UNIVERSITÀ DI PISA

Facoltà di Ingegneria

Laurea Specialistica in Ingegneria dell'Automazione

Tesi di laurea

LPV MODELLING, IDENTIFICATION AND CONTROL INCLUDING FAULT TOLERANCE MECHANISMS: APPLICATION TO A TWO-DEGREE OF FREEDOM HELICOPTER

Candidato:

Damiano Rotondo

Relatori:	Controrelatore:
Prof. Mario Innocenti	*Prof. Lorenzo Pollini*
Prof. Vicenç Puig	
Prof.ssa Fatiha Nejjari	

Sessione di Laurea del 07/10/2011
Anno accademico 2010/2011
Consultazione consentita

A sierma, mama e frama
ca era dittu ca l'er'a fa'
e nah! lu sta' fazzu!

Mi dediĉas ĉi magistriĝan tezon
al la miaj gepatroj kaj frato.

ABSTRACT

This thesis presents the LPV (Linear Parameter Varying) modelling, identification and control with fault tolerance mechanisms of a Twin Rotor MIMO System (TRMS).

The non-linear model of the TRMS is transformed into a quasi-LPV model and then approximated in a polytopic way. The model parameters are calibrated by means of a non-linear least-square identification approach. Once the calibrated model is obtained, a simulator is built and validated against real data.

Using the LPV LMI (Linear Matrix Inequalities) pole placement approach, an LPV observer and an LPV controller are designed and tested both in simulation and on the real system in order to prove its effectiveness and performance.

Finally, a Fault Tolerant Control strategy using virtual sensors and virtual actuators is applied in an LPV way. In order to implement these approaches, fault estimation is required. To this aim, a method to identify the faults is implemented formulating a parameter estimation problem. The methodology is tested both in simulation and on the real system, showing its effectiveness and its limits.

SOMMARIO

Questa tesi presenta la modellazione, l'identificazione ed il controllo LPV (Linear Parameter Varying, sistemi lineari con parametri varianti) con meccanismi di tolleranza ai guasti di un Twin Rotor MIMO System (TRMS).

Il modello non lineare del TRMS è trasformato in un modello quasi-LPV ed è approssimato in maniera politopica. Dopodiché, i parametri del modello sono stati identificati con un approccio d'identificazione basato sui minimi quadrati non lineari. Una volta ottenuto un modello calibrato, un simulatore è stato costruito e validato con dati reali.

Usando un approccio di piazzamento dei poli mediante risoluzione di disuguaglianze matriciali lineari (LMI), sono stati progettati un osservatore ed un controllore LPV, e questi sono stati testati sia sul simulatore che sul sistema reale, provando la loro efficacia e le loro prestazioni.

Infine, una strategia di controllo tollerante ai guasti basata sull'idea dei sensori e degli attuatori virtuali è stata applicata in maniera LPV. Per implementare tale strategia, una stima dei guasti è richiesta; per questo motivo è stato implementato un metodo per identificare i guasti attraverso la formulazione di un problema di stima dei parametri. La metodologia è stata applicata al simulatore ed al sistema reale, mostrando la sua efficacia, ma anche i suoi limiti.

INDEX

1 – INTRODUCTION

1.1 – Background

This thesis describes the LPV modelling, identification and control including fault tolerance mechanisms of a Twin-Rotor MIMO System (TRMS) developed by Feedback Instruments Limited for control experiments. This system resembles a simplified behaviour of a conventional helicopter with less degree of freedom. The system is perceived as a challenging engineering problem owing to its high non-linearity, cross-coupling between its two axes, and inaccessibility of some of its states and outputs for measurements.

The modelling of such a system has been addressed in several papers. In particular, in **[Rahideh, 2007]** two physical models based on Newtonian and Lagrangian approaches are presented and compared. Further modelling improvements can be found in **[Gabriel, 2008]** and **[Christensen, 2006].**

The experimental identification of the parameters of the TRMS model has been addressed in several papers too. In **[Ahmad, 2002]**, a dynamic model of a one-degree-of-freedom TRMS in hover is obtained using a black-box system identification technique. The extracted model is employed for designing and implementing a feedforward/open-loop control. In **[Darus, 2004]**, the parameters of the TRMS in hovering position are identified through the utilization of a genetic algorithm optimization technique based on one-step-ahead prediction. In **[Alam, 2007]**, a particle swarm optimization (PSO) algorithm that uses time-varying inertia weight factor and time-varying acceleration coefficients is used to extract both 1-DOF models for both vertical (pitch) and horizontal (yaw) channels, and a 2-DOF parametric model that takes into account cross-coupling between the channels. In **[Toha, 2008]**, an adaptive neuro-fuzzy inference system (ANFIS) network design is deployed and used for modelling the TRMS.

Regarding the control, a nonlinear predictive control has been presented in **[Dutka, 2003]**. The non-linearity is handled by converting the state-dependent state-space representation into the linear time-varying representation. In **[Lopez, 2003a]** and **[Lopez, 2004]**, the control of the twin rotor system using feedback linearization techniques (as full state linearization and input output linearization) has been suggested. In **[Lopez, 2003b]**, a H_∞ controller for helicopter dynamics is proposed. Later, a non linear H_∞ approach for handling the coupling taken as disturbance that should be rejected is introduced in **[Lopez, 2005]**. The resulting controller exhibited attributes of a non-linear PID with time-varying constants according to the operating point. In **[Ahmed, 2009]**, a sliding mode control is considered by defining a sliding surface that allows to deal with cross-coupling inherent in the twin rotor dynamics.

Fault Tolerant Control (FTC) is a new idea recently introduced in the research literature **[Blanke, 2003]** which allows keeping current performance close to desirable one and preserving stability conditions in presence of components and/or instrument faults. Accommodation capability of a control system depends on many factors such as severity of the failure, the robustness of the nominal system and mechanisms that introduce redundancy in sensors and/or actuators. From the point of view of the control strategies, the literature considers two main groups of techniques: the *active* and the *passive* (**[Zhang, 2008]** for a review). Most of the FTC methods have been proposed for LTI systems. The LPV theory, used in this thesis, is mainly used for designing controllers for non-faulty systems, but recently it has also been used for active FTC **[Rodrigues, 2007]**. Recently, virtual sensors and virtual actuators have been proposed as a fault accommodation approach **[Blanke, 2003]**. The main idea of this FTC method is to reconfigure the control loop such that the nominal controller could still be used without need of retuning it. This means that the faulty plant is adapted to the nominal controller instead of adapting the controller to the faulty plant. That is, the faulty plant together with the virtual actuator/sensor block allows the controller to see the same plant as before the fault.

1.2 – Objectives

The aim of this thesis is to design a controller for the TRMS in an LPV way. The controller will comprehend some active fault tolerance mechanisms to avoid loss of stability and to reduce the loss of performances due to the occurrence of a fault either in one of the sensors or in one of the actuators. To reach this aim, the TRMS dynamics must be modelled and its parameters have to be identified. Afterwards, the control strategy and the fault tolerance mechanisms will be implemented and tested both on a simulator and on the real system.

1.3 – Contributions

The contributions of this thesis are the following:

- the Newtonian model of the TRMS proposed in **[Rahideh, 2007]** is modified and transformed into a quasi-LPV model using the transformation method proposed by **[Shamma, 1992a]**;
- a non-linear identification approach that is based on formulating a non-linear least square problem is used to estimate the parameters of the non-linear model of the TRMS. The obtained values are then used both for control design purposes and for building a Matlab simulator to test the control strategies before applying them to the real set-up;
- the TRMS model is approximated by a polytopic LPV model such that LPV control theory can be applied. This control methodology allows to design a state feedback gain-scheduling controller that is able to adapt the control gains with the operating point using a scheduling function and scheduling variables;
- an LPV observer is designed to estimate the TRMS states;
- LPV controllers are designed to control the yaw and the pitch angles, and their performances are compared;
- active fault tolerance mechanisms are designed in an LPV way and added to the control loop, so as to give the system fault-rejection property;
- the designed observer, controller and fault tolerance mechanisms are tested both in the simulation environment and with the real system;

1.4 – Outline of the thesis

The thesis is organized as follows:

Chapter 1 – Introduction

Chapter 2 – Preliminaries

This chapter introduces some basic concepts that are needed later in this thesis. This thesis proposes the use of a Linear Parameter Varying (LPV) approach to control a TRMS system. Thus, some theory about LPV systems is introduced, and a way to obtain an LPV representation from a classical non-linear representation is briefly explained. Two types of LPV models are described in this chapter. First, an LPV modelling approach based on “hiding” the non-linearities in the time-varying parameter vector (“absolute representation”). Then, another LPV representation is presented, based on state transformation, where the equilibrium point is not included explicitly, but the new time-varying parameters depend on the equilibrium point (“relative representation”).

Most of the thesis relies heavily on linear matrix inequalities (LMIs) that need to be solved in order to place the eigenvalues of some closed-loop system in a desired subregion of the complex plane. Concepts dealing with pole placement in an LMI region will be briefly discussed.

Finally, basic concepts about Fault Tolerant Control will be resumed.

Chapter 3 - TRMS Modelling

In this chapter, the Twin Rotor MIMO System, a laboratory setup developed by Feedback Instruments Limited for control experiments, is described. At first, a physical model is obtained by considering the dynamics of the DC motors, of the propellers and the horizontal and vertical subsystem dynamics. Then, techniques described in Chapter 2 will be used to obtain a TRMS LPV representation: an LPV absolute representation of the TRMS is used as an intermediate step to obtain the so-called "Shamma representation" (based on state transformation) that can be approximated by a polytopic LPV representation that is the model used for control design purposes in this thesis.

Chapter 4 – TRMS Identification

In this chapter, the unknown parameters that describe the TRMS dynamics are identified. The procedure that has been used is explained and the results of the identification are shown. At the end of the chapter, some advices to improve the goodness of the model are given, and the bounding box approach is used to obtain a polytopic LPV representation for the system, to be used for control design purposes.

Chapter 5 – TRMS LPV Control

In this chapter, the TRMS is controlled by an LPV state feedback controller. The controller is designed using different LPV models, all obtained from the non-linear model described in Chapter 3 and identified in Chapter 4. The effectiveness of the control strategy is shown and compared to basic PID control strategy.

Chapter 6 –LPV Fault-Tolerant Control for the TRMS

In this chapter, active fault-tolerance mechanisms are designed in both cases of sensor and actuator faults. The fault tolerance mechanisms are tested both in simulation and on the real system.

Chapter 7 – Conclusions and Future Works

Appendix A – Tables of TRMS Quantities

Appendix B – Linear Matrix Inequalities

2 – PRELIMINARIES

2.1 – Introduction

In recent years an emerging approach to control theory known as Linear Parameter Varying (LPV) Control, has been developed as an alternative to classical gain-scheduling techniques. Classical gain-scheduling is a method to design controllers for dynamical systems over a wide performance envelope. The controller obtained is based on the interpolation of a family of locally linearized controllers, but no guarantees are given on its stability and robustness other than at the design points. LPV Control synthesis fits into the gain scheduling framework, while adding stability and robustness guarantees. The strength of the LPV approach lies in the extension of well-known methods for linear optimal control, including the use of linear matrix inequalities (LMIs), to the design of gain scheduled LPV controllers. A condition to apply LPV control synthesis is to transform the nonlinear model of the system into an LPV model; hence LPV modelling becomes a key issue in the design of LPV controllers. Luckily, many nonlinear systems of practical interest can be represented as quasi-LPV systems, where *quasi* is added because the scheduling parameters do not depend only on external signals, but on system variables too.

In recent years, Active Fault Tolerant Control (AFTC) has become an important topic in control theory, and techniques have been developed to reduce and compensate the effects of the fault on the behaviour of the controlled system. Among these techniques, virtual sensors and virtual actuators have been created to adapt the faulty plant to the nominal controller, so as to hide the fault.

In Section 2.2, LPV theory will be reviewed; in Section 2.3, some theory about controller design with pole placement constraints will be explained. Lastly, in Section 2.4, some basic informations about fault tolerant control will be given.

2.2 – LPV Modelling

2.2.1 – Linear Parameter Varying Systems

In gain-scheduling, the basic concept is to linearize the non-linear system model at different operating points resulting in a collection of local LTI descriptions of the plant. Then, subsequently, LTI controllers are designed for each local operating point considered. These controllers are interpolated to give a global control solution to the entire operation regime. The used interpolation function is called the *scheduling function* and it is dependent on the current operating point of the plant. To describe the changes of the operating point, a signal is introduced, which is called the *scheduling signal*. In this way, the parameters of the resulting controller are dependent on this varying signal; hence the name *parameter-varying*, while the dynamic relation between system signals is still *linear*. Due to many successful applications of this design methodology, gain-scheduling has become popular in industrial applications, even if guarantees for overall stability of the designed LPV controllers was not available and the possibility of malfunction has existed. This was resolved by the introduction of interpolation based methods that guarantee global stability. In the mean time, researchers realized that in general many non-linear systems can be converted into an LPV form. Approaches that provided direct LPV models for gain-scheduling without the laborious process of NL system modelling or identification have appeared. The LPV field has evolved rapidly in the last decades and has become a promising framework for modern industrial control with a growing number of applications like aircrafts, re-entry vehicles, automobiles, wind turbines, induction motors, servo systems, wafer steppers, internet web servers, CD-players and environmental modelling. For further informations see **[Tóth, 2010]**.

2.2.2 – Polytopic LPV Representation

In **[Shamma, 1992a]**, linear parameter-varying systems (LPV) are defined as linear time-varying plants whose state-space matrices are fixed functions of some vector of varying parameters $\boldsymbol{\psi}(k)$. They are described by state-space equations of the form:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}(\boldsymbol{\psi}(k))\mathbf{x}(k) + \mathbf{B}(\boldsymbol{\psi}(k))\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}(\boldsymbol{\psi}(k))\mathbf{x}(k) + \mathbf{D}(\boldsymbol{\psi}(k))\mathbf{u}(k) \end{cases} \tag{2.2.1}$$

where $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ denotes the state vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ denotes the control inputs and $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ represents the sensor outputs. The matrices have the following dimensions: $\mathbf{A}(\boldsymbol{\psi}(k)) \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B}(\boldsymbol{\psi}(k)) \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{C}(\boldsymbol{\psi}(k)) \in \mathbb{R}^{n_y \times n_x}$. $\boldsymbol{\psi}(k)$ is the system vector of time-varying parameters of dimension n_ψ that changes with the operating point scheduled by some measured system variables $\mathbf{p}(k)$ that can be estimated using some known function $\boldsymbol{\psi}(k) = f_\psi(\mathbf{p}(k))$, known as scheduling function.

LPV systems can either be viewed as linear time-invariant (LTI) plants subject to time-varying parametric uncertainty $\boldsymbol{\psi}(k)$, or as linear time-varying plants, or as a result from the linearization of nonlinear plants along the trajectories of the parameter $\boldsymbol{\psi}$.

In **[Apkarian, 1995]** and **[Chilali, 1996]**, linear matrix inequality techniques are applied for gain-scheduled control of LPV systems. These techniques have been developed both for continuous-time systems and for discrete-time systems. Since the controller for the Twin Rotor MIMO System will be designed as a discrete-time system, only the theory for LPV discrete-time systems will be reviewed in the following.

The kind of LPV systems considered are those whose time-varying parameter vector $\boldsymbol{\psi}(k)$ varies within a polytope Θ and whose input and output matrices $\mathbf{B}$ and $\mathbf{C}$ do not depend on the vector $\boldsymbol{\psi}$. More precisely, in this kind of LPV system, the state matrix ranges in a polytope of matrices defined as the convex hull of a finite number N of matrices. Each polytope vertex corresponds to a particular value of the scheduling variable $\boldsymbol{\psi}(k)$. In other words:

$$\begin{pmatrix} \mathbf{A}(\boldsymbol{\psi}(k)) & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \in Co\left\{ \begin{pmatrix} \mathbf{A}_j & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{pmatrix}, j = 1, \ldots, N \right\} := \sum_{j=1}^{N} \alpha_k^j(\boldsymbol{\psi}(k)) \begin{pmatrix} \mathbf{A}_j & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{pmatrix} \tag{2.2.2}$$

with $\alpha_k^j(\boldsymbol{\psi}(k)) \geq 0$ and $\sum_{j=1}^{N} \alpha_k^j(\boldsymbol{\psi}(k)) = 1$, where each j^{th} model is called a *vertex* system. Because of this property, this type of LPV systems is referred as polytopic.

A common approach to design a controller/observer for the LPV system (2.2.1) is to approximate it by a polytopic LPV system. The simplest polytopic approximation relies on bounding each LPV parameter by an interval. This approximation is known as *bounding box* approach. An alternative approach known as *small hull* approach proposed by **[Kumar, 2000]** is based on the fact that not only the range of varying parameters is known, but also intelligent choices can be made about the intermediate values that the varying parameter $\boldsymbol{\psi}(k)$ can take. From this set of parameter values, the convex hull that contains them is obtained using the quick hull algorithm **[Barber, 1996]**.

Proceeding with one of these two approaches, the polytopic LPV system (2.2.1) can be expressed as follows:

$$\mathbf{x}(k+1) = \sum_{j=1}^{N} \alpha_k^j(\boldsymbol{\psi}(k))\left[\mathbf{A}_j\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k)\right] \tag{2.2.3}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \tag{2.2.4}$$

where $\mathbf{x}(k) \in \mathbb{R}^{n_x}$ denotes the state vector, $\mathbf{u}(k) \in \mathbb{R}^{n_u}$ denotes the control inputs and $\mathbf{y}(k) \in \mathbb{R}^{n_y}$ denotes the sensor outputs. Here $\mathbf{A}_j \in \mathbb{R}^{n_x \times n_x}$, $\mathbf{B} \in \mathbb{R}^{n_x \times n_u}$ and $\mathbf{C} \in \mathbb{R}^{n_y \times n_x}$ are time-invariant matrices defined for j^{th} model. The polytopic system is scheduled through functions designed as follows: $\alpha_k^j(\boldsymbol{\psi}_k)$, $\forall j \in [1,...,N]$ that lie in a convex set:

$$\Omega = \left\{ \alpha_k^j(\boldsymbol{\psi}(k)) \in \mathbb{R}^N, \alpha_k(\boldsymbol{\psi}(k)) = \left[\alpha_k^1(\boldsymbol{\psi}(k)),...,\alpha_k^N(\boldsymbol{\psi}(k))\right]^T, \alpha_k^j(\boldsymbol{\psi}(k)) \geq 0, \forall j, \sum_{j=1}^{N} \alpha_k^j(\boldsymbol{\psi}(k)) = 1 \right\} \quad (2.2.5)$$

2.2.3 – Types of LPV Models

There are three possible approaches to obtain an LPV model:

1. *Physical modelling*: The model is obtained by physical laws taking into account that the model parameters vary according to the system states, the operating conditions and/or external factors.
2. *Multi-model identification*: It is a two-step procedure where:
   - LTI models are identified at several different equilibrium (operating condition) by classical methods;
   - a global multi-model is obtained by interpolating among the local LTI models.

   The multiple model approach obtained by physical laws or identification can be viewed as a single linear parameter varying (LPV) global model. The parameters of the resultant global model are the result of interpolating the local estimated parameters by suitable set-membership functions.
3. *LPV identification*: This procedure is an extension of the classical identification (via linear regression). In this case, the varying parameters of the system (external parameters and/or space state) are assumed unknown a priori. However, they are measured or estimated upon operation of the system. The identified model parameters depend on the previous varying parameters.

The physical modelling will be used in this work to obtain the LPV model through two approaches. One is based on **[Kwiatkowski, 2006]** (denoted as "Absolute LPV" in the following) and the other is based on **[Shamma, 1992b]** (denoted as "Shamma LPV" in the following).

2.2.4 – Absolute LPV Representation

This approach is introduced by **[Kwiatkowski, 2006]**. The method facilitates the construction of a suitable LPV model from a given non-linear system:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \\ \mathbf{y} = \mathbf{g}(\mathbf{x},\mathbf{u}) \end{cases} \quad (2.2.6)$$

where ***x*** is the state, ***u*** is the input and ***y*** denotes an available measured output.

The idea is to express the model (2.2.6) in the following way in order to facilitate the procedure:

$$\begin{cases} \dot{x}_i = \sum_{j=1}^{rx_i} f_{ij}(\mathbf{x},\mathbf{u}), i = 1,...,n_x \\ y_i = \sum_{j=1}^{ry_i} g_{ij}(\mathbf{x},\mathbf{u}), i = 1,...,n_y \end{cases} \quad (2.2.7)$$

The integers rx_i and ry_i denote the number of the terms in each state and output of (2.2.6). In the following, there is no difference in handling a state or an output. For this reason, in the following, the state equations are considered. Each term is decomposed into its numerator α_{ij}, denominator β_{ij} and a constant factor k_{ij}:

$$\dot{x}_i = \sum_{j=1}^{rx_i} k_{ij} \frac{\alpha_{ij}(\mathbf{x},\mathbf{u})}{\beta_{ij}(\mathbf{x},\mathbf{u})}, i = 1,\ldots,n_x \tag{2.2.8}$$

The numerator is factored to determine the possibilities of "hiding" the nonlinearities in the parameters:

$$\alpha_{ij} = \prod_{q=1}^{n_x} \prod_{r=1}^{n_u} h_{ij}(\mathbf{x},\mathbf{u}) x_q^{\mu_q} u_r^{\nu_r} \tag{2.2.9}$$

where $h_{ij}(\mathbf{x},\mathbf{u})$ denotes a "non-factorisable" term, $\mu_q \in \mathbb{Z}^+$ and $\nu_r \in \mathbb{Z}^+$ are the integer power of the state x_q and input u_r, respectively.

The classification of a term is determined by the structure of its numerator. Two classes can be distinguished:

- *Non-factorizable numerator,* K_0: For a term of this class a factor of the state $x_q^{\mu_q}$ or input $u_r^{\nu_r}$ as an element of the system matrix or input matrix could not be chosen. An example of K_0 is $f_{ij} = \sin x_1$.
- *Positive power of factor,* K_p: This class consists of terms that have numerators with positive integer powers of a state variable $x_q^{\mu_q}$ or input $u_r^{\nu_r}$. An example of K_p is $f_{ij} = 4x_1^2$.

According to this term classification, parameter components ϑ_{ij} can be chosen as follows:

$$\vartheta_{ij}^a = k_{ij} \frac{\alpha_{ij}(\mathbf{x},\mathbf{u})}{\beta_{ij} x_l}, l = 1, n_x \tag{2.2.10}$$

$$\vartheta_{ij}^b = k_{ij} \frac{\alpha_{ij}(\mathbf{x},\mathbf{u})}{\beta_{ij} u_l}, l = 1, n_u \tag{2.2.11}$$

where if the numerator is K_0, the parameter can be taken using n_x possible assignments to the system matrix ***A*** and n_u possible assignments to the input matrix ***B***. If the numerator is K_p the parameter is a factor of the numerator (see details in **[Kwiatkowski, 2006]**).

2.2.5 – Shamma LPV Representation

The absolute representation has the restriction of giving good performances only in case of slow variations in the scheduling variables. This means that a slow variation assumption is necessary in order to provide guarantees on the overall design.

A reformulation of LPV theory to overcome this limitation has been proposed in **[Shamma, 1992b]**. This approach is called *state transformation* because a quasi-LPV model (a system is said to be *quasi-LPV* when some of the scheduling variable are not exogenous but states of the system) is obtained through exact transformations of the nonlinear states.

Consider a nonlinear system of the form:

$$\begin{cases} \begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1) \\ \mathbf{f}_2(\mathbf{x}_1) \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{11}(\mathbf{x}_1) & \mathbf{A}_{12}(\mathbf{x}_1) \\ \mathbf{A}_{21}(\mathbf{x}_1) & \mathbf{A}_{22}(\mathbf{x}_1) \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{B}_1(\mathbf{x}_1) \\ \mathbf{B}_2(\mathbf{x}_1) \end{bmatrix} \mathbf{u} \\ \mathbf{y} = \mathbf{x}_1 \end{cases} \tag{2.2.12}$$

The system is nonlinear, but the nonlinearity is entirely contained in the output variable, $\mathbf{y}=\mathbf{x}_1$. Assume that there exist differentiable functions $\mathbf{x}_{2eq}$ and $\mathbf{u}_{eq}$ such that for every $\mathbf{x}_1$,

$$\begin{bmatrix}0\\0\end{bmatrix}=\begin{bmatrix}\mathbf{f}_1(\mathbf{x}_1)\\\mathbf{f}_2(\mathbf{x}_1)\end{bmatrix}+\mathbf{A}(\mathbf{x}_1)\begin{bmatrix}\mathbf{x}_1\\\mathbf{x}_{2eq}(\mathbf{x}_1)\end{bmatrix}+\begin{bmatrix}\mathbf{B}_1(\mathbf{x}_1)\\\mathbf{B}_2(\mathbf{x}_1)\end{bmatrix}\mathbf{u}_{eq}(\mathbf{x}_1) \tag{2.2.13}$$

Then, by defining new states and inputs:

$$\xi_1 := \mathbf{x}_1 \tag{2.2.14}$$

$$\xi_2 := \mathbf{x}_2-\mathbf{x}_{2eq}(\mathbf{x}_1) \tag{2.2.15}$$

$$\mathbf{v} := \mathbf{u}-\mathbf{u}_{eq}(\mathbf{x}_1) \tag{2.2.16}$$

$$\tilde{\mathbf{A}}_{22}(\xi_1) := \mathbf{A}_{22}(\xi_1)-\left.\frac{d\mathbf{x}_{2eq}}{d\mathbf{x}_1}\right|_{\mathbf{x}_1=\xi_1}\mathbf{A}_{12}(\xi_1) \tag{2.2.17}$$

$$\tilde{\mathbf{B}}_2(\xi_1) := \mathbf{B}_2(\xi_1)-\left.\frac{d\mathbf{x}_{2eq}}{d\mathbf{x}_1}\right|_{\mathbf{x}_1=\xi_1}\mathbf{B}_1(\xi_1) \tag{2.2.18}$$

a quasi-LPV system is obtained:

$$\begin{cases}\begin{bmatrix}\dot{\xi}_1\\\dot{\xi}_2\end{bmatrix}=\begin{bmatrix}\mathbf{0} & \mathbf{A}_{12}(\xi_1)\\\mathbf{0} & \tilde{\mathbf{A}}_{22}(\xi_1)\end{bmatrix}\begin{bmatrix}\xi_1\\\xi_2\end{bmatrix}+\begin{bmatrix}\mathbf{B}_1(\xi_1)\\\tilde{\mathbf{B}}_2(\xi_1)\end{bmatrix}\mathbf{v}\\ \mathbf{y}=\xi_1\end{cases} \tag{2.2.19}$$

Although the system representation in (2.2.19) has a linearized appearance, it still exactly represents the original nonlinear system. The LPV form is not equivalent to a Jacobian linearization around an operating point. For further informations see **[Balas, 2002]**.

2.3 – *Design with Pole Placement Constraints*

An LMI approach for pole placement design is described in **[Chilali, 1996]**. The main motivation for seeking pole clustering in specific regions of the complex plane is that, by constraining the closed-loop eigenvalues to lie in a prescribed region, stability can be guaranteed and a satisfactory transient response can be ensured.

Let $\bar{D}$ be a subregion of the unit circle. A dynamical system $\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)$ is called $\bar{D}$-stable if all its poles lie in $\bar{D}$ (that is, all eigenvalues of the matrix $\mathbf{A}$ lie in $\bar{D}$). By extension, $\mathbf{A}$ is then called $\bar{D}$-stable. When $\bar{D}$ is the entire unit circle, this notion reduces to asymptotic stability, which is characterized in LMI terms by the Lyapunov theorem.

Hereafter, $\otimes$ denotes the Kronecker product of matrices, and the notation $\mathbf{M}=[\mu_{kl}]_{1\le k,l\le m}$ means that $\mathbf{M}$ is an $m\times m$ matrix with generic entry μ_{kl}.

Definition (LMI Regions): A subset $\bar{D}$ of the complex plane is called an LMI region if there exist a symmetric matrix $\boldsymbol{\alpha}=[\alpha_{kl}]_{1\le k,l\le m}\in\mathbb{R}^{m\times m}$ and a matrix $\boldsymbol{\beta}=[\beta_{kl}]_{1\le k,l\le m}\in\mathbb{R}^{m\times m}$ such that:

$$\bar{D}=\{z\in\mathbb{C}:\mathbf{f}_{\bar{D}}(z)<0\} \tag{2.3.1}$$

with:

$$\mathbf{f}_{\bar{D}}(z):=\boldsymbol{\alpha}+z\boldsymbol{\beta}+\bar{z}\boldsymbol{\beta}^T=[\alpha_{kl}+\beta_{kl}z+\beta_{lk}\bar{z}]_{1\le k,l\le m} \tag{2.3.2}$$

In other words, an LMI region is a subset of the complex plane that is representable by an LMI in z and $\bar{z}$. As a result, LMI regions are convex and symmetric with respect to the real axis.
Using Gutman's theorem for LMI regions (**[Gutman, 1981]**), pole location in a given LMI region can be characterized in terms of the $m \times m$ block matrix

$$\mathbf{M}_{\bar{D}}(\mathbf{A},\mathbf{X}) := \boldsymbol{\alpha} \otimes \mathbf{X} + \boldsymbol{\beta} \otimes (\mathbf{AX}) - \boldsymbol{\beta}^T \otimes (\mathbf{AX})^T = \left[\alpha_{kl}\mathbf{X} + \beta_{kl}\mathbf{AX} + \beta_{lk}\mathbf{XA}^T\right]_{1\le k,l\le m} \tag{2.3.3}$$

as follows.

Theorem: The matrix $\mathbf{A}$ is $\bar{D}$-stable if and only if there exists a symmetric matrix $\mathbf{X}$ such that:

$$\begin{cases} \mathbf{M}_{\bar{D}}(\mathbf{A},\mathbf{X}) < 0 \\ \mathbf{X} > 0 \end{cases} \tag{2.3.4}$$

$\mathbf{M}_{\bar{D}}(\mathbf{A},\mathbf{X})$ in (2.3.3) and $\mathbf{f}_{\bar{D}}(z)$ in (2.3.2) are related by the substitution $(\mathbf{X},\mathbf{AX},\mathbf{XA}^T) \leftrightarrow (1,z,\bar{z})$.

The LMI regions which are of some interest for the design of the LPV controller/observer are the disk of radius r and center $(-q,0)$ and the vertical strip consisting of complex numbers $w = x + jy$ such that $x_{\min} < x < x_{\max}$.

The disk of radius r and center $(-q,0)$ is an LMI region with characteristic function:

$$\mathbf{f}_{\bar{D}}(z) = \begin{pmatrix} -r & q+z \\ q+\bar{z} & -r \end{pmatrix} \tag{2.3.5}$$

In this case (2.3.4) leads to:

$$\begin{cases} \begin{pmatrix} -r\mathbf{X} & q\mathbf{X}+\mathbf{AX} \\ q\mathbf{X}+\mathbf{XA}^T & -r\mathbf{X} \end{pmatrix} < 0 \\ \mathbf{X} > 0 \end{cases} \tag{2.3.6}$$

The vertical strip consisting of complex numbers $w = x + jy$ such that $x_{\min} < x < x_{\max}$ is an LMI region with characteristic function:

$$\mathbf{f}_{\bar{D}}(z) = \begin{pmatrix} x_{\min} - z/2 - \bar{z}/2 & 0 \\ 0 & -x_{\max} + z/2 + \bar{z}/2 \end{pmatrix} \tag{2.3.7}$$

In this case (2.3.4) leads to:

$$\begin{cases} \begin{pmatrix} x_{\min}\mathbf{X} - \frac{1}{2}(\mathbf{AX}+\mathbf{XA}^T) & 0 \\ 0 & -x_{\max}\mathbf{X} + \frac{1}{2}(\mathbf{AX}+\mathbf{XA}^T) \end{pmatrix} < 0 \\ \mathbf{X} > 0 \end{cases} \tag{2.3.8}$$

The LMI describing the region of the complex plane obtained through intersection of the circle and the vertical strip can be obtained easily, because the class of LMI regions is invariant under set intersection. Specifically, given two LMI regions $\bar{D}_1$ and $\bar{D}_2$ and their associated characteristic functions $\mathbf{f}_{\bar{D}_1}$ and $\mathbf{f}_{\bar{D}_2}$, the intersection $\bar{D} = \bar{D}_1 \cap \bar{D}_2$ is also an LMI region with characteristic function:

$$\mathbf{f}_{\bar{D}_1 \cap \bar{D}_2} = diag\left(\mathbf{f}_{\bar{D}_1}, \mathbf{f}_{\bar{D}_2}\right) \tag{2.3.9}$$

A consequence of this intersection property is that simultaneous clustering constraints can be expressed as a system of LMIs in the same variable $\mathbf{X}$ without introducing any conservatism. Given two LMI regions $\bar{D}_1$ and $\bar{D}_2$, a matrix $\mathbf{A}$ is both $\bar{D}_1$-stable and $\bar{D}_2$-stable if and only if there exists a positive definite matrix $\mathbf{X}$ such that $\mathbf{M}_{\bar{D}_1}(\mathbf{A},\mathbf{X})<0$ and $\mathbf{M}_{\bar{D}_2}(\mathbf{A},\mathbf{X})<0$.

Consider a linear time-invariant (LTI) system described by:

$$\begin{cases}\mathbf{x}(k+1)=\mathbf{A}\mathbf{x}(k)+\mathbf{B}\mathbf{u}(k)\\ \mathbf{y}(k)=\mathbf{C}\mathbf{x}(k)\end{cases} \tag{2.3.10}$$

under state-feedback control law $\mathbf{u}(k)=\mathbf{K}\mathbf{x}(k)$. The problem to be solved consists of finding a state-feedback gain $\mathbf{K}$ that places the closed-loop poles in some LMI stability region $\bar{D}$ with characteristic function (2.3.2).

The pole-placement constraint is satisfied if and only if there exists $\mathbf{X}_{\bar{D}}>0$ such that:

$$\left[\alpha_{kl}\mathbf{X}_{\bar{D}}+\beta_{kl}(\mathbf{A}+\mathbf{B}\mathbf{K})\mathbf{X}_{\bar{D}}+\beta_{lk}\mathbf{X}_{\bar{D}}(\mathbf{A}+\mathbf{B}\mathbf{K})^T\right]_{1\le k,l\le m}<0 \tag{2.3.11}$$

By means of the auxiliary variable $\mathbf{\Gamma}:=\mathbf{K}\mathbf{X}$ the matrix inequality (2.3.11) becomes an LMI that can be solved through convex optimization techniques:

$$\begin{cases}\left[\alpha_{kl}\mathbf{X}+\beta_{kl}\mathbf{U}(\mathbf{X},\mathbf{\Gamma})+\beta_{lk}\mathbf{U}(\mathbf{X},\mathbf{\Gamma})^T\right]<0\\ \mathbf{U}(\mathbf{X},\mathbf{\Gamma}):=\mathbf{A}\mathbf{X}+\mathbf{B}\mathbf{\Gamma}\end{cases} \tag{2.3.12}$$

This result can be easily extended to systems described by a polytopic state-space model with the structure of (2.2.2). The problem consists in computing state-feedback gains $\mathbf{K}_j$ and a single Lyapunov matrix $\mathbf{X}>0$ such that $\mathbf{M}_{\bar{D}}(\mathbf{A}_j+\mathbf{B}\mathbf{K}_j,\mathbf{X})<0$ for $j=1,...,N$. The use of a single Lyapunov function over the entire operating range guarantees $\bar{D}$-stability for each couple of state-space matrices $(\mathbf{A},\mathbf{B})$ in the polytope (2.2.2).

Summarising, let $\bar{D}$ be any LMI region, and suppose that the LPV system is quadratically $\bar{D}$-stabilizable with Lyapunov matrix $\mathbf{X}$ and state-feedback gains $\mathbf{K}_j$, and let $\mathbf{\Gamma}_j=\mathbf{K}_j\mathbf{X}$. Writing the condition $\mathbf{M}_{\bar{D}}(\mathbf{A}+\mathbf{B}\mathbf{K},\mathbf{X})<0$ at each vertex $(\mathbf{A}_j,\mathbf{B})$ of the polytope (2.2.2) yields the following necessary conditions on $\mathbf{X}$, $\mathbf{\Gamma}_j$:

$$\begin{cases}\left[\alpha_{kl}\mathbf{X}+\beta_{kl}(\mathbf{A}_j\mathbf{X}+\mathbf{B}\mathbf{\Gamma}_j)+\beta_{lk}(\mathbf{A}_j\mathbf{X}+\mathbf{B}\mathbf{\Gamma}_j)^T\right]_{k,l}<0\\ \mathbf{X}>0\end{cases} \tag{2.3.13}$$

for $j=1,...,N$.

Conditions (2.3.13) are necessary and sufficient for quadratic *D*-stabilizability.

This result can be used for the design of an LPV observer thanks to the duality property.

Consider the linear time-invariant (LTI) system with observer described by

$$\begin{cases}\mathbf{x}(k+1)=\mathbf{A}x(k)+\mathbf{B}u(k)\\ \mathbf{y}(k)=\mathbf{C}\mathbf{x}(k)\\ \hat{\mathbf{x}}(k+1)=\mathbf{A}\hat{\mathbf{x}}(k)+\mathbf{B}\mathbf{u}(k)+\mathbf{L}(\hat{\mathbf{y}}(k)-\mathbf{y}(k))\\ \hat{\mathbf{y}}(k)=\mathbf{C}\hat{\mathbf{x}}(k)\end{cases} \tag{2.3.14}$$

The problem to be solved consists of finding an observer gain $\mathbf{L}$ that places the poles of the observer in some LMI stability region $\bar{D}$ with characteristic function (2.3.2).

This is equivalent to the problem of finding a state-feedback gain $\mathbf{L}^T$ that places the poles of the closed-loop dual system in $\bar{D}$:

$$\begin{cases} \mathbf{z}(k+1) = \mathbf{A}^T \mathbf{z}(k) + \mathbf{C}^T \mathbf{v}(k) \\ \mathbf{w}(k) = \mathbf{B}^T \mathbf{z}(k) \end{cases} \tag{2.3.15}$$

under the state-feedback control law $\mathbf{v}(k) = \mathbf{L}^T \mathbf{z}(k)$. In other words, $\bar{D}$-detectability of the pair $(\mathbf{A}, \mathbf{C})$ is equivalent to $\bar{D}$-stabilizability of $(\mathbf{A}^T, \mathbf{C}^T)$.

2.4 – *Concepts and Basic Principles of Fault Tolerant Control*

A Fault Tolerant Control (FTC) system is a control system designed to retain some portion of its control integrity in the event of a fault. It is based on some kind of automatic reconfiguration made once a malfunction has been detected and isolated. Because of the need of the fault to be detected and isolated, FDI (Fault Detection and Isolation) plays an important role in the FTC. The main advantage of FTC over other measures for fault tolerance is the fact that FTC makes "intelligent" use of the redundancies included in the system.

There are two ways to approach a fault in FTC, so methods are usually divided into two classes: passive (PFTC) and active (AFTC). PFTC is based on robust controller design techniques and aims at synthesizing one robust controller that makes the closed-loop system insensitive to certain faults. This approach does not need FDI and is computationally simpler, even though the price to pay is that there is a loss of performance in the system behaviour when there is no fault. On the other hand, the design of AFTC is based on controller redesign, or selecting/mixing of pre-designed controllers. It needs the presence of an FDI scheme, responsible for detecting, localizing and estimating on-line the fault severity.

This thesis will focus on two techniques: AFTC using virtual sensors and AFTC using virtual actuators. The approaches are similar techniques that have been developed for LTI systems, even though hereafter the theory will be adapted to LPV models in order to be applied to a non-linear system, that is the Twin Rotor MIMO System.

For further information about the field of FTC, the reader could find them in **[Blanke, 2003]**, **[Patton, 1997]** and **[Zhang, 2008]**.

2.4.1 – Control reconfiguration for actuator or sensor failures

Both the virtual sensor technique and the virtual actuator technique belong to the FTC class of techniques based on control reconfiguration. When faults appear either in the actuators or in the sensors, the control loops brought about by the nominal controllers are modified. The goal of the reconfiguration is to stabilise the faulty process and to keep it operational with sufficient performance. In particular, instead of adapting the controller to the faulty plant, a reconfiguration block is used to adapt the faulty plant to the nominal controller. The faulty plant together with the reconfiguration block should produce, for a given input the same (or approximately the same) output as the nominal plant. Hence, the controller "sees" the same plant as before the fault and reacts in the same way as before. In this way, the nominal controller remains an unchanged block of the control loop.

In case of a sensor break-down, the reconfiguration block results from the application of a Luenberger observer to reconstruct the faulty output. It is called a "virtual sensor", because it reconstructs that element y_i of the output vector from the other measured output signals that the faulty sensor does no longer measure.

If an actuator becomes faulty, the reconfiguration block is obtained in a dual way. The reconfiguration block is called a "virtual actuator", because it acts like the faulty actuator but replaces the effect of this actuator by using the control input of the other actuators appropriately.

The reconfigured controller, which is to be applied to the faulty plant, consists of the nominal controller and the reconfiguration block.
The overall idea is presented in ***Fig. 1***.

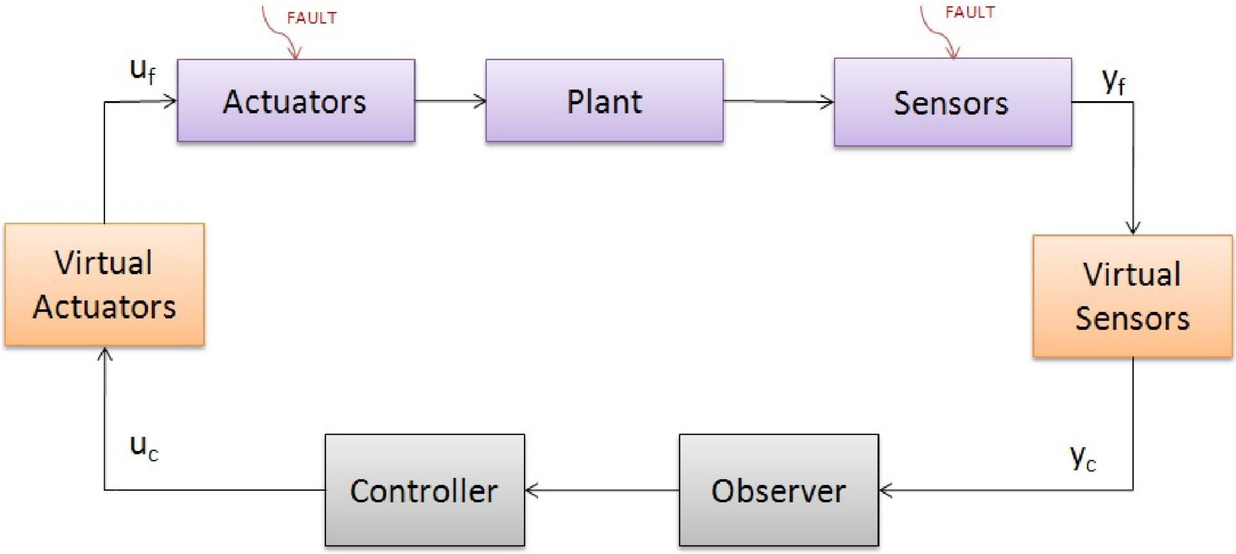

Fig. 1 - Virtual actuators and virtual sensors

3 – TRMS MODELLING

3.1 – The Twin-Rotor MIMO System

The twin-rotor multiple input-multiple output (MIMO) system (TRMS) is a laboratory set-up developed by Feedback Instruments Limited **[Feedback, 33-007-4M5]** for control experiments. The system is perceived as a challenging engineering problem owing to its high non-linearity, cross-coupling between its two axes, and inaccessibility of some of its states and outputs for measurements. An accurate dynamic model of the system is thus required to achieve control objectives satisfactorily.

The TRMS is an aero-dynamical system similar to a helicopter. At both ends of its beam there are two propellers driven by DC motors, each perpendicular to the other one. The beam can rotate freely in the horizontal and vertical planes, in such a way that its ends move on spherical surfaces. The joined beam can be moved by changing the motors supply voltages, thus controlling the rotational speed of the propellers. There is a counter-weight fixed to the beam which is used for balancing the angular momentum in a stable equilibrium position.

The rotor generating the vertical movement is called the main rotor. It enables the TRMS to pitch, which is a rotation in the vertical plane around the horizontal axes. The rotor generating the horizontal movement is called the tail rotor. It enables the TRMS to yaw, which is a rotation in the horizontal plane around the vertical axes.

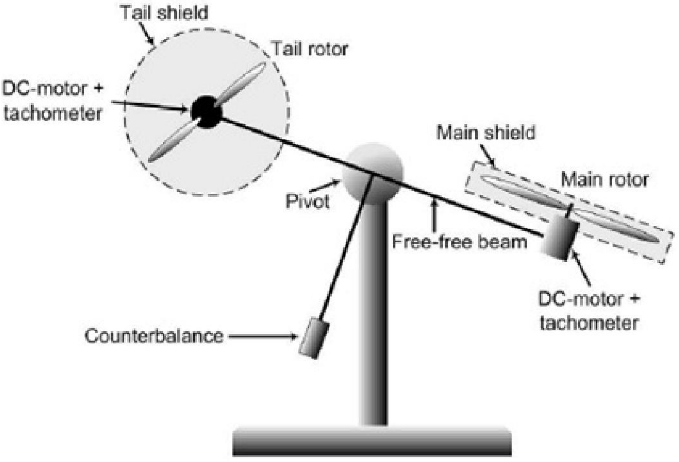

Fig. 2 – The Twin Rotor MIMO System

3.2 – Non-Linear Model of the TRMS

In this section a mathematical model of the TRMS is described. The crude dynamic model of the system supplied by the manufacturer does not represent the system dynamic precisely, as all the effective forces are not taken into consideration. Accurate models are proposed in **[Rahideh, 2007]**, **[Gabriel, 2008]**, **[López, 1998]** and **[Christensen, 2006]**. Each of these models contemplates dynamical effects that have not been considered by the manufacturer, and leads to a set of non-linear differential equations.

The strategy to describe the TRMS is to split the system into simpler subsystems: the DC-motors, the propellers and the beam. The first two subsystems have independent dynamics, that is, the main motor does not affect the behaviour of the tail motor, and vice versa. The same is true for the propellers. On the other hand, the dynamics of the beam is strongly non-linear with the presence of interaction phenomenons between the horizontal and the vertical dynamics.

3.2.1 – DC-Motors

The TRMS possesses two permanent magnet DC motors, one for the main propeller and the other for the tail. The motors are considered to be identical, though they work with different loads since the propellers are of unequal sizes.

The electric relationship between the applied motor voltages u_h and u_v, the angular velocities of the rotors ω_h and ω_v, and the armature currents of the motors i_{ah} and i_{av} can be described by the following differential equation:

$$L_a \frac{di_{ah}}{dt} = -R_a i_{ah} - k_a \omega_h + k_1 u_h \tag{3.2.1}$$

$$L_a \frac{di_{av}}{dt} = -R_a i_{av} - k_a \omega_v + k_2 u_v \tag{3.2.2}$$

The resistance R_a is assumed to be constant, thus neglecting a non-linear phenomenon due to a non-linear voltage drop across the carbon brushes of the commutator of the DC motors, that would be better approximated using a piecewise constant resistance, as proposed by Knudsen and Jensen in **[Knudsen, 1995]**.

Moreover, the term $L_a \, di_a / dt$ could be disregarded, because the time constant of the electrical part is small compared to that of the mechanical part. As a result the dynamics of the system is not altered significantly by omitting the inductance of the motor. This leads to the following equations:

$$R_a i_{ah} + k_a \omega_h - k_1 u_h = 0 \tag{3.2.3}$$

$$R_a i_{av} + k_a \omega_v - k_2 u_v = 0 \tag{3.2.4}$$

3.2.2 – Propellers

The propellers are each hard coupled to the motors so that the composed moments of inertia J_{tr} and J_{mr} are obtained by combining the moments of inertia of both the propeller and the motor.

There are four types of frictions affecting the rotation of the propeller (***Fig. 3***). These frictions can be characterized as **[Andersen, 2005]**:

- The viscous friction τ_B, proportional to the angular speed ω;
- The Coloumb friction τ_c, constant, which depends on the sign of ω when rotating. If the motor torque applied is less than τ_n, the Coloumb friction τ_c is equal to the negative of the applied torque;
- The stiction τ_s only occurs when ω is zero (or very close to zero);
- The drag friction τ_D is due to the air resistance experienced by the rotor, and is proportional to the square of the angular speed ω.

The non-linear Coloumb friction τ_c and stiction τ_s can be neglected, thus obtaining a simpler model of the propeller that fits almost in the same way the real behaviour, even though it differs when the angular speed approaches 0 rad/s. Around 0 rad/s the lack of stiction or Coulomb friction in the models is seen, but since the working points of the rotors will not be close to 0 rad/s, the resulting error will be very small.

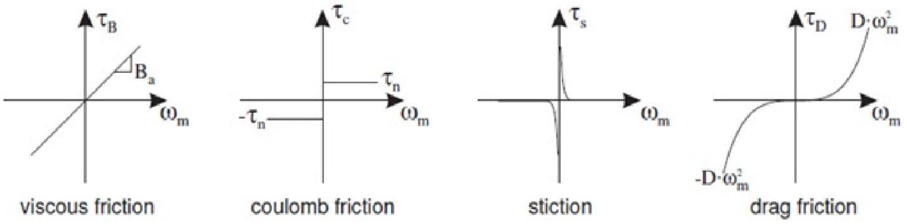

Fig. 3 - The four types of frictions affecting the rotation of the motor

Under these assumptions, the mechanical behaviour of the propellers can be described by the following first-order non-linear differential equations:

$$J_{tr}\frac{d\omega_h}{dt} = k_a i_{ah} - B_{tr}\omega_h - f_1(\omega_h) \tag{3.2.5}$$

$$J_{mr}\frac{d\omega_v}{dt} = k_a i_{av} - B_{mr}\omega_v - f_4(\omega_v) \tag{3.2.6}$$

where:

$$f_1(\omega_h) = \begin{cases} k_{thp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{thn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases} \tag{3.2.7}$$

$$f_4(\omega_v) = \begin{cases} k_{tvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{tvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases} \tag{3.2.8}$$

By disregarding the electrical dynamics, and taking into account (3.2.3) and (3.2.4), equations (3.2.5) and (3.2.6) become:

$$J_{tr}\frac{d\omega_h}{dt} = \frac{k_a k_1}{R_a}u_h - \left(\frac{k_a^2}{R_a} + B_{tr}\right)\omega_h - f_1\left(\omega_h\right) \tag{3.2.9}$$

$$J_{mr}\frac{d\omega_v}{dt} = \frac{k_a k_2}{R_a}u_v - \left(\frac{k_a^2}{R_a} + B_{mr}\right)\omega_v - f_4\left(\omega_v\right) \tag{3.2.10}$$

3.2.3 – Horizontal dynamics

The horizontal dynamics of the TRMS beam can be described by the following equations:

$$\begin{aligned}\frac{d\Omega_h}{dt} &= \frac{l_t f_2\left(\omega_h\right)\cos\theta_v - k_{oh}\Omega_h - f_3\left(\theta_h\right) + f_6\left(\theta_v\right)}{D\cos^2\theta_v + E\sin^2\theta_v + F} \\ &+ \frac{k_m\omega_v\sin\theta_v\Omega_v\left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v\right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right)^2} \\ &+ \frac{k_m\cos\theta_v\left(\frac{k_a k_2}{R_a}u_v - \left(B_{mr} + \frac{k_a^2}{R_a}\right)\omega_v - f_4\left(\omega_v\right)\right)}{J_{mr}\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right)}\end{aligned} \tag{3.2.11}$$

$$\frac{d\theta_h}{dt} = \Omega_h \tag{3.2.12}$$

where:

$$\theta_v^0 = \arctan\left(\frac{A-B}{C}\right) \tag{3.2.13}$$

$$A = \left(\frac{m_t}{2} + m_{tr} + m_{ts}\right) l_t \tag{3.2.14}$$

$$B = \left(\frac{m_m}{2} + m_{mr} + m_{ms}\right) l_m \tag{3.2.15}$$

$$C = \left(\frac{m_b}{2} l_b + m_{cb} l_{cb}\right) \tag{3.2.16}$$

$$D = \left(\frac{m_m}{3} + m_{mr} + m_{ms}\right) l_m^2 + \left(\frac{m_t}{3} + m_{tr} + m_{ts}\right) l_t^2 \tag{3.2.17}$$

$$E = \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2 \tag{3.2.18}$$

$$F = m_{ms} r_{ms}^2 + \frac{m_{ts}}{2} r_{ts}^2 \tag{3.2.19}$$

$$f_2\left(\omega_h\right) = \begin{cases} k_{fhp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{fhn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases} \tag{3.2.20}$$

$$f_3(\theta_h) = \begin{cases} k_{chp}\theta_h & \text{if } \theta_h \geq 0 \\ k_{chn}\theta_h & \text{if } \theta_h < 0 \end{cases} \tag{3.2.21}$$

$$f_6\left(\theta_v\right) = \begin{cases} k_{cvp}\left(\theta_v - \theta_v^0\right)^2 & \text{if } \theta_v \geq \theta_v^0 \\ k_{cvn}\left(\theta_v - \theta_v^0\right)^2 & \text{if } \theta_v < \theta_v^0 \end{cases} \tag{3.2.22}$$

and where:

- $l_t f_2\left(\omega_h\right)\cos\theta_v$ is the propulsive torque due to the tail propeller rotation;
- $k_{oh}\Omega_h$ is the torque due to the viscous friction force, in the horizontal plane;
- $f_3\left(\theta_h\right)$ is the torque due to the cables connecting the sensors and the motors. Though they are lightweighted and hang relatively freely, there is a torsion force in the horizontal axis that has to be considered. Although this torque is non-linear, a linear approximation provides a satisfactory level of accuracy;
- $f_6\left(\theta_v\right)$ takes into account a coupling phenomenon that was not described by the model defined in **[Rahideh, 2007]**. Such a phenomenon appeared during the TRMS identification (see next chapter).
- θ_v^0 is the equilibrium point for the vertical angle corresponding to $u_v = 0$;
- $D\cos^2\theta_v + E\sin^2\theta_v + F$ is the horizontal moment of inertia. It can be calculated as the sum of the following components:

$$J_{h1} = \frac{m_m}{3}\left(l_m \cos\theta_v\right)^2 \qquad \text{(beam of main rotor)} \tag{3.2.23}$$

$$J_{h2} = \frac{m_t}{3}\left(l_t \cos\theta_v\right)^2 \qquad \text{(beam of tail rotor)} \tag{3.2.24}$$

$$J_{h3} = \frac{m_b}{3}\left(l_b \sin\theta_v\right)^2 \qquad \text{(beam of counterweight)} \tag{3.2.25}$$

$$J_{h4} = m_{tr}\left(l_t \cos\theta_v\right)^2 \qquad \text{(tail rotor)} \tag{3.2.26}$$

$$J_{h5} = m_{mr}\left(l_m \cos\theta_v\right)^2 \qquad \text{(main rotor)} \tag{3.2.27}$$

$$J_{h6} = m_{cb}\left(l_{cb} \cos\theta_v\right)^2 \qquad \text{(counterweight)} \tag{3.2.28}$$

$$J_{h7} = \frac{m_{ts}}{2} r_{ts}^2 + m_{ts}\left(l_t \cos\theta_v\right)^2 \qquad \text{(shield tail rotor)} \tag{3.2.29}$$

$$J_{h8} = m_{ms} r_{ms}^2 + m_{ms}\left(l_m \cos\theta_v\right)^2 \qquad \text{(shield main rotor)} \tag{3.2.30}$$

The remaining terms of the equation are due to phenomenona of electro-mechanical coupling, because the engines driving the horizontal and the vertical movement, respectively affect the other direction.

3.2.4 – Vertical dynamics

The vertical dynamics of the TRMS beam can be described by the following equations:

$$\begin{aligned}\frac{d\Omega_v}{dt} = {} & \frac{l_m f_5\left(\omega_v\right) + k_g \Omega_h f_5\left(\omega_v\right)\cos\theta_v - k_{ov}\Omega_v - \Omega_h^2 H \sin\theta_v \cos\theta_v}{J_v} \\ & + \frac{g\left(\left(A-B\right)\cos\theta_v - C\sin\theta_v\right)}{J_v} + \frac{k_t\left(\frac{k_a k_1}{R_a} u_h - \left(B_{tr} + \frac{k_a^2}{R_a}\right)\omega_h - f_1(\omega_h)\right)}{J_v J_{tr}}\end{aligned} \tag{3.2.31}$$

$$\frac{d\theta_v}{dt} = \Omega_v \tag{3.2.32}$$

where:

$$H = A l_t + B l_m + \frac{m_b}{2} l_b^2 + m_{cb} l_{cb}^2 \tag{3.2.33}$$

$$f_5\left(\omega_v\right) = \begin{cases} k_{fvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{fvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases} \tag{3.2.34}$$

- $l_m f_5\left(\omega_v\right)$ is the propulsive torque due to the main propeller rotation;
- $k_g \Omega_h f_5\left(\omega_v\right)\cos\theta_v$ is the torque of the gyroscopic force;
- $k_{ov}\Omega_v$ is the torque due to the viscous friction force in the vertical plane;
- $g\left(\left(A-B\right)\cos\theta_v - C\sin\theta_v\right)$ is the torque due to the gravity force that acts perpendicularly to the earth's surface, thus affecting only the angle θ_v;
- $\Omega_h^2 H \sin\theta_v \cos\theta_v$ is the torque due to the centrifugal force;
- $k_t\left(k_a i_{ah} - B_{tr}\omega_h - f_1\left(\omega_h\right)\right)$ is due to an electro-mechanical coupling between the engines driving the horizontal and the vertical movement;
- J_v is the vertical moment of inertia. It can be calculated as the sum of the following components:

$$J_{v1} = m_{mr} l_m^2 \qquad \text{(main rotor)} \tag{3.2.35}$$

$$J_{v2} = m_m \frac{l_m^2}{3}$$ (beam of main rotor) (3.2.36)

$$J_{v3} = m_{cb} l_{cb}^2$$ (counterweight) (3.2.37)

$$J_{v4} = m_b \frac{l_b^2}{3}$$ (beam of counterweight) (3.2.38)

$$J_{v5} = m_{tr} l_t^2$$ (tail rotor) (3.2.39)

$$J_{v6} = m_t \frac{l_t^2}{3}$$ (beam of tail rotor) (3.2.40)

$$J_{v7} = \frac{m_{ms}}{2} r_{ms}^2 + m_{ms} l_m^2$$ (shield main rotor) (3.2.41)

$$J_{v8} = \frac{m_{ts}}{2} r_{ts}^2 + m_{ts} l_t^2$$ (shield tail rotor) (3.2.42)

3.2.5 – The overall model

The TRMS mathematical model can be expressed as a non-linear state space model, where the input vector is $\mathbf{u} = \begin{bmatrix} u_h & u_v \end{bmatrix}^T$, with u_h the input voltage of the tail motor and u_v the input voltage of the main motor. The system state is the column vector $\mathbf{x} = \begin{bmatrix} \omega_h & \Omega_h & \theta_h & \omega_v & \Omega_v & \theta_v \end{bmatrix}^T$. ω_h is the rotational velocity of the tail rotor, ω_v is the rotational velocity of the main rotor, Ω_h is the angular velocity around the horizontal axis, Ω_v is the angular velocity around the vertical axis, θ_h is the yaw angle of the beam and θ_v is the pitch angle of the beam.

The mathematical model of the TRMS becomes the set of 6 non-linear differential equations:

$$\frac{d\omega_h}{dt} = \frac{k_a k_1}{J_{tr} R_a} u_h - \left(\frac{B_{tr}}{J_{tr}} + \frac{k_a^2}{J_{tr} R_a} \right) \omega_h - \frac{f_1(\omega_h)}{J_{tr}} \tag{3.2.43}$$

$$\begin{aligned} \frac{d\Omega_h}{dt} &= \frac{l_t f_2(\omega_h)\cos\theta_v - k_{oh}\Omega_h - f_3(\theta_h) + f_6(\theta_v)}{D\cos^2\theta_v + E\sin^2\theta_v + F} \\ &+ \frac{k_m \omega_v \sin\theta_v \Omega_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v \right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F \right)^2} \\ &+ \frac{k_m \cos\theta_v \left(\frac{k_a k_2}{R_a} u_v - \left(B_{mr} + \frac{k_a^2}{R_a} \right) \omega_v - f_4(\omega_v) \right)}{J_{mr} \left(D\cos^2\theta_v + E\sin^2\theta_v + F \right)} \end{aligned} \tag{3.2.44}$$

$$\frac{d\theta_h}{dt} = \Omega_h \tag{3.2.45}$$

$$\frac{d\omega_v}{dt} = \frac{k_a k_2}{J_{mr} R_a} u_v - \left(\frac{B_{mr}}{J_{mr}} + \frac{k_a^2}{J_{mr} R_a} \right) \omega_v - \frac{f_4(\omega_v)}{J_{mr}} \tag{3.2.46}$$

$$\frac{d\Omega_v}{dt} = \frac{l_m f_5\left(\omega_v\right) + k_g \Omega_h f_5\left(\omega_v\right)\cos\theta_v - k_{ov}\Omega_v - \Omega_h^2 H \sin\theta_v \cos\theta_v}{J_v}$$

$$+ \frac{g\left(\left(A-B\right)\cos\theta_v - C\sin\theta_v\right)}{J_v} + \frac{k_t\left(\frac{k_a k_1}{R_a}u_h - \left(B_{tr} + \frac{k_a^2}{R_a}\right)\omega_h - f_1(\omega_h)\right)}{J_v J_{tr}} \tag{3.2.47}$$

$$\frac{d\theta_v}{dt} = \Omega_v \tag{3.2.48}$$

where constants $A,\ B,\ C,\ D,\ E,\ F,\ H$ are defined as in (3.2.14), (3.2.15), (3.2.16), (3.2.17), (3.2.18), (3.2.19), (3.2.33) and:

$$J_v = \left(\frac{m_t}{3} + m_{tr} + m_{ts}\right) l_t^2 + \left(\frac{m_m}{3} + m_{mr} + m_{ms}\right) l_m^2 + \frac{m_{ms}}{2} r_{ms}^2 + m_{ts} r_{ts}^2 + \frac{m_b}{3} l_b^2 + m_{cb} l_{cb}^2 \tag{3.2.49}$$

and the functions f_i are defined as in (3.2.7), (3.2.20), (3.2.21), (3.2.8) and (3.2.34).

3.3 – TRMS LPV Representation

In order to design an LPV controller and an LPV observer for the Twin Rotor MIMO System, an LPV mathematical model is required. LPV representations of given nonlinear or parameter dependent plants are usually not unique, and different models lead to different achievable performance when LMI techniques are used to design LPV controllers and observers. The non-linear model that has been described in Section 3.2.5 can be expressed in an LPV absolute form according to the procedure described in Section 2.2.3. The LPV absolute representation can be used to obtain another LPV model based on state transformation, according to the procedure described in Section 2.2.4.

3.3.1 – TRMS Absolute representation

The non-linear model of TRMS can be transformed in the following LPV representation:

$$\begin{bmatrix} \dot{\omega}_h(t) \\ \dot{\Omega}_h(t) \\ \dot{\theta}_h(t) \\ \dot{\omega}_v(t) \\ \dot{\Omega}_v(t) \\ \dot{\theta}_v(t) \end{bmatrix} = \mathbf{A}_{abs}(\mathbf{x}(t)) \begin{bmatrix} \omega_h(t) \\ \Omega_h(t) \\ \theta_h(t) \\ \omega_v(t) \\ \Omega_v(t) \\ \theta_v(t) - \theta_v^0 \end{bmatrix} + \mathbf{B}_{abs}(\mathbf{x}(t)) \begin{bmatrix} u_h(t) \\ u_v(t) \end{bmatrix} \tag{3.3.1}$$

where:

$$\mathbf{A}_{abs}(\mathbf{x}(t)) = \begin{pmatrix} a_{11}(\mathbf{x}(t)) & 0 & 0 & 0 & 0 & 0 \\ a_{21}(\mathbf{x}(t)) & a_{22}(\mathbf{x}(t)) & a_{23}(\mathbf{x}(t)) & a_{24}(\mathbf{x}(t)) & a_{25}(\mathbf{x}(t)) & a_{26}(\mathbf{x}(t)) \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44}(\mathbf{x}(t)) & 0 & 0 \\ a_{51}(\mathbf{x}(t)) & a_{52}(\mathbf{x}(t)) & 0 & a_{54}(\mathbf{x}(t)) & a_{55} & a_{56}(\mathbf{x}(t)) \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.3.2}$$

$$\mathbf{B}_{abs}(\mathbf{x}(t)) = \begin{pmatrix} b_{11} & 0 \\ 0 & b_{22}(\mathbf{x}(t)) \\ 0 & 0 \\ 0 & b_{42} \\ b_{51} & 0 \\ 0 & 0 \end{pmatrix} \tag{3.3.3}$$

$$a_{11}(\mathbf{x}(t)) = -\frac{k_a^2/R_a + B_{tr} + f_1(\omega_h)/\omega_h}{J_{tr}} \tag{3.3.4}$$

$$a_{21}(\mathbf{x}(t)) = \frac{l_t \cos\theta_v \, f_2(\omega_h)/\omega_h}{D\cos^2\theta_v + E\sin^2\theta_v + F} \tag{3.3.5}$$

$$a_{22}(\mathbf{x}(t)) = \frac{-k_{oh}}{D\cos^2\theta_v + E\sin^2\theta_v + F} \tag{3.3.6}$$

$$a_{23}(\mathbf{x}(t)) = \frac{-f_3(\theta_h)/\theta_h}{D\cos^2\theta_v + E\sin^2\theta_v + F} \tag{3.3.7}$$

$$a_{24}(\mathbf{x}(t)) = -\frac{k_m \cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F}\left(\frac{k_a^2/R_a + B_{mr} + f_4(\omega_v)/\omega_v}{J_{mr}}\right) \tag{3.3.8}$$

$$a_{25}(\mathbf{x}(t)) = \frac{k_m\omega_v \sin\theta_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v\right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right)^2} \tag{3.3.9}$$

$$a_{26}(\mathbf{x}(t)) = \frac{f_6(\theta_v)/\left(\theta_v - \theta_v^0\right)}{D\cos^2\theta_v + E\sin^2\theta_v + F} \tag{3.3.10}$$

$$a_{44}(\mathbf{x}(t)) = -\frac{k_a^2/R_a + B_{mr} + f_4(\omega_v)/\omega_v}{J_{mr}} \tag{3.3.11}$$

$$a_{51}(\mathbf{x}(t)) = -\frac{k_t}{J_v}\left(\frac{k_a^2/R_a + B_{tr} + f_1(\omega_h)/\omega_h}{J_{tr}}\right) \tag{3.3.12}$$

$$a_{52}(\mathbf{x}(t)) = \frac{k_g \cos\theta_v \, f_5(\omega_v)/\omega_v - \Omega_h H \sin\theta_v \cos\theta_v}{J_v} \tag{3.3.13}$$

$$a_{54}(\mathbf{x}(t)) = \frac{l_m \, f_5(\omega_v)/\omega_v}{J_v} \tag{3.3.14}$$

$$a_{55} = -\frac{k_{ov}}{J_v} \tag{3.3.15}$$

$$a_{56}(\mathbf{x}(t)) = \frac{g\left(\left(A - B\right)\cos\theta_v - C\sin\theta_v\right)}{J_v\left(\theta_v - \theta_v^0\right)} \tag{3.3.16}$$

$$b_{11} = \frac{k_a k_1}{J_{tr} R_a} \tag{3.3.17}$$

$$b_{22}(\mathbf{x}(t)) = \frac{k_m \cos\theta_v k_a k_2}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right) R_a J_{mr}} \tag{3.3.18}$$

$$b_{42} = \frac{k_a k_2}{J_{mr} R_a} \tag{3.3.19}$$

$$b_{51} = \frac{k_a k_1 k_t}{J_v J_{tr} R_a} \tag{3.3.20}$$

A discrete-time LPV model can be obtained using Euler approximation with a sampling time T_s:

$$\begin{aligned}
\begin{bmatrix} \omega_h(k+1) \\ \Omega_h(k+1) \\ \theta_h(k+1) \\ \omega_v(k+1) \\ \Omega_v(k+1) \\ \theta_v(k+1) \end{bmatrix} &= \begin{pmatrix} 1+a_{11}T_s & 0 & 0 & 0 & 0 & 0 \\ a_{21}T_s & 1+a_{22}T_s & a_{23}T_s & a_{24}T_s & a_{25}T_s & a_{26}T_s \\ 0 & T_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+a_{44}T_s & 0 & 0 \\ a_{51}T_s & a_{52}T_s & 0 & a_{54}T_s & 1+a_{55}T_s & a_{56}T_s \\ 0 & 0 & 0 & 0 & T_s & 1 \end{pmatrix} \begin{bmatrix} \omega_h(k) \\ \Omega_h(k) \\ \theta_h(k) \\ \omega_v(k) \\ \Omega_v(k) \\ \theta_v(k) \end{bmatrix} + \mathbf{B}_{abs} T_s \begin{bmatrix} u_h(k) \\ u_v(k) \end{bmatrix} \\
&= \begin{pmatrix} a_{11}^{absd} & 0 & 0 & 0 & 0 & 0 \\ a_{21}^{absd} & a_{22}^{absd} & a_{23}^{absd} & a_{24}^{absd} & a_{25}^{absd} & a_{26}^{absd} \\ 0 & T_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44}^{absd} & 0 & 0 \\ a_{51}^{absd} & a_{52}^{absd} & 0 & a_{54}^{absd} & a_{55}^{absd} & a_{56}^{absd} \\ 0 & 0 & 0 & 0 & T_s & 1 \end{pmatrix} \begin{bmatrix} \omega_h(k) \\ \Omega_h(k) \\ \theta_h(k) \\ \omega_v(k) \\ \Omega_v(k) \\ \theta_v(k) \end{bmatrix} + \begin{pmatrix} b_{11}^{absd} & 0 \\ 0 & b_{22}^{absd} \\ 0 & 0 \\ 0 & b_{42}^{absd} \\ b_{51}^{absd} & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} u_h(k) \\ u_v(k) \end{bmatrix}
\end{aligned} \tag{3.3.21}$$

where the dependence of the elements of the matrices on the state vector has been omitted.

3.3.2 – TRMS Shamma Representation

An LPV model for the Twin Rotor MIMO System can be obtained through state transformation from (3.3.1) by considering:

$$\begin{aligned} \mathbf{x}_1 &= \left[\theta_h \quad \theta_v\right]^T \\ \mathbf{x}_2 &= \left[\omega_h \quad \Omega_h \quad \omega_v \quad \Omega_v\right]^T \end{aligned} \tag{3.3.22}$$

The first step is to obtain the functions $\mathbf{x}_{2eq}$ and $\mathbf{u}_{eq}$ as defined in (2.2.13), that corresponds, in this case, to calculate the equilibrium points for $\omega_h^{eq}(\theta_h,\theta_v)$, $\Omega_h^{eq}(\theta_h,\theta_v)$, $\omega_v^{eq}(\theta_h,\theta_v)$, $\Omega_v^{eq}(\theta_h,\theta_v)$, $u_h^{eq}(\theta_h,\theta_v)$, $u_v^{eq}(\theta_h,\theta_v)$ by solving the equation system:

$$\begin{cases} a_{11}\omega_h^{eq} + b_{11}u_h^{eq} = 0 \\ a_{21}\omega_h^{eq} + a_{22}\Omega_h^{eq} + a_{23}\theta_h + a_{24}\omega_v^{eq} + a_{25}\Omega_v^{eq} + a_{26}\theta_v^{eq} + b_{22}u_v^{eq} = 0 \\ \Omega_h^{eq} = 0 \\ a_{44}\omega_v^{eq} + b_{42}u_v^{eq} = 0 \\ a_{51}\omega_h^{eq} + a_{52}\Omega_h^{eq} + a_{54}\omega_v^{eq} + a_{55}\Omega_v^{eq} + a_{56}\theta_v + b_{51}u_h^{eq} = 0 \\ \Omega_v^{eq} = 0 \end{cases} \tag{3.3.23}$$

where $a_{11,}a_{21},\ldots,a_{56},b_{11},\ldots,b_{51}$ are defined from (3.3.4) to (3.3.20), and their dependence on $\mathbf{x}(t)$ has been omitted.

Solving (3.3.23) leads to:

$$
\begin{aligned}
\omega_h^{eq} &= -\frac{a_{23}}{a_{21}}\theta_h - \frac{a_{26}}{a_{21}}\theta_v + \left(\frac{a_{44}b_{22}}{a_{21}b_{42}} - \frac{a_{24}}{a_{21}}\right)\omega_v^{eq} \\
\Omega_h^{eq} &= 0 \\
\omega_v^{eq} &= \frac{-a_{23}a_{51}b_{11}b_{42}\theta_h + a_{11}b_{51}a_{23}b_{42}\theta_h + a_{56}b_{11}a_{21}b_{42}\theta_v - a_{26}a_{51}b_{11}b_{42}\theta_v + a_{11}a_{26}b_{42}b_{51}}{a_{24}a_{51}b_{11}b_{42} - a_{44}b_{22}a_{51}b_{11} - a_{54}b_{11}a_{21}b_{42} - a_{11}b_{51}a_{24}b_{42} + a_{11}b_{51}a_{44}b_{22}} \\
\Omega_v^{eq} &= 0 \\
u_h^{eq} &= -\frac{a_{11}}{b_{11}}\omega_h^{eq} \\
u_v^{eq} &= -\frac{a_{44}}{b_{42}}\omega_v^{eq}
\end{aligned}
\tag{3.3.24}
$$

or, equivalently:

$$
\begin{aligned}
\omega_h^{eq} &= \frac{f_3(\theta_h)\omega_h - f_6(\theta_v)\omega_h}{l_t \cos\theta_v f_2(\omega_h)} \\
\Omega_h^{eq} &= 0 \\
\omega_v^{eq} &= \frac{g\omega_v\left(C\sin\theta_v + \left(B-A\right)\cos\theta_v\right)}{l_m f_5(\omega_v)} \\
\Omega_v^{eq} &= 0 \\
u_h^{eq} &= \frac{k_a + R_a\left(B_{tr} + \left(f_1(\omega_h)/\omega_h\right)\right)/k_a}{k_1}\omega_h^{eq} \\
u_v^{eq} &= \frac{k_a + R_a\left(B_{mr} + \left(f_4(\omega_v)/\omega_v\right)\right)/k_a}{k_2}\omega_v^{eq}
\end{aligned}
\tag{3.3.25}
$$

By defining new states and inputs, a new LPV model of TRMS can be obtained through (2.2.19). Then, this model can be reshaped by changing the order of the state variables, obtaining:

$$
\begin{bmatrix} \dot{\widehat{\omega}}_h(t) \\ \dot{\Omega}_h(t) \\ \dot{\theta}_h(t) \\ \dot{\widehat{\omega}}_v(t) \\ \dot{\Omega}_v(t) \\ \dot{\theta}_v(t) \end{bmatrix}
=
\begin{pmatrix}
a_{11} & -\dfrac{d\omega_h^{eq}}{d\theta_h} & 0 & 0 & -\dfrac{d\omega_h^{eq}}{d\theta_v} & 0 \\
a_{21} & a_{22} & 0 & a_{24} & a_{25} & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & a_{44} & -\dfrac{d\omega_v^{eq}}{d\theta_v} & 0 \\
a_{51} & a_{52} & 0 & a_{54} & a_{55} & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\begin{bmatrix} \widehat{\omega}_h(t) \\ \Omega_h(t) \\ \theta_h(t) \\ \widehat{\omega}_v(t) \\ \Omega_v(t) \\ \theta_v(t) \end{bmatrix}
+ \mathbf{B}_{abs}(\mathbf{x}(t)) \begin{bmatrix} \widehat{u}_h(t) \\ \widehat{u}_v(t) \end{bmatrix}
\tag{3.3.26}
$$

where $\widehat{\omega}_{h/v}(t) = \omega_{h/v}(t) - \omega_{h/v}^{eq}(t)$ and $\widehat{u}_{h/v}(t) = u_{h/v}(t) - u_{h/v}^{eq}(t)$ and:

$$
\frac{d\omega_h^{eq}}{d\theta_h} = \begin{cases} \dfrac{k_{chp}\omega_h}{l_t\cos\theta_v f_2(\omega_h)} & if\ \theta_h \geq 0 \\[2ex] \dfrac{k_{chn}\omega_h}{l_t\cos\theta_v f_2(\omega_h)} & if\ \theta_h < 0 \end{cases}
\tag{3.3.27}
$$

$$\frac{d\omega_h^{eq}}{d\theta_v} = \begin{cases} \dfrac{\omega_h \left(f_3(\theta_h)\sin\theta_v - 2k_{cvp}\left(\theta_v - \theta_v^0\right)\cos\theta_v - k_{cvp}\left(\theta_v - \theta_v^0\right)^2 \sin\theta_v \right)}{l_t f_2(\omega_h)\cos^2\theta_v} & if\ \theta_v \geq 0 \\ \dfrac{\omega_h \left(f_3(\theta_h)\sin\theta_v - 2k_{cvn}\left(\theta_v - \theta_v^0\right)\cos\theta_v - k_{cvn}\left(\theta_v - \theta_v^0\right)^2 \sin\theta_v \right)}{l_t f_2(\omega_h)\cos^2\theta_v} & if\ \theta_v < 0 \end{cases} \tag{3.3.28}$$

$$\frac{d\omega_v^{eq}}{d\theta_v} = \frac{g\omega_v\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{l_m f_5(\omega_v)} \tag{3.3.29}$$

A discrete-time LPV model can be obtained using Euler approximation with a sampling time T_s:

$$\begin{bmatrix} \widehat{\omega}_h(k+1) \\ \Omega_h(k+1) \\ \theta_h(k+1) \\ \widehat{\omega}_v(k+1) \\ \Omega_v(k+1) \\ \theta_v(k+1) \end{bmatrix} = \begin{pmatrix} 1+a_{11}T_s & -\dfrac{d\omega_h^{eq}}{d\theta_h}T_s & 0 & 0 & -\dfrac{d\omega_h^{eq}}{d\theta_v}T_s & 0 \\ a_{21}T_s & 1+a_{22}T_s & 0 & a_{24}T_s & a_{25}T_s & 0 \\ 0 & T_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1+a_{44}T_s & -\dfrac{d\omega_v^{eq}}{d\theta_v}T_s & 0 \\ a_{51}T_s & a_{52}T_s & 0 & a_{54}T_s & 1+a_{55}T_s & 0 \\ 0 & 0 & 0 & 0 & T_s & 1 \end{pmatrix} \begin{bmatrix} \widehat{\omega}_h(k) \\ \Omega_h(k) \\ \theta_h(k) \\ \widehat{\omega}_v(k) \\ \Omega_v(k) \\ \theta_v(k) \end{bmatrix} + \mathbf{B}_{abs}(\mathbf{x}(k)) \begin{bmatrix} \widehat{u}_h(k) \\ \widehat{u}_v(k) \end{bmatrix}$$

$$= \begin{pmatrix} a_{11}^{shad}(\mathbf{x}_k) & a_{12}^{shad}(\mathbf{x}_k) & 0 & 0 & a_{15}^{shad}(\mathbf{x}_k) & 0 \\ a_{21}^{shad}(\mathbf{x}_k) & a_{22}^{shad}(\mathbf{x}_k) & 0 & a_{24}^{shad}(\mathbf{x}_k) & a_{25}^{shad}(\mathbf{x}_k) & 0 \\ 0 & T_s & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{44}^{shad}(\mathbf{x}_k) & a_{45}^{shad}(\mathbf{x}_k) & 0 \\ a_{51}^{shad}(\mathbf{x}_k) & a_{52}^{shad}(\mathbf{x}_k) & 0 & a_{54}^{shad}(\mathbf{x}_k) & a_{55}^{shad} & 0 \\ 0 & 0 & 0 & 0 & T_s & 1 \end{pmatrix} \begin{bmatrix} \widehat{\omega}_h(k) \\ \Omega_h(k) \\ \theta_h(k) \\ \widehat{\omega}_v(k) \\ \Omega_v(k) \\ \theta_v(k) \end{bmatrix} + \begin{pmatrix} b_{11}^{shad} & 0 \\ 0 & b_{22}^{shad}(\mathbf{x}_k) \\ 0 & 0 \\ 0 & b_{44}^{shad} \\ b_{51}^{shad} & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} \widehat{u}_h(k) \\ \widehat{u}_v(k) \end{bmatrix} \tag{3.3.30}$$

3.3.3 – Polytopic LPV Model of the TRMS

The procedure to obtain an *interval polytopic LPV model* for the Twin Rotor MIMO System is here described.

Each of the state variables that influence the elements of the state-space matrices (3.3.21) or (3.3.30) is assumed to take values in an interval.

These intervals are defined as:

$$\begin{aligned} \omega_h &\in \left[\underline{\omega}_h, \overline{\omega}_h\right] \\ \Omega_h &\in \left[\underline{\Omega}_h, \overline{\Omega}_h\right] \\ \theta_h &\in \left[\underline{\theta}_h, \overline{\theta}_h\right] \\ \omega_v &\in \left[\underline{\omega}_v, \overline{\omega}_v\right] \\ \theta_v &\in \left[\underline{\theta}_v, \overline{\theta}_v\right] \end{aligned}$$

For each of the scheduling parameters $a_{11}^d(\mathbf{x}_k), a_{12}^d(\mathbf{x}_k), \ldots, a_{55}^d(\mathbf{x}_k), b_{22}^d(\mathbf{x}_k)$ their minimum and maximum values are calculated as:

$$\underline{a}_{ij}^d = \min_{\omega_h, \Omega_h, \theta_h, \omega_v, \theta_v} a_{ij}^d(\mathbf{x}_k) \tag{3.3.31}$$

$$\overline{a}_{ij}^d = \max_{\omega_h, \Omega_h, \theta_h, \omega_v, \theta_v} a_{ij}^d(\mathbf{x}_k) \tag{3.3.32}$$

$$\underline{b}_{22}^{d} = \min_{\omega_h,\Omega_h,\theta_h,\omega_v,\theta_v} b_{22}^{d}(\mathbf{x}_k) \tag{3.3.33}$$

$$\overline{b}_{22}^{d} = \max_{\omega_h,\Omega_h,\theta_h,\omega_v,\theta_v} b_{22}^{d}(\mathbf{x}_k) \tag{3.3.34}$$

The scheduling parameter vector is assumed to take values in a hyperrectangle of $\mathbb{R}^n$ called the parameter box, where each parameter a_{ij}^d varies between the extreme values $\underline{\alpha}_{ij}^d$ and $\overline{\alpha}_{ij}^d$. When each of the scheduling parameters takes an extreme value, one of the vertex systems in (2.2.2) is generated.

The strong nonlinearity of some of the scheduling parameters makes difficult to find the exact optima over the entire range of values that the state space variables can take. In the following, explicit formulas to calculate $\underline{a}_{ij}^d$ and $\overline{a}_{ij}^d$, $\underline{b}_{22}^d$ and $\overline{b}_{22}^d$ will be given for both the so-called Absolute and the Shamma LPV Model. Notice that *min* and *max* denote that the optimum for the expression inside the brackets has to be found by means of numerical optimization methods. It also should be considered that some LPV parameters are not defined when $\omega_h = 0$, and that they take very high values when ω_h is near zero, so an additional limit on ω_h will be imposed:

$$\omega_h \in \left[\underline{\omega}_h, \omega_h^-\right] \cup \left[\omega_h^+, \overline{\omega}_h\right]$$

The same is true for ω_v, thus leading to:

$$\omega_v \in \left[\underline{\omega}_v, \omega_v^-\right] \cup \left[\omega_v^+, \overline{\omega}_v\right]$$

as well as for θ_v when its value is near to θ_v^0, leading to:

$$\theta_v \in \left[\underline{\theta}_v, \theta_v^-\right] \cup \left[\theta_v^+, \overline{\theta}_v\right]$$

3.3.3.1 – Polytopic Absolute LPV Model of the TRMS

$$a_{11}^{absd} = \begin{cases} 1 - \dfrac{T_s}{J_{tr}}\left(\dfrac{k_a^2}{R_a} + B_{tr} + k_{thp}\omega_h\right) & if\, \omega_h \geq 0 \\ 1 - \dfrac{T_s}{J_{tr}}\left(\dfrac{k_a^2}{R_a} + B_{tr} - k_{thn}\omega_h\right) & if\, \omega_h < 0 \end{cases} \tag{3.3.35}$$

$$\underline{a}_{11}^{absd} = 1 - \frac{T_s}{J_{tr}}\left(\frac{k_a^2}{R_a} + B_{tr} + \max\left\{k_{thp}\overline{\omega}_h, -k_{thn}\underline{\omega}_h\right\}\right) \tag{3.3.36}$$

$$\overline{a}_{11}^{absd} = 1 - \frac{T_s}{J_{tr}}\left(\frac{k_a^2}{R_a} + B_{tr}\right) \tag{3.3.37}$$

$$a_{21}^{absd} = \begin{cases} \dfrac{l_t T_s \cos\theta_v k_{fhp}\omega_h}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\, \omega_h \geq 0 \\ \dfrac{-l_t T_s \cos\theta_v k_{fhn}\omega_h}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\, \omega_h < 0 \end{cases} \tag{3.3.38}$$

$$\underline{a}_{21}^{absd} = l_t T_s \min\left\{k_{fhp}\omega_h^+, -k_{fhn}\omega_h^-\right\} \min_{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]} \left\{\frac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F}\right\} \tag{3.3.39}$$

$$\overline{a}_{21}^{absd} = l_t T_s \max\left\{k_{fhp}\overline{\omega}_h, -k_{fhn}\underline{\omega}_h\right\} \max_{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]} \left\{\frac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F}\right\} \tag{3.3.40}$$

$$a_{22}^{absd} = 1 - \frac{k_{oh} T_s}{D\cos^2\theta_v + E\sin^2\theta_v + F} \tag{3.3.41}$$

$$\underline{a}_{22}^{absd} = 1 - k_{oh} T_s \max_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{1}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.42}$$

$$\overline{a}_{22}^{absd} = 1 - k_{oh} T_s \min_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{1}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.43}$$

$$a_{23}^{absd} = \begin{cases} \dfrac{-k_{chp} T_s}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\, \theta_h \geq 0 \\ \dfrac{-k_{chn} T_s}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\, \theta_h < 0 \end{cases} \tag{3.3.44}$$

$$\underline{a}_{23}^{absd} = -T_s \max\left\{k_{chp}, k_{chn}\right\} \max_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{1}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.45}$$

$$\overline{a}_{23}^{absd} = -T_s \min\left\{k_{chp}, k_{chn}\right\} \min_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{1}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.46}$$

$$a_{24}^{absd} = \begin{cases} -k_m T_s \dfrac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F} \left(\dfrac{k_a^2 / R_a + B_{mr} + k_{tvp}\omega_v}{J_{mr}} \right) & if\, \omega_v \geq 0 \\ -k_m T_s \dfrac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F} \left(\dfrac{k_a^2 / R_a + B_{mr} - k_{tvn}\omega_v}{J_{mr}} \right) & if\, \omega_v < 0 \end{cases} \tag{3.3.47}$$

$$\underline{a}_{24}^{absd} = -\frac{k_m T_s}{J_{mr}} \left(\frac{k_a^2}{R_a} + B_{mr} + \max\left\{k_{tvp}\overline{\omega}_v, k_{tvn}\underline{\omega}_v\right\} \right) \max_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.48}$$

$$\overline{a}_{24}^{absd} = -\frac{k_m T_s}{J_{mr}} \left(\frac{k_a^2}{R_a} + B_{mr} \right) \min_{\theta_v \in [\underline{\theta}_v, \overline{\theta}_v]} \left\{ \frac{\cos\theta_v}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right\} \tag{3.3.49}$$

$$a_{25}^{absd} = \frac{T_s k_m \omega_v \sin\theta_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v \right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F \right)^2} \tag{3.3.50}$$

$$\underline{a}_{25}^{absd} = k_m T_s \min_{\substack{\omega_v \in [\underline{\omega}_v, \overline{\omega}_v] \\ \theta_v \in [\underline{\theta}_v, \overline{\theta}_v]}} \left\{ \frac{\omega_v \sin\theta_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v \right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F \right)^2} \right\} \tag{3.3.51}$$

$$\overline{a}_{25}^{absd} = k_m T_s \max_{\substack{\omega_v \in [\underline{\omega}_v, \overline{\omega}_v] \\ \theta_v \in [\underline{\theta}_v, \overline{\theta}_v]}} \left\{ \frac{\omega_v \sin\theta_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v \right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F \right)^2} \right\} \tag{3.3.52}$$

$$a_{26}^{absd} = \begin{cases} \dfrac{T_s k_{cvp}\left(\theta_v - \theta_v^0\right)}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\,\theta_v \geq \theta_v^0 \\ \dfrac{T_s k_{cvn}\left(\theta_v - \theta_v^0\right)}{D\cos^2\theta_v + E\sin^2\theta_v + F} & if\,\theta_v < \theta_v^0 \end{cases} \tag{3.3.53}$$

$$\underline{a}_{26}^{absd} = T_s k_{cvn} \min_{\theta_v \in \left[\underline{\theta}_v, \theta_v^0\right]} \left(\frac{\theta_v - \theta_v^0}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right) \tag{3.3.54}$$

$$\overline{a}_{26}^{absd} = T_s k_{cvp} \max_{\theta_v \in \left[\theta_v^0, \overline{\theta}_v\right]} \left(\frac{\theta_v - \theta_v^0}{D\cos^2\theta_v + E\sin^2\theta_v + F} \right) \tag{3.3.55}$$

$$a_{44}^{absd} = \begin{cases} 1 - \dfrac{k_a^2 / R_a + B_{mr} + k_{tvp}\omega_v}{J_{mr}} & if\,\omega_v \geq 0 \\ 1 - \dfrac{k_a^2 / R_a + B_{mr} - k_{tvn}\omega_v}{J_{mr}} & if\,\omega_v < 0 \end{cases} \tag{3.3.56}$$

$$\underline{a}_{44}^{absd} = 1 - \frac{T_s}{J_{mr}} \left(\frac{k_a^2}{R_a} + B_{mr} + \max\left\{ k_{tvp}\overline{\omega}_v, -k_{tvn}\underline{\omega}_v \right\} \right) \tag{3.3.57}$$

$$\overline{a}_{44}^{absd} = 1 - \frac{T_s}{J_{mr}} \left(\frac{k_a^2}{R_a} + B_{mr} \right) \tag{3.3.58}$$

$$a_{51}^{absd} = \frac{k_t}{J_v} a_{11}^{absd} \tag{3.3.59}$$

$$\underline{a}_{51}^{absd} = \frac{k_t}{J_v} \left(1 - \frac{T_s}{J_{tr}} \left(\frac{k_a^2}{R_a} + B_{tr} + \max\left\{ k_{thp}\overline{\omega}_h, -k_{thn}\underline{\omega}_h \right\} \right) \right) \tag{3.3.60}$$

$$\overline{a}_{51}^{absd} = \frac{k_t}{J_v} \left(1 - \frac{T_s}{J_{tr}} \left(\frac{k_a^2}{R_a} + B_{tr} \right) \right) \tag{3.3.61}$$

$$a_{52}^{absd} = \begin{cases} \dfrac{T_s\left(k_g \cos\theta_v k_{fvp}\omega_v - \Omega_h H \sin\theta_v \cos\theta_v\right)}{J_v} & if\,\omega_v \geq 0 \\ \dfrac{T_s\left(-k_g \cos\theta_v k_{fvn}\omega_v - \Omega_h H \sin\theta_v \cos\theta_v\right)}{J_v} & if\,\omega_v < 0 \end{cases} \tag{3.3.62}$$

$$\underline{a}_{52}^{absd} = \min_{\substack{\omega_v \in \left[\underline{\omega}_v, \overline{\omega}_v\right] \\ \Omega_h \in \left[\underline{\Omega}_h, \overline{\Omega}_h\right] \\ \theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]}} \left\{ a_{52}^{absd} \right\} \tag{3.3.63}$$

$$\overline{a}_{52}^{absd} = \max_{\substack{\omega_v \in \left[\underline{\omega}_v, \overline{\omega}_v\right] \\ \Omega_h \in \left[\underline{\Omega}_h, \overline{\Omega}_h\right] \\ \theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]}} \left\{ a_{52}^{absd} \right\} \tag{3.3.64}$$

$$a_{54}^{absd} = \begin{cases} \dfrac{l_m k_{fvp} \omega_v T_s}{J_v} & if\, \omega_v \geq 0 \\ -\dfrac{l_m k_{fvn} \omega_v T_s}{J_v} & if\, \omega_v < 0 \end{cases} \tag{3.3.65}$$

$$\underline{a}_{54}^{absd} = 0 \tag{3.3.66}$$

$$\overline{a}_{54}^{absd} = \max\left(\frac{l_m k_{fvp} \overline{\omega}_v T_s}{J_v}, -\frac{l_m k_{fvn} \underline{\omega}_v T_s}{J_v} \right) \tag{3.3.67}$$

$$a_{56}^{absd} = \frac{T_s g \left(\left(A - B \right) \cos\theta_v - C \sin\theta_v \right)}{J_v \left(\theta_v - \theta_v^0 \right)} \tag{3.3.68}$$

$$\underline{a}_{56}^{absd} = \frac{T_s g}{J_v} \min_{\theta_v \in \left[\underline{\theta}_v, \theta_v^- \right] \cup \left[\theta_v^+, \overline{\theta}_v \right]} \left\{ \frac{\left(\left(A - B \right) \cos\theta_v - C \sin\theta_v \right)}{\left(\theta_v - \theta_v^0 \right)} \right\} \tag{3.3.69}$$

$$\overline{a}_{56}^{absd} = \frac{T_s g}{J_v} \max_{\theta_v \in \left[\underline{\theta}_v, \theta_v^- \right] \cup \left[\theta_v^+, \overline{\theta}_v \right]} \left\{ \frac{\left(\left(A - B \right) \cos\theta_v - C \sin\theta_v \right)}{\left(\theta_v - \theta_v^0 \right)} \right\} \tag{3.3.70}$$

$$b_{22}^{absd} = \frac{k_m k_a k_2 T_s \cos\theta_v}{\left(D \cos^2\theta_v + E \sin^2\theta_v + F \right) R_a J_{mr}} \tag{3.3.71}$$

$$\underline{b}_{22}^{absd} = \frac{k_m k_a k_2 T_s}{R_a J_{mr}} \min_{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v \right]} \left\{ \frac{\cos\theta_v}{D \cos^2\theta_v + E \sin^2\theta_v + F} \right\} \tag{3.3.72}$$

$$\overline{b}_{22}^{absd} = \frac{k_m k_a k_2 T_s}{R_a J_{mr}} \max_{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v \right]} \left\{ \frac{\cos\theta_v}{D \cos^2\theta_v + E \sin^2\theta_v + F} \right\} \tag{3.3.73}$$

3.3.3.2 – Polytopic Shamma LPV Model of the TRMS

As some of the elements of the state space matrices are the same in both the Absolute and the Shamma LPV model, formulas to calculate the minimum and maximum value taken by the scheduling parameters of the Shamma LPV model are given only for those parameters which are different with respect to the Absolute LPV model. Thus:

$$a_{12}^{shad} = \begin{cases} \dfrac{-k_{chp} T_s}{l_t \cos\theta_v k_{fhp} \omega_h} & \theta_h \geq 0, \omega_h \geq 0 \\ \dfrac{k_{chp} T_s}{l_t \cos\theta_v k_{fhn} \omega_h} & \theta_h \geq 0, \omega_h < 0 \\ \dfrac{-k_{chn} T_s}{l_t \cos\theta_v k_{fhp} \omega_h} & \theta_h < 0, \omega_h \geq 0 \\ \dfrac{k_{chn} T_s}{l_t \cos\theta_v k_{fhn} \omega_h} & \theta_h \geq 0, \omega_h < 0 \end{cases} \tag{3.3.74}$$

$$\underline{a}_{12}^{shad} = -\frac{T_s}{l_t} \max\left\{ \frac{1}{k_{fhp} \omega_h^+}, \frac{-1}{k_{fhn} \omega_h^-} \right\} \max\left\{ k_{chp}, k_{chn} \right\} \max\left\{ \frac{1}{\cos\overline{\theta}_v}, \frac{1}{\cos\underline{\theta}_v} \right\} \tag{3.3.75}$$

$$\overline{a}_{12}^{shad} = -\frac{T_s}{l_t}\min\left\{\frac{1}{k_{fhp}\overline{\omega}_h}, \frac{-1}{k_{fhn}\underline{\omega}_h}\right\}\min\left\{k_{chp}, k_{chn}\right\} \tag{3.3.76}$$

$$a_{15}^{shad} = \begin{cases} -\dfrac{T_s\left(k_{chp}\theta_h\sin\theta_v - 2k_{cvp}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvp}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhp}\omega_h\cos^2\theta_v} & if\,\theta_v \geq 0, \theta_h \geq 0, \omega_h \geq 0 \\ \dfrac{T_s\left(k_{chp}\theta_h\sin\theta_v - 2k_{cvp}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvp}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhn}\omega_h\cos^2\theta_v} & if\,\theta_v \geq 0, \theta_h \geq 0, \omega_h < 0 \\ -\dfrac{T_s\left(k_{chn}\theta_h\sin\theta_v - 2k_{cvp}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvp}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhp}\omega_h\cos^2\theta_v} & if\,\theta_v \geq 0, \theta_h < 0, \omega_h \geq 0 \\ \dfrac{T_s\left(k_{chn}\theta_h\sin\theta_v - 2k_{cvp}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvp}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhn}\omega_h\cos^2\theta_v} & if\,\theta_v \geq 0, \theta_h < 0, \omega_h < 0 \\ -\dfrac{T_s\left(k_{chp}\theta_h\sin\theta_v - 2k_{cvn}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvn}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhp}\omega_h\cos^2\theta_v} & if\,\theta_v < 0, \theta_h \geq 0, \omega_h \geq 0 \\ \dfrac{T_s\left(k_{chp}\theta_h\sin\theta_v - 2k_{cvn}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvn}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhn}\omega_h\cos^2\theta_v} & if\,\theta_v < 0, \theta_h \geq 0, \omega_h < 0 \\ -\dfrac{T_s\left(k_{chn}\theta_h\sin\theta_v - 2k_{cvn}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvn}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhp}\omega_h\cos^2\theta_v} & if\,\theta_v < 0, \theta_h < 0, \omega_h \geq 0 \\ \dfrac{T_s\left(k_{chn}\theta_h\sin\theta_v - 2k_{cvn}\left(\theta_v-\theta_v^0\right)\cos\theta_v - k_{cvn}\left(\theta_v-\theta_v^0\right)^2\sin\theta_v\right)}{l_t k_{fhn}\omega_h\cos^2\theta_v} & if\,\theta_v < 0, \theta_h < 0, \omega_h \geq 0 \end{cases} \tag{3.3.77}$$

$$\underline{a}_{15}^{shad} = \min_{\substack{\omega_h \in \left[\underline{\omega}_h, \overline{\omega}_h\right] \\ \theta_h \in \left[\underline{\theta}_h, \overline{\theta}_h\right] \\ \theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]}} a_{15}^{shad} \tag{3.3.78}$$

$$\overline{a}_{15}^{shad} = \max_{\substack{\omega_h \in \left[\underline{\omega}_h, \overline{\omega}_h\right] \\ \theta_h \in \left[\underline{\theta}_h, \overline{\theta}_h\right] \\ \theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right]}} a_{15}^{shad} \tag{3.3.79}$$

$$a_{45}^{shad} = \begin{cases} \dfrac{gT_s\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{l_m k_{fvp}\omega_v} & \omega_v \geq 0 \\ \dfrac{-gT_s\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{l_m k_{fvn}\omega_v} & \omega_v < 0 \end{cases} \tag{3.3.80}$$

$$\underline{a}_{45}^{shad} = \frac{gT_s}{l_m}\min\left\{\min_{\substack{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right] \\ \omega_v \in \left[\omega_v^+, \overline{\omega}_v\right]}}\left\{-\frac{\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{k_{fvp}\omega_v}\right\}, \min_{\substack{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right] \\ \omega_v \in \left[\underline{\omega}_v, \omega_v^-\right]}}\left\{\frac{\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{k_{fvn}\omega_v}\right\}\right\} \tag{3.3.81}$$

$$\underline{a}_{45}^{shad} = \frac{gT_s}{l_m}\max\left\{\max_{\substack{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right] \\ \omega_v \in \left[\omega_v^+, \overline{\omega}_v\right]}}\left\{-\frac{\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{k_{fvp}\omega_v}\right\}, \max_{\substack{\theta_v \in \left[\underline{\theta}_v, \overline{\theta}_v\right] \\ \omega_v \in \left[\underline{\omega}_v, \omega_v^-\right]}}\left\{\frac{\left[C\cos\theta_v + (A-B)\sin\theta_v\right]}{k_{fvn}\omega_v}\right\}\right\} \tag{3.3.82}$$

3.4 – Summary

In this chapter a nonlinear model of the Twin-Rotor MIMO System has been described. Such a model has been used to obtain the so-called LPV Absolute Representation. Obtaining this kind of representation was a necessary step to obtain another LPV model by means of State Transformation. Finally, some formulas to obtain the interval polytopic LPV model have been given.

4 – TRMS IDENTIFICATION

4.1 - Introduction

In this chapter, a procedure to identify the unknown parameters of the TRMS is presented. This procedure is based on the same idea that has been used in TRMS modelling, that is to split the system into simpler subsystems and identify each model for its own.

The parameters affect the behaviour of the model of the Twin-Rotor MIMO System that has been obtained in the previous chapter. Some of these parameters can be considered to be known, as their values have been given by the manufacturer and are supposed to be reliable. Other parameters are considered to be unknown, thus they need to be estimated.

In this way, unknown parameters of the motor/propellers subsystems are identified at first. Then, unknown parameters of the aerodynamical part are identified. A numerical value of these parameters will be obtained through identification. The identification procedure can be divided into stages. Each stage returns the values of some of the unknown parameters, and makes use of both the known parameters and the unknown parameters that have been identified in previous stages.

Section 4.2 will resume in a table what is considered to be known and what is not.

In section 4.3, 4.4, 4.5 and 4.6 each identification stage will be described in detail.

In section 4.7 the model of the TRMS will be validated.

At last, in section 4.8, the LPV Polytopic TRMS Model will be obtained.

4.2 – Table of unknown and known parameters

Hereafter, ***Table 1*** lists the parameters that affect the behaviour of the Twin-Rotor MIMO System. The values of those parameters that can be considered to be known will be listed too. The identification stage in which the unknown parameters will be estimated is indicated by means of a number. Stage 1 is the tail propeller identification; stage 2 is the main propeller identification; stage 3 is the identification of the horizontal aeromechanical dynamics when the TRMS is driven by the tail propeller; stage 4 is the identification of the horizontal aeromechanical dynamics when the TRMS is driven by the main propeller; stage 5 is the identification of the vertical aeromechanical dynamics when the TRMS is driven by the main propeller; and stage 6 is the identification of the vertical aeromechanical dynamics when the TRMS is driven by the tail propeller.

Known Parameter	Value	Parameter to identify	Identification Stage
R_a	8Ω	J_{tr}	1
L_a	$0.86\,mH$	B_{tr}	1
k_a	$0.0202\,Nm/A$	k_{thp}	1
k_1	6.5	k_{thn}	1
k_2	8.5	J_{mr}	2
l_t	$0.282\,m$	B_{mr}	2
l_m	$0.246\,m$	k_{tvp}	2
l_b	$0.290\,m$	k_{tvn}	2
l_{cb}	$0.276\,m$	k_{fhp}	3
r_{ms}	$0.155\,m$	k_{fhn}	3

r_{ts}	$0.100\,m$	k_{chp}	3
m_t	$0.015\,kg$	k_{chn}	3
m_{tr}	$0.221\,kg$	k_{oh}	3
m_{ts}	$0.119\,kg$	k_{cvp}	4
m_m	$0.014\,kg$	k_{cvn}	4
m_{mr}	$0.236\,kg$	k_m	4
m_{ms}	$0.219\,kg$	k_{fvp}	5
m_b	$0.022\,kg$	k_{fvn}	5
m_{cb}	$0.068\,kg$	k_{ov}	5
k_g	0.2	k_t	6

Table 1 – Parameters of the Twin-Rotor MIMO System

4.3 – Tail propeller identification

4.3.1 – Identification procedure

In this section, a procedure to obtain values for the parameters that affect the tail propeller dynamics is explained. This procedure is based on the knowledge of the non-linear model of the tail propeller (3.2.43)-(3.2.7):

$$\frac{d\omega_h}{dt} = \frac{k_a k_1}{J_{tr} R_a} u_h - \left(\frac{B_{tr}}{J_{tr}} + \frac{k_a^2}{J_{tr} R_a} \right) \omega_h - \frac{f_1(\omega_h)}{J_{tr}} \tag{4.3.1}$$

$$f_1(\omega_h) = \begin{cases} k_{thp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{thn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases} \tag{4.3.2}$$

It is assumed that all the parameters except J_{tr}, B_{tr}, k_{thp} and k_{thn} are known, as their values are given by the manufacturer. The identification procedure should identify values of the unknown parameters in such a manner that the non-linear model behaviour resembles the real behaviour of the TRMS.

It is assumed to have at disposal N sets of data $\{u_h^i(k), \omega_h^i(k)\}$ where $i = 1,\ldots,N$ and $k = 1,\ldots,K_i$. K_i is the number of samples of the *i*-th set of data.

The identification procedure tries to find the minimum of the objective function:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \left(\omega_h^i(k) - \hat{\omega}_h^i(k) \right)^2 \tag{4.3.3}$$

where $\hat{\omega}_h^i(k)$ is the solution of (4.3.1).

4.3.2 – Results

The identification procedure has been applied to 14 sets of data. It must be noted that while all the sets of data can be used to identify J_{tr} and B_{tr}, a numerical value for k_{thp} can be obtained only from those sets of data where the input voltage u_h is positive. Analogously, a numerical value for k_{thn} can be obtained only from those sets of data where the input voltage u_h is negative.

Each set of data consists of 6000 samples, corresponding to an experiment that lasts 60 seconds, i.e., sampling data with a sample period of 0.01 s. Each set of data is the response of the tail propeller to an input voltage signal. Such a signal can be of two types: a step signal, that is a constant value from 0 to 60 s; or a step signal with a pseudo-random binary sequence overlapped from 30 s to 60 s. An example of such a signal, with step amplitude of 0.4 V and PRBS amplitude of 0.1 V, is shown in ***Fig. 4***.

The sets of data used in this identification stage are briefly resumed in ***Table 2*** and ***Table 3***.

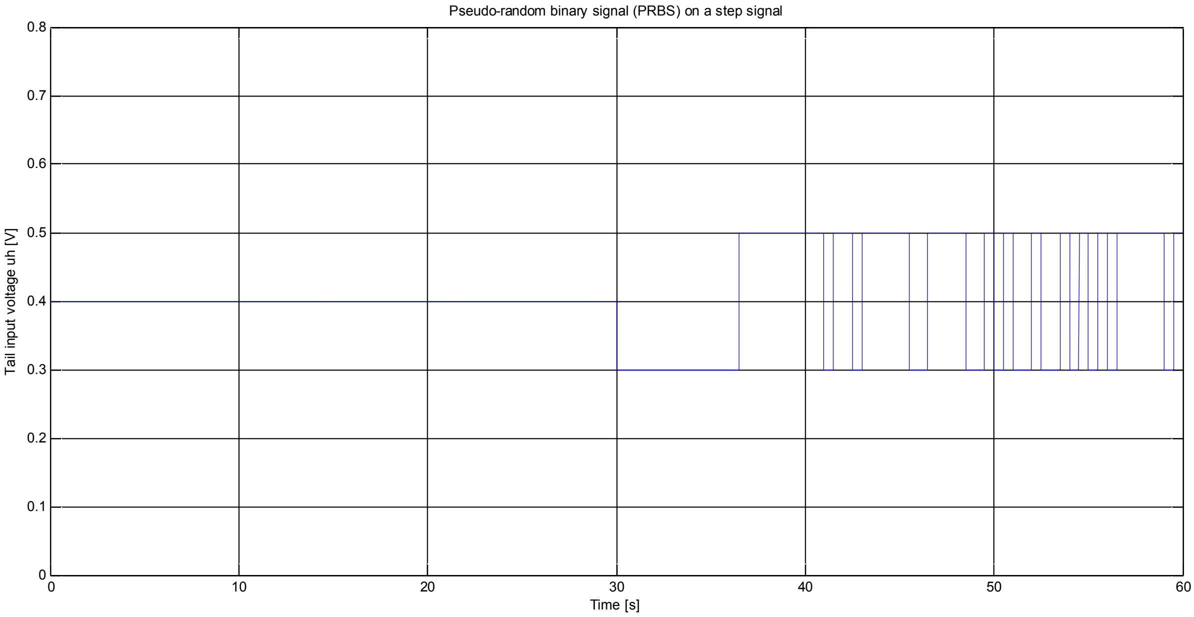

Fig. 4 – An example of PRBS (Pseudo-random binary sequence) overlapped on a step signal

	1	2	3	4	5	6	7	8
Step amplitude (0-60s) (V)	0.4	0.5	0.6	0.7	0.8	0.9	1.1	1.3
PRBS amplitude (30s-60s) (V)	0.1	0	0.1	0	0.1	0	0	0

Table 2 – Sets of data (positive input voltage u_h)

	9	10	11	12	13	14
Step amplitude (0-60s) (V)	-0.3	-0.4	-0.5	-0.6	-0.7	-0.7
PRBS amplitude (30s-60s) (V)	0.1	0	0.1	0	0.1	0

Table 3 – Sets of data (negative input voltage u_h)

The minimum of Eq. (4.3.3) for these sets of data is found by means of *fmincon*, a Matlab function of the Optimization Toolbox. This function attempts to find a constrained minimum of a scalar function of several variables (in this case, the unknown parameters). This is generally referred to as *constrained nonlinear optimization* or *nonlinear programming*. A first constraint on the parameters to be found is that they must have positive values. Another constraint on these parameters is due to numerical issues: the *fmincon* function has proved to fail in finding correct values for the unknown parameters when no assumption is made on their order of magnitude. Therefore, a lower and upper bound for each parameter has to be imposed.

By imposing these constraints on the unknown parameters:

$$J_{tr} \in \left[10^{-3}\ F, 10^{-1}\ F\right]$$

$$B_{tr} \in \left[10^{-4}\ \Omega^{-1}, 10^{-1}\ \Omega^{-1}\right]$$

$$k_{thp} \in \left[10^{-4}\ V^{-1}\Omega^{-1}, 5\cdot 10^{-2}\ V^{-1}\Omega^{-1}\right]$$

$$k_{thn} \in \left[10^{-4}\ V^{-1}\Omega^{-1}, 5\cdot 10^{-2}\ V^{-1}\Omega^{-1}\right]$$

the following results have been obtained using the *fmincon* function:

$$J_{tr} = 0.0101\ F$$

$$B_{tr} = 0.0097\ \Omega^{-1}$$

$$k_{thp} = 0.0032\ V^{-1}\Omega^{-1}$$

$$k_{thn} = 4.8951\cdot 10^{-4}\ V^{-1}\Omega^{-1}$$

Comparisons between the real data and the simulation of Eq. (4.3.1) with the previous values are shown in ***Fig. 5*** and ***Fig. 6***.

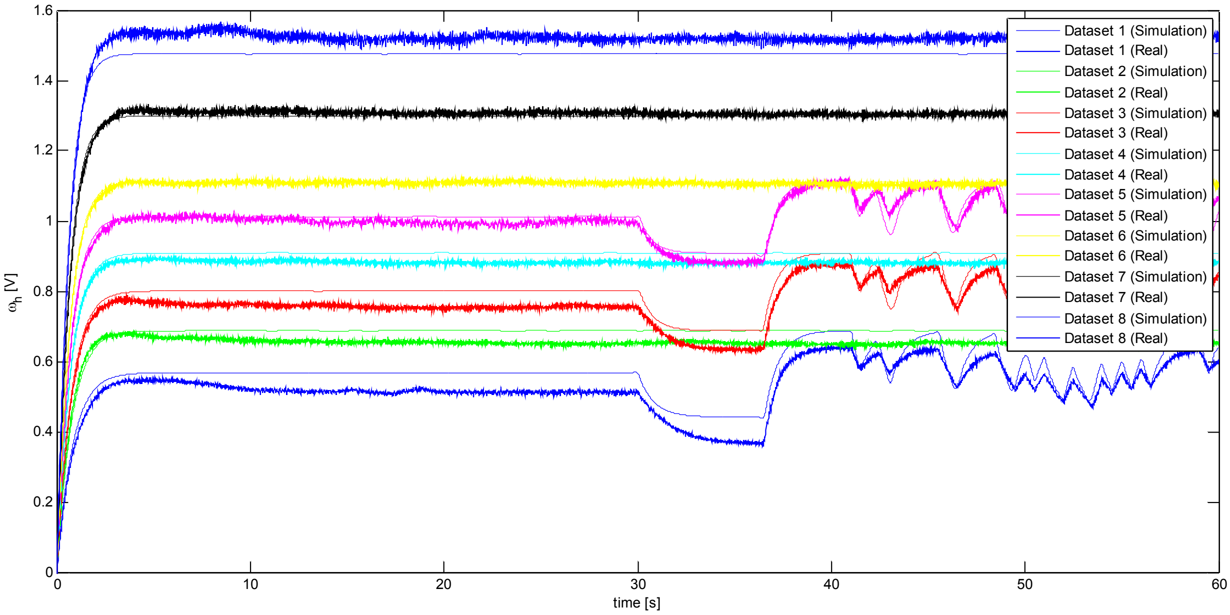

Fig. 5 - Comparison between real and simulation data (tail propeller - positive input u_h)

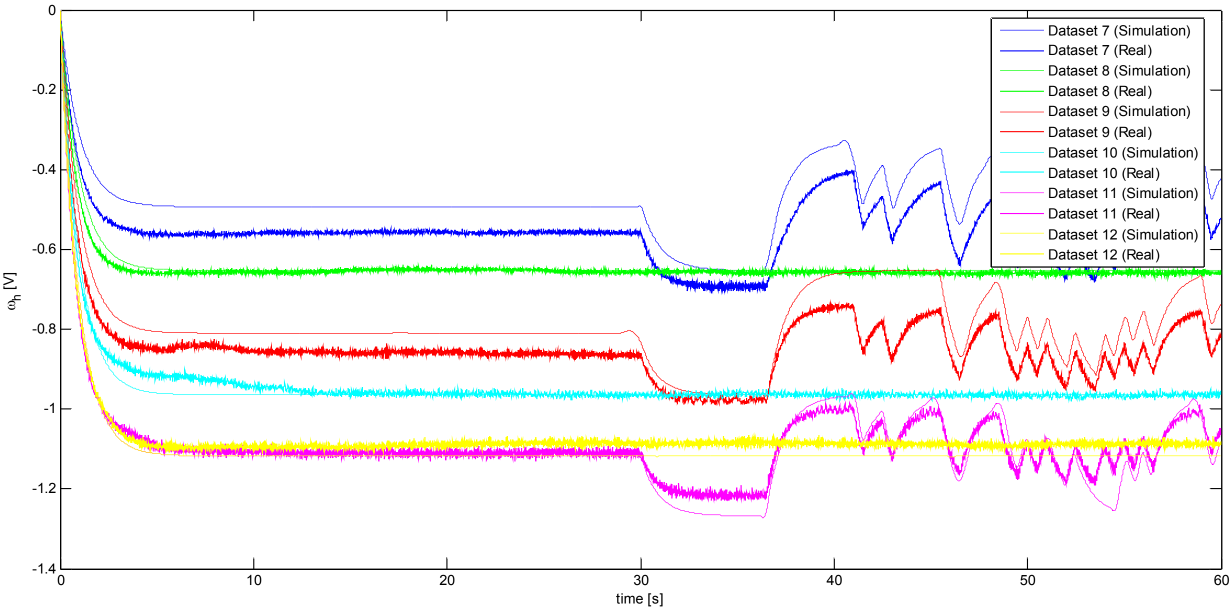

Fig. 6 - Comparison between real and simulation data (tail propeller - negative input u_h)

Table 4 shows the maximum absolute errors and mean absolute errors for each dataset. As it can be seen in the table and in the figures, the identification procedure gives good performance in most cases. However, this is not true for all sets of data, probably due to unmodelled effects as, for example, the stiction, that occurs when ω_h is either zero or very close to zero. As explained in Chapter 3, the lack of stiction and Coulomb friction in the model around zero can be seen, but since the working points of the rotors will not be close to this value, the resulting error does not degrade too much the goodness of the results.

Dataset	Maximum Absolute Error	Mean Absolute Error	Mean of the Dataset
1	0.0853 (16.31%)	0.0406 (7.76%)	0.5231
2	0.0541 (8.35%)	0.0321 (4.96%)	0.6478
3	0.0643 (8.44%)	0.0328 (4.30%)	0.7616
4	0.0482 (5.53%)	0.0254 (2.91%)	0.8726
5	0.0413 (4.15%)	0.0128 (1.28%)	0.9957
6	0.0287 (2.63%)	0.0061 (0.56%)	1.0945
7	0.0389 (3.01%)	0.0085 (0.66%)	1.2919
8	0.0925 (6.15%)	0.0457 (3.04%)	1.5057
9	0.1246 (22.35%)	0.0766 (13.74%)	-0.5574
10	0.0585 (9.04%)	0.0070 (1.08%)	-0.6478
11	0.0993 (11.74%)	0.0539 (6.37%)	-0.8462
12	0.0626 (6.64%)	0.0099 (1.05%)	-0.9429
13	0.0758 (6.96%)	0.0169 (1.55%)	-1.0886
14	0.0515 (4.80%)	0.0278 (2.59%)	-1.0730

Table 4 - Maximum Absolute Error and Mean Absolute Error (Tail Propeller Identification)

4.4 – Main propeller identification

4.4.1 – Identification procedure

In this section, a procedure to obtain values for the parameters that affect the main propeller dynamics is explained. This procedure is analogous to the one described in the previous section for the identification of the tail propeller dynamics parameters, and is based on the knowledge of the non-linear model of the main propeller (3.2.46)-(3.2.8):

$$\frac{d\omega_v}{dt} = \frac{k_a k_2}{J_{mr} R_a} u_v - \left(\frac{B_{mr}}{J_{mr}} + \frac{k_a^2}{J_{mr} R_a} \right) \omega_v - \frac{f_4(\omega_v)}{J_{mr}} \tag{4.4.1}$$

$$f_4(\omega_v) = \begin{cases} k_{tvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{tvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases} \tag{4.4.2}$$

It is assumed that all the parameters except J_{mr}, B_{mr}, k_{tvp} and k_{tvn} are known, as their values are given by the manufacturer. The identification procedure should identify some values for the unknown parameters in such a manner that the non-linear model behaviour resembles the real behaviour of the TRMS.

It is assumed to have at disposal N sets of data $\{u_v^i(k), \omega_v^i(k)\}$ where $i = 1, \ldots, N$ and $k = 1, \ldots, K_i$. K_i is the number of samples of the i-th set of data.

The identification procedure finds the minimum of the objective function:

$$J = \sum_{i=1}^{N} \sum_{k=1}^{K_i} \left(\omega_v^i(k) - \hat{\omega}_v^i(k) \right)^2 \tag{4.4.3}$$

where $\hat{\omega}_v^i(k)$ is the solution of Eq. (4.4.1).

4.4.2 – Results

The identification procedure has been applied to 10 sets of data. It must be noted that while all the sets of data can be used to identify J_{mr} and B_{mr}, a numerical value for k_{tvp} can be obtained only from those sets of data where the input voltage u_v is positive and a numerical value for k_{tvn} can be obtained only with a negative input voltage u_v.

Each set of data consists of 6000 samples, corresponding to an experiment that lasts 60 seconds, i.e., with a sample period of 0.01 s. Each set of data is the response of the main propeller to an input voltage signal. Such a signal can be of two types: a step signal, that is a constant value from 0 to 60 s; or a step signal with a pseudo-random binary sequence overlapped from 30 s to 60 s. The sets of data used in this identification stage are briefly resumed in ***Table 5*** and ***Table 6***.

Dataset	1	2	3	4	5	6	7
Step amplitude (0-60s) (V)	0.6	0.6	0.8	0.8	1.0	1.2	1.4
PRBS amplitude (30s-60s) (V)	0.1	0	0.1	0	0	0	0

Table 5 - Sets of data (Positive input voltages u_v)

Dataset	8	9	10
Step amplitude (0-60s) (V)	-0.4	-0.6	-0.8
PRBS amplitude (30s-60s) (V)	0.1	0.1	0.1

Table 6 – Sets of data (Negative input voltages u_v)

The implementation of this identification stage is similar to the tail propeller one, and makes use of the Matlab function *fmincon*. Analogously, because of convergence issues, a lower bound and an upper bound for each parameter must be imposed.

These results have been obtained by imposing these constraints on the unknown parameters:

$$J_{mr} \in \left[10^{-3}\,F, 10^{-1}\,F\right]$$

$$B_{mr} \in \left[10^{-4}\,\Omega^{-1}, 10^{-1}\,\Omega^{-1}\right]$$

$$k_{tvp} \in \left[10^{-4}\,V^{-1}\Omega^{-1}, 5\cdot 10^{-2}\,V^{-1}\Omega^{-1}\right]$$

$$k_{tvn} \in \left[10^{-4}\,V^{-1}\Omega^{-1}, 5\cdot 10^{-2}\,V^{-1}\Omega^{-1}\right]$$

The *fmincon* function has returned the following values:

$$J_{mr} = 0.0562\,F$$

$$B_{mr} = 0.0032\,\Omega^{-1}$$

$$k_{tvp} = 0.0177\,V^{-1}\Omega^{-1}$$

$$k_{tvn} = 0.0154\,V^{-1}\Omega^{-1}$$

A comparison between the real data and the numerical solution of (4.4.1) with such values is shown in ***Fig.*** *7* and ***Fig. 8***. ***Table*** *7* shows the maximum absolute errors and the mean absolute errors for each dataset. As it can be seen in the table and in the figures, the identification procedure gives good results in most cases.

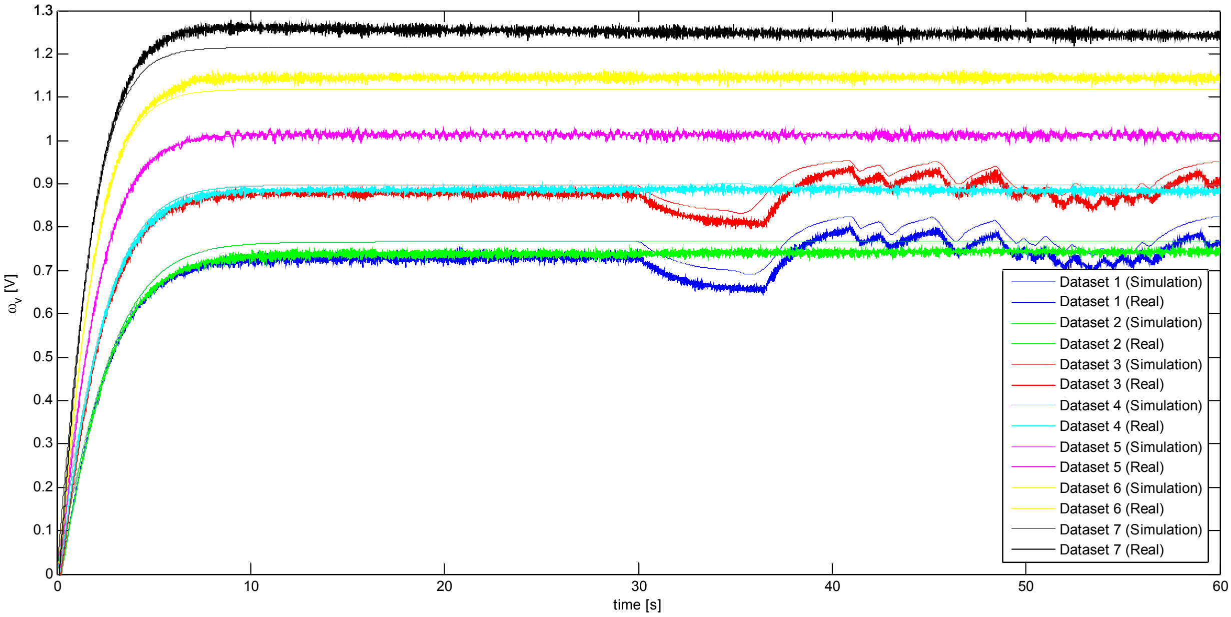

Fig. 7 - Comparison between real and simulation data (main propeller – positive input u_v)

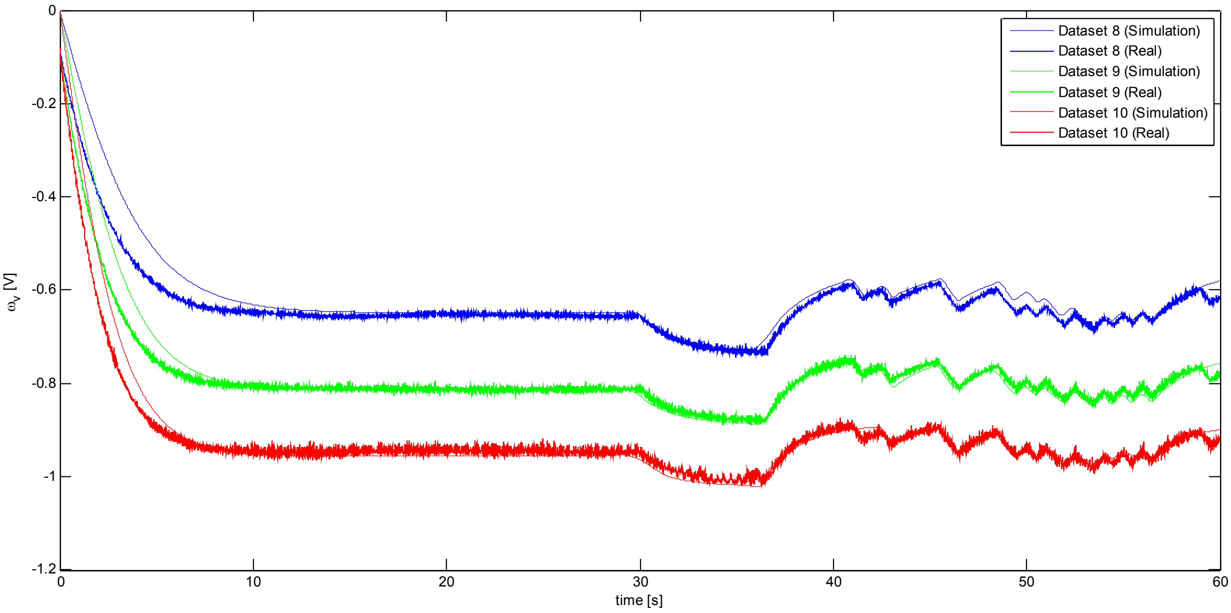

Fig. 8 - Comparison between real and simulation data (main propeller - negative input u_v)

Dataset	Maximum Absolute Error	Mean Absolute Error	Mean of the Dataset
1	0.0763 (10.86%)	0.0342 (4.86%)	0.7026
2	0.0816 (11.50%)	0.0259 (3.64%)	0.7097
3	0.0991 (11.72%)	0.0233 (2.75%)	0.8454
4	0.0739 (8.65%)	0.0126 (1.48%)	0.8536
5	0.0819 (8.37%)	0.0050 (0.51%)	0.9796
6	0.0888 (8%)	0.0254 (2.28%)	1.1101
7	0.1128 (9.28%)	0.0344 (2.83%)	1.2147
8	0.1274 (20.23%)	0.0204 (3.24%)	-0.6295
9	0.1271 (16.20%)	0.0154 (1.97%)	-0.7843
10	0.1263 (13.76%)	0.0162 (1.76%)	-0.9180

Table 7 - Maximum Absolute Error and Mean Absolute Error (Main Propeller Identification)

4.5 – Horizontal aeromechanical identification

4.5.1 – Identification Procedure

In this section, a procedure to obtain values for the parameters that describe the horizontal aeromechanical dynamics is explained. This procedure is based on the knowledge of the non-linear model (3.2.44)-(3.2.45)-(3.2.20)-(3.2.21)-(3.2.22). At first, the horizontal dynamics due to an input voltage acting on the tail propeller is identified, that is what has been called "Identification stage 3", and values for the unknown parameters $k_{fhp}, k_{fhn}, k_{oh}, k_{chp}, k_{chn}$ are found. Then, the horizontal dynamics due to an input voltage acting on the main propeller is identified, that is what has been called "Identification stage 4", and the unknown parameters k_m, k_{cvp}, k_{cvn} are estimated.

4.5.1.1 – Identification stage 3

When no input voltage acts on the main propeller, Eqs. (3.2.44) and (3.2.45) simplify to:

$$\frac{d\Omega_h}{dt} = \frac{l_t f_2(\omega_h)\cos\theta_v^0 - k_{oh}\Omega_h - f_3(\theta_h)}{D\cos^2\theta_v^0 + E\sin^2\theta_v^0 + F} \tag{4.5.1}$$

$$\frac{d\theta_h}{dt} = \Omega_h \tag{4.5.2}$$

where θ_v^0 is the equilibrium value for the vertical angle when no input voltage acts on the main propeller.

$$f_2(\omega_h) = \begin{cases} k_{fhp}\omega_h^2 & \text{if } \omega_h \geq 0 \\ -k_{fhn}\omega_h^2 & \text{if } \omega_h < 0 \end{cases} \tag{4.5.3}$$

$$f_3(\theta_h) = \begin{cases} k_{chp}\theta_h & \text{if } \theta_h \geq 0 \\ k_{chn}\theta_h & \text{if } \theta_h < 0 \end{cases} \tag{4.5.4}$$

Replacing Eq. (4.5.2) in Eq. (4.5.1) leads to:

$$\ddot{\theta}_h + \frac{k_{oh}}{K_v^0}\dot{\theta}_h + \frac{f_3(\theta_h)}{K_v^0} = \frac{l_t f_2(\omega_h)\cos\theta_v^0}{K_v^0} \tag{4.5.5}$$

$$K_v^0 = D\cos^2\theta_v^0 + E\sin^2\theta_v^0 + F \tag{4.5.6}$$

Laplace transform of (4.5.5) leads to:

$$\Theta_h(s) = \frac{K}{1 + 2\xi T_w s + T_w^2 s^2}\Psi_h(s) \tag{4.5.7}$$

where $\Psi_h(s)$ is the Laplace transform of $\omega_h^2(t)$.

$$K = \begin{cases} \dfrac{l_t k_{fhp}\cos\theta_v^0}{k_{chp}} & if\, \omega_h \geq 0 \\ -\dfrac{l_t k_{fhn}\cos\theta_v^0}{k_{chn}} & if\, \omega_h < 0 \end{cases} \tag{4.5.8}$$

$$2\xi T_w = \frac{k_{oh}}{k_{chp/n}} \tag{4.5.9}$$

$$T_w^2 = \frac{K_v^0}{k_{chp/n}} \tag{4.5.10}$$

Hence, the knowledge of K, ξ and T_w can be used for estimation of the unknown parameters:

$$k_{chp} = \frac{K_v^0}{\left(T_w^{(p)}\right)^2} \tag{4.5.11}$$

$$k_{chn} = \frac{K_v^0}{\left(T_w^{(n)}\right)^2} \tag{4.5.12}$$

$$k_{oh}^{(p)} = 2\xi^{(p)} T_w^{(p)} k_{chp} \tag{4.5.13}$$

$$k_{oh}^{(n)} = 2\xi^{(n)} T_w^{(n)} k_{chn} \tag{4.5.14}$$

$$k_{fhp} = \frac{K^{(p)} k_{chp}}{l_t \cos\theta_v^0} \tag{4.5.15}$$

$$k_{fhn} = -\frac{K^{(n)} k_{chn}}{l_t \cos\theta_v^0} \tag{4.5.16}$$

Values for $K^{(p)}$ and $K^{(n)}$ can be found in a simple way through the knowledge of steady-state values of ω_h and θ_h:

$$K^{(p/n)} = \frac{\theta_h^\infty}{\left(\omega_h^\infty\right)^2} \tag{4.5.17}$$

Values for $T_w^{(p)}$, $T_w^{(n)}$, $\xi^{(p)}$ and $\xi^{(n)}$ can be estimated by means of some optimization method.

It is assumed to have at disposal N sets of data $\{\,\omega_h^i(k)\,,\theta_h^i(k)\,\}$ where $i = 1,\ldots,N$, $k = 1,\ldots,K_i$ and K_i is the number of samples of the i-th set of data. By applying the described procedure to each set of data, different values for the unknown parameters can be obtained. Such values can be used to define lower bounds and upper bounds for the values of the unknown parameters. Then, the minimum of the objective function:

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K_i}\left(\theta_h^i(k) - \hat{\theta}_h^i(k)\right)^2 \tag{4.5.18}$$

where $\hat{\theta}_h^i(k)$ is the solution of Eqs. (4.5.1)-(4.5.2), can be found.

4.5.1.2 – Identification stage 4

When no input voltage acts on the tail propeller, Eqs. (3.2.44) and (3.2.45) simplify to:

$$\begin{aligned}\frac{d\Omega_h}{dt} = {} & \frac{k_m \omega_v \sin\theta_v \Omega_v \left(D\cos^2\theta_v - E\sin^2\theta_v - F - 2E\cos^2\theta_v\right)}{\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right)^2} \\ & + \frac{-k_{oh}\Omega_h - f_3\left(\theta_h\right) + f_6\left(\theta_v\right)}{D\cos^2\theta_v + E\sin^2\theta_v + F} + \frac{k_m \cos\theta_v \left(\dfrac{k_a k_2}{R_a} u_v - \left(B_{mr} + \dfrac{k_a^2}{R_a}\right)\omega_v - f_4\left(\omega_v\right)\right)}{J_{mr}\left(D\cos^2\theta_v + E\sin^2\theta_v + F\right)}\end{aligned} \tag{4.5.19}$$

$$\frac{d\theta_h}{dt} = \Omega_h \tag{4.5.20}$$

$$f_6(\theta_v) = \begin{cases} k_{cvp}\left(\theta_v - \theta_v^0\right)^2 & \text{if } \theta_v \geq \theta_v^0 \\ k_{cvn}\left(\theta_v - \theta_v^0\right)^2 & \text{if } \theta_v < \theta_v^0 \end{cases} \tag{4.5.21}$$

It is assumed to have at disposal N sets of data $\{\omega_v^i(k), \theta_v^i(k), \theta_h^i(k)\}$ where $i = 1,\ldots,N$, $k = 1,\ldots,K_i$ and K_i is the number of samples of the *i*-th set of data.

The identification procedure should identify some values for k_m, k_{cvp} and k_{cvn}, as all the other parameter values are given by the manufacturer or have been identified previously.

If the input voltage in the main propeller is a step signal, at steady-state:

$$\Omega_v^\infty = \Omega_h^\infty = \frac{k_a k_2}{R_a} u_v^\infty - \left(B_{mr} + \frac{k_a^2}{R_a} \right) \omega_v^\infty - f_4\left(\omega_v^\infty\right) = 0 \tag{4.5.22}$$

and Eq. (4.5.19) reduces to:

$$\frac{-f_3\left(\theta_h^\infty\right) + f_6\left(\theta_v^\infty\right)}{D\cos^2\theta_v^\infty + E\sin^2\theta_v^\infty + F} = 0 \tag{4.5.23}$$

or, equivalently:

$$-k_{chp}\theta_h^\infty + k_{cvp}\left(\theta_v^\infty - \theta_v^0\right)^2 = 0 \quad \left(\theta_v \geq \theta_v^0\right) \tag{4.5.24}$$

$$-k_{chp}\theta_h^\infty + k_{cvn}\left(\theta_v^\infty - \theta_v^0\right)^2 = 0 \quad \left(\theta_v < \theta_v^0\right) \tag{4.5.25}$$

A first estimation of k_{cvp} and k_{cvn} can be obtained from Eqs. (4.5.24) and (4.5.25) as:

$$k_{cvp/n} = \frac{k_{chp}\theta_h^\infty}{\left(\theta_v^\infty - \theta_v^0\right)^2} \tag{4.5.26}$$

Later, the minimum over k_m, k_{cvp} and k_{chn} of the objective function:

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K_i}\left(\theta_h^i(k) - \hat{\theta}_h^i(k)\right)^2 \tag{4.5.27}$$

where $\hat{\theta}_h^i(k)$ is the solution of Eqs. (4.5.19)-(4.5.20), should be found. $\hat{\theta}_h$ can be calculated by considering that both ω_v and θ_v are data provided by the sensors installed in the TRMS.

4.5.2 – Results

4.5.2.1 – Identification Stage 3

The identification procedure has been applied to 6 sets of data. It must be noted that while all the sets of data can be used to identify k_{oh}, numerical values for k_{fhp} and k_{chp} can be obtained only from those sets of data where the angular speed of the tail rotor ω_h is positive. Analogously, numerical values for k_{fhn} and k_{chn} can be obtained only from those sets of data where the angular speed of the tail rotor ω_h is negative.

Each set of data consists of 6000 samples, corresponding to an experiment that lasts 60 seconds, obtained with a sample period of 0.01 s, obtained as the response of the TRMS to an input voltage step signal. The sets of inputs used in this identification stage are briefly resumed in ***Table 8***.

Dataset	1	2	3	4	5	6
Step amplitude (0-60s) (V)	0.7	0.9	1.1	1.3	-0.6	-0.7

Table 8 - Tail input voltages u_h for horizontal aeromechanical identification

Expressions from Eqs. (4.5.11) to (4.5.16) have been used to obtain a preliminary estimation of the unknown parameters. Values for $K^{(p)}$ and $K^{(n)}$ have been obtained using Eq. (4.5.17). Values for $T_w^{(p)}$, $T_w^{(n)}$, $\xi^{(p)}$ and $\xi^{(n)}$ have been obtained by means of *pem* Matlab function. Results of this are briefly resumed in ***Table 9***:

Dataset	K	ξ	T_w	k_{fhp}	k_{fhn}	k_{oh}	k_{chp}	k_{chn}
1	0.6108	0.4096	2.5782	0.0178		0.0141	0.0067	
2	0.5266	0.3140	2.4196	0.0174		0.0115	0.0076	
3	0.5231	0.3608	2.3816	0.0178		0.0135	0.0078	
4	0.5020	0.4624	2.2250	0.0196		0.0185	0.0090	
5	-1.2331	0.6436	3.1611		0.0239	0.0181		0.0044
6	-1.6599	0.6754	3.3375		0.0288	0.0180		0.0040

Table 9 – Preliminary estimation of horizontal aeromechanical parameters

The minimum of (4.5.18) can be found by means of *fmincon*. The results obtained in the previous stage can be used to impose constraints on the unknown parameters:

$$k_{fhp} \in \left[0.0174\, kg\, m^2\, s^{-2}\, V^{-2}, 0.0196\, kg\, m^2\, s^{-2}\, V^{-2}\right]$$

$$k_{fhn} \in \left[0.0239\, kg\, m^2\, s^{-2}\, V^{-2}, 0.0288\, kg\, m^2\, s^{-2}\, V^{-2}\right]$$

$$k_{chp} \in \left[0.0067\, kg\, m^2\, s^{-2}, 0.0090\, kg\, m^2\, s^{-2}\right]$$

$$k_{chn} \in \left[0.0040\, kg\, m^2\, s^{-2}, 0.0044\, kg\, m^2\, s^{-2}\right]$$

$$k_{oh} \in \left[0.0115\, kg\, m^2\, s^{-1}, 0.0185\, kg\, m^2\, s^{-1}\right]$$

The *fmincon* function has returned the following values:

$$k_{fhp} = 0.0187\, kg\, m^2\, s^{-2}\, V^{-2}$$

$$k_{fhn} = 0.0271\, kg\, m^2\, s^{-2}\, V^{-2}$$

$$k_{chp} = 0.0085\, kg\, m^2\, s^{-2}$$

$$k_{chn} = 0.0043\, kg\, m^2\, s^{-2}$$

$$k_{oh} = 0.0154\, kg\, m^2\, s^{-1}$$

A comparison between the real data and the simulation of Eqs. (4.5.1)-(4.5.2) with such values is shown in ***Fig. 9*** and ***Fig. 10***. ***Table 10*** shows the maximum absolute errors, the mean absolute errors and the steady-state errors for each dataset. The identification procedure gives good performances in most cases. As 4 experiments out of 6 dealt with positive input voltages, the best performances are achieved with these sets of data. This is not a problem, as in the following chapters the system will be operated with positive set points.

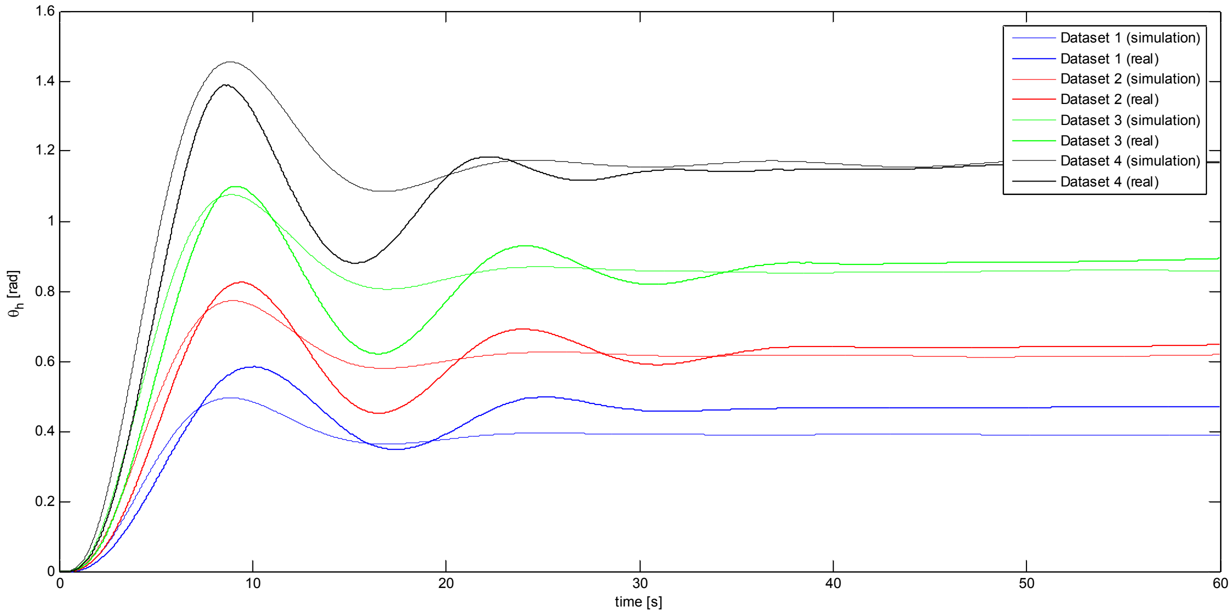

Fig. 9 - Comparison between real and simulation data – Identification Stage 3 (positive u_h)

Dataset	Max Absolute Error	Mean Absolute Error	Steady-State error
1	6.3°	3.7°	-4.7°
2	7.4°	2.1°	-1.6°
3	10.7°	2.6°	-2°
4	13.8°	3.1°	0.1°
5	23.7°	13.4°	-11.5°
6	19.3°	12.5°	17.5°

Table 10 – Max, Mean and Steady-State errors for Identification Stage 3

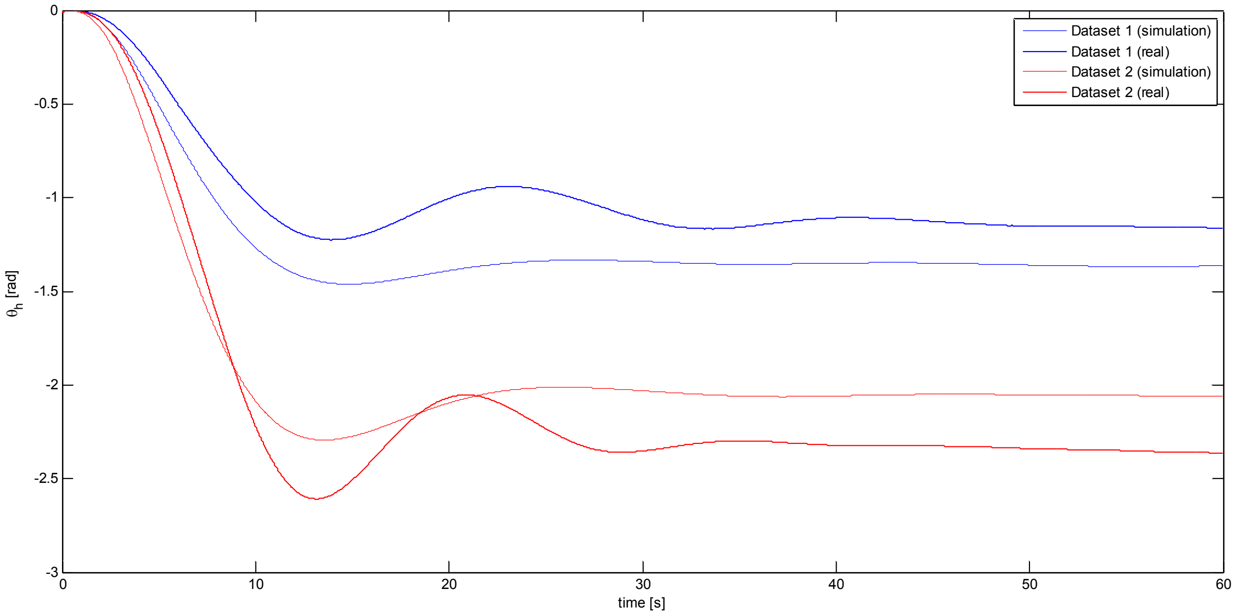

Fig. 10 - Comparison between real and simulation data – Identification Stage 3 (negative u_h)

4.5.2.2 – Identification Stage 4

The identification procedure has been applied to 9 sets of data. It must be noted that while all the sets of data can be used to identify k_m, a numerical value for k_{cvp} can be obtained only from those sets of data where the input voltage of the main propeller u_v is positive. Analogously, a numerical value for k_{cvn} can be obtained only from those sets of data where the input voltage of the main propeller u_v is negative.

The sets of inputs used in this identification stage are briefly resumed in ***Table 11***.

Dataset	1	2	3	4	5	6	7	8	9
Step amplitude (0-60s) (V)	0.6	0.8	1.0	1.2	1.4	-0.7	-0.8	-0.9	-1.0

Table 11 - Main input voltages u_v for horizontal aeromechanical identification

Eq. (4.5.26) has been used to obtain a preliminary estimation of k_{cvp} and k_{cvn}, as resumed in ***Table 12***.

Dataset	1	2	3	4	5	6	7	8	9
$k_{cvp}\left(kg\,m^2\,s^{-2}\right)$	0.0125	0.0163	0.0225	0.0211	0.0208				
$k_{cvn}\left(kg\,m^2\,s^{-2}\right)$						0.0031	0.0084	0.0132	0.0145

Table 12 - Estimation of k_{cvp} and k_{cvn}

These values can be used to impose some constraint on the unknown parameters as follows:

$$k_m \in \left[10^{-3}\,kg\,m^2\,s^{-1}\,V^{-1}, 5\cdot 10^{-2}\,kg\,m^2\,s^{-1}\,V^{-1}\right]$$

$$k_{cvp} \in \left[0.0125\,kg\,m^2\,s^{-2}, 0.0225\,kg\,m^2\,s^{-2}\right]$$

$$k_{cvn} \in \left[0.0031\,kg\,m^2\,s^{-2}, 0.0145\,kg\,m^2\,s^{-2}\right]$$

Then, the *fmincon* has returned the following values:

$$k_m = 0.0122\,kg\,m^2\,s^{-1}\,V^{-1}$$

$$k_{cvp} = 0.0207\,kg\,m^2\,s^{-2}$$

$$k_{cvn} = 0.0094\,kg\,m^2\,s^{-2}$$

A comparison between the real data and the simulation of Eqs. (4.5.19)-(4.5.20) with such values is shown in ***Fig. 11*** and ***Fig. 12***. ***Table 13*** shows the maximum absolute errors, the mean absolute errors and the steady-state errors for each dataset. The simulation has proved to give worse performance when $u_v < 0$ than when $u_v > 0$. This is not a big problem, as in the following chapters the system will be operated with positive inputs.

Dataset	Max Absolute Error	Mean Absolute Error	Steady-State error
1	9.15°	3.63°	3.14°
2	10.89°	4.49°	3.60°
3	11.80°	3.13°	2.02°
4	12.01°	3.15°	2.47°
5	12.19°	3.02°	2.30°
6	22.29°	3.94°	1.30°
7	24.67°	3.61°	0.28°
8	24.71°	4.24°	1.47°
9	24.54°	4.86°	2.56°

Table 13 - Max, Mean and Steady-State errors for Identification Stage 4

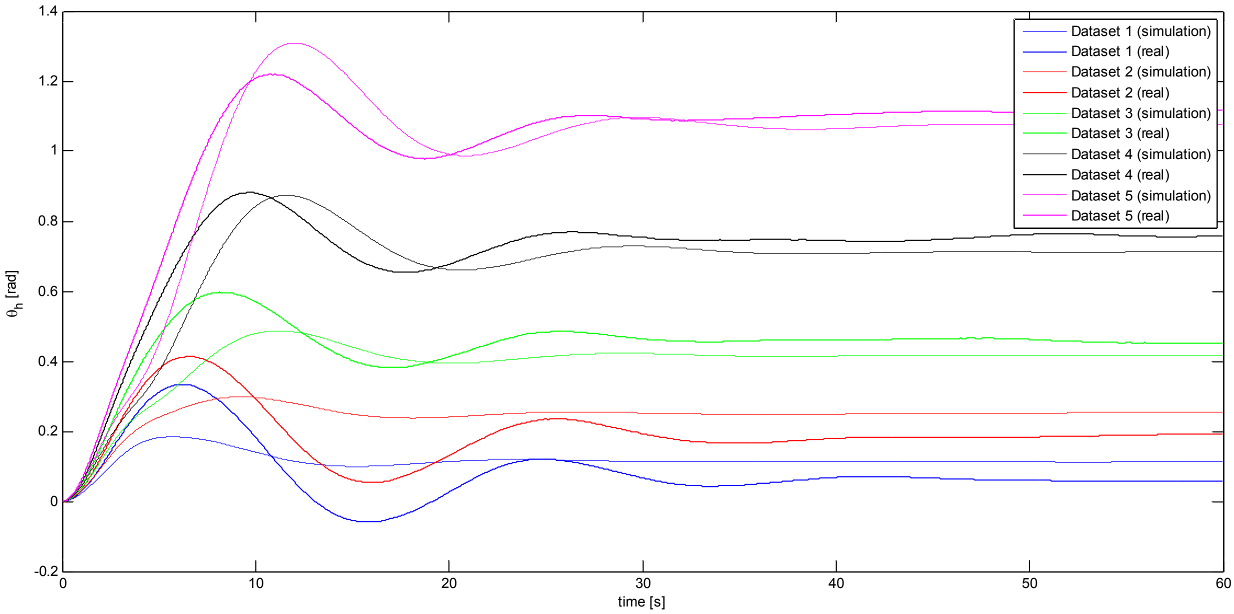

Fig. 11 - Comparison between real and simulation data – Identification Stage 4 (positive u_v)

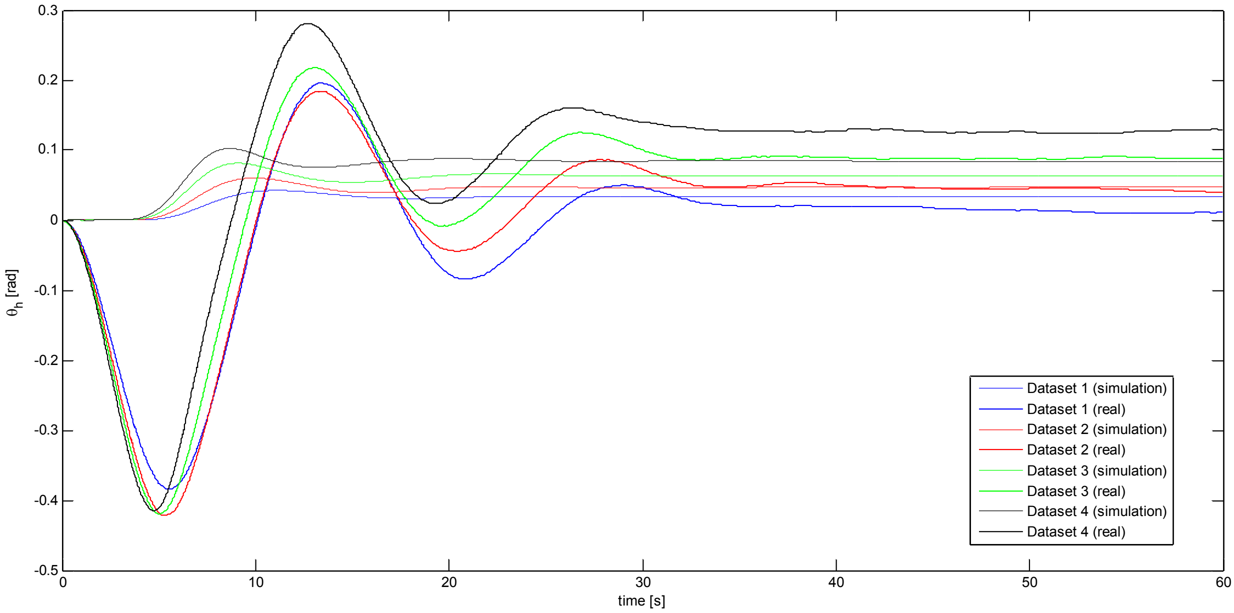

Fig. 12 - Comparison between real and simulation data – Identification Stage 4 (negative u_v)

4.6 – Vertical aeromechanical identification

4.6.1 – Identification procedure

In this section, a procedure to obtain values for the parameters that describe the vertical aeromechanical dynamics is explained. This procedure is based on the knowledge of the non-linear model (3.2.47)-(3.2.48). At first, the vertical dynamics due to an input voltage acting on the main propeller is identified, that is what has been called "Identification Stage 5", that will return values for the unknown parameters k_{fvp}, k_{fvn} and k_{ov}. Then, the vertical dynamics due to an input voltage acting on the tail propeller is identified, that is what has been called "Identification Stage 6", that give a value for the unknown parameter k_t.

4.6.1.1 – Identification Stage 5

When no input voltage acts on the tail propeller, Eqs. (3.2.47) and (3.2.48) simplify to:

$$\frac{d\Omega_v}{dt} = \frac{l_m f_5(\omega_v) + k_g \Omega_h f_5(\omega_v)\cos\theta_v - k_{ov}\Omega_v - 0.5\Omega_h^2 H \sin 2\theta_v + g\left((A-B)\cos\theta_v - C\sin\theta_v\right)}{J_v} \tag{4.6.1}$$

$$\frac{d\theta_v}{dt} = \Omega_v \tag{4.6.2}$$

$$f_5(\omega_v) = \begin{cases} k_{fvp}\omega_v^2 & \text{if } \omega_v \geq 0 \\ -k_{fvn}\omega_v^2 & \text{if } \omega_v < 0 \end{cases} \tag{4.6.3}$$

The identification procedure should identify some values for the unknown parameters k_{fvp}, k_{fvn} and k_{ov} in such a manner that the non-linear model behaviour resembles the real behaviour of the TRMS.

At steady-state, as $\Omega_h^\infty = 0$ and $\Omega_v^\infty = 0$, Eq. (4.6.1) reduces to:

$$\frac{l_m f_5(\omega_v^\infty) + g\left((A-B)\cos\theta_v^\infty - C\sin\theta_v^\infty\right)}{J_v} = 0 \tag{4.6.4}$$

Thus, a preliminary estimation of k_{fvp} and k_{fvn} can be obtained as:

$$k_{fvp} = \frac{g\left((B-A)\cos\theta_v^\infty + C\sin\theta_v^\infty\right)}{l_m\left(\omega_v^\infty\right)^2} \tag{4.6.5}$$

$$k_{fvn} = -\frac{g\left((B-A)\cos\theta_v^\infty + C\sin\theta_v^\infty\right)}{l_m\left(\omega_v^\infty\right)^2} \tag{4.6.6}$$

Later, the minimum over k_{fvp}, k_{fvn} and k_{ov} of the objective function:

$$J = \sum_{i=1}^{N}\sum_{k=1}^{K_i}\left(\theta_v^i(k) - \hat{\theta}_v^i(k)\right)^2 \tag{4.6.7}$$

where $\hat{\theta}_v^i(k)$ is the solution of Eq. (4.6.1)-(4.6.2), should be found. $\hat{\theta}_v$ can be calculated by considering that both ω_v and Ω_h are data provided from the sensors.

4.6.1.2 – Identification Stage 6

It has been noticed experimentally that the coupling effect between the tail propeller and the vertical aeromechanical behaviour of the TRMS beam can be ignored, as for each value of u_h between its minimum value and its maximum value the TRMS has shown to remain in its equilibrium vertical angle θ_v^0. Hence, $k_t = 0$ is a good approximation of the real value of this constant.

4.6.2 – Results

The identification procedure has been applied to 10 sets of data. It must be noted that while all the sets of data can be used to identify k_{ov}, numerical values for k_{fvp} can be obtained only from those sets of data where the angular speed of the main rotor ω_v is positive. Analogously, numerical values for k_{fvn} can be obtained only from those sets of data where the angular speed of the main rotor ω_v is negative.

Each set of data consists of 6000 samples, corresponding to an experiment that lasts 60 seconds, with a sample period of 0.01 s, obtained from the response of the TRMS to an input voltage step signal. The sets of inputs used in this identification stage are briefly resumed in ***Table 14***.

Dataset	1	2	3	4	5	6	7	8	9	10
Step amplitude (0-60s) (V)	0.4	0.6	0.8	1.0	1.2	1.4	-0.7	-0.8	-0.9	-1.0

Table 14 – Main input voltages u_v for vertical aeromechanical identification

Eqs. (4.6.5) and (4.6.6) have been used to obtain a preliminary estimation of k_{fvp} and k_{fvn}, as shown in ***Table 15***:

Dataset	1	2	3	4	5	6
k_{fvp}	0.3161	0.3873	0.4261	0.4220	0.4301	0.4339
Dataset	7	8	9	10		
k_{fvn}	0.2392	0.2435	0.2515	0.3461		

Table 15 – Estimation of k_{fvp} and k_{fvn}

The minimum of (4.6.7) can be found by means of *fmincon*. The results obtained in the previous stage can be used to impose constraints on the unknown parameters:

$$k_{fvp} \in \left[0.3161\, kg\, m^2\, s^{-2}\, V^{-2}, 0.4339\, kg\, m^2\, s^{-2}\, V^{-2}\right]$$

$$k_{fvn} \in \left[0.2392\, kg\, m^2\, s^{-2}\, V^{-2}, 0.3461\, kg\, m^2\, s^{-2}\, V^{-2}\right]$$

$$k_{ov} \in \left[10^{-3}\ kg\, m^2\, s^{-1}, 10\, kg\, m^2\, s^{-1}\right]$$

Under the previous considerations, the *fmincon* has returned the following values:

$$k_{fvp} = 0.4311\, kg\, m^2\, s^{-2}\, V^{-2}$$

$$k_{fvn} = 0.2850\, kg\, m^2\, s^{-2}\, V^{-2}$$

$$k_{ov} = 0.0326\, kg\, m^2\, s^{-1}$$

A comparison between the real data and the simulation of Eqs. (4.6.1)-(4.6.2) with such values is shown in ***Fig. 13*** and ***Fig. 14***. ***Table 16*** shows the maximum absolute errors, the mean absolute errors and the steady-state error for each dataset. The results prove to be satisfactory, and it is hard to improve them without modifying the non-linear model of the system, including some non-linear effects that have been neglected.

Dataset	Max Absolute Error	Mean Absolute Error	Steady-State error
1	4.76º	2.13º	1.95º
2	6.10º	1.32º	1.32º
3	8.45º	0.64º	0.17º
4	12.05º	0.93º	0.48º
5	13.87º	0.81º	0.06º
6	17.62º	1.35º	0.45º
7	5.60º	2.02º	1.91º
8	5.73º	2.14º	1.97º
9	6.39º	2.13º	1.83º
10	6.99º	2.00º	1.67º

Table 16 – Max, Mean and Steady-State errors for Identification Stage 5

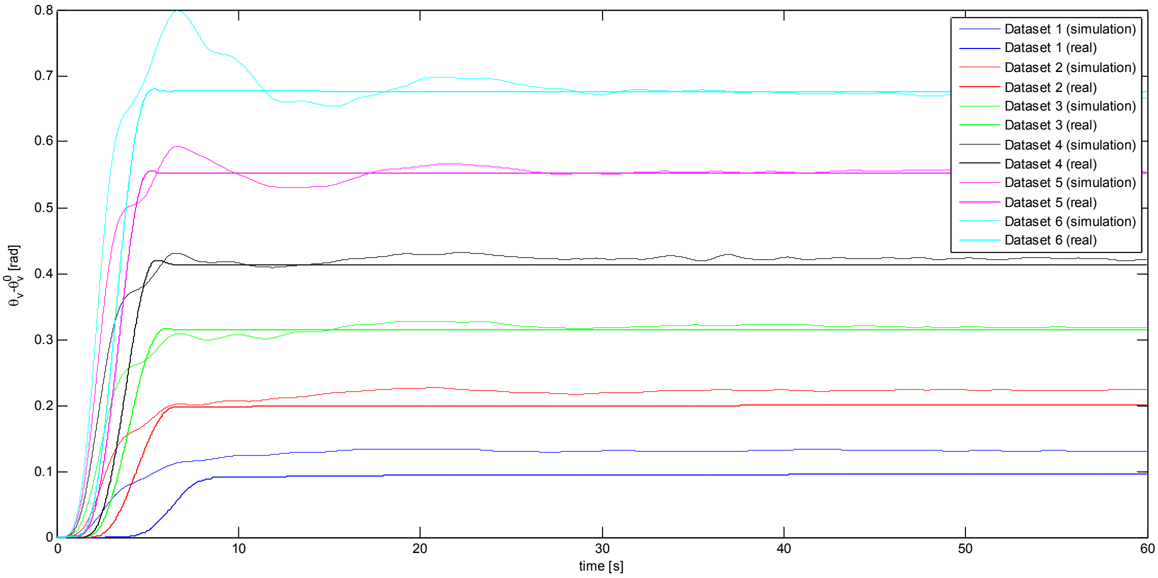

Fig. 13 - Comparison between real and simulation data – Identification Stage 5 (positive u_v)

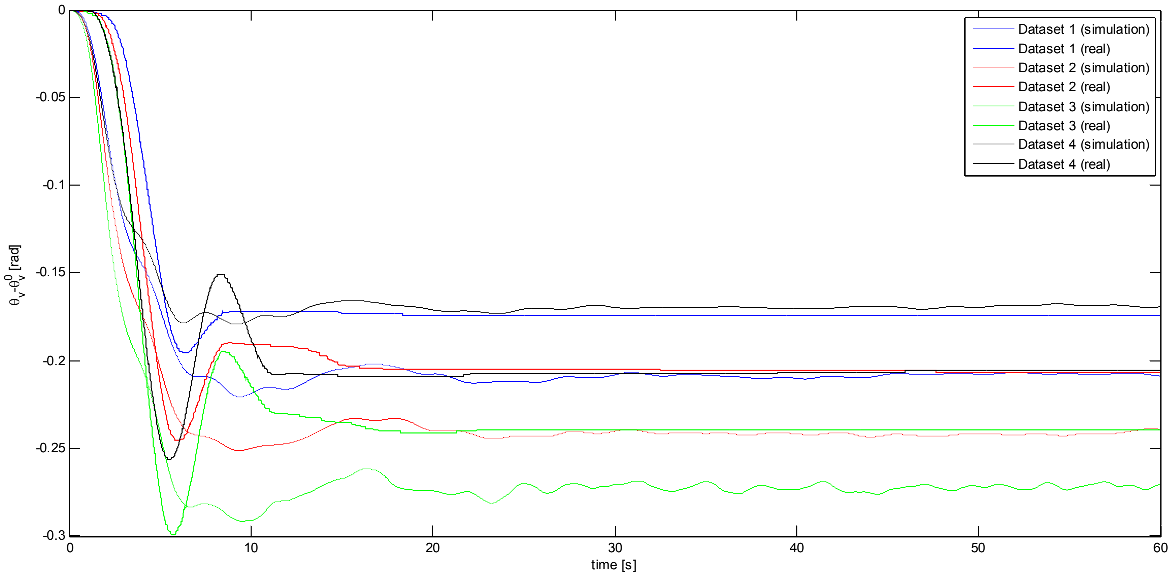

Fig. 14 - Comparison between real and simulation data – Identification Stage 5 (negative u_v)

4.7 – Validation and remarks

In order to validate the values obtained for each parameter by means of identification, such results have been tested with various inputs. The simulation of the non-linear model has been compared to the data obtained from the real system in two different scenarios. In both scenarios, the input voltages for both the main and the tail propeller are sinusoidal with mean value different from zero (see ***Table 17***).

Scenario	Input	Mean value	Sine amplitude	Sine frequency
1	u_h	0.7 V	0.3 V	0.1 Hz
1	u_v	0.7 V	0.3 V	0.05 Hz
2	u_h	0.6 V	0.3 V	0.1 Hz
2	u_v	0.6 V	0.1 V	0.05 Hz

Table 17 – Input voltages for validation of the TRMS model

The comparisons between the simulation data and the real data for the 1st scenario are shown in ***Fig. 15***, ***Fig. 16***, ***Fig. 17*** and ***Fig. 18***.

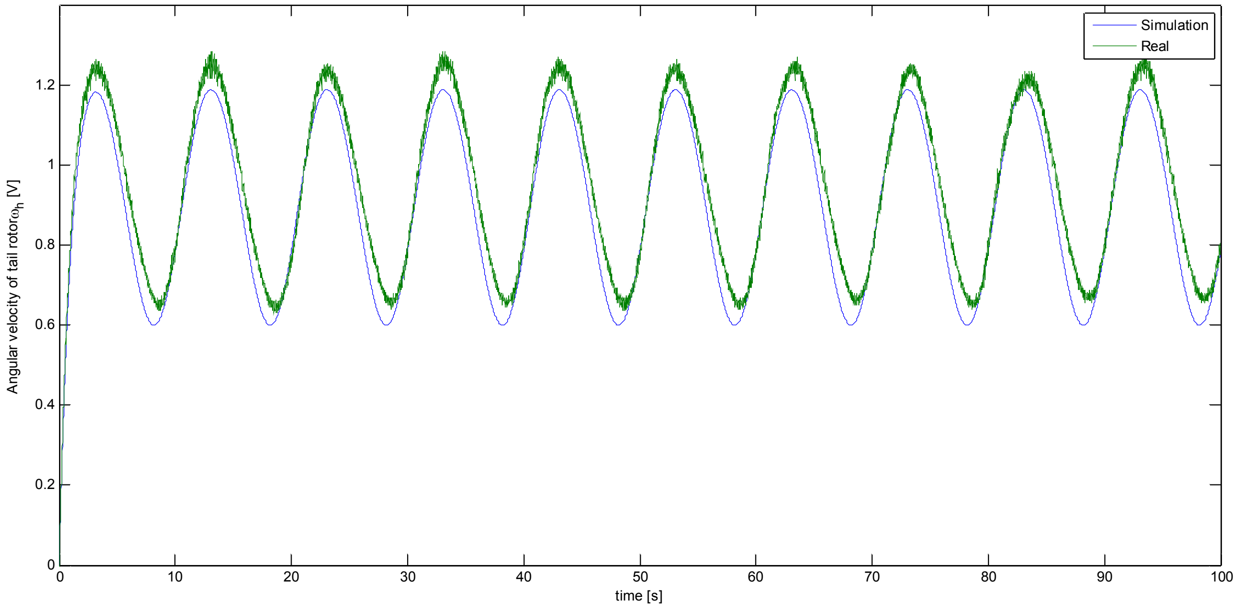

Fig. 15 – Angular velocity of tail rotor in validation scenario 1

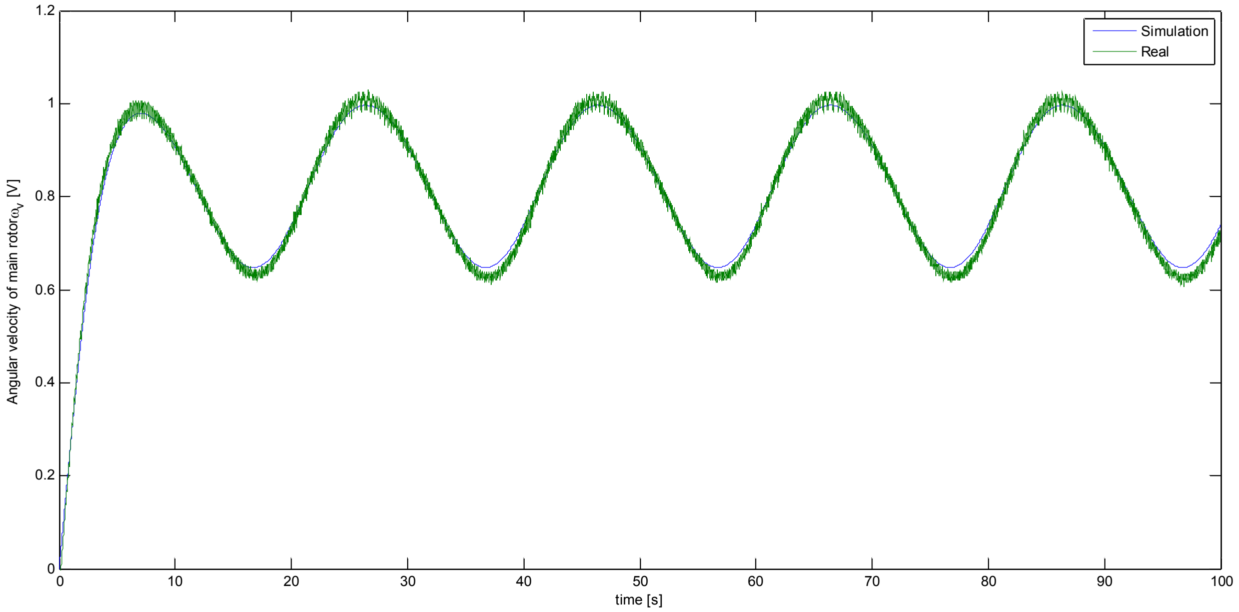

Fig. 16 - Angular velocity of main rotor in validation scenario 1

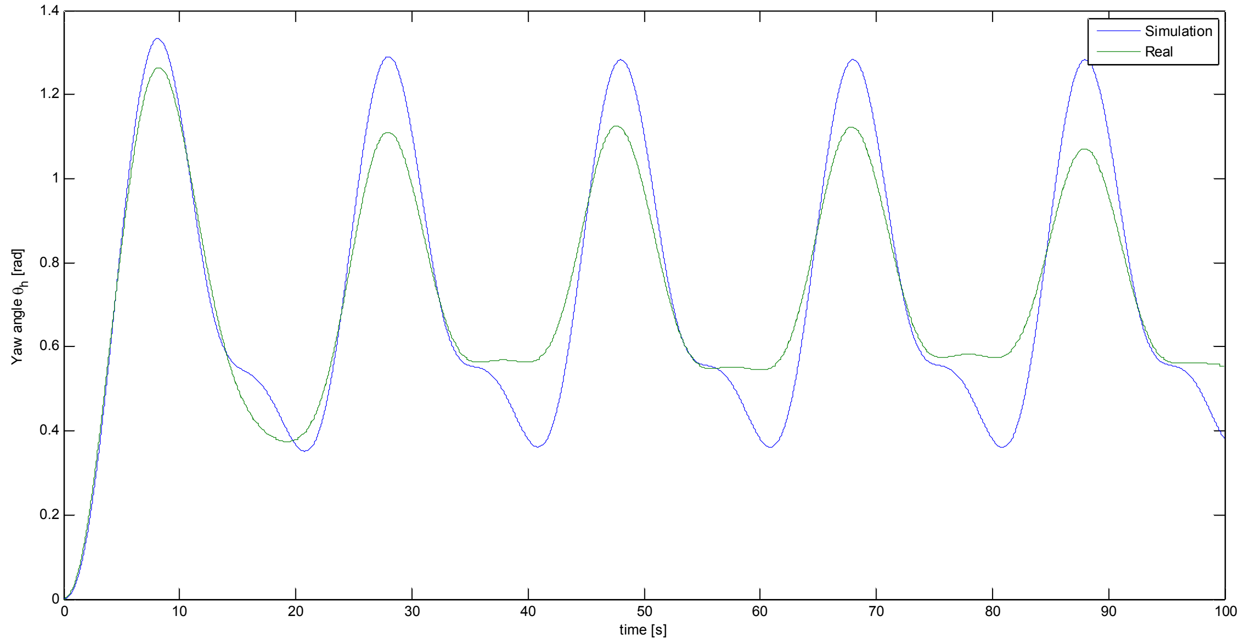

Fig. 17 – Yaw angle of the beam in validation scenario 1

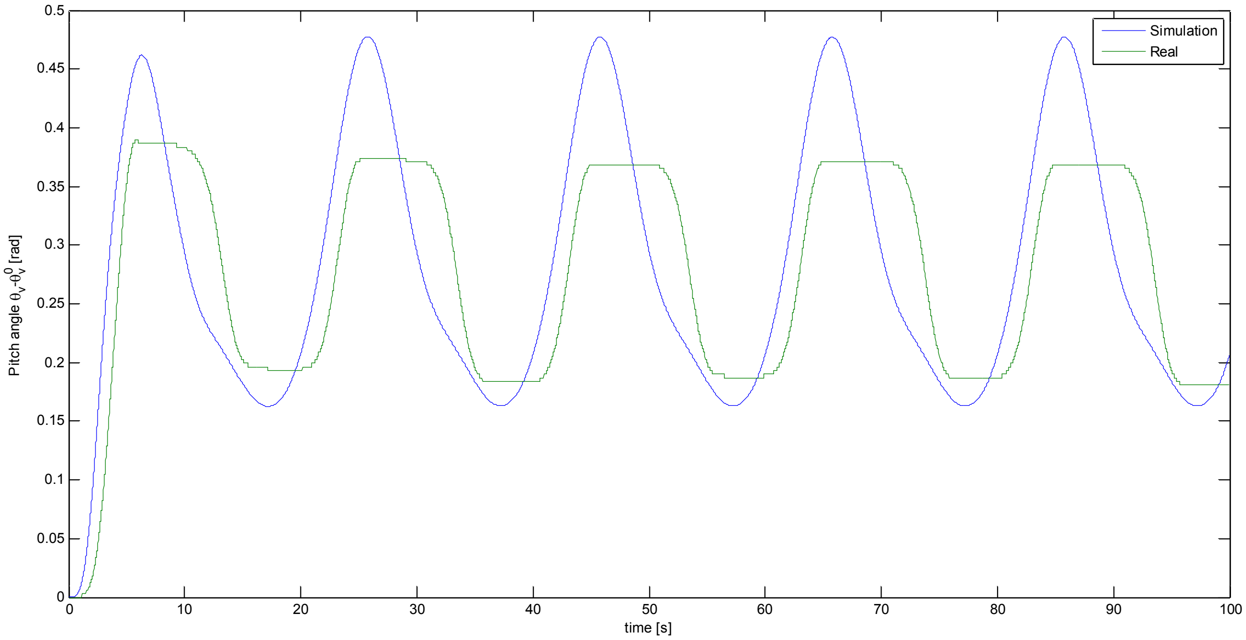

Fig. 18 – Pitch angle θ_v with respect to its equilibrium θ_v^0 in validation scenario 1

Analogously, comparisons between the simulation data and the real data for the 2^{nd} scenario are shown in ***Fig. 19***, ***Fig. 20***, ***Fig. 21*** and ***Fig. 22***.

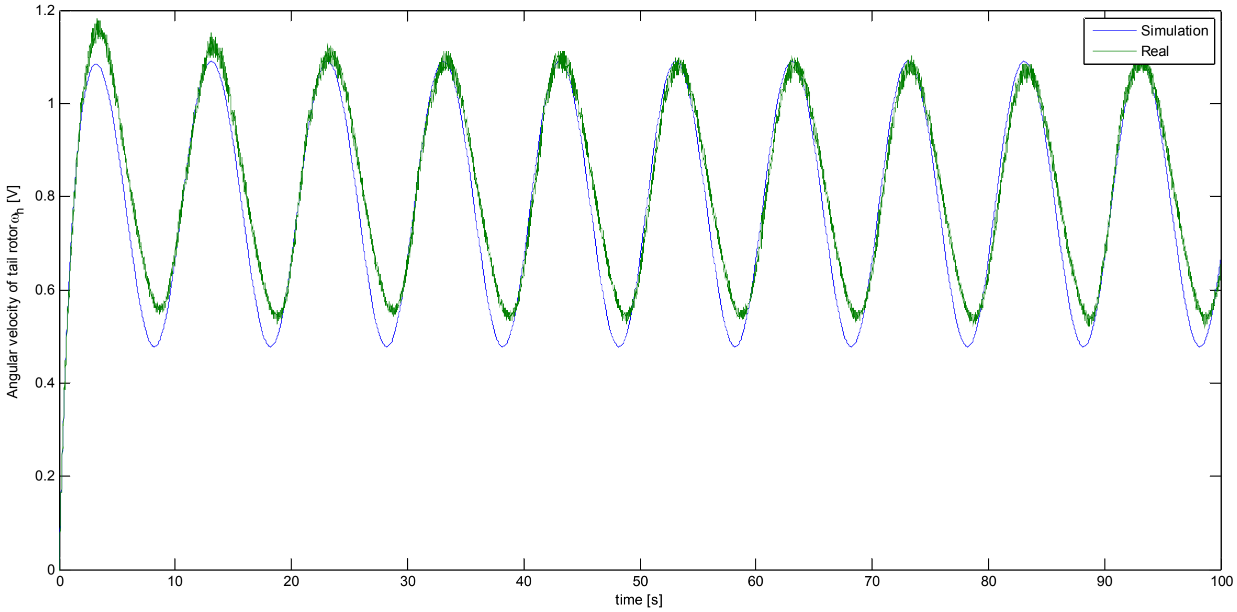

Fig. 19 - Angular velocity of tail rotor in validation scenario 2

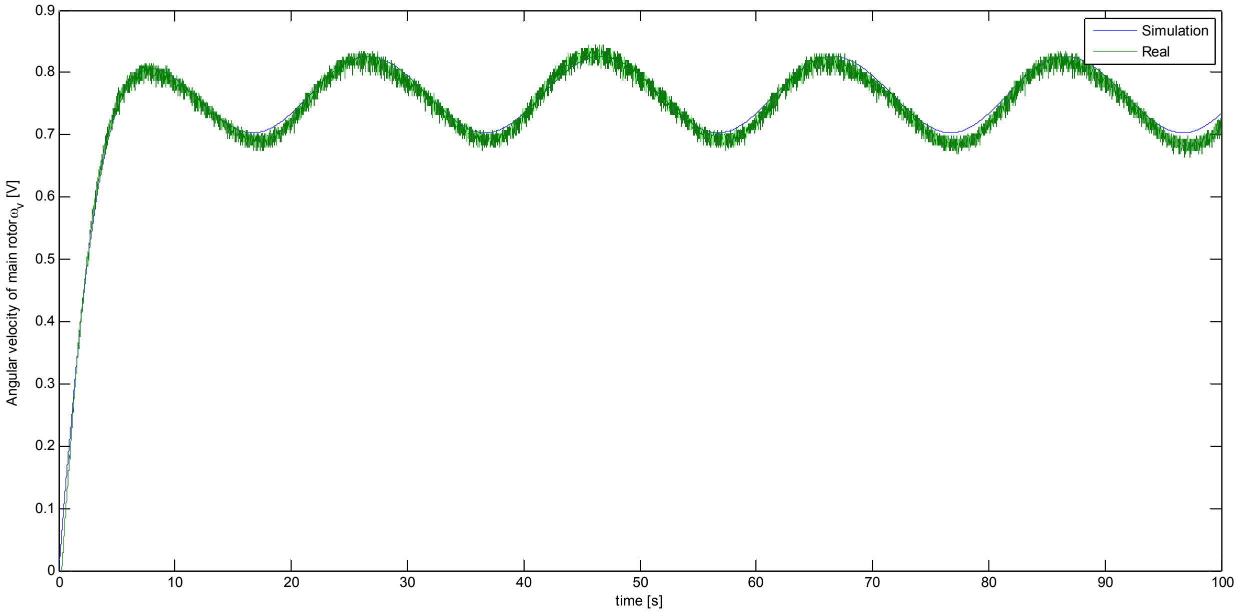

Fig. 20 - Angular velocity of main rotor in validation scenario 2

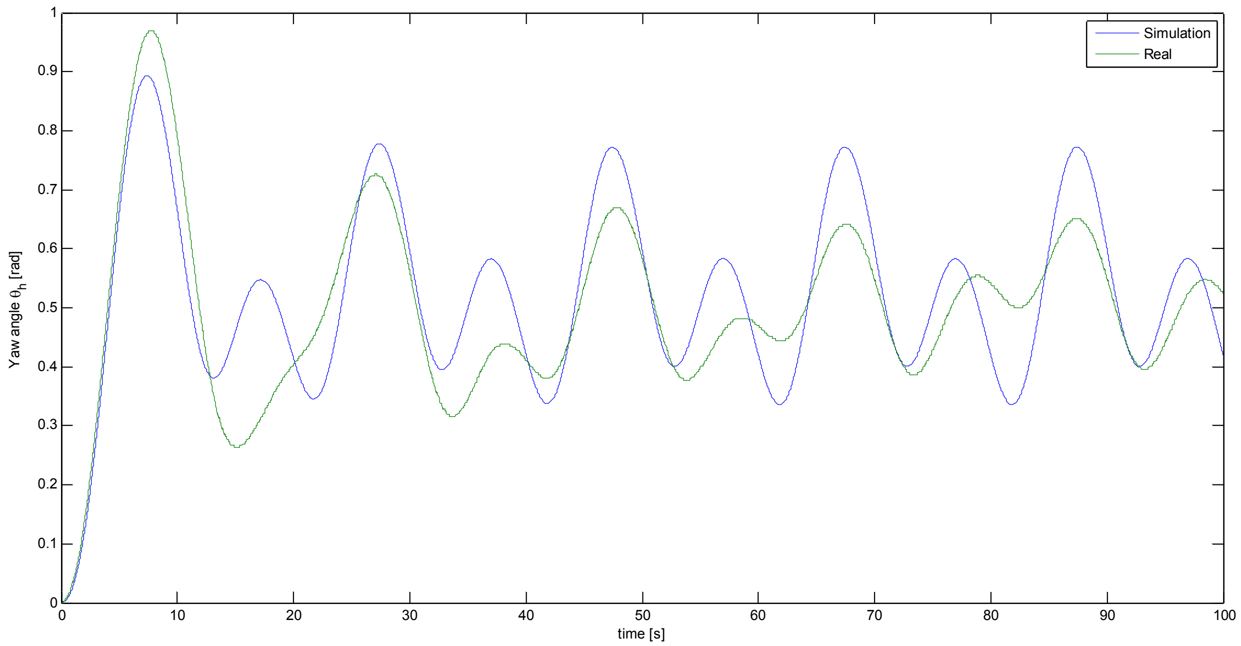

Fig. 21 - Yaw angle in validation scenario 2

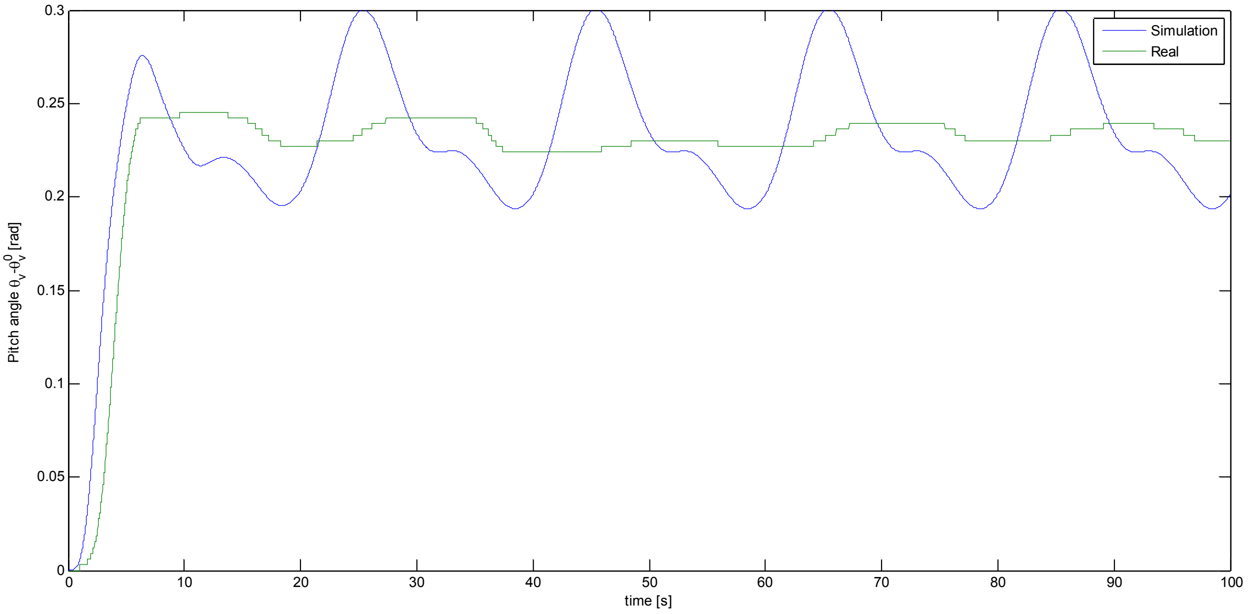

Fig. 22 - Pitch angle θ_v with respect to its equilibrium θ_v^0 in validation scenario 2

The results prove to be quite satisfactory. The validation shows that the model approximates the behaviour of both the propellers quite accurately. Nevertheless, the model and the real system show slightly different behaviours when the yaw and the pitch angle responses are compared. This is not a big problem, as the control will compensate imperfections in the model.

The goodness of the model could be improved through a better modelling of the non-linearities. This will not be done in this thesis, as it deals with LPV modelling and control, and such non-linearities belong to the class of non-linearities that cannot be modelled in an LPV way.

However, in the following, some advices are given on how the model could be improved by considering such non-linearities.

Although the model approximates quite well the behaviour of both the propellers, the two behaviours diverge when the angular velocities of either the main or the tail propeller are near to zero. This is because the non-linear Coulomb friction and the stiction have been neglected. Considering these effects, the quality of the model will be improved.

In **[Rahideh, 2007]**, the torque of the friction force, covering viscous, Coulomb and static frictions, is considered to have a piecewise linear behaviour. When the angular velocities of the beam Ω_h and Ω_v are near zero, the gain between the angular velocity and the torque is higher, as shown in ***Fig. 23***. These two effects have not been considered in our model, but considering them would improve the goodness of the model, even though it would complicate the identification of the parameters too.

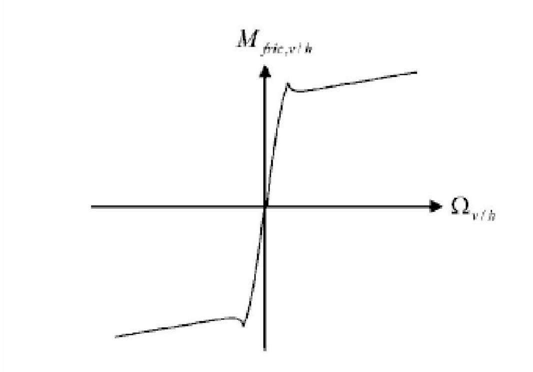

Fig. 23 – Torque of the friction force

4.8 – Construction of the polytopic LPV model

Once the parameters of the non linear model have been identified, a polytopic LPV model can be obtained by applying the procedure that has been described in the previous chapter. Each of the state variables that influence the values of the elements of the state-space matrices is assumed to take values in an interval. The extremes of these intervals are imposed by limitations on the input voltages that can be applied to the tail and to the main rotor and by physical constraints on the minimum and maximum values that the yaw angle and the pitch angle can take. Such extremes can either be found by simulations of the non-linear model ($\underline{\omega}_h, \overline{\omega}_h, \underline{\omega}_v, \overline{\omega}_v, \underline{\Omega}_h, \overline{\Omega}_v$) or by measuring the minimum and maximum yaw and pitch angles that the physical constraints allow the TRMS beam to reach ($\underline{\theta}_h, \overline{\theta}_h, \underline{\theta}_v, \overline{\theta}_v$).

In the following, it will be considered that $u_h \in [-2.5\,V, 2.5\,V]$ and $u_v \in [-2.5\,V, 2.5\,V]$, leading to:

$$\omega_h \in [-3.5\,V, 2.5\,V]$$

$$\Omega_h \in [-2\,rad/s, 2\,rad/s]$$

$$\omega_v \in [-1.8\,V, 1.8\,V]$$

At the same time, the physical structure of the TRMS imposes that:

$$\theta_h \in [-3.4\,rad, 2.4\,rad]$$

$$\theta_v \in [0.7\,rad, 1.7\,rad]$$

As some of the LPV parameters are not defined when $\omega_h = 0$, and they take very high values when ω_h is near zero, some other limits on ω_h will be imposed, that is:

$$\omega_h \in [-3.5\,V, \omega_h^-] \cup [\omega_h^+, 2.5\,V]$$

with $\omega_h^- = -0.3\,V$ and $\omega_h^+ = 0.3\,V$.

The same is true for ω_v and θ_v, thus leading to:

$$\omega_v \in \left[-1.8\,V, \omega_v^-\right] \cup \left[\omega_v^+, 1.8\,V\right]$$
$$\theta_v \in \left[-0.7\,rad, \theta_v^-\right] \cup \left[\theta_v^+, 1.7\,rad\right]$$

with $\omega_v^- = -0.05\,V$ and $\omega_v^+ = 0.05\,V$, $\theta_v^- = \theta_v^0 - 0.01 rad$ and $\theta_v^+ = \theta_v^0 + 0.01 rad$.
All the models have been obtained with a sample time $T_s = 0.01 s$.

4.8.1 – Polytopic Absolute LPV Model

The limits lead to the extreme values for each parameter as defined in ***Table 18***. $\underline{a}_{ij}^{absd}$ and $\overline{a}_{ij}^{absd}$ have been calculated according to formulas in section **[3.3.3.1]**. $\tilde{a}_{ij}^{absd}$ is the mean value of each parameter. It has been calculated by dividing the region of the admissible states in the state-space in a grid and by calculating a_{ij}^{absd} at each point of the grid. Afterwards, the mean $\tilde{a}_{ij}^{absd}$ of all the obtained values has been calculated. Finally, Δa_{ij}^{absd} is the variation of each LPV parameter, calculated as its maximum value minus its minimum value.

	$\underline{a}_{ij}^{absd}$ *or* $\underline{b}_{ij}^{absd}$	$\overline{a}_{ij}^{absd}$ *or* $\overline{b}_{ij}^{absd}$	$\tilde{a}_{ij}^{absd}$ *or* $\tilde{b}_{ij}^{absd}$	Δa_{ij}^{absd} *or* Δb_{ij}^{absd}
a_{11}^{absd}	0.9840	0.9903	0.9881	0.0063
a_{21}^{absd}	2.5888e-4	0.0048	0.0017	0.0045
a_{22}^{absd}	0.9895	0.9975	0.9949	0.0079
a_{23}^{absd}	-0.0058	-7.0366e-4	-0.0020	0.0051
a_{24}^{absd}	-0.0013	-1.1549e-4	-6.6765e-4	0.0012
a_{25}^{absd}	-0.0075	0.0070	-8.5261e-6	0.0145
a_{26}^{absd}	-0.0045	0.0156	0.0018	0.0200
a_{44}^{absd}	0.9947	0.9994	0.9971	0.0047
a_{51}^{absd}	0	0	0	0
a_{52}^{absd}	-0.0092	0.0254	0.0060	0.0346
a_{54}^{absd}	0	0.0247	0.0109	0.0247
a_{55}^{absd}	0.9949	0.9949	0.9949	0
a_{56}^{absd}	-0.0412	-0.0240	-0.0372	0.0171
b_{11}^{absd}	0.0163	0.0163	0.0163	0
b_{22}^{absd}	7.6242e-4	9.7038e-4	8.7045e-4	2.0796e-4
b_{42}^{absd}	0.0038	0.0038	0.0038	0
b_{51}^{absd}	0	0	0	0

Table 18 – Extreme values, mean and variation of the Absolute LPV parameters

Notice that the variability of the parameter b_{22}^{absd} is very small; hence, the matrix $\mathbf{B}_{abs}(\mathbf{x}(t))$ can be approximated with a constant matrix, where $b_{22} = \tilde{b}_{22}^{absd}$, and LPV design techniques that make use of LMI regions can be applied.

4.8.2 – Polytopic Shamma LPV Model

These limits lead to the extreme values for each parameter as defined in ***Table 19***. $\underline{a}_{ij}^{shad}$ and $\overline{a}_{ij}^{shad}$ have been calculated according to formulas in section **[3.3.3.2]**. $\tilde{a}_{ij}^{shad}$ is the mean value of each parameter. It has been calculated by dividing the region of the admissible states in the state-space in a grid and by calculating a_{ij}^{shad} at each point of the grid. Afterwards, the mean $\tilde{a}_{ij}^{shad}$ of all the obtained values has been calculated. Finally, Δa_{ij}^{shad} is the variation of each LPV parameter, calculated as its maximum value minus its minimum value.

	$\underline{a}_{ij}^{shad}$ *or* $\underline{b}_{ij}^{shad}$	$\overline{a}_{ij}^{shad}$ *or* $\overline{b}_{ij}^{shad}$	$\tilde{a}_{ij}^{shad}$ *or* $\tilde{b}_{ij}^{shad}$	Δa_{ij}^{shad} *or* Δb_{ij}^{shad}
a_{11}^{shad}	0.6802	0.8040	0.7625	0.1238
a_{12}^{shad}	-4.3202	-0.0375	-0.3976	4.2827
a_{15}^{shad}	-44.7408	56.3527	2.2551	101.0935
a_{21}^{shad}	0.0052	0.0955	0.0342	0.0903
a_{22}^{shad}	0.7907	0.9496	0.9067	0.1589
a_{24}^{shad}	-0.0269	-0.0029	-0.0131	0.0241
a_{25}^{shad}	-0.1501	0.1407	-2.27e-4	0.2908
a_{44}^{shad}	0.8939	0.9857	0.9414	0.0917
a_{45}^{shad}	-15.0186	1.9448	-2.0167	16.9634
a_{51}^{shad}	0	0	0	0
a_{52}^{shad}	-0.1777	0.5079	0.1286	0.6855
a_{54}^{shad}	0.0109	0.4950	0.2179	0.4841
b_{22}^{shad}	0.0152	0.0194	0.0171	0.0042

Table 19 - Extreme values, mean and variation of the Absolute LPV parameters

Notice that the variability of the parameter b_{22}^{shad} is very small; hence, the matrix $\mathbf{B}_{sha}(\mathbf{x}(t))$ can be approximated with a constant matrix, where $b_{22} = \tilde{b}_{22}^{shad}$, and LPV design techniques that make use of LMI regions can be applied.

4.9 – Summary

In this chapter the unknown parameters of the model of the Twin-Rotor MIMO System have been identified. The results have been validated and some advices of how the quality of the model could be improved have been given. However, the model has proved to show good performances in approximating the real behaviour of the system.

Finally, an LPV polytopic model has been obtained by calculating the extreme values for each element in the state-space matrices, as explained in the previous chapter.

5 – TRMS LPV CONTROL

5.1 – Polytopic LPV control loop

The LPV system can be controlled by a state feedback control with a tracking reference input as proposed in **[Franklin, 1997]**. The feedback control law is based on the classical state feedback, even though the feedback matrix is designed in an LPV way and thus scheduled by the vector $\boldsymbol{\psi}(k)$:

$$\mathbf{u}(k) = \mathbf{K}\left(\boldsymbol{\psi}(k)\right)\mathbf{x}(k) \tag{5.1.1}$$

A nonlinear function N_{ux} is added to the state feedback control law (5.1.1). The basic idea in determining the function N_{ux} is that it should transform the reference input $\mathbf{r}(k)$, expressed as either a desired value or a trajectory of the system outputs, to a state reference $\mathbf{x}_r(k)$ and a feedforward control action $\mathbf{u}_r(k)$ that correspond to an equilibrium point for this $\mathbf{r}(k)$.
Thus, the control law can be expressed as follows:

$$\mathbf{u}(k) = \mathbf{u}_r(k) + \mathbf{K}\left(\boldsymbol{\psi}(k)\right)\left(\mathbf{x}(k) - \mathbf{x}_r(k)\right) \tag{5.1.2}$$

It is assumed that $\mathbf{x}(k)$ is not available and the estimated state $\hat{\mathbf{x}}(k)$ should be used instead. Consequently, an *LPV state observer* is used to provide such state estimation:

$$\hat{\mathbf{x}}(k+1) = \mathbf{A}\left(\boldsymbol{\psi}(k)\right)\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\left(\boldsymbol{\psi}(k)\right)\left(\hat{\mathbf{y}}(k) - \mathbf{y}(k)\right) \tag{5.1.3}$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k) \tag{5.1.4}$$

where $\hat{\mathbf{x}}(k) \in \mathbb{R}^{n_x}$ and $\hat{\mathbf{y}}(k) \in \mathbb{R}^{n_y}$ are the estimated state and output variables, respectively. The matrix $\mathbf{L}\left(\boldsymbol{\psi}(k)\right) \in \mathbb{R}^{n_x \times n_y}$ is the gain of the state LPV observer.
The overall LPV control scheme is represented in ***Fig. 24***.

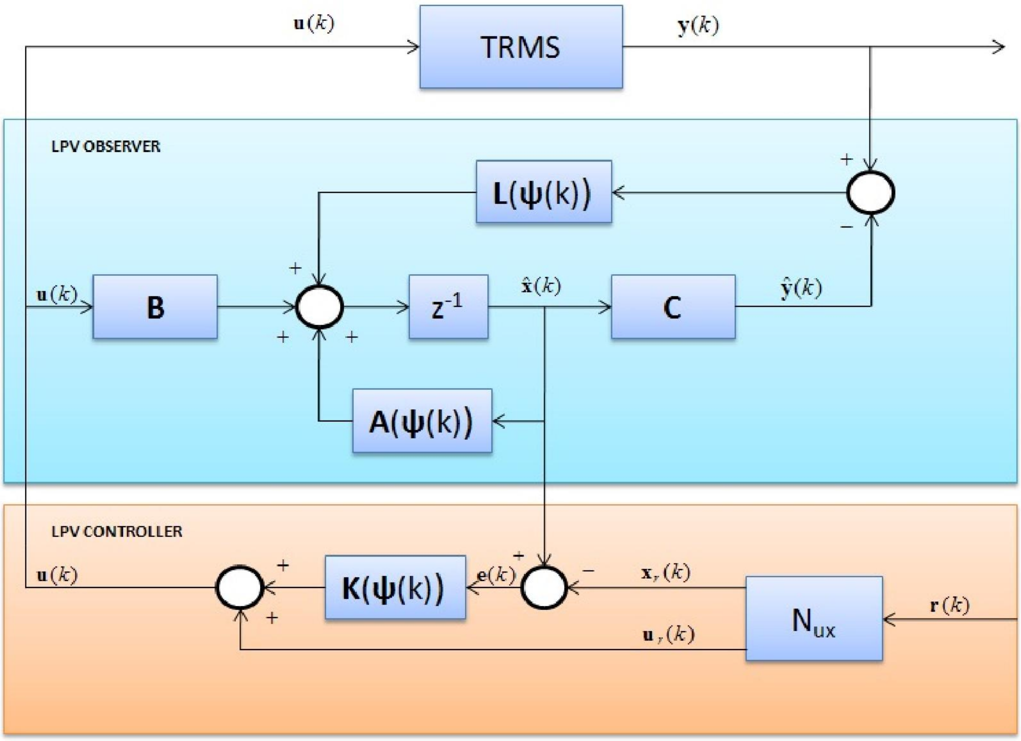

Fig. 24 – LPV Controller/Observer Scheme

The control law (5.1.2) can be implemented using the polytopic representation of the system (2.2.3)-(2.2.4) as:

$$\mathbf{u}(k) = \mathbf{u}_r(k) + \sum_{j=1}^{N} \alpha_k^j \left(\boldsymbol{\psi}(k)\right)\mathbf{K}_j \left(\hat{\mathbf{x}}(k) - \mathbf{x}_r(k)\right) \tag{5.1.5}$$

where $\mathbf{K}_j \in \mathbb{R}^{n_u \times n_x}$ is the controller gain for the j^{th} model, that can be calculated by solving (2.3.13):

$$\begin{cases} \left[\alpha_{kl}\mathbf{X} + \beta_{kl}\left(\mathbf{A}_j\mathbf{X} + \mathbf{B}\boldsymbol{\Gamma}_j\right) + \beta_{lk}\left(\mathbf{A}_j\mathbf{X} + \mathbf{B}\boldsymbol{\Gamma}_j\right)^T\right]_{k,l} < 0 \\ \mathbf{X} > 0 \end{cases} \tag{5.1.6}$$

Analogously, the *LPV state observer* (5.1.3)-(5.1.4) used to estimate $\hat{\mathbf{x}}(k)$ can be implemented using the polytopic representation of the system as:

$$\hat{\mathbf{x}}(k+1) = \sum_{j=1}^{N} \alpha_k^j \left(\boldsymbol{\psi}(k)\right)\left[\mathbf{A}_j\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}_j\left(\mathbf{y}(k) - \hat{\mathbf{y}}(k)\right)\right] \tag{5.1.7}$$

$$\hat{\mathbf{y}}(k) = \mathbf{C}\hat{\mathbf{x}}(k) \tag{5.1.8}$$

where $\hat{\mathbf{x}}(k) \in \mathbb{R}^{n_x}$ and $\hat{\mathbf{y}}(k) \in \mathbb{R}^{n_y}$ are the estimated state and output, respectively. The matrices $\mathbf{L}_j \in \mathbb{R}^{n_x \times n_y}$ are the state observer gains for each j^{th} model, that can be calculated by solving (2.3.13) for the dual system (2.3.15).

5.2 – LPV state observer

In the following some LPV state observers for the TRMS are designed using the LPV models that have been previously obtained and discussed; both the absolute and the Shamma LPV representation are used for design purpose and the results obtained with the real TRMS are compared and commented.

5.2.1 – LPV state observer design

The *LPV state observer* is designed with LMIs (2.3.13) assuming that the eigenvalues are placed in an LMI region that is the intersection between the disk of radius r and center $(-q, 0)$ and a vertical strip defined by the extreme values $S_{\min}$ and $S_{\max}$. The problem is equivalent to that of finding state feedback matrices, one for each vertex system, that place the eigenvalues of the closed-loop dual vertex system in the LMI region, thus leading to the following LMIs:

$$\begin{cases} \begin{pmatrix} -r\mathbf{X} & q\mathbf{X} + \mathbf{A}_j^T\mathbf{X} + \mathbf{C}^T\boldsymbol{\Gamma}_j & 0 & 0 \\ q\mathbf{X} + \mathbf{X}\mathbf{A}_j + \boldsymbol{\Gamma}_j^T\mathbf{C} & -r\mathbf{X} & 0 & 0 \\ 0 & 0 & S_{\min}\mathbf{X} - \frac{1}{2}\left(\mathbf{A}_j^T\mathbf{X} + \mathbf{X}\mathbf{A}_j + \mathbf{C}^T\boldsymbol{\Gamma}_j + \boldsymbol{\Gamma}_j^T\mathbf{C}\right) & 0 \\ 0 & 0 & 0 & -S_{\max}\mathbf{X} + \frac{1}{2}\left(\mathbf{A}_j^T\mathbf{X} + \mathbf{X}\mathbf{A}_j + \mathbf{C}^T\boldsymbol{\Gamma}_j + \boldsymbol{\Gamma}_j^T\mathbf{C}\right) \end{pmatrix} < 0 \\ \mathbf{X} > 0 \end{cases} \tag{5.2.1}$$

The vertices state observer gains are then obtained as:

$$\mathbf{L}_j = \left(\boldsymbol{\Gamma}_j\mathbf{X}^{-1}\right)^T \tag{5.2.2}$$

It must be noted that the application of the LPV theory would need to solve LMIs (5.2.1) for 2^N vertices, that is, $2^N + 1$ LMIs where N is the number of varying parameters. This leads to some computational issues, as the time needed to solve such a number of LMIs could be quite big, as it increases in an exponential way with N, as resumed in ***Table 20***. Hence, the design of the LPV observer is made by considering that only the $\tilde{N}$ most varying parameters vary, and all the others are fixed at their mean value.

Number of varying parameters	Time needed to solve LMIs
1	$\approx 4s$
2	$\approx 6s$
3	$\approx 14s$
4	$\approx 38s$
5	$\approx 98s$
6	$\approx 327s$
7	$\approx 1270s$
8	$\approx 7150s$

Table 20 – Time needed to solve the LMIs when the number of varying parameters of the polytopic model varies

5.2.2 – LPV state observer results

The LMI region where the observer eigenvalues are placed should be chosen in such a way that the observer is fast enough to track the data coming from the sensors without loss of information. As Matlab simulations are done assuming a perfect knowledge of the TRMS (that is, the system and its model match perfectly) and absence of noise in the data coming from the sensors, a comparison between the observers is done applying them to the real system, where the observer should be able to filter the noise in the data. Indeed, the best observer has been chosen by means of a *trial and error* procedure done on the real TRMS, where the region to place the observer poles has been moved until good performances have been obtained.

Due to computational issues that have been described previously, the design of the LPV state observer has been implemented by considering that only the $\tilde{N}$ most varying parameters vary. The selection $\tilde{N} = 6$ has shown to be a good compromise between computation time and performance.

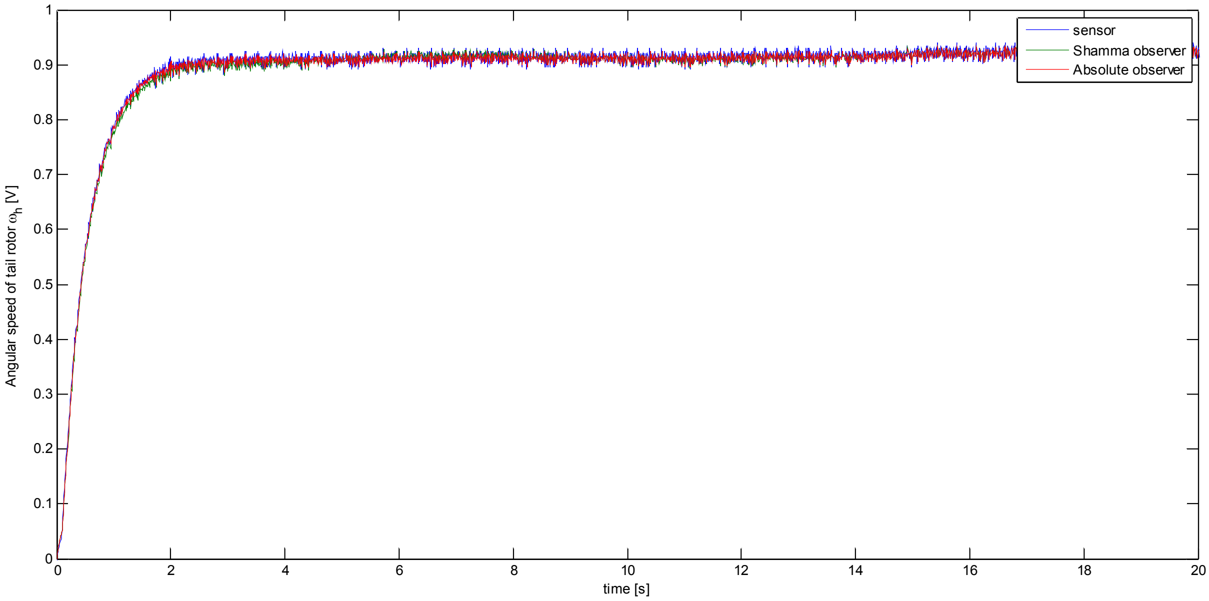

Fig. 25 – Comparison between sensor and LPV observer estimation – Angular speed of tail rotor

Choosing $\tilde{N}=6$ means that the vector of scheduling parameters is $\psi_{abs}=\left[a_{22}^{absd},a_{25}^{absd},a_{26}^{absd},a_{52}^{absd},a_{54}^{absd},a_{56}^{absd}\right]$ for the Absolute LPV model and $\psi_{sha}=\left[a_{12}^{shad},a_{15}^{shad},a_{25}^{shad},a_{45}^{shad},a_{52}^{shad},a_{54}^{shad}\right]$ for the Shamma LPV model, respectively.

Good results and satisfactory performances are obtained when the LMI region is chosen as a disk with center 0.2 and radius 0.1, that is: $q=-0.2$, $r=0.1$, $S_{\min}=0.1$ and $S_{\max}=0.3$.

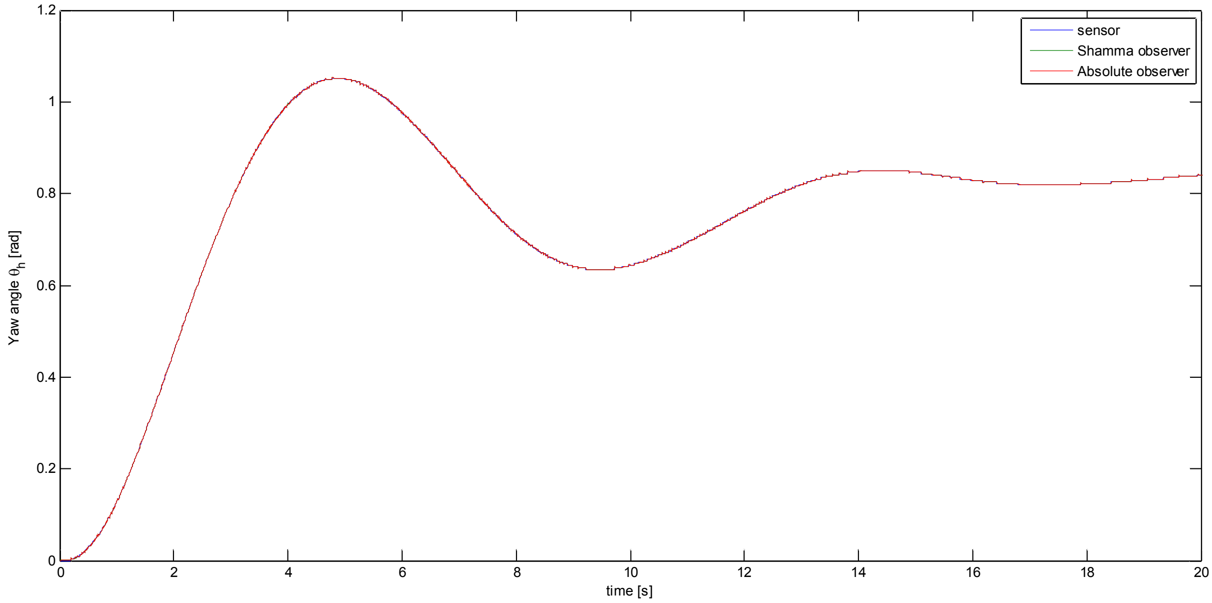

Fig. 26 – Comparison between sensor and LPV observer estimation – Yaw angle

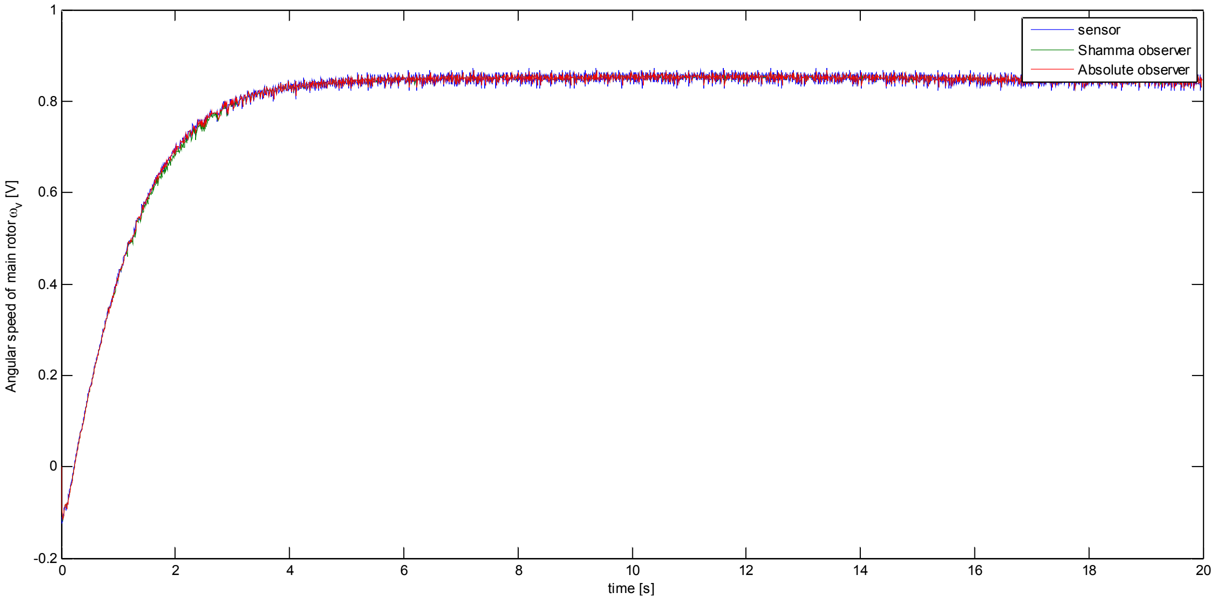

Fig. 27 – Comparison between sensor and LPV observer estimation – Angular speed of main rotor

Fig. 25, ***Fig. 26***, ***Fig. 27*** and ***Fig. 28*** show the behaviour of the observers when no state feedback control acts on the TRMS. The both observers show very good performances. Actually, this result is not surprising, as the noise corrupting the data collected from the sensors is quite small. Anyhow, the presence of an observer in the control loop becomes essential when a fault appears, as it will be shown in next chapters.

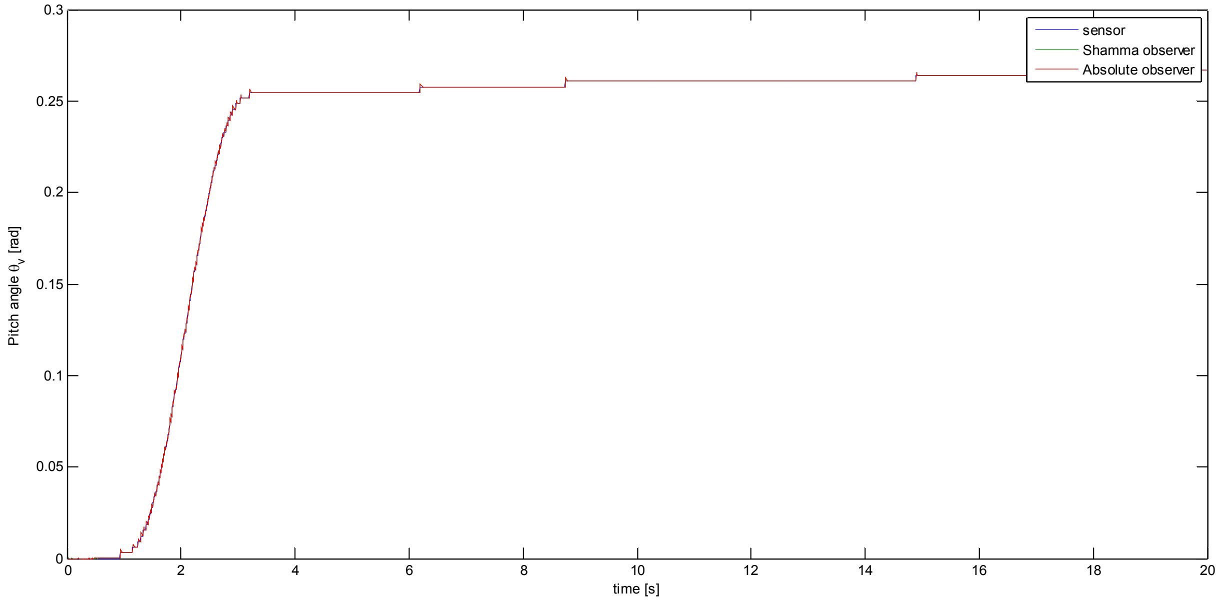

Fig. 28 - Comparison between sensor and LPV observer estimation – Pitch angle

The observers designed using, respectively, the absolute LPV model and the Shamma LPV model show good tracking performance. However, it has been noticed that the computation time needed to solve the LMIs for the Shamma model is 3-4 times bigger than the computation time needed for the absolute model. As the solution of the LMIs is obtained offline, this is not an important issue when comparing the efficiency of the two methods to design an LPV observer.

5.3 – 6-States-based LPV Controller

In the following some LPV state controllers for the TRMS are designed using the LPV models that have been previously obtained and discussed; both the absolute and the Shamma LPV representation are used for design purpose and the results obtained are compared and commented.

5.3.1 – The N_{ux} function

The N_{ux} function transforms the reference trajectory $\mathbf{r}(k)$ to a state reference $\mathbf{x}_r(k)$ and a feedforward control action $\mathbf{u}_r(k)$ that correspond to an equilibrium point for the reference $\mathbf{r}(k)$. The trajectory is given by a desired yaw angle $\theta_h^{ref}(k)$ and a desired pitch angle $\theta_v^{ref}(k)$. These desired angles, along with the actual angles $\theta_h(k)$ and $\theta_v(k)$, are used to obtain a reference for the horizontal and vertical velocities $\Omega_h^{ref}(k)$ and $\Omega_v^{ref}(k)$. The desired horizontal and vertical velocities are considered to be the velocities needed to drive the system from the actual angles $\theta_h(k)$ and $\theta_v(k)$ to the desired ones $\theta_h^{ref}(k)$ and $\theta_v^{ref}(k)$ in the times τ_h and τ_v, respectively:

$$\Omega_h^{ref}(k) = \left(\theta_h^{ref}(k) - \theta_h(k)\right) / \tau_h \tag{5.3.1}$$

$$\Omega_v^{ref}(k) = \left(\theta_v^{ref}(k) - \theta_v(k)\right) / \tau_v \tag{5.3.2}$$

The state reference $\mathbf{x}_r(k)$ is given by:

$$\mathbf{x}_r(k) = \left[\omega_h^{ref}(k) \;\; \Omega_h^{ref}(k) \;\; \theta_h^{ref}(k) \;\; \omega_v^{ref}(k) \;\; \Omega_v^{ref}(k) \;\; \theta_v^{ref}(k)\right]^T \tag{5.3.3}$$

while the feedforward action $\mathbf{u}_r(k)$ is given by:

$$\mathbf{u}_r(k)=\left[u_h^{ref}(k)\ \ u_v^{ref}(k)\right] \tag{5.3.4}$$

where $\omega_h^{ref}(k)$, $\omega_v^{ref}(k)$, $u_h^{ref}(k)$ and $u_v^{ref}(k)$ can be obtained from the set of non-linear differential equations that describes the behaviour of the TRMS by imposing all derivatives equal to zero and $\Omega_{h/v}(k)=\Omega_{h/v}^{ref}(k)$, $\theta_{h/v}(k)=\theta_{h/v}^{ref}(k)$:

$$\begin{cases}
\dfrac{k_a k_1}{J_{tr}R_a}u_h^{ref}-\left(\dfrac{B_{tr}}{J_{tr}}+\dfrac{k_a^2}{J_{tr}R_a}\right)\omega_h^{ref}-\dfrac{f_1(\omega_h^{ref})}{J_{tr}}=0 \\
\dfrac{l_t f_2\left(\omega_h^{ref}\right)\cos\theta_v^{ref}-k_{oh}\Omega_h^{ref}-f_3\left(\theta_h^{ref}\right)+f_6\left(\theta_v^{ref}\right)}{D\cos^2\theta_v^{ref}+E\sin^2\theta_v^{ref}+F} \\
+\dfrac{k_m\omega_v^{ref}\sin\theta_v^{ref}\Omega_v^{ref}\left(D\cos^2\theta_v^{ref}-E\sin^2\theta_v^{ref}-F-2E\cos^2\theta_v^{ref}\right)}{\left(D\cos^2\theta_v^{ref}+E\sin^2\theta_v^{ref}+F\right)^2} \\
+\dfrac{k_m\cos\theta_v^{ref}\left(\dfrac{k_a k_2}{R_a}u_v^{ref}-\left(B_{mr}+\dfrac{k_a^2}{R_a}\right)\omega_v^{ref}-f_4\left(\omega_v^{ref}\right)\right)}{J_{mr}\left(D\cos^2\theta_v^{ref}+E\sin^2\theta_v^{ref}+F\right)}=0 \\
\dfrac{k_a k_2}{J_{mr}R_a}u_v^{ref}-\left(\dfrac{B_{mr}}{J_{mr}}+\dfrac{k_a^2}{J_{mr}R_a}\right)\omega_v^{ref}-\dfrac{f_4(\omega_v^{ref})}{J_{mr}}=0 \\
\dfrac{l_m f_5\left(\omega_v^{ref}\right)+k_g\Omega_h^{ref}f_5\left(\omega_v^{ref}\right)\cos\theta_v^{ref}-k_{ov}\Omega_v^{ref}-\left(\Omega_h^{ref}\right)^2 H\sin\theta_v^{ref}\cos\theta_v^{ref}}{J_v} \\
+\dfrac{g\left(\left(A-B\right)\cos\theta_v^{ref}-C\sin\theta_v^{ref}\right)}{J_v}+\dfrac{k_t\left(\dfrac{k_a k_1}{R_a}u_h^{ref}-\left(B_{tr}+\dfrac{k_a^2}{R_a}\right)\omega_h^{ref}-f_1(\omega_h^{ref})\right)}{J_v J_{tr}}=0
\end{cases} \tag{5.3.5}$$

Solving (5.3.5) leads to the solution:

$$f_5\left(\omega_v^{ref}\right)=\frac{g\left(C\sin\theta_v^{ref}+(B-A)\cos\theta_v^{ref}\right)+k_{ov}\Omega_v^{ref}+\left(\Omega_h^{ref}\right)^2 H\sin\theta_v^{ref}\cos\theta_v^{ref}}{l_m+k_g\Omega_h^{ref}\cos\theta_v^{ref}} \tag{5.3.6}$$

$$f_2\left(\omega_h^{ref}\right)=\frac{f_3\left(\theta_h^{ref}\right)-f_6\left(\theta_v^{ref}\right)+k_{oh}\Omega_h^{ref}-\frac{k_m\omega_v^{ref}\sin\theta_v^{ref}\Omega_v^{ref}\left(D\cos^2\theta_v^{ref}-E\sin^2\theta_v^{ref}-F-2E\cos^2\theta_v^{ref}\right)}{D\cos^2\theta_v^{ref}+E\sin^2\theta_v^{ref}+F}}{l_t\cos\theta_v^{ref}} \tag{5.3.7}$$

$$u_h^{ref}=\frac{\left(B_{tr}R_a+k_a^2\right)\omega_h^{ref}+f_1\left(\omega_h^{ref}\right)R_a}{k_a k_1} \tag{5.3.8}$$

$$u_v^{ref}=\frac{\left(B_{mr}R_a+k_a^2\right)\omega_v^{ref}+f_4\left(\omega_v^{ref}\right)R_a}{k_a k_2} \tag{5.3.9}$$

5.3.2 – LPV state controller design

The *LPV state controller* is designed with LMIs (2.3.13) assuming that the eigenvalues are placed in an LMI region that is the intersection between a disk of radius r and center $\left(-q,0\right)$ and a vertical strip defined by the extreme values $S_{\min}$ and $S_{\max}$, thus leading to the following LMIs:

$$\begin{cases}\begin{pmatrix} -r\mathbf{X} & q\mathbf{X}+\mathbf{A}_j\mathbf{X}+\mathbf{B}\mathbf{\Gamma}_j & 0 & 0 \\ q\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\mathbf{\Gamma}_j^T\mathbf{B}^T & -r\mathbf{X} & 0 & 0 \\ 0 & 0 & S_{\min}\mathbf{X}-\frac{1}{2}\left(\mathbf{A}_j\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\mathbf{B}\mathbf{\Gamma}_j+\mathbf{M}_j^T\mathbf{\Gamma}_j^T\right) & 0 \\ 0 & 0 & 0 & -S_{\max}\mathbf{X}+\frac{1}{2}\left(\mathbf{A}_j\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\mathbf{B}\mathbf{\Gamma}_j+\mathbf{\Gamma}_j^T\mathbf{B}^T\right) \end{pmatrix}<0 \\ \mathbf{X}>0 \end{cases} \qquad (5.3.10)$$

The vertices state controller gains are then obtained as:

$$\mathbf{K}_j=\mathbf{\Gamma}_j\mathbf{X}^{-1} \qquad (5.3.11)$$

Because of computational issues, the design of the LPV controller is made by considering that only the $\tilde{N}$ most varying parameters vary, and all the others are fixed at their mean value, in the same way as the LPV observer was designed.

5.3.3 – Simulation Results

At first, both the controller based on the absolute model and the controller based on the Shamma model were tested in simulation. The responses of the overall system in both cases were compared to the ones that are obtained without the control loop, that is, by just applying the feedforward action calculated by means of the N_{ux} function.

The simulation was performed by considering the system to be perfectly described by its non-linear differential equations, whose parameters had been correctly estimated by the identification procedure. However, a saturation was added to reproduce the limits of the control action, that is $\left[-2.5V,2.5V\right]$, for both the tail motor voltage u_h and the main motor voltage u_v.

The experiment designed to compare the effectiveness of the control lasts 50 seconds. The reference angle trajectory is stepwise and a change in the set-point occurs at 25 seconds:

$$\begin{cases}\theta_h^{ref}=0.5, \quad \theta_v^{ref}=0.3 \qquad if \quad 0s\le t<25s \\ \theta_h^{ref}=0.8, \quad \theta_v^{ref}=0.5 \qquad if \quad 25s\le t<50s\end{cases} \qquad (5.3.12)$$

In ***Fig. 29*** and in ***Fig. 30***, the yaw and the pitch angle responses are shown, both when no control acts on the system and when an LPV controller acts on it. Both figures were obtained using the controllers designed by solving (5.3.10) and placing the poles of the closed-loop system in the region of the complex plane that is the intersection between the unit circle and the vertical strip with values $S_{\min}=0$ and $S_{\max}=1$ (a semicircle). The time constants that appear in (5.3.1) and (5.3.2) were chosen as $\tau_h=1s$ and $\tau_v=1s$.

In both figures it can be seen that the control system is able to make the system faster and to stabilize it at the desired yaw and pitch angles.

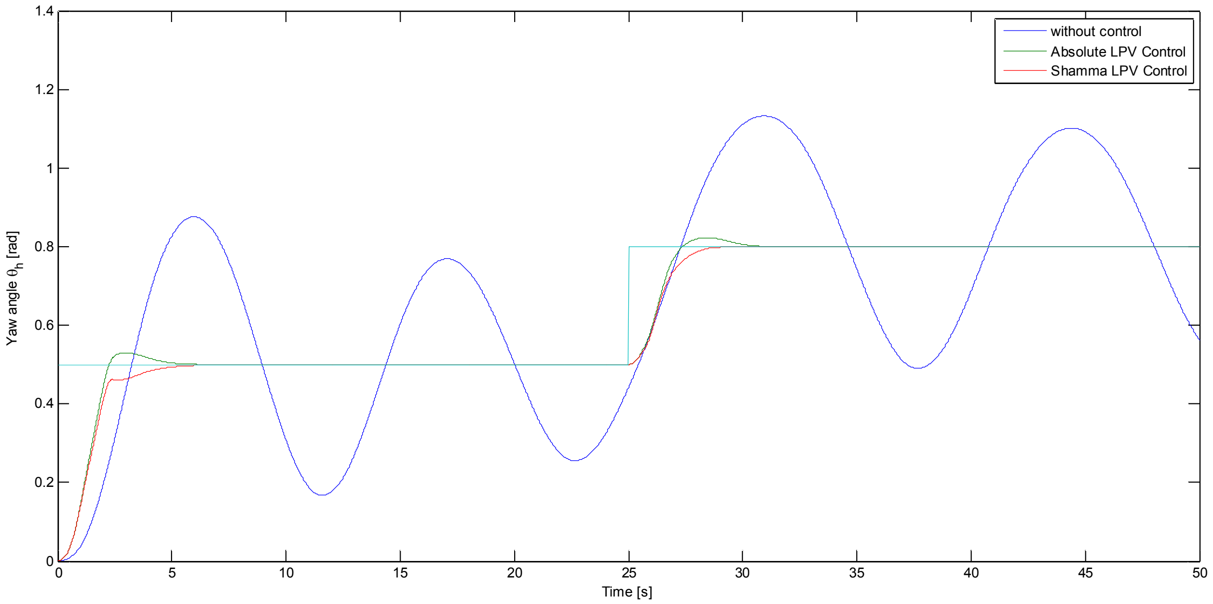

Fig. 29 - Comparison between the response of the yaw angle without control, with absolute LPV control and with Shamma LPV control (simulation data)

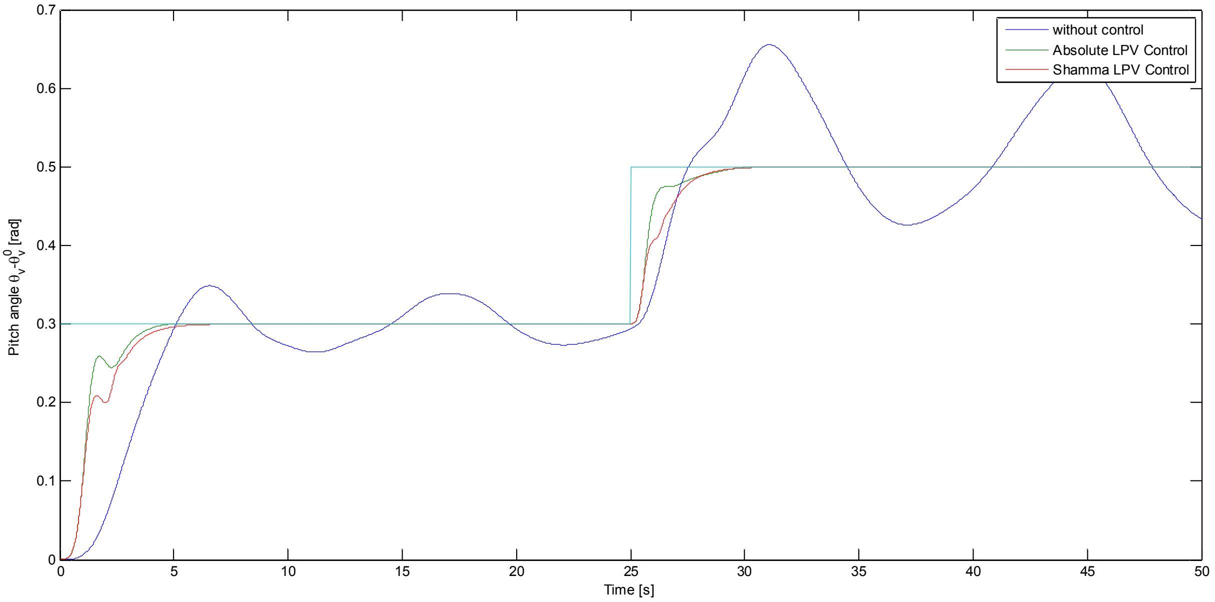

Fig. 30 – Comparison between the response of the pitch angle without control, with absolute LPV control and with Shamma LPV control (simulation data)

Afterwards, the region of the complex plane where to place the poles of the closed-loop system was varied, and some simulations were made to compare the effectiveness of the control strategy. During this step, it has been noticed that it is not possible to put the poles in whatever region. For example, if the desired region is the intersection between the circle with center $(0.8,0)$ and radius $r=0.1$ and the vertical strip with extreme values $S_{\min}=0.8$ and $S_{\max}=0.9$, the LMI constraints are not feasible, and the poles are not correctly placed in the desired region, as it can be seen in ***Fig. 31***. This may suggest that, depending on the desired region, the LPV models could lose the property of *D*-stabilizability.

In particular, it has been noticed that some of the closed-loop poles tend to remain near the point $(1,0)$ of the complex plane. Hence, to avoid the loss of *D*-stabilizability, it is suggested to choose desired regions that contain this point.

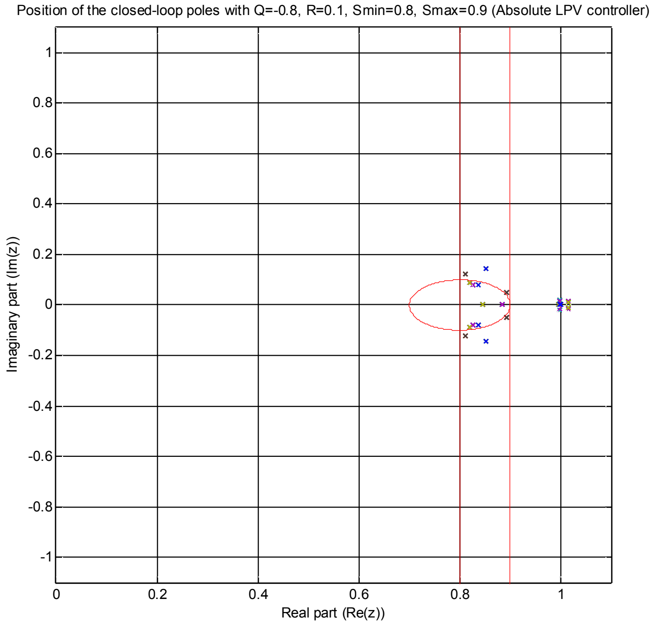

Fig. 31 - Position of the closed-loop poles with Q=-0.8, R=0.1, Smin=0.8, Smax=0.9 (Absolute LPV controller)

Varying the region of the complex plane where to put the poles, the performance of the closed-loop system changes, as it can be seen in ***Fig. 32*** and ***Fig. 33***. However, there is not a big improvement and both the absolute and the Shamma model give similar performance when used to design the control matrices $\mathbf{K}_j$.

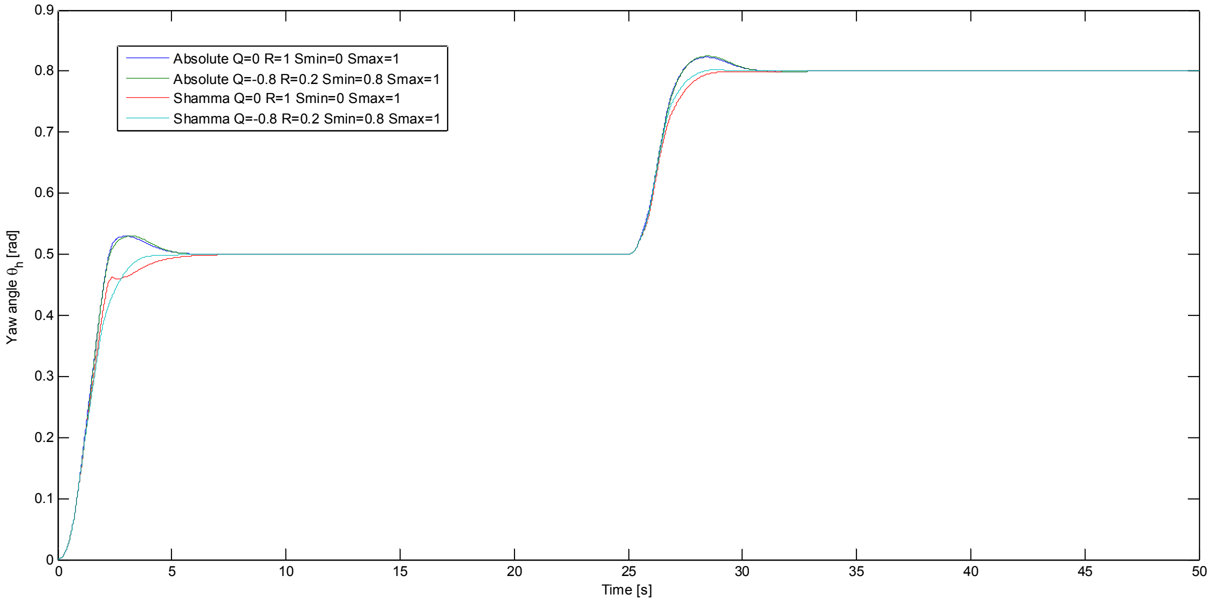

Fig. 32 - Comparison between the responses of the yaw angle of the controlled system when the region where to place the closed-loop poles varies (simulation data)

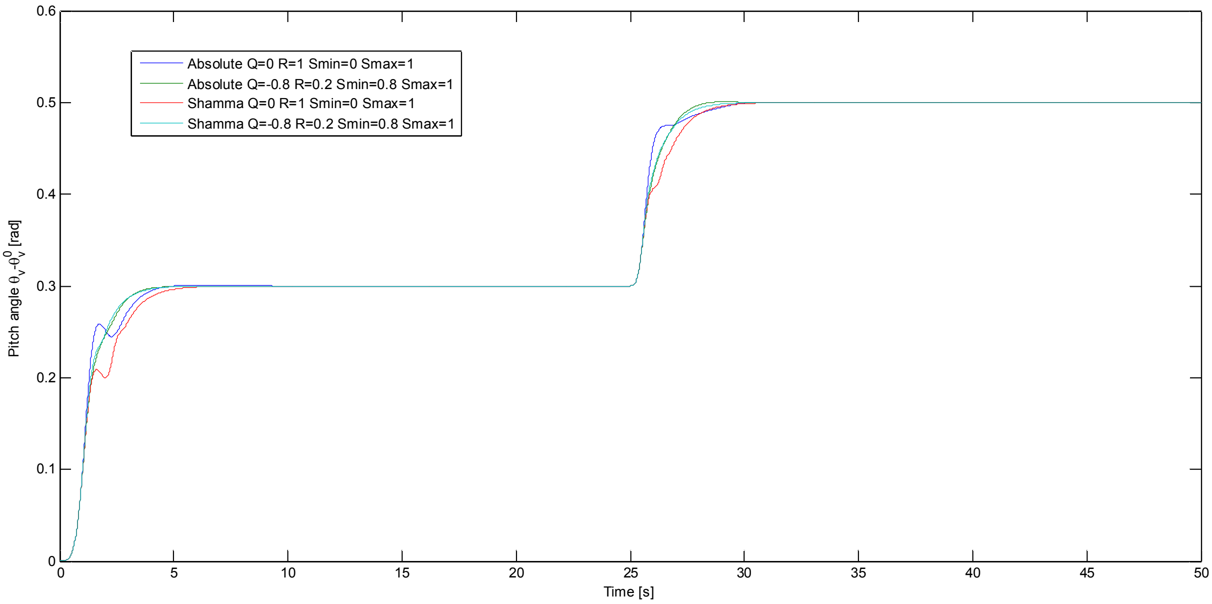

Fig. 33 - Comparison between the responses of the pitch angle of the controlled system when the region where to place the poles varies (simulation data)

Taking a look into the values taken by the input voltages u_h (***Fig. 34***) and u_v (***Fig. 35***), it can be seen that the motors work at their maximum voltages when a change in the reference angle occurs. This puts a limit in the performances of the control system, that cannot be improved because of the saturation.

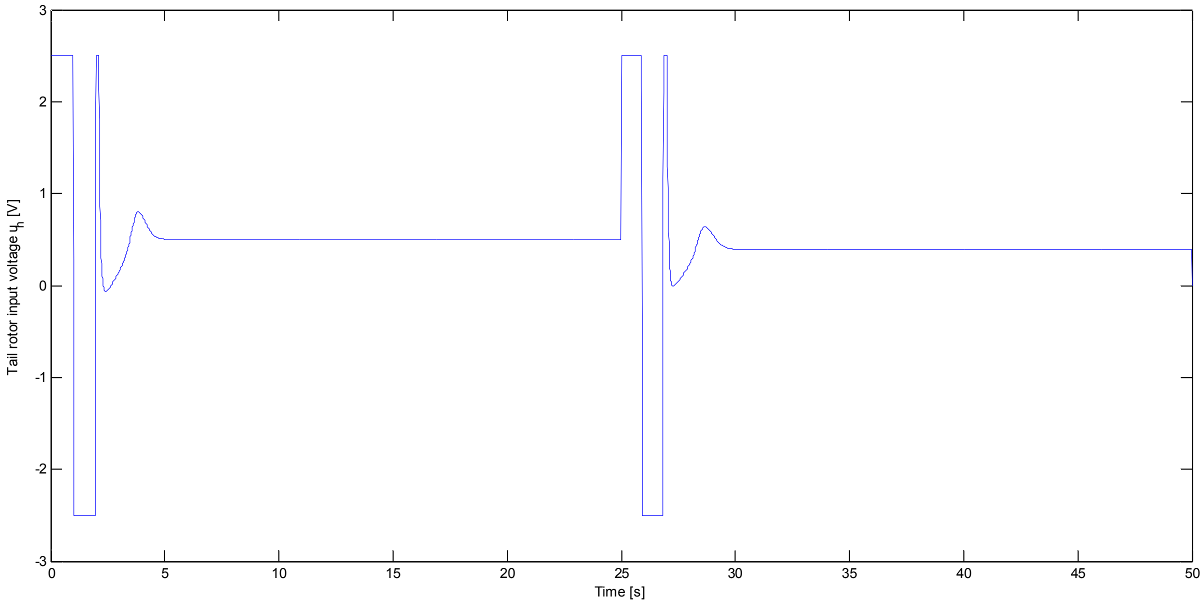

Fig. 34 - Tail motor input voltage u_h (simulation with Shamma controller with $Q = -0.8$, $R = 0.2$, $S_{\min} = 0.8$, $S_{\max} = 1$)

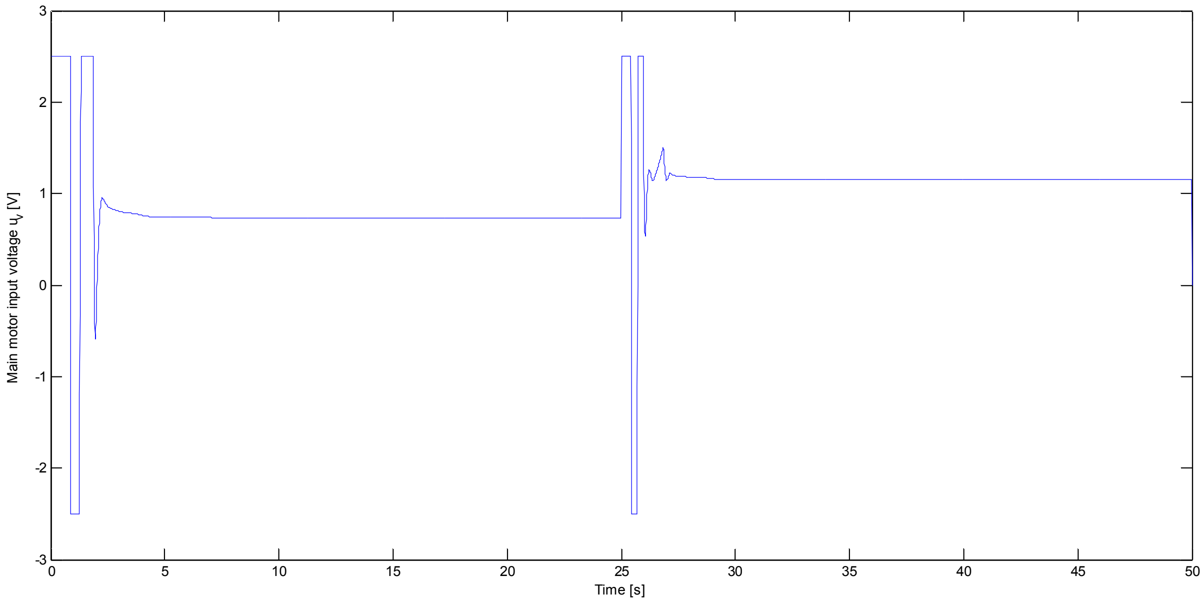

Fig. 35 - Main motor input voltage u_v (simulation with Shamma controller with $Q = -0.8$, $R = 0.2$, $S_{\min} = 0.8$, $S_{\max} = 1$)

5.3.4 – Practical results

Afterwards, the controller has been tested when applied to the real TRMS. The experiment to compare the effectiveness of the control was the same as in simulation experiments.

The obtained results are shown in ***Fig. 36***, ***Fig. 37***, ***Fig. 38*** and ***Fig. 39***. It can be seen that all the controllers make the system stable, even though those obtained using the Absolute LPV model for the TRMS show better performances during the transient with respect to those obtained using the Shamma LPV model. Hence, starting from this point, the Shamma LPV model will not be used anymore.

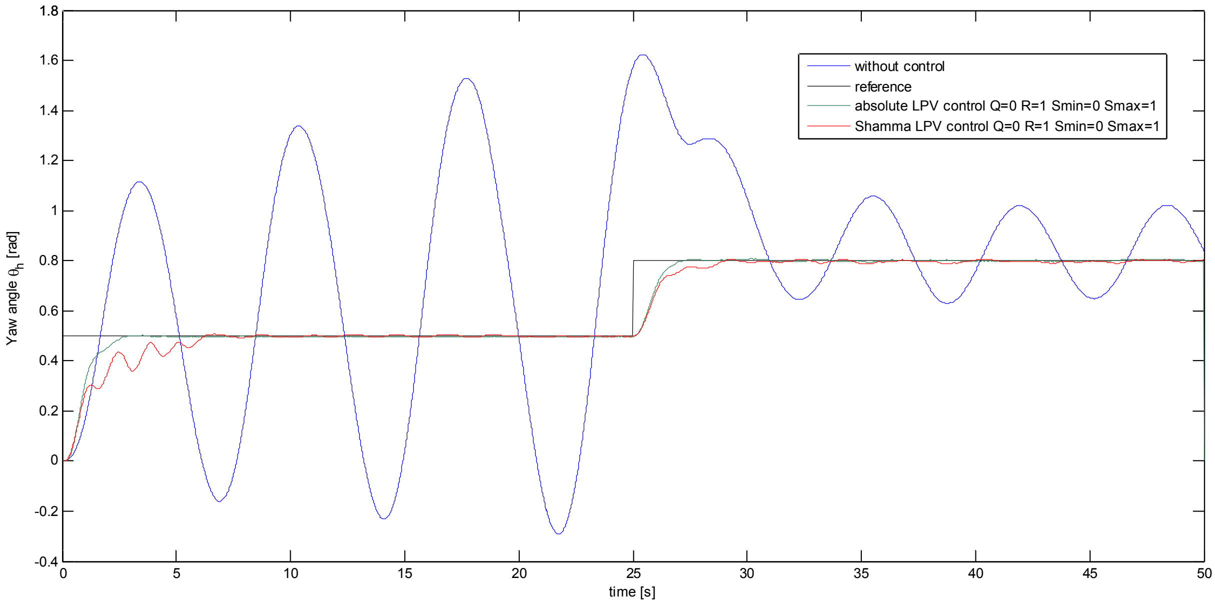

Fig. 36 - Comparison between the response of the yaw angle without control, with absolute LPV control and with Shamma LPV control (real experiment)

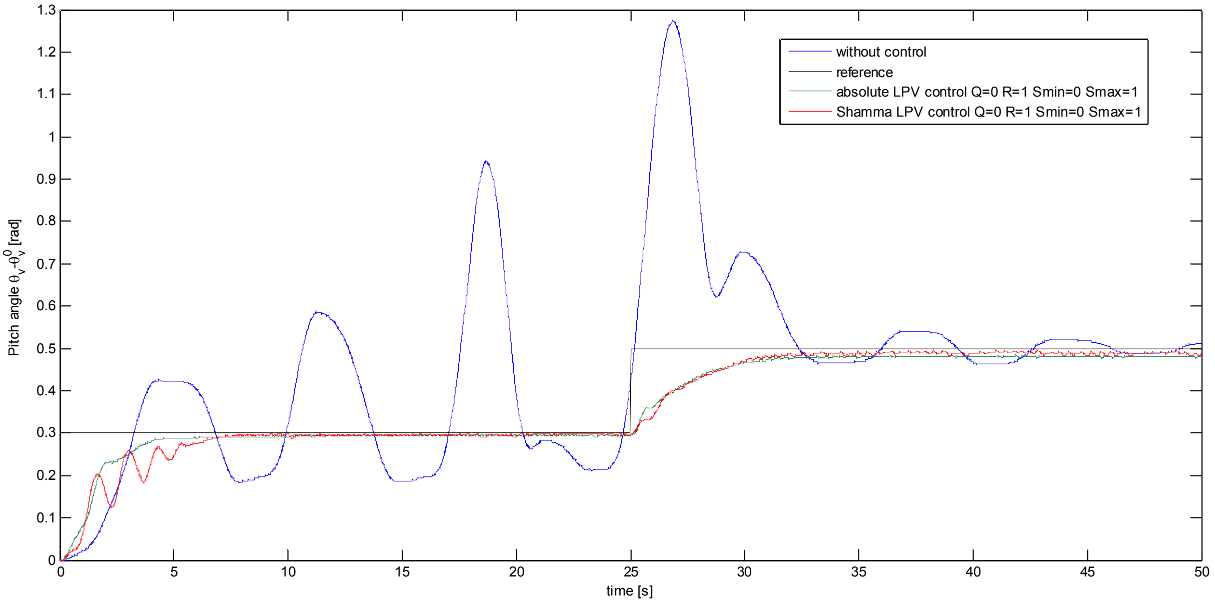

Fig. 37 - Comparison between the response of the pitch angle without control, with absolute LPV control and with Shamma LPV control (real experiment)

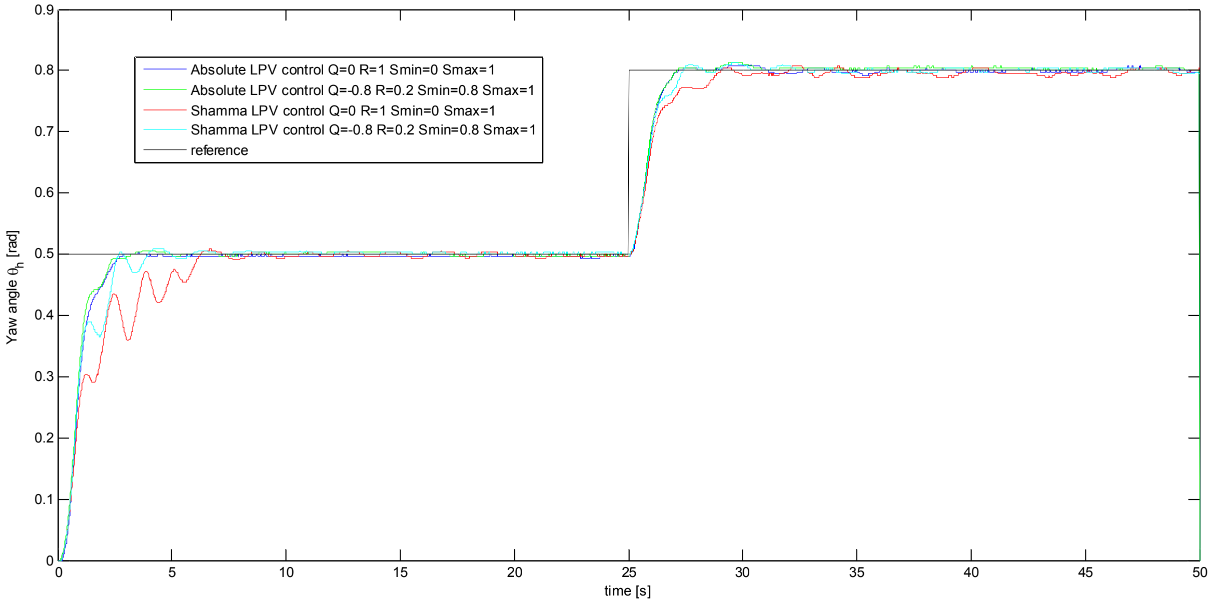

Fig. 38 - Comparison between the responses of the yaw angle of the controlled system when the region where to place the closed-loop poles varies (real experiment)

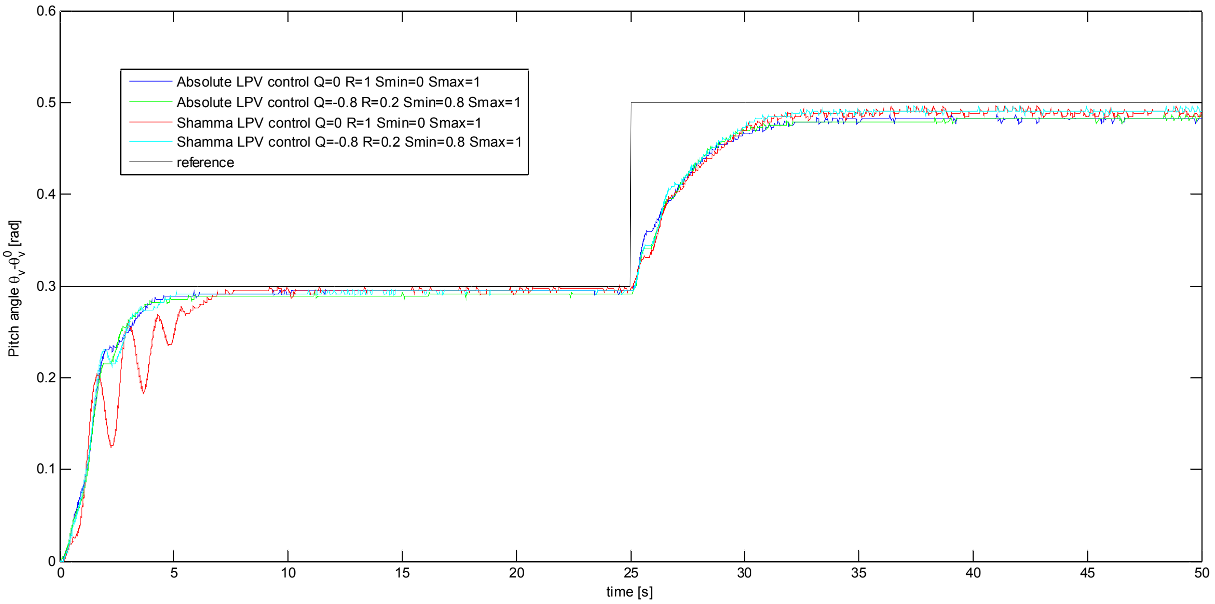

Fig. 39 - Comparison between the responses of the pitch angle of the controlled system when the region where to place the closed-loop poles varies (real experiment)

5.4 – 4-States-based LPV Controller with Local Control Loops

Another strategy to control the TRMS is to split it into subsystems, as it has been done for identification purposes. This allows controlling each subsystem for its own, resulting in the scheme of ***Fig. 40***.

At first, a control loop is designed for the tail rotor, in order to control its angular speed to a desired value. In this way, the controlled tail rotor dynamics becomes faster than when there is no control loop. Then, the same is made for the main rotor.

Assuming that the dynamical behaviour of the rotors is fast enough to be negligible, the non-linear differential equations that describe the aeromechanical part of the TRMS can be rewritten in a state-space form by considering the angular speeds of the two rotors as inputs of the system. Then, a control loop can be designed in an LPV way.

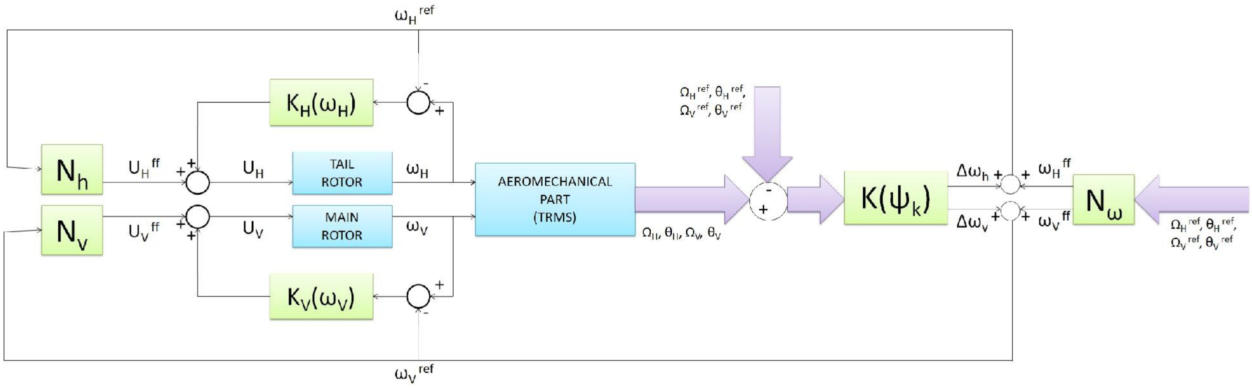

Fig. 40 - 4-State LPV Controller with local control loops

5.4.1 – Tail rotor local control loop

The equation that describes the dynamical behaviour of the tail rotor (3.2.43)

$$\frac{d\omega_h}{dt} = \frac{k_a k_1}{J_{tr} R_a} u_h - \left(\frac{B_{tr}}{J_{tr}} + \frac{k_a^2}{J_{tr} R_a} \right) \omega_h - \frac{f_1(\omega_h)}{J_{tr}} \tag{5.4.1}$$

can be rewritten in the LPV absolute form:

$$\frac{d\omega_h}{dt} = a_h^{abs}(\omega_h)\omega_h + b_h^{abs}u_h \tag{5.4.2}$$

where:

$$a_h^{abs}(\omega_h) = -\frac{k_a^2/R_a + B_{tr} + f_1(\omega_h)/\omega_h}{J_{tr}}$$

$$b_h^{abs} = \frac{k_a k_1}{J_{tr} R_a}$$

A polytopic discrete-time LPV model can be generated in an easy way using Euler approximation with a sampling time T_h. Then, applying the bounding box approach to the resulting a_h^{absd}:

$$\underline{a}_h^{absd} = 1 - \frac{T_h}{J_{tr}}\left(\frac{k_a^2}{R_a} + B_{tr} + \max\left\{k_{thp}\overline{\omega}_h, -k_{thn}\underline{\omega}_h\right\}\right)$$

$$\overline{a}_h^{absd} = 1 - \frac{T_h}{J_{tr}}\left(\frac{k_a^2}{R_a} + B_{tr}\right)$$

The tail rotor can be controlled through the following control law:

$$u_h(k) = u_h^{ff}(\omega_h^{ref}(k)) + K_h(\omega_h(k))\left[\omega_h(k) - \omega_h^{ref}(k)\right] \tag{5.4.3}$$

where K_h is a gain designed in an LPV way and scheduled by the current tail rotor angular speed $\omega_h(k)$, obtained directly from the corresponding sensor or, as an alternative, from the observer $\hat{\omega}_h(k)$.

$\omega_h^{ref}(k)$ is the reference for the angular speed of the tail rotor. It is one of the outputs of the controller that drives the aeromechanical part of the TRMS towards the desired θ_h^{ref} and θ_v^{ref}.

$u_h^{ff}(\omega_h^{ref}(k))$ is the feedforward input voltage, calculated from (5.4.1) as:

$$u_h^{ff}(k) = \frac{\left(B_{tr}R_a + k_a^2\right)\omega_h^{ref}(k) - f_1(\omega_h^{ref}(k))R_a}{k_a k_1} \tag{5.4.4}$$

The *LPV state controller* is designed with LMIs (2.3.13) assuming that the eigenvalue of the closed-loop system is placed in an LMI region that is the intersection between the disk of radius r_h and center $(-q_h, 0)$ and a vertical strip defined by the extreme values $S_{\min h}$ and $S_{\max h}$. In this case, a single eigenvalue has to be placed. Then, the radius r_h and the extreme values $S_{\min h}$ and $S_{\max h}$ can be chosen small enough to reduce the desired region to a point of the real axis, without compromising the feasibility of the LMIs.

The eigenvalue has been moved throughout the segment of the real axis [0,1] until a good performance of the closed-loop system was found, that is, the steady-state error in case of a step input was close to zero. Moving the eigenvalues towards 0 has the effect of increasing the proportional constant K_h of the controller, with the resulting effect of reducing the steady-state error.

After some trials, the controller has shown satisfactory performances when the LMI region was chosen as the disk with center 0.9 and radius 0.001, that is: $q = -0.9$, $r = 0.001$, $S_{\min} = 0.899$ and $S_{\max} = 0.991$. The controller works with a sampling time $T_h = 0.01s$.

Fig. 41 shows the results obtained in simulation environment. The presence of the control loop, in this case, makes faster the rotor, as the 1% settling time decreases from 3s to 0.5s. Because of the presence of an operational limit in the value that u_h can take, it is not possible to decrease further the settling time.

Fig. 42 shows the results obtained when the controller is applied to the real system. In this case, the effect of the control loop is not only to make the system faster, but also to compensate the errors in modelling, thus driving the rotor to the desired angular velocity with almost zero steady-state error.

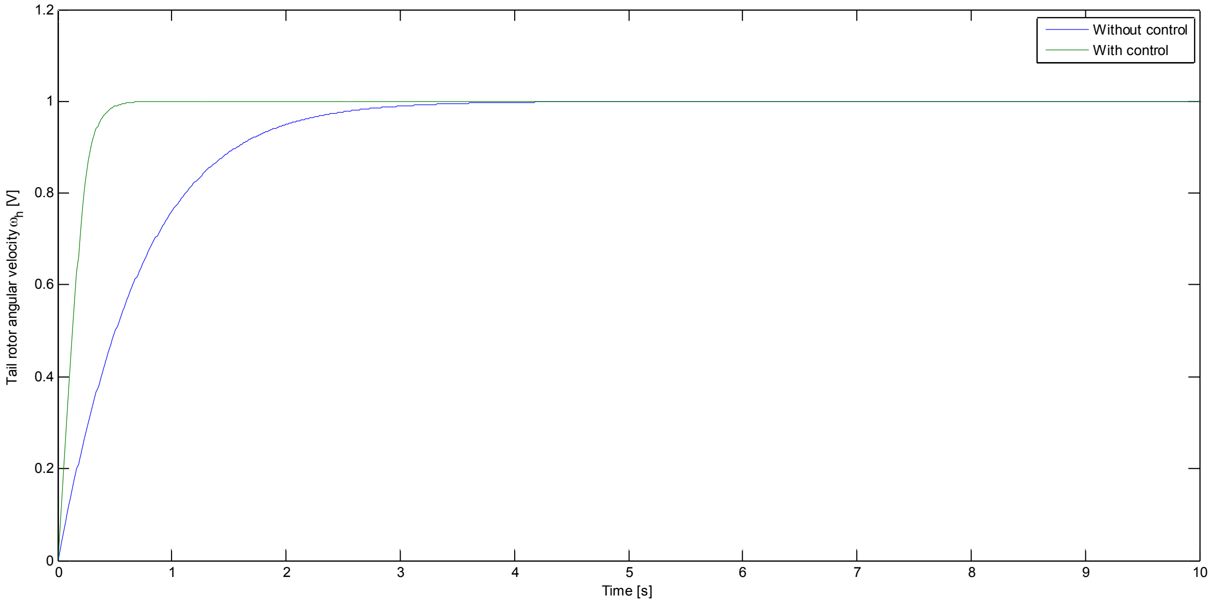

Fig. 41 - Simulation of the tail rotor angular velocity without control and with control

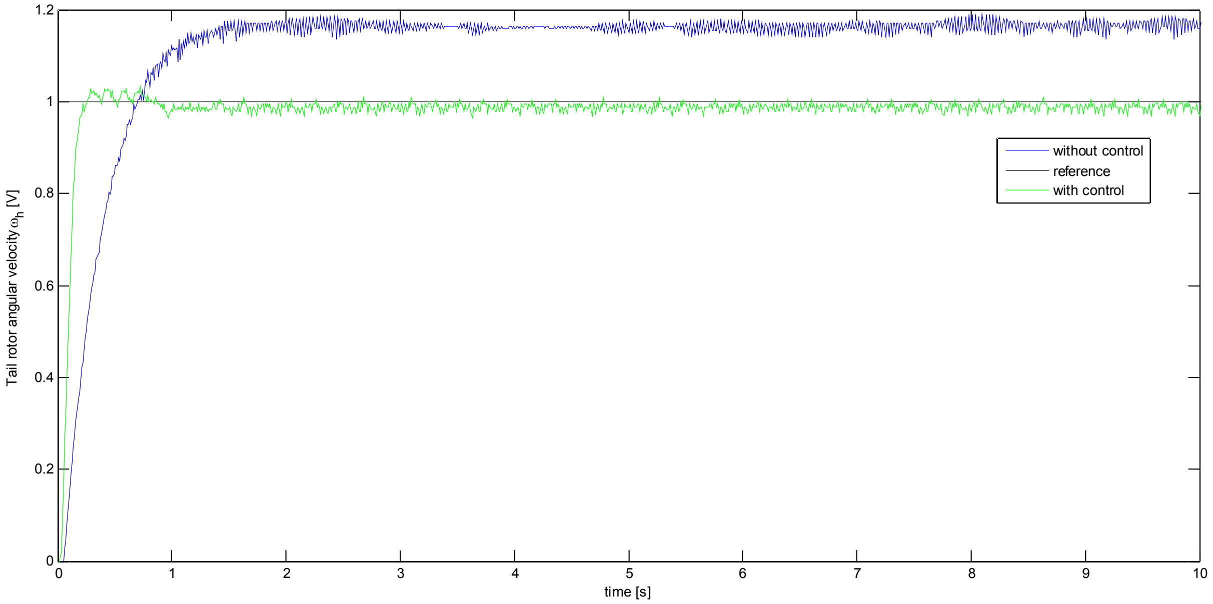

Fig. 42 – Time response of the tail rotor angular velocity without control and with control (real experiment)

5.4.2 – Main rotor local control loop

The equation that describes the dynamical behaviour of the main rotor (3.2.46):

$$\frac{d\omega_v}{dt}=\frac{k_a k_2}{J_{mr}R_a}u_v-\left(\frac{B_{mr}}{J_{mr}}+\frac{k_a^2}{J_{mr}R_a}\right)\omega_v-\frac{f_4(\omega_v)}{J_{mr}} \tag{5.4.5}$$

can be rewritten in the LPV absolute form:

$$\frac{d\omega_v}{dt}=a_v(\omega_v)\omega_v+b_v u_v \tag{5.4.6}$$

where:

$$a_v^{abs}(\omega_v)=-\frac{k_a^2/R_a+B_{mr}+f_4(\omega_v)/\omega_v}{J_{mr}}$$

$$b_v^{abs}=\frac{k_a k_2}{J_{mr}R_a}$$

A polytopic discrete-time LPV model can be generated in an easy way using Euler approximation with a sampling time T_v. Then, applying the bounding box approach to the resulting a_v^{absd}:

$$\underline{a}_v^{absd}=1-\frac{T_v}{J_{mr}}\left(\frac{k_a^2}{R_a}+B_{mr}+\max\left\{k_{tvp}\overline{\omega}_v,-k_{tvn}\underline{\omega}_v\right\}\right)$$

$$\overline{a}_v^{absd}=1-\frac{T_v}{J_{mr}}\left(\frac{k_a^2}{R_a}+B_{mr}\right)$$

The main rotor can be controlled through the following control law:

$$u_v(k)=u_v^{ff}(\omega_v^{ref}(k))+K_v(\omega_v(k))\left[\omega_v(k)-\omega_v^{ref}(k)\right] \tag{5.4.7}$$

where K_v is a gain designed in an LPV way and scheduled by the main rotor angular speed $\omega_v(k)$, obtained directly from the corresponding sensor or, as an alternative, from the observer $\hat{\omega}_v(k)$.

$\omega_v^{ref}(k)$ is the desired angular speed of the main rotor. It is one of the outputs of the controller that drives the aeromechanical part of the TRMS towards the desired θ_h^{ref} and θ_v^{ref}.

$u_v^{ff}(\omega_v^{ref}(k))$ is the feedforward input voltage, calculated from (5.4.5) as:

$$u_v^{ff}(k)=\frac{\left(B_{mr}R_a+k_a^2\right)\omega_v^{ref}(k)-f_4(\omega_v^{ref}(k))R_a}{k_a k_2} \tag{5.4.8}$$

The *LPV state controller* is designed with LMIs (2.3.13) assuming that the eigenvalue is placed in an LMI region that is the intersection between the disk of radius r_v and center $(-q_v,0)$ and a vertical strip defined by the extreme values $S_{\min v}$ and $S_{\max v}$. The radius r_v and the extreme values $S_{\min v}$ and $S_{\max v}$ can be chosen small enough to reduce the desired region to a point of the real axis, without compromising the feasibility of the LMIs.

The eigenvalue has been moved throughout the segment of the real axis [0,1] until a good performance of the closed-loop system was found, that is, the steady-state error in case of a step input was close to zero. Moving the eigenvalues towards 0 has the effect of increasing the proportional constant K_v of the controller, with the resulting effect of reducing the steady-state error.

After some trials, the controller has shown satisfactory performances when the LMI region was chosen as the disk with center 0.97 and radius 0.001, that is: $q = -0.97$, $r = 0.001$, $S_{\min} = 0.969$ and $S_{\max} = 0.971$. The controller works with a sampling time $T_v = 0.01s$.

Fig. 43 shows results obtained in simulation environment. The presence of the control loop, in this case, makes faster the rotor, as the 1% settling time decreases from 7.6s to 1.8s. Because of the presence of an operational limit in the value that u_v can take, it is not possible to decrease further the settling time.

Fig. 44 shows the results obtained when the controller is applied to the real system. In this case, the effect of the control loop is not only to make the system faster, but also to compensate the errors in modelling, thus driving the rotor to the desired angular velocity with almost zero steady-state error.

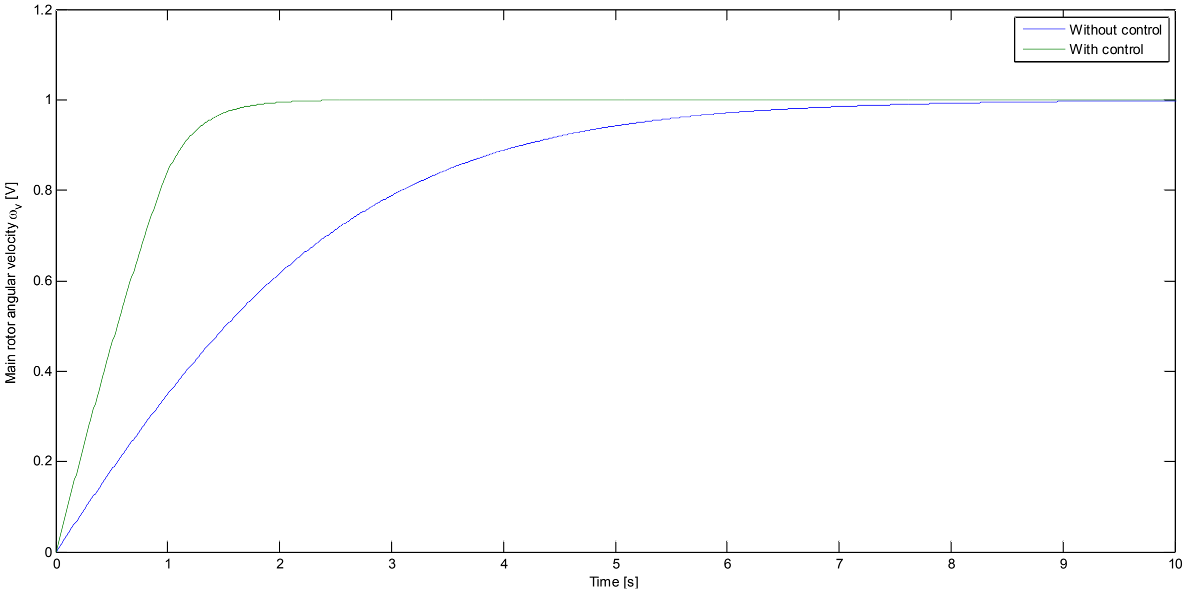

Fig. 43 - Simulation of the main motor angular velocity without and with control

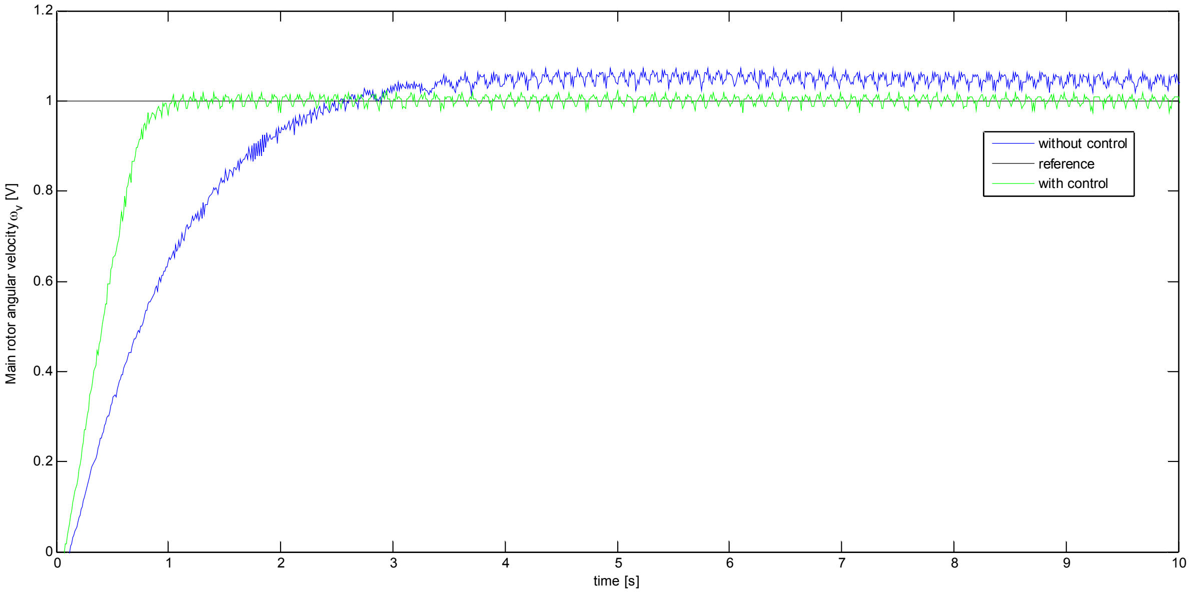

Fig. 44 - Time response of the main rotor angular velocity without control and with control (real experiment)

5.4.3 – Aeromechanical control loop: the reduced model

The equations that describe the dynamical behaviour of the aeromechanical part of the TRMS (3.2.44), (3.2.45), (3.2.47) and (3.2.48), can be rewritten by neglecting the dynamics of the rotors and by considering their angular velocities as inputs of the system. In order to make the system linear with respect to the inputs, $f_2\left(\omega_h\right)$ and $f_5\left(\omega_v\right)$ are used as inputs instead of ω_h and ω_v. The resulting reduced order model can be rewritten in the following LPV absolute form:

$$\begin{bmatrix}\dot{\Omega}_h(t)\\ \dot{\theta}_h(t)\\ \dot{\Omega}_v(t)\\ \dot{\theta}_v(t)\end{bmatrix}=\begin{pmatrix}a_{11}(\psi(t)) & a_{12}(\psi(t)) & a_{13}(\psi(t)) & a_{14}(\psi(t))\\ 1 & 0 & 0 & 0\\ a_{31}(\psi(t)) & 0 & a_{33} & a_{34}(\psi(t))\\ 0 & 0 & 1 & 0\end{pmatrix}\begin{bmatrix}\Omega_h(t)\\ \theta_h(t)\\ \Omega_v(t)\\ \theta_v(t)\end{bmatrix}+\begin{pmatrix}b_{11}(\psi(t)) & 0\\ 0 & 0\\ 0 & b_{32}(\psi(t))\\ 0 & 0\end{pmatrix}\begin{bmatrix}f_2\left(\omega_h(t)\right)\\ f_5\left(\omega_v(t)\right)\end{bmatrix} \tag{5.4.9}$$

where:

$$a_{11}(\psi(t))=-\frac{k_{oh}}{D\cos^2\theta_v+E\sin^2\theta_v+F} \tag{5.4.10}$$

$$a_{12}(\psi(t))=-\frac{f_3\left(\theta_h\right)/\theta_h}{D\cos^2\theta_v+E\sin^2\theta_v+F} \tag{5.4.11}$$

$$a_{13}(\psi(t))=\frac{k_m\omega_v\sin\theta_v\left(D\cos^2\theta_v-E\sin^2\theta_v-F-2E\cos^2\theta_v\right)}{\left(D\cos^2\theta_v+E\sin^2\theta_v+F\right)^2} \tag{5.4.12}$$

$$a_{14}(\psi(t))=\frac{f_6\left(\theta_v\right)/\left(\theta_v-\theta_v^0\right)}{D\cos^2\theta_v+E\sin^2\theta_v+F} \tag{5.4.13}$$

$$a_{31}(\psi(t))=-\Omega_h H\sin\theta_v\cos\theta_v/J_v \tag{5.4.14}$$

$$a_{33}=-k_{ov}/J_v \tag{5.4.15}$$

$$a_{34}(\psi(t))=\frac{g\left(\left(A-B\right)\cos\theta_v-C\sin\theta_v\right)/\left(\theta_v-\theta_v^0\right)}{J_v} \tag{5.4.16}$$

$$b_{11}(\psi(t))=\frac{l_t\cos\theta_v}{D\cos^2\theta_v+E\sin^2\theta_v+F} \tag{5.4.17}$$

$$b_{32}(\psi(t))=\frac{l_m+k_g\Omega_h\cos\theta_v}{J_v} \tag{5.4.18}$$

A polytopic discrete-time LPV model can be generated in an easy way using Euler approximation with a sampling time T_s, then applying the bounding box approach. Then, the parameter-varying elements of the matrix $\mathbf{B}$ can be approximated with their mean value, so as to obtain an LPV model with a constant input matrix.

5.4.4 – Aeromechanical control loop: the control signal

The controller has two outputs: $\omega_h^{ref}(k)$, reference for the tail rotor angular speed, and $\omega_v^{ref}(k)$, reference for the main rotor angular speed. Both the outputs are inputs to the local control loops that control the tail and the main rotor, and are obtained from the corresponding $f_2^{ref}(k)$ and $f_5^{ref}(k)$, taking into account equations (3.2.20) and (3.2.34):

$$\omega_h^{ref}(k) = \begin{cases} \sqrt{\dfrac{f_2^{ref}(k)}{k_{fhp}}} & if\, f_2^{ref}(k) \geq 0 \\ -\sqrt{-\dfrac{f_2^{ref}(k)}{k_{fhn}}} & if\, f_2^{ref}(k) < 0 \end{cases} \tag{5.4.19}$$

$$\omega_v^{ref}(k) = \begin{cases} \sqrt{\dfrac{f_5^{ref}(k)}{k_{fvp}}} & if\, f_5^{ref}(k) \geq 0 \\ -\sqrt{-\dfrac{f_5^{ref}(k)}{k_{fvn}}} & if\, f_5^{ref}(k) < 0 \end{cases} \tag{5.4.20}$$

Both $f_2^{ref}(k)$ and $f_5^{ref}(k)$ are the sum of two parts:

$$\begin{bmatrix} f_2^{ref}(k) \\ f_5^{ref}(k) \end{bmatrix} = \begin{bmatrix} f_2^{ff}\left(\Omega_h^{ref}(k), \theta_h^{ref}(k), \Omega_v^{ref}(k), \theta_v^{ref}(k)\right) \\ f_5^{ff}\left(\Omega_h^{ref}(k), \theta_h^{ref}(k), \Omega_v^{ref}(k), \theta_v^{ref}(k)\right) \end{bmatrix} + \mathbf{K}\left(\boldsymbol{\psi}(k)\right) \begin{bmatrix} \Omega_h(k) - \Omega_h^{ref}(k) \\ \theta_h(k) - \theta_h^{ref}(k) \\ \Omega_v(k) - \Omega_v^{ref}(k) \\ \theta_v(k) - \theta_v^{ref}(k) \end{bmatrix} \tag{5.4.21}$$

where:

$$f_2^{ff}(k) = \frac{k_{oh}\Omega_h^{ref}(k) + f_3(\theta_h^{ref}(k)) - f_6(\theta_v^{ref}(k)) - \frac{k_m \omega_v^{ff}(k) \sin\theta_v^{ref}(k) \Omega_v^{ref}(k)\left(D\cos^2\theta_v^{ref}(k) - E\sin^2\theta_v^{ref}(k) - F - 2E\cos^2\theta_v^{ref}(k)\right)}{D\cos^2\theta_v^{ref}(k) + E\sin^2\theta_v^{ref}(k) + F}}{l_m} \tag{5.4.22}$$

$$f_5^{ff}(k) = \frac{k_{ov}\Omega_v^{ref}(k) + \left(\Omega_h^{ref}(k)\right)^2 H\sin\theta_v^{ref}(k)\cos\theta_v^{ref}(k) - g\left(\left(A - B\right)\cos\theta_v^{ref}(k) - C\sin\theta_v^{ref}(k)\right)}{l_m + k_g\Omega_h^{ref}(k)\cos\theta_v^{ref}(k)} \tag{5.4.23}$$

$$\omega_v^{ff}(k) = \begin{cases} \sqrt{\dfrac{f_5^{ff}(k)}{k_{fvp}}} & if\, f_5^{ff}(k) \geq 0 \\ -\sqrt{-\dfrac{f_5^{ff}(k)}{k_{fvn}}} & if\, f_5^{ff}(k) < 0 \end{cases} \tag{5.4.24}$$

$\mathbf{K}(\boldsymbol{\psi}(k))$ is the LPV controller based on the 4-states LPV model (5.4.9). It is designed in the same way as the controllers that have been described in the previous paragraphs, by solving LMIs (2.3.13). In this case the scheduling vector is:

$$\boldsymbol{\psi}(k) = \left[a_{11}^{absd}(\mathbf{x}(k)), a_{12}^{absd}(\mathbf{x}(k)), a_{13}^{absd}(\mathbf{x}(k)), a_{14}^{absd}(\mathbf{x}(k)), a_{31}^{absd}(\mathbf{x}(k)), a_{34}^{absd}(\mathbf{x}(k))\right]$$

where each element of the scheduling vector is given by (5.4.10)-(5.4.16).

$\theta_h^{ref}(k)$ and $\theta_v^{ref}(k)$ are determined offline; $\Omega_h^{ref}(k)$ and $\Omega_v^{ref}(k)$ are calculated as:

$$\Omega_h^{ref}(k) = \left(\theta_h^{ref}(k) - \theta_h(k)\right) / \tau_h \tag{5.4.25}$$

$$\Omega_v^{ref}(k) = \left(\theta_v^{ref}(k) - \theta_v(k)\right) / \tau_v \tag{5.4.26}$$

where $\theta_h(k)$ and $\theta_v(k)$ are the actual yaw and pitch angles. The desired horizontal and vertical velocities are chosen to be the velocities needed to drive the system from the actual angles $\theta_h(k)$ and $\theta_v(k)$ to the desired ones $\theta_h^{ref}(k)$ and $\theta_v^{ref}(k)$ in the times τ_h and τ_v, respectively.

5.4.5 – Aeromechanical control loop: results

At first, the controller of the aeromechanical part of the TRMS has been tested in simulation. The sampling time has been chosen as $T_s = 0.01$ and the controller has been designed both with $q = 0$, $r = 1$, $S_{\min} = 0$, $S_{\max} = 1$, and with $q = -0.8$, $r = 0.2$, $S_{\min} = 0.8$, $S_{\max} = 1$. The references for the horizontal and vertical angular speeds of the beam $\Omega_{h/v}^{ref}$ have been calculated through (5.4.25) and (5.4.26) by assuming that $\tau_h = 1s$ and $\tau_v = 1s$.

In order to compare the effectiveness of this control strategy with the one that was previously used based on the 6-states LPV model, the same reference angle trajectory is used:

$$\begin{cases} \theta_h^{ref} = 0.5, \theta_v^{ref} = 0.3 & if\, 0s \le t < 25s \\ \theta_h^{ref} = 0.8, \theta_v^{ref} = 0.5 & if\, 25s \le t < 50s \end{cases}$$

In ***Fig. 45*** and ***Fig. 46***, the response of the system with the 4-states LPV controller is compared with the one obtained using the Shamma 6-state LPV Controller designed with $q = -0.8$, $r = 0.2$, $S_{\min} = 0.8$ and $S_{\max} = 1$.

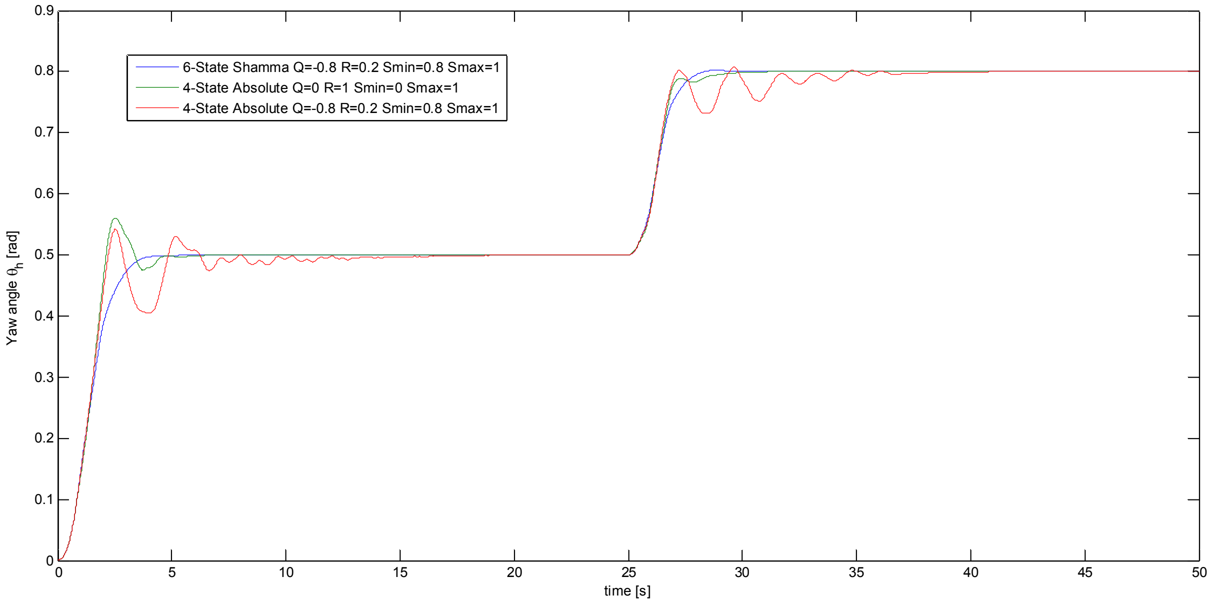

Fig. 45 - Comparison between the responses of the yaw angle of the system controlled by a 6-states LPV controller and the system controlled by a 4-states LPV controller (simulation data)

It can be seen that this control strategy is able to stabilize the system as well as the previous strategy, even though there is a loss of performances in the transient response.

Similar results are obtained when the controller is tested with the real TRMS; as shown in ***Fig. 47*** and ***Fig. 48***. In this case, the TRMS is driven towards the desired yaw and pitch angles, but there is the presence of some oscillations both in the transient and in steady-state. However, it could be possible that a change in the region of the complex plane where to place the closed-loop poles could lead to better results.

Reducing the order of the system from 6 to 4 has the consequence of reducing the computation time needed to solve the LMIs (2.3.13). This is an important issue when an integral action is added to the system, as such an addition increases the order of the system from 6 to 8 and an 8-states LPV controller would need too much computation time to solve the LMIs. A 4-states LPV controller with integral action is described in the next paragraph.

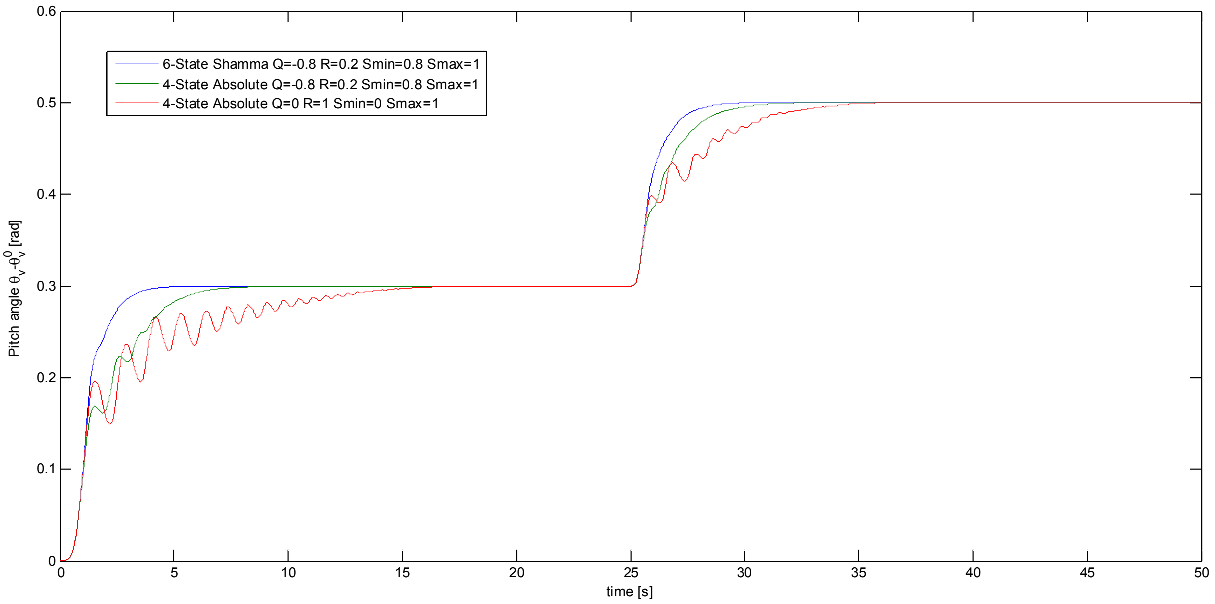

Fig. 46 - Comparison between the responses of the pitch angle of the TRMS controlled by a 6-states LPV controller and the TRMS controlled by a 4-states LPV controller (simulation data)

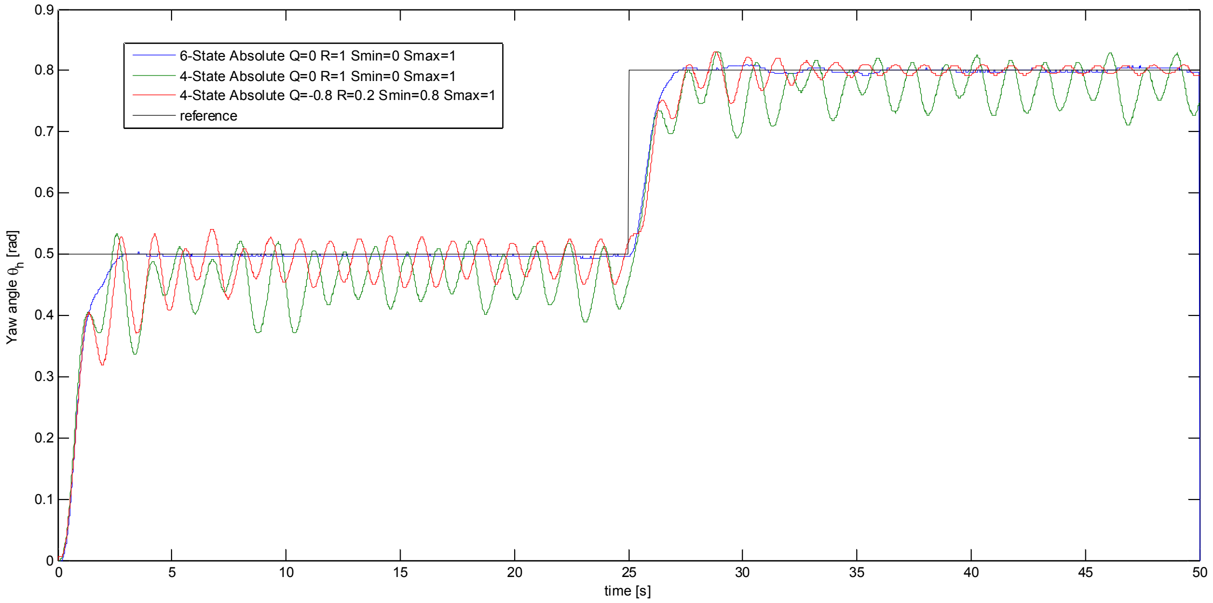

Fig. 47 - Comparison between the responses of the yaw angle of the TRMS controlled by a 6-states LPV controller and the TRMS controlled by a 4-states LPV controller (real data)

5.5 – 4-States LPV Controller with Integral Action and Local Control Loops

5.5.1 – LPV Integral Control by State Augmentation

Integral control is needed in order to eliminate the steady-state errors due to some imperfections in either the modelling or the identification of the TRMS. For this reason, some form of integral control is typically included in most control systems. The control that has been used so far is an extension of state-space design methods to LPV systems: state-space designs will not naturally include an integral action unless additional steps are taken. A method to add an integral action to classic state-space control is described in **[Franklin, 1997]** and it is based on state augmentation: the state is augmented in such a way that the control system achieves zero steady-state error for a general class of reference and disturbance signals.

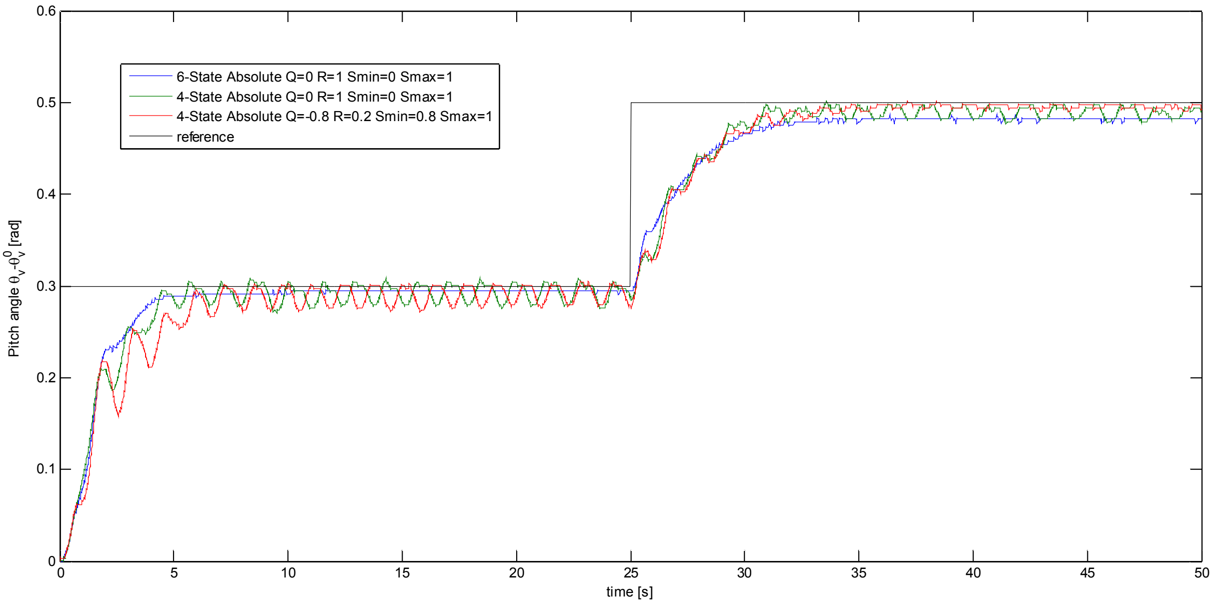

Fig. 48 - Comparison between the responses of the pitch angle of the TRMS controlled by a 6-states LPV controller and the TRMS controlled by a 4-states LPV controller (real data)

The idea is to add a state to obtain an integral of the error signal. This integrator is implemented as part of the controller equations. Then, the integral is fed-back along with the estimated or measured state. To accomplish the design of the feedback gains for both the integral and the original state vector, the model of the plant is augmented and an integral error output is added to the existing plant state output. This augmented model is then used as before to calculate the feedback control gains for the augmented model.

More specifically, the state of the system:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A}\left(\boldsymbol{\psi}(k)\right)\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) \end{cases} \tag{5.4.27}$$

is augmented with $\mathbf{x}_I$, the integral of the error, $\mathbf{e} = \mathbf{y} - \mathbf{r}$. The discrete integral is simply a summation of all past values of $\mathbf{e}(k)$, which results in the difference equation:

$$\mathbf{x}_I(k+1) = \mathbf{x}_I(k) + \mathbf{e}(k) = \mathbf{x}_I(k) + \mathbf{C}\mathbf{x}(k) - \mathbf{r}(k) \tag{5.4.28}$$

therefore arriving at the augmented plant model:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{x}_I(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}\left(\boldsymbol{\psi}(k)\right) & \mathbf{0} \\ \mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{x}_I(k) \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \mathbf{u}(k) - \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{r}(k) \tag{5.4.29}$$

With this revised definition of the system, the design technique already used can be applied directly for control law design.

The addition of extra poles for the integrator state element will typically lead to a deteriorated command input response compared to that obtained without integral control.

5.5.2 – Augmented model of the TRMS

The LPV model of TRMS can be modified by augmenting the state in order to add an integral action to the control. Using the Euler approximation with a sampling time T_s, the model (5.4.9) becomes:

$$\begin{bmatrix} \Omega_h(k+1) \\ \theta_h(k+1) \\ \Omega_v(k+1) \\ \theta_v(k+1) \\ I_h(k+1) \\ I_v(k+1) \end{bmatrix} = \begin{pmatrix} 1+a_{11}(\mathbf{x}(k))T_s & a_{12}(\mathbf{x}(k))T_s & a_{13}(\mathbf{x}(k))T_s & a_{14}(\mathbf{x}(k))T_s & 0 & 0 \\ T_s & 1 & 0 & 0 & 0 & 0 \\ a_{31}(\mathbf{x}(k))T_s & 0 & 1+a_{33}T_s & a_{34}(\mathbf{x}(k))T_s & 0 & 0 \\ 0 & 0 & T_s & 1 & 0 & 0 \\ 0 & T_s & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & T_s & 0 & 1 \end{pmatrix} \begin{bmatrix} \Omega_h(k) \\ \theta_h(k) \\ \Omega_v(k) \\ \theta_v(k) \\ I_h(k) \\ I_v(k) \end{bmatrix}$$

$$+ \begin{pmatrix} \tilde{b}_{11}T_s & 0 \\ 0 & 0 \\ 0 & \tilde{b}_{32}T_s \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \begin{bmatrix} f_2(\omega_h(k)) \\ f_5(\omega_v(k)) \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} \begin{bmatrix} \theta_h^{ref}(k) \\ \theta_v^{ref}(k) \end{bmatrix} \qquad (5.4.30)$$

A polytopic discrete-time LPV model can be generated applying the bounding box approach.

5.5.3 –Results

The controller has been designed with $q = -0.8$, $r = 0.2$, $S_{\min} = 0.8$, $S_{\max} = 1$, choosing a sampling time of $T_s = 0.01$. The reference for the horizontal and vertical angular speeds of the beam $\Omega_{h/v}^{ref}$ have been calculated through (5.4.25) and (5.4.26) by assuming that $\tau_h = 1s$ and $\tau_v = 1s$.

In order to compare the effectiveness of this control with the ones previously designed using the 6-states LPV model and the 4-states LPV model (without integral action), the same reference angle trajectory is used:

$$\begin{cases} \theta_h^{ref} = 0.5, \theta_v^{ref} = 0.3 & if\, 0s \le t < 25s \\ \theta_h^{ref} = 0.8, \theta_v^{ref} = 0.5 & if\, 25s \le t < 50s \end{cases}$$

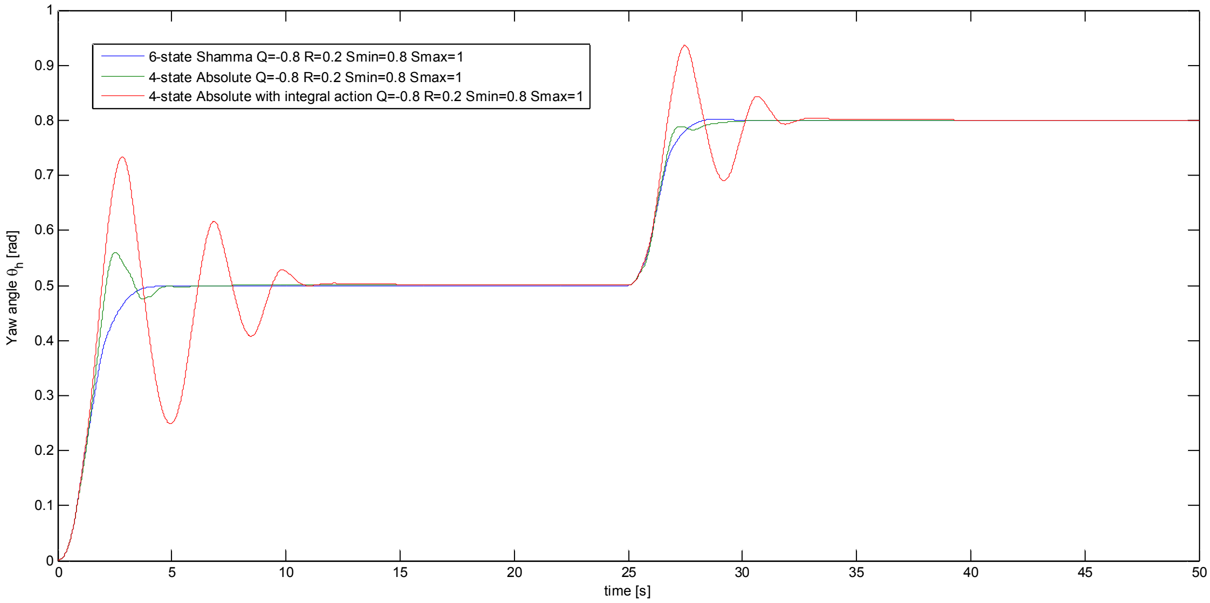

Fig. 49 - Comparison between different control strategies for the yaw angle θ_h (simulation data)

Fig. 49 and ***Fig. 50*** show results obtained in simulation, compared with those that had been obtained previously by applying other control strategies. It can be seen that the introduction of an integral action causes a loss of performance in the transient.

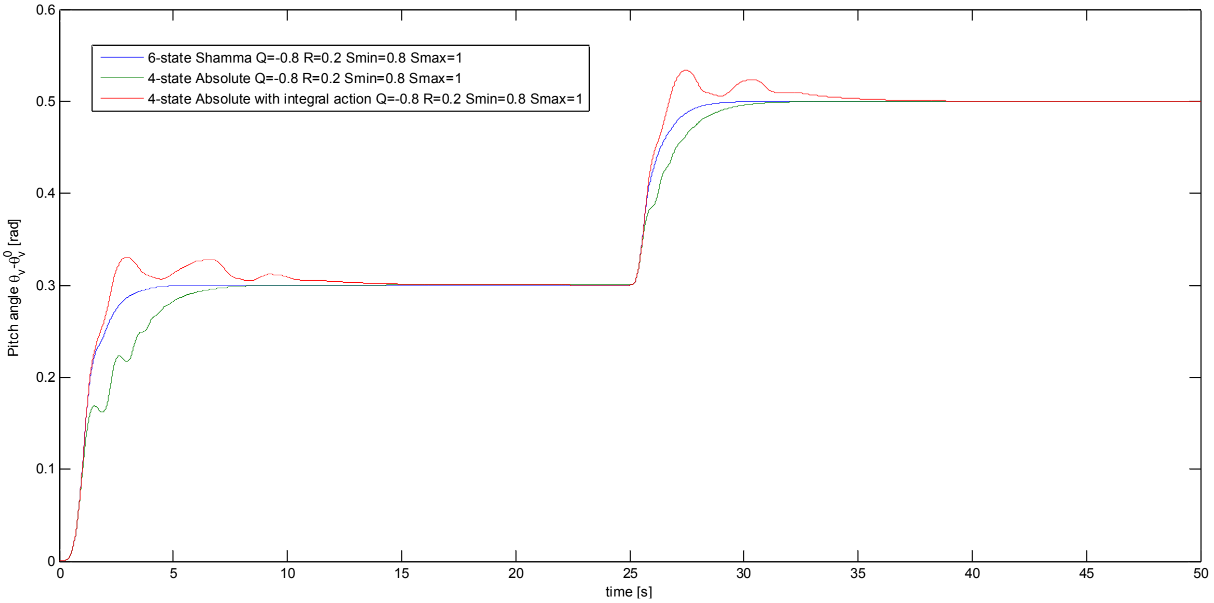

Fig. 50 - Comparison between different control strategies for the pitch angle θ_v (simulation data)

Despite the loss of performances appeared in simulation, the presence of an integral action eliminates the steady-state error when applied to the real system, as it can be seen in ***Fig. 51*** and ***Fig. 52***.

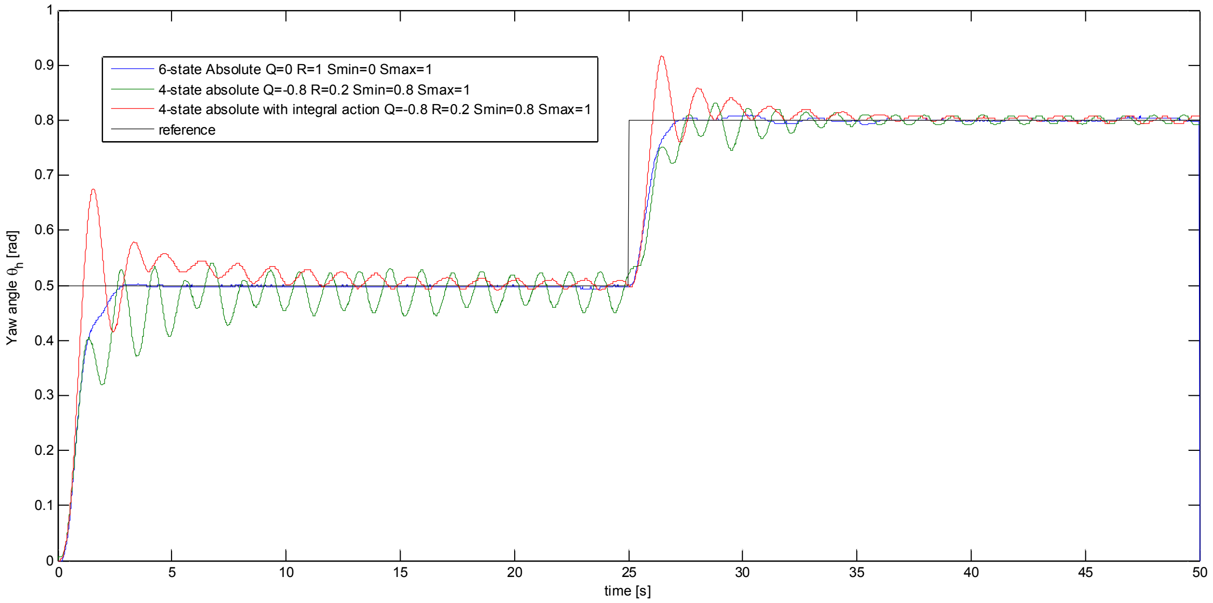

Fig. 51 - Comparison between different control strategies for the yaw angle θ_h (real data)

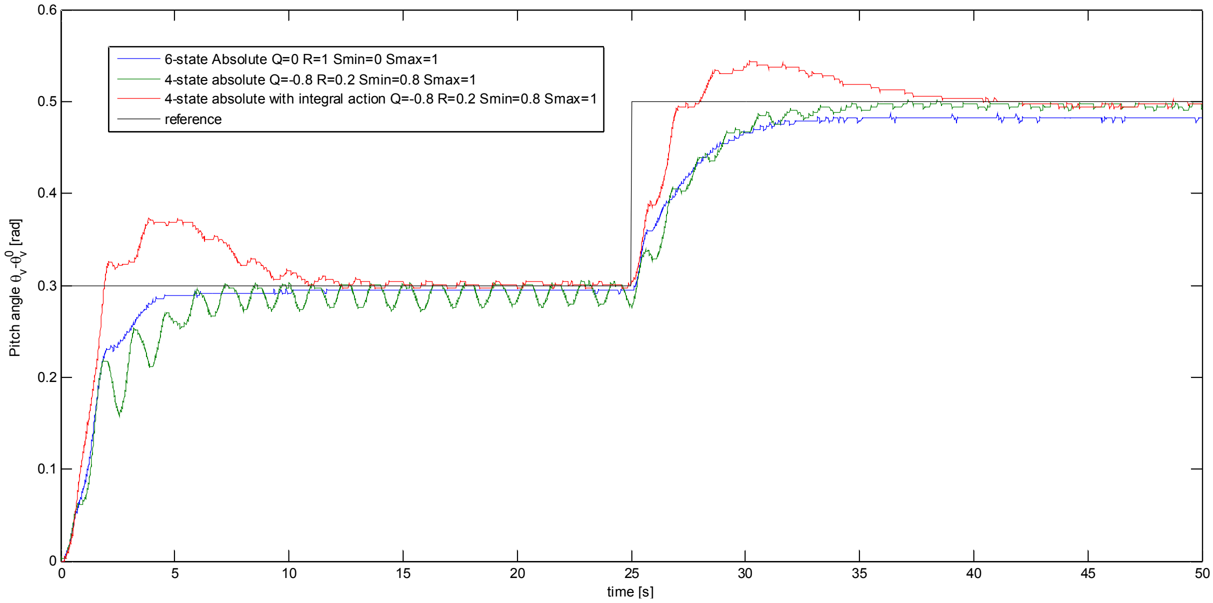

Fig. 52 - Comparison between different control strategies for the pitch angle θ_v (real data)

5.6 – Disturbance rejection

The effectiveness of both the controllers, the 6-states LPV controller and the 4-states one with local control loops and integral action, was tested when some disturbances were applied to the TRMS. The experiment was done on the real system, where the TRMS had to follow the angle references $\theta_h^{ref} = 0.6$ and $\theta_v^{ref} = 0.4$. Exterior disturbances were provoked manually and the control systems proved to be able to reject them. Again, the better transient performances were obtained with the 6-states controller (***Fig. 53*** and ***Fig. 54***), while the 4-states controller with integral action could eliminate the steady-state error (***Fig. 55*** and ***Fig. 56***). Hence, the 6-states LPV controller will be the one to be used as the nominal controller in Fault-Tolerant Control schemes, that are the subject of the next chapter.

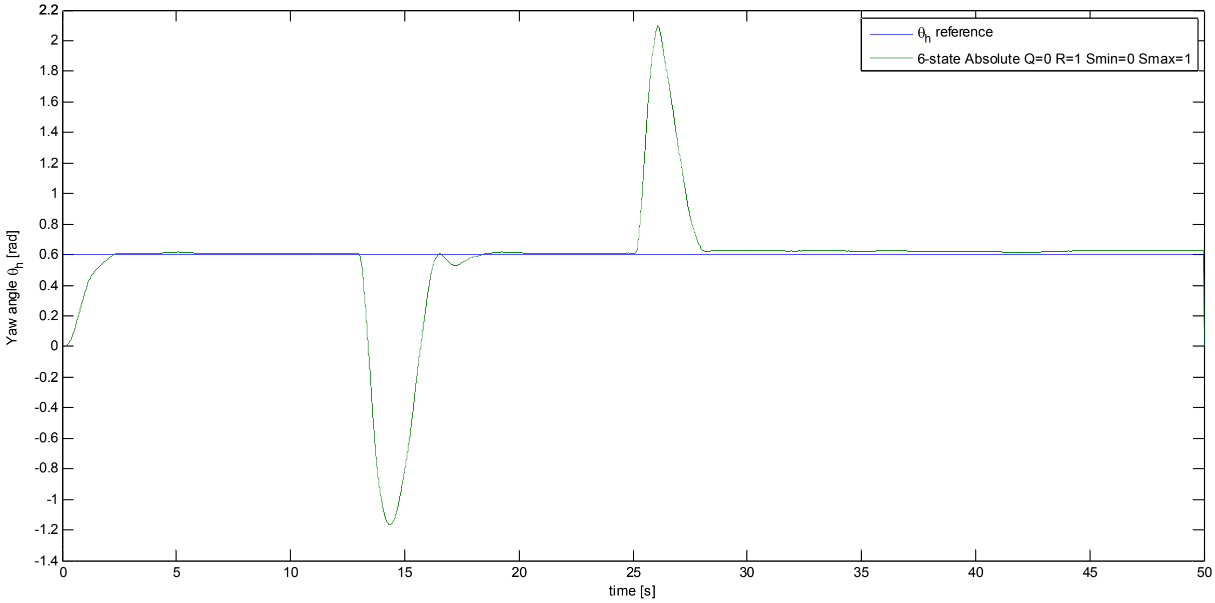

Fig. 53 - Disturbance rejection with 6-states LPV controller (yaw angle)

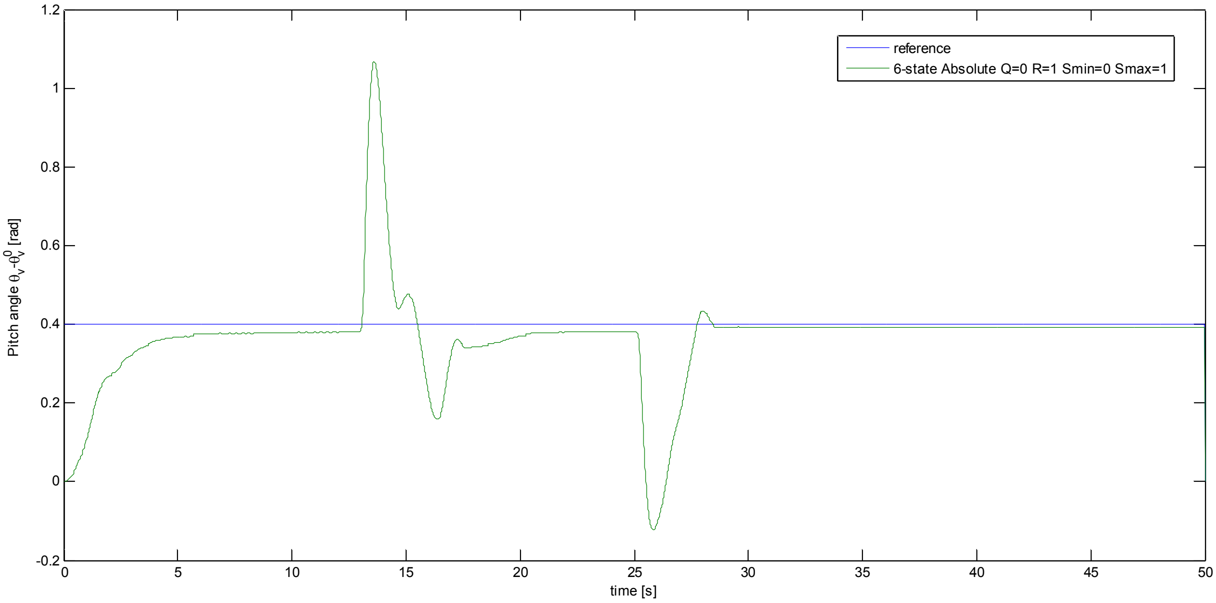

Fig. 54 – Disturbance rejection with 6-states LPV controller (pitch angle)

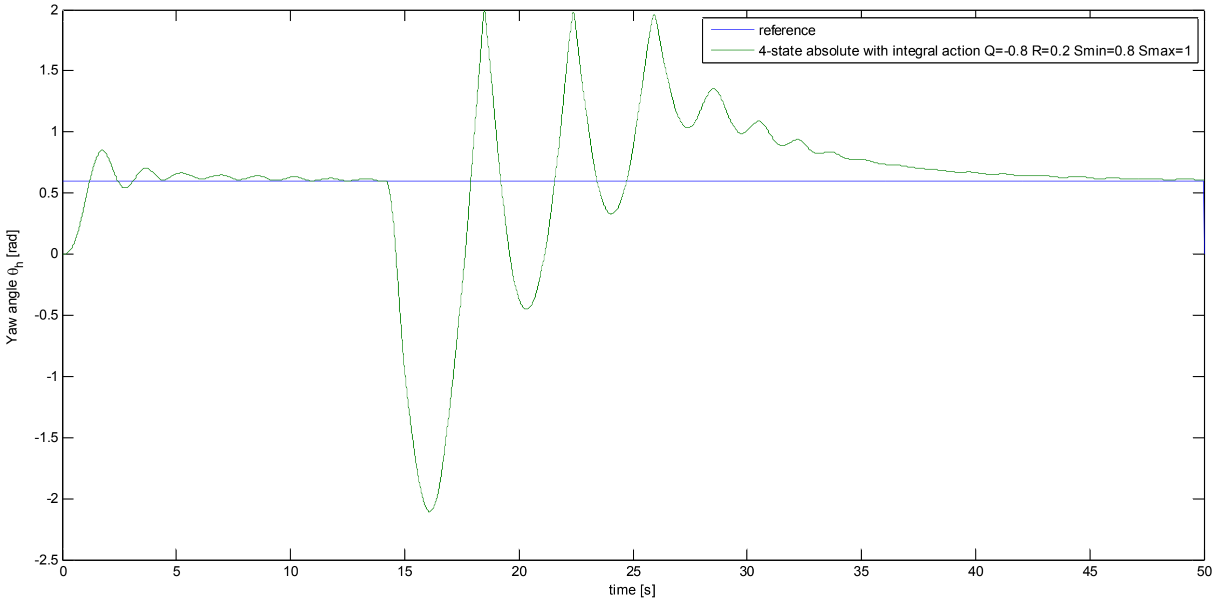

Fig. 55 – Disturbance rejection with 4-states LPV controller and integral action (yaw angle)

5.7 – Comparison with PID control strategy

In order to evaluate the goodness of the LPV state-feedback control strategy, a comparison with a PID control strategy is carried out.

Two controllers were tuned for both the yaw and the pitch angle, by applying the Ziegler-Nichols methods for PID tuning.

The step response of the pitch angle is S-shaped and can be approximated with the time response of a first-order system with a time delay as in ***Fig.** 57*. Thus, the open-loop Ziegler-Nichols method can be used to tune the PID controller.

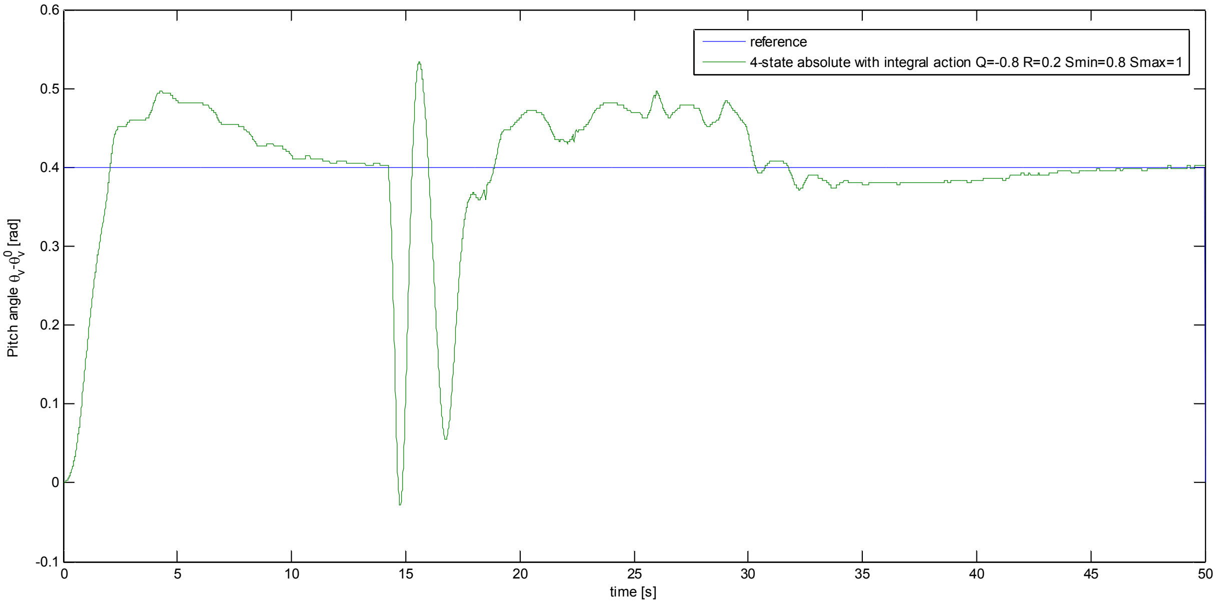

Fig. 56 - Disturbance rejection with 4-states LPV controller and integral action (pitch angle)

The slope of the tangent at the inflection point of the reaction curve is $S = 0.434$, while the time delay is $t_d = 1.12s$. Hence, the optimum gains are obtained as $k_P = 1.2 / SL = 2.47$, $k_I = k_P / T_I = k_P / 2t_d = 1.10$ and $k_D = k_P T_D = 0.5 k_P t_d = 0.69$.

The step response of the yaw angle is not S-shaped, so the open-loop method can not be used. Hence, the closed-loop method must be used: the loop is closed and the proportional gain is increased until the system becomes marginally stable and continuous oscillations begin (*Fig. 58*).

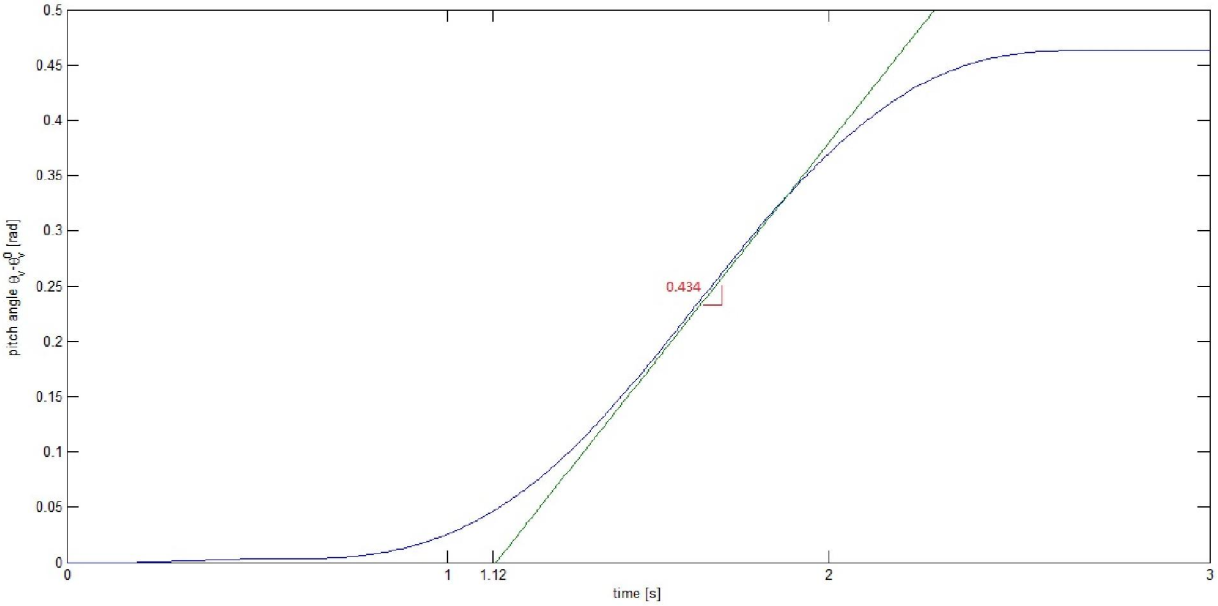

Fig. 57 – Open-loop Ziegler-Nichols method: step response of pitch angle with u_v = 1 V

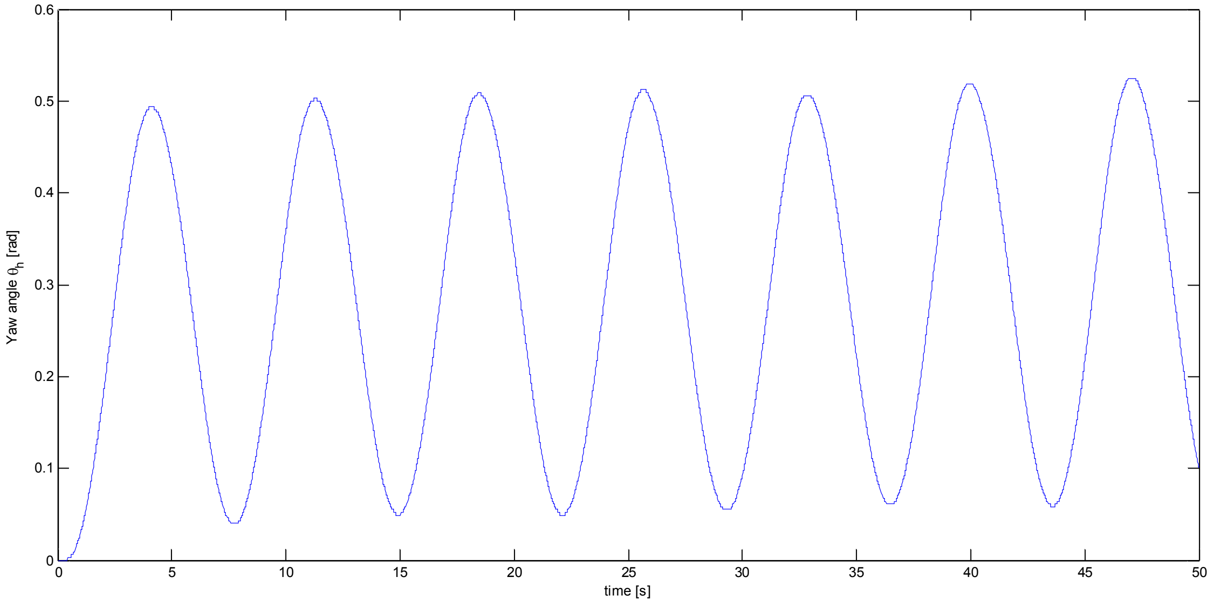

Fig. 58 – Closed-loop Ziegler-Nichols method: response of yaw angle with $K_u = 1.3$

The corresponding gain is $K_u = 1.3$ and the period of oscillation is $P_u = 7s$. These values lead to the following gains of the PID controller: $k_p = 1.6k_u = 2.08$, $k_I = 2k_P / P_u = 0.59$ and $k_D = k_P P_u / 8 = 1.82$.

The system was tested with the angle trajectory (5.3.12), and results are shown in ***Fig. 59***. In order to decrease the oscillations of the yaw angle and to make the response of the pitch angle faster, the constants were modified according to ***Table 21***.

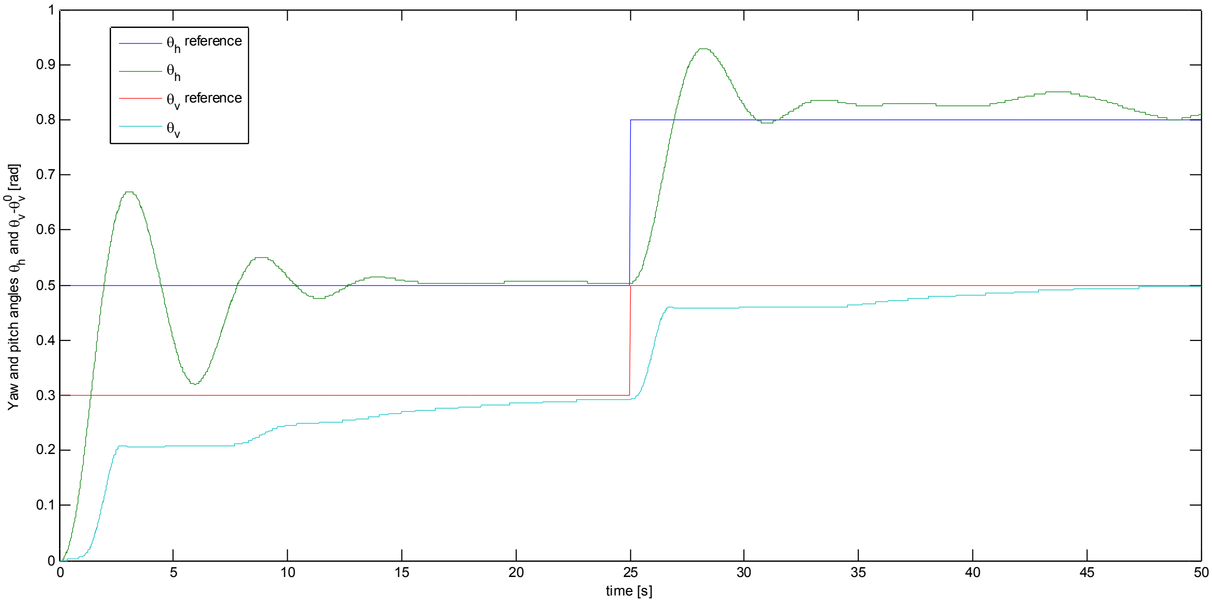

Fig. 59 – PID control of yaw and pitch angles with constants tuned by Ziegler-Nichols methods

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
k_P	Decrease	Increase	Small change	Decrease	Degrade
k_I	Decrease	Increase	Increase	Decrease significantly	Degrade
k_D	Minor decrease	Minor decrease	Minor decrease	No effect in theory	Improve if k_D is small

Table 21 – Effects of increasing a parameter independently

Constants that assure performances comparable to those obtained with the 6-states LPV controller were found: $k_{Ph} = 3.12$, $k_{Ih} = 0.71$, $k_{Dh} = 4.55$, $k_{Pv} = 2.96$, $k_{Iv} = 1.32$ and $k_{Dv} = 1.38$. Results without disturbances are shown in ***Fig. 60*** and ***Fig. 61***, while results when some disturbances are applied to the TRMS are shown in ***Fig. 62*** and ***Fig. 63***.

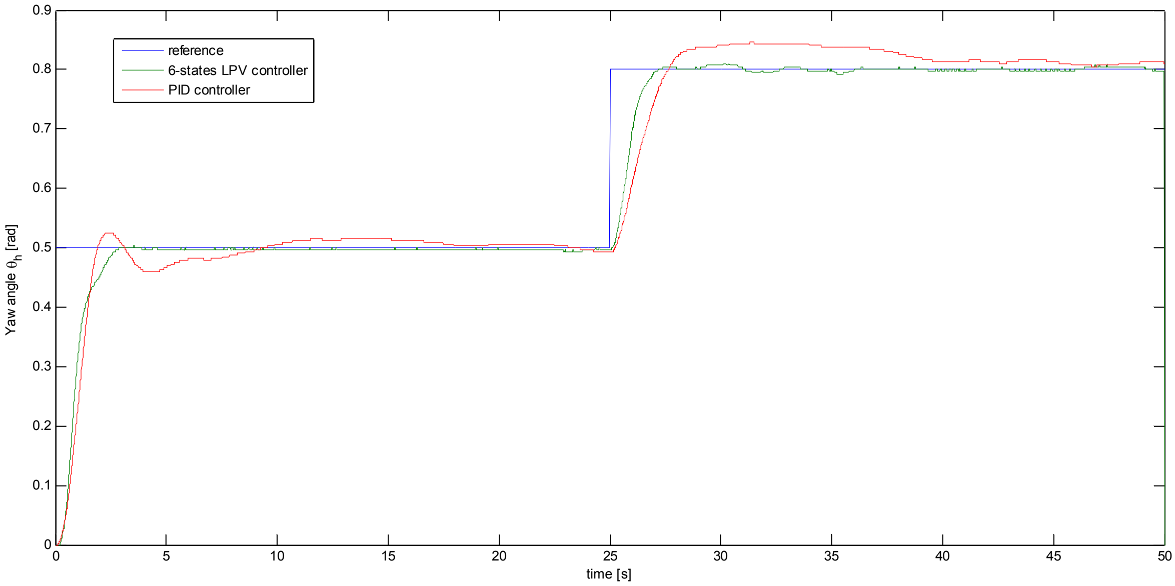

Fig. 60 – Comparison between 6-states LPV controller and PID controller (yaw angle)

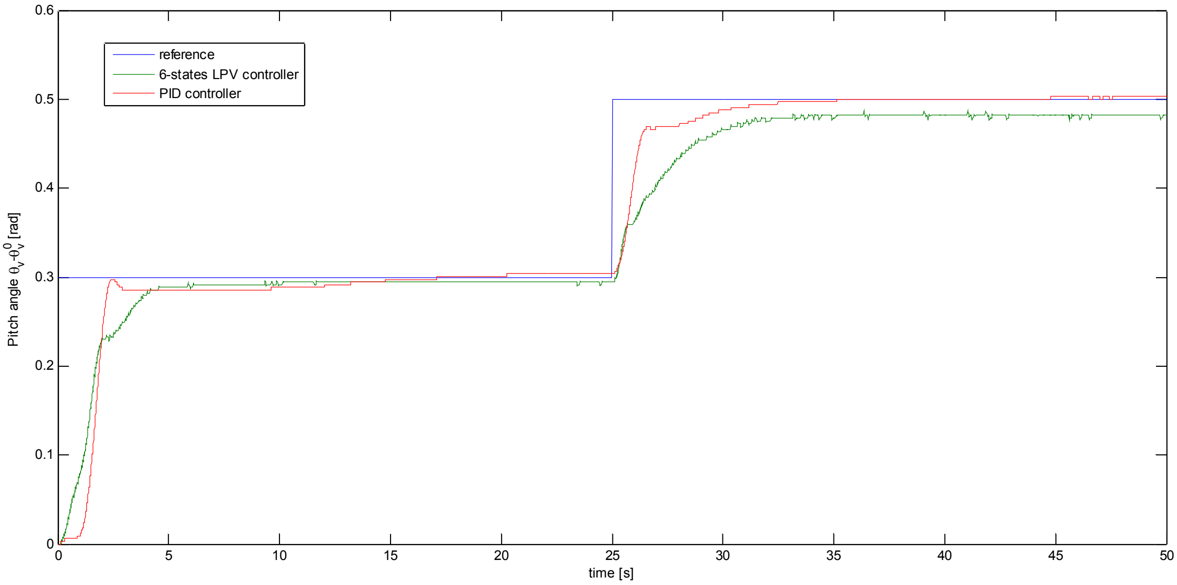

Fig. 61 - Comparison between 6-states LPV controller and PID controller (pitch angle)

It can be seen that the system controlled with PIDs fails in rejecting the disturbances even though its performances in following a stepwise reference are similar to those of the 6-states LPV controller.

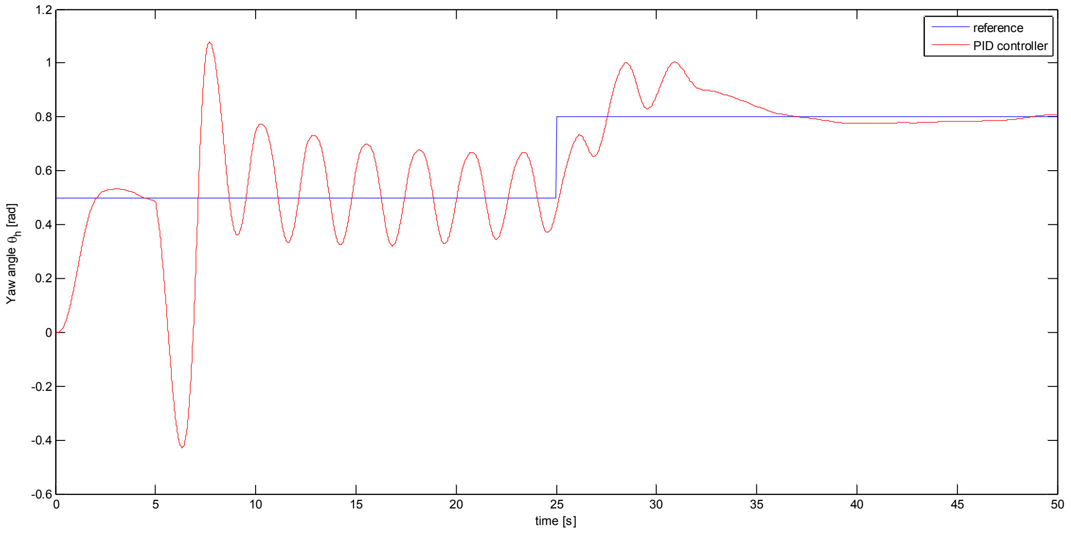

Fig. 62 - Results with PID controller in presence of disturbances (yaw angle)

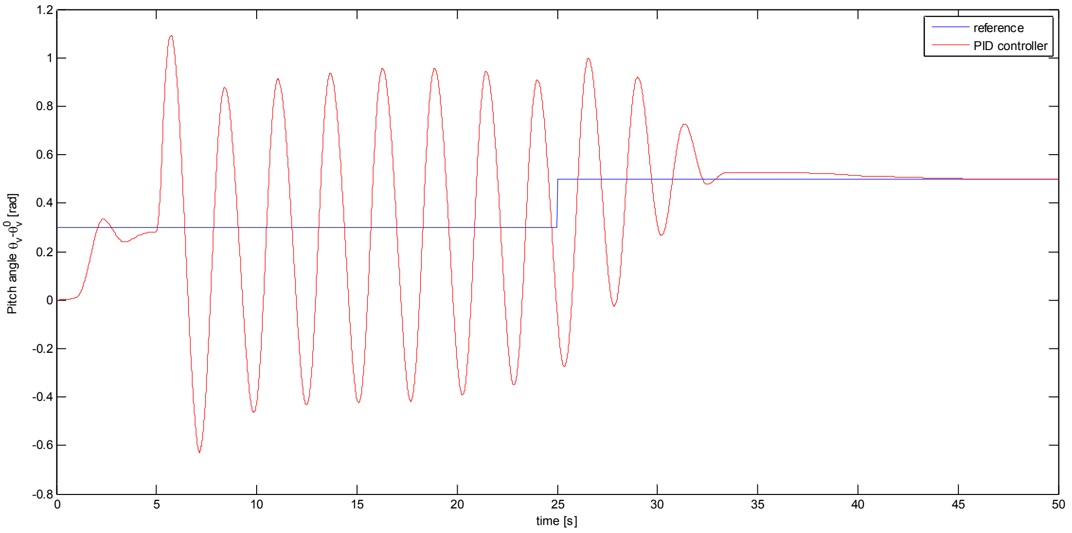

Fig. 63 - Results with PID controller in presence of disturbances (pitch angle)

5.8 – Summary

In this chapter an observer and a controller have been designed according to the LPV theory. The observer proved to work well both when applied to the simulation model and when applied to the real TRMS, as the noise corrupting the measurements of the sensors was very small. The controller was designed with different LPV models:

- a 6-states LPV model, obtained from the non-linear model by neglecting the dynamics of the armature currents;
- a 4-states LPV model with local control loops for the tail/main rotor, where the problem of controlling the TRMS is split into simpler control problems;
- a 4-states LPV model with local control loops for the tail/main rotor with integral action, where the state of the aeromechanical model is augmented in order to add an integral

action which permits to drive the system to the desired yaw and pitch angles with zero steady-state error.

The 6-states LPV controller proved to give the best transient response even though, when tested with the real system, exhibited a steady-state error different from zero. On the other hand, the 4-states LPV model with local control loops, augmented to add an integral action, had the worst transient response but was able to drive the TRMS to the desired angles with zero steady-state error.
The 6-states LPV controller proved to reject rapidly disturbances and to give better results than those obtained with PID controllers. Because of all these reasons, it was chosen as the nominal controller to which fault tolerance mechanisms will be applied in next chapter.

6 –LPV FAULT-TOLERANT CONTROL FOR THE TRMS

6.1 – Introduction

Recently, virtual sensors and virtual actuators for linear systems have been proposed as a fault accommodation approach (see, e.g. **[Blanke, 2003]**). The main idea of this Fault Tolerant Control (FTC) method is to reconfigure the control loop such that the nominal controller could be still used without need of retuning. The plant with the faulty sensor/actuator is modified adding the virtual sensor/actuator block that masks the fault and allows the controller to see the same plant as before the fault occurrence.

In case of a sensor break-down, the reconfiguration block results from the application of a Luenberger observer to reconstruct the faulty output. It is called a *"virtual sensor"*, because it reconstructs that element y_i of the output vector from the other measured output signals that the faulty sensor does no longer measure.

In case of an actuator break-down, the reconfiguration block can be obtained in a dual way. In this case, it is called *"virtual actuator"*, because it acts like the faulty actuator but replaces the effect of this actuator by using the control input of the other actuators appropriately. The reconfigured controller, that will be applied to the faulty plant, consists of the nominal controller and the reconfiguration block.

The faulty process and the reconfiguration block together become the so-called reconfigured plant, which is connected to the nominal controller (***Fig. 64***). If the reconfigured plant behaves like the nominal plant, the loop consisting of the reconfigured plant and the controller behaves like the nominal closed-loop system. This is true for an arbitrary nominal controller.

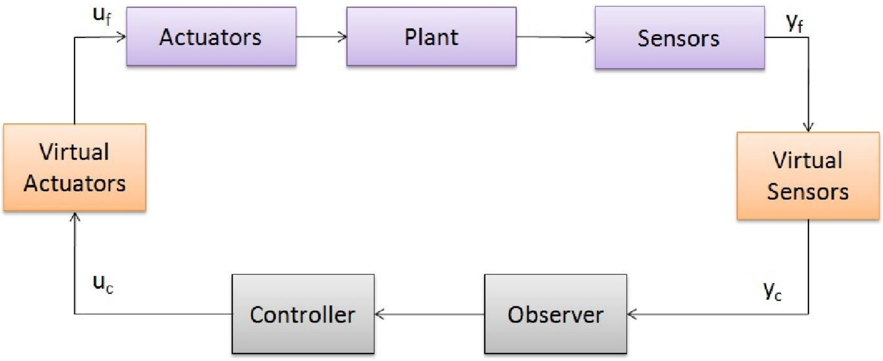

Fig. 64 – Virtual sensors and virtual actuators

In order to implement an active fault tolerant strategy, a fault estimation algorithm is needed. The fault estimation problem can be formulated as a parameter estimation problem in such a way that any parameter estimation algorithm (such as least squares, generalized/extended least squares, instrumental variables, maximum likelihood, extended Kalman filter and others) could be used.

Hereafter, the virtual sensor/actuator theory will be extended to LPV systems, so that it can be applied to the case of study that is the Twin Rotor MIMO System.

6.2 – Reconfiguration problem

Before explaining the reconfiguration method, the problem to be solved is formally stated. The model of the nominal process is supposed to be given in state-space LPV discrete-time form:

$$\begin{cases}\mathbf{x}(k+1) = \mathbf{A}(\boldsymbol{\psi}(k))\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{y}(k) \quad = \mathbf{C}\mathbf{x}(k) \\ \mathbf{x}(0) \quad = \mathbf{x}_0\end{cases} \tag{6.2.1}$$

The nominal process is stabilised by a nominal controller with output $\mathbf{u}(k)$ and inputs $\hat{\mathbf{x}}(k)$, the actual estimated state, and the actual reference state $\mathbf{x}_{ref}(k)$. The controller is supposed to be an LPV feedback controller with a feedforward action:

$$\mathbf{u}_c(k) = \mathbf{K}(\boldsymbol{\psi}(k))\left(\hat{\mathbf{x}}(k) - \mathbf{x}_{ref}(k)\right) + \mathbf{u}_{ff}(k) \tag{6.2.2}$$

The state $\hat{\mathbf{x}}(k)$ is supposed to be estimated through the nominal LPV state observer:

$$\begin{cases}\hat{\mathbf{x}}(k+1) = \mathbf{A}(\boldsymbol{\psi}(k))\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}(\boldsymbol{\psi}(k))\left(\hat{\mathbf{y}}(k) - \mathbf{y}_c(k)\right) \\ \hat{\mathbf{y}}(k) \quad = \mathbf{C}\hat{\mathbf{x}}(k) \\ \hat{\mathbf{x}}(0) \quad = \hat{\mathbf{x}}_0\end{cases} \tag{6.2.3}$$

The process, the observer and the controller form the nominal control loop for $\mathbf{u}(k) = \mathbf{u}_c(k)$ and $\mathbf{y}(k) = \mathbf{y}_c(k)$:

$$\begin{cases}\mathbf{x}(k+1) = \mathbf{A}(\boldsymbol{\psi}(k))\mathbf{x}(k) + \mathbf{B}\mathbf{K}(\boldsymbol{\psi}(k))\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}_{ff}(k) \\ \hat{\mathbf{x}}(k+1) = \left(\mathbf{A}(\boldsymbol{\psi}(k)) + \mathbf{B}\mathbf{K}(\boldsymbol{\psi}(k))\right)\hat{\mathbf{x}}(k) + \mathbf{B}\mathbf{u}_{ff}(k) + \mathbf{L}(\boldsymbol{\psi}(k))\left(\mathbf{C}\hat{\mathbf{x}}(k) - \mathbf{y}(k)\right) \\ \mathbf{y}(k) \quad = \mathbf{C}\mathbf{x}(k) \\ \mathbf{x}(0) \quad = \mathbf{x}_0 \\ \hat{\mathbf{x}}(0) \quad = \mathbf{x}_0\end{cases} \tag{6.2.4}$$

The control loop is assumed to be stable and to satisfy the performance requirements concerning the set-point tracking.

6.2.1 – Fault cases

In case of a fault of sensor *i*, the *i*-th row of the matrix $\mathbf{C}$ is changed and the new matrix is denoted by $\mathbf{C}_f$. The faults are considered to be multiplicative, which means that the output of sensor *i* is multiplied by a constant ϕ_i, thus:

$$\mathbf{C}_f = \mathbf{\Phi}\mathbf{C} \tag{6.2.5}$$

with $\mathbf{\Phi} = diag\left\{\phi_1, \phi_2, ..., \phi_{n_y}\right\}$ and $0 \le \phi_i \le 1$ such that the extreme values $\phi_i = 0$ and $\phi_i = 1$ denote a total loss of the *i*-th sensor and the healthy *i*-th sensor, respectively.

In case of a fault of actuator *j*-th, the *j*-th column of the matrix $\mathbf{B}$ is changed and the new matrix is denoted by $\mathbf{B}_f$. Analogously, the faults are considered to be multiplicative, that is, the output of actuator *j* is multiplied by a constant β_j, thus:

$$\mathbf{B}_f = \mathbf{B}\boldsymbol{\beta} \tag{6.2.6}$$

with $\boldsymbol{\beta} = diag\left\{\beta_1, \beta_2, ..., \beta_{n_u}\right\}$ and $0 \le \beta_j \le 1$ such that the extreme values $\beta_j = 0$ and $\beta_j = 1$ denote a total loss of the *j*-th actuator and the healthy *j*-th actuator, respectively.

It is assumed that the faulty process is still controllable and observable.

The input and output of the faulty plant are denoted by $\mathbf{u}_f$ and $\mathbf{y}_f$, respectively.

6.2.2 – Reconfiguration task

The goal is to find a reconfigured controller that makes the closed-loop system satisfy the following conditions:

Strong reconfiguration goal – The controller should make the reconfigured control loop behave in exactly the same way as the nominal control loop, that is, the relation $\mathbf{y}_f(k)=\mathbf{y}(k)$ should hold. This strong goal is only feasible in very special cases. Therefore, a weaker goal is defined in terms of the dynamical and the static behaviour of the reconfigured loop.

Weak reconfiguration goal – The weak goal consists of a static and a dynamical part. Considering the static behaviour, the output $\mathbf{y}_f$ of the reconfigured loop should have the same value as for the nominal system. This means that the relation $\mathbf{y}_f(k)\rightarrow\mathbf{y}(k)$ for $k\rightarrow\infty$ should hold. The transient behaviour is determined by the poles and zeros of the system which should not differ significantly in the nominal and the reconfigured control loop. Additional poles (and zeros) are allowed only if they are fast enough not to dominate the system behaviour.

6.3 – Virtual sensors

Hereafter, a reconfiguration block that reconstructs a measurement y_i when a fault in the *i*-th sensor occurs is described. The main idea is to use an observer for the faulty system, which represents the main part of the reconfiguration block to be built. This block is called *virtual sensor* due to its function of replacing a faulty sensor.
The faulty system under consideration is described by the state-space model:

$$\begin{cases}\mathbf{x}_f(k+1)=\mathbf{A}\left(\psi(k)\right)\mathbf{x}_f(k)+\mathbf{B}\mathbf{u}_f(k)\\ \mathbf{y}_f(k)=\mathbf{C}_f\mathbf{x}_f(k)\\ \mathbf{x}_f(0)=\mathbf{x}_{f0}\end{cases} \tag{6.3.1}$$

where the sensor failure is reflected by the matrix $\mathbf{C}_f$ and $\mathbf{x}_f$, $\mathbf{u}_f$ and $\mathbf{y}_f$ are the system state, input and output vector in faulty conditions.
If the condition $Kern\left(\mathbf{C}_f\right)\subseteq Kern\left(\mathbf{C}\right)$ is satisfied, where *Kern* denotes the kernel of a matrix, the complete output vector $\mathbf{y}$ can be reconstructed from $\mathbf{y}_f$. This condition means that the measurement information obtained by the healthy output vector $\mathbf{y}$ is the same as the information obtained by the remaining sensor through $\mathbf{y}_f$. The condition can be written in an equivalent form as:

$$rank\left(\mathbf{C}_f\right)=rank\begin{pmatrix}\mathbf{C}\\ \mathbf{C}_f\end{pmatrix} \tag{6.3.2}$$

If condition (6.3.2) is satisfied, then the reconfiguration problem is solved by choosing:

$$\mathbf{y}(k)=\mathbf{P}\mathbf{y}_f(k) \tag{6.3.3}$$

where:

$$\mathbf{P}=\mathbf{C}\mathbf{C}_f^{+}=\mathbf{C}\mathbf{C}_f^{T}\left(\mathbf{C}_f\mathbf{C}_f^{T}\right)^{-1} \tag{6.3.4}$$

Notice that by choosing $\mathbf{P}$ as in (6.3.4), $\mathbf{C}=\mathbf{P}\mathbf{C}_f$.
Situations where the condition (6.3.2) is satisfied include the following:

- The fault has changed the sensitivity of the sensor, but the signal is not completely lost. Hence $y_f = ay$ holds for some scalar *a*. This is the case when γ_i in matrix $\mathbf{\Gamma}$ of (6.2.5) is different from zero.
- The faulty sensor has at least one parallel redundant sensor. The matrix **P** switches the output to the redundant sensor.
- An analytic relation between the faulty output and several other output values exists, which can be reformulated by using the matrix **P**.

The later two cases are only possible if $\mathbf{C}$ does not have full rank.

The new control structure obtained by choosing $\mathbf{y}(k)$ as in (6.3.3) can be interpreted as consisting of a reconfiguration block:

$$\begin{aligned} &\mathbf{y}_c(k) = \mathbf{P}\mathbf{y}_f(k) + \mathbf{y}_\Delta \\ &\mathbf{u}_f(k) = \mathbf{u}_c(k) \end{aligned} \tag{6.3.5}$$

That is, if the condition (6.3.2) holds, the virtual sensor is a static reconfiguration block.

The general case is, however, when the condition (6.3.2) is violated. In this case, the reconfiguration block includes a state observer and a direct feedthrough.

6.3.1 – Definition of the virtual sensor

Consider the faulty plant (6.3.1). The virtual sensor is described as the system:

$$\begin{cases} \mathbf{x}_v(k+1) = \mathbf{A}_v\left(\boldsymbol{\psi}(k)\right)\mathbf{x}_v(k) + \mathbf{B}_v\mathbf{u}_c(k) - \mathbf{L}_v\left(\boldsymbol{\psi}(k)\right)\mathbf{y}_f(k) \\ \mathbf{u}_f(k) = \mathbf{u}_c(k) \\ \mathbf{y}_c(k) = \mathbf{C}_v\mathbf{x}_v(k) + \mathbf{P}_v\mathbf{y}_f(k) \\ \mathbf{x}_v(0) = \mathbf{x}_{v0} \end{cases} \tag{6.3.6}$$

with the state $\mathbf{x}_v \in \mathbb{R}^n$ and matrices:

$$\mathbf{A}_v\left(\boldsymbol{\psi}(k)\right) = \mathbf{A}\left(\boldsymbol{\psi}(k)\right) + \mathbf{L}_v\left(\boldsymbol{\psi}(k)\right)\mathbf{C}_f \tag{6.3.7}$$

$$\mathbf{B}_v = \mathbf{B} \tag{6.3.8}$$

$$\mathbf{C}_v = \mathbf{C} - \mathbf{P}_v\mathbf{C}_f \tag{6.3.9}$$

$\mathbf{P}_v$ and $\mathbf{L}_v$ denote matrices that can be freely chosen; notice that $\mathbf{L}_v$ is scheduled by the vector of varying parameters $\boldsymbol{\psi}(k)$.

The main part of the virtual sensor is the state observer with the state vector $\mathbf{x}_v(k)$. The complete output $\mathbf{y}_c(k)$ of the plant can be approximately determined: $\mathbf{y}_c(k) \approx \mathbf{C}\mathbf{x}_v(k)$. This observation result is improved by using the available sensor values and by observing only the difference between the nominal and the faulty output. In a generalised form, this approach is represented by the output equation of (6.3.6) where the matrix $\mathbf{P}_v$ is a design parameter. For $\mathbf{P}_v = \mathbf{0}$ only observed values are used.

6.3.2 – Model of the reconfigured plant

The plant together with the virtual sensor and the nominal state observer is described by equations (6.3.1), (6.2.3) and (6.3.6):

$$\begin{pmatrix} \mathbf{x}_f(k+1) \\ \mathbf{x}_v(k+1) \\ \hat{\mathbf{x}}(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{A}(\boldsymbol{\psi}(k)) & \mathbf{0} & \mathbf{0} \\ -\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f & \mathbf{A}(\boldsymbol{\psi}(k)) + \mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f & \mathbf{0} \\ -\mathbf{L}(\boldsymbol{\psi}(k))\mathbf{P}_v\mathbf{C}_f & -\mathbf{L}(\boldsymbol{\psi}(k))\left(\mathbf{C} - \mathbf{P}_v\mathbf{C}_f\right) & \mathbf{A}(\boldsymbol{\psi}(k)) + \mathbf{L}(\boldsymbol{\psi}(k))\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}_f(k) \\ \mathbf{x}_v(k) \\ \hat{\mathbf{x}}(k) \end{pmatrix} + \begin{pmatrix} \mathbf{B} \\ \mathbf{B} \\ \mathbf{B} \end{pmatrix} \mathbf{u}_c(k) \tag{6.3.10}$$

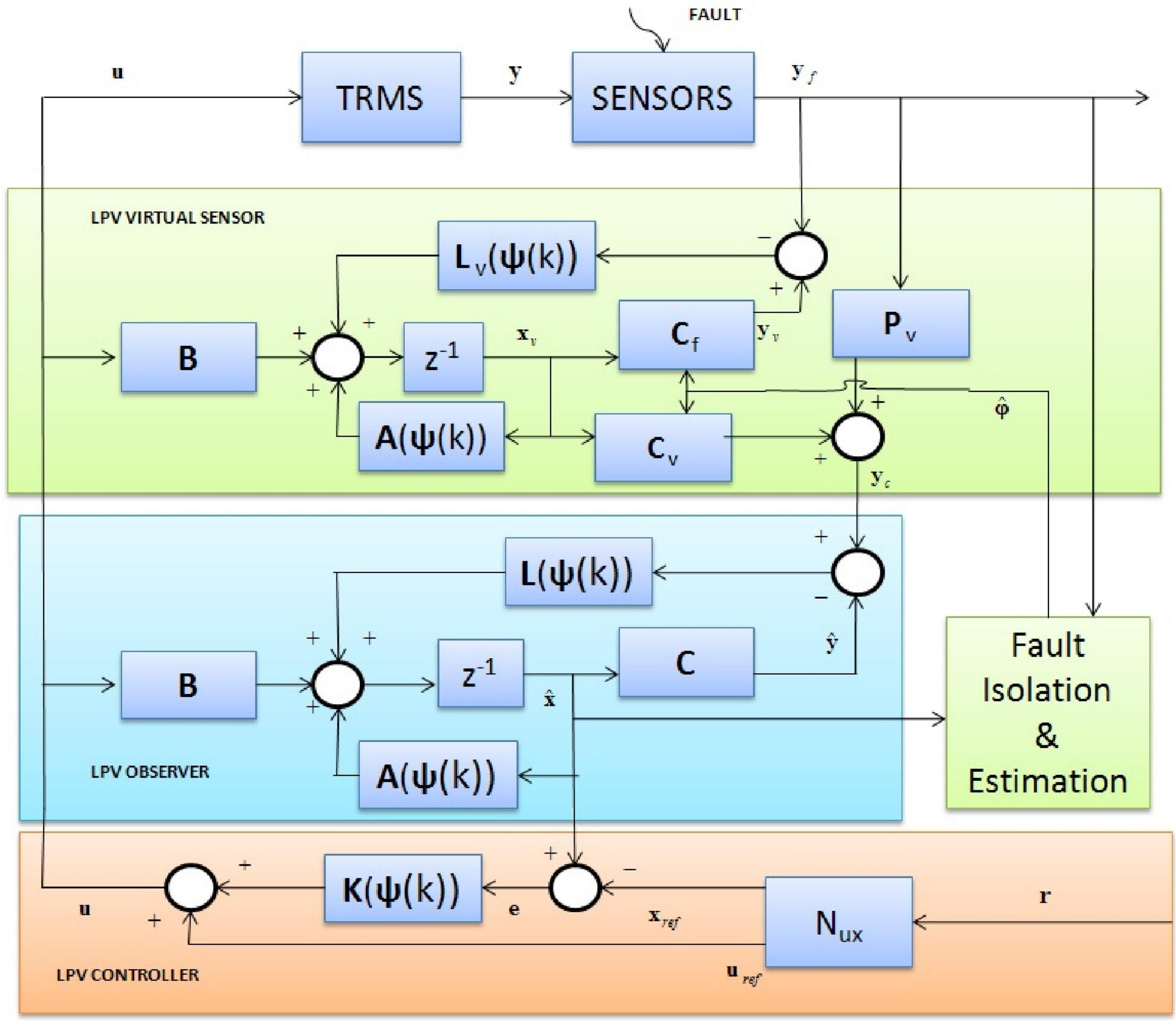

Fig. 65 – Virtual sensor scheme

A state transformation is performed in order to introduce the virtual sensor error $\mathbf{x}_\Delta(k)=\mathbf{x}_v(k)-\mathbf{x}_f(k)$ and the observation error $\mathbf{x}_\delta(k)=\hat{\mathbf{x}}(k)-\mathbf{x}_v(k)$. Then, the system (6.3.10) is equivalent to:

$$\begin{pmatrix}\mathbf{x}_f(k+1)\\ \mathbf{x}_\Delta(k+1)\\ \mathbf{x}_\delta(k+1)\end{pmatrix}=\begin{pmatrix}\mathbf{A}(\boldsymbol{\psi}(k)) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f & \mathbf{0}\\ \mathbf{0} & \mathbf{L}(\boldsymbol{\psi}(k))\left(\mathbf{P}_v\mathbf{C}_f-\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f\right) & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}(\boldsymbol{\psi}(k))\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{x}_f(k)\\ \mathbf{x}_\Delta(k)\\ \mathbf{x}_\delta(k)\end{pmatrix}+\begin{pmatrix}\mathbf{B}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix}\mathbf{u}_c(k) \quad (6.3.11)$$

For the analysis of the closed-loop behaviour the model of the reconfigured plant is combined with the linear feedback controller (6.2.2):

$$\begin{pmatrix}\mathbf{x}_f(k+1)\\ \mathbf{x}_\Delta(k+1)\\ \mathbf{x}_\delta(k+1)\end{pmatrix}=\begin{pmatrix}\mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{BK}(\boldsymbol{\psi}(k)) & \mathbf{BK}(\boldsymbol{\psi}(k)) & \mathbf{BK}(\boldsymbol{\psi}(k))\\ \mathbf{0} & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f & \mathbf{0}\\ \mathbf{0} & \mathbf{L}(\boldsymbol{\psi}(k))\left(\mathbf{P}_v\mathbf{C}_f-\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f\right) & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}(\boldsymbol{\psi}(k))\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{x}_f(k)\\ \mathbf{x}_\Delta(k)\\ \mathbf{x}_\delta(k)\end{pmatrix} \quad (6.3.12)$$

$$+\begin{pmatrix}-\mathbf{BK}(\boldsymbol{\psi}(k))\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix}\mathbf{x}_{ref}(k)+\begin{pmatrix}\mathbf{B}\\ \mathbf{0}\\ \mathbf{0}\end{pmatrix}\mathbf{u}_{ff}(k)$$

The autonomous closed-loop behaviour can be analysed by imposing $\mathbf{u}_{ff} = \mathbf{0}$ and $\mathbf{x}_{ref} = \mathbf{0}$, thus (6.3.12) simplifies to:

$$\begin{pmatrix} \mathbf{x}_f(k+1) \\ \mathbf{x}_\Delta(k+1) \\ \mathbf{x}_\delta(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{A}+\mathbf{BK}(\boldsymbol{\psi}(k)) & \mathbf{BK}(\boldsymbol{\psi}(k)) & \mathbf{BK}(\boldsymbol{\psi}(k)) \\ \mathbf{0} & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{L}(\boldsymbol{\psi}(k))\left(\mathbf{P}_v\mathbf{C}_f - \mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f\right) & \mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}(\boldsymbol{\psi}(k))\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}_f(k) \\ \mathbf{x}_\Delta(k) \\ \mathbf{x}_\delta(k) \end{pmatrix} \qquad (6.3.13)$$

Looking at this model, the *separation principle* can be applied: the matrix $\mathbf{K}(\boldsymbol{\psi}(k))$ influences the behaviour of the process state $\mathbf{x}_f(k)$ through the submatrix $\mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{BK}(\boldsymbol{\psi}(k))$ (nominal controller). $\mathbf{L}_v(\boldsymbol{\psi}(k))$ affects the behaviour of the virtual sensor error $\mathbf{x}_\Delta(k)$ through the submatrix $\mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}_v(\boldsymbol{\psi}(k))\mathbf{C}_f$ (LPV virtual sensor). Finally, the observer error $\mathbf{x}_\delta$ is affected by the matrix $\mathbf{L}(\boldsymbol{\psi}(k))$ through the submatrix $\mathbf{A}(\boldsymbol{\psi}(k))+\mathbf{L}(\boldsymbol{\psi}(k))\mathbf{C}$ (LPV state observer).

The set σ of eigenvalues of the reconfigured closed-loop system (6.3.13) consists of the set of eigenvalues of the nominal closed-loop system (6.2.4) plus the set of eigenvalues of the virtual sensor (6.3.6) and the set of eigenvalues of the LPV observer (6.2.3):

$$\sigma = \sigma\{\mathbf{A}+\mathbf{BK}\} \cup \sigma\{\mathbf{A}+\mathbf{L}_v\mathbf{C}_f\} \cup \sigma\{\mathbf{A}+\mathbf{LC}\}$$

Thus, the closed loop LPV controller, the LPV virtual sensor and LPV state observer can be designed independently.

The analysis has shown how the virtual sensor works. The direct feedthrough $\mathbf{P}$ reconstructs or at least approximates the output $\mathbf{y}_c$ of the faultless plant from the remaining output $\mathbf{y}_f$. If the condition (6.3.2) is satisfied and $\mathbf{P}$ is chosen according to (6.3.4), the virtual sensor shrinks to a static reconfiguration block because $\mathbf{C}_v = \mathbf{0}$. This solution to the reconfiguration problem coincides with the solution obtained by the model-matching approach. The strong reconfiguration goal is satisfied. If the condition (6.3.2) is not satisfied, the virtual sensor reconstructs the missing sensor information.

6.3.3 – Polytopic LPV Virtual Sensor

The LPV virtual sensor (6.3.6) can be defined in a polytopic way as:

$$\begin{cases} \mathbf{x}_v(k+1) = \sum_{j=1}^{N} \alpha_k^j(\boldsymbol{\psi}(k))\left[\mathbf{A}_j\mathbf{x}_v(k)+\mathbf{Bu}(k)+\mathbf{L}_{v,j}\left(\mathbf{y}_v(k)-\mathbf{y}_f(k)\right)\right] \\ \mathbf{y}_v(k) = \mathbf{C}_f\mathbf{x}_v(k) \end{cases} \qquad (6.3.14)$$

where $\mathbf{x}_v(k) \in \mathbb{R}^{n_x}$ and $\mathbf{y}_v(k) \in \mathbb{R}^{n_y}$ are the virtual state and output, respectively. The matrices $L_{v,j} \in \mathbb{R}^{n_x \times n_y}$ are the gains of the LPV virtual sensor defined for each j^{th} model.

The reconstruction of the system output vector $\mathbf{y}_c(k)$ is given by:

$$\mathbf{y}_c(k) = \mathbf{P}\mathbf{y}_f(k) + \mathbf{C}\mathbf{x}_v(k) - \mathbf{P}\mathbf{C}_f\mathbf{x}_v(k) \qquad (6.3.15)$$

6.3.4 – LPV Virtual Sensor Design

Resuming, the design of LPV virtual sensor implies selecting:

- matrices $\mathbf{K}_j$ of (5.1.5) in order to guarantee closed-loop stability of the original system assuming that the pair $(\mathbf{A}(\boldsymbol{\psi}(k)),\mathbf{B})$ is stabilizable for all $\boldsymbol{\psi} \in \boldsymbol{\Psi}$;

- matrices $\mathbf{L}_{v,j}$ of (6.3.14) in order to correct the LPV virtual sensor error of the reconfigured system assuming that the pair $\left(\mathbf{A}(\boldsymbol{\psi}(k)), \mathbf{C}_f\right)$ is detectable for all $\boldsymbol{\psi} \in \boldsymbol{\Psi}$;
- matrices $\mathbf{L}_j$ of (5.1.7) in order to correct the LPV state observer error of the reconfigured system assuming that the pair $\left(\mathbf{A}(\boldsymbol{\psi}(k)), \mathbf{C}\right)$ is detectable for all $\boldsymbol{\psi} \in \boldsymbol{\Psi}$;
- a matrix $\mathbf{P}$ such that $\mathbf{C} = \mathbf{P}\mathbf{C}_f$ assuming that the rank condition (6.3.2) is satisfied.

The design of the LPV virtual sensor is done by applying LMI techniques described in *Chapter 2* to place the poles in a desired region of the complex plane. Thus, equation (2.3.13) becomes:

$$\begin{cases} \begin{pmatrix} -r\mathbf{X} & q\mathbf{X} + \mathbf{A}_j^T\mathbf{X} + \mathbf{C}_f^T\boldsymbol{\Gamma}_j & 0 & 0 \\ q\mathbf{X} + X\mathbf{A}_j + \boldsymbol{\Gamma}_j^T\mathbf{C}_f & -r\mathbf{X} & 0 & 0 \\ 0 & 0 & S_{\min}\mathbf{X} - \frac{1}{2}\left(\mathbf{A}_j^T\mathbf{X} + \mathbf{X}\mathbf{A}_j + \mathbf{C}_f^T\boldsymbol{\Gamma}_j + \boldsymbol{\Gamma}_j^T\mathbf{C}_f\right) & 0 \\ 0 & 0 & 0 & -S_{\max}\mathbf{X} + \frac{1}{2}\left(\mathbf{A}_j^T\mathbf{X} + \mathbf{X}\mathbf{A}_j + \mathbf{C}_f^T\boldsymbol{\Gamma}_j + \boldsymbol{\Gamma}_j^T\mathbf{C}_f\right) \end{pmatrix} < 0 \\ \mathbf{X} > 0 \end{cases} \tag{6.3.16}$$

Finally, the vertices of the virtual sensor gains are then obtained as:

$$\mathbf{L}_{v,j} = \left(\boldsymbol{\Gamma}_j\mathbf{X}^{-1}\right)^T \tag{6.3.17}$$

6.3.6 – Sensor fault isolation and estimation

Hereafter, the procedure to isolate and estimate the multiplicative sensor faults is explained. The hypothesis of having at disposal a fault detection algorithm is made. This means that, as soon as a fault appears in the system, the fault detection algorithm is able to detect that there is a fault, and the procedure for fault isolation and estimation is started.

This procedure is based on the fact that, in presence of an abrupt fault in a sensor, there is a difference between the output of the sensor and the corresponding output of the state observer. Such a difference can be used to estimate the magnitude of the fault through a parameter estimation approach.

The approach to be used is the so-called "recursive least-square method" with a forgetting factor $\varsigma(k)$ **[Ljung, 1987]**. The linear regression is written down as:

$$y_{f,i} = \phi_i \sum_{l=1}^{n_y} c_{i,l}\hat{x}_l(k) \tag{6.3.18}$$

where $y_{f,i}$ is the i^{th} output sensor when a fault ϕ_i occurs and $\mathbf{c}_i$ is the i^{th} row of matrix $\mathbf{C}$.

The fault estimation is thus given by:

$$\left\{\begin{array}{l} P_i(k) = \dfrac{\left[P_i(k-1) - \dfrac{P_i(k-1)^2 \left(\sum\limits_{l=1}^{n_y} c_{i,l}\hat{x}_l(k) \right)^2}{\varsigma_i(k) + P_i(k-1)\left(\sum\limits_{l=1}^{n_y} c_{i,l}\hat{x}_l(k) \right)^2} \right]}{\varsigma_i(k)} \\ \hat{\phi}_i(k) = \hat{\phi}_i(k-1) + \Phi_i(k)\left[y_{f,i}(k) - \hat{\phi}_i(k-1)\sum\limits_{l=1}^{n_y} c_{i,l}\hat{x}_l(k) \right] \\ \Phi_i(k) = \dfrac{P_i(k-1)\sum\limits_{l=1}^{n_y} c_{i,l}\hat{x}_l(k)}{\varsigma_i(k) + P_i(k-1)\left(\sum\limits_{l=1}^{n_y} c_{i,l}\hat{x}_l(k) \right)^2} \end{array}\right. \qquad (6.3.19)$$

where $P_i(k)$ is the variance of the fault estimation $\hat{\phi}_i(k)$ and $\Phi_i(k)$ is the gain of the algorithm at instant *k*. It should be noticed that if no fault tolerance mechanism is used, the output of the state observer would eventually converge to the sensor one resulting in a fault estimation approximately equal to 1 (no fault) even when a fault occurs in the system. However, the presence of the fault tolerance mechanism holds the two outputs at different values.

The method explained here has proved to be able not only to estimate the fault magnitude, but also to isolate the fault (that is, to identify which sensor is faulty), taking into account that a fault in a sensor has been detected.

6.3.7 – Application to TRMS

The Twin Rotor MIMO System has 4 sensors, for measuring $\omega_h, \theta_h, \omega_v, \theta_v$ respectively. When a fault occurs (denoted with $\phi_1, \phi_2, \phi_3, \phi_4$ respectively), the fault identification/estimation approach is applied to discover which sensor is faulty and which is the corresponding magnitude. As each sensor corresponds to a state-space variable, equation (6.3.19) can be rewritten in a simplified way. For example, to estimate the magnitude of the fault in the ω_h sensor ϕ_1, equation (6.3.19) reduces to:

$$\left\{\begin{array}{l} P_1(k) = \dfrac{\left[P_1(k-1) - \dfrac{P_1(k-1)^2 \hat{\omega}_h(k)^2}{\varsigma_1(k) + P_1(k-1)\hat{\omega}_h(k)^2} \right]}{\varsigma_1(k)} \\ \hat{\phi}_1(k) = \hat{\phi}_1(k-1) + \Phi_1(k)\left[\omega_h^{f.sensor}(k) - \hat{\phi}_1(k-1)\hat{\omega}_h(k) \right] \\ \Phi_1(k) = \dfrac{P_1(k-1)\hat{\omega}_h(k)}{\varsigma_1(k) + P_1(k-1)\hat{\omega}_h(k)^2} \end{array}\right. \qquad (6.3.20)$$

The values provided from the fault estimation $\hat{\phi}_1, \hat{\phi}_2, \hat{\phi}_3, \hat{\phi}_4$ are used to apply the virtual sensor mechanism, so obtaining tolerance to the fault. In particular, if the faulty sensor is not completely broken, the rank condition (6.3.2) satisfies and the virtual sensor reduces to a static block designed in a model matching way. On the other hand, if the faulty sensor is completely broken, the rank condition (6.3.2) is no longer satisfied, hence there is the need to activate the corresponding virtual sensor.

In the first case (virtual sensor through model matching), equation (6.3.4) gives:

$$\mathbf{P} = \begin{pmatrix} 1/\hat{\phi}_1 & 0 & 0 & 0 \\ 0 & 1/\hat{\phi}_2 & 0 & 0 \\ 0 & 0 & 1/\hat{\phi}_3 & 0 \\ 0 & 0 & 0 & 1/\hat{\phi}_4 \end{pmatrix} \tag{6.3.21}$$

In the second case (virtual sensor through observer), the LPV virtual sensors (one for each sensor) are obtained by solving (6.3.16) where $\mathbf{C}_f$ is obtained from $\mathbf{C}$ by eliminating the row corresponding to the faulty sensor. For example, if the sensor that measures the tail rotor angular speed ω_h breaks ($\phi_1 = 0$), the corresponding $\mathbf{C}_f$ is:

$$\mathbf{C}_f = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.3.22}$$

In this case, as $\hat{\phi}_i$ converges to 0 without taking the exact value, a threshold is used to determine if it is the model matching or the observer that has to be applied.

6.4 – Virtual sensors results

6.4.1 – Sensor fault scenarios

Different fault scenarios have been designed to test the effectiveness of the fault tolerance mechanism. In all the scenarios, the TRMS is asked to follow the angle references (5.3.12):

$$\begin{cases} \theta_h^{ref} = 0.5, \quad \theta_v^{ref} = 0.3 \quad if \quad 0s \le t < 25s \\ \theta_h^{ref} = 0.8, \quad \theta_v^{ref} = 0.5 \quad if \quad 25s \le t < 50s \end{cases} \tag{6.4.1}$$

The faults always occur at time $t = 20s$. Fault scenarios differ in which sensor is faulty and which is its magnitude, as resumed in the following table:

Scenario	1	2	3	4	5	6	7	8
Faulty sensor	ω_h	ω_h	θ_h	θ_h	ω_v	ω_v	θ_v	θ_v
Fault magnitude	0.5	0	0.5	0	0.5	0	0.5	0

Table 22 – Fault scenarios (virtual sensors)

The forgetting factors in (6.3.19) have been chosen as $\varsigma_1(k) = \varsigma_2(k) = \varsigma_3(k) = \varsigma_4(k) = 0.99\,\forall k$. The virtual sensor observers design parameters have been chosen as resumed in ***Table 23***.

6.4.2 – Simulation results

Hereafter results obtained in simulation environment are presented. For each scenario four figures are shown. The first figure shows the fault and its estimation. The second figure shows the state variable that corresponds to the faulty sensor: the real value, the measured value (with fault) and the reconstructed value are compared. The third figure shows the response of the yaw angle both when a fault tolerance mechanism is implemented and when there is not such a mechanism. Analogously, the fourth figure shows the response of the pitch angle with the same comparison.

	ω_h virtual sensor	θ_h virtual sensor	ω_v virtual sensor	θ_v virtual sensor
q	-0.9	-0.95	-0.7	0
r	0.05	0.05	0.3	1
$S_{\min}$	0.9	0.95	0.7	0
$S_{\max}$	0.95	1	1	1

Table 23 – Virtual sensor observers design parameters

To prove the convergence of the virtual sensor observer to the real state, when a fault appears the state of the virtual sensor is set to half the state of the nominal observer. This will not be done when the virtual sensor will be applied to the real TRMS, as the most intelligent choice for the virtual sensor initial condition is to set it at the nominal observer condition at the moment the fault appears.

6.4.2.1 – Fault scenario 1

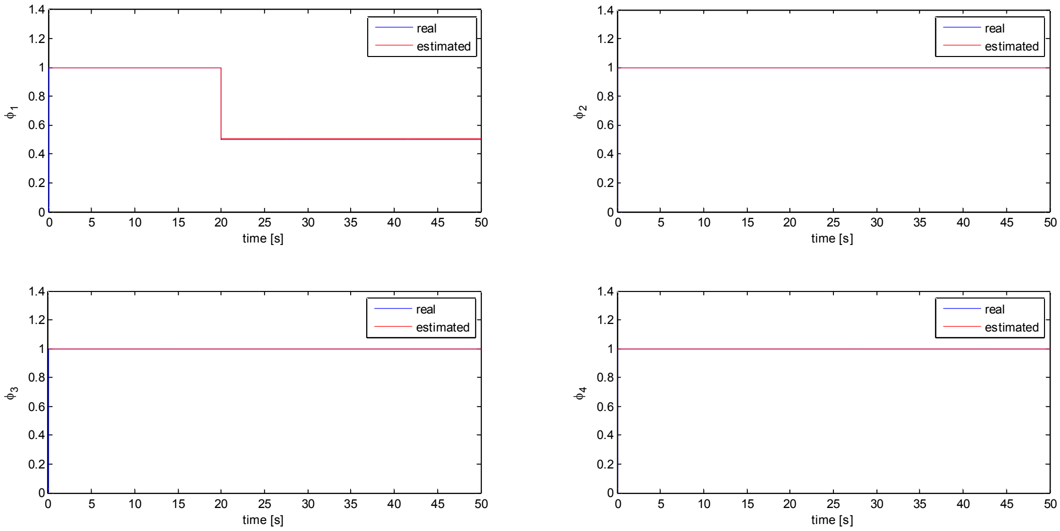

Fig. 66 – Fault scenario 1 (simulation): fault estimation

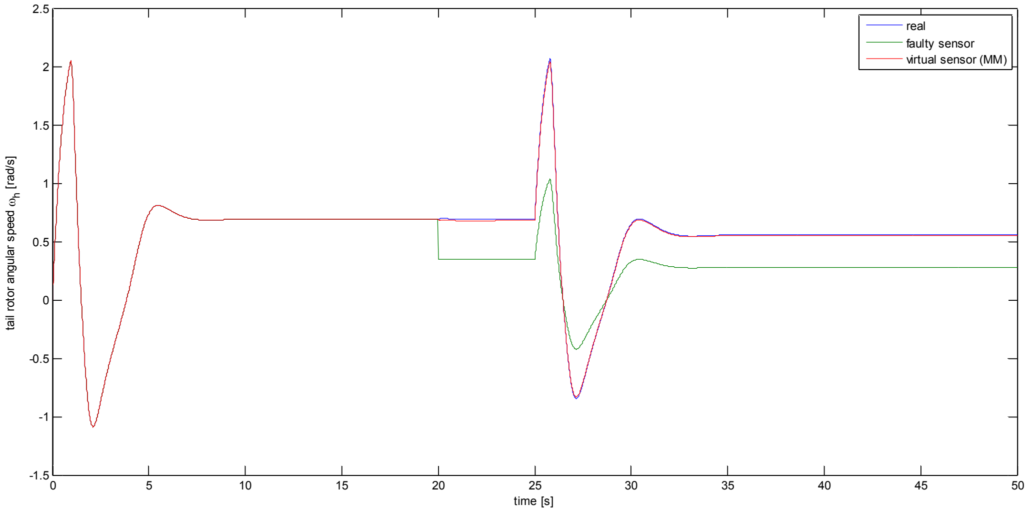

Fig. 67 – Fault scenario 1 (simulation): faulty output reconstruction

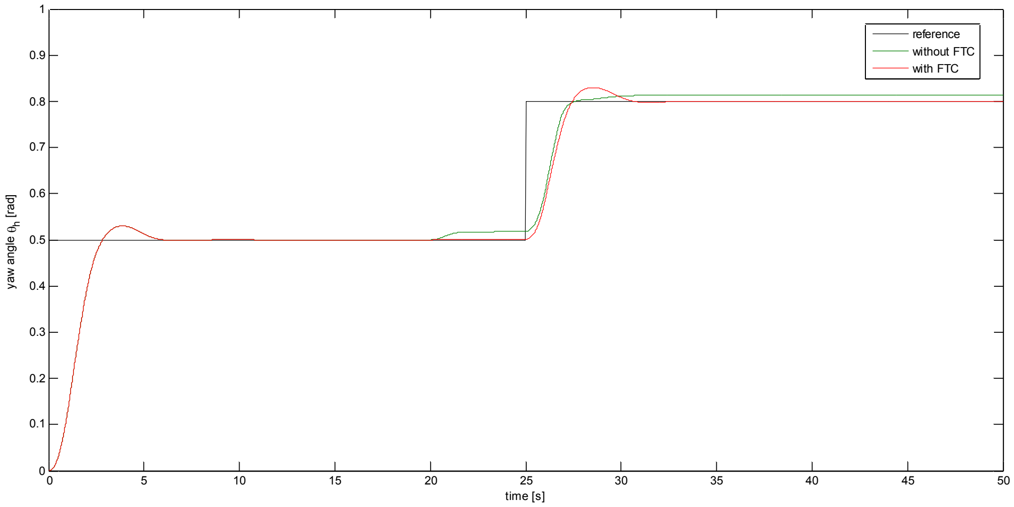

Fig. 68 – Fault scenario 1 (simulation): yaw angle response

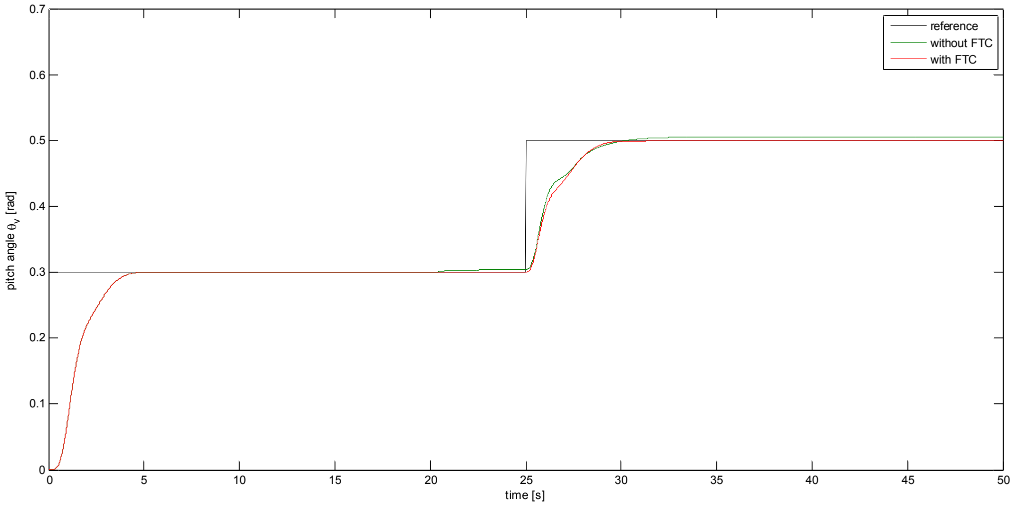

Fig. 69 – Fault scenario 1 (simulation): pitch angle response

6.4.2.2 – Fault scenario 2

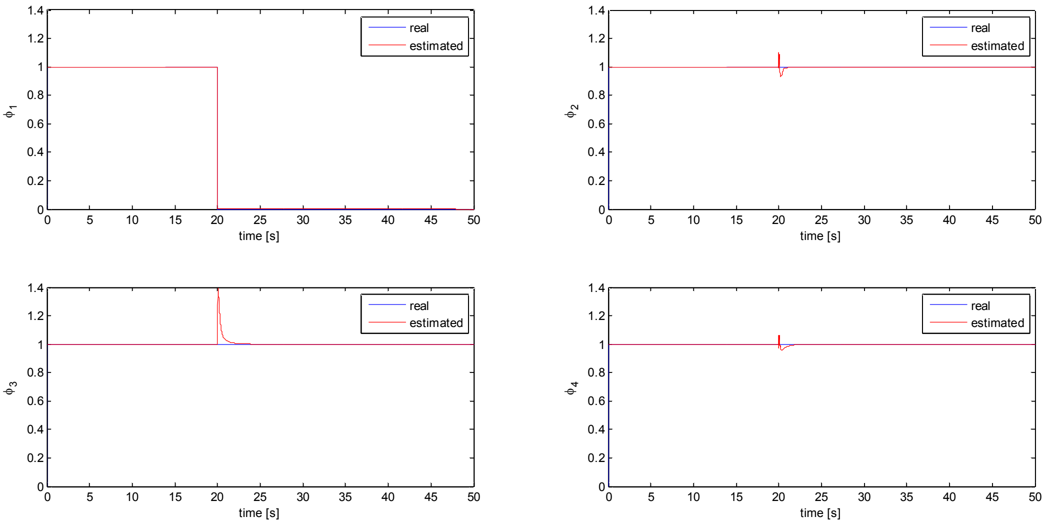

Fig. 70 – Fault scenario 2 (simulation): fault estimation

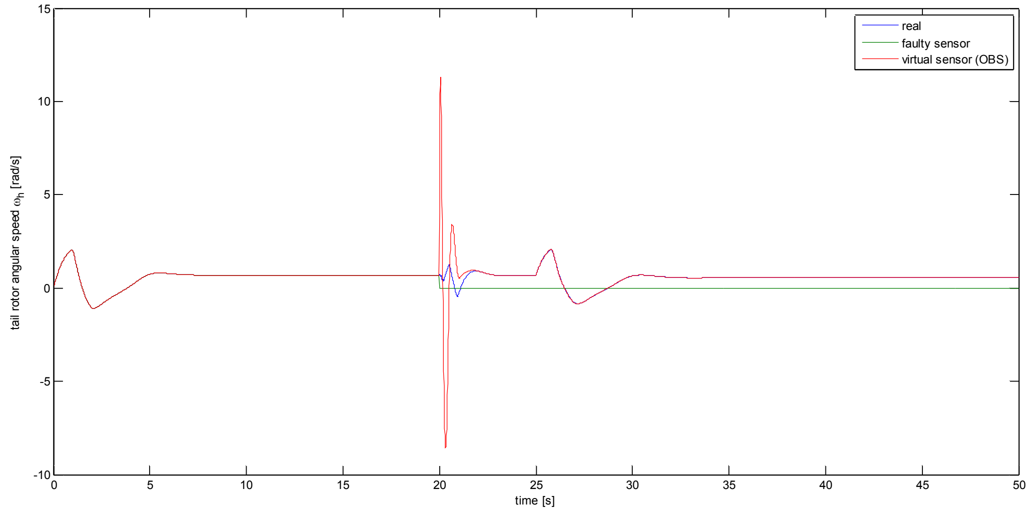

Fig. 71 – Fault scenario 2 (simulation): faulty output reconstruction

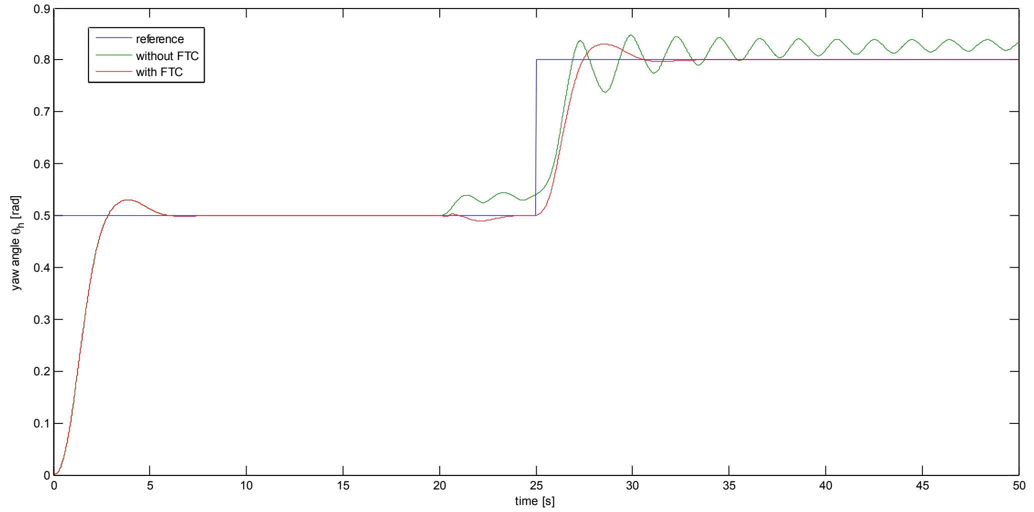

Fig. 72 – Fault scenario 2 (simulation): yaw angle response

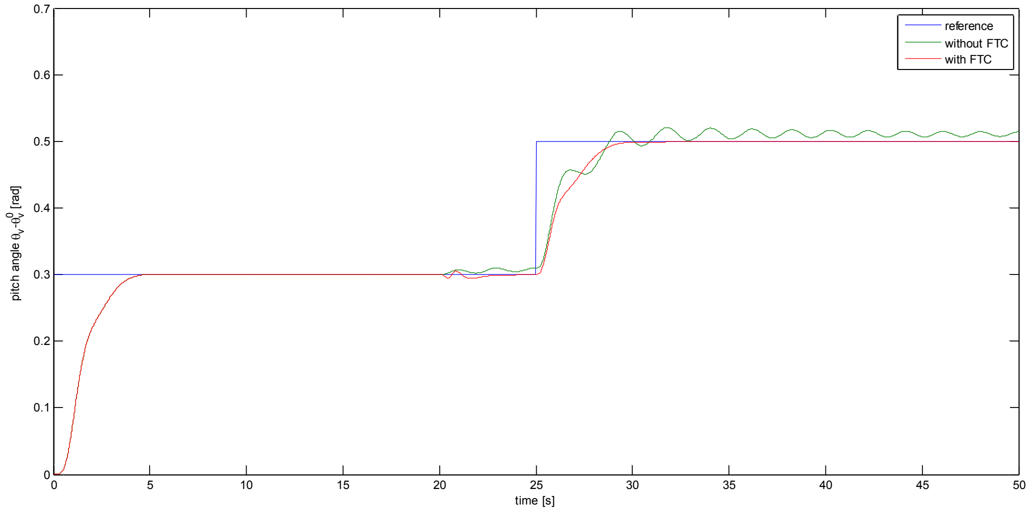

Fig. 73 – Fault scenario 2 (simulation): pitch angle response

6.4.2.3 – Fault scenario 3

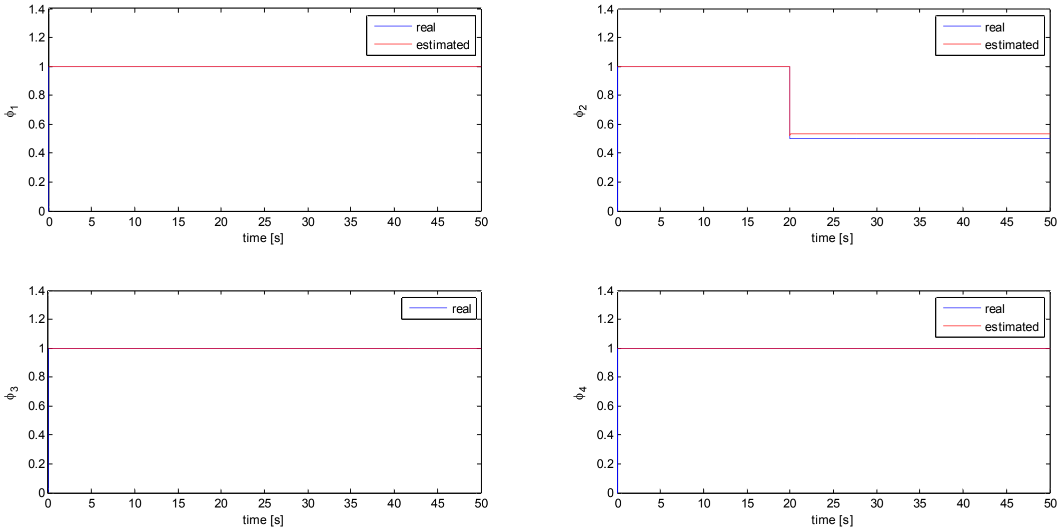

Fig. 74 – Fault scenario 3 (simulation): fault estimation

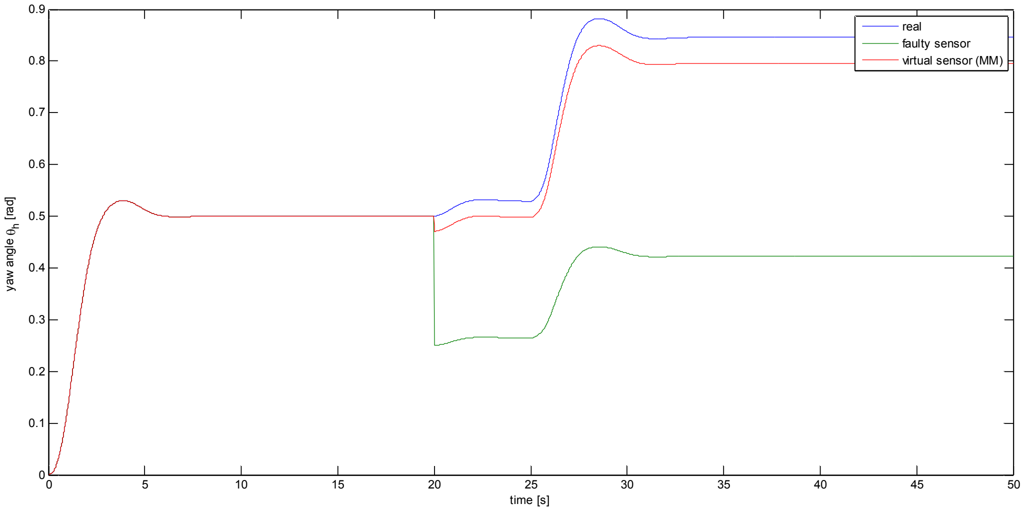

Fig. 75 – Fault scenario 3 (simulation): faulty output reconstruction

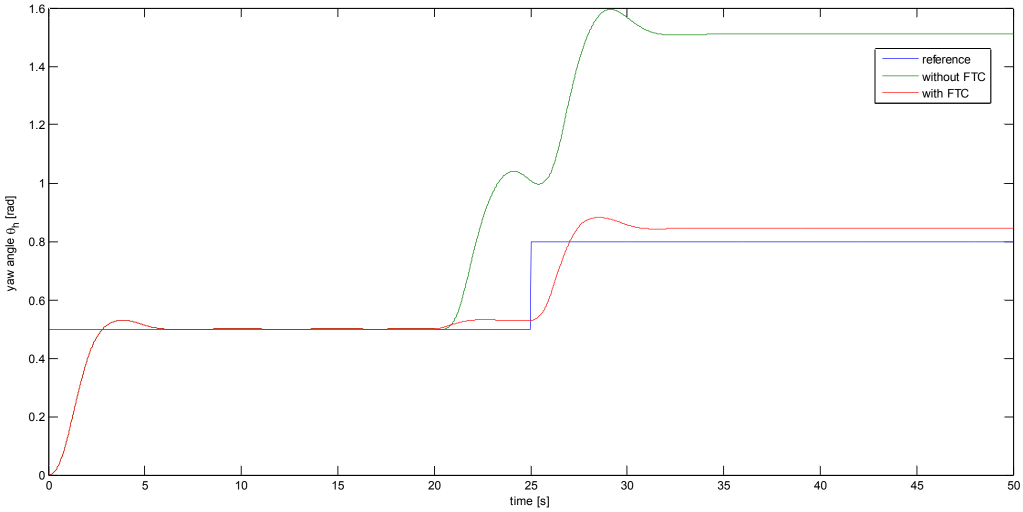

Fig. 76 – Fault scenario 3 (simulation): yaw angle response

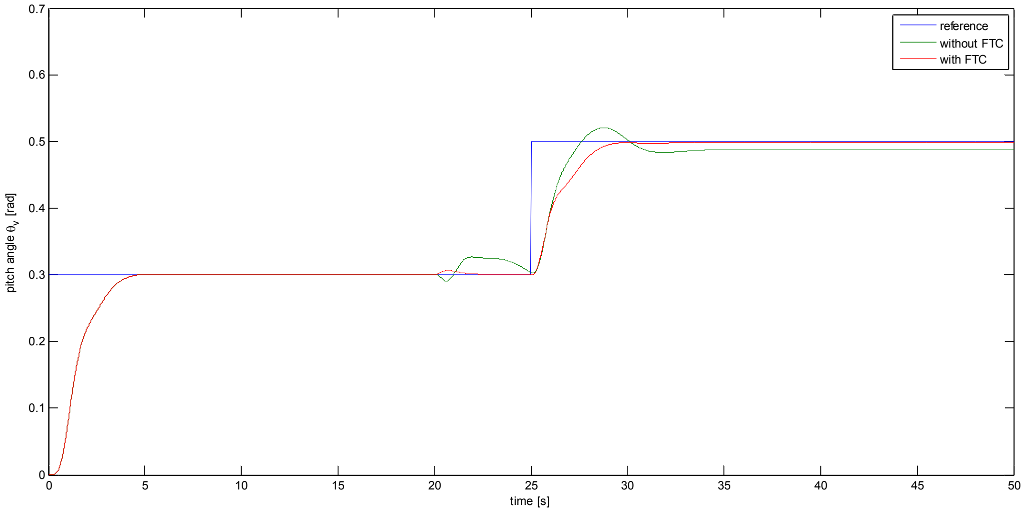

Fig. 77 – Fault scenario 3 (simulation): pitch angle response

6.4.2.4 – Fault scenario 4

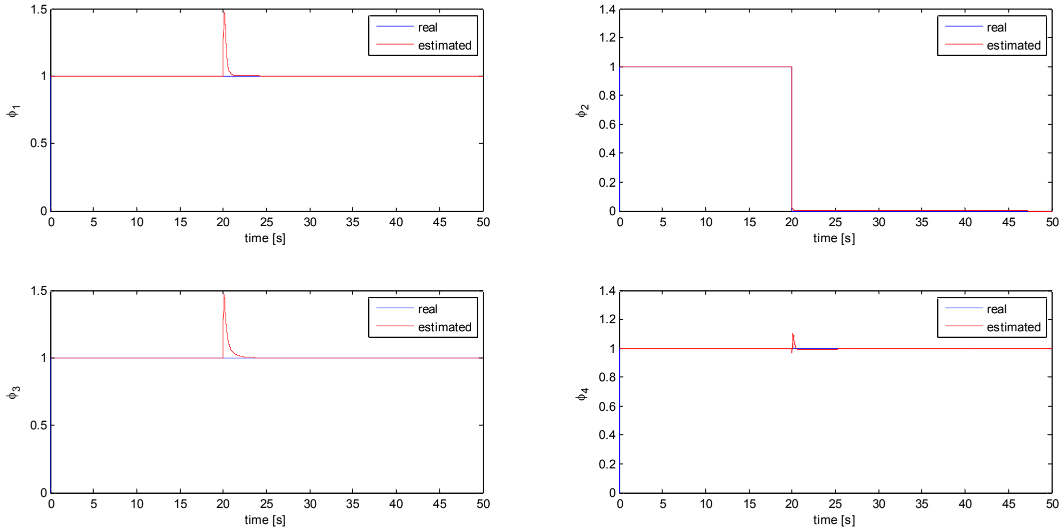

Fig. 78 – Fault scenario 4 (simulation): fault estimation

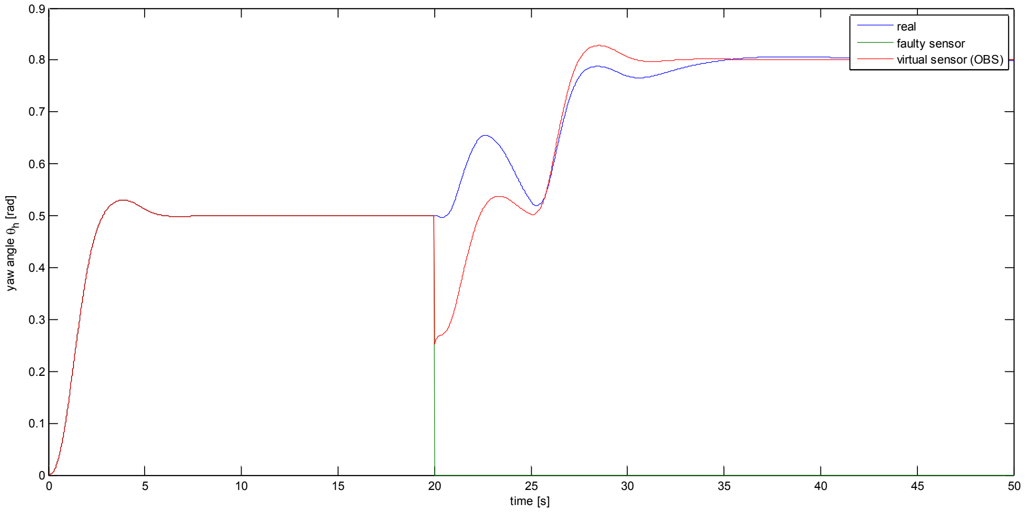

Fig. 79 – Fault scenario 4 (simulation): faulty output reconstruction

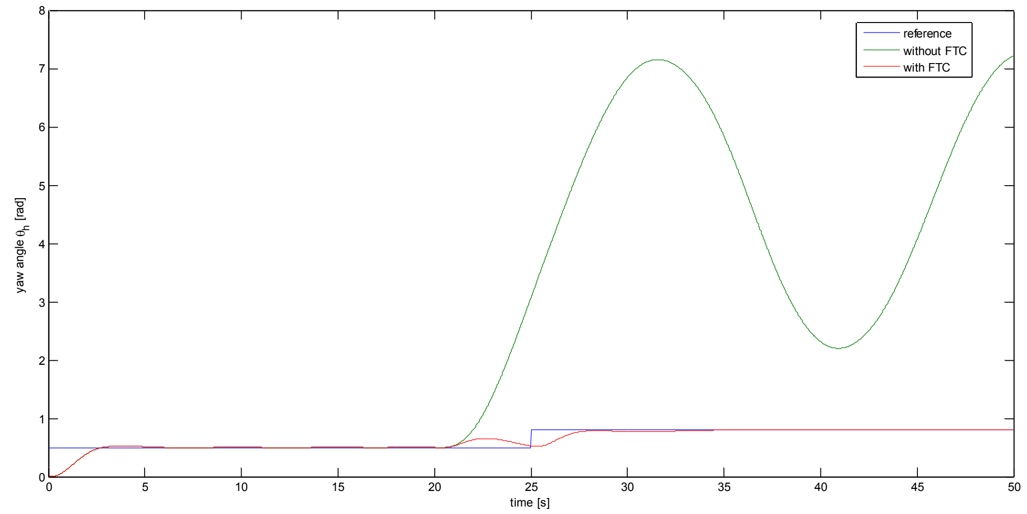

Fig. 80 – Fault scenario 4 (simulation): yaw angle response

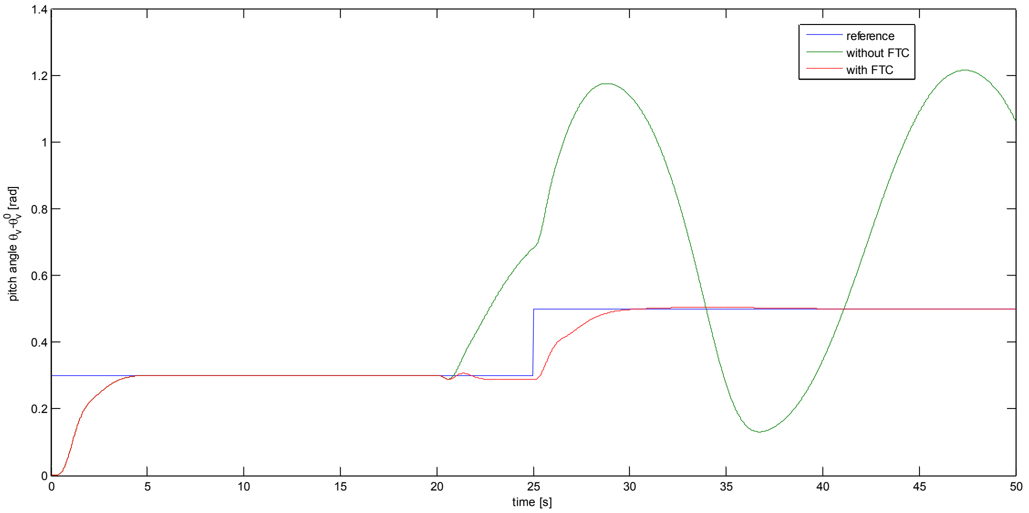

Fig. 81 – Fault scenario 4 (simulation): pitch angle response

6.4.2.5 – Fault scenario 5

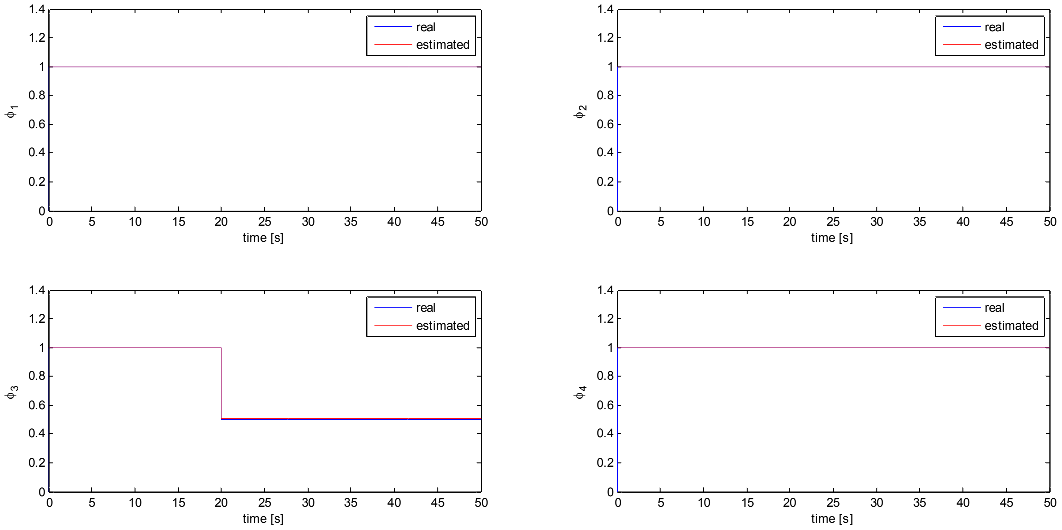

Fig. 82 – Fault scenario 5 (simulation): fault estimation

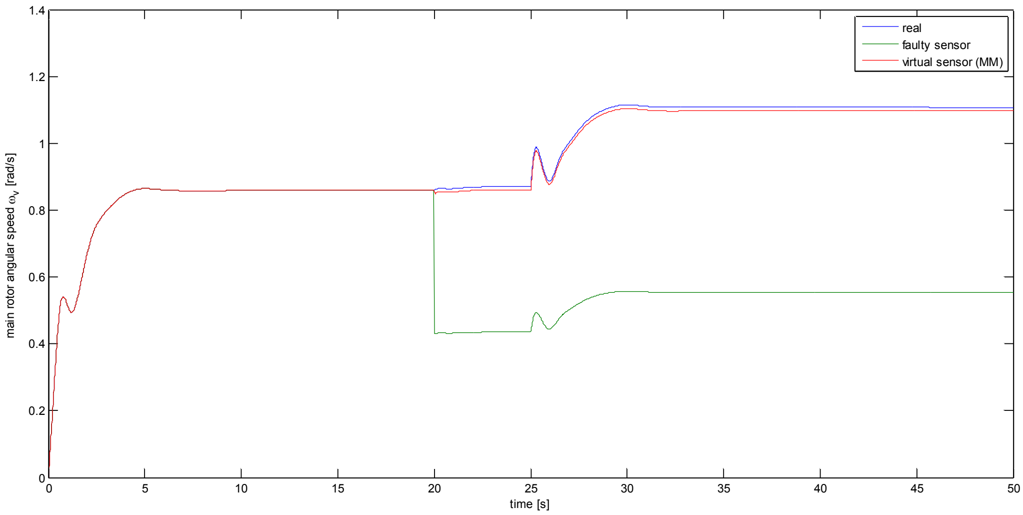

Fig. 83 – Fault scenario 5 (simulation): faulty output reconstruction

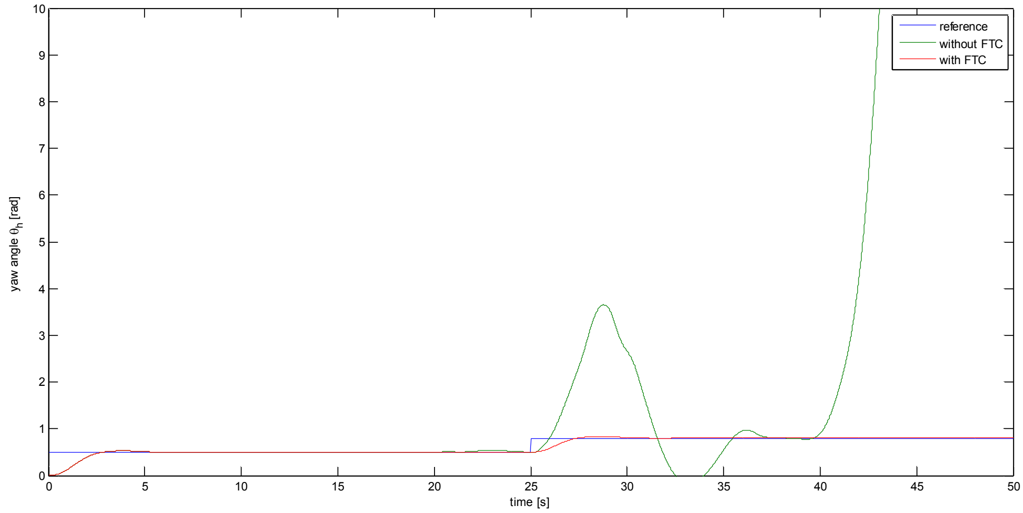

Fig. 84 – Fault scenario 5 (simulation): yaw angle response

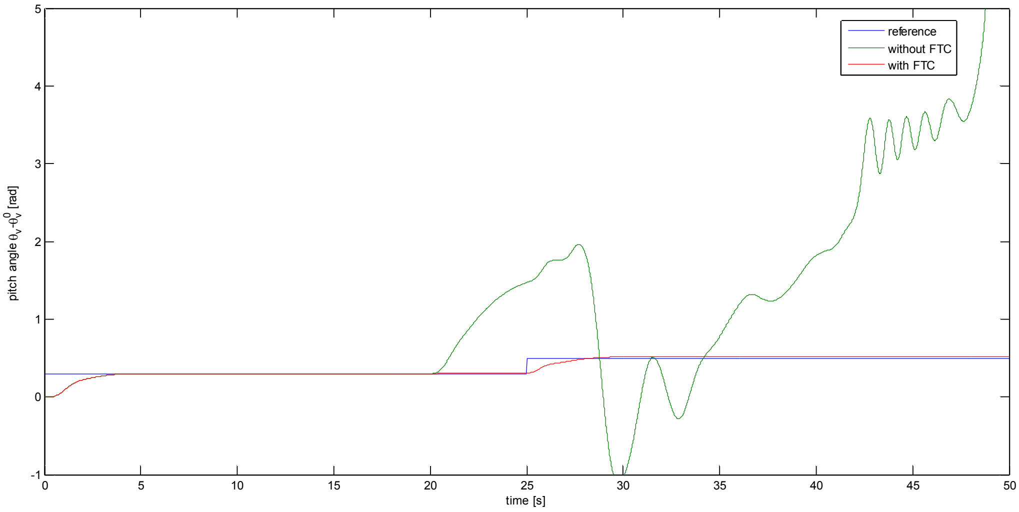

Fig. 85 – Fault scenario 5 (simulation): pitch angle response

6.4.2.6 – Fault scenario 6

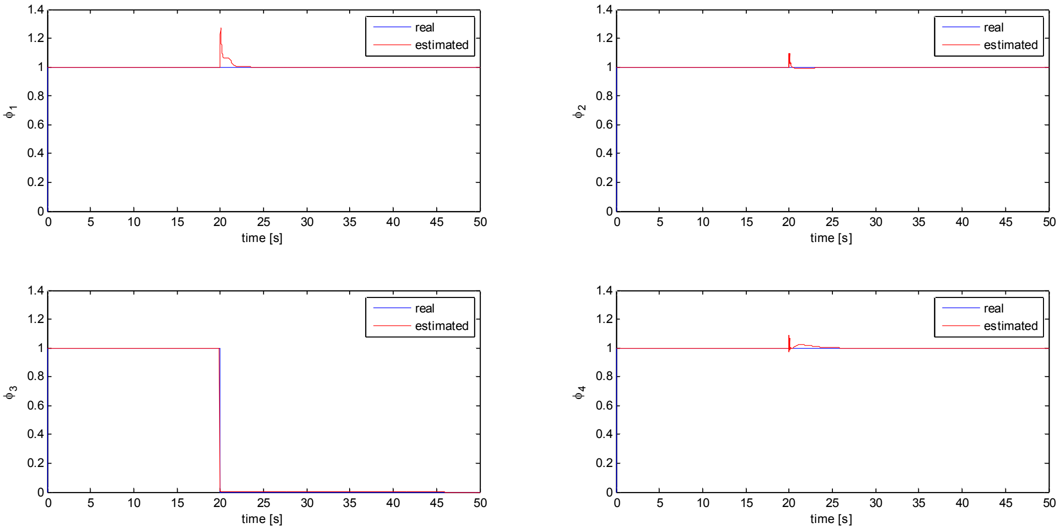

Fig. 86 – Fault scenario 6 (simulation) – Fault estimation

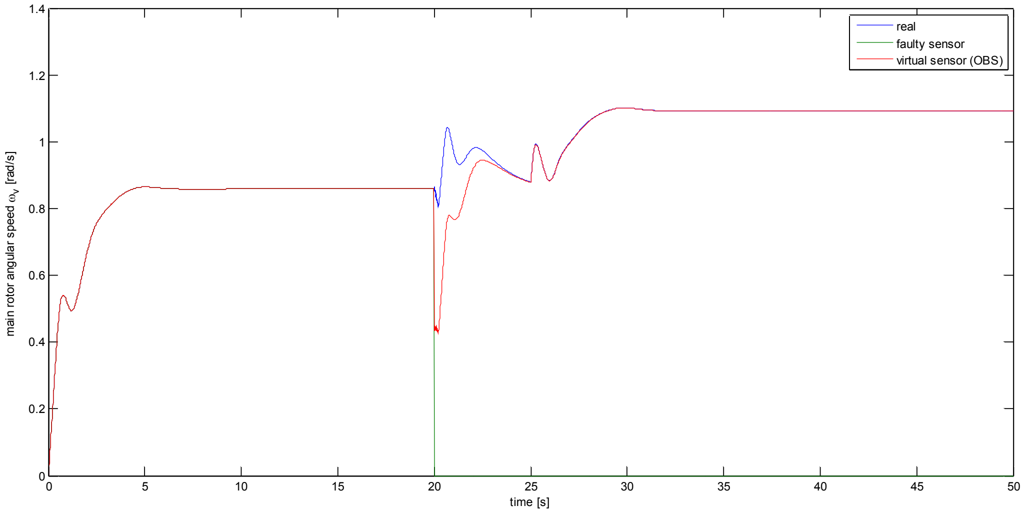

Fig. 87 – Fault scenario 6 (simulation): faulty output reconstruction

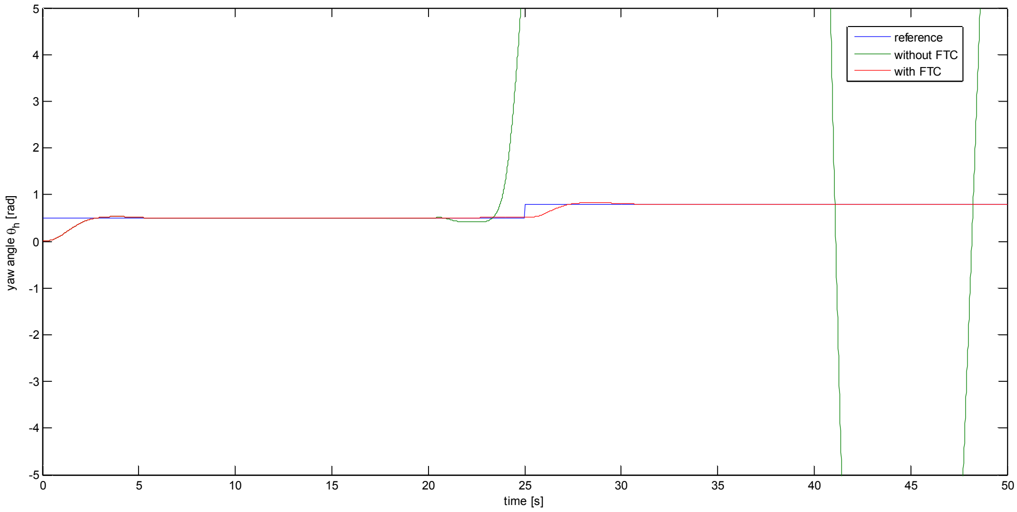

Fig. 88 – Fault scenario 6 (simulation): yaw angle response

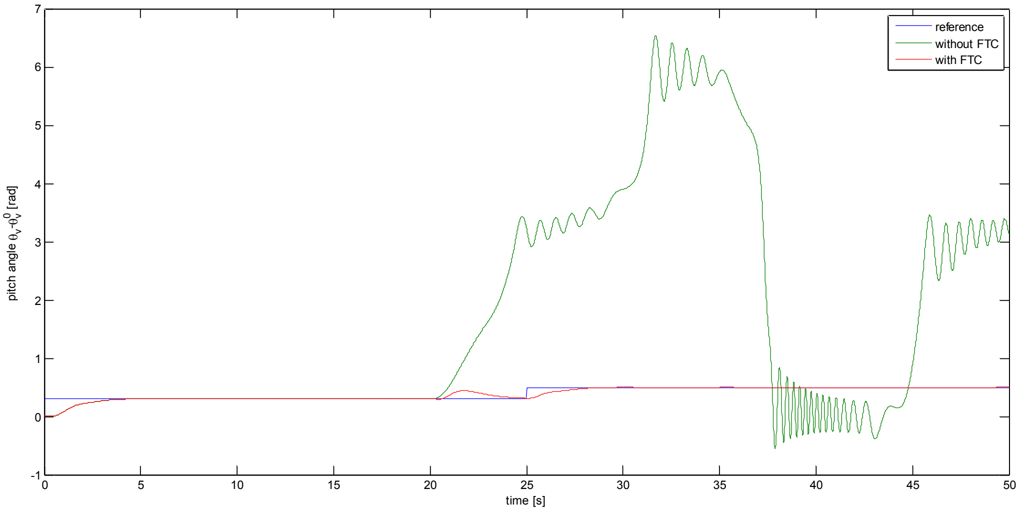

Fig. 89 - Fault scenario 6 (simulation): pitch angle response

6.4.2.7 – Fault scenario 7

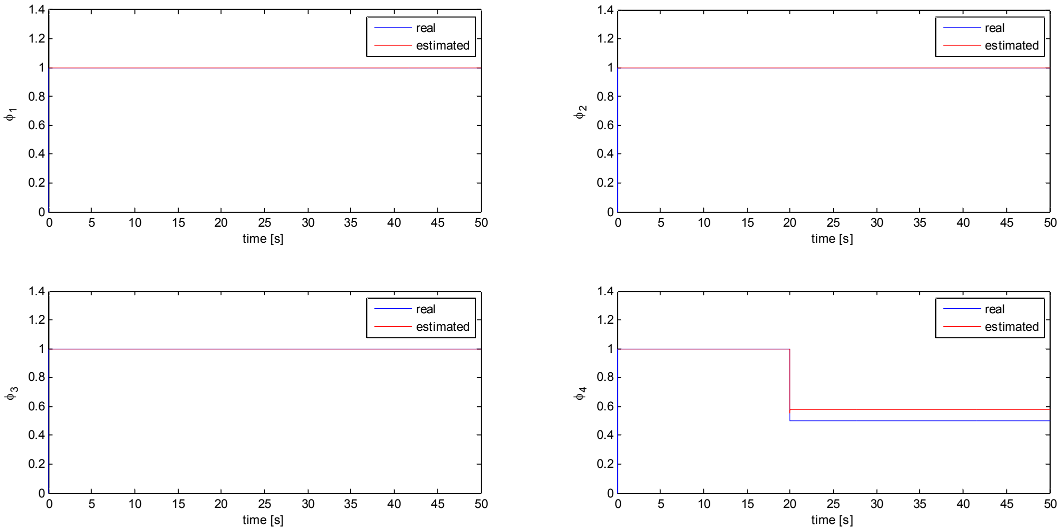

Fig. 90 – Fault scenario 7 (simulation): fault estimation

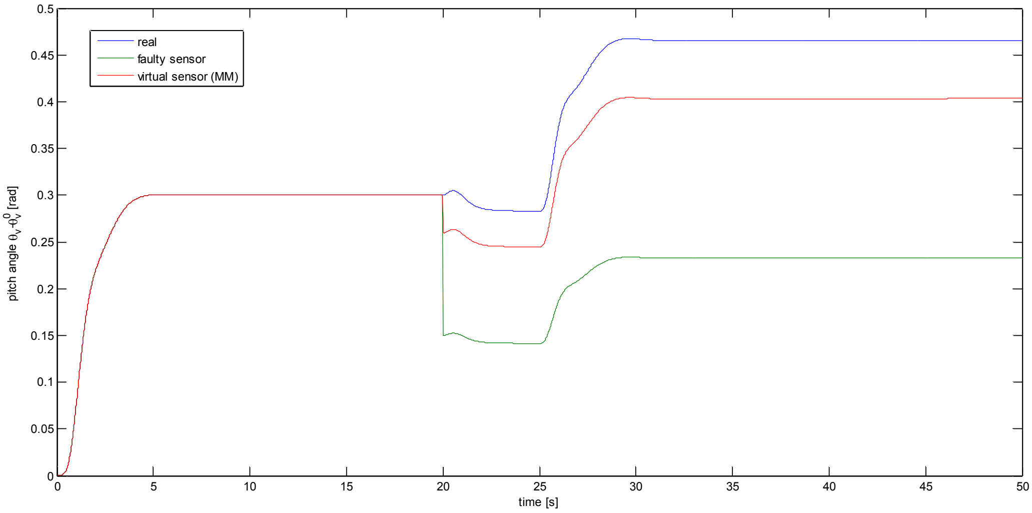

Fig. 91 – Fault scenario 7 (simulation): faulty output reconstruction

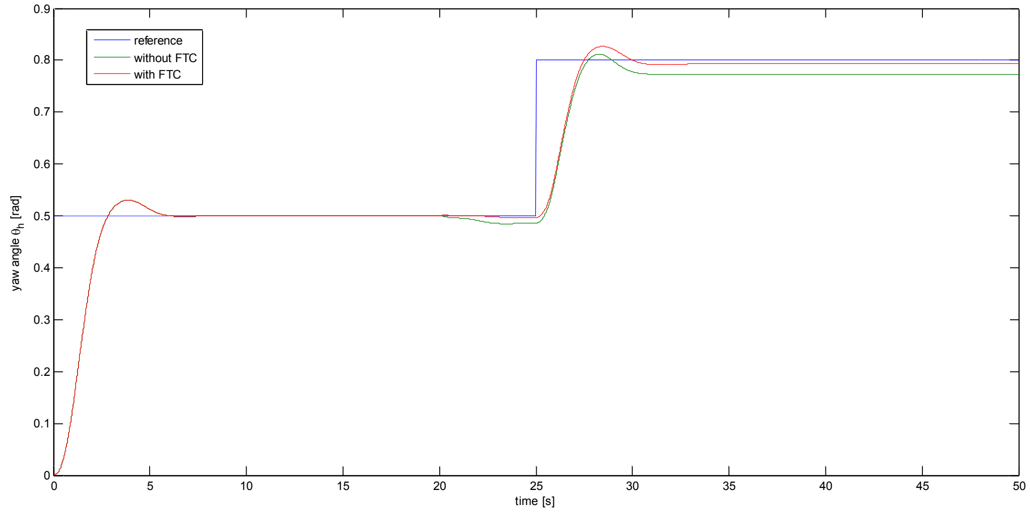

Fig. 92 – Fault scenario 7 (simulation): yaw angle response

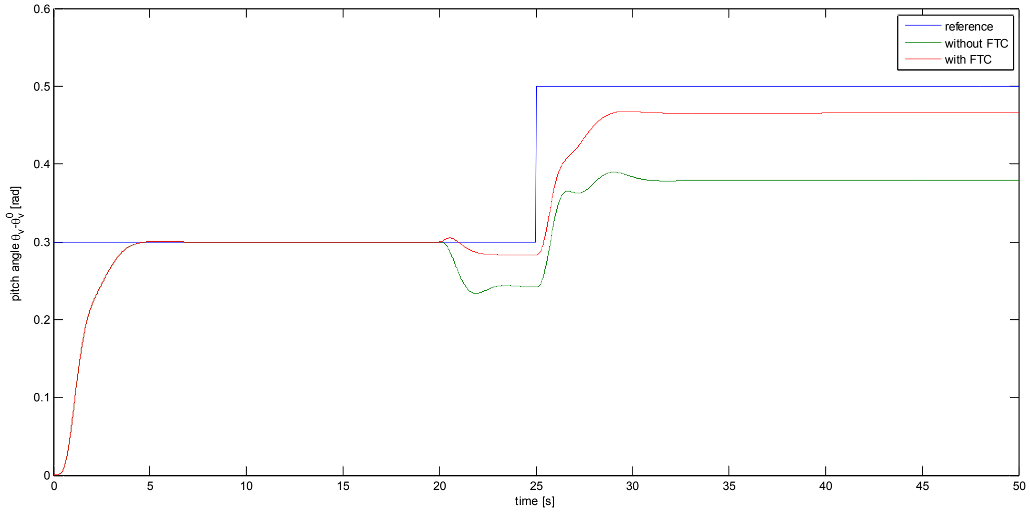

Fig. 93 – Fault scenario 7 (simulation): pitch angle response

6.4.2.8 – Fault scenario 8

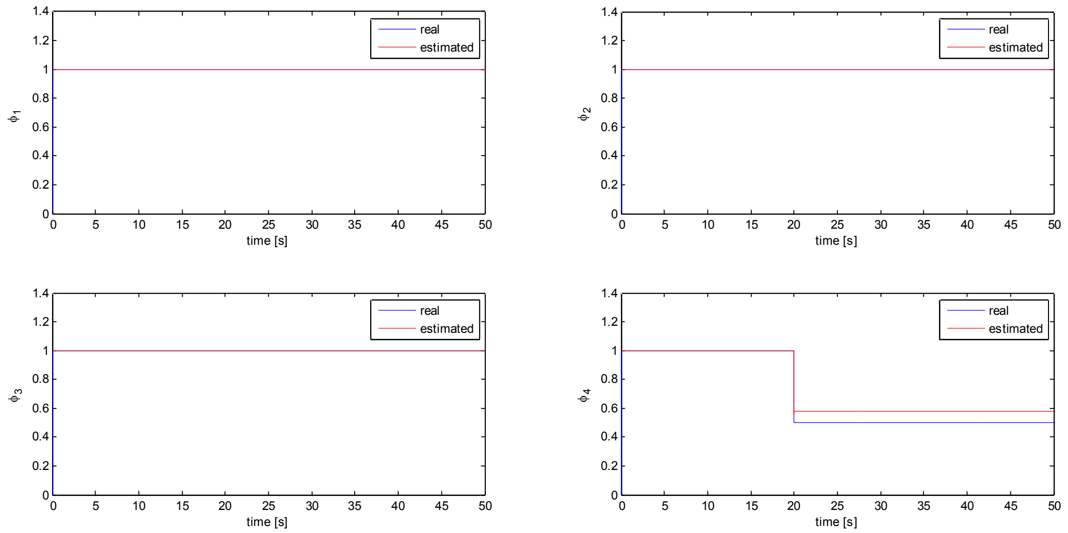

Fig. 94 – Fault scenario 8 (simulation): fault estimation

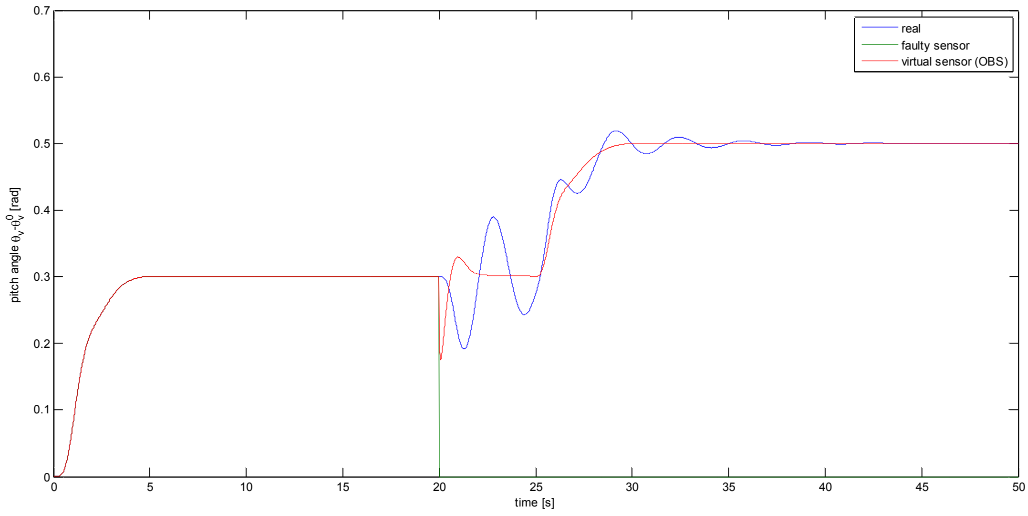

Fig. 95 – Fault scenario 8 (simulation): faulty output reconstruction

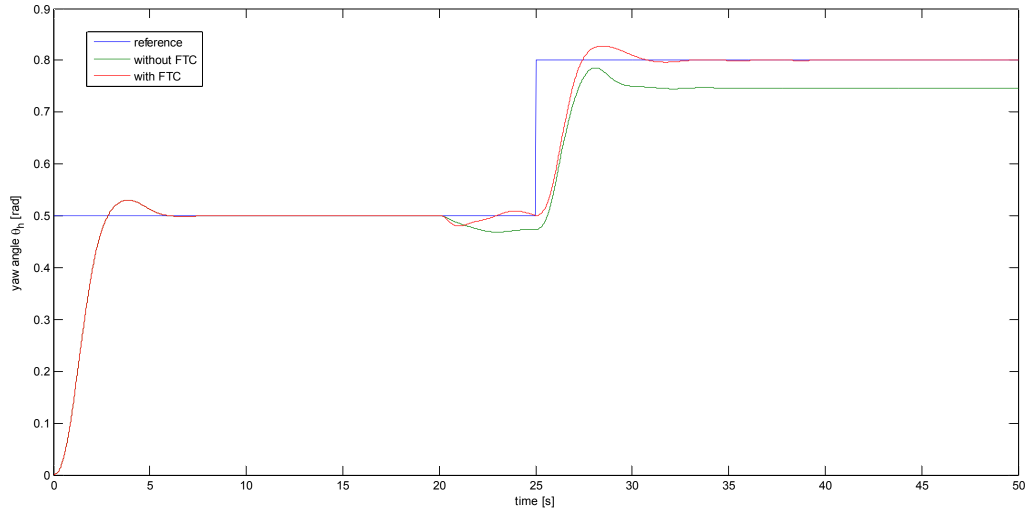

Fig. 96 – Fault scenario 8 (simulation): yaw angle response

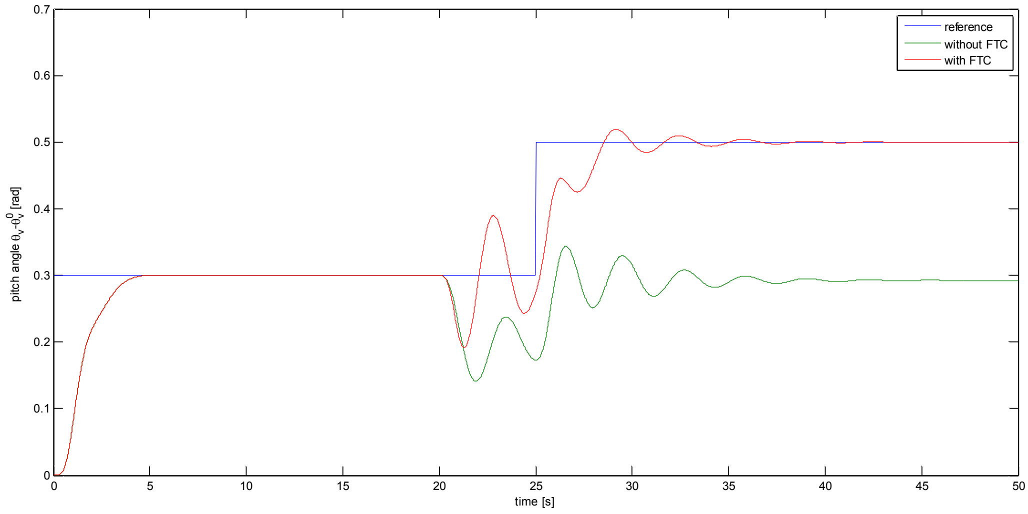

Fig. 97 – Fault scenario 8 (simulation): pitch angle response

6.4.3 – Practical results

In this section, the results obtained with the real TRMS are presented. Analogously to the previous section “Simulation results”, four figures are shown for each scenario. When a fault occurs, the virtual sensor block is activated. It can behave either as a static block (that is the same of Model Matching), or as a dynamic observer. In the second case, the state of the virtual sensor must be initialized as soon as the fault appears. The most intelligent choice is to set its value to the same as the one of the nominal observer.

6.4.3.1 – Fault scenario 1

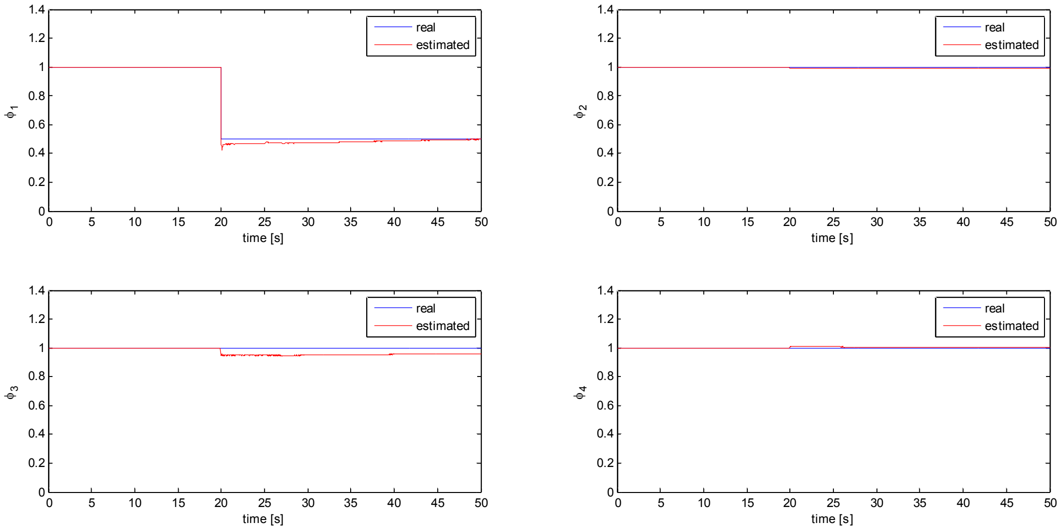

Fig. 98 – Fault scenario 1 (real): fault estimation

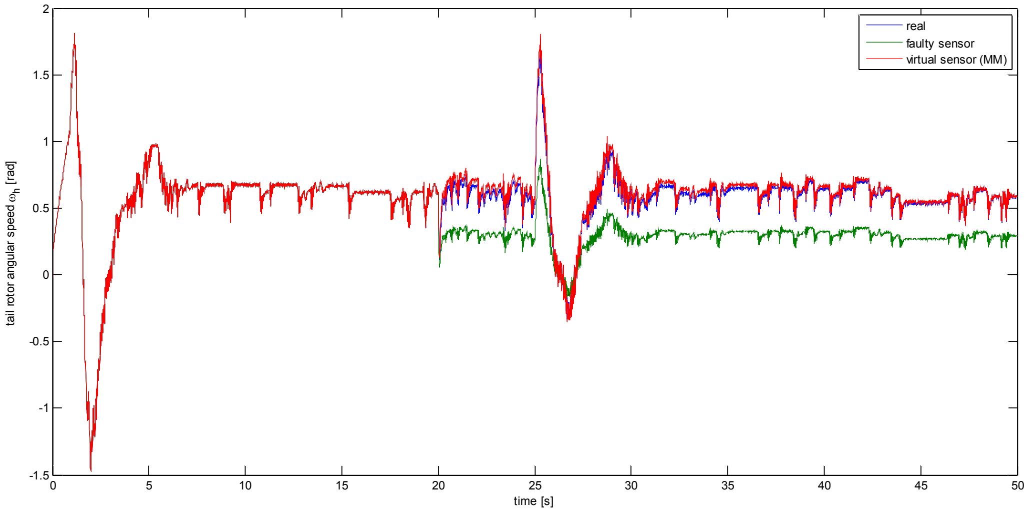

Fig. 99 - Fault scenario 1 (real): faulty output reconstruction

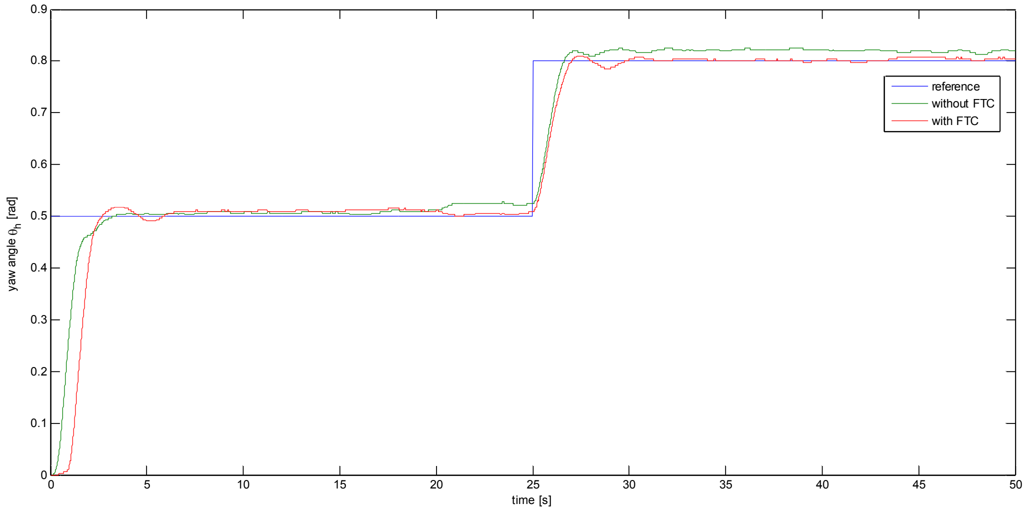

Fig. 100 – Fault scenario 1 (real): yaw angle response

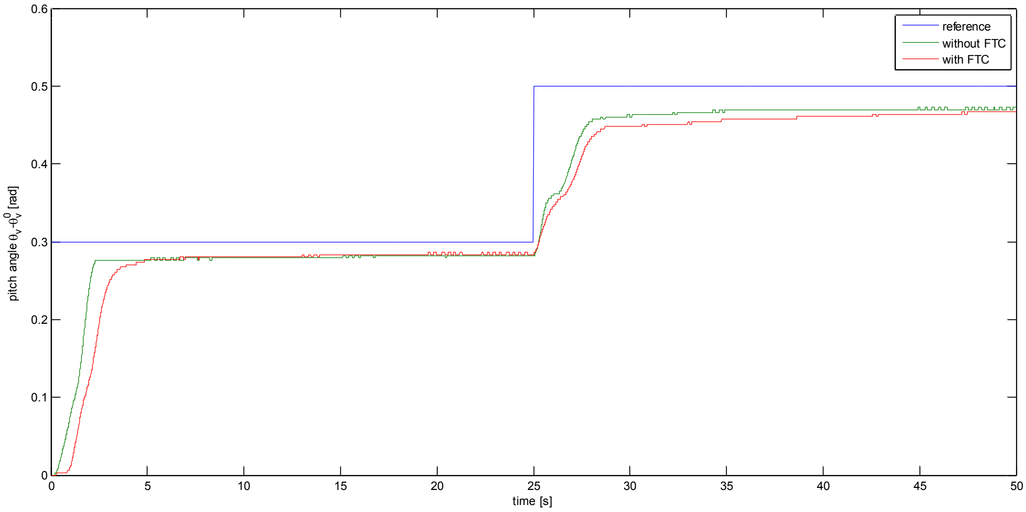

Fig. 101 – Fault scenario 1 (real): pitch angle response

6.4.3.2 – Fault scenario 2

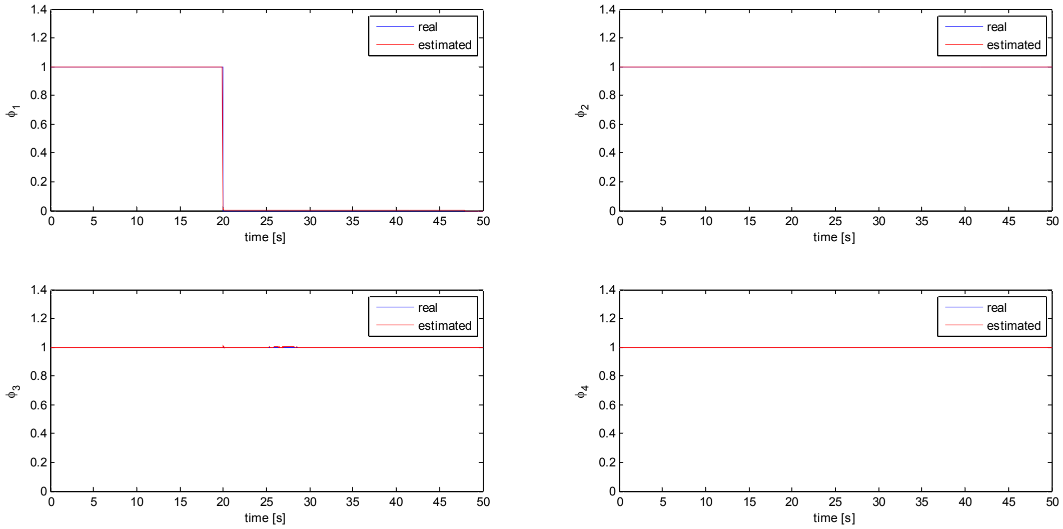

Fig. 102 – Fault scenario 2 (real): fault estimation

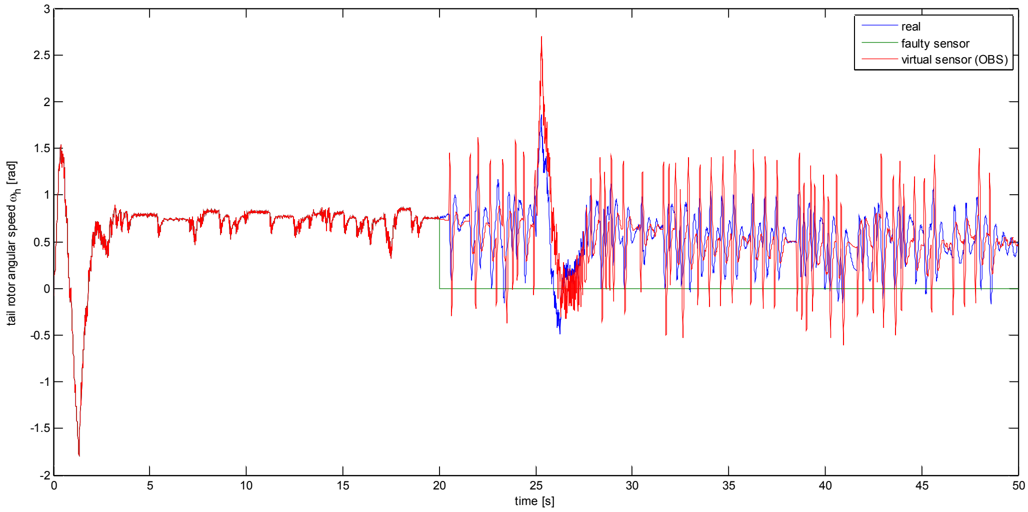

Fig. 103 – Fault scenario 2 (real): faulty output reconstruction

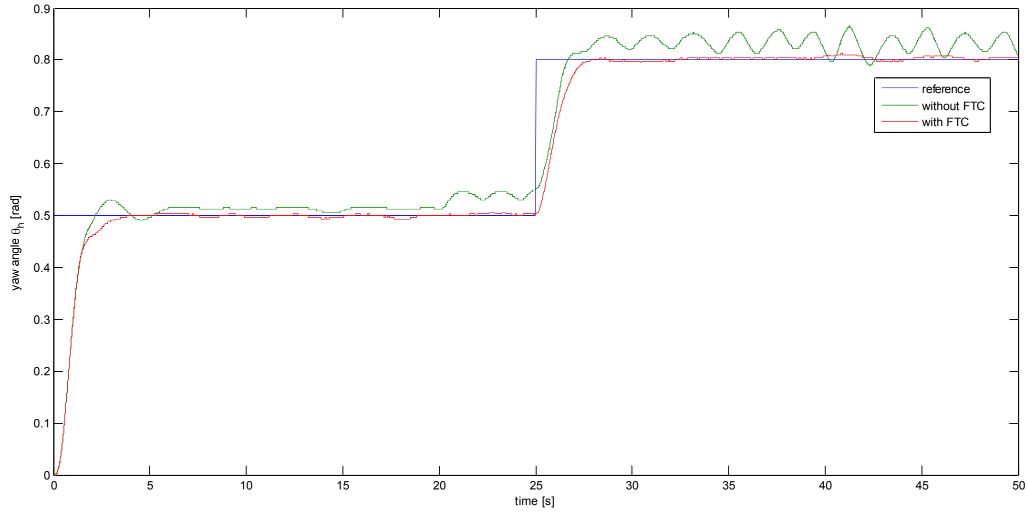

Fig. 104 – Fault scenario 2 (real): yaw angle response

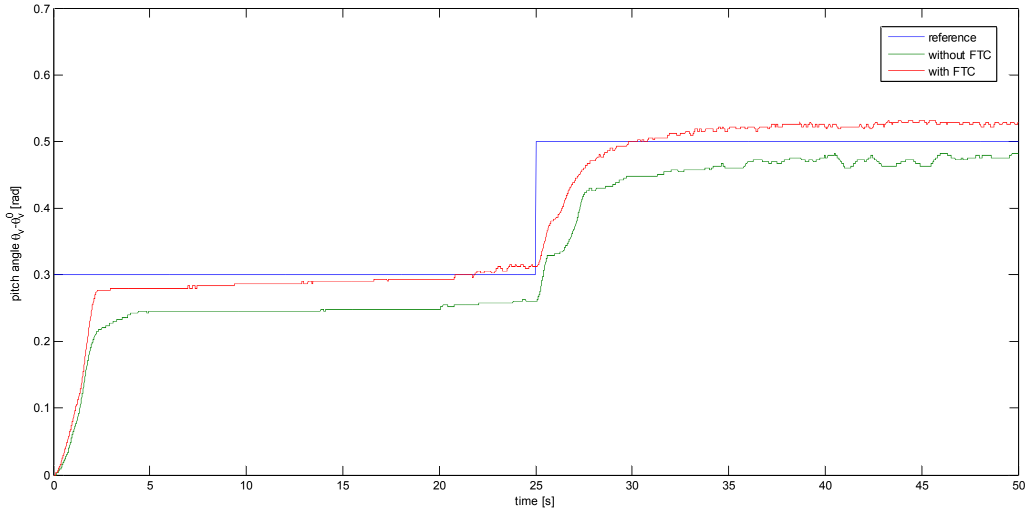

Fig. 105 – Fault scenario 2 (real): pitch angle response

6.4.3.3 – Fault scenario 3

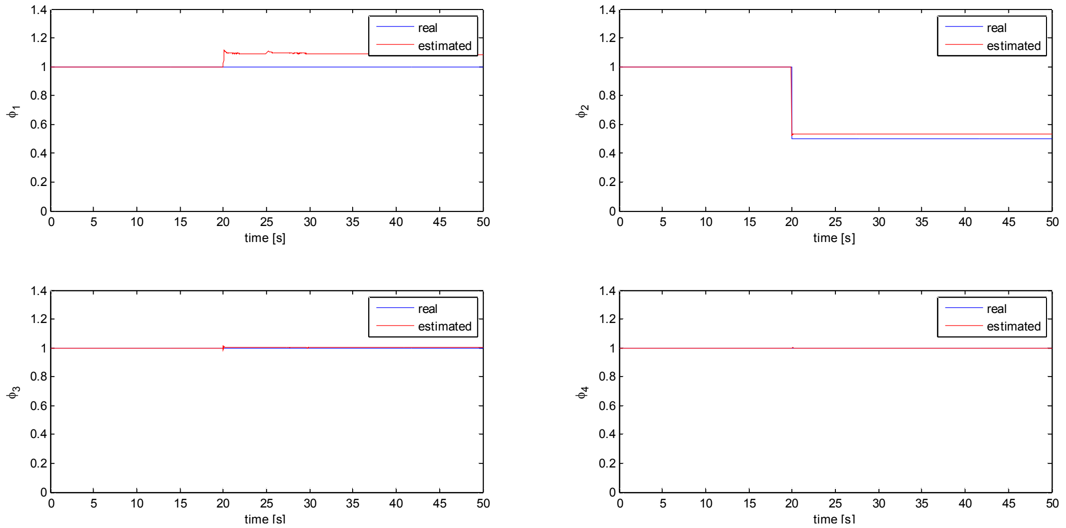

Fig. 106 – Fault scenario 3 (real): fault estimation

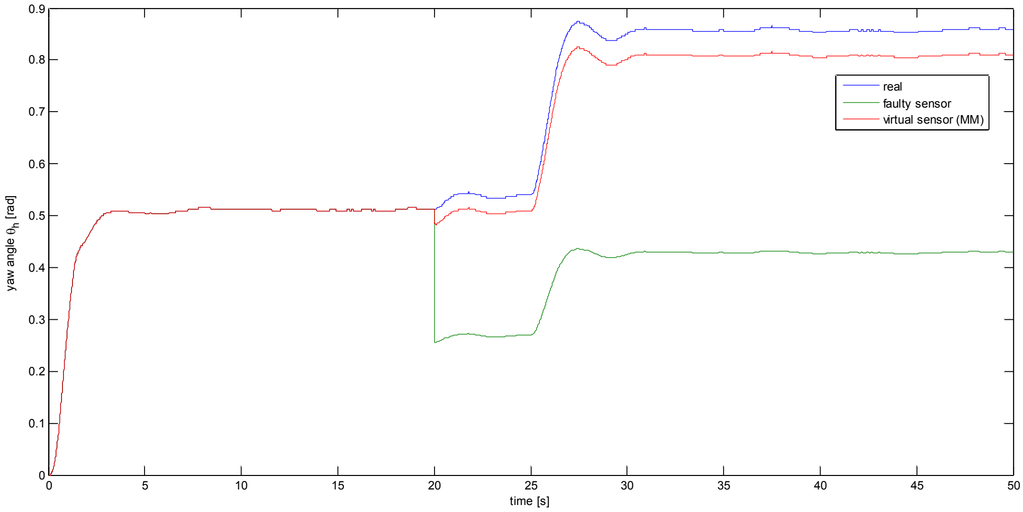

Fig. 107 – Fault scenario 3 (real): faulty output reconstruction

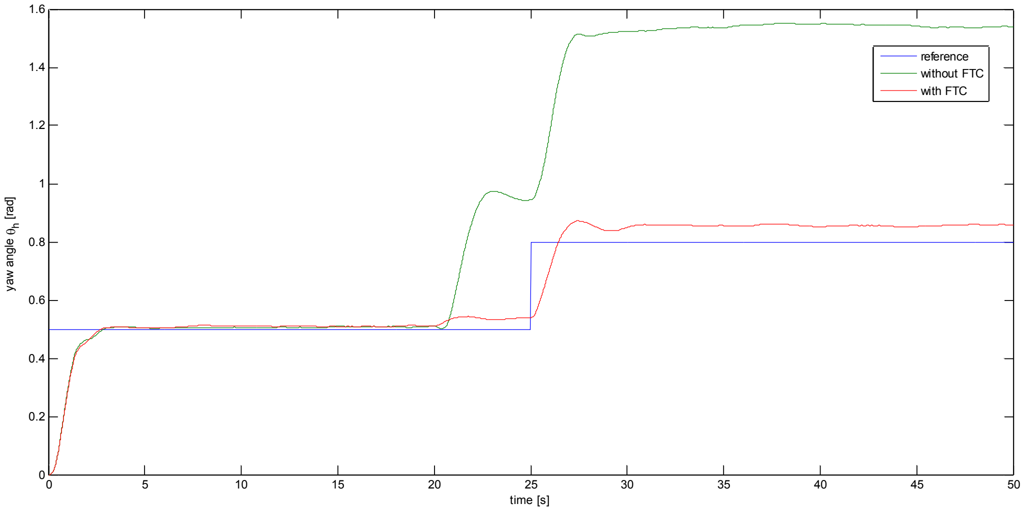

Fig. 108 – Fault scenario 3 (real): yaw angle response

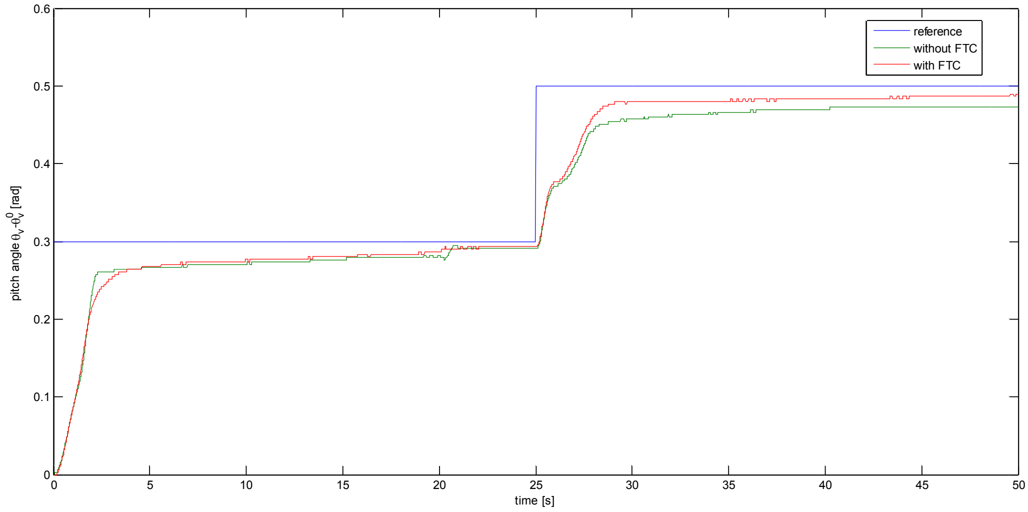

Fig. 109 – Fault scenario 3 (real): pitch angle response

6.4.3.4 – Fault scenario 4

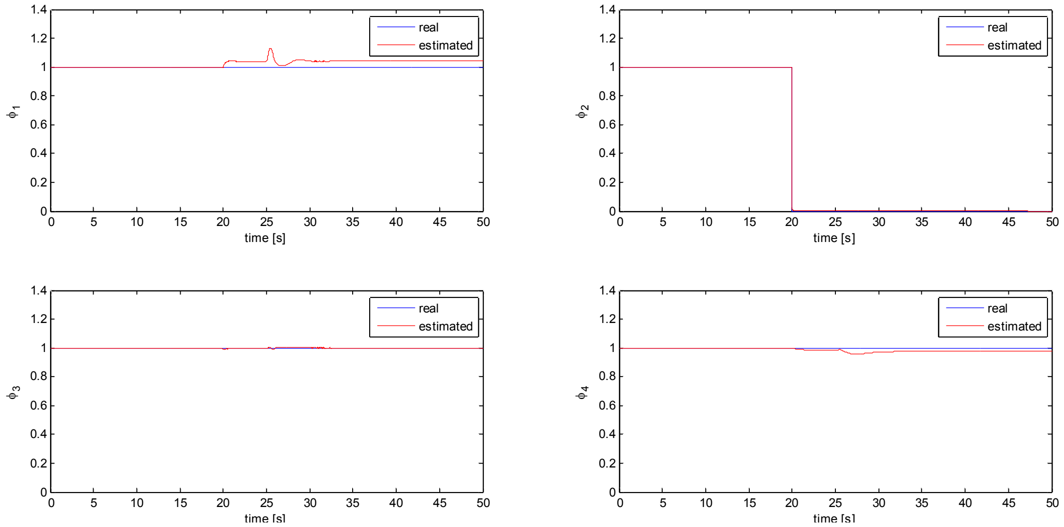

Fig. 110 – Fault scenario 4 (real): fault estimation

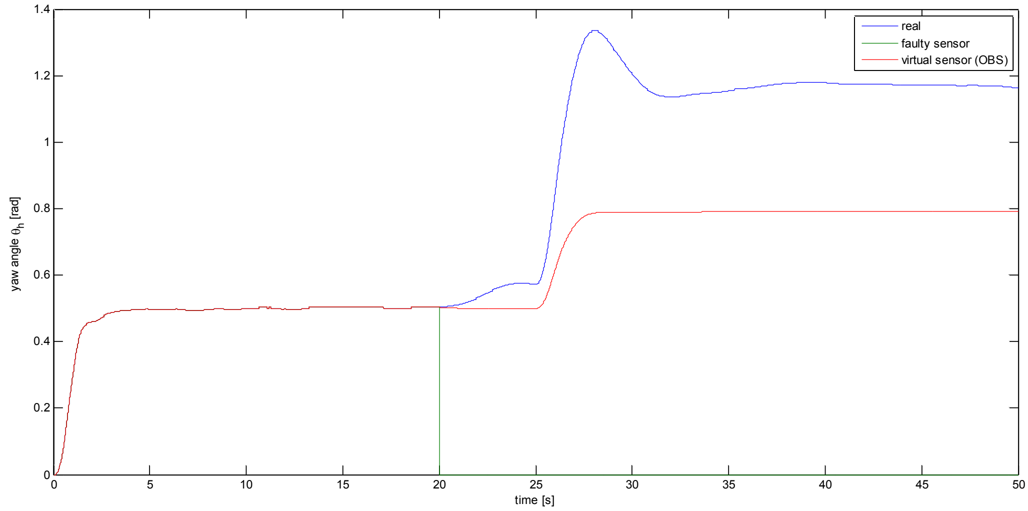

Fig. 111 – Fault scenario 4 (real): faulty output reconstruction

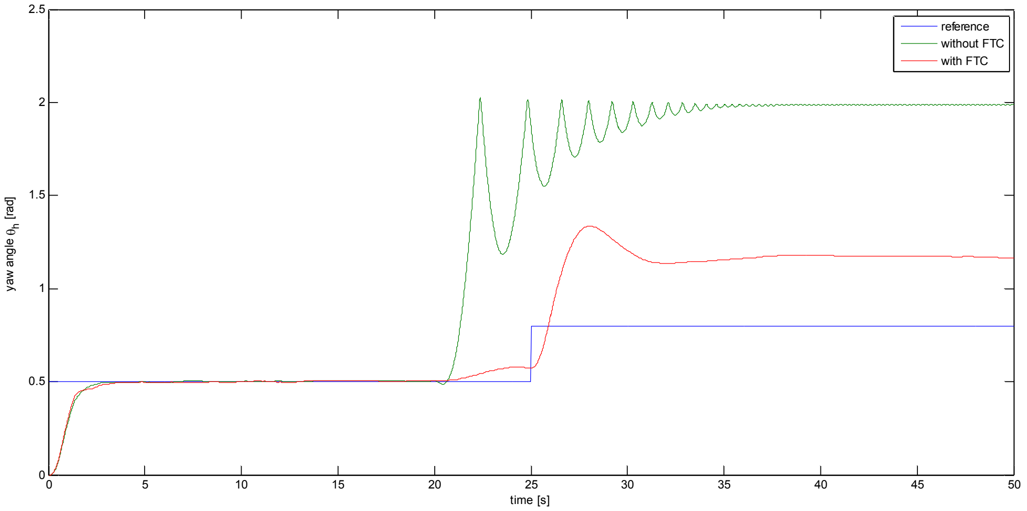

Fig. 112 – Fault scenario 4 (real): yaw angle response

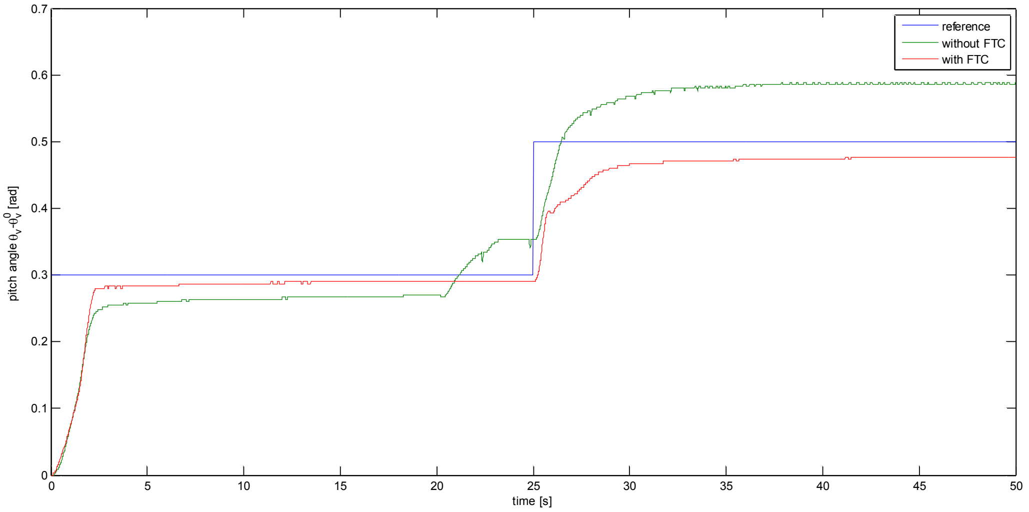

Fig. 113 – Fault scenario 4 (real): pitch angle response

6.4.3.5 – Fault scenario 5

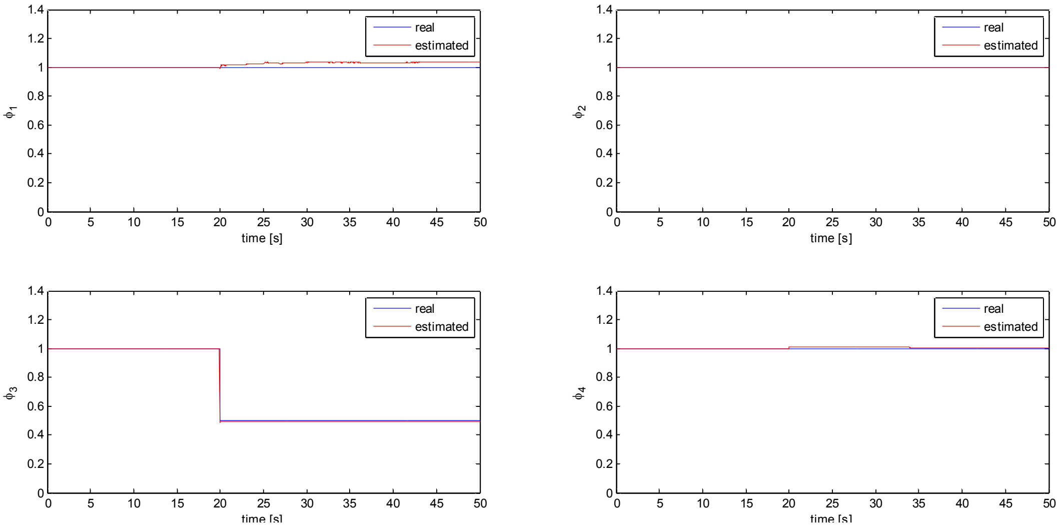

Fig. 114 – Fault scenario 5 (real): fault estimation

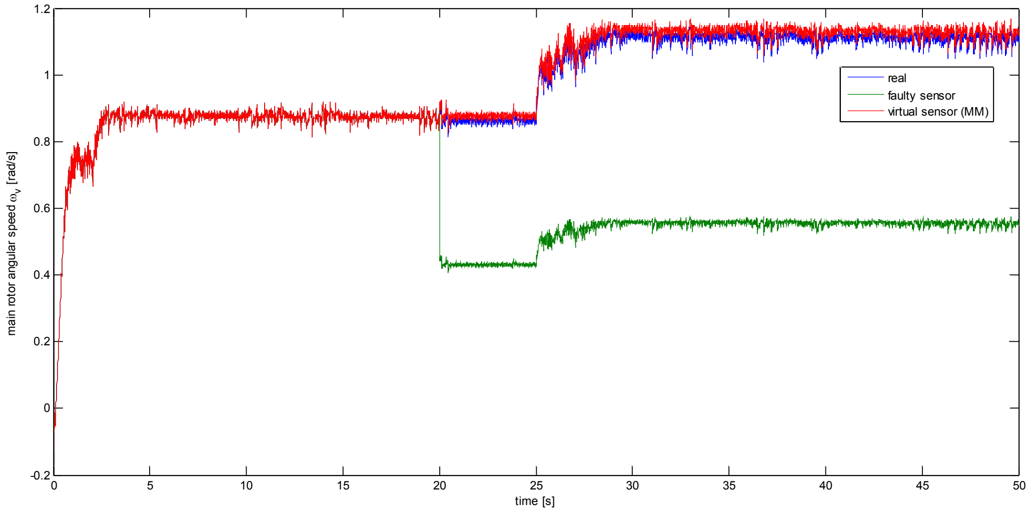

Fig. 115 – Fault scenario 5 (real): faulty output reconstruction

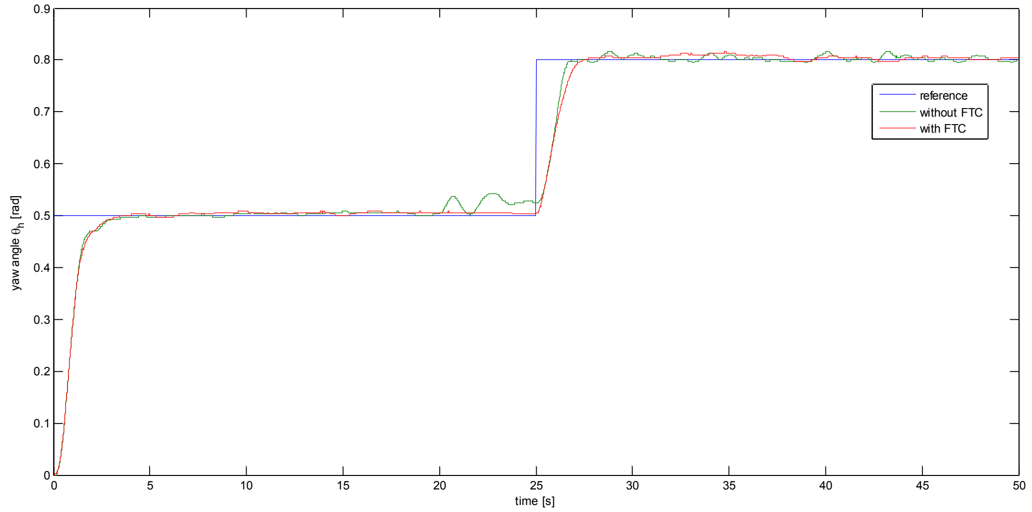

Fig. 116 – Fault scenario 5 (real): yaw angle response

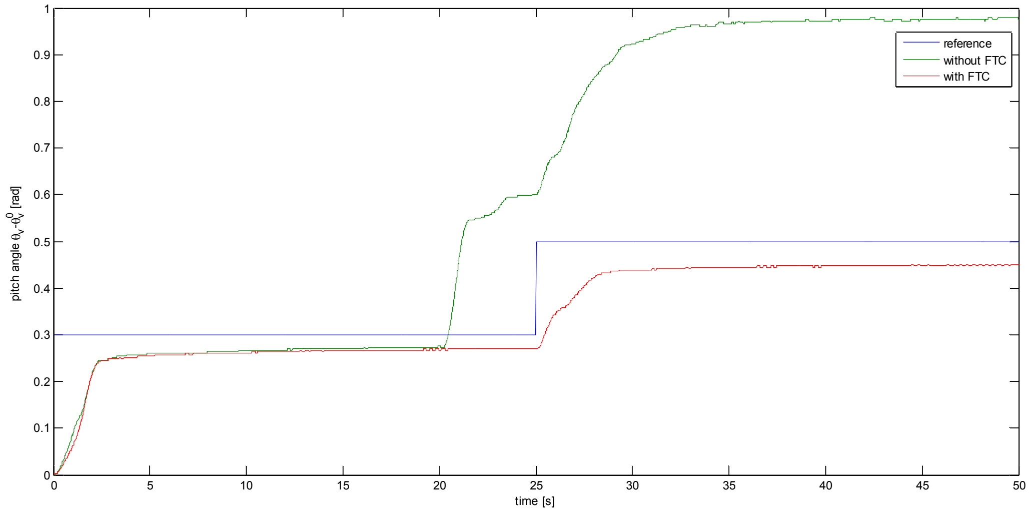

Fig. 117 – Fault scenario 5 (real): pitch angle response

6.4.3.6 – Fault scenario 6

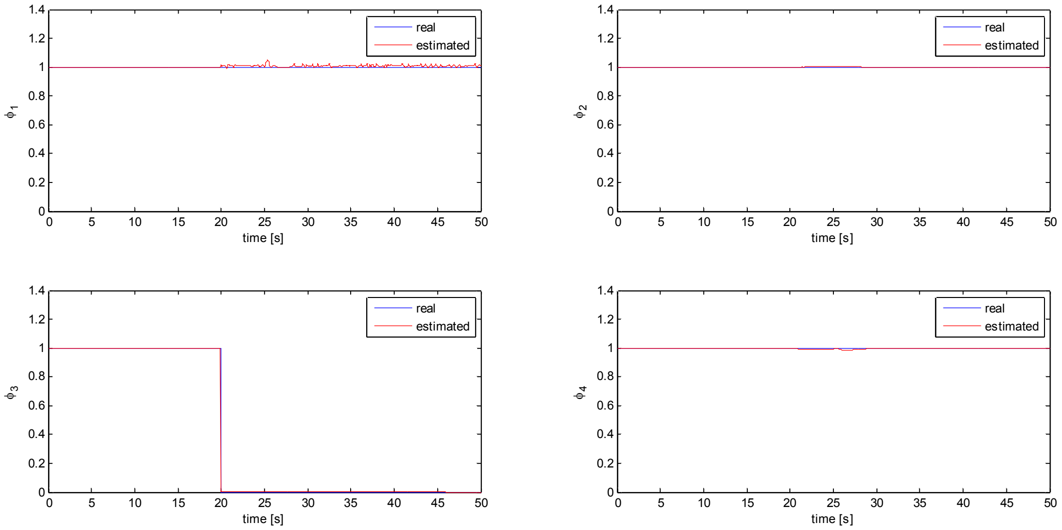

Fig. 118 – Fault scenario 6 (real): fault estimation

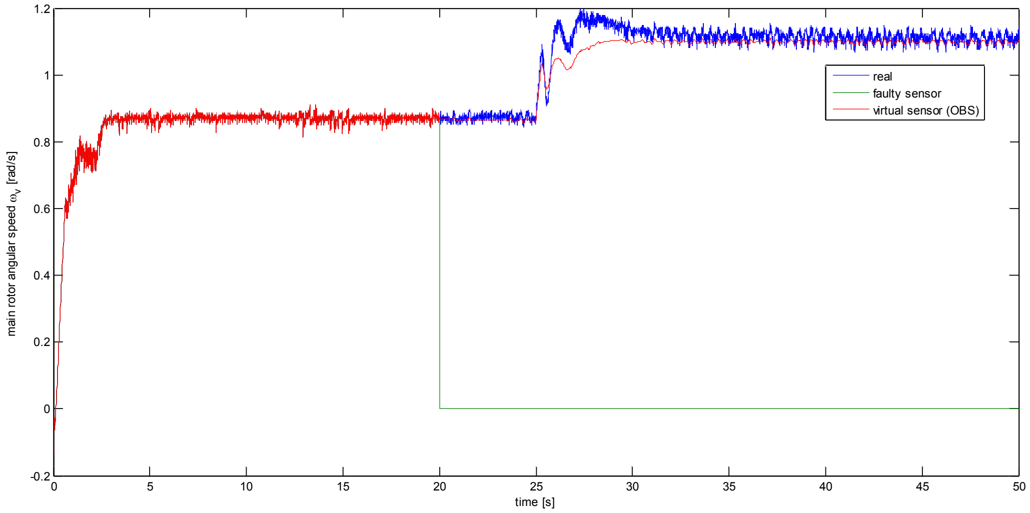

Fig. 119 – Fault scenario 6 (real): faulty output reconstruction

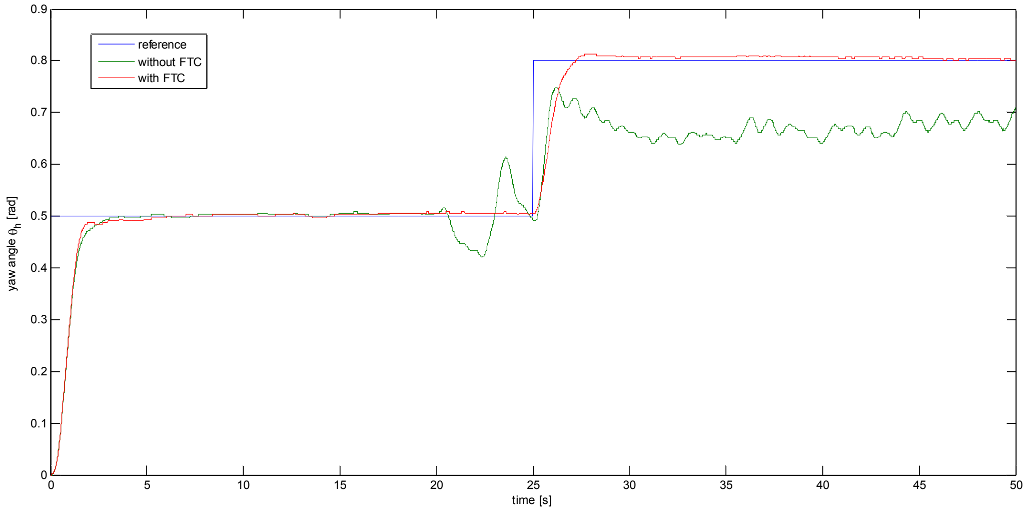

Fig. 120 – Fault scenario 6 (real): yaw angle response

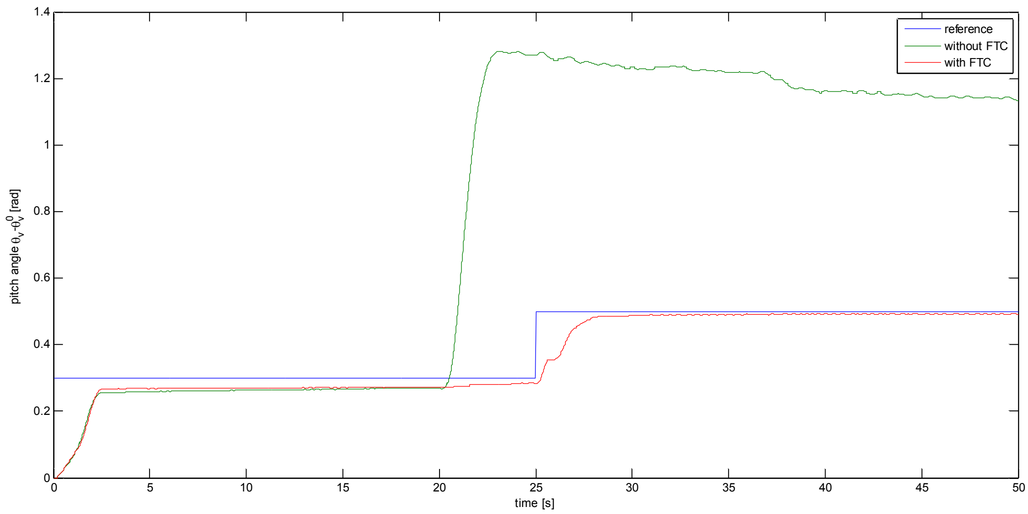

Fig. 121 – Fault scenario 6 (real): pitch angle response

6.4.3.7 – Fault scenario 7

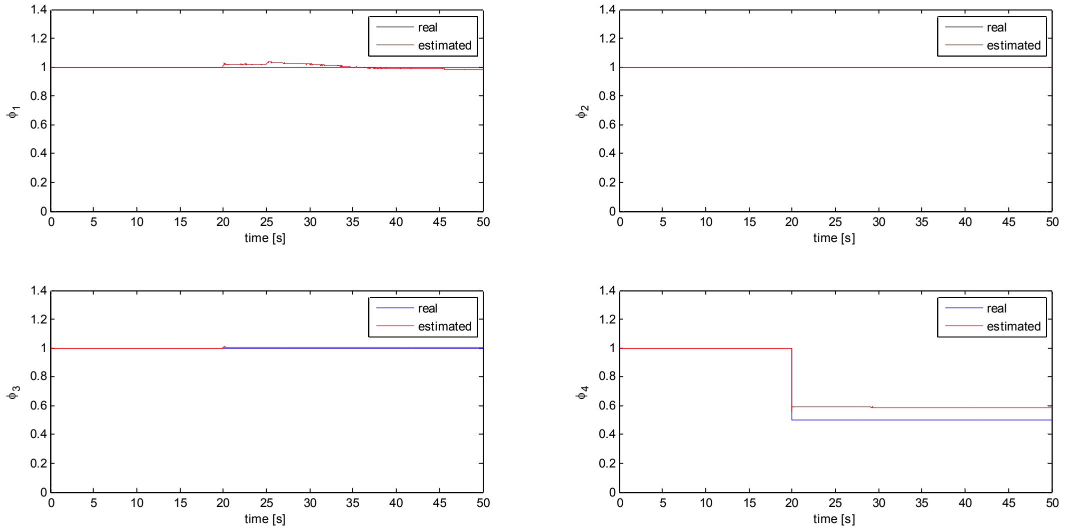

Fig. 122 – Fault scenario 7 (real): fault estimation

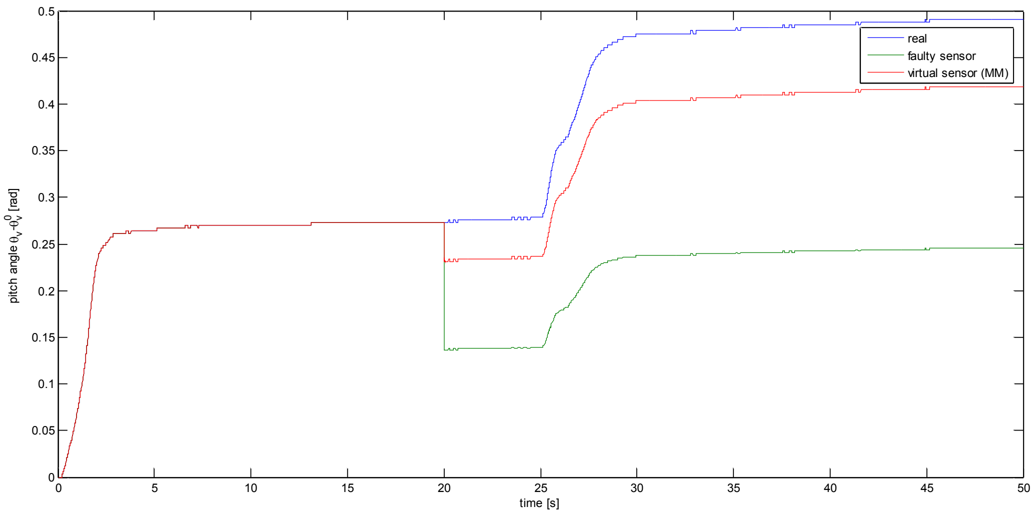

Fig. 123 – Fault scenario 7 (real): faulty output reconstruction

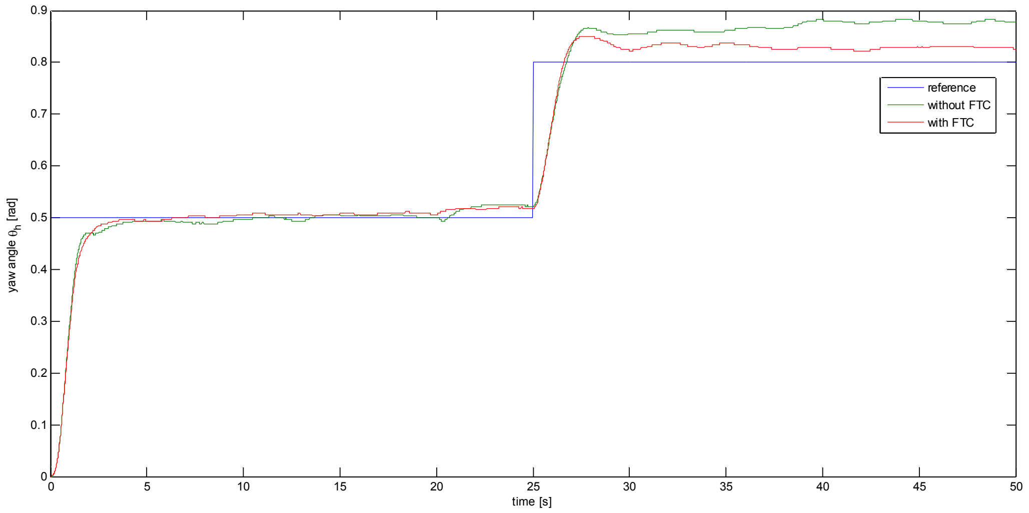

Fig. 124 – Fault scenario 7 (real): yaw angle response

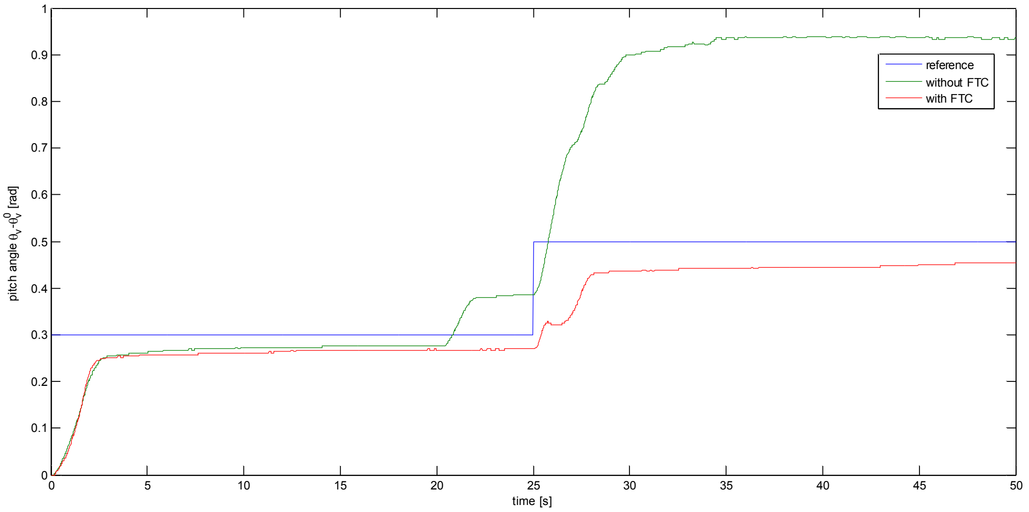

Fig. 125 – Fault scenario 7 (real): pitch angle response

6.4.3.8 – Fault scenario 8

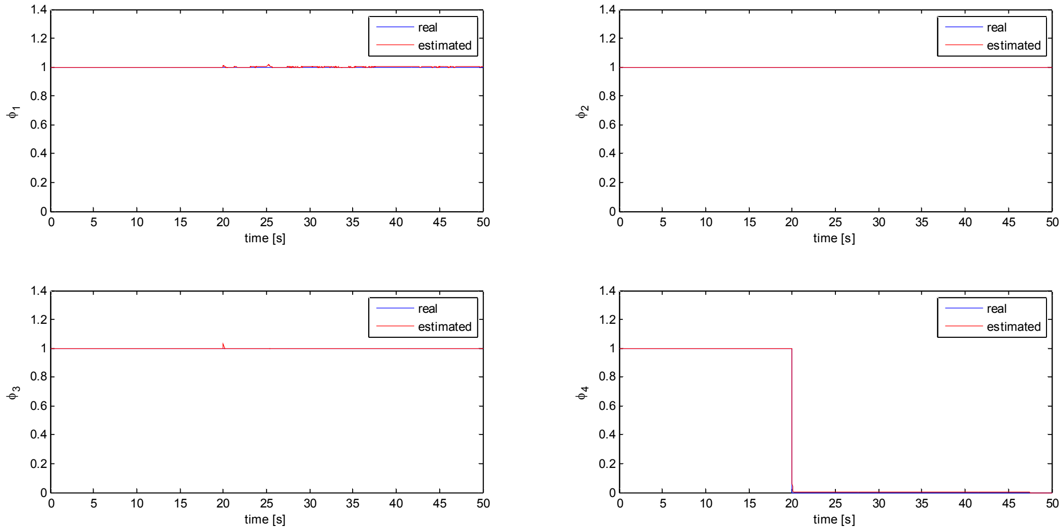

Fig. 126 – Fault scenario 8 (real): fault estimation

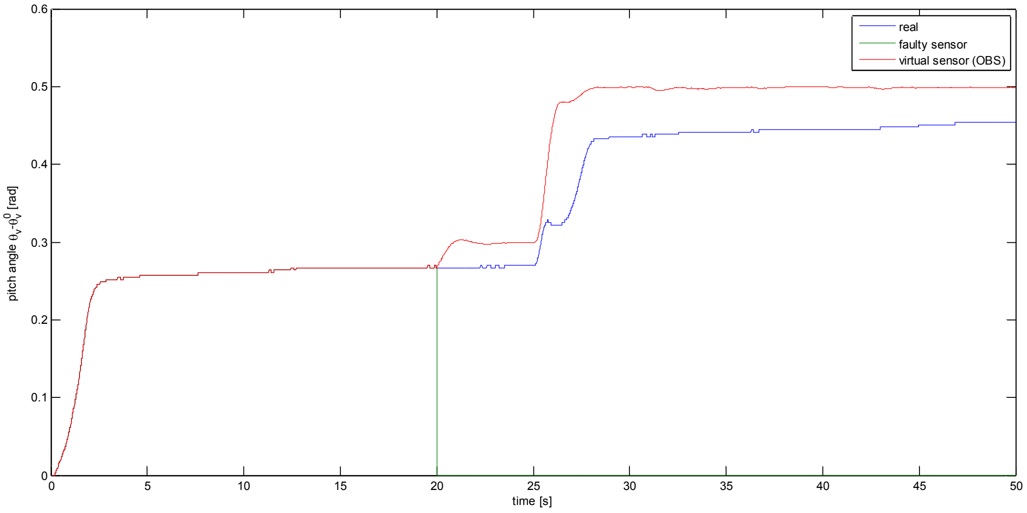

Fig. 127 – Fault scenario 8 (real): faulty output reconstruction

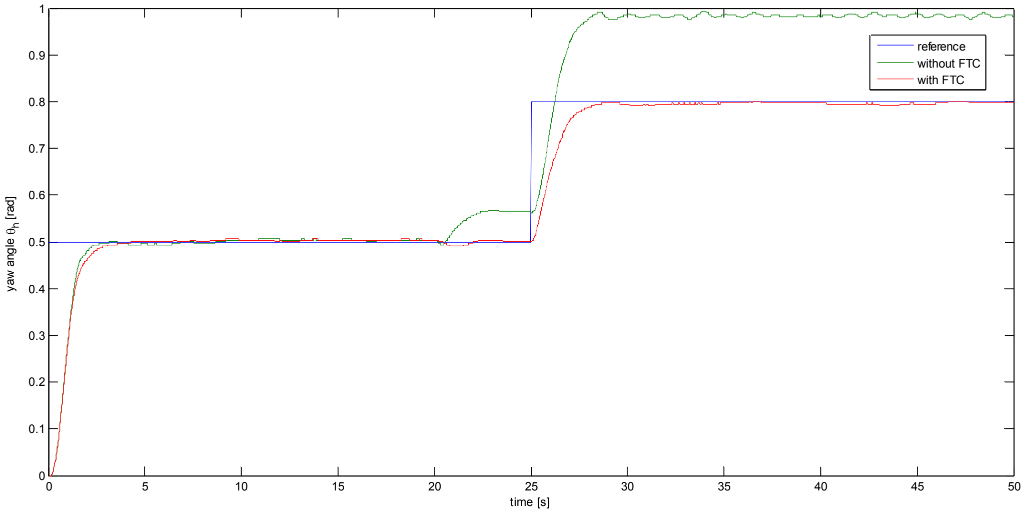

Fig. 128 – Fault scenario 8 (real): yaw angle response

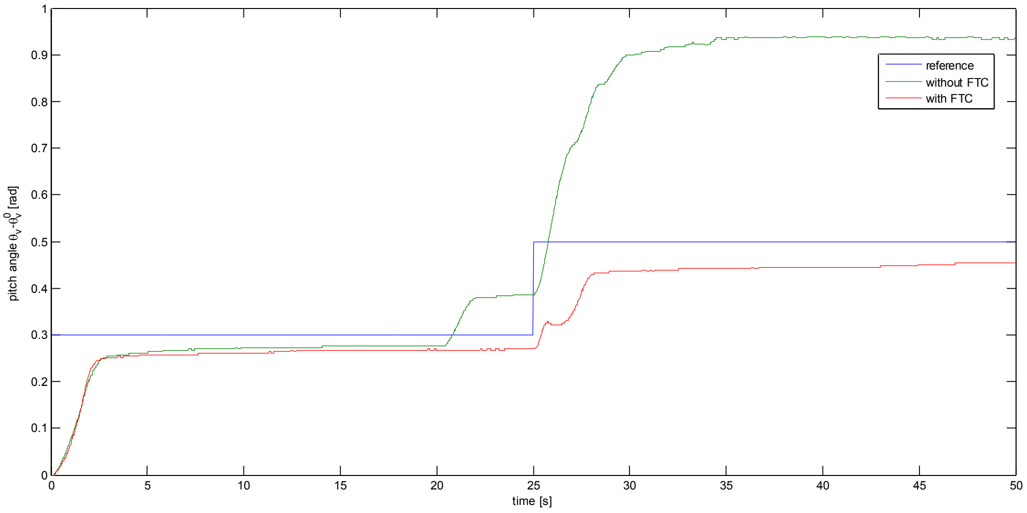

Fig. 129 – Fault scenario 8 (real): pitch angle response

6.4.4 – Comments

The fault identification/estimation strategy proved to work efficiently both in the simulation environment (e.g. ***Fig. 66***) and when applied to the real system (e.g. ***Fig. 98***).

The virtual sensor strategy for Fault Tolerant Control showed to improve the performance of the control system in all cases when applied in a simulation environment. When the sensor was not completely broken, the goodness of the faulty output reconstruction depended on the fault estimation result. Sometimes the reconstruction was almost perfect, as in scenario 1 (partial fault in ω_h sensor, ***Fig. 67***); sometimes it was not, as in scenario 3 (partial fault in θ_h sensor, ***Fig. 75***), and this led to a small offset between the real output of the plant and the corresponding virtual sensor output. When the sensor was completely broken, the performance of the control system depended on the convergence of the virtual sensor observer to the real state. It proved to converge in all cases, as it was supposed to be because the system is observable also when 3 sensors are used instead of 4. Moreover, the presence of the FTC strategy avoided the loss of stability due to the loss of some sensor outputs, as in scenario 4, 5 and 6, where instability occurred (e.g. ***Fig. 80*** and ***Fig. 81***). This proved that, as long as the system is observable and the model is considered to match perfectly the behaviour of the real plant, the FTC strategy gives very good performances.

The virtual sensor strategy proved to work quite efficiently also when applied to the real system. However, a big loss of performances in scenario 4 was noticed (***Fig. 80***), even though it was not so big as when no fault tolerance mechanism was implemented, leading the system to instability. This loss of performance was due to the biggest limit of virtual sensor strategy, that is the fact of being an FTC strategy based on the model of the system. In presence of strong differences between the real behaviour of the TRMS and the model behaviour, an offset appeared between the desired angle and the real one, as the one to be controlled was not the real angle but the virtual sensor angle. Such a difference could be reduced through a better identification of the model parameters, even though it has been noticed that some of the parameters are strongly time-varying (e.g. the constants $k_{chp}, k_{chn}, k_{cvp}, k_{cvn}$ that take into account dynamical effects due to the cables and that influence the steady-state gain of the system).

6.5 – Virtual actuators

This section describes a solution to the reconfiguration problem for actuator failures by introducing the notion of virtual actuators, that are the dual systems of virtual sensors.

The system under consideration is described by the state-space model:

$$\begin{cases} \mathbf{x}_f(k+1) = \mathbf{A}\left(\boldsymbol{\psi}(k)\right)\mathbf{x}_f(k) + \mathbf{B}_f\mathbf{u}_f(k) \\ \mathbf{y}_f(k) = \mathbf{C}\mathbf{x}_f(k) \\ \mathbf{x}_f(0) = \mathbf{x}_{f0} \end{cases} \tag{6.5.1}$$

where the actuator failure is reflected by the matrix $\mathbf{B}_f$.

If the condition $\mathrm{Im}\left(\mathbf{B}_f\right) \supseteq \mathrm{Im}\left(\mathbf{B}\right)$ is satisfied, where Im denotes the image of a matrix, exact model-matching can be reached. The condition can be written in an equivalent form as:

$$rank\left(\mathbf{B}_f\right) = rank\left(\mathbf{B} \;\; \mathbf{B}_f\right) \tag{6.5.2}$$

If condition (6.5.2) is satisfied, the reconfigured controller is given by a static reconfiguration block:

$$\mathbf{u}(k) = \mathbf{N}\mathbf{u}_c(k) \tag{6.5.3}$$

where:

$$\mathbf{N} = \mathbf{B}_f^{+}\mathbf{B} = \left(\mathbf{B}_f^T\mathbf{B}_f\right)^{-1}\mathbf{B}_f^T\mathbf{B} \tag{6.5.4}$$

Notice that by choosing $\mathbf{N}$ as in (6.5.4), $\mathbf{B}_f\mathbf{N} = \mathbf{B}$.

However, the general case is when the condition (6.5.2) is violated.

6.5.1 – Definition of the virtual actuator

Consider the faulty plant (6.5.1). The virtual actuator is defined as the system:

$$\begin{cases} \mathbf{x}_v(k+1) = \mathbf{A}_v\left(\boldsymbol{\psi}(k)\right)\mathbf{x}_v(k) + \mathbf{B}_v\mathbf{u}_c(k) \\ \mathbf{u}_f(k) = \mathbf{C}_v\left(\boldsymbol{\psi}(k)\right)\mathbf{x}_v(k) + \mathbf{D}_v\mathbf{u}_c(k) \\ \mathbf{y}_c(k) = \mathbf{C}\mathbf{x}_v(k) + \mathbf{y}_f(k) \\ \mathbf{x}_v(0) = \mathbf{x}_{v0} \end{cases} \tag{6.5.5}$$

with the state $\mathbf{x}_v \in \mathbb{R}^n$ and matrices:

$$\mathbf{A}_v\left(\boldsymbol{\psi}(k)\right) = \mathbf{A}\left(\boldsymbol{\psi}(k)\right) + \mathbf{B}_f\mathbf{M}_v\left(\boldsymbol{\psi}(k)\right) \tag{6.5.6}$$

$$\mathbf{B}_v = \mathbf{B} - \mathbf{B}_f\mathbf{N}_v \tag{6.5.7}$$

$$\mathbf{C}_v\left(\boldsymbol{\psi}(k)\right) = \mathbf{M}_v\left(\boldsymbol{\psi}(k)\right) \tag{6.5.8}$$

$$\mathbf{D}_v = \mathbf{N}_v \tag{6.5.9}$$

$\mathbf{M}_v$ and $\mathbf{N}_v$ denote matrices that can be freely chosen. Notice that $\mathbf{M}_v$ is scheduled by the vector of varying parameters $\boldsymbol{\psi}(k)$.

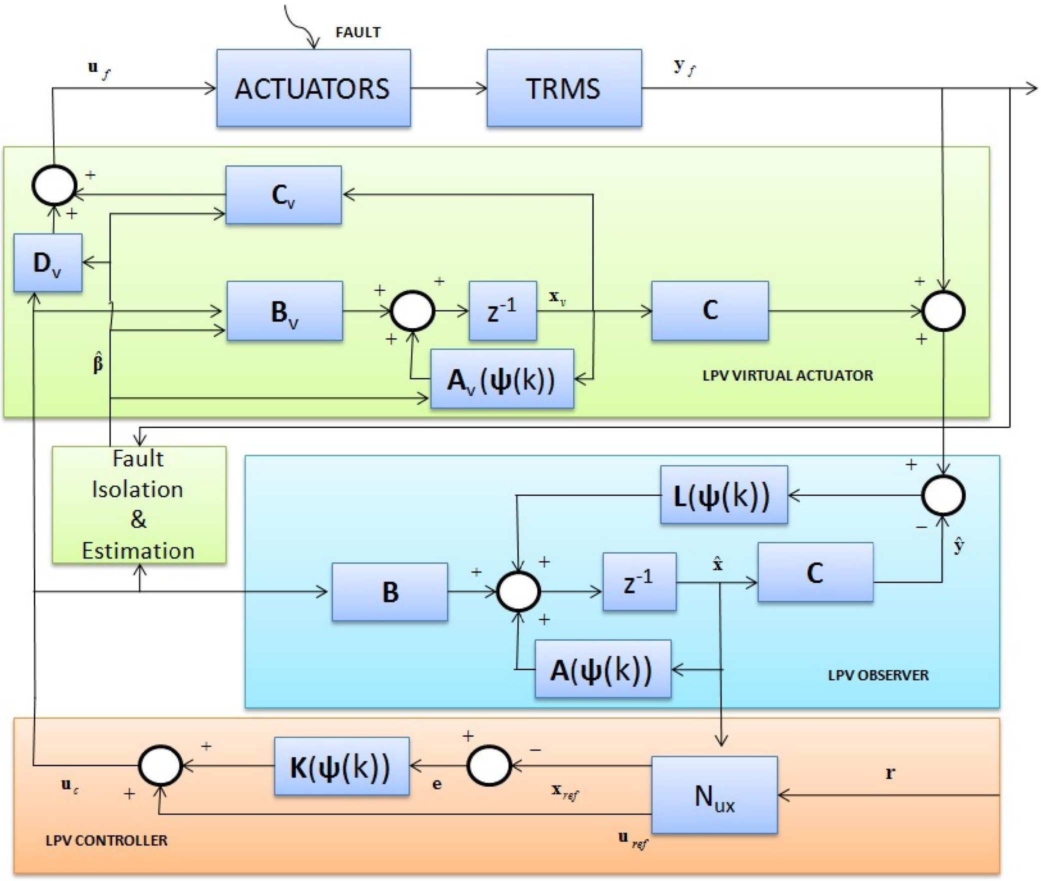

Fig. 130 – Virtual actuator scheme

6.5.2 – Model of the reconfigured plant

The plant together with the virtual actuator leads to the following model of the reconfigured plant:

$$\begin{cases}\begin{pmatrix}\mathbf{x}_f(k+1)\\ \mathbf{x}_v(k+1)\\ \hat{\mathbf{x}}(k+1)\end{pmatrix}=\begin{pmatrix}\mathbf{A}(\psi(k)) & \mathbf{B}_f\mathbf{M}_v(\psi(k)) & \mathbf{0}\\ \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k)) & \mathbf{0}\\ -\mathbf{L}(\psi(k))\mathbf{C} & -\mathbf{L}(\psi(k))\mathbf{C} & \mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{x}_f(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}+\begin{pmatrix}\mathbf{B}_f\mathbf{N}_v\\ \mathbf{B}-\mathbf{B}_f\mathbf{N}_v\\ \mathbf{B}\end{pmatrix}\mathbf{u}_c(k)\\ \mathbf{y}_c(k)=\begin{pmatrix}\mathbf{C} & \mathbf{C} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{x}_f(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}\end{cases}\tag{6.5.10}$$

The introduction of the new state $\mathbf{x}_w(k)=\mathbf{x}_f(k)+\mathbf{x}_v(k)$ leads to the following equivalent model:

$$\begin{cases}\begin{pmatrix}\mathbf{x}_w(k+1)\\ \mathbf{x}_v(k+1)\\ \hat{\mathbf{x}}(k+1)\end{pmatrix}=\begin{pmatrix}\mathbf{A}(\psi(k)) & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k)) & \mathbf{0}\\ -\mathbf{L}(\psi(k))\mathbf{C} & \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C}\end{pmatrix}\begin{pmatrix}\mathbf{x}_w(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}+\begin{pmatrix}\mathbf{B}\\ \mathbf{B}-\mathbf{B}_f\mathbf{N}_v\\ \mathbf{B}\end{pmatrix}\mathbf{u}_c(k)\\ \mathbf{y}_c(k)=\begin{pmatrix}\mathbf{C} & \mathbf{0} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{x}_w(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}\end{cases}\tag{6.5.11}$$

Note that the state $\mathbf{x}_v(k)$ of the second subsystem is not observable by $\mathbf{y}_c$. Hence, this state does not influence the input-output behaviour of the reconfigured plant, whose model can be reduced to:

$$\begin{cases}\mathbf{x}(k+1)=\mathbf{A}(\psi(k))\mathbf{x}(k)+\mathbf{B}\mathbf{u}_c(k)\\ \mathbf{y}_c(k)=\mathbf{C}\mathbf{x}(k)\end{cases}\tag{6.5.12}$$

The model has the same input-output behaviour as the nominal plant for arbitrary parameter matrices $\mathbf{M}_v$ and $\mathbf{N}_v$. Hence, the virtual actuator yields to a reconfigured plant that satisfies the fault-hiding goal.

6.5.3 – Separation principle for the virtual actuator

The reconfigured closed-loop system consists of the reconfigured plant and the controller (6.2.2). If the transformed model is used, the reconfigured closed-loop system is described by:

$$\begin{aligned}\begin{pmatrix}\mathbf{x}_w(k+1)\\ \mathbf{x}_v(k+1)\\ \hat{\mathbf{x}}(k+1)\end{pmatrix}&=\begin{pmatrix}\mathbf{A}(\psi(k)) & \mathbf{0} & \mathbf{B}\mathbf{K}(\psi(k))\\ \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k)) & (\mathbf{B}-\mathbf{B}_f\mathbf{N}_v)\mathbf{K}(\psi(k))\\ -\mathbf{L}(\psi(k))\mathbf{C} & \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C}+\mathbf{B}\mathbf{K}(\psi(k))\end{pmatrix}\begin{pmatrix}\mathbf{x}_w(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}\\ &+\begin{pmatrix}-\mathbf{B}\mathbf{K}(\psi(k))\\ (-\mathbf{B}+\mathbf{B}_f\mathbf{N}_v)\mathbf{K}(\psi(k))\\ -\mathbf{B}\mathbf{K}(\psi(k))\end{pmatrix}\mathbf{x}_{ref}(k)+\begin{pmatrix}\mathbf{B}\\ \mathbf{B}-\mathbf{B}_f\mathbf{N}_v\\ \mathbf{B}\end{pmatrix}\mathbf{u}_{ff}(k)\end{aligned}\tag{6.5.13}$$

The autonomous closed-loop behaviour can be analysed by imposing $\mathbf{u}_{ff}=0$ and $\mathbf{x}_{ref}=0$, thus (6.5.13) simplifies to:

$$\begin{pmatrix}\mathbf{x}_w(k+1)\\ \mathbf{x}_v(k+1)\\ \hat{\mathbf{x}}(k+1)\end{pmatrix}=\begin{pmatrix}\mathbf{A}(\psi(k)) & \mathbf{0} & \mathbf{B}\mathbf{K}(\psi(k))\\ \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k)) & (\mathbf{B}-\mathbf{B}_f\mathbf{N}_v)\mathbf{K}(\psi(k))\\ -\mathbf{L}(\psi(k))\mathbf{C} & \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C}+\mathbf{B}\mathbf{K}(\psi(k))\end{pmatrix}\begin{pmatrix}\mathbf{x}_w(k)\\ \mathbf{x}_v(k)\\ \hat{\mathbf{x}}(k)\end{pmatrix}\tag{6.5.14}$$

A state transformation is performed in order to introduce the observation error $\hat{\mathbf{e}}(k)=\hat{\mathbf{x}}(k)-\mathbf{x}_w(k)$:

$$\begin{pmatrix} \mathbf{x}_w(k+1) \\ \mathbf{x}_v(k+1) \\ \hat{\mathbf{e}}(k+1) \end{pmatrix} = \begin{pmatrix} \mathbf{A}(\psi(k))+\mathbf{BK}(\psi(k)) & \mathbf{0} & \mathbf{BK}(\psi(k)) \\ (\mathbf{B}-\mathbf{B}_f\mathbf{N}_v)\mathbf{K}(\psi(k)) & \mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k)) & (\mathbf{B}-\mathbf{B}_f\mathbf{N}_v)\mathbf{K}(\psi(k)) \\ \mathbf{0} & \mathbf{0} & \mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{x}_w(k) \\ \mathbf{x}_v(k) \\ \hat{\mathbf{e}}(k) \end{pmatrix} \quad (6.5.15)$$

Looking at this model, the *separation principle* can be applied: the matrix $\mathbf{K}(\psi(k))$ influences the behaviour of the state $\mathbf{x}_w(k)$ through the submatrix $\mathbf{A}(\psi(k))+\mathbf{BK}(\psi(k))$ (nominal controller). $\mathbf{M}_v(\psi(k))$ affects the behaviour of the virtual actuator state $\mathbf{x}_v(k)$ through the submatrix $\mathbf{A}(\psi(k))+\mathbf{B}_f\mathbf{M}_v(\psi(k))$ (LPV virtual actuator). Finally, the observer error $\hat{\mathbf{e}}(k)$ is affected by the matrix $\mathbf{L}(\psi(k))$ through the submatrix $\mathbf{A}(\psi(k))+\mathbf{L}(\psi(k))\mathbf{C}$ (LPV state observer).

The set σ of the reconfigured closed-loop system (6.5.15) consists of the set of eigenvalues of the nominal closed-loop system (6.2.4) plus the set of eigenvalues of the virtual actuator (6.5.5) and the set of eigenvalues of the LPV observer (6.2.3):

$$\sigma = \sigma\{\mathbf{A}+\mathbf{BK}\} \cup \sigma\{\mathbf{A}+\mathbf{B}_f\mathbf{M}_v\} \cup \sigma\{\mathbf{A}+\mathbf{LC}\}$$

Thus, the closed loop LPV controller, the LPV virtual actuator and the LPV state observer can be designed independently. This is true for arbitrary matrices $\mathbf{M}_v$ and $\mathbf{N}_v$ of the virtual actuator. Clearly, the matrix $\mathbf{M}_v$ has to be chosen so that the matrix $\mathbf{A}+\mathbf{B}_f\mathbf{M}_v$ has eigenvalues with negative real parts in order to ensure the stability of the reconfigured closed-loop system. This can be done if and only if the pair $(\mathbf{A},\mathbf{B}_f)$ is stabilizable.

The matrices $\mathbf{M}_v$ and $\mathbf{N}_v$ can be chosen in such a way that the difference between the behaviour of the nominal and the reconfigured closed-loop system is minimized. A complete reconfiguration is possible if the matrix $\mathbf{N}_v$ can be chosen such that the matrix $\mathbf{B}_v$ vanishes. This is the case when the virtual actuator reduces to the static reconfiguration block. If this is not true, the static reconfiguration block does not solve the reconfiguration problem, and the dynamical part of the virtual actuator becomes active. The generalised virtual actuator permits reducing the effect of the virtual actuator if the matrix $\mathbf{B}_v$ can be made very small by choosing the matrix $\mathbf{N}_v$ appropriately.

6.5.4 – Polytopic LPV Virtual Actuator

The LPV virtual actuator (6.5.5) can be defined in a polytopic way as:

$$\begin{cases} \mathbf{x}_v(k+1) = \sum_{j=1}^{N} \alpha_k^j(\psi(k))\left[\mathbf{A}_j\mathbf{x}_v(k)+\mathbf{B}_f\mathbf{M}_{v,j}\mathbf{x}_v(k)+\mathbf{Bu}_c(k)-\mathbf{B}_f\mathbf{N}_v\mathbf{u}_c(k)\right] \\ \mathbf{u}_f(k) = \sum_{j=1}^{N} \alpha_k^j(\psi(k))\left[\mathbf{M}_{v,j}\mathbf{x}_v(k)+\mathbf{N}_v\mathbf{u}_c(k)\right] \\ \mathbf{y}_c(k) = \mathbf{Cx}_v \end{cases} \quad (6.5.16)$$

The matrices $\mathbf{M}_{v,j} \in \mathbb{R}^{n_u \times n_x}$ are the gains of the LPV virtual actuator defined for each j^{th} model.

6.5.5 – LPV Virtual Actuator Design

Resuming, the design of the LPV virtual actuator implies selecting:

- matrices $\mathbf{K}_j$ of (5.1.5) in order to guarantee closed-loop stability of the original system assuming that the pair $(\mathbf{A}(\psi(k)),\mathbf{B})$ is stabilizable for all $\psi \in \mathbf{\Psi}$;

- matrices $\mathbf{M}_{v,j}$ of (6.5.16) assuming that the pair $\left(\mathbf{A}(\boldsymbol{\psi}(k)),\mathbf{B}_f\right)$ is stabilizable for all $\boldsymbol{\psi}\in\boldsymbol{\Psi}$;
- matrices $\mathbf{L}_j$ of (5.1.7) in order to correct the LPV state observer error of the reconfigured system assuming that the pair $\left(\mathbf{A}(\boldsymbol{\psi}(k)),\mathbf{C}\right)$ is detectable for all $\boldsymbol{\psi}\in\boldsymbol{\Psi}$;
- matrix $\mathbf{N}_v$ such that $\mathbf{B}=\mathbf{B}_f\mathbf{N}_v$ assuming that the rank condition (6.5.2) is satisfied.

The design of the LPV virtual actuator is done by applying LMI techniques described in *Chapter 2* to place the poles in a desired region of the complex plane. Thus, equation (2.3.13) becomes:

$$\begin{cases}\begin{pmatrix} -r\mathbf{X} & q\mathbf{X}+\mathbf{A}_j\mathbf{X}+\mathbf{B}_f\boldsymbol{\Gamma}_j & 0 & 0 \\ q\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\boldsymbol{\Gamma}_j^T\mathbf{B}_f^T & -r\mathbf{X} & 0 & 0 \\ 0 & 0 & S_{\min}\mathbf{X}-\frac{1}{2}\left(\mathbf{A}_j\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\mathbf{B}_f\boldsymbol{\Gamma}_j+\boldsymbol{\Gamma}_j^T\mathbf{B}_f^T\right) & 0 \\ 0 & 0 & 0 & -S_{\max}\mathbf{X}+\frac{1}{2}\left(\mathbf{A}_j\mathbf{X}+\mathbf{X}\mathbf{A}_j^T+\mathbf{B}_f\boldsymbol{\Gamma}_j+\boldsymbol{\Gamma}_j^T\mathbf{B}_f^T\right)\end{pmatrix}<0 \\ X>0\end{cases} \quad (6.5.17)$$

Finally, the vertices of the virtual actuator gains are then obtained as:

$$\mathbf{M}_{v,j}=\boldsymbol{\Gamma}_j\mathbf{X}^{-1} \quad (6.5.18)$$

6.5.6 – Fault isolation and estimation for the TRMS actuators

Hereafter, the procedure to isolate and estimate the multiplicative actuator faults is explained. We suppose having at disposal a Fault Detection device. This means that, as soon as a fault appears in the system, the Fault Detection device is able to detect that there is a fault, and the procedure for fault isolation and estimation is started.

This procedure is based on the fact that, in presence of a fault in an actuator, there appears a difference between the behaviour of the real actuator and the behaviour of its model. Such a difference can be used to estimate the magnitude of the fault through a parameter estimation approach.

The approach to be used is the so-called "Recursive Least Square method" with a forgetting factor $\varsigma(k)$ **[Ljung, 1987]**.

The model behaviour of the actuator is assumed to be described by the following LPV discrete-time equation:

$$\omega^{sim}(k+1)=a\left(\omega^{sim}(k)\right)\omega^{sim}(k)+bu(k) \quad (6.5.19)$$

where a is given by (3.3.4) for the tail rotor and (3.3.11) for the main rotor and b is given by (3.3.17) for the tail rotor and (3.3.19) for the main rotor, respectively.

The faulty behaviour of the actuator is assumed to be described by (β is the fault magnitude):

$$\omega^{fault}(k+1)=a\left(\omega^{fault}(k)\right)\omega^{fault}(k)+\beta bu(k) \quad (6.5.20)$$

The assumptions $\omega^{sim}(k+1)\simeq\omega^{sim}(k)$ and $\omega^{fault}(k+1)\simeq\omega^{fault}(k)$ lead to:

$$\omega^{sim}(k)\left(1-a\left(\omega^{sim}(k)\right)\right)=bu(k) \quad (6.5.21)$$

$$\omega^{fault}(k)\left(1-a\left(\omega^{fault}(k)\right)\right)=\beta bu(k) \quad (6.5.22)$$

Thus, the regression equation can be written as:

$$\frac{\omega^{fault}(k)\left(1-a\left(\omega^{fault}(k)\right)\right)}{\omega^{sim}(k)\left(1-a\left(\omega^{sim}(k)\right)\right)}=\beta \quad (6.5.23)$$

The fault estimation is thus given by:

$$\begin{cases} P_i(k) = \dfrac{\left[P_i(k-1) - \dfrac{P_i(k-1)^2}{\varsigma_i(k) + P_i(k-1)}\right]}{\varsigma_i(k)} \\ \hat{\beta}_i(k) = \hat{\gamma}_i(k-1) + \Phi_i(k)\left[\dfrac{\omega^{fault}(k)\left(1 - a(\omega^{fault}(k))\right)}{\omega^{sim}(k)\left(1 - a(\omega^{sim}(k))\right)} - \hat{\beta}_i(k-1)\right] \\ \Phi_i(k) = \dfrac{P_i(k-1)}{\varsigma_i(k) + P_i(k-1)} \end{cases} \tag{6.5.24}$$

where $P_i(k)$ is the variance of the fault estimation $\hat{\phi}_i(k)$ and $\Phi_i(k)$ is the gain of the algorithm at instant *k*. The Twin Rotor MIMO System has 2 actuators, that directly affect the dynamics of ω_h and ω_v respectively, through the input voltages u_h and u_v. When a fault occurs (denoted by β_1 and β_2, respectively), the fault isolation/estimation approach is applied to discover which actuator is faulty and which is the corresponding magnitude. The values provided from the fault estimation $\hat{\beta}_1$ and $\hat{\beta}_2$ are used to apply the virtual actuator mechanism, thus obtaining fault tolerance. It has been noticed that a complete loss of an actuator leads the system to the loss of the stabilizability property; hence, a fault tolerance approach for faults in the actuators of the TRMS is possible only as long as the fault is partial. In this case, the rank condition (6.5.2) is satisfied and the virtual actuator reduces to a static block designed in a model-matching way.

In this case, equation (6.5.4) gives:

$$\mathbf{N}_v = \begin{pmatrix} \dfrac{1}{\hat{\beta}_1} & 0 \\ 0 & \dfrac{1}{\hat{\beta}_2} \end{pmatrix} \tag{6.5.25}$$

6.6 – Virtual actuator results

6.6.1 – Actuator fault scenarios

Different fault scenarios have been designed to test the effectiveness of the fault tolerance mechanism. In all the scenarios, the TRMS is asked to follow the angle references (5.3.12):

$$\begin{cases} \theta_h^{ref} = 0.5, \quad \theta_v^{ref} = 0.3 \quad if \quad 0s \le t < 25s \\ \theta_h^{ref} = 0.8, \quad \theta_v^{ref} = 0.5 \quad if \quad 25s \le t < 50s \end{cases} \tag{6.6.1}$$

Faults always occur at time $t = 20s$. Fault scenarios differ in which actuator is faulty and which is its magnitude, as resumed in the following table:

Scenario	9	10
Faulty actuator	u_h	u_v
Fault magnitude	0.5	0.5

Fig. 131 – Fault scenarios (virtual actuators)

The forgetting factors in (6.5.24) have been chosen as $\varsigma_1(k) = \varsigma_2(k) = 0.9\,\forall k$.

6.6.2 – Simulation results

Hereafter, the results obtained in simulation environment are presented. For each scenario four figures are shown. The first figure shows the fault and its estimation. The second figure shows the system input variable that corresponds to the faulty actuator: the output of the nominal controller, the faulty value and the virtual actuator value are compared (the desired behaviour is when the output of the nominal controller and the virtual actuator value are the same). The third figure shows the response of the yaw angle both when a fault tolerance mechanism is implemented and when there is not such a mechanism. Analogously, the fourth figure shows the response of the pitch angle with the same comparison.

6.6.2.1 – Fault scenario 9

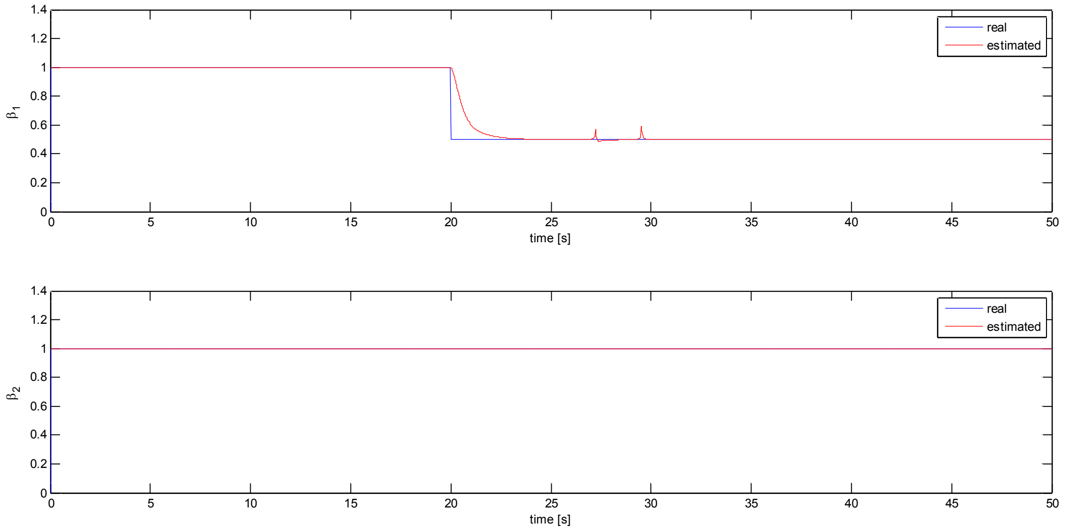

Fig. 132 – Fault scenario 9 (simulation): fault estimation

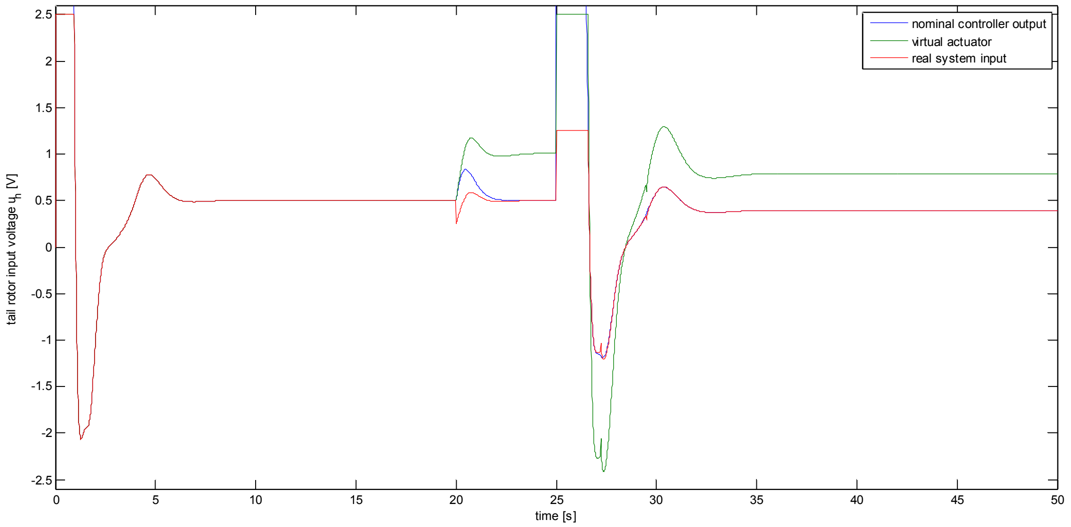

Fig. 133 – Fault scenario 9 (simulation): faulty input reconstruction

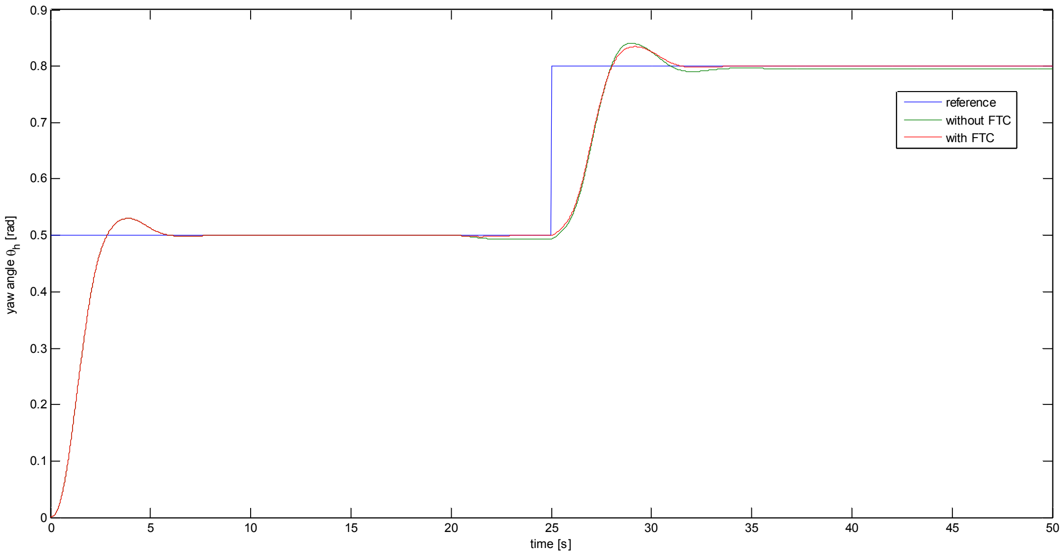

Fig. 134 – Fault scenario 9 (simulation): yaw angle response

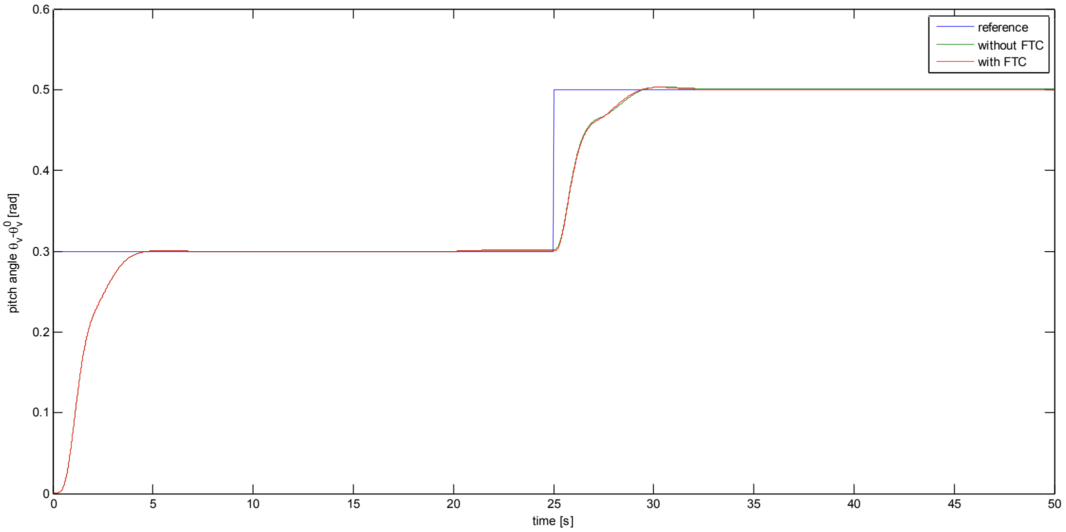

Fig. 135 – Fault scenario 9 (simulation): pitch angle response

6.6.2.2 – Fault scenario 10

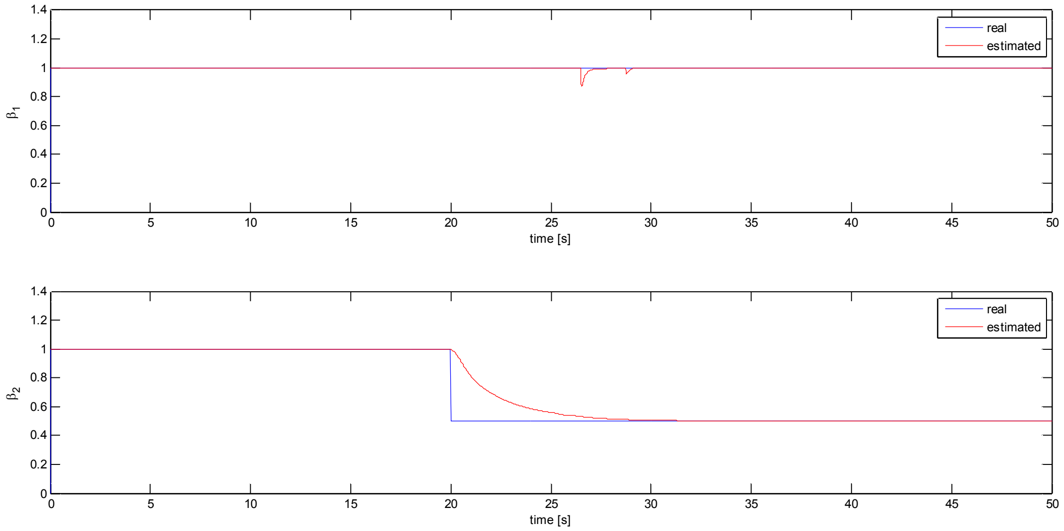

Fig. 136 – Fault scenario 10 (simulation): fault estimation

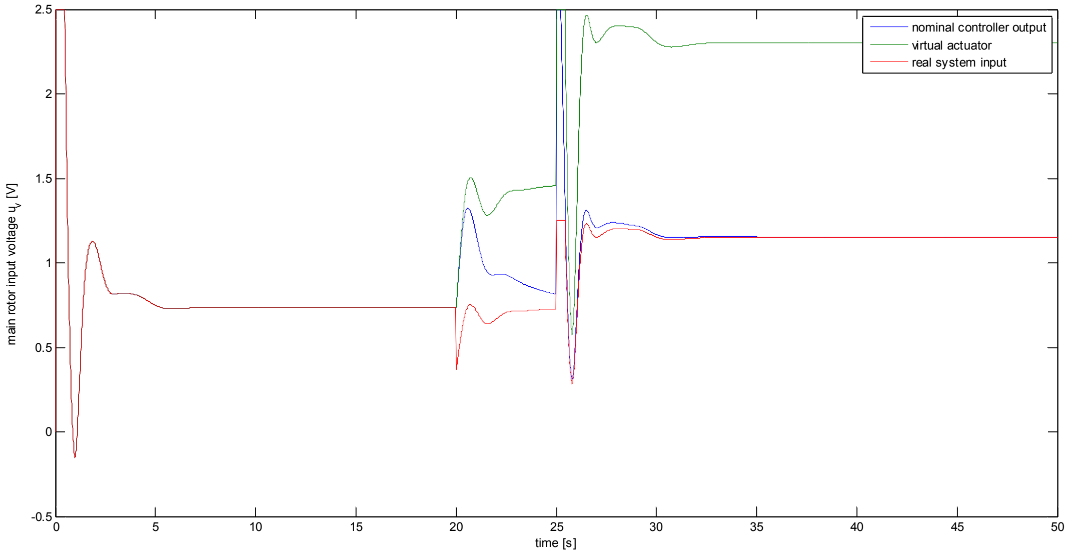

Fig. 137 – Fault scenario 10 (simulation): faulty input reconstruction

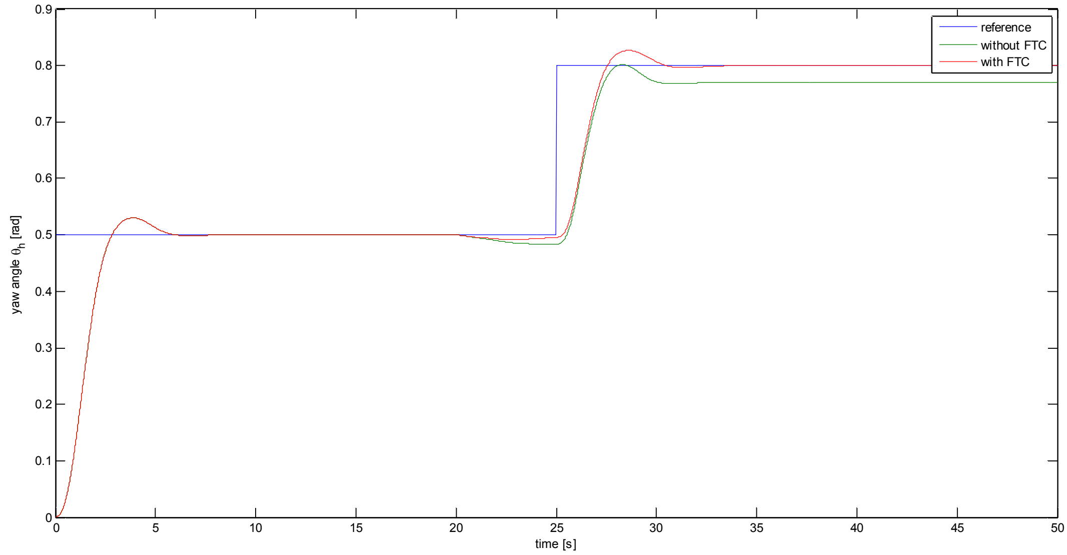

Fig. 138 – Fault scenario 10 (simulation): yaw angle response

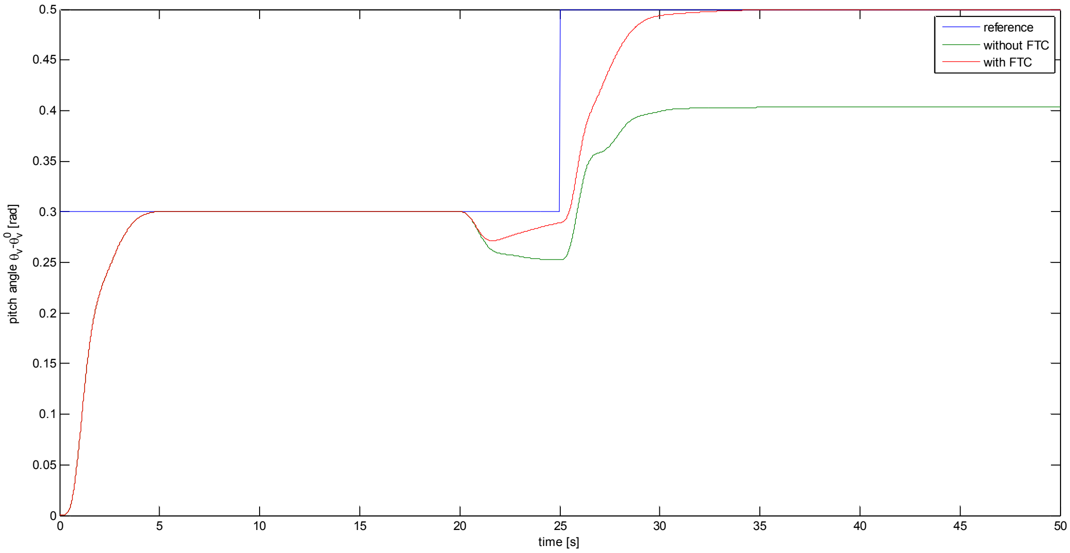

Fig. 139 – Fault scenario 10 (simulation): pitch angle response

6.6.3 – Practical results

In this section, the results obtained with the real TRMS are presented. Analogously to the previous section “Simulation results”, four figures are shown for each scenario. When a fault occurs, the virtual actuator static block is activated.

6.6.3.1 – Fault scenario 9

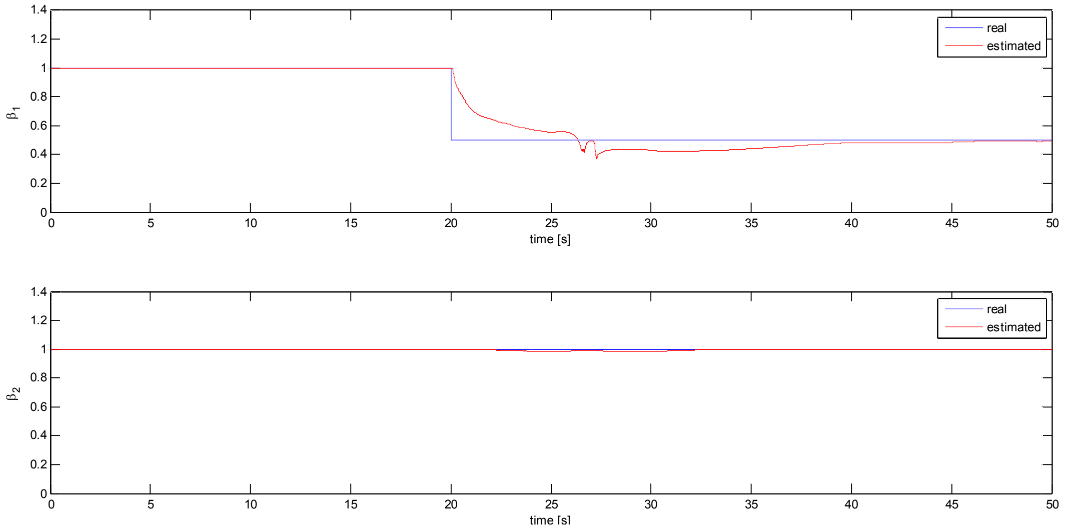

Fig. 140 – Fault scenario 9 (real): fault estimation

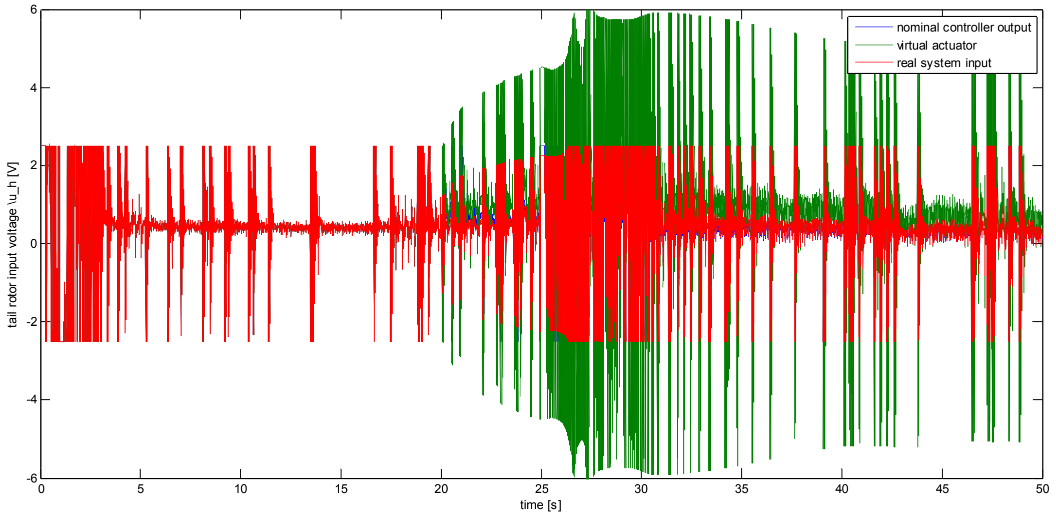

Fig. 141 – Fault scenario 9 (real): faulty input reconstruction

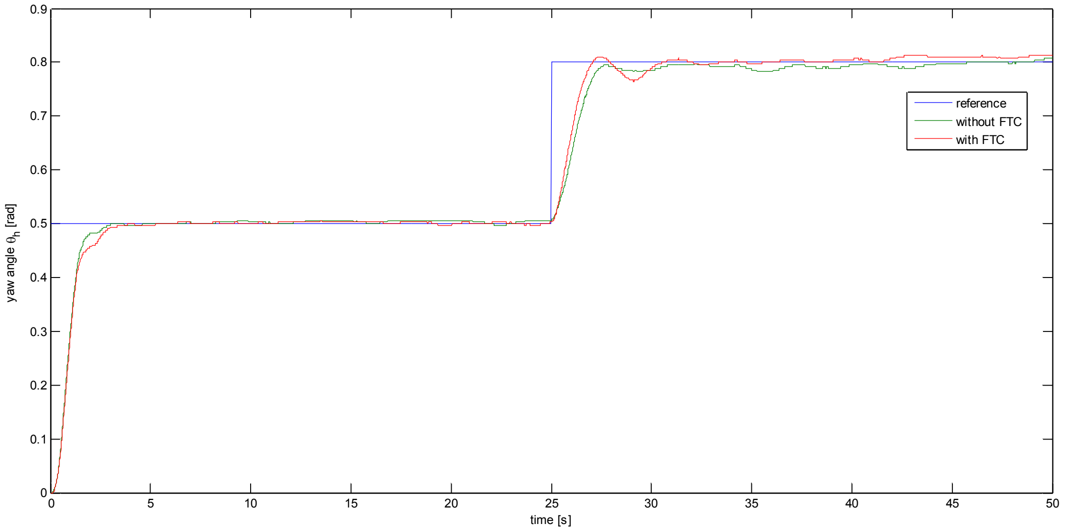

Fig. 142 – Fault scenario 9 (real): yaw angle response

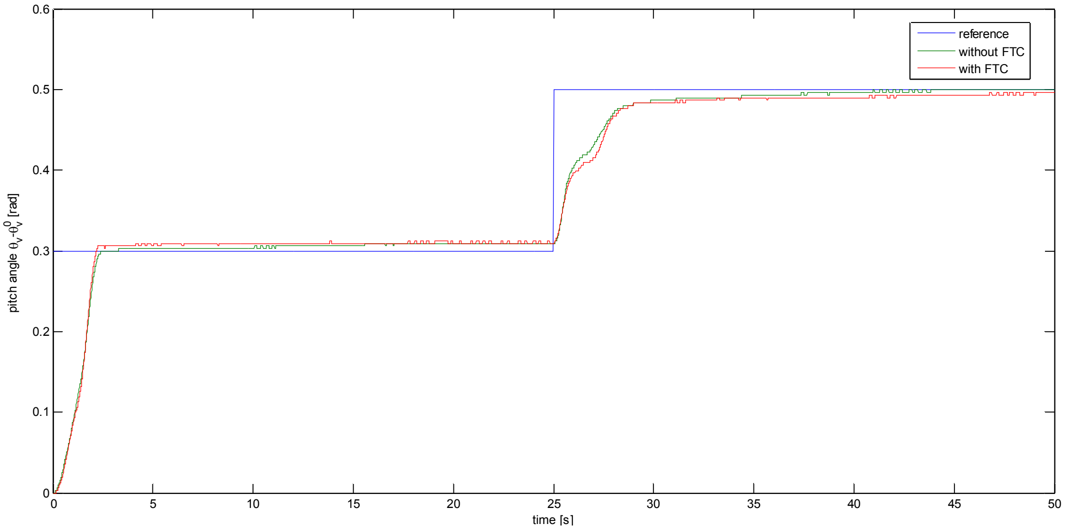

Fig. 143 – Fault scenario 9 (real): pitch angle response

6.6.3.2 – Fault scenario 10

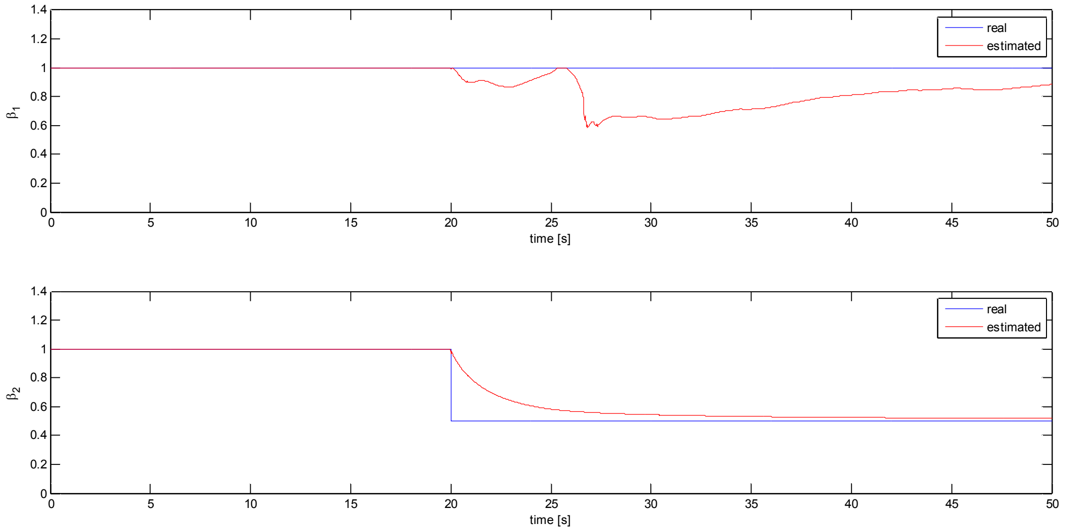

Fig. 144 – Fault scenario 10 (real): fault estimation

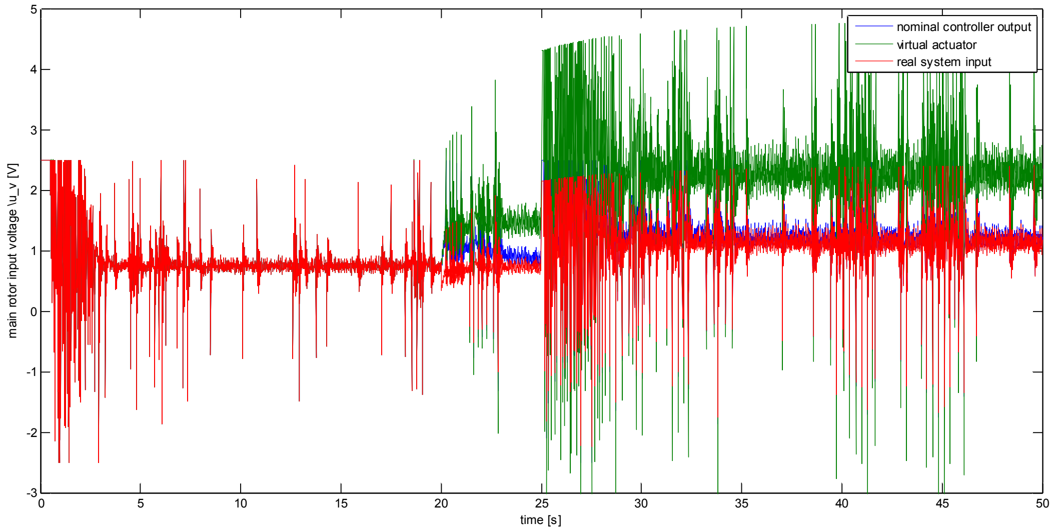

Fig. 145 – Fault scenario 10 (real): faulty input reconstruction

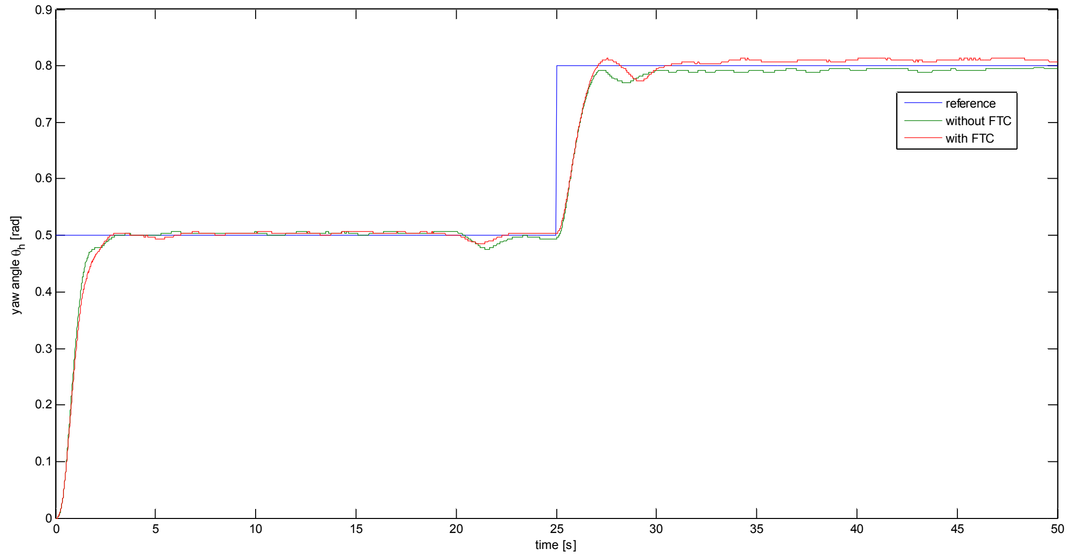

Fig. 146 – Fault scenario 10 (real): yaw angle response

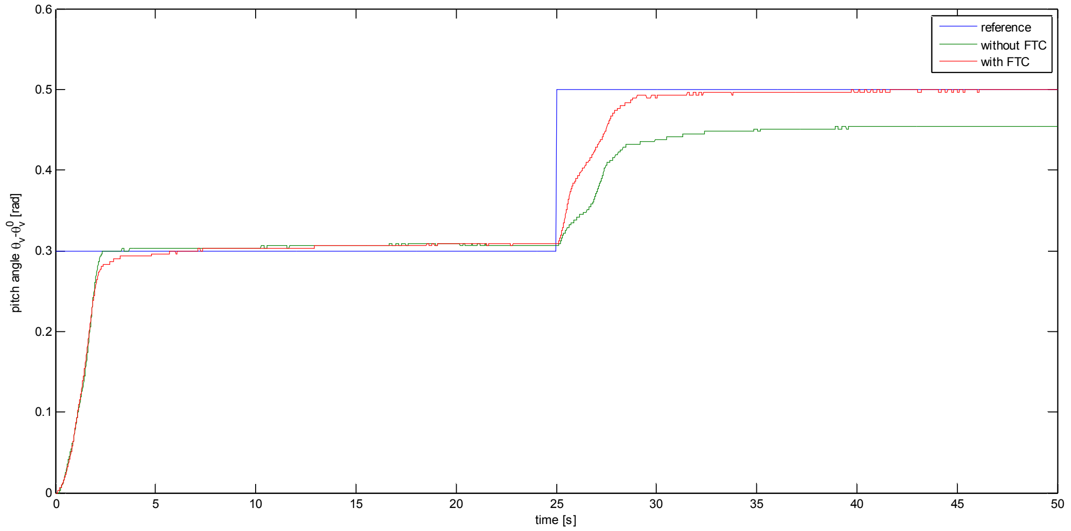

Fig. 147 – Fault scenario 10 (real): pitch angle response

6.4.4 – Comments

The fault isolation/estimation strategy proved to work efficiently both in simulation environment (***Fig. 132*** and ***Fig. 136***) and when applied to the real system (***Fig. 140*** and ***Fig. 144***).

A gain fault in the tail actuator showed not to be critical for the control performance in both cases. However, in simulation, the fault occurrence made the control model to differ from the controlled one, and this caused the appearance of a steady-state error that was eliminated by the fault tolerance mechanism (***Fig. 134***).

A gain fault in the main actuator showed to cause a big loss of performance in the control system, which demonstrated not to be able to control the system at the desired yaw and pitch angles as long as no fault tolerance mechanism was implemented. The virtual actuator strategy proved to work efficiently both in simulation environment (***Fig. 138*** and ***Fig. 139***) and when applied to the real system (***Fig. 146*** and ***Fig. 147***).

6.7 – Summary

In this chapter a Fault Tolerant LPV virtual sensor and actuator approach for non-linear systems that can be approximated by an LPV model has been proposed. This FTC method adapts the faulty plant to the nominal LPV controller instead of adapting the LPV controller to the faulty plant. In this way, the faulty plant together with the LPV virtual sensor or actuator block allows the LPV controller to see the same plant as before the fault. The LPV virtual sensor and the LPV virtual actuator are designed using polytopic LPV techniques and LMI regions.

Virtual sensors and virtual actuators belong to the so-called Active FTC methods. In order to apply this kind of methods, fault detection, isolation and estimation are needed. Assuming that the fault had already been detected, a procedure for isolating which sensor/actuator is faulty and for estimating the magnitude of the fault has been proposed. Such a method is based on recursive least squares, as the fault is considered to be a parameter that needs to be estimated.

The fault isolation and estimation method and the virtual sensor and virtual actuator strategy have been tested both in a simulation framework and by applying them to the real system.

The proposed methodology showed to work well in most cases, even though there appeared limits due to modelling imperfections and to the characteristics of the TRMS system.

7 – CONCLUSIONS AND FUTURE WORKS

7.1 – Conclusions

The main idea behind this thesis was the design of a control with fault tolerance mechanisms for non-linear systems that can be represented as LPV models, with application to a Twin Rotor MIMO System (TRMS). To reach this aim, an LPV model was obtained and identified with real data.
The conclusions that have been extracted in this thesis are the following:

- Modelling non-linear MIMO systems in an LPV fashion is an efficient approach that permits the use of linear methodologies for designing a controller and an observer, taking into account both coupling and non-linear effects;
- Least-squares methods proved to be a simple and anyway good strategy for identifying the parameters of a non-linear model through comparison of the output of a non-linear simulator with data coming from the real system;
- The proposed control strategy has shown its effectiveness and performance both to drive the system to the desired yaw and pitch angles from the equilibrium and when an external disturbance was applied. Different LPV models were used and the one that gave the best performances was the Absolute LPV model with 6 states. In addition, the LPV controller showed better performances when compared with a PID controller tuned through Ziegler-Nichols empirical methods;
- The proposed fault tolerance mechanisms worked well in simulation, even though they showed their limits when applied to the real system. In particular, as the proposed fault tolerance mechanisms are based on the model of the plant, the model should fit the reality as well as possible. Moreover, the virtual actuator strategy could not be applied when one of the TRMS actuators was completely broken. This suggests that the proposed strategy would give better performances if applied to systems where there is a strong redundancy of both sensors and actuators.

7.2 – Future works

To continue the research proposed in this thesis, some ideas are outlined below:

- Some improvements in the control performances could be obtained through a better modelling of the TRMS. Some effects were not considered in the non-linear model used in this thesis, and including them could lead to better performances, even though this would make more difficult the identification procedure, as the number of parameters to be estimated would increase;
- The TRMS showed a time-varying behaviour; in particular, there were some parameters of the model (e.g. those related with the cable effects) whose value could vary strongly. This would suggest adding an adaptive mechanism to the control. For example, the LPV box could be calculated by assuming that those parameters take values in an interval, and the aim of the adaptive mechanism would be to estimate online the parameters values and use them both for LPV scheduling and for calculating the feedforward control action. Alternatively, robustness could be added offline to the control, by extending robust control methods to LPV systems;
- The approach used for obtaining a polytopic LPV model was the bounding box; wrapping the varying parameters into a box is a conservative approach that leads to considering combinations of parameters that would never appear; hence, a convex hull approach

could lead to a less conservative model that better fits the reality, thus increasing the control performances;

- The region where to place the poles of the closed-loop system and of the observer could be chosen through optimum control techniques. Moreover, the presence of the saturation in the control loop can be taken into account to choose properly the LMI regions;
- In this thesis it has been assumed that fault detection was already available, and only a fault isolation and estimation method was developed. A fault detection mechanism should be developed and integrated to the overall scheme;
- The developed fault isolation and estimation method proved to be sufficient to identify where a fault occurred and which was the magnitude of the fault; however, no robustness guarantee has been given and more complex strategies could be investigated to assure a robust fault diagnosis (these strategies could implicitly include fault detection);
- The applied fault tolerance strategies showed limits in their application to the TRMS because of the scarcity of sensors and actuators in the chosen case of study: virtual sensor and virtual actuators could lead to better performances when applied to systems where a strong redundance of sensors and actuators is available. It would be of interest to apply these techniques to systems that are similar to the TRMS, but that have the advantage of having at disposal more sensors or actuators (e.g. the quadrotor, an aircraft propelled by four rotors; in this case, a complete fault in one of the rotors would presumably not lead to the loss of stabilizability, and the dynamical part of the virtual actuator could be used). Another interesting application could be a distributed system, where lots of sensors and actuators are available (e.g. a water supply network).

BIBLIOGRAPHICAL REFERENCES

[Ahmad, 2002] – S. Ahmad, A. Chipperfield, and M.O. Tokhi, "Dynamical modelling and open-loop control of a twin rotor multi-input multi-output system*", Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. 216, no. 6, pp. 477-496, 2002.

[Ahmed, 2009] – Q. Ahmed, A.I. Bhatti, and S. Iqbal, "Nonlinear Robust Decoupling Control Design for Twin Rotor System", 7th Asian Control Conference, Hong Kong, China, 2009.

[Alam, 2007] – M. S. Alam and M.O. Tokhi, "Modelling of a twin rotor system: a particle swarm optimization approach", *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* Vol. 221, no. 3, pp. 353-374, 2007.

[Andersen, 2005] – P. Andersen and T.S. Pedersen, "Modeldannelse", Lecture note for the course "Modelling" at 6th semester at the Department of Control Engineering, Aalborg University, 2005.

[Apkarian, 1995] – P. Apkarian, P. Gahinet and G. Becker, "Self-Scheduled H_∞ Control of Linear Parameter-varying Systems: a Design Example". *Automatica,* Vol. 31, no. 9, pp. 1251-1261, 1995.

[Balas, 2002] – G.J. Balas, "Linear parameter-varying control and its application to a turbofan engine", *International Journal of Robust and Nonlinear Control,* Vol.12, pp. 763-796, 2002.

[Barber, 1996] – C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The Quickhull algorithm for convex hulls", *ACM Trans. On Mathematical Software,* Vol. 22, no. 4, pp. 469-483, 1996.

[Blanke, 2003] – M. Blanke, M. Kinnaert, J. Lunze and M. Staroswiecki, **Diagnosis and Fault-Tolerant Control**, Springer-Verlag, Berlin Heidelberg, 2003.

[Chilali, 1996] – M. Chilali, P. Gahinet, "H_∞ Design with Pole Placement Constraints: An LMI Approach". *IEEE Transactions on Automatic Control,* Vol. 41, no. 3, pp. 358-367, 1996.

[Christensen, 2006] – H. Vie Christensen et al., “Modelling and Control of a Twin-Rotor MIMO System”, Report of Department of Control Engineering Institute of Electronic Systems of Aalborg University, 2006.

[Darus, 2004] – I. Z. M. Darus, F. M. Aldebrez, and M.O. Tokhi, “Parametric Modelling of a Twin Rotor System using Genetic Algorithms”, First International Symposium on Control, Communications and Signal Processing, 2004.

[Dutka, 2003] – A. S. Dutka, W. Ordys, and M. J. Grimble, “Non-Linear Predictive Control of 2 dof helicopter model”, 2nd IEEE Conference on Decision and Control Maui, Hawaii, USA, 2003.

[Feedback, 33-007-4M5] – Twin Rotor MIMO System 33-007-4M5, Advanced Teaching Manual (Feedback Instruments Limited, Crowborough, UK).

[Franklin, 1997] – G. F. Franklin, J. D. Powell, and M. L. Workman, **Digital Control of Dynamic Systems**, 3rd ed. Addison Wesley Longman, 1997.

[Gabriel, 2008] – C. Gabriel, “Modelling, Simulation and Control of a Twin Rotor MIMO-System”, Project Thesis at Department of Systems Engineering and Control of Universidad Politécnica de Valencia, 2008.

[Gutman, 1981] – S. Gutman, E. I. Jury, “A general theory for matrix root clustering in subregions of the complex plane”, *IEEE Transactions on Automatic Control,* Vol. AC-26*,* pp. 853-863, 1981.

[Knudsen, 1995] – M. Knudsen and J. G. Jensen, “Estimation of Non-Linear DC Models Using a Sensitivity Approach”, 1995.

[Kumar, 2000] – A. Kumar and M. R. Anderson, “A comparison of LPV Modeling Techniques for Aircraft Control”, AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver. AIAA-2000-4458, 2000.

[Kwiatkowski, 2006] – A. Kwiatkowski, M. T. Boll, H. Werner, "Automated Generation and Assessment of Affine LPV Models", Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, pp. 6690-6695, 2006.

[Ljung, 1987] – L. Ljung, **System Identification: Theory for the User**, Prentice-Hall, 1987.

[López, 1998] – M. López Martínez, "Modelo y control LQR por Par Calculado de un Sistema Mecanico de 2 gdl impulsado por dos helices", Proyecto Final de Carrera. Escuela Superior de Ingenieros (Supervising Director: L. Fernando Castaño).

[Lopez, 2003a] – M. Lopez Martinez and F. Rubio, "Control of a laboratory helicopter using feedback linearization", 2003 European Control Conference, Cambridge, UK, 2003.

[Lopez, 2003b] – M. Lopez Martinez, M.G. Ortega, and F. Rubio, "An Hinf controller of the twin rotor laboratory equipment", 11th IEEE International Conference on Emerging Technologies and Factory Automation, 2003.

[Lopez, 2004] – M. Lopez Martinez, J. Diaz, M.G. Ortega and F. Rubio, "Control of a Laboratory Helicopter using Switched 2-step Feedback Linearization*", 2004 American Control Conference, Boston, Massachussets, USA, 2004.*

[Lopez, 2005] – M. Lopez Martinez, C. Vivas, and M.G. Ortega, "A multivariable nonlinear Hinf controller for a laboratory helicopter", 44th IEEE Conference on Decision and Control and European Control Conference, Seville Spain, 2005.

[Patton, 1997] – R. J. Patton, "Fault-Tolerant Control Systems: the 1997 situation". Proceedings of the IFAC Symposium: SAFEPROCESS '97, Hull, UK, 2:1033-1055, 1997

[Rahideh, 2007] – A. Rahideh and M.H. Shaheed, "Mathematical Dynamic Modelling of a Twin-Rotor Multiple Input – Multiple Output System", *ImechE Journal of Systems and Control Engineering,* 221(1):89-101, 2007.

[Rodrigues, 2007] – M. Rodrigues, D. Theilliol, S. Aberkane, and D. Sauter, “Fault Tolerant Control Design for Polytopic LPV Systems”, *International Journal of Applied Mathematics and Computer Science*, Vol. 17, no. 1, pp. 27-37, 2007.

[Shamma, 1992a] – Shamma, J.F. and J.R. Cloutier, “A linear parameter-varying approach to gain scheduled missile autopilot design”. Proceedings of American Control Conference, Chicago, IL, USA, pp. 1317-1321, 1992.

[Shamma, 1992b] – J.S. Shamma, M. Athans, “Gain Scheduling: Potential Hazards and Possible Remedies”, *IEEE Control Systems Magazine*, Vol. 12, pp. 101-107, 1992.

[Toha, 2008], S.F. Toha, M.O. Tokhi and Z. Hussain, “ANFIS Modelling of a Twin Rotor System”, *UKACC Internation Conference on Control,* 2008.

[Tóth, 2010] – R. Tóth, **Modeling and Identification of Linear Parameter-Varying Systems**, 1st ed. Springer, 2010.

[Zhang, 2008] – Y. Zhang, J. Jiang. “Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems”. *Annual Reviews in Control*, Vol. 32, no. 2*,* pp. 229-252, 2008.

APPENDIX A – TABLES OF TRMS QUANTITIES

TABLE OF DC MOTOR QUANTITIES AND PARAMETERS

Symbol	Description	Unit
i_a	Armature current of a DC motor	A
i_{ah}	Armature current of tail DC motor	A
i_{av}	Armature current of main DC motor	A
R_a	Armature resistance of a DC motor	Ω
R_{ah}	Armature resistance of tail DC motor	Ω
R_{av}	Armature resistance of main DC motor	Ω
L_a	Armature inductance of a DC motor	H
L_{ah}	Armature inductance of tail DC motor	H
L_{av}	Armature inductance of main DC motor	H
k_a	Motor constant of a DC motor	
k_{ah}	Motor constant of tail DC motor	
k_{av}	Motor constant of main DC motor	
ω_h	Angular speed of tail rotor	V
ω_v	Angular speed of main rotor	V
k_1	Positive constant of tail motor	
k_2	Positive constant of main motor	
u_h	Horizontal voltage control input	V
u_v	Vertical voltage control input	V
B_{tr}	Viscous friction coefficient of the tail DC motor	Ω^{-1}
B_{mr}	Viscous friction coefficient of the main DC motor	Ω^{-1}
$k_{thp/n}$	Drag friction coefficients of the tail DC motor	$V^{-1}\Omega^{-1}$
$k_{tvp/n}$	Drag friction coefficients of the main DC motor	$V^{-1}\Omega^{-1}$

TABLE OF LENGTHS AND MASSES OF THE TRMS

Symbol	Description	Unit
l_t	Length of the tail part of the beam	m
l_m	Length of the main part of the beam	m
l_b	Length of the counterweight beam	m
l_{cb}	Distance between the counterweight and the joint	m
m_t	Mass of the tail part of the beam	kg
m_{tr}	Mass of the tail DC motor	kg
m_{ts}	Mass of the tail shield	kg
m_m	Mass of the main part of the beam	kg
m_{mr}	Mass of the main DC motor	kg
m_{ms}	Mass of the main shield	kg

m_b	Mass of the counterweight beam	kg
m_{cb}	Mass of the counterweight	kg
r_{ms}	Radius of the main shield	m
r_{ts}	Radius of the tail shield	m
g	Gravitational acceleration	m/s^2

TABLE OF THE QUANTITIES OF THE TRMS MECHANICS AND AERODYNAMICS

Symbol	Description	Unit
J_{tr}	Moment of inertia in the tail DC motor	F
J_{mr}	Moment of inertia in the main DC motor	F
J_h	Horizontal moment of inertia	$kg\, m^2$
J_v	Vertical moment of inertia	$kg\, m^2$
k_{fvp}	Positive constant	$kg\, m\, s^{-2}\, V^{-2}$
k_{fvn}	Positive constant	$kg\, m\, s^{-2}\, V^{-2}$
k_{fhp}	Positive constant	$kg\, m\, s^{-2}\, V^{-2}$
k_{fhn}	Positive constant	$kg\, m\, s^{-2}\, V^{-2}$
k_{chp}	Positive constant	$kg\, m^2\, s^{-2}$
k_{chn}	Positive constant	$kg\, m^2\, s^{-2}$
k_{oh}	Viscous friction coefficient, positive constant	$kg\, m^2\, s^{-1}$
k_{ov}	Viscous friction coefficient, positive constant	$kg\, m^2\, s^{-1}$
k_m	Positive constant	$kg\, m^2\, s^{-1} V^{-1}$
k_t	Positive constant	$kg\, m^2\, s^{-1} V^{-1}$
k_g	Gyroscopic constant	m s
τ_g	Gravitational torque	Nm
τ_c	Centripetal torque	Nm
$\tau_{f,v}$	Friction torque from bearings, etc. in vertical plane	Nm
θ_v	Angular position of the TRMS in the vertical plane	rad
θ_h	Angular position of the TRMS in the horizontal plane	rad
Ω_v	Angular velocity of the TRMS in the vertical plane	rad/s
Ω_h	Angular velocity of the TRMS in the horizontal plane	rad/s
f_1	Drag friction non-linear function of the tail propeller	
f_2	Non-linear function of the aerodynamic force from the tail rotor	N
f_3	Torque of the flat cable force	Nm
f_4	Drag friction non-linear function of the main propeller	
f_5	Non-linear function of the aerodynamic force from the main rotor	N

APPENDIX B – LINEAR MATRIX INEQUALITIES

MATRIX INEQUALITIES

A square matrix $\mathbf{X}$ is *Hermitian* if $\mathbf{X} = \mathbf{X}^T$. Let $\mathbf{F}$ be a Hermitian matrix and $\mathbf{h} \in \mathbb{C}^n$, the *quadratic form* is defined as:

$$|\mathbf{h}|_F^2 \triangleq \mathbf{h}^H \mathbf{F} \mathbf{h} \qquad (7.1.1)$$

$\mathbf{F}$ is said to be *positive semidefinite* if:

$$|\mathbf{h}|_F^2 \geq 0 \quad \forall \mathbf{h} \qquad (7.1.2)$$

and $\mathbf{F}$ is said to be *positive definite* if:

$$\exists \varepsilon > 0 : |\mathbf{h}|_F^2 \geq \varepsilon |\mathbf{h}|^2 \quad \forall \mathbf{h} \qquad (7.1.3)$$

Negative semidefinite and *negative definite* matrices are defined similarly.

Now let $\mathbf{F}$ be a hermitian matrix function of a vector of *decision variables* $\mathbf{x}$. $\mathbf{F}(\mathbf{x}) > 0$ is called a *matrix inequality* in $\mathbf{x}$. The *feasibility set* of a matrix inequality is defined as:

$$\mathbf{X}_{feas} \triangleq \{\mathbf{x} : \mathbf{F}(\mathbf{x}) > 0\} \qquad (7.1.4)$$

and the matrix inequality is said to be *feasible* if $\mathbf{X}_{feas}$ is non-empty.

Matrix inequalities arise in many control analysis and synthesis problems. There is no general way to solve them, except when $\mathbf{F}$ depends affinely on $\mathbf{x}$. In this case, the affine matrix inequlity can be solved with convex methods and it is called a *Linear Matrix Inequality* (LMI) and fast and efficient numerical solvers are available. If a problem can be cast as a finite-dimensional LMI it can therefore be considered practically solved.

THE YALMIP TOOLBOX

YALMIP is a modelling language for advanced modelling and solution of convex and nonconvex optimization problems. It is implemented as a free toolbox for MATLAB. The language is consistent with standard MATLAB syntax, thus making it extremely simple to use for anyone familiar with the Mathworks software. It implements a large amount of modelling tricks, allowing the user to concentrate on the high-level model, while YALMIP takes care of the low-level modelling.

The modelling language supports a large number of optimization classes, such as linear, quadratic, second order cone, semidefinite, mixed integer conic, geometric, local and global polynomial, multiparametric, bilevel and robust programming.

One of the central ideas in YALMIP is to concentrate on the language and the higher level algorithms, while relying on external solvers for the actual computations. However, YALMIP also implements internal algorithms for global optimization, mixed integer programming, multiparametric programming, sum-of-squares programming and robust optimization.

A YALMIP EXAMPLE: SEMIDEFINITE PROGRAMMING

This example illustrates the definition and solution of a simple semidefinite programming problem.

Given a linear dynamic system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, the goal is to prove stability by finding a symmetric matrix $\mathbf{P}$ satisfying:

$$\begin{cases} \mathbf{P} \geq 0 \\ \mathbf{A}^T \mathbf{P} + \mathbf{P}\mathbf{A} \leq 0 \end{cases}$$

At first, a stable matrix **A** and symmetric matrix **P** have to be defined (square matrices are symmetric by default):

```
A = [-1 2 0 ; -3 -4 1 ; 0 0 -2];
P = sdpvar(3,3);
```

Later, the constraints must be defined:

```
F = [P >= 0, A'*P+P*A <= 0];
```

To avoid the zero solution or an unbounded solution, the trace of the matrix is constrained:

```
F = [F, trace(P) == 1];
```

At this point, the problem can be solved. A feasible solution is needed, so one argument is sufficient when *solvesdp* is called to solve the problem:

```
solvesdp(F);
Pfeasible = double(P);
```

For more informations about YALMIP, check **http://users.isy.liu.se/johanl/yalmip/**