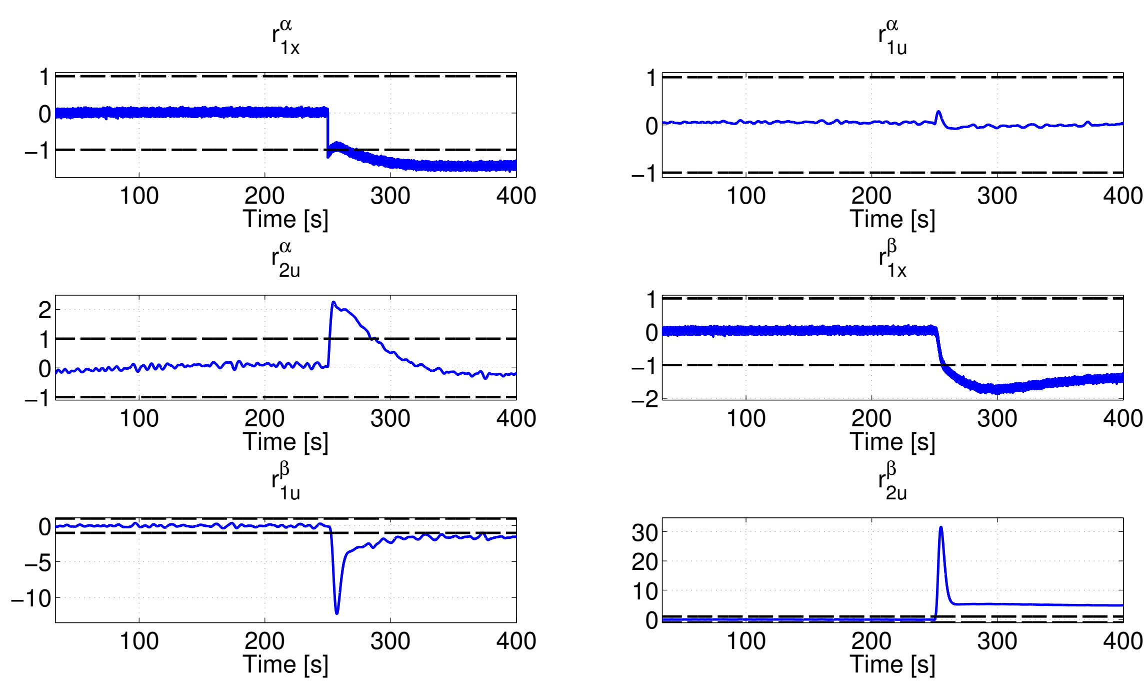

Figure 3.34: Multiplicative fault measure x_2 normalized (z_α and z_β set)

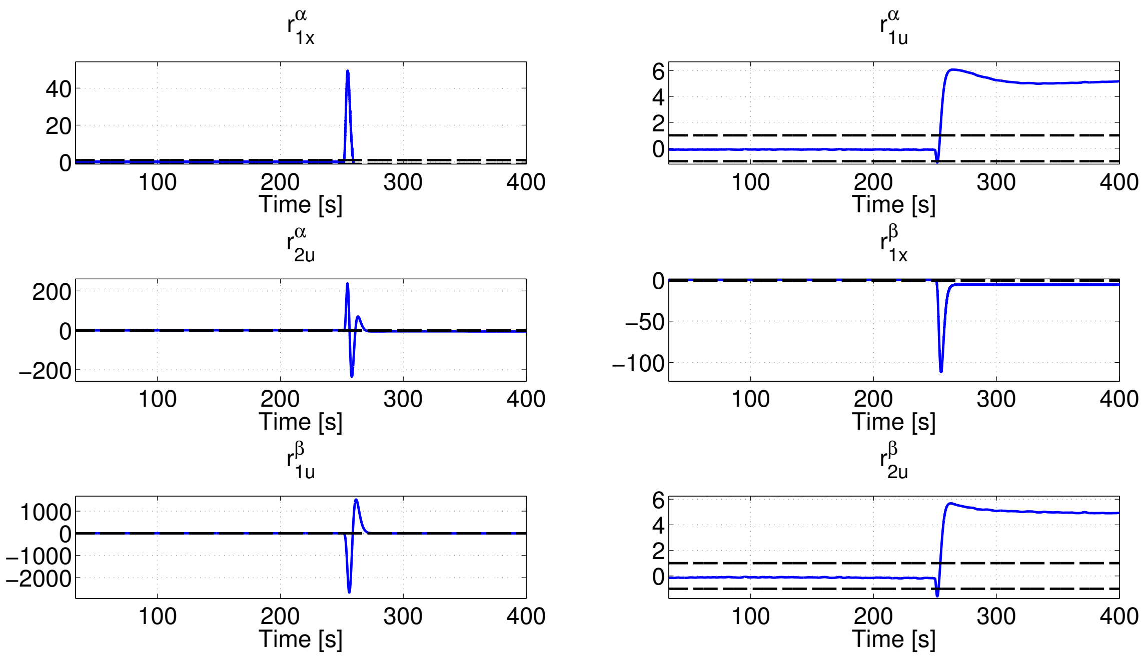

Figure 3.35: Additive fault measure x_3 normalized (z_α and z_β set)

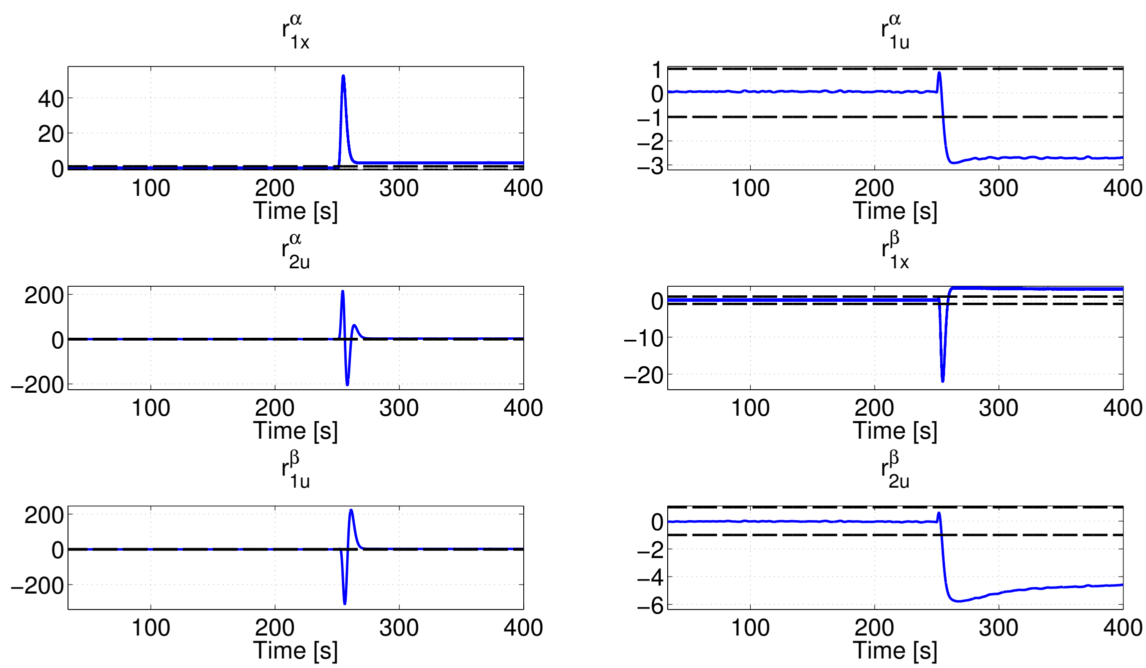

Figure 3.36: Multiplicative fault measure x_3 normalized (z_α and z_β set)

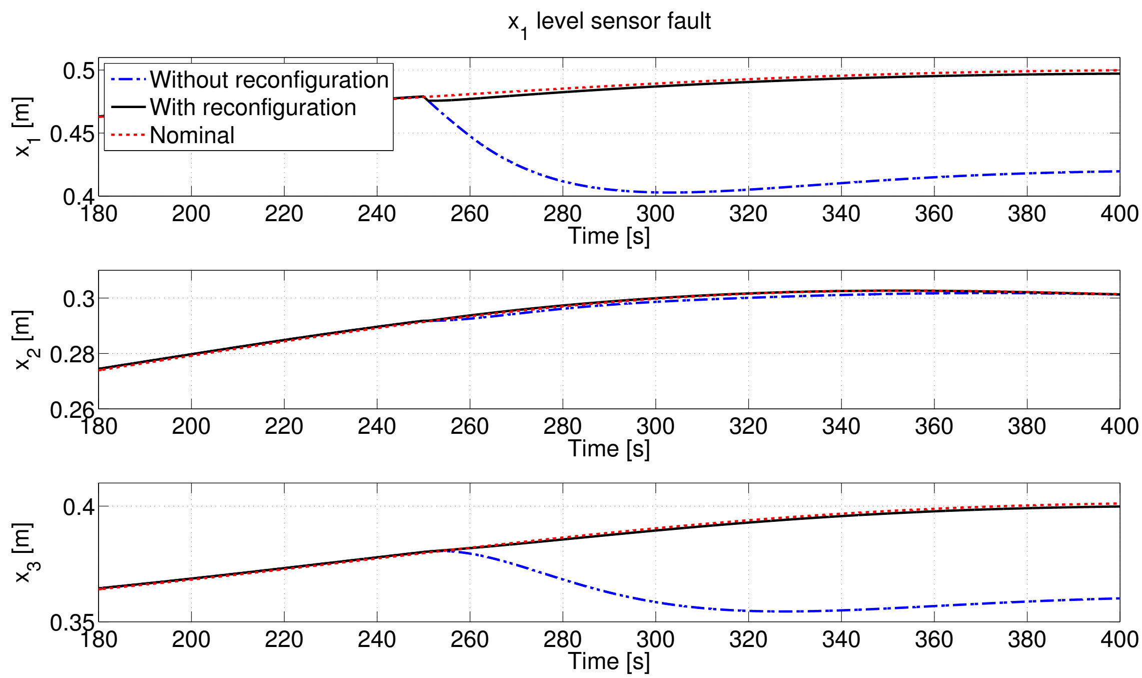

Figure 3.37: Reconfiguration after additive fault in x_1

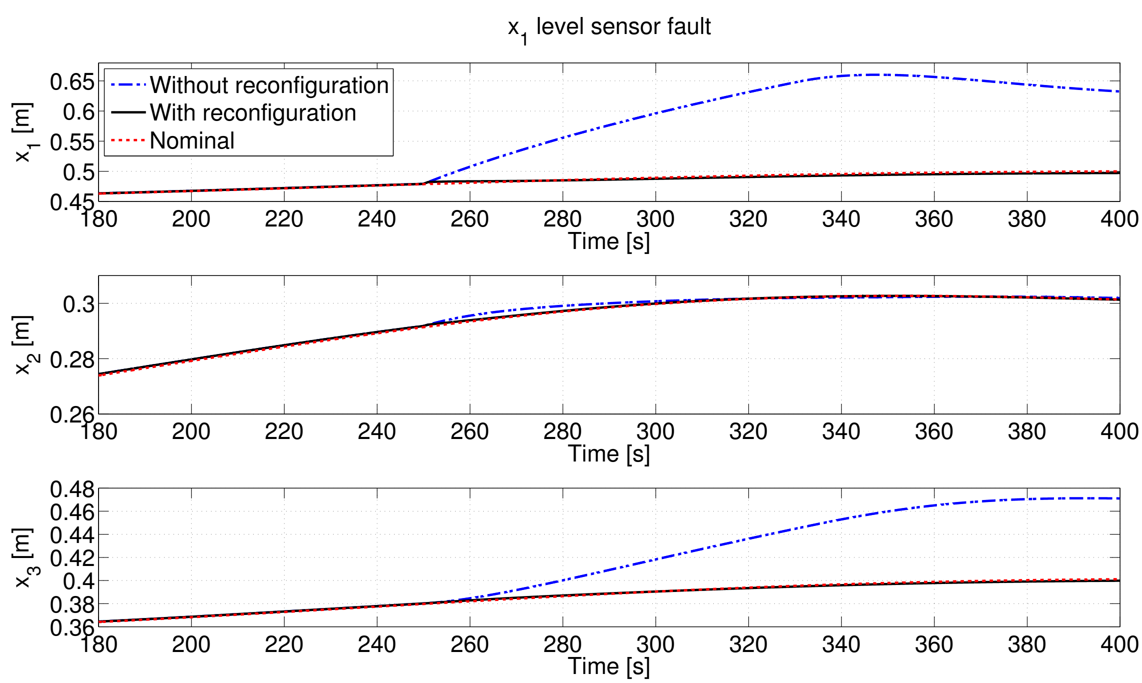

Figure 3.38: Reconfiguration after multiplicative fault in x_1

3.4 CONCLUSION

The proposed approach presented in chapter 2 is tested in this chapter. Two different systems were selected, a UAV quadrotor and a three tank system. For the quadrotor only one set of flat outputs is found, however control reconfiguration can be done partially. For the three tank process two sets fulfilling the conditions to exploit at maximum the technique are found, this has as consequence the fact that any fault may be detected and isolated. Fault detection and isolation is carried out by simply comparing the residual amplitude and a pre-defined fix threshold. For this specific system, even if in one fault case one of the residuals is affected but does not exceed the threshold, every single fault can be detected and isolated. This problem can be avoided by changing threshold-based isolation mechanism for a more adequate one.

Control reconfiguration is carried out by simply switching between the faulty and the unfaulty measure. The proposed approach uses the non-uniqueness property of the flat output vector. Unfaulty signals to reconfigure all the fault measures can only be obtained, if, at least, one element inside one of the flat output vector is differentially coupled and algebraically independent of one of the elements inside any of the other flat output vectors.

Additionally, for additive faults the fault amplitude can be estimated by simply subtracting the faulty version from the fault-free one. This information could be useful in the future, in order to plan an optimal trajectory after failure.

FINAL CONCLUSION

In this thesis, a flatness-based FTC approach is presented. This approach can be equally applied to nonlinear and linear systems. The FTC proposed approach take advantage of the non-uniqueness property of the flat output vector, in fact if there exists at least two set of flat outputs and at least one of their internal elements are algebraically independent, the FDI could be improved in a considerable manner. This operation is enhanced because if the assumption presented is verified, the number of residual signals is augmented. As a consequence the probability to obtain an individual fault signature for each fault augments too.

Real applicability is verified in two different nonlinear systems: a quadrotor UAV and a three tank system, for the first only one set of flat outputs could be found, however thanks that the internal decoupling present in this system each single sensor fault could be detected and isolated. Additionally thanks that the properties of the flat systems, fault-free references of the system states are available. Such signals are used to change the controller reference. This action hides the fault to the controller, and as a consequence the system is not affected by the fault.

The second system is a classical three tank process. Contrarily to the UAV, two set of flat outputs are found. As a consequence the number of residues is augmented and every single fault could be detected. Additionally the fault-free references could be used for reconfiguration.

The time derivatives of the flat outputs needed to obtain the residues and the unfaulty references are computed with high-gain observers coupled with a low pass filter in both systems. However the internal parameters of each block are tuned according to the system. For instance the delay introduced for the low-pass filter is not a key parameter in the three tank process, because the dynamic of this system is slow. As a consequence the filter is designed to eliminate noise and it does not take into account the resulting time delay. On the other hand the dynamics of the UAV is faster, so the design of the filter needs special attention. In fact, if the cut-off frequency is large enough, the time delay induced in the estimates will be considerable. However, if the cut-off frequency is low the time delay will decrease but the noise will increase. As a consequence a trade-off needs to be found.

Even if the technique shows to be effective to counteract the fault effect for both systems, the proposed approach has some limitations. The fact that the flat outputs has to be system states or linear combination of them could reduce the applicability. Another important point is the fact that because the technique is based on flatness it becomes necessary to compute the time derivatives of noisy signals, which could be rather difficulty when the time derivatives mount in order.

Future work Chapter 3 summarized the results presented in this dissertation. The proposed FTC technique and their applications to improve the fault detection and reconfiguration of nonlinear systems were described. Besides the remarkable features of the proposed methods, there is room for further improvements. Below, we outline some directions for possible extension of the work.

- Obtaining two sets of flat outputs could be a hard task. A future direction of this work could consist in developing an automatic algorithm to do this computations or at least present the necessary conditions, in which it exists two or more sets of flat outputs.
- The FDI decision is taken by a simply fixed algorithm, even if the technique is effective, in future work a more sophisticated decision algorithm could be tested.

- Reconfiguration is carried by changing the faulty signal for an estimated reference. This change is carried out by means of a switch. Such an action could produce instability. Future work will be developed to study this phenomenon and give solutions to avoid instability.

- Another pending issue is a real application to a nonlinear system. Additionally for the UAV most of the dynamics are neglected. This could prevent the application in a real UAV. In order to avoid this, the model presented in this manuscript could be change for a more accurate one.

- Reconfiguration shows its applicability to the UAV, however there is some limitations regarding the fault size, another interesting work could be to investigate the restructuring of the control loop.

ABBREVIATIONS AND NOTATIONS

Abbreviations

Abbreviation	Meaning
AFTCS	Active Fault Tolerant Control Systems.
ARR	Analytical Redundancy Relations.
DOF	Degrees Of Freedom.
EKF	Extended Kalman Filter.
ELS	Extended Least Squares.
FDI	Fault Detection and Isolation.
FPRG	Fundamental Problem in Residual Generation.
HOT	High Order Terms.
LQ	Linear Quadratic.
LS	Least Squares.
MPC	Model Predictive Control.
NIO	Nonlinear Identity Observer.
PID	Proportional Integral Derivative.
RLS	Recursive Least Squares.
UAV	Unmanned Aerial Vehicle.

Bibliography

[1] ADJALLAH, K., D. MAQUIN and J. RAGOT, «Nonlinear observer-based fault detection», in *Proceedings of the Third IEEE Conference on Control Applications, Glasgow, Scotland*, IEEE, pages. 1115–1120, 1994.

[2] ANTRITTER, F., B. MÜLLER and J. DEUTSCHER, «Tracking control for nonlinear flat systems by linear dynamic output feedback», *Proceedings Nonlinear control systems, Stuttgart, Germany*, 2004.

[3] BARBOT, J.-P., M. FLIESS and T. FLOQUET, «An algebraic framework for the design of nonlinear observers with unknown inputs», in *46th IEEE Conference on Decision and Control, New Orleans, Louisiana USA*, IEEE, pages. 384–389, 2007.

[4] BENOSMAN, M., *Passive Fault Tolerant Control, Robust control, Theory and applications*, ISBN: 978-953-307-229-6, InTech, DOI: 10.5772/14334. Available from: http://www.intechopen.com/books/robust-control-theory-and-applications/passive-fault-tolerant-control, 2011.

[5] BODSON, M. and J. E. GROSZKIEWICZ, «Multivariable adaptive algorithms for reconfigurable flight control», *IEEE Transactions on Control Systems Technology*, **5**(2), pages. 217–229, 1997.

[6] CAGLAYAN, A., S. ALLEN and K. WEHMULLER, «Evaluation of a second generation reconfiguration strategy for aircraft flight control systems subjected to actuator failure/surface damage», in *Proceedings of the IEEE National Aerospace and Electronics Conference*, IEEE, pages. 520–529, 1988.

[7] CASBEER, D. W., D. B. KINGSTON, R. W. BEARD and T. W. MCLAIN, «Cooperative forest fire surveillance using a team of small unmanned air vehicles», *International Journal of Systems Science*, **37**(6), pages. 351–360, 2006.

[8] CASTILLO, P., A. DZUL and R. LOZANO, «Real-time stabilization and tracking of a four-rotor mini rotorcraft», *IEEE Transactions on Control Systems Technology*, **12**(4), pages. 510–516, 2004.

[9] CAZAURANG, F., *Commande robuste des systÈmes plats application à la commande d'une machine synchrone*, PhD. thesis, Université Sciences et Technologies-Bordeaux I, 1997.

[10] CAZAURANG, F. and L. LAVIGNE, «Satellite path planning by flatness approach», *International Review of Aerospace Engineering*, **2**(3), pages. 123–132, 2009.

[11] CHAMSEDDINE, A., Y. ZHANG, C. A. RABBATH, C. JOIN and D. THEILLIOL, «Flatness-Based Trajectory Planning/Replanning for a Quadrotor Unmanned Aerial Vehicle», *IEEE Transactions on Aerospace and Electronic Systems*, **48**(4), pages. 2832–2848, 2012.

[12] CHARLET, B., J. LÉVINE and R. MARINO, «Sufficient conditions for dynamic state feedback linearization», *Journal on Control and Optimization*, **29**(1), pages. 38–57, 1991.

[13] CHEN, J. and R. J. PATTON, *Robust model-based fault diagnosis for dynamic systems*, Kluwer academic publishers, 1999.

[14] CHOW, E. and A. WILLSKY, «Analytical redundancy and the design of robust failure detection systems», *IEEE Transactions on Automatic Control*, **29**(7), pages. 603–614, 1984.

[15] CHRISTOPHE, C., *Surveillance des systèmes non linéaires: Application aux machines électriques*, PhD. thesis, Université des sciences et technologies de Lille, 2001.

[16] CLARK, R. N., D. C. FOSTH and V. M. WALTON, «Detecting instrument malfunctions in control systems», *IEEE Transactions on Aerospace and Electronic Systems*, (4), pages. 465–473, 1975.

[17] COMTET VARGA, G., *Surveillance des systèmes non linéaires- Application aux machines asynchrones*, PhD. thesis, Université des Sciences et Technologies de Lille, 1997.

[18] COWLING, I. D., O. A. YAKIMENKO, J. F. WHIDBORNE and A. K. COOKE, «A prototype of an autonomous controller for a quadrotor UAV», in *European Control Conference, Kos, Greece*, pages. 1–8, 2007.

[19] CRASSIDIS, J. L. and J. L. JUNKINS, *Optimal Estimation of Dynamic Systems*, Chapman and Hall/CRC, 2004.

[20] DAAFOUZ, J., M. FLIESS, G. MILLÉRIOUX *et al.*, «Une approche intrinsèque des observateurs linéaires à entrées inconnues», in *Conférence internationale francophone d'automatique (CIFA 2006), Bordeaux, France*, 2006.

[21] DALMASSO, I., I. GALLETTI, R. GIULIANO and F. MAZZENGA, «WiMAX networks for emergency management based on UAVs», in *IEEE First AESS European Conference on Satellite Telecommunications (ESTEL), Rome, Italy*, IEEE, pages. 1–6, 2012.

[22] DEVASIA, S. and B. PADEN, «Exact output tracking for nonlinear time-varying systems», in *Proceedings of the 33rd IEEE Conference on Decision and Control, Lake Buena Vista, Florida, USA*, volume 3, IEEE, pages. 2346–2355, 1994.

[23] DING, S. X., *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*, Springer, 2008.

[24] FINN, R. L. and D. WRIGHT, «Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications», *Computer Law & Security Review*, **28**(2), pages. 184–194, 2012.

[25] FLIESS, M., C. JOIN, M. MBOUP and H. SIRA-RAMÍREZ, «Compression différentielle de transitoires bruités», *Comptes Rendus Mathematique*, **339**(11), pages. 821–826, 2004.

[26] FLIESS, M., C. JOIN, M. MBOUP, H. SIRA-RAMIREZ *et al.*, «Analyse et représentation de signaux transitoires: application à la compression, au débruitage et à la détection de ruptures», in *Actes 20e Coll. GRETSI, Louvain-la-Neuve, France, available on http://hal.inria.fr/inria-00001115*, 2005.

[27] FLIESS, M., J. LÉVINE, P. MARTIN and P. ROUCHON, «Flatness and defect of nonlinear systems: introductory theory and examples», *International journal of control*, **61**(6), pages. 1327–1361, 1995.

[28] FRANK, P., G. SCHRIER and E. A. GARCIA, «Nonlinear observers for fault detection and isolation», in *New Directions in nonlinear observer design*, Springer, pages. 399–422, 1999.

[29] FRANK, P. M. and M. BLANKE, «Fault diagnosis and fault-tolerant control», in *Control Systems, Robotics and Automation, from Encyclopedia of Life Support Systems (EOLSS), Developed Under the Auspices of the UNESCO*, Eolss publishers,Oxford UK, 2004.

[30] GAO, Z. and P. J. ANTSAKLIS, «Stability of the pseudo-inverse method for reconfigurable control systems», *International Journal of Control*, **53**(3), pages. 717–729, 1991.

[31] GAO, Z. and P. J. ANTSAKLIS, «Reconfigurable control system design via perfect model following», *International Journal of Control*, **56**(4), pages. 783–798, 1992.

[32] GERTLER, J. and D. SINGER, «A new structural framework for parity equation-based failure detection and isolation», *Automatica*, **26**(2), pages. 381–388, 1990.

[33] GOERZEN, C., Z. KONG and B. METTLER, «A survey of motion planning algorithms from the perspective of autonomous UAV guidance», *Journal of Intelligent and Robotic Systems*, **57**(1-4), pages. 65–100, 2010.

[34] HENGY, D. and P. FRANK, «Component failure detection via nonlinear state observers», in *Proc. of IFAC Workshop on Fault Detection and Safety in Chemical Plants*, pages. 153–157, 1986.

[35] HORN, J., J. BAMBERGER, P. MICHAU and S. PINDL, «Flatness-based clutch control for automated manual transmissions», *Control Engineering Practice*, **11**(12), pages. 1353–1359, 2003.

[36] HUGENHOLTZ, C. H., B. J. MOORMAN, K. RIDDELL and K. WHITEHEAD, «Small unmanned aircraft systems for remote sensing and earth science research», *Eos, Transactions American Geophysical Union*, **93**(25), pages. 236–236, 2012.

[37] ISERMANN, R., *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*, Springer, 2006.

[38] JOIN, C., H. SIRA-RAMIREZ and M. FLIESS, «Control of an uncertain three-tank-system via on-line parameter identification and fault detection», in *Proc. 16th IFAC World Congress on Automatic Control, Prague, Czech Republic*, 2005.

[39] KANTNER, M., B. BODENHEIMER, P. BENDOTTI and R. M. MURRAY, «An experimental comparison of controllers for a vectored thrust, ducted fan engine», in *Proceedings of the American Control Conference, Seattle, Washington USA*, volume 3, IEEE, pages. 1956–1961, 1995.

[40] KOLCHIN, E. R., *Differential algebra & algebraic groups*, volume 54, Academic press, 1973.

[41] KRISHNASWAMI, V. and G. RIZZONI, «A survey of observer based residual generation for FDI», in *IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Helsinki, Finland*, IFAC, pages. 34–37, 1994.

[42] KRÖGER, T., *On-Line Trajectory Generation in Robotic Systems*, *Springer Tracts in Advanced Robotics*, volume 58, Springer, Berlin, Heidelberg, Germany, January 2010.

[43] LAVIGNE, L., *Outils d'analyse et de synthèse des lois de commande robuste des systèmes dynamiques plats.*, PhD. thesis, Université Sciences et Technologies-Bordeaux I, 2003.

[44] LEUSCHEN, M. L., I. D. WALKER and J. R. CAVALLARO, «Fault residual generation via nonlinear analytical redundancy», *IEEE Transactions on Control Systems Technology*, **13**(3), pages. 452–458, 2005.

[45] LÉVINE, J., *Analysis and control of nonlinear systems*, Springer Berlin, 2009.

[46] LÉVINE, J., «On necessary and sufficient conditions for differential flatness», *Applicable Algebra in Engineering, Communication and Computing*, **22**(1), pages. 47–90, 2011.

[47] LOOZE, D., J. WEISS, J. ETERNO and N. BARRETT, «An automatic redesign approach for restructurable control systems», *IEEE Control Systems Magazine*, **5**(2), pages. 16–22, 1985.

[48] LÓPEZ FERNÁNDEZ, J. L. and C. RAVENTÓS OLIVELLA, «Model based fault detection using differential flatness», *5th AIAA-PEGASUS Student conference, Toulouse, France*, 2009.

[49] LOUEMBET, C., *Génération de trajectoires optimales pour systèmes différentiellement plats: application aux manoeuves d'attitude sur orbite*, PhD. thesis, Université de Bordeaux 1, 2007.

[50] LOUEMBET, C., F. CAZAURANG and A. ZOLGHADRI, «Motion planning for flat systems using positive B-splines: An LMI approach», *Automatica*, **46**(8), pages. 1305–1309, 2010.

[51] LOZANO-PEREZ, T., «A simple motion-planning algorithm for general robot manipulators», *IEEE Journal of Robotics and Automation*, **3**(3), pages. 224–238, 1987.

[52] LUNZE, J. and J. RICHTER, «Control Reconfiguration: Survey of Methods and Open Problems», Bericht Nr. 2006.08, Ruhr-University of Bochum, June 2006.

[53] MAI, P., C. JOIN and J. REGER, «An example of flatness based fault tolerant control using algebraic derivative estimation», in *IAR Workshop, Nancy*, 2006.

[54] MAI, P., C. JOIN, J. REGER *et al.*, «Flatness-based Fault Tolerant Control of a nonlinear MIMO system using algebraic derivative estimation», in *3rd IFAC Symposium on System, Structure and Control, Foz do Iguassu, Brazil*, 2007.

[55] MARTIN, P., *Contribution à l'étude des systèmes différentiellement plats*, PhD. thesis, École des mines de Paris, 1992.

[56] MARTIN, P., S. DEVASIA and B. PADEN, «A different look at output tracking: Control of a VTOL aircraft», *Automatica*, **32**(1), pages. 101–107, 1996.

[57] MARTIN, P. and P. ROUCHON, «Any (controllable) driftless system with m inputs and m+ 2 states is flat», in *Proceedings of the 34th IEEE Conference on Decision and Control, New Orleans, Los Angeles USA*, volume 3, IEEE, pages. 2886–2891, 1995.

[58] MARTÍNEZ TORRES, C., L. LAVIGNE, F. CAZAURANG, E. ALCORTA GARCIA and D. DIAZ, «Control reconfiguration for differentially flat systems.», in *Congreso Nacional de Control Automático (AMCA). Ensenada, Mexico*, 2013.

[59] MARTÍNEZ TORRES, C., L. LAVIGNE, F. CAZAURANG, E. ALCORTA GARCIA and D. DIAZ, «Fault detection and isolation on a three tank system using differential flatness.», in *European Control Conference. Zurich, Switzerland*, 2013.

[60] MARTÍNEZ TORRES, C., L. LAVIGNE, F. CAZAURANG, E. ALCORTA GARCIA and D. DIAZ, «Fault tolerant control of a three tank system: a flatness based approach», in *2nd International Conference on Control and Fault-Tolerant Systems. Nice, France*, 2013.

[61] MBOUP, M., C. JOIN and M. FLIESS, «A revised look at numerical differentiation with an application to nonlinear feedback control», in *Mediterranean Conference on Control & Automation, Athens, Greece*, IEEE, pages. 1–6, 2007.

[62] MILAM, M. B., R. FRANZ, J. E. HAUSER and R. M. MURRAY, «Receding horizon control of vectored thrust flight experiment», *IEE Proceedings-Control Theory and Applications*, **152**(3), pages. 340–348, 2005.

[63] MOGENS, B., K. MICHEL, L. JAN and S. MARCEL, *Diagnosis and Fault-tolerant control*, second edition, Springer, 2006.

[64] MORIO, V., *Contribution au développement d'une loi de guidage autonome par platitude. Application à une mission de rentrée atmosphérique.*, PhD. thesis, Université Sciences et Technologies-Bordeaux I, 2009.

[65] MORIO, V., A. FALCOZ and F. CAZAURANG, «On the design of a flatness-based guidance algorithm for the terminal area energy management of a winged-body vehicle», in *Proc. of the 17th IFAC Symposium on Automatic Control in Aerospace, Toulouse, France*, 2007.

[66] NAN, Z., D. ANTOINE, D. ANDREI and M.-C. FELIXO, «A differential flatness approach for rotorcraft fault detection», in *27th Chinese Control Conference, Kunming, China*, IEEE, pages. 237–241, 2008.

[67] NGUANG, S. K., P. ZHANG and S. X. DING, «Parity relation based fault estimation for nonlinear systems: An LMI approach», *International Journal of Automation and Computing*, **4**(2), pages. 164–168, 2007.

[68] NONAMI, K., F. KENDOUL, S. SUZUKI, W. WANG and D. NAKAZAWA, *Autonomous Flying Robots: Unmanned Aerial Vehicles and Micro Aerial Vehicles*, Springer Publishing Company, Incorporated, 2010.

[69] NØRGAARD, M., N. K. POULSEN and O. RAVN, «New developments in state estimation for nonlinear systems», *Automatica*, **36**(11), pages. 1627–1638, 2000.

[70] NOURA, H., D. THEILLIOL, J.-C. PONSART and A. CHAMSEDDINE, *Fault-tolerant control systems: Design and practical applications*, Springer, 2009.

[71] QUADRAT, A. and D. ROBERTZ, «Constructive computation of flat outputs of a class of multidimensional linear systems with variable coefficients», in *Proceedings of the Mathematical Theory of Networks and Systems , Kyoto, Japan*, pages. 24–28, 2006.

[72] RICHTER, J., *Reconfigurable Control of Nonlinear Dynamical Systems*, PhD. thesis, Fakultät für Elektrotechnik, Ruhr-Universität Bochum, 2010.

[73] RICHTER, J. H., *Reconfigurable control of nonlinear dynamical systems: a fault-hiding approach*, volume 408, Springer, 2011.

[74] ROTHFUSS, R., J. RUDOLPH and M. ZEITZ, «Flatness based control of a nonlinear chemical reactor model», *Automatica*, **32**(10), pages. 1433–1439, 1996.

[75] SIRA-RAMÍREZ, H. and S. K. AGRAWAL, *Differentially flat systems*, volume 17, Birkhauser, 2004.

[76] STAROSWIECKI, M., «Fault tolerant control: the pseudo-inverse method revisited», in *Proc. 16th IFAC World Congress, Prague, Czech Republic*, 2005.

[77] STAROSWIECKI, M., J. CASSAR and G. COMTET-VARGA, «Analytic redundancy relations for state affine systems», in *proceedings of the fourth European control conferences, Brussels, Belgium*, 1997.

[78] STAROSWIECKI, M. and G. COMTET-VARGA, «Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems», *Automatica*, **37**(5), pages. 687–699, 2001.

[79] STENGEL, R. F., «Intelligent failure-tolerant control», *Control Systems Magazine, IEEE*, **11**(4), pages. 14–23, 1991.

[80] STUMPER, J.-F., F. SVARICEK and R. KENNEL, «Trajectory Tracking Control with Flat Inputs and a Dynamic Compensator», in *Proceedings ot the European Control conference, Budapest, Hungary*, pages. 248–253, 2009.

[81] SURYAWAN, F., J. DE DONA and M. SERON, «Fault detection, isolation, and recovery using spline tools and differential flatness with application to a magnetic levitation system», in *Conference on Control and Fault-Tolerant Systems (SysTol), Nice, France*, IEEE, pages. 293–298, 2010.

[82] VAN NIEUWSTADT, M. J. and R. M. MURRAY, «Real-time trajectory generation for differentially flat systems», *International Journal of Robust and Nonlinear Control*, **8**(11), pages. 995–1020, 1998.

[83] VASILJEVIC, L. K. and H. K. KHALIL, «Error bounds in differentiation of noisy signals by high-gain observers», *Systems & Control Letters*, **57**(10), pages. 856–862, 2008.

[84] VENKATASUBRAMANIAN, V., R. RENGASWAMY and S. N. KAVURI, «A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies», *Computers & Chemical Engineering*, **27**(3), pages. 313–326, 2003.

[85] VENKATASUBRAMANIAN, V., R. RENGASWAMY, S. N. KAVURI and K. YIN, «A review of process fault detection and diagnosis: Part III: Process history based methods», *Computers & Chemical Engineering*, **27**(3), pages. 327–346, 2003.

[86] VENKATASUBRAMANIAN, V., R. RENGASWAMY, K. YIN and S. N. KAVURI, «A review of process fault detection and diagnosis: Part I: Quantitative model-based methods», *Computers & Chemical Engineering*, **27**(3), pages. 293–311, 2003.

[87] YANG, Q., A. AITOUCHE and B. O. BOUAMAMA, «Fault detection and isolation of pem fuel cell system by analytical redundancy», in *Control & Automation (MED), 2010 18th Mediterranean Conference on*, IEEE, pages. 1371–1376, 2010.

[88] YU, M., D. WANG, M. LUO and D. H. ZHANG, «FDI and fault estimation based on differential evolution and analytical redundancy relations», in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, IEEE, pages. 1341–1346, 2010.

[89] ZEITZ, M., «The extended Luenberger observer for nonlinear systems», *Systems & Control Letters*, **9**(2), pages. 149–156, 1987.

[90] ZERAR, M., F. CAZAURANG and A. ZOLGHADRI, «Robust tracking of nonlinear MIMO uncertain flat systems», in *International Conference on Systems, Man and Cybernetics, The Hague, Netherlands*, volume 1, IEEE, pages. 536–541, 2004.

[91] ZHANG, N., A. DONCESCU, A. C. B. RAMOS and F. MORA-CAMINO, «Fault Detection for Difference Flat Systems», in *Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong*, volume 2, 2012.

[92] ZHANG, Y. and J. JIANG, «Bibliographical review on reconfigurable fault-tolerant control systems», *Annual Reviews in Control*, **32**(2), pages. 229–252, 2008.

[93] ZHOU, K., J. C. DOYLE, K. GLOVER *et al.*, *Robust and optimal control*, volume 40, Prentice Hall New Jersey, 1996.

LIST OF FIGURES

List of tables

APPENDIX A

TRAJECTORY GENERATION BY POLYNOMIAL APPROACH

This appendix recalls the construction of trajectories, by using the polynomial approach.

With the given initial and final conditions c_{ini} and c_{fin}, the trajectory generation problem consists in create a function $f(t)$ which fulfills those constraints.

This is a boundary condition problem, that can be easily solved by considering polynomial functions such as:

$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + ... + a_n t^n \tag{A.1}$$

Where a_i when $i = 0, 1, 2...n$ are polynomial coefficients, and t is the time.

The degree (n) of the polynomial depends on the number of boundary conditions (co) that must be verified and on the desired “smoothness” of the trajectory. This degree has to be at least equals to the number of constraints, minus one.

Mathematically, these conditions may be expressed in matrix form as:

$$M * a = b \tag{A.2}$$

Where M is a known $(n + 1) * (n + 1)$ matrix, composed by the time part of the equation A.1, b is the vector containing the known constraints. a contains the unknown coefficients.

The value of the coefficients can be easily computed by using the next expression

$$a = M^{-1} b \tag{A.3}$$

Example A.1 *Given the initial and final conditions, for the position* p_{ini}, p_{fin}*, velocity* v_{ini}, v_{fin} *and acceleration* a_{ini}, a_{fin}*. Construct the polynomial function which fulfills the constraints. Since it exists six boundary conditions, the minimum degree of the polynomial function has to be five.*

$$f(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \tag{A.4}$$

where the six parameters a_0*,* a_1*,* a_2*,* a_3*,* a_4 *and* a_5 *must be defined so that the boundary conditions are satisfied. Let us define the boundary conditions as follows:*

$$\begin{aligned} p_{ini} &= 0 \\ p_{fin} &= 5 \end{aligned} \tag{A.5}$$

$$\begin{aligned} v_{ini} &= 0 \\ v_{fin} &= 0 \end{aligned} \tag{A.6}$$

$$\begin{aligned} a_{ini} &= 0 \\ a_{fin} &= 0 \end{aligned} \tag{A.7}$$

$t_i = 0s$ *and* $t_f = 400s$*.*

In order to compute the values of the coefficients, the next steps has to be followed.

- *Compute the time derivatives of the selected polynomial function A.4, until the maximum time derivative becomes equal to* co*.*

$$\begin{aligned} f(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ \dot{f}(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{f}(t) &= 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{aligned} \tag{A.8}$$

- *Substitute the value of the* $t_i = 0$ *in the first equations. Equals each equations to the initial conditions. This operation will give the values of the first three coefficients,* a_0*,* a_1 *and* a_2*.*

$$\begin{aligned} a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 &= 0 \\ a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 &= 0 \\ 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 &= 0 \end{aligned} \tag{A.9}$$

- *Construct the matrix form A.2. Substitute t by t_f in the equations A.9 and sort them into the matrix M together with the three equations presented above. The rest of the coefficients (a_3, a_4 and a_5.) are find by inverting the matrix M as show in equation A.3.*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ (t_f)^5 & (t_f)^4 & (t_f)^3 & (t_f)^2 & t_f & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5(t_f)^4 & 4(t_f)^3 & 3(t_f)^2 & 2t_f & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20(t_f)^3 & 12(t_f)^2 & 6t_f & 2 & 0 & 0 \end{bmatrix} * \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.10}$$

The value of the coefficients are.

$$\begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 2.9297 * 10^{-12} \\ -2.9297 * 10^{-9} \\ 7.8125 * 10^{-7} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.11}$$

Once the coefficients found, it suffices of create a time vector from t_i to t_f to obtain the desired trajectory. Figure A.1 shows the trajectories of position, velocity and acceleration.

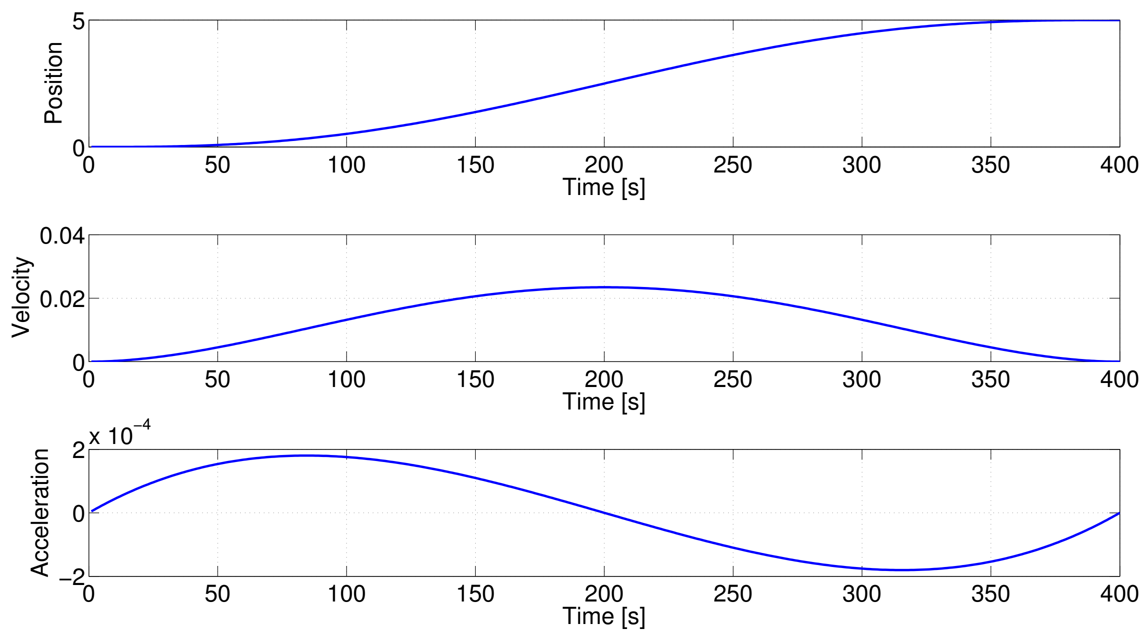

Figure A.1: Trajectories