

# $\mathcal{L}_1$ -based Model Following Control of an Identified Helicopter Model in Hover

**Giacomo Picardi**

giacomo.picardi@tuebingen.mpg.de

Master student, University of Pisa

Max Plank Institute for Biological Cybernetics

Tübingen, Germany

April 4, 2016



# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
1.1	MyCopter project and PAV concept . . . . .	11
1.2	A brief history of adaptive control . . . . .	13
1.3	Literature review and contribution . . . . .	14
1.4	Outline . . . . .	15
<b>2</b>	<b>Helicopter Model Description</b>	<b>17</b>
2.1	Robinson R44 Raven II helicopter . . . . .	17
2.2	Identification process . . . . .	17
2.3	Model description . . . . .	18
<b>3</b>	<b>Baseline Controller Design</b>	<b>21</b>
3.1	Goals . . . . .	21
3.1.1	PAV reference dynamics . . . . .	21
3.2	Baseline controller structure . . . . .	22
3.3	Performance of the Baseline Controller . . . . .	22
3.3.1	Uncertainties in the Augmented Helicopter . . . . .	23
<b>4</b>	<b>Adaptive Controller Design</b>	<b>29</b>
4.1	Uncertain Plant . . . . .	29
4.2	State Predictor . . . . .	30
4.3	Adaptation Law . . . . .	30
4.4	Control Law . . . . .	31
4.4.1	Choice of outputs . . . . .	31
4.4.2	Tuning of $\Gamma$ and $C(s)$ . . . . .	32
<b>5</b>	<b>Verification and validation</b>	<b>33</b>
5.1	Results . . . . .	33
<b>6</b>	<b>Conclusions and Future Work</b>	<b>41</b>
6.1	Conclusion . . . . .	41
6.2	Future Work . . . . .	42
<b>7</b>	<b>Appendix</b>	<b>43</b>
7.1	Appendix A . . . . .	43
7.2	Appendix B . . . . .	45

7.2.1	helicopter_model.m . . . . .	45
7.2.2	helicopter_model_uncertainties_ON.m . . . . .	50
7.2.3	pidCL.m . . . . .	56
7.2.4	pidCL_uncertainties_ON.m . . . . .	58
7.2.5	Adaptation_acc.m . . . . .	60
7.2.6	montecarlo_acc.m . . . . .	61
7.2.7	data_analysis_acc.m . . . . .	64
7.3	Appendix C . . . . .	72
7.3.1	Augmented helicopter . . . . .	72
7.3.2	Augmented helicopter with adaptive controller . . . . .	73
7.3.3	Adaptive Control Law . . . . .	74
7.3.4	State Predictor . . . . .	75
7.3.5	Estimation Law . . . . .	76

## **Acknowledgments**

## Abstract

The aim of this study is to augment the uncertain dynamics of the helicopter in order to resemble the dynamics of a new kind of vehicle, the so called Personal Aerial Vehicle. To achieve this goal a two step procedure is proposed. First, the helicopter model dynamics is augmented with a PID-based dynamic controller. Such controller implements a model following on the nominal helicopter model without uncertainties. Then, an  $\mathcal{L}_1$  adaptive controller is designed to restore the nominal responses of the augmented helicopter when variations in the identified parameters are considered. The performance of the adaptive controller is evaluated via Montecarlo simulations. The results show that the application of the adaptive controller to the augmented helicopter dynamics can significantly reduce the effects of uncertainty due to the identification of the helicopter model. For implementation reasons the adaptive controller was applied to a subset of the outputs of the system. However, the under actuation typical of helicopters makes the tracking of the nominal responses good also on the not directly adapted outputs.

The thesis is articulated in three main steps. First of all the nominal plant is considered, i.e. no uncertainty is considered in the identified parameters and a baseline controller is designed to track the PAV dynamics. When the uncertainty in the values of all identified parameters is considered the dynamic augmentation does not perform as desired. Then, an adaptive controller is designed to recover the nominal stability and performance. Finally the design is validated with Montecarlo simulations.

## List of Abbreviations and Acronyms

ACAH	Attitude Command Attitude Hold
GPS	Global Positioning System
IFCS	Intelligent Flight Control Program
IRAC	Integrated Resilient Aircraft Control Project
JDAM	Joint Direct Attack Munition
MRAC	Model Reference Adaptive Control
PATS	Personal Aerial Transportation Systems
PAV	Personal Air Vehicle
PID	Proportional Integral Derivative
PIO	Pilot Induced Oscillations
TRC	Translational Rate Command

## Nomenclature

$p$	angular velocity components of helicopter along fuselage x-axis, [rad/s]
$q$	angular velocity components of helicopter along fuselage y-axis, [rad/s]
$r$	angular velocity components of helicopter along fuselage z-axis, [rad/s]
$u$	translational velocity component of the helicopter along fuselage x-axis, [m/s]
$v$	translational velocity component of the helicopter along fuselage y-axis, [m/s]
$w$	translational velocity component of the helicopter along fuselage z-axis, [m/s]
$x$	state vector of helicopter model and (after 3.2) of augmented helicopter model
$y$	output vector of helicopter and augmented helicopter model
$p_{B_{aug}}$	component of uncertainty vector in the span of $B_{aug}$
$p_{B_{um}}$	component of uncertainty vector in the span of $B_{um}$
$u_{ad}$	adaptive control input
$u_{bsl}$	baseline control input
$\hat{x}$	augmented helicopter state prediction
$N$	Number of Montecarlo simulations
$P$	Lyapunov Equation solution from adaptive control law
$Q$	Lyapunov Equation constant term from adaptive control law
$A_{aug}$	state transition matrix augmented helicopter model
$A_{he}$	state transition matrix of helicopter model
$B_{aug}$	input matrix of augmented helicopter model
$B_{he}$	input matrix of helicopter model
$B_{um}$	input matrix of unmatched uncertainties of augmented helicopter model
$C(s)$	low pass filter of adaptive control law
$C_{aug}$	output transition matrix of augmented helicopter model
$C_{he}$	output transition matrix of helicopter model
$H_m(s)$	transfer function of nominal augmented helicopter model
$H_{um}(s)$	transfer function between unmatched uncertainties and output of the augmented helicopter model
$M_{lat}$	TRC reference model from lateral cyclic to lateral translational velocity
$M_{lon}$	TRC reference model from longitudinal cyclic to longitudinal translational velocity
$M_{col}$	TRC reference model from collective lever to vertical translational velocity
$M_{ped}$	ACAH reference model from pedals to yaw rate
$T_s$	sampling time, [s]
$T_{save}$	save time in Montecarlo simulations, [s]
$\phi$	Roll Euler angle, [rad]
$\theta$	Pitch Euler angle, [rad]
$\psi$	Yaw Euler angle, [rad]
$\rho$	Identified parameter vector
$\delta_{col}$	Collective pilot input, [grad]
$\delta_{lat}$	Lateral cyclic pilot input, [grad]
$\delta_{lon}$	Longitudinal cyclic pilot input, [grad]
$\delta_{ped}$	Pedal pilot input, [grad]
$\hat{\sigma}$	unmatched uncertainties estimation
$\bar{\rho}$	Identified parameter vector, nominal values
$\Gamma$	Adaptation Gain



$\Delta A$	State transition matrix perturbation of augmented helicopter model
$\Delta \rho$	Identified parameter variations vector



# Chapter 1

## Introduction

The aim of this study is to augment the uncertain dynamics of the helicopter in order to resemble the dynamics of a new kind of vehicle, the so called Personal Aerial Vehicle. To achieve this goal a two step procedure is proposed. First, the helicopter model dynamics is augmented with a PID-based dynamic controller. Such controller implements a model following on the nominal helicopter model without uncertainties. Then, an  $\mathcal{L}_1$  adaptive controller is designed to restore the nominal responses of the augmented helicopter when variations in the identified parameters are considered.

The present thesis was driven by two main motivations. On one hand the concept of PAV played a fundamental role in the definition of the control goals. On the other, recent growing interest in adaptive control applications inspired the choice of the control technique. In particular an adaptive control technique based on a reference model was applied. This introduction gives an overview of the two cornerstones of the work. Along with this, the introduction includes a literature review to understand how the present document contributes to technical progress.

### 1.1 MyCopter project and PAV concept

Nowadays ground based transportation has come to saturation. Huge jams regularly form in large metropolitan areas all over the world. Delays, fuel consumption and pollution resulting from this seriously impact the output of global economies. The delays resulting from this congestion have been estimated to cost billions of euros per year in Europe only [1]. A possible solution could be exploiting the vertical dimension moving part of the commuting traffic from the ground to the air. The European project myCopter aimed to address this problem. The main goal of such a project is well stated by its slogan: *enabling technologies for personal aerial transportation systems* (PATs). Of course such an ambitious goal can only be achieved with a structured and complex program. The issues that need to be addressed are not only merely technical. The aims of the project are revolutionary and the social and economical implications cannot be neglected.

A fundamental block of myCopter project is the definition of the personal air vehicle (PAV) concept. A PAV is a new kind of vehicle whose dynamics is ideal and represent the theoretical reference that a real vehicle should have in order to be easy and safe to fly



Figure 1.1: Logo of European project myCopter

by naive pilots. The PAV dynamics were defined by the University of Liverpool [2]. The



Figure 1.2: A PAV concept

exact tuning of the responses was achieved through to a series of experiments involving non expert pilots rating the degree of effort required to drive in a simulated environment [1]. Implementing a physical vehicle whose dynamics resemble the PAV dynamics can be done in many several ways all of them resulting in different implications. A key feature a PAV should have is the ability of landing and taking off vertically. This is very important because otherwise, with airplane-like aircrafts the ability of door-to-door transportation would get lost and ad hoc infrastructures would be needed. In the last decades a considerable number of prototypes have been proposed for personal transportation purposes but none has achieved mass production. Among the reasons for this are the costs and excessive amount of training required to get a license compared to cars. For PAVs to be massively accessible it would be necessary to reduce the cost of training to obtain a PAV license the same way it is possible to obtain a driving license today [2].

As already stated the PAV concept is purely theoretical. A step forward in the direction of the physical implementation consists in the work of Geluardi at the Max Plank Institute for Biological Cybernetics. The core of his research is the idea of using helicopters as a base to a PAV. On one hand light civil helicopters feature many properties that a PAV should have, e.g. vertical take off and landing, size and number of seats. On the other hand they are extremely difficult to control because of their unstable, non minimum phase and highly coupled behavior. The transition from an helicopter to a PAV can be performed with the design of a control system. Such a system is meant to enhance stability and to augment the



Figure 1.3: Prototypes for personal transportation. From left to right: Terrafugia, NASA Puffin, Moller Skycar

dynamics of the helicopter in order to resemble the reference developed by the University of Liverpool. More details on the PAV dynamics will be given in chapter 2 when discussing the control objectives.

## 1.2 A brief history of adaptive control

After World War II there was a great interest in the development of high performance aircraft. These aircraft operated in a very wide flight envelope, spanning a very large range of speeds and altitudes. In this very diverse conditions the parameters variations are considerable and the effect of non linearities become non negligible. The first motivation for research in the field of Adaptive Control was to design a single controller with a mechanism to regulate the values of its parameters according to variations in the aircraft parameters. In a few words it was an alternative to gain scheduling.

After some successful application, suddenly the interest in adaptive control dropped in 1967 after a flight test in which the aircraft was destroyed and the pilot killed [10]. The instability of the adaptive control system was among the causes of the tragedy.

In the last decade there was a renewed interest in adaptive control. This time one of the main motivation for researching and applying adaptive control was the recovery of stability and adequate performance in case of failures or external disturbances.

The group of Prof. Calise from the Georgia Institute of Technology focused on two main topics. On one hand he applied adaptive control to exploit the redundancies of control surfaces to handle failures and bad atmospheric conditions in the case of piloted aircraft [11]. On the other hand, within the Joint Direct Attack Munition (JDAM) project, he proved that Model Reference Adaptive Control was a viable technique to reduce the importance of an accurate modeling and thus saving time and money [12].

Boing X-45 was an important test bench for adaptive control. Wise and Lavretsky evaluate the performance of MRAC in simulation [13] and  $\mathcal{L}_1$  adaptive control was applied in simulation for the case of actuator failures [14].

The goal of the Intelligent Flight Control System Project (IFCS) was to optimize performance of aircraft both in nominal and failure conditions. During this project adaptive controllers showed a tendency to increase Pilot Induced Oscillations (PIO) [15].

One of the most recent project on adaptive control is the Integrated Resilient Aircraft

Control Project (IRAC). The main aim of the project was to investigate the applicability and compare against each other different adaptive methods and applying them to the Generic Transport Model developed at NASA.

Many progresses were made in the field of adaptive control in the last decades. However, many problems remain open at the time of writing. The biggest issue to face is the problem of certification. Up to now, much of the tuning and evaluation is performed using Montecarlo simulation. This technique is time consuming and in some cases does not guarantee anything. The development of metrics to evaluate the performance of adaptive control is a fundamental step to be made before adaptive control will be applicable on large scale.

### 1.3 Literature review and contribution

The idea of this thesis started from the conclusion of the work of Geluardi [16]. The aim of his work was applying  $H_\infty$  and  $\mu$ -synthesis control methods to an identified light civil helicopter in hover [17] in order to augment its dynamics and resemble the dynamics of a PAV. Excellent results were obtained when no uncertainties in the identified model parameters were considered. The augmented helicopter followed the PAV model in the frequency range of interest, i.e. where the model is valid ( $[0.1, 20]rad/s$ ). However, when variations with respect to identified parameters were considered the proposed control architecture did not behave well and handling qualities degraded from level 1 to level 2 and 3 of the ADS-33E-PRF standards [18].

To overcome the limitation of robust control law, adaptive control methods can be used. Adaptive controller performs an online estimation of the uncertainty and produces a control input to reduce the undesirable deviations of the uncertain system from the nominal behavior [19].

The goal of this paper is to design an adaptive control loop for compensating uncertainties of an identified civil light helicopter in hover. The dynamics of the helicopter is first augmented with a PID-based controller to resemble translational rate command (TRC) response types of a PAV. Then, an adaptive controller augments the combined dynamics of an helicopter and PIDs to reject the effects of uncertainties in the identified parameters.

Many successful applications of adaptive control can be found in the field of small scale helicopter control. In the work of Guerreiro et al. [20]  $\mathcal{L}_1$ -adaptive control theory was used to provide attitude and velocity stabilization of an autonomous small scale rotorcraft. The adaptive controller was designed to reject the effects of wind disturbances. In this paper, however, we focus on a full scale manned helicopter and the adaptive controller is designed to reject the effects of identification uncertainties. In the work of Bichlmeier et al. [21] an  $\mathcal{L}_1$ -adaptive controller was designed to augment a preexisting Proportional Integral Derivative (PID) baseline controller on a full scale manned helicopter. The aim of the adaptive controller was to maintain handling and flying qualities in situations the baseline controller was not designed for or performed poorly. The adaptive controller only compensated for uncertainties that enter the system dynamics through the control channel, i.e. *matched* uncertainties. Nevertheless the model considered in the present paper is affected by uncertainties that do not fall in the matched category, the so called *unmatched* uncertainties. Few works in literature refer to adaptive control with unmatched uncertainties. For example in [22]  $\mathcal{L}_1$ -

adaptive control is applied in the case of unmatched uncertainties to the NASA AirStar aircraft. To the best of our knowledge, however, adaptive control has never been applied to an augmented full scale helicopter model to reject the effects of unmatched uncertainties.

## 1.4 Outline

This thesis is organized as follows:

- **Chapter 2** presents the identified helicopter model. First, some specifications on the helicopter used are given. Then, a quick description of the identification method used is given. Finally, the resulting state space model is described along with the uncertainties derived from the identification process.
- In **Chapter 3** the design of the baseline dynamic controller is addressed. First, the control goals are stated. Then the baseline controller architecture is presented and the achievement of the control goals is shown. Finally, the global dynamics of the augmented system is obtained in order to show how uncertainties affect the augmented system.
- **Chapter 4** describes the adaptive control technique implemented.  $\mathcal{L}_1$  is presented very generally. Then, the theory is revised for our application and the most important design choices are explained.
- In **Chapter 5** the proposed architecture is tested against different parameter realizations. The Montecarlo study performed is described and the results are presented.
- In **Chapter 6** a discussion on the overall work is given together with the conclusion. The most relevant aspect of the work are highlighted and future possibilities discussed.
- Matlab code and simulink schemes are reported in the the Appendix.





# Chapter 2

## Helicopter Model Description

### 2.1 Robinson R44 Raven II helicopter

The helicopter model used in this thesis is an identified model of a Robinson R44 Raven II (see fig. ??) in hover. The Robinson R44 Raven II is a four-seat light helicopter produced



Figure 2.1: Robinson R44 Raven II.

by Robinson Helicopter Company. It was chosen because is affordable and has limited dimension. Those characteristic perfectly fit the idea of myCopter project of promoting a brand new transportation system. Fig. ?? shows the dimensions of the rotorcraft.

### 2.2 Identification process

The design of a dynamic controller requires a mathematical model of the system. In this work the mathematical model of hovering dynamics of the helicopter was obtained via identification techniques [17]. The identification process consists of taking measurements of inputs and outputs of the system and using such data in combination with some knowledge of the system dynamics to derive a mathematical model.

In our case the piloted inputs from cyclic stick ( $\delta_{lat}$ ,  $\delta_{lon}$ ), collective lever ( $\delta_{col}$ ) and pedals ( $\delta_{ped}$ ) were recorded by four optical sensors directly mounted on the commands. The outputs considered are taken from two Global Positioning System antennas and an Inertial Measurement Unit. Data were collected while performing several piloted frequency sweeps

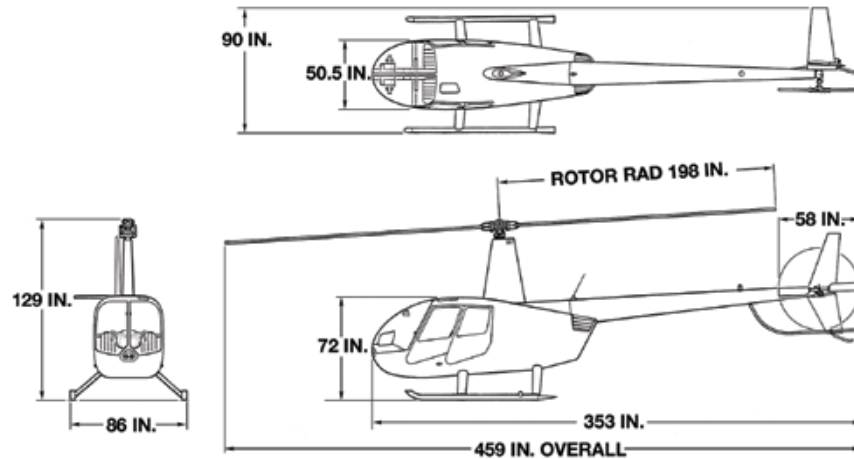


Figure 2.2: Robinson R44 Raven II, dimensions in inches.

and doublets around the hover trim condition [16]. After a first non parametric analysis a transfer function model was implemented to take into account fundamental dynamics, order of the system and level of coupling.

The identified parameters  $\rho$  of the helicopter model are not known exactly. Each identified parameter can be written in the form:

$$\rho_i = \bar{\rho}_i + \Delta\rho_i \quad (2.1)$$

where  $\bar{\rho}_i$  represents the nominal value and  $\Delta\rho_i$  is the unknown perturbation of the  $i$ -th identified parameter, with  $i = 1, 2, \dots, 28$ . The expected standard deviation of  $\Delta\rho_i$  are defined by Cramer-Rao bounds (CR) computed during the identification. Specifically,  $\Delta\rho_i \in [-3CR_i, +3CR_i]$  with 99% of probability [24].

## 2.3 Model description

The mathematical model resulting from the identification method [17] is a state space form model. This model considers pilot inputs from cyclic stick ( $\delta_{lat}$  and  $\delta_{lon}$ ), collective lever ( $\delta_{col}$ ) and tail rotor pedals ( $\delta_{ped}$ ) ???. The outputs considered are the translational velocities

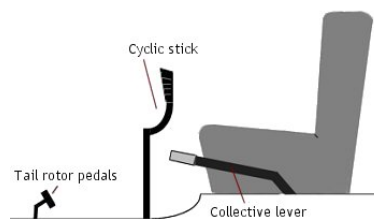


Figure 2.3: Pilot commands of a helicopter

in body frame  $u, v, w$ , the angular rates in body frame  $p, q, r$  and the roll and pitch angles

$\phi, \theta$ . The model considered is highly realistic because it takes into account rotor-fuselage coupling dynamics such as flapping, inflow and lead lag. The resulting dynamics is described in eq. 2.2.

$$\begin{aligned}\dot{x} &= A_{he}(\bar{\rho} + \Delta\rho) + B_{he}(\bar{\rho} + \Delta\rho)u \\ y &= C_{he}x\end{aligned}\tag{2.2}$$

where  $x \in \mathbb{R}^{17}$  is the helicopter state vector,  $u \in \mathbb{R}^4$  is the input vector from the pilot and  $y \in \mathbb{R}^8$  is the output vector. The matrices  $A_{he}$ ,  $B_{he}$  and  $C_{he}$  are of appropriate dimensions. It is important to note that  $A_{he}$  and  $B_{he}$  are not exactly known because they depend on the uncertain identified parameters  $\rho$ .

The dynamics of the identified helicopter are non minimum phase, unstable and fully coupled among the four control axes.



# Chapter 3

## Baseline Controller Design

In this chapter the helicopter dynamics in eq. 2.2 are augmented with a dynamic controller in order to follow the PAV reference model. No uncertainty in the identified parameter vector is considered, i.e.  $\rho = \bar{\rho}$ .

### 3.1 Goals

The goal of the baseline controller is to augment the dynamics of the helicopter and to follow the reference model of the PAV.

The PAV reference model was defined during MyCopter Project by the University of Liverpool [23]. These reference dynamics follow the guidelines for the definition of level one handling qualities defined in the ADS-33E-PRF [18], but are modified to be more suitable for non expert pilots. The definition of the reference dynamics was done with extensive experiments with naive pilots in a simulator. The study showed that non pilot performed better when Translational Rate Command (TRC) response type are used for the lateral, longitudinal and vertical translational degrees of freedom. A Rate Command Attitude Hold (RCAH) response type is implemented in yaw.

#### 3.1.1 PAV reference dynamics

The PAV reference dynamics is expressed in terms of transfer functions. Following the guidelines of ADS-33E-PRF in order to ensure handling qualities of level 1 the augmented helicopter must respond like a first order system to a step command with appropriate rising time. The corresponding transfer functions are:

$$\begin{aligned} M_{lat} &= \frac{v_{ref}}{\delta_{lat}} = \frac{1.4}{1.25s+1}, & \left[ \frac{m/s}{deg} \right] \\ M_{lon} &= \frac{u_{ref}}{\delta_{lon}} = \frac{1.4}{0.7s+1}, & \left[ \frac{m/s}{deg} \right] \\ M_{col} &= \frac{w_{ref}}{\delta_{col}} = \frac{0.502}{0.25s+1}, & \left[ \frac{m/s}{deg} \right] \\ M_{ped} &= \frac{r_{ref}}{\delta_{ped}} = \frac{1}{0.82s+1}, & \left[ \frac{rad/s}{deg} \right] \end{aligned} \tag{3.1}$$

## 3.2 Baseline controller structure

A model following approach was implemented to achieve response types selected as reference dynamics for each axis (Model reference dynamics [M]) (see eq.3.1).

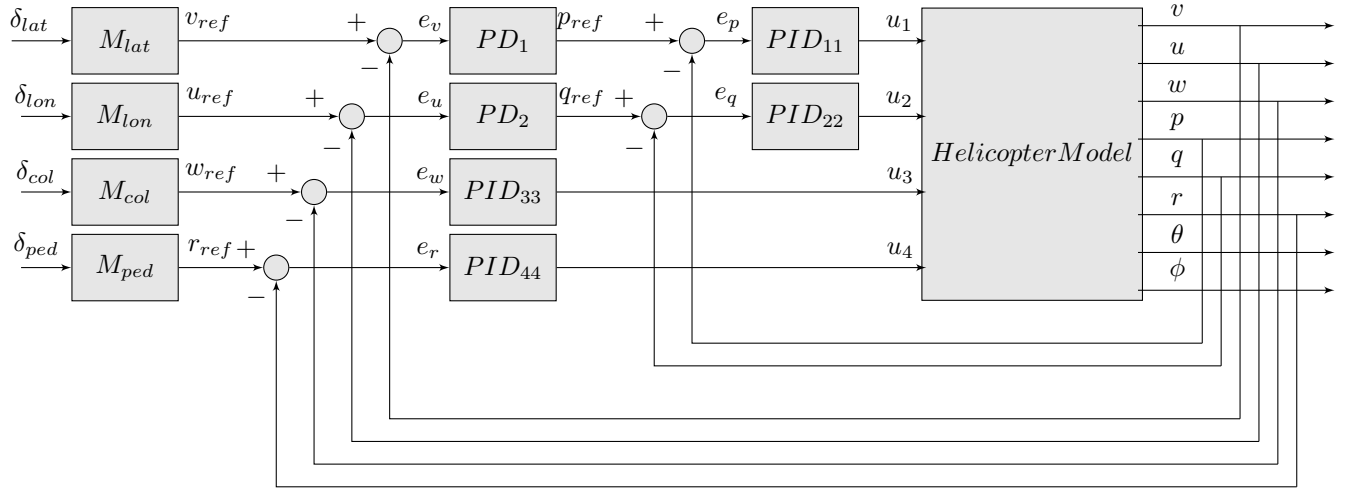


Figure 3.1: Scheme of the helicopter augmented with PID-based controller and reference dynamics

To achieve handling qualities requirements of PAV, the helicopter dynamics was augmented with a PID-based controller as shown in figure ???. A multi loop controller architecture is implemented on the lateral and longitudinal axes. On the outer loop two PD controllers provide roll and pitch rate references ( $p_{ref}$ ),  $q_{ref}$  expressed in terms of the error on lateral and longitudinal velocities ( $e_v$ ,  $e_u$ ). The inner loop tracks the angular rate reference and stabilizes the dynamics.

A single loop controller architecture is implemented on the vertical and yaw axis.  $PID_{33}$  and  $PID_{44}$  define the inputs of the helicopter in terms of the error on vertical velocity and yaw rate ( $e_w$ ,  $e_r$ ), respectively. Tuning of the baseline controller was done to minimize the difference between the augmented helicopter dynamics and the model reference over the  $PID$ s parameters in the range of frequencies where the identified model is valid ( $[0.1, 20]$  rad/s). However, the proportional gain of  $PID_{44}$  was manually incremented to better decouple vertical velocity and yaw rate.

## 3.3 Performance of the Baseline Controller

The presented control architecture achieves adequate model tracking in the frequency range of interest as shown in fig. 3.2a-3.2d.

To investigate whether the resulting augmented dynamic system satisfies handling qualities requirements of ADS-33E-PRF standard [18], responses to a step command must be considered.

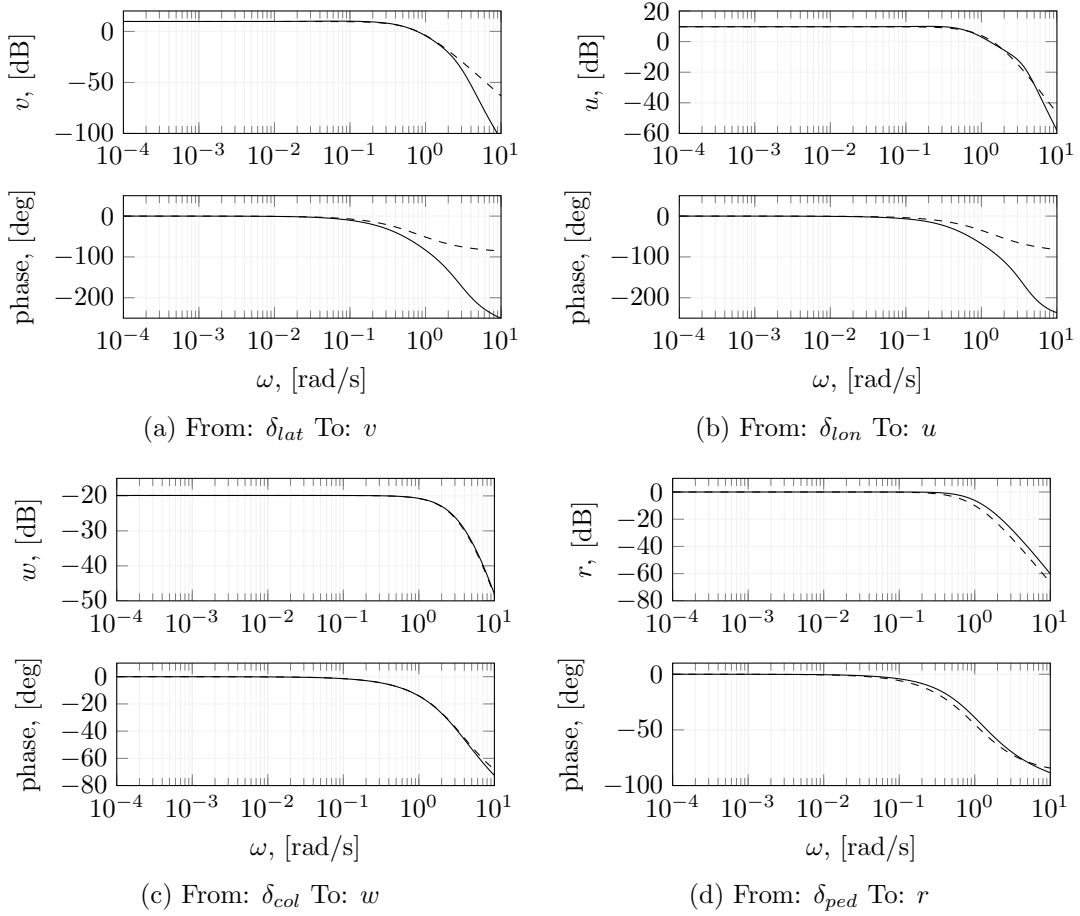


Figure 3.2: Bode plot comparison. Dashed: Reference Model, Solid: Augmented Helicopter.

Figures 3.3a-3.3d show that the augmented helicopter behaves approximately like a first order system, with rising time very close to the reference model in every channel and negligible overshoot in response 3.3c. Therefore the ADS-33E-Prf requirements are satisfied. However, the responses  $v$  to  $\delta_{lat}$  and  $u$  to  $\delta_{lon}$  of the augmented helicopter present a delay due to the non minimum phase of the helicopter dynamics.

### 3.3.1 Uncertainties in the Augmented Helicopter

After the dynamic augmentation with PID controllers and reference dynamics  $M$ , all uncertainties only affect the state transition matrix of the system. The proof can be found in Appendix A. The resulting augmented system can be written in the form:

$$\begin{aligned} \dot{x}_{aug} &= A_{aug}(\bar{\rho} + \Delta\rho)x_{aug} + B_{aug}u_{aug} \\ y &= C_{aug}x_{aug} \end{aligned} \quad (3.2)$$

where  $x_{aug} \in \mathbb{R}^{31}$ ,  $u_{aug} = [v_{cmd}, u_{cmd}, w_{cmd}, r_{cmd}] \in \mathbb{R}^4$  and  $y \in \mathbb{R}^8$ ,  $A_{aug} \in \mathbb{R}^{31 \times 31}$ ,  $B_{aug} \in \mathbb{R}^{31 \times 4}$ ,  $C_{aug} \in \mathbb{R}^{8 \times 31}$ . From now on we will refer to the state and input vectors of the augmented helicopter as  $x$  and  $u$ , respectively, to reduce the use of subscripts.

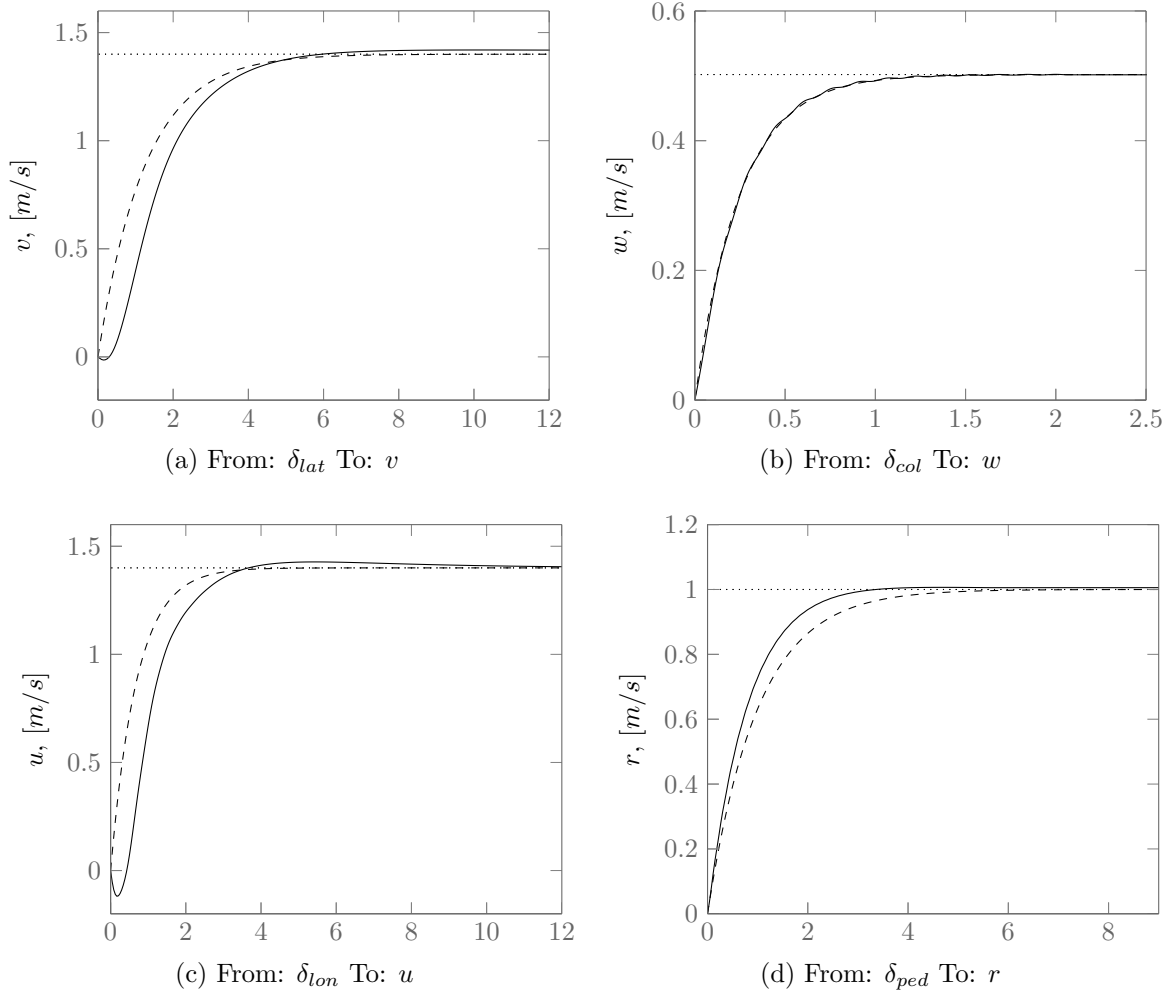


Figure 3.3: Step responses comparison. Dashed: Reference Model, Solid: Augmented Helicopter.

To apply the adaptive control technique presented in the following chapter, the system in 3.2 must be rewritten in a convenient form. First, the nominal dynamics are separated from the perturbation as follows:

$$\begin{aligned} \dot{x} &= (A_{aug}(\bar{\rho}) + \Delta A(\bar{\rho} + \Delta\rho))x + B_{aug}u \\ y &= C_{aug}x \end{aligned} \quad (3.3)$$

where  $\Delta A$  is unknown and accounts for the uncertainty due to  $\Delta\rho$ . Then, the vector  $\Delta Ax$  can be written as the sum of two vectors: the component of  $\Delta Ax$  along the span of  $B_{aug}$  (matched uncertainties) and along its orthogonal complement  $B_{um}$  (unmatched uncertainties). Here  $B_{um} \in \mathbb{R}^{31 \times 27}$  is a constant matrix such that  $B_{aug}^T B_{um} = 0$  and  $[B_{aug} \ B_{um}]$  has full rank. The two components can be expressed as:

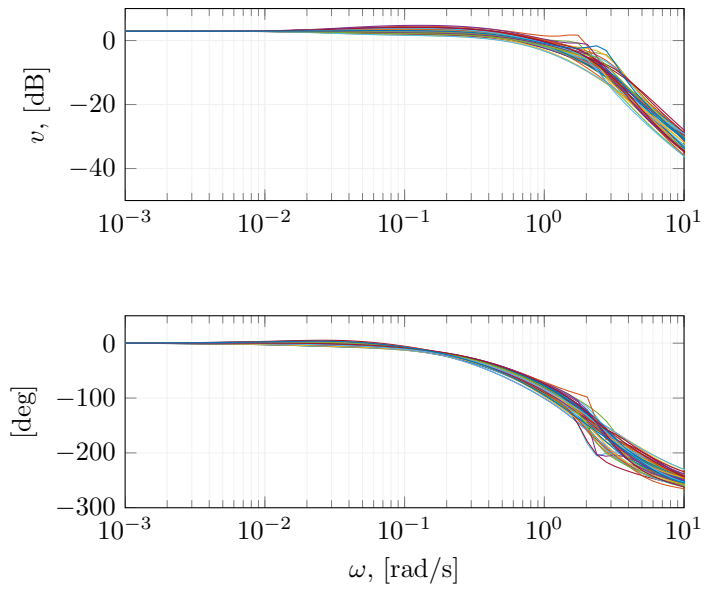
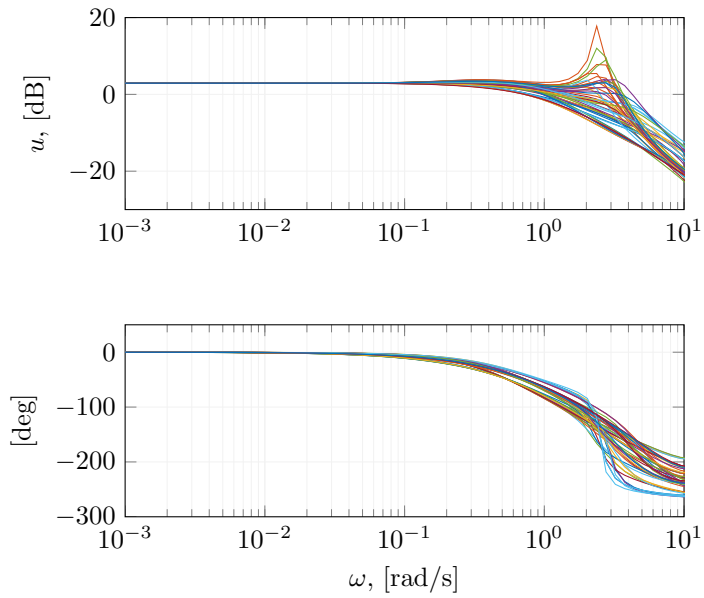
$$\begin{aligned} p_{B_{aug}} &= B_{aug} B_{aug}^+ \Delta Ax, \\ p_{B_{um}} &= B_{um} B_{um}^+ \Delta Ax. \end{aligned} \quad (3.4)$$

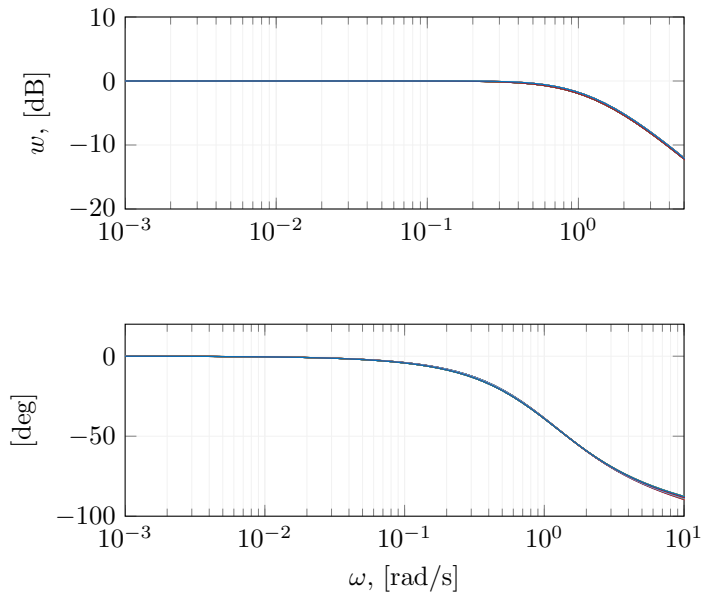
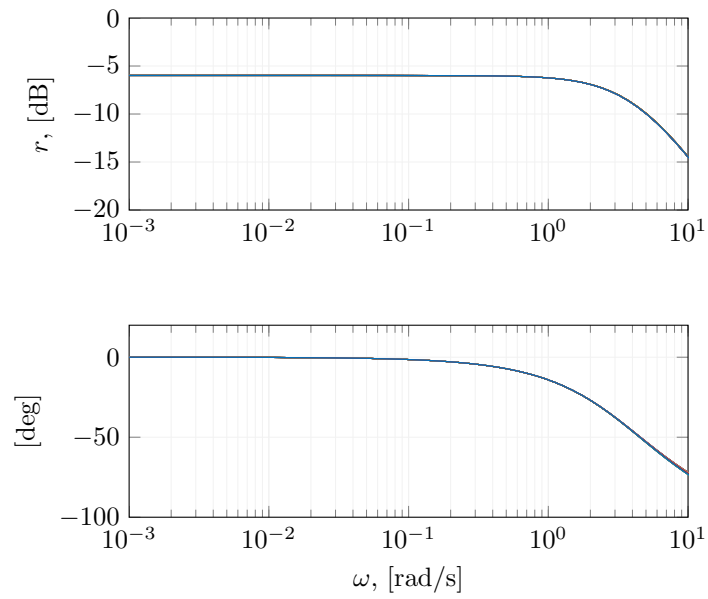


where superscript  $+$  indicates the Moore-Penrose pseudoinverse. As proven in Appendix A, in our case the component of  $\Delta A$  along  $\text{span}(B_{aug})$  is null. Thus, the resulting form of the augmented helicopter dynamics 3.2 is:

$$\begin{aligned}\dot{x} &= A_{aug}x + B_{aug}u + p_{B_{um}} \\ y &= C_{aug}x\end{aligned}\tag{3.5}$$

When the considered unmatched uncertainties are included, the baseline controller presented in Figure ?? fails to track the reference model. Figures ?? - ?? show Bode diagrams of  $(u, v, w, r)$  from  $(\delta_{lat}, \delta_{lon}, \delta_{col}, \delta_{ped})$  respectively for 50 random parameter variations. Clearly the responses are different for different uncertain parameters realizations. In particular the uncertainty is larger when an input on the cyclic is considered. This makes the handling qualities of the augmented helicopter degrade. To overcome this limitation, an  $\mathcal{L}_1$  adaptive controller was designed.

(a) From:  $\delta_{lat}$  To:  $v$ (b) From:  $\delta_{lon}$  To:  $u$

(a) From:  $\delta_{col}$  To:  $w$ (b) From:  $\delta_{ped}$  To:  $r$



# Chapter 4

## Adaptive Controller Design

This section gives an overview of  $\mathcal{L}_1$ -control theory. The contents of [25] are revised for our implementation. The implemented adaptive control architecture is shown in fig. 4.1.

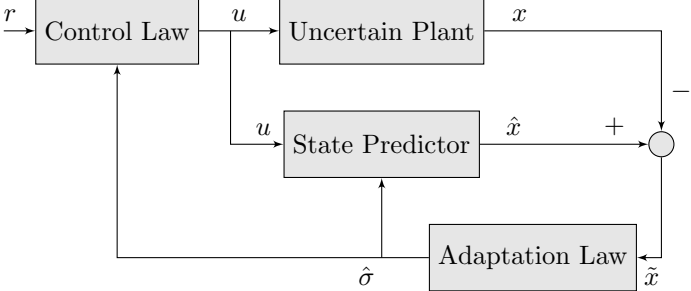


Figure 4.1: Adaptive control architecture

State predictor and adaptation law provide an estimation of the state of the augmented helicopter  $\hat{x}$  and an estimation of the uncertainties  $\hat{\sigma}$ , respectively. Then, the control law block creates a control signal based on  $\hat{\sigma}$  to compensate for the effect of uncertainties in the system dynamics. Each block will be described in a subsection along with the faced implementation issues.

### 4.1 Uncertain Plant

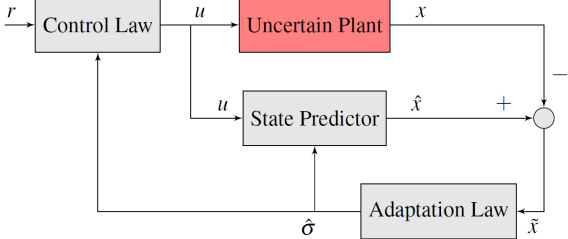


Figure 4.2: The uncertain plant block contains the augmented helicopter dynamics

Many applications of adaptive control in literature place the adaptive controller right around the uncertain element (the helicopter model). Nevertheless, when the compensation of unmatched uncertainties is required, adaptive controller must be applied to a stable systems [25]. Since the open loop helicopter model is unstable, our choice was to implement the adaptive controller to the stable augmented helicopter model.

## 4.2 State Predictor

The *State Predictor* block produces an estimation of the augmented helicopter state  $\hat{x}$  based on the control input of the augmented helicopter  $u$  and the estimation of the unmatched uncertainty  $\hat{\sigma}$ . The prediction of the augmented helicopter state is calculated with the following equation:

$$\dot{\hat{x}}(t) = A_{aug}\hat{x}(t) + B_{aug}u(t) + B_{um}\hat{\sigma}(t) \quad (4.1)$$

where  $\hat{\sigma}(t) \in \mathbb{R}^{n-m}$  is the estimation of the unmatched uncertainties computed in the *Adaptation Law* block. Note that the equations of the state predictor 4.1 and of the uncertain plant 3.5 have the same form. The only difference between the two is that the state predictor dynamics includes the estimation of the uncertainties instead of the actual uncertainties.

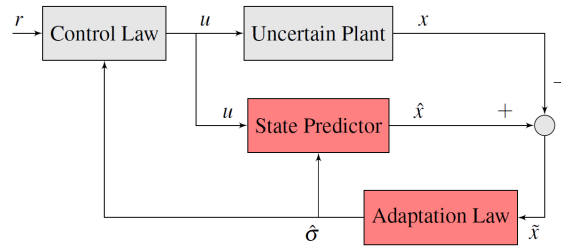


Figure 4.3: State Predictor and Adaptation Law block work in a loop to provide unmatched uncertainties estimation

## 4.3 Adaptation Law

The *Adaptation Law* block provides an estimation of the unmatched uncertainties based on the difference between the state of the augmented helicopter  $x$  and its prediction  $\hat{x}$ . The estimation of unmatched uncertainties  $\hat{\sigma}$  is computed as follows:

$$\begin{aligned} \hat{\sigma}(t) &\triangleq \hat{\sigma}_1(t)\|x(t)\|_{L_\infty} + \hat{\sigma}_2(t), \\ \dot{\hat{\sigma}}_1(t) &= -\Gamma(\tilde{x}^T(t)PB_{um})^T\|x(t)\|_{L_\infty}, \\ \dot{\hat{\sigma}}_2(t) &= -\Gamma(\tilde{x}^T(t)PB_{um})^T. \end{aligned} \quad (4.2)$$

where  $\tilde{x} \triangleq \hat{x} - x$  is the error between the predicted state and the state of the plant.  $P \in \mathbb{R}^{n \times n}$  is a symmetric, positive definite matrix that solves the algebraic Lyapunov equation  $A_{aug}^T P + P A_{aug} = -Q$ , with  $Q \in \mathbb{R}^{n \times n}$  definite positive. The choice of parameter  $P$  is related to the proof of stability of the implemented control architecture [25]. The design parameter

$\Gamma \in \mathbb{R}^+$  is the adaptive gain. The tuning of  $\Gamma$  is critic for the implemented adaptive controller and will be discussed later in this section.

## 4.4 Control Law

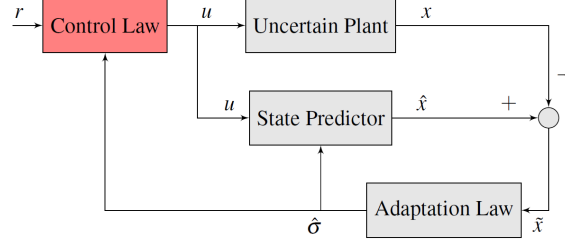


Figure 4.4: Control Law block provides the input for the augmented helicopter

The *Control Law* block produces a control signal that compensates for the effect of unmatched uncertainties using the unmatched uncertainties estimation  $\hat{\sigma}$ . The estimated effect of unmatched uncertainties on the output of the augmented helicopter can be calculated as  $H_{um}(s)\hat{\sigma}$ , where:

$$H_{um}(s) \triangleq C_{aug}(s\mathbb{I}_n - A_{aug})B_{um} \in \mathbb{C}^{4 \times 27} \quad (4.3)$$

Eq. 4.3 is the transfer function from unmatched uncertainties to output. The input that cancels such effect is calculated as  $H_m^{-1}(s)H_{um}(s)\hat{\sigma}(s)$  where  $H_m(s)$  is the transfer function that describes the nominal behavior of the augmented helicopter:

$$H_m(s) \triangleq C_{aug}(s\mathbb{I}_n - A_{aug})B_{aug} \in \mathbb{C}^{4 \times 4} \quad (4.4)$$

Finally, the control signal is obtained by adding the term for unmatched uncertainties compensation to the pilot input and by applying a low pass filter  $C(s)$  as:

$$u(s) = -C(s)(H_m^{-1}(s)H_{um}(s)\hat{\sigma}(s) - r(s)) \quad (4.5)$$

In eq. 4.5,  $r(s)$  is the Laplace transform of the pilot input and  $C(s) \in \mathbb{C}^{4 \times 4}$  is a proper and stable low-pass filter with unit steady state gain.

To implement the control law in eq. 4.5,  $H_{um}(s)$  (eq. 4.3) must be stable and  $H_m(s)$  (eq. 4.4) must be invertible and have a stable inverse. Since the adaptive controller designed in this paper is applied to the augmented helicopter, the term 4.3 is stable. To achieve  $H_m(s)$  invertible and with stable inverse we had to select 4 of the 8 helicopter outputs such that  $H_m(s)$  is minimum phase (i.e. all zeros  $\in \mathbb{C}^-$ ).

### 4.4.1 Choice of outputs

The outputs of the helicopter model can be divided into 4 groups:

- $u, q, \theta$  refer to the longitudinal axis;

- $v, p, \phi$  refer to the lateral axis;
- $w$  refers to the vertical axis;
- $r$  refers to the yaw axis.

The variables in each group strongly depend on each other because the helicopter is under-actuated. Therefore, selecting four outputs with one variable per control axis might improve the tracking of all variables. Based on these considerations the outputs selected were  $w, r, \theta, \phi$ .

#### 4.4.2 Tuning of $\Gamma$ and $C(s)$

Finally, the choice of the value of the adaptation gain  $\Gamma$  and the design of the low-pass filter  $C(s)$  were addressed. Large values of  $\Gamma$  make the uncertainties estimation fast but introduce high frequency components in the control signal that may lead to instability. Low values of  $\Gamma$  produce a slower, yet less noisy estimation. The filter  $C(s)$  is commonly designed as a low-pass filter that reduces the destabilizing effect of high frequency components that derive from high values of  $\Gamma$ , thus allowing for a fast estimation. However, in literature no procedure exists to design the filter  $C(s)$  and the adaptation gain  $\Gamma$  in the case of multi-input multi-output system with unmatched uncertainties [26]. Our approach was to adjust the bandwidth of  $C(s)$  to cut off the high frequencies introduced by the estimation, while maintaining the fast augmented helicopter dynamics needed for compensating the uncertainties. Then,  $\Gamma$  was tuned by starting from an arbitrarily high value and progressively reducing it until the resulting design was stable. The approach based on these considerations led to the following choice of parameters.

Parameter	Value
$\Gamma$	$10^5$
$Q$	$\mathbb{I}_{31}$
$C(s)$	$\frac{1}{(1+\frac{s}{50})^4}$

Table 4.1: Parameter selection



# Chapter 5

## Verification and validation

The performance of the proposed control architecture were validated against different realizations of the uncertain parameters vector  $\rho$  via Montecarlo simulation. First, a doublet

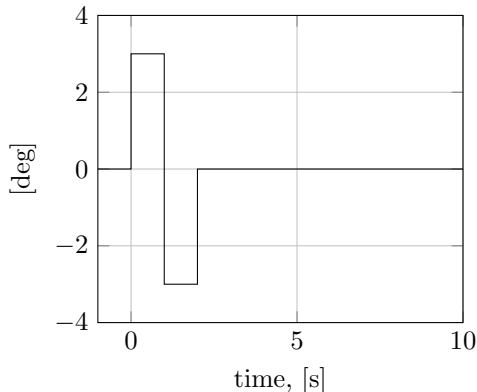


Figure 5.1: Doublet

signal as shown in fig. 5.1 was applied to each input of the nominal augmented helicopter without uncertainties. Then, the responses of the uncertain system were evaluated with the adaptive controller switched on and off. The test shown in the following of this section uses 500 random variation on all identified parameters. Recalling the definition of uncertain parameters in eq. 2.1,  $\Delta\rho_i$  was randomly selected as  $\pm CR_i$ , i.e. the maximum or minimum possible value with 99% of probability [27].

### 5.1 Results

In this section the results of the Montecarlo study are shown and commented. The responses of all the outputs of the augmented helicopter to a doublet in each input channel are shown. For every output the nominal response of the augmented helicopter without parametric uncertainties are plotted together with the response of the system when the adaptive controller is switched on and off. For every input plots are divided into two groups. First,  $w$ ,  $r$ ,  $\theta$  and  $\phi$  are shown, then the remaining output. This division reflects the division into directly and indirectly adapted output.

Figures 5.2a-5.2h shows the responses of directly adapted output to a double on  $\delta_{lat}$ . In the mean case the adaptive controller improves the tracking of nominal responses and significantly reduces the standard deviation of the responses with respect to identified parameter variations.

Very similar results are obtained when non directly adapted outputs are considered. Figures 5.3a-5.3h show the responses of  $v$ ,  $u$ ,  $p$ ,  $q$ . Thanks to the under actuation typical of helicopters the adaptive controller achieves good tracking of the responses of the nominal system and reduces the standard deviation at every time.

Figures 5.4a-5.4h shows the responses of directly adapted output to a double on  $\delta_{lon}$ . The results are similar to the previous case. In the mean case the adaptive controller improves the tracking of nominal responses and significantly reduces the standard deviation of the responses with respect to identified parameter variations.

Very similar results are obtained when non directly adapted outputs are considered. Figures 5.5a-5.5h show the responses of  $v$ ,  $u$ ,  $p$ ,  $q$ . Thanks to the under actuation typical of helicopters the adaptive controller achieves good tracking of the responses of the nominal system and reduces the standard deviation at every time.

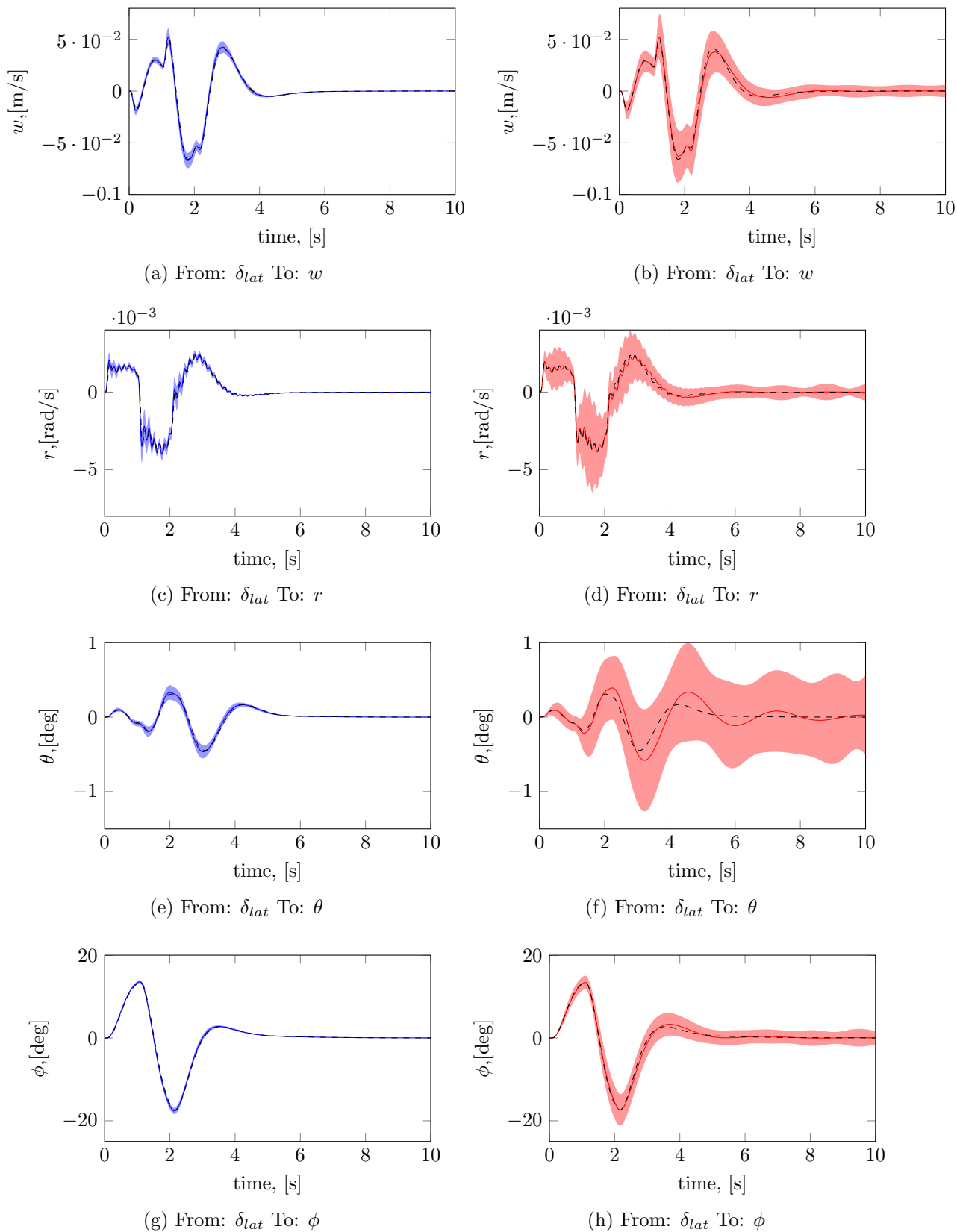
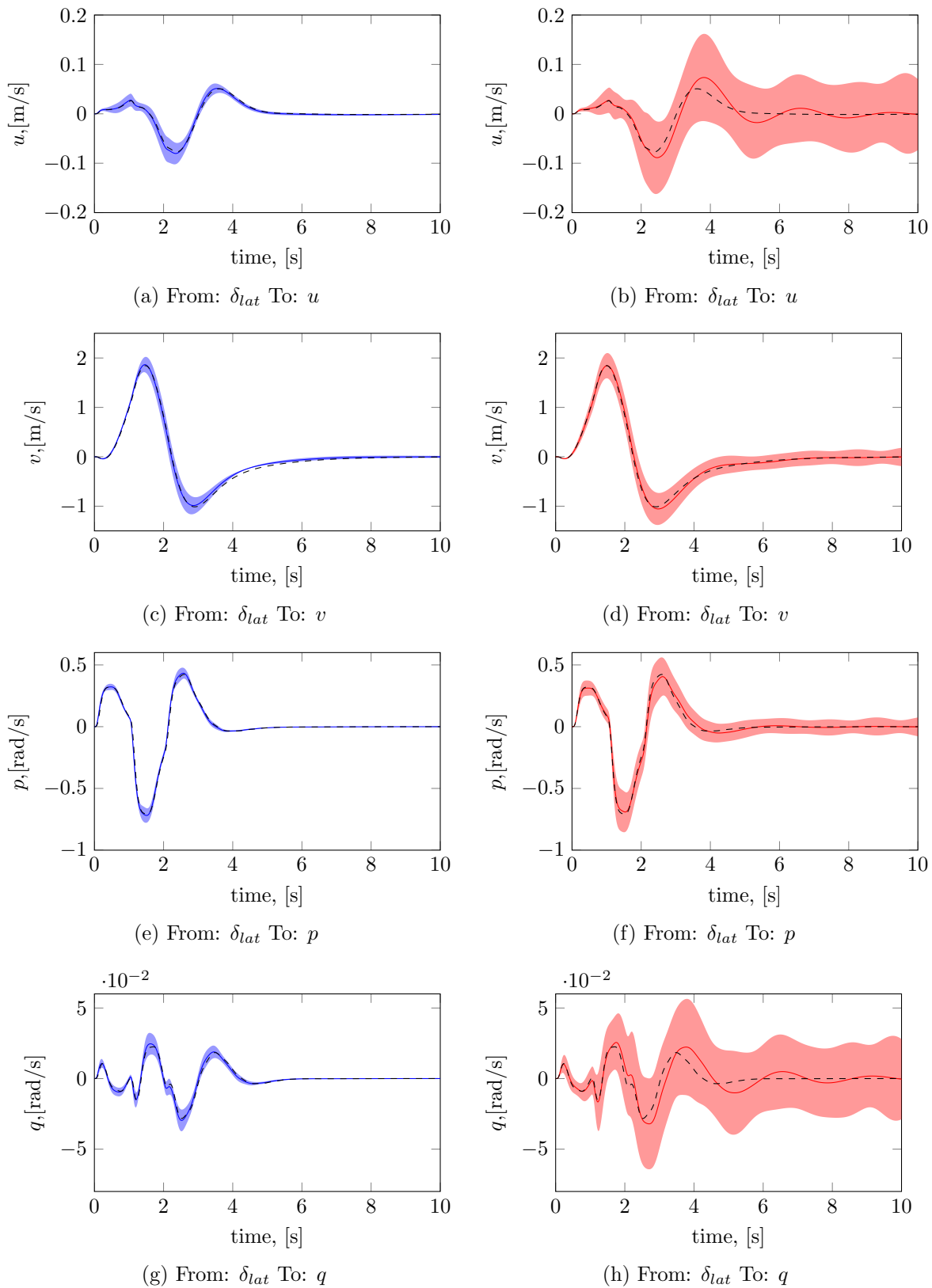


Figure 5.2: Adaptive ON/OFF comparison. Dashed black: Nominal response, Solid blue: Adaptive ON mean, Shadow blue: Adaptive ON SD, Solid red: Adaptive OFF mean, Shadow red: Adaptive OFF SD.



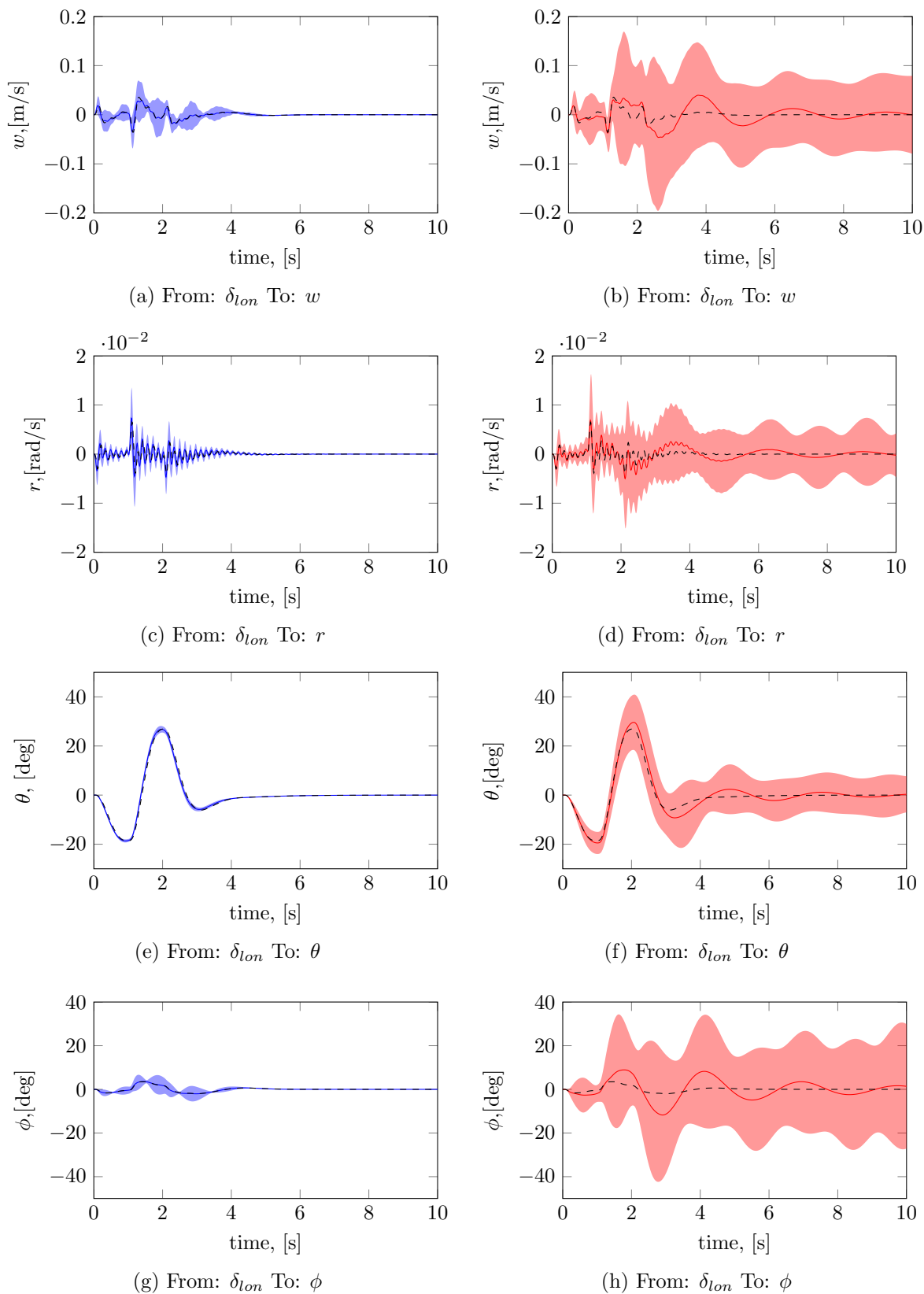


Figure 5.4: Adaptive ON/OFF comparison. Dashed black: Nominal response, Solid blue: Adaptive ON mean, Shadow blue: Adaptive ON SD, Solid red: Adaptive OFF mean, Shadow red: Adaptive OFF SD.

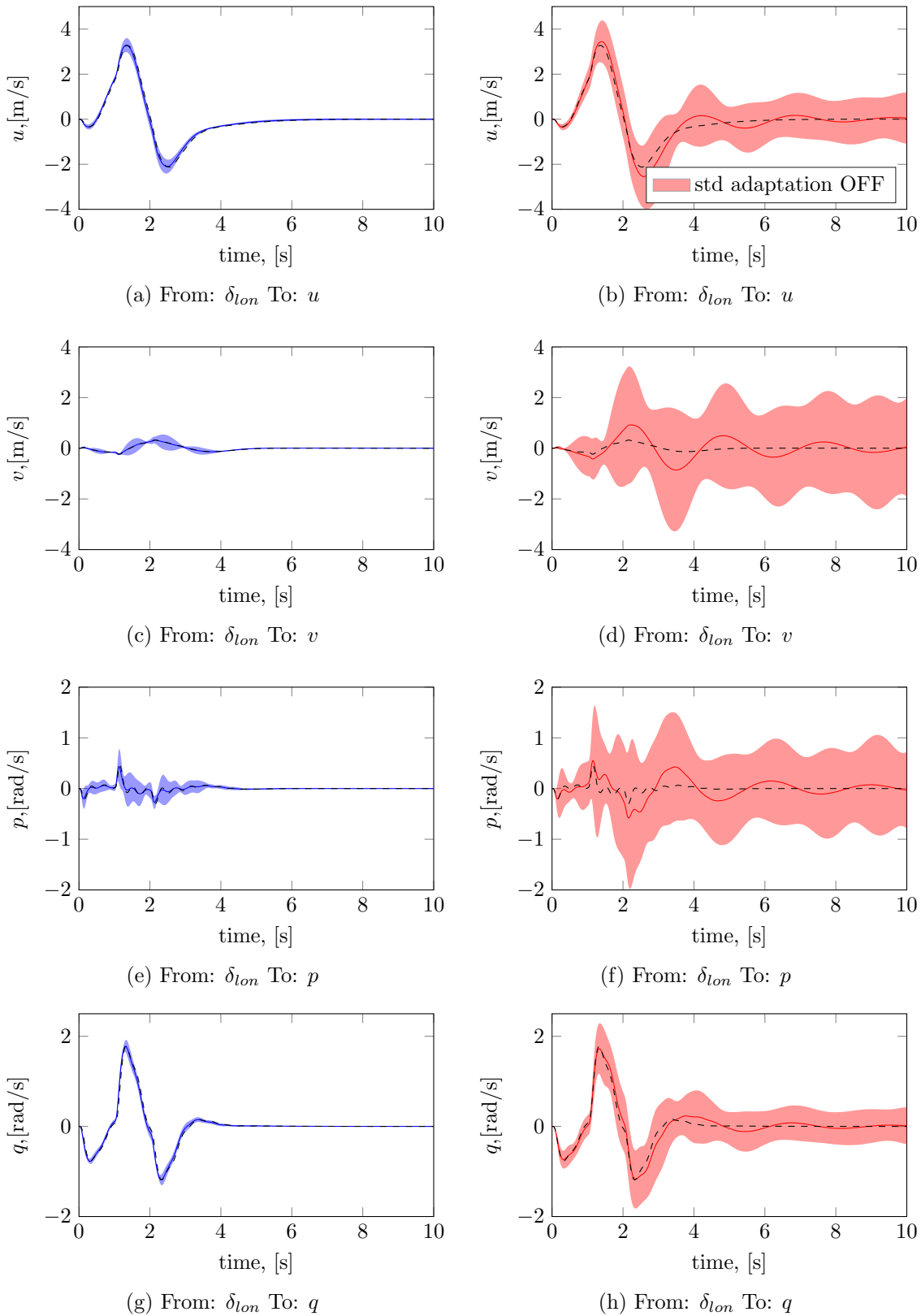


Figure 5.5: Adaptive ON/OFF comparison. Dashed black: Nominal response, Solid blue: Adaptive ON mean, Shadow blue: Adaptive ON SD, Solid red: Adaptive OFF mean, Shadow red: Adaptive OFF SD.

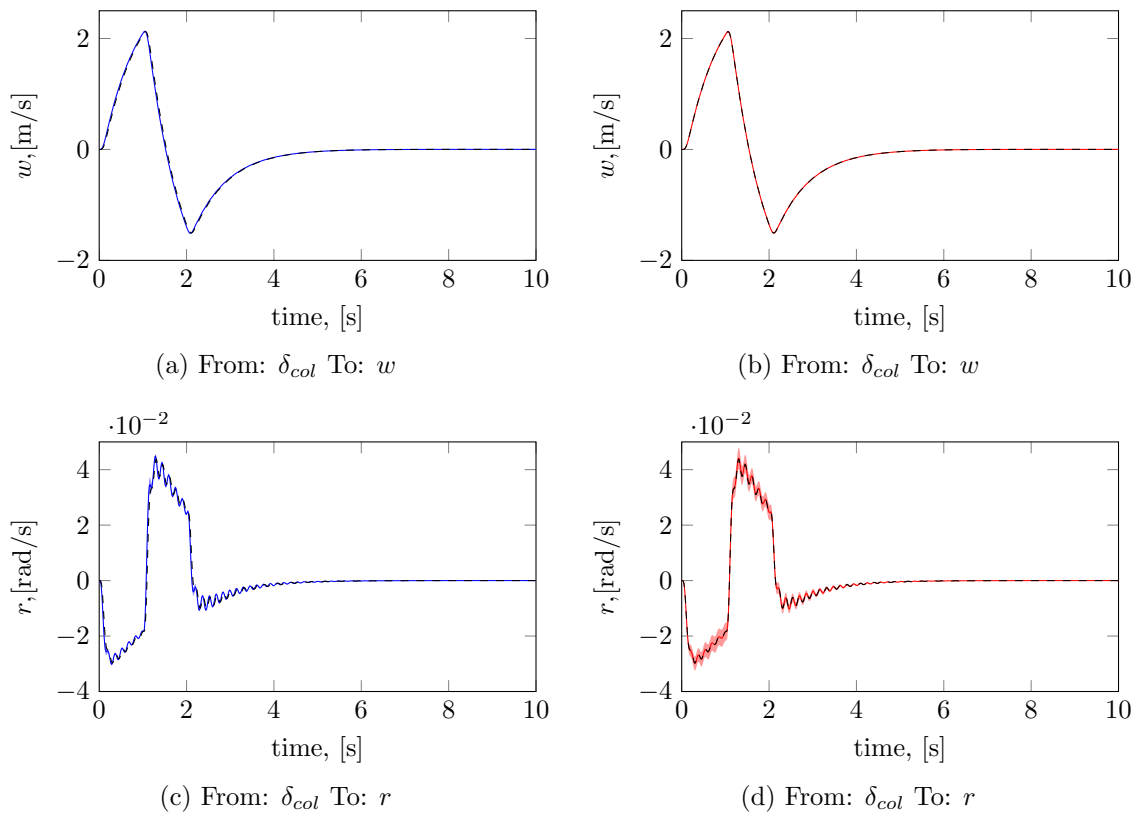


Figure 5.6: Adaptive ON/OFF comparison. Dashed black: Nominal response, Solid blue: Adaptive ON mean, Shadow blue: Adaptive ON SD, Solid red: Adaptive OFF mean, Shadow red: Adaptive OFF SD.

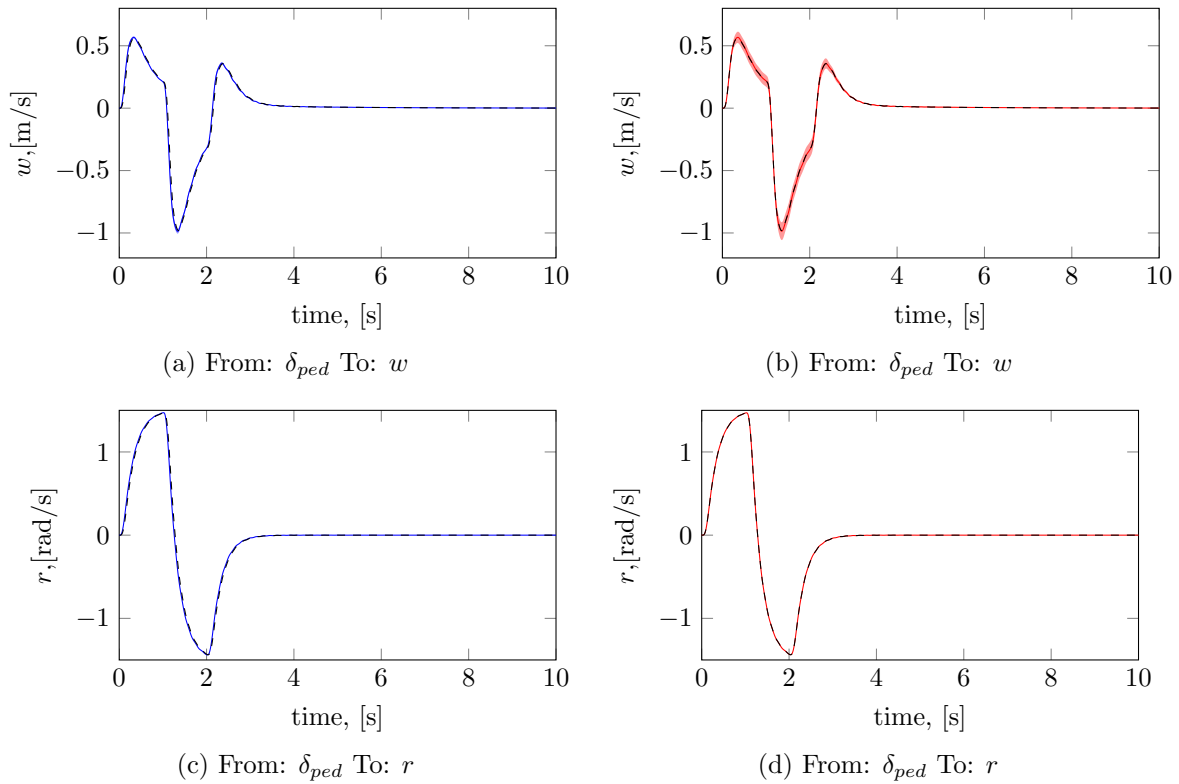


Figure 5.7: Adaptive ON/OFF comparison. Dashed black: Nominal response, Solid blue: Adaptive ON mean, Shadow blue: Adaptive ON SD, Solid red: Adaptive OFF mean, Shadow red: Adaptive OFF SD.



# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusion

The study described in this thesis consisted in a two step solution of the problem of augmenting the dynamics of a civil light helicopter model in hover in order to resemble the reference dynamic of a PAV.

The PAV reference dynamics is defined in terms of input-output behavior. It is based on the handling qualities ADS-33E-PRF standard but was tuned for naive pilots.

The first step of the proposed solution is the design of a dynamic augmentation with a PID-based controller. A classical architecture has been used to implement a model following. The tuning of the parameters of the PIDs was done minimizing a cost index in terms of  $H_\infty$ -norm. The design was validated under the assumption of no uncertainty in the identified parameter of the helicopter model. The helicopter dynamics augmented with the baseline controller tracks the PAV reference model in the frequency range of interest. A further validation was done analysing the step responses to follows the ADS-33E-PRF guidelines. The system showed first order like responses with adequate rising time. However, when uncertainty in the identified parameters was taken into account the behavior of the augmented helicopter was not satisfactory any more. The same limitation was highlighted by previous works where optimal and robust control methods were applied.

The second step of the proposed solution was intended to overcome this limitation. An adaptive control loop based on  $\mathcal{L}_1$  theory was implemented.  $\mathcal{L}_1$ -control is essentially a Model Reference Adaptive Control (MRAC) where the control input is low-pass filtered. The reference model in this phase is the augmented helicopter in nominal conditions and the goal of this additional control loop is to reject the effect of parametric uncertainties. The adaptive controller was applied to a subset of the augmented helicopter outputs because the method required the system to be non minimum phase and square.

The proposed solution was validated against several parameter realizations via a Monte Carlo study. A doublet input was applied to all the piloted input and the responses of the augmented helicopter with and without the adaptive controller are compared with the responses of the nominal system.

The adaptive controller significantly improved the tracking of the nominal responses in the mean case and reduced the standard deviation of the responses. The same improvement

were shown by the outputs that are not directly applied, thanks to the under actuated nature of helicopters.

## 6.2 Future Work

This thesis is meant as a first investigation on the possibility of using adaptive control to achieve the goal of augmenting the helicopter dynamics and resemble the behavior of a PAV.

The first step forward in this investigation could be the set up of an experiment with a mobile platform simulator. For example the CyberMotion simulator from the Department of Professor Bühlhoff at the Max Plank Institute for Biological Cybernetic could be a valid candidate. Such a study could point out how the adaptive controller behaves when a naive pilot uses the augmented helicopter and how the improvements in the responses achieved with it are perceived by the experiment participants.

Another interesting development is the addition of State Observer (e.g a Kalman Filter) to estimate the state vector of the augmented helicopter based on the output from the Inertial Navigation Systems and the GPS. Testing and validation of the system with this addition are crucial to a future implementation on a real helicopter.

# Chapter 7

## Appendix

### 7.1 Appendix A

In this appendix we obtain the state space model of the system in fig. ???. The helicopter state space model is given in eq. 2.2 where in matrix  $C_{he}$  the feedback outputs  $[u, v, w, p, q, r]$  are selected. The *PID*-based controller model can be expressed in state space form as follows:

$$\begin{aligned}\dot{x}_c &= A_c x_c + B_{c1} y + B_{c2} r \\ u &= C_c x_c + D_{c1} y + D_{c2} r\end{aligned}\quad (7.1)$$

where  $x_c \in \mathbb{R}^{n_c}$  is the controller state vector,  $u$  the control input from 2.2,  $y$  is the feedback output from 2.2, and  $r \in \mathbb{R}^{n_r}$  represents the reference vector. The real matrices ( $A_c, B_{c1}, B_{c2}, C_c, D_{c1}, D_{c2}$ ) are of appropriate dimensions and describe the controller dynamics.

The state space model for the closed loop system of helicopter and *PID*-based controller is:

$$\begin{aligned}\dot{x}_{cl} = \begin{bmatrix} \dot{x}_{he} \\ \dot{x}_c \end{bmatrix} &= A_{cl} x_{cl} + B_{cl} r \\ y &= C_{cl} x_{cl} + D_{cl} r\end{aligned}\quad (7.2)$$

where matrices ( $A_{cl}, B_{cl}, C_{cl}, D_{cl}$ ) are:

$$\begin{aligned}A_{cl}(\rho) &= \begin{bmatrix} A_{he}(\rho) + B_{he}(\rho)D_{c1}C_{he} & B_p(\rho)C_{he} \\ B_{c1}C_{he} & A_c \end{bmatrix} \\ B_{cl}(\rho) &= \begin{bmatrix} B_{he}(\rho)D_{c2} \\ B_{c2} \end{bmatrix} \\ C_{cl} &= C_{he} \\ D_{cl} &= 0\end{aligned}\quad (7.3)$$

As shown in fig. ??? the reference model [M] is in cascade with the system in 7.3. Since the reference model is strictly proper, its state space form is:

$$\begin{aligned}\dot{x}_r &= A_r x_r + B_r u_r \\ r &= C_r x_r\end{aligned}\quad (7.4)$$

where  $x_r \in \mathbb{R}^{n_r}$  is the state,  $u_r \in \mathbb{R}^{m_r}$  is the external command vector and,  $r$  is the reference vector from 7.1. The real matrices ( $A_r, B_r, C_r$ ) are of appropriate dimensions and describe

the reference dynamics. Finally, the cascade of the reference model [M] and the helicopter augmented with the *PID – based* controller can be expressed in state space form as follows:

$$\begin{aligned} \dot{x} = \begin{bmatrix} \dot{x}_{cl} \\ \dot{x}_r \end{bmatrix} &= Ax + Bu_r \\ y &= Cx + Du_r \end{aligned} \quad (7.5)$$

where matrices  $(A, B, C, D)$  are:

$$\begin{aligned} A(\rho) &= \begin{bmatrix} A_{cl}(\rho) & B_{cl}(\rho)C_r \\ 0 & A_r \end{bmatrix} & B &= \begin{bmatrix} 0 \\ B_r \end{bmatrix} \\ C &= C_{cl} & D &= 0 \end{aligned} \quad (7.6)$$

From 7.6 we observe that:

- Matrix  $B$  only depends on the reference dynamics and thus is not affected by uncertainty;
- All the uncertainties lie in the upper rows of  $A$  and in the corresponding rows matrix  $B$  is null. Thus, all uncertainties are unmatched because they are not in the span of  $B$ .

## 7.2 Appendix B

The working code developed in this thesis is reported in this appendix. The following files are provided:

- **helicopter\_model.m** creates the nominal helicopter state space model.
- **helicopter\_model\_uncertainties\_ON.m** creates the uncertain helicopter state space model. It is possible to change the identified parameters values at the beginning of the file (variable *x*).
- **pidCL.m** creates the state space model of the augmented helicopter in nominal conditions. The parameters of the PID controllers are set here. Calls `helicopter_model.m` and uses the `connect` to create the state space model.
- **pidCL\_uncertainties\_ON.m** is the same as `pidCL`, but uses `helicopter_model_uncertainties_ON.m` to create the state space model of the augmented helicopter. The PID values are unchanged.
- **Adaptation\_acc.m** contains the design of the adaptive controller. From here one can change the tuning knobs of the controller.
- **montecarlo\_acc.m** is the scripts to run the montecarlo simulation. From here all the parameters of the montecarlo simulation can be modified. Note that the script compiles the simulink schemes in C to speed the simulation up. Additional inputs must be included from the script.
- **data\_analysis\_acc.m** loads the data from the montecarlo simulation and creates different plots.

### 7.2.1 helicopter\_model.m

```
% clear
% clc
close all
load('C:\Users\utente\Desktop\Tesi Magistrale\Matlab\useful_data\X');
x = X;

%-----%
%Calculate the final transfer matrix and the single matrices
%-----%
    s = tf('s');
%     m = 1126.4; %mass in kg of the R44 in take off
    m = 1015; %mass in kg of the R44 in take off
%     m = x(42);
    ch = 0.254; %main rotor blade chord meters
    g = 9.81;
    T0 = m*g;
    Omega = 42; %angular velocity main rotor in rad/s
```

```

radius = 5.029; %radius in meters
    U0 = 0;
    V0 = 0;
    W0 = 0;
    P0 = 0;
    Q0 = 0;
    R0 = 0;
    PHI0 = 0;
    THETA0 = 0;
    k1 = 1;
    k2 = 1;
    k3 = 1;
    wa = 0;
    sigmaa = 2*ch/(pi*radius);% main rotor solidity
%weight coefficients
    Wg = 1.0;
    Wp = 0.01745;

% [r,~] = size(Tc);

C0 = 0.639;
% C0 = 200;
rho = 1.2;%kg/m^3
%-----%
%System matrices
%-----%
% gamma = 5.73*rho*ch*radius^4/341.3393;
gamma = 5.73*rho*ch*radius^4/341.3393;
etact_a = (rho*pi*(radius^2)*(Omega*radius)^2)/m;
ni0 = sqrt(T0/(2*rho*pi*radius^2*(Omega*radius)^2));
    % u, v, w, p, q, r, phi, theta, betalc,betals, x1, x2,...
    %eta-q y1, y2 eta_p ni, beta0, beta0_dot

M = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, etact_a*0.543/(Omega^2*radius*C0), ...
      etact_a*4*ni0/(3*Omega), 0;...
      0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 0, 0, 0, 0, 0, 0, 0, .0259, 0, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, .0259, 0, 0,...
      0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;...

```

```

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, ...
0 0, 0, 0 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, ...
0 0, 0, 0 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, ...
1 0, 0, 0 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 1, 0, 0 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0, 1, 0 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0, -1, 1 0, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0, 0, 0 1, 0, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0, 0, 0 0, 1, 0, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, ...
0 0, 0, 0 0, 0, 1, 0, 0, 0, 0, 0, 0;...
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0, 0, 0 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
1];

% u, v, w, p, q, r, phi, theta, betalc,betals, x1, x2,...
%eta-q y1, y2 eta_p ni, beta0, beta0_dot,
F = [ x(1), +R0, -Q0, 0, -W0, +V0, 0, -g, g, 0, 0, 0, 0, 0, 0, 0, 0, ...
0 0 ;...
-R0, x(6), +P0, +W0, 0, -U0, g, 0, 0, -g, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0 0 ;...
+Q0, -P0, x(9), x(10)-V0, +U0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0 -etact_a*(4*ni0/(Omega*radius)), 0
0 ;...
0, x(13), 0, 0, 0, 0, 0, 0,
-k2*Q0, 0, 0, 0, 0, x(20), 0, 0, 0
0 0, 0, 0 0, 0, 0, 0, 0,
0 ;...
x(14), x(2), 0, x(5), 0, 0, 0, 0,
0, 0, 0, 0, x(18), 0, 0, 0, 0,
0, 0, 0 0, 0, 0, 0,
0 ;...
0, 0, x(12), x(3)-k3*Q0, -k1*R0-k3*P0, x(15)-k1*Q0,
0, 0, 0, x(22), 0, 0, 0, 0,
0 0, 0, 0
0 ;...
0, 0, 0, 1, sin(PHI0)*tan(THETA0), cos(PHI0)*tan(THETA0),
0, wa*sec(THETA0), 0, 0, 0, 0, 0, 0,
0 0, 0, 0
0 ;...
0, 0, 0, 0, 0, cos(THETA0), -sin(THETA0), -
0, 0, 0, 0, 0, 0, 0, 0,
0 0 ;...
0, 0, 0, 0, 0, -.0259,
0, 0, 0, 0, -1, 0, 0, 0,
0, 0, 0 0, 0, 0,
0 ;...

```

```

0,      0,      0,      -.0259,      0,      0,      0,      0
0,      0,      0,      0,      x(17),      -1,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      0,      1,      0
0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      x(23),      -205.5209,      -2.8672,      0
0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      x(25),      x(26),      0
0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0,      1,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      x(24),      0,      0,      0,      0,      0,      -205.5209,      -2.8672,      0
0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      x(27),      0
x(28),      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      (-75*pi*Omega/32)*(ni0+5.73*sigmaa/16)*C0,      0,      (-25*pi*Omega*radius),      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0
1;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      -Omega*gamma/(6*radius),      0,      0,      0,      0,      0,      0,      -Omega^2,      0,      0
-Omega*gamma/8
];

%      lat,      long,      ped,      col
G = [      0,      0,      0,      0;...      % u
        0,      0,      0,      0;...      % v
        0,      0,      0,      0;...      % w
        0,      0,      0,      0;...      % p
        0,      0,      0,      0;...      % q
        0,      0,      x(21),      0;...      % r
        0,      0,      0,      0;...      % phi
        0,      0,      0,      0;...      % theta
        x(7),      x(16),      0,      0;...      % betalc
        x(11),      x(8),      0,      0;...      % betals
        0,      0,      0,      0;...      % x1
        0,      0,      0,      0;...      % x2

```





```

0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0      0,      0,      0,      0      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0      0,      0,      0,      0      0,      0,      0
];

```

```

A = M\F;
B = M\G;
C = H0+H1*(M\F);
D = H1*(M\G);
% Eliminate accelerations
C = C([1,2,3,4,5,6,10,11],:);
D = D([1,2,3,4,5,6,10,11],:);

[T,U] = minreal(ss(A,B,C,D));

```

## 7.2.2 helicopter\_model\_uncertainties\_ON.m

```

% clear
% clc
% close all
load('D:\Uni\Tesi Magistrale\Matlab\useful_data\X');
load('D:\Uni\Tesi Magistrale\Matlab\useful_data\CR');
% load('D:\Uni\Tesi Magistrale\Matlab\useful_data\worstcase_unc');

%random
% x = zeros(length(X),1);
% for i = 1:length(X)
%     x(i) = normrnd(X(i),3*CR(i));
% end
% x = X;
%max or min value
% for i = 1:length(X)
%     if round(rand(1,1)) == 0
%         x(i) = X(i) + 3*CR(i);
%     else
%         x(i) = X(i) -3*CR(i);
%     end
% end
% % worst case
% x(1) = perfmargunc.x1;
% x(2) = perfmargunc.x2;
% x(3) = perfmargunc.x3;
% x(4) = perfmargunc.x4;
% x(5) = perfmargunc.x5;
% x(6) = perfmargunc.x6;
% x(7) = perfmargunc.x7;
% x(8) = perfmargunc.x8;
% x(9) = perfmargunc.x1;
% x(10) = perfmargunc.x10;
% x(11) = perfmargunc.x11;

```

```

% x(12) = perfmargunc.x12;
% x(13) = perfmargunc.x13;
% x(14) = perfmargunc.x14;
% x(15) = perfmargunc.x15;
% x(16) = perfmargunc.x16;
% x(17) = perfmargunc.x17;
% x(18) = perfmargunc.x18;
% x(19) = 0;
% x(20) = perfmargunc.x20;
% x(21) = perfmargunc.x21;
% x(22) = perfmargunc.x22;
% x(23) = perfmargunc.x23;
% x(24) = perfmargunc.x24;
% x(25) = perfmargunc.x25;
% x(26) = perfmargunc.x26;
% x(27) = perfmargunc.x27;
% x(28) = perfmargunc.x28;
% x = xunc;
x = rho(1, :);
%-----%
%Calculate the final transfer matrix and the single matrices
%-----%
    s = tf('s');
%     m = 1126.4; %mass in kg of the R44 in take off
    m = 1015; %mass in kg of the R44 in take off
%     m = x(42);
    ch = 0.254; %main rotor blade chord meters
    g = 9.81;
    T0 = m*g;
    Omega = 42; %angular velocity main rotor in rad/s
    radius = 5.029; %radius in meters
    U0 = 0;
    V0 = 0;
    W0 = 0;
    P0 = 0;
    Q0 = 0;
    R0 = 0;
    PHI0 = 0;
    THETA0 = 0;
    k1 = 1;
    k2 = 1;
    k3 = 1;
    wa = 0;
    sigmaa = 2*ch/(pi*radius); % main rotor solidity
%weight coefficients
    Wg = 1.0;
    Wp = 0.01745;

% [r,~] = size(Tc);

C0 = 0.639;
% C0 = 200;
rho = 1.2;%kg/m^3
%-----%

```



```

beta0_dot,
F = [
    x(1),      +R0,      -Q0,      0,      0,      -W0,
+V0,          0,      -g,      g,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,
0;
    -R0,      x(6),      +P0,      +W0,      0,
-U0,          g,      0,      0,      -g,      0,      0,
0,      0,      0,      0,      0,      0,      0,
0;
    +Q0,      -P0,      x(9),      x(10)-V0,      +U0,
0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      -etact_a*(4*ni0/(Omega*radius)),      0,
0;
    0,      x(13),      0,      0,      0,
-k2*Q0,      0,      0,      0,      x(20),      0,      0,
0,      0,      0,      0,      0,      0,
0;
    x(14),      x(2),      0,      x(5),      0,
0,      0,      0,      x(18),      0,      0,      0,
0,      0,      0,      0,      0,      0,
0;
    0,      0,      x(12),      x(3)-k3*Q0,      -k1*R0-k3*P0,
x(15)-k1*Q0,      0,      0,      0,      x(22),      0,      0,
0,      0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      1,      sin(PHI0)*tan(THETA0),      cos(PHI0)*tan(THETA0),
0,      wa*sec(THETA0),      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      0,      cos(THETA0),
-sin(THETA0),      -wa*cos(THETA0),      0,      0,      0,      0,
0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      0,      -0.0259,
0,      0,      0,      0,      -1,      0,      0,      0,
0,      0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      -0.0259,      0,
0,      0,      0,      0,      x(17),      -1,      0,      0,
0,      0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      0,      0,      0,      1,      0,
0,      0,      0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      0,      0,      x(23),
0,      0,      0,      0,      0,      0,      -205.5209,      -2.8672,      0,
0,      0,      0,      0,      0,      0,
0;
    0,      0,      0,      0,      0,      0,      x(25),      x(26),      0,
0,      0,      0,      0,      0,      0,
0;
    0;

```

```

0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0
0,      1,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      x(24),      0,
0,      0,      0,      0,      0,      0,      0,      0,      0 -205.5209, -2.
0,      0,      0,      0,      0,      0,      0,      0,      0
0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,
x(28),      0,      0,      0,      0,      0,      0,      0,      0 x(27),
0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0,      0
1;...
0,      0,      0,      0,      0,      0,      0,      0,      0
0,      0,      0,      0,      0,      0,      0,      0,      0
-Omega*gamma/8,      -Omega*gamma/(6*radius),      -Omega^2,
];

%      lat,      long,      ped,      col
G = [      0,      0,      0,      0;... % u
        0,      0,      0,      0;... % v
        0,      0,      0,      0;... % w
        0,      0,      0,      0;... % p
        0,      0,      0,      0;... % q
        0,      0,      x(21),      0;... % r
        0,      0,      0,      0;... % phi
        0,      0,      0,      0;... % theta
        x(7),      x(16),      0,      0;... % betalc
        x(11),      x(8),      0,      0;... % betals
        0,      0,      0,      0;... % x1
        0,      0,      0,      0;... % x2
        0,      0,      0,      0;... % eta-q
        0,      0,      0,      0;... % y1
        0,      0,      0,      0;... % y2
        0,      0,      0,      0;... % eta-p
        0,      0,      0,      (25*pi*(Omega^2)*radius/32)*(5.73*sigmaa/8)*C0*(x(4));... %
        0,      0,      0,      0;... % beta0
        0,      0,      0,      ((Omega^2)*gamma/8)*(x(4));... % beta0_dot
];

%      u,      v,      w,      p,      q,      r,      phi,      theta,      betalc,betals, x1,
x2,      eta-q      y1,      y2,      eta-p      ni,      beta0,      beta0_dot

H0 = [      1,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
        0,      1,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...

```

```

0,      0,      0,      1,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      1,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      W0,      -V0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      -W0,      0,      U0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      V0,      -U0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      1,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      1,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
];

```

```

H1 = [
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0;...
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      1,      0,      0,      0,      0,      0,
0,      0,      0,      1,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      1,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      1,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,      0,
];

```

```

A = M\F;
B = M\G;
C = H0+H1*(M\F);
D = H1*(M\G);
% Eliminate accelerations
C = C([1,2,3,4,5,6,10,11],:);
D = D([1,2,3,4,5,6,10,11],:);

```

```
[Tun,U] = minreal(ss(A,B,C,D));
```

### 7.2.3 pidCL.m

```

%% Addpath & Load heli model
% clear all;
% close all;
% clc;

addpath('D:\Uni\Tesi Magistrale\Matlab')
addpath('D:\Uni\Tesi Magistrale\Matlab\Mimotools')
addpath('D:\Uni\Tesi Magistrale\Matlab\utility_functions')
helicopter_model

%% PID controllers
p1 = 0.00685573330460408;
d1 = 0.206685839902812;
n1 = 222172.099709302;
PD1 = p1 + d1*n1/(1+n1/s);

p2 = -0.050968413980726;
d2 = -0.180732929270159;
n2 = 573.055407375731;
PD2 = p2 + d2*n2/(1+n2/s);

p3 = -19.2413361184279;
i3 = -7.4980503154103;
d3 = -2.75013277646159;
n3 = 77049.385285613;
PID3 = p3 + i3/s + d3*n3/(1+n3/s);

p4 = 200.29405677187591;
i4 = 0.682713668257837;
d4 = 3.53887257037233;
n4 = 21764.2373310266;
PID4 = p4 + i4/s + d4*n4/(1+n4/s);

p11 = 11.4689389172983;
i11 = 10.678603429402;
d11 = 0.171845473406865;
n11 = 144.103005394287;
PID11 = p11 + i11/s + d11*n11/(1+n11/s);

p22 = 37.0840650215077;
i22 = 57.7121950431289;
d22 = 1.00175371267254;
n22 = 165669.450652458;
PID22 = p22 + i22/s + d22*n22/(1+n22/s);

PIDin = [PID11 0; 0 PID22];
PIDout = [PD1 0 0 0; 0 PD2 0 0; 0 0 PID3 0; 0 0 0 PID4];

lat2v = tf(1.4,[1.25 1]);
lon2u = tf(1.4,[0.7 1]);

```



```

ped2r = tf(0.502,[0.25 1]);
col2w = tf(1,[0.82 1]);%*exp(-0.25*s);

% H =[lat2v/1.4 0 0 0; 0 lon2u/1.4 0 0; 0 0 col2w 0; 0 0 0 ped2r/0.502];

%% Closed Loop Linear Fractional Transformation Representation
%
% Pzw = [zeros(11,2) T(1:11,3:4)];
% Pzu = T(:,1:2);
% Pyw = [-eye(2) swaprow(T(10:11,3:4),1,2)];
% Pyu = swaprow(T(10:11,1:2),1,2);
%
% Tzw = Pzw + Pzu*PIDin/(eye(2) - Pyu*PIDin)*Pyw;
%
% Myu = [Tzw(2,:);
%        Tzw(1,:);
%        Tzw(3,:);
%        Tzw(6,:)];
% tic
% try
%     CL = -Tzw*PIDout/(eye(4) - Myu*PIDout)*H;
% catch
%     toc
% end
% toc
% step(CL);

%% Use of connect to speed up (?)
PID11.u = 'e_p';
PID11.y = 'u1';
PID22.u = 'e_q';
PID22.y = 'u2';
PD1.u = 'e_v';
PD1.y = 'p_ref';
PD2.u = 'e_u';
PD2.y = 'q_ref';
PID3.u = 'e_w';
PID3.y = 'u3';
PID4.u = 'e_r';
PID4.y = 'u4';
lat2v.u = 'lat_com';
lat2v.y = 'v_ref';
lon2u.u = 'lon_com';
lon2u.y = 'u_ref';
ped2r.u = 'ped_com';
ped2r.y = 'r_ref';
col2w.u = 'col_com';
col2w.y = 'w_ref';
uT = {'u1','u2','u3','u4'};
T.u = uT;
yT = {'u','v','w','p','q','r','theta','phi'};
T.y = yT;
sum1 = sumblk('e_v = v_ref - v');
sum2 = sumblk('e_u = u_ref - u');

```

```

sum3 = sumblk('e_w = w_ref - w');
sum4 = sumblk('e_r = r_ref - r');
sum5 = sumblk('e_q = q_ref - q');
sum6 = sumblk('e_p = p_ref - p');

rT = {'lat_com', 'lon_com', 'col_com', 'ped_com'};
CL = connect(PD1,PD2,PID3,PID4,PID11,PID22,...
            T,sum1,sum2,sum3,sum4,sum5,sum6,...
            lat2v,lon2u,col2w,ped2r,...
            rT,...
            yT);

```

## 7.2.4 pidCL\_uncertainties\_ON.m

```

%% Addpath & Load heli model

addpath('D:\Uni\Tesi Magistrale\Matlab')
addpath('D:\Uni\Tesi Magistrale\Matlab\Mimotools')
addpath('D:\Uni\Tesi Magistrale\Matlab\utility_functions')
helicopter_model_uncertainties_ON

%% PID controllers
p1 = 0.00685573330460408;
d1 = 0.206685839902812;
n1 = 222172.099709302;
PD1 = p1 + d1*n1/(1+n1/s);

p2 = -0.050968413980726;
d2 = -0.180732929270159;
n2 = 573.055407375731;
PD2 = p2 + d2*n2/(1+n2/s);

p3 = -19.2413361184279;
i3 = -7.4980503154103;
d3 = -2.75013277646159;
n3 = 77049.385285613;
PID3 = p3 + i3/s + d3*n3/(1+n3/s);

p4 = 200.29405677187591;
i4 = 0.682713668257837;
d4 = 3.53887257037233;
n4 = 21764.2373310266;
PID4 = p4 + i4/s + d4*n4/(1+n4/s);

p11 = 11.4689389172983;
i11 = 10.678603429402;
d11 = 0.171845473406865;
n11 = 144.103005394287;
PID11 = p11 + i11/s + d11*n11/(1+n11/s);

p22 = 37.0840650215077;
i22 = 57.7121950431289;

```

```

d22 = 1.00175371267254;
n22 = 165669.450652458;
PID22 = p22 + i22/s + d22*n22/(1+n22/s);

PIDin = [PID11 0; 0 PID22];
PIDout = [PD1 0 0 0; 0 PD2 0 0; 0 0 PID3 0; 0 0 0 PID4];

lat2v = tf(1.4,[1.25 1]);
lon2u = tf(1.4,[0.7 1]);
ped2r = tf(0.502,[0.25 1]);
col2w = tf(1,[0.82 1]);%*exp(-0.25*s);

% H =[lat2v/1.4 0 0 0; 0 lon2u/1.4 0 0; 0 0 col2w 0; 0 0 0 ped2r/0.502];

%% Closed Loop Linear Fractional Transformation Representation
%
% Pzw = [zeros(11,2) T(1:11,3:4)];
% Pzu = T(:,1:2);
% Pyw = [-eye(2) swaprow(T(10:11,3:4),1,2)];
% Pyu = swaprow(T(10:11,1:2),1,2);
%
% Tzw = Pzw + Pzu*PIDin/(eye(2) - Pyu*PIDin)*Pyw;
%
% Myu = [Tzw(2,:);
%        Tzw(1,:);
%        Tzw(3,:);
%        Tzw(6,:)];
% tic
% try
%     CL = -Tzw*PIDout/(eye(4) - Myu*PIDout)*H;
% catch
%     toc
% end
% toc
% step(CL);

%% Use of connect to speed up (?)
PID11.u = 'e_p';
PID11.y = 'u1';
PID22.u = 'e_q';
PID22.y = 'u2';
PD1.u = 'e_v';
PD1.y = 'p_ref';
PD2.u = 'e_u';
PD2.y = 'q_ref';
PID3.u = 'e_w';
PID3.y = 'u3';
PID4.u = 'e_r';
PID4.y = 'u4';
lat2v.u = 'lat_com';
lat2v.y = 'v_ref';
lon2u.u = 'lon_com';
lon2u.y = 'u_ref';
ped2r.u = 'ped_com';

```

```

ped2r.y = 'r_ref';
col2w.u = 'col_com';
col2w.y = 'w_ref';
uT = {'u1', 'u2', 'u3', 'u4'};
Tun.u = uT;
yT = {'u', 'v', 'w', 'p', 'q', 'r', 'theta', 'phi'};
Tun.y = yT;
sum1 = sumblk('e_v = v_ref - v');
sum2 = sumblk('e_u = u_ref - u');
sum3 = sumblk('e_w = w_ref - w');
sum4 = sumblk('e_r = r_ref - r');
sum5 = sumblk('e_q = q_ref - q');
sum6 = sumblk('e_p = p_ref - p');

rT = {'lat_com', 'lon_com', 'col_com', 'ped_com'};
CL_un = connect(PD1, PD2, PID3, PID4, PID11, PID22, ...
               Tun, sum1, sum2, sum3, sum4, sum5, sum6, ...
               lat2v, lon2u, col2w, ped2r, ...
               rT, ...
               yT);

```

## 7.2.5 Adaptation\_acc.m

```

%% Addpath & Load heli models
% clear all;
% close all;
% clc;

addpath('D:\Uni\Tesi Magistrale\Matlab')
addpath('D:\Uni\Tesi Magistrale\Matlab\pid')
addpath('D:\Uni\Tesi Magistrale\Matlab\Hinf.Mu')

pidCL
pidCL.uncertainties_ON
s = tf('s');
n = length(CL.a(1, :));
m = length(CL.b(1, :));
l = length(CL.c(:, 1));

% deltaA = CL_un.a - CL.a;
deltaA = zeros(n);
% deltaA = [10*eye(n-m) zeros(n-m, m); zeros(m, n)];
% deltaA matched
% deltaA = CL.b*ones(4, 31);

Bu = null(CL.b');
B = [CL.b Bu];
Binv = inv(B);
f = B\(\deltaA);
f1 = f(1:4, :);
f2 = f(5:end, :);
w0 = eye(m);

```

```

%Altre combo fase minima (PID 3 4 5 6),
Hm = CL([3 6 7 8],:);
Hum = ss(CL.a,Bu,CL.c,[CL.d zeros(1,n-2*m)]);
Hum = Hum([3 6 7 8],:);
Hminv = minreal((inv(Hm)/(1+s/1e4)^4),1e-9);

%% Tuning knob
P = lyap(CL.a',1*eye(n));
PBm = P*CL.b;
PBum = P*Bu;
K = 10*eye(m);
Asp = -0.1*eye(n);
Ts = 1e-5;
Tsave = 1e-2;
Adgainum = inv(B)*inv(Asp*(eye(n)-exp(Asp*Ts)))*exp(Asp*Ts);
% funziona...non eccellente, lento
% Gamma = 1e2;
% C = ss(1/(1+s/10)^4*eye(m));
% Gamma = 1e6;
% C = ss(1/(1+s/50)^4*eye(m));
Gamma = 1e5;
C = ss(1/(1+s/50)^5*eye(m)); %funziona senza proj
% C = ss([1/(1+s/50)^4 0 0 0;
%       0 1/(1+s/10)^4 0 0;
%       0 0 1/(1+s/50)^4 0;
%       0 0 0 1/(1+s/50)^4]); %provetta
% C = ss(1/(1+s/100)^4*eye(m)); % prova con proj
adOFF = 0;

%%
Hmtilde = ss(CL.a+deltaA, CL.b, eye(n), zeros(n,m));
CLref = minreal((eye(n) + Hmtilde*Hminv*Hum*f2)\Hmtilde,1e-3);

```

## 7.2.6 montecarlo\_acc.m

```

clear all;
close all;
clc;
addpath('D:\Uni\Tesi Magistrale\Matlab')
addpath('D:\Uni\Tesi Magistrale\Matlab\pid')
load('C:\Users\gpicardi\Desktop\simulator_accelerator\rho.mat');
unc_param = rho;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Parametri simulazioni
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

N = 1;
Nsave = 500;
Tsave = 1e-2;
simtime = 10;
samples = simtime/Tsave+1;

```

```

minind = 1;
maxind = Nsave;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Indicizzazione adaptive & uncertain:
% 1)input 2)simulation 3)signal 4)time
% Indicizzazione nominal:
% 1)input 2)signal 3)time
% Indicizzazione adinput:
% 1)simulation 2)input 3)signal 4)time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

adaptive = zeros(4,N,8,samples);
uncertain = zeros(4,N,8,samples);
unstable = zeros(N,1);
unc_unst = zeros(N,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Montecarlo per il sistema incerto con meccanismo adattivo acceso
% (adaptive) e spento (uncertain)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load_system('adaptive_acc');
load_system('nonadaptive_acc');

% Input
parSim.LoadExternalInput = 'on';
parSim.ExternalInput = 'DeltaA, latstep, lonstep, collstep, pedstep';
parSim.StopTime = num2str(simtime);
parSim.StartTime = num2str(0);

% Fast simulation
% See: http://undocumentedmatlab.com/blog/improving-simulink-performance/
% Note:
% - the use of time structures as inputs has improved a lot the performance
% (compared to using buses) with accelerator mode.
% This is not valid for 'rapid' mode.
parSim.SimulationMode = 'accelerator';
parSim.SFSimEnableDebug = 'off';
parSim.SFSimEcho = 'off';
parSim.InlineParams = 'off';

% Logging
parSim.SaveTime = 'off';
parSim.SaveOutput = 'off';

% Load paramNameValStruct for fast simulations
fname = fieldnames(parSim);
fval = struct2cell(parSim);
simOpt = [fname, fval]';
simOpt = simOpt(:);
set_param('adaptive_acc', simOpt{:});
set_param('nonadaptive_acc', simOpt{:});

c = 1; % counter for unstable
offset = input('offset incertezze: ');

```

```

cpunumber = input('numero simulazione: ');
for iter = 1:N %this for changes the realization of uncertainty
    try

        xunc = unc_param(iter+offset,:);
        Adaptation_acc
        DeltaA.time = [0 simtime];
        DeltaA.signals.values(:, :,1) = deltaA;
        DeltaA.signals.values(:, :,2) = deltaA;
        DeltaA.signals.dimensions = [n,n];

        for cmd = 1:4 % seleziona l'ingresso attivo
            latstep.time = [0 simtime]';
            latstep.signals.values = [0, 0]';
            latstep.signals.dimensions = 1;
            lonstep.time = [0 simtime]';
            lonstep.signals.values = [0, 0]';
            lonstep.signals.dimensions = 1;
            collstep.time = [0 simtime]';
            collstep.signals.values = [0, 0]';
            collstep.signals.dimensions = 1;
            pedstep.time = [0 simtime]';
            pedstep.signals.values = [0, 0]';
            pedstep.signals.dimensions = 1;
            switch cmd
                case 1
                    latstep.signals.values = [1, 1]';
                case 2
                    lonstep.signals.values = [1, 1]';
                case 3
                    collstep.signals.values = [1, 1]';
                case 4
                    pedstep.signals.values = [1, 1]';
            end
            %Sistema incerto senza adattamento

            % Load parameters that change in the Model Workspace
            hws = get_param('nonadaptive_acc', 'modelworkspace');
            hws.assignin('DeltaA',DeltaA);
            hws.assignin('latstep',latstep);
            hws.assignin('lonstep',lonstep);
            hws.assignin('collstep',collstep);
            hws.assignin('pedstep',pedstep);

            tic
            sim('nonadaptive_acc');
            toc
            %renaming output to workspace (refs are always the same)
            uncertain(cmd,iter,1:3,:) = uvw';
            uncertain(cmd,iter,7:8,:) = phitheta';
            uncertain(cmd,iter,4:6,:) = pqr';

            %Sistema incerto con adattamento

```

```

% Load parameters that change in the Model Workspace
hws = get_param('adaptive_acc', 'modelworkspace');
hws.assignin('DeltaA',DeltaA);
hws.assignin('latstep',latstep);
hws.assignin('lonstep',lonstep);
hws.assignin('collstep',collstep);
hws.assignin('pedstep',pedstep);

tic
sim('adaptive_acc');
%renaming output to workspace (refs are always the same)
adaptive(cmd,iter,1:3,:) = squeeze(uvw);
adaptive(cmd,iter,7:8,:) = squeeze(phitheta);
adaptive(cmd,iter,4:6,:) = squeeze(pqr);
toc

end
catch
unstable(iter) = 1;
unc_unst(iter) = iter + offset;
end
% if mod(iter,Nsave) == 0
%   adsave = adaptive(:,minind:maxind,:);
%   uncsave = uncertain(:,minind:maxind,:);
%   unstabletmp = unstable(minind:maxind);
%   adsave = adsave(:,~unstabletmp,:); %#ok<NASGU>
%   uncsave = uncsave(:,~unstabletmp,:); %#ok<NASGU>
%   save(strcat('montecarlo_data',num2str(cpunumber),num2str(...
%   floor(iter/Nsave))), 'uncsave','adsave');
%   adsave = zeros(4,Nsave,8,samples);
%   uncsave = zeros(4,Nsave,8,samples);
%   minind = maxind + 1;
%   maxind = maxind + maxind;
% end
end
save(strcat('montecarlo',num2str(cpunumber)), 'adaptive','uncertain');
save(strcat('unstable',num2str(cpunumber)), 'unstable');

```

## 7.2.7 data\_analysis\_acc.m

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plots and data processing from montecarlo.m
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all
close all
clc

% load('montecarlo11');
load('nominal');
load('montecarlo_data');
load('uncertain_filtered');
%adaptive = adaptive(:,1:1000,:);

```



```

%uncertain = uncertain(:,1:length(squeeze(adaptive(1, :, 1, 1))), :, :);
%
N = length(squeeze(adaptive(1, :, 1, 1)));
samples = length(squeeze(adaptive(1, 1, 1, :)));
%mean and sd of adapted and non adapted output signals over the
%uncertainties realization
mean_ad = zeros(4, 8, samples);
sd_ad = zeros(4, 8, samples);
mean_un = zeros(4, 8, samples);
sd_un = zeros(4, 8, samples);
%error wrt nominal signal (for all realization for all time)
error_ad = zeros(4, N, 8, samples);
error_un = zeros(4, N, 8, samples);
reler_ad = zeros(4, N, 8, samples);
reler_un = zeros(4, N, 8, samples);
%mean and sd of error for adapted and non output over the uncertainty
%realizations
mean_err_ad = zeros(4, 8, samples);
sd_err_ad = zeros(4, 8, samples);
mean_err_un = zeros(4, 8, samples);
sd_err_un = zeros(4, 8, samples);
%mean and max error over time (varies with the uncertainty realization)
mean_err_ad_par = zeros(4, 8, N);
mean_err_un_par = zeros(4, 8, N);
max_err_ad_par = zeros(4, 8, N);
max_err_un_par = zeros(4, 8, N);
%0.05 and 0.95 percentiles of mean and max error for uncertain and
%adaptive
Qmean_err_ad = zeros(4, 8, 2);
Qmean_err_un = zeros(4, 8, 2);
Qmax_err_ad = zeros(4, 8, 2);
Qmax_err_un = zeros(4, 8, 2);

%Compute error for adapted and non adapted
for iter=1:N
    error_ad(:, iter, :, :) = abs(nominal-squeeze(adaptive(:, iter, :, :)));
    error_un(:, iter, :, :) = abs(nominal-squeeze(uncertain(:, iter, :, :)));
    reler_ad(:, iter, :, :) = squeeze(error_ad(:, iter, :, :))./abs(nominal);
    reler_un(:, iter, :, :) = squeeze(error_un(:, iter, :, :))./abs(nominal);
end

%Compute mean and sd for adapted and non adapted over uncertainties
%realizations. Resulting signals depend on time
for cmd=1:4
    for output=1:8
        tmp = adaptive(cmd, :, output, :);
        mean_ad(cmd, output, :) = mean(tmp);
        sd_ad(cmd, output, :) = std(tmp);

        tmp = uncertain(cmd, :, output, :);
        mean_un(cmd, output, :) = mean(tmp);
        sd_un(cmd, output, :) = std(tmp);
    %
        tmp = error_ad(cmd, :, output, :);

```

```

mean_err_ad(cmd,output,:) = mean(tmp);
sd_err_ad(cmd,output,:) = std(tmp);

tmp = error_un(cmd,:,output,:);
mean_err_un(cmd,output,:) = mean(tmp);
sd_err_un(cmd,output,:) = std(tmp);
end
end

% %Compute the mean and max error over time (depend on parameter
% %realization)
for cmd=1:4
    for output=1:8
        for iter=1:N
            mean_err_ad_par(cmd,output,iter) = ...
                mean(squeeze(error_ad(cmd,iter,output,:)));
            mean_err_un_par(cmd,output,iter) = ...
                mean(squeeze(error_un(cmd,iter,output,:)));
            max_err_ad_par(cmd,output,iter) = ...
                max(squeeze(error_ad(cmd,iter,output,:)));
            max_err_un_par(cmd,output,iter) = ...
                max(squeeze(error_un(cmd,iter,output,:)));
        end
        Qmean_err_ad(cmd,output,:)=quantile(mean_err_ad_par(cmd,output,:),...
            [0.05, 0.95]);
        Qmean_err_un(cmd,output,:) = quantile(mean_err_un_par(cmd,output,:),...
            [0.05, 0.95]);
        Qmax_err_ad(cmd,output,:) = quantile(max_err_ad_par(cmd,output,:),...
            [0.05, 0.95]);
        Qmax_err_un(cmd,output,:) = quantile(max_err_un_par(cmd,output,:),...
            [0.05, 0.95]);
    end
end

%% Boxplot
for cmd=1:4
    for output=[3,6]%1:8

        switch cmd
            case 1
                in = 'lateral stick';
            case 2
                in = 'longitudinal stick';
            case 3
                in = 'collective';
            case 4
                in = 'pedal';
        end

        switch output
            case 1
                out = 'u';
                unit = '[m/s]';
            case 2

```

```

        out = 'v';
        unit = '[m/s]';
    case 3
        out = 'w';
        unit = '[m/s]';
    case 4
        out = 'p';
        unit = '[rad/s]';
    case 5
        out = 'q';
        unit = '[rad/s]';
    case 6
        out = 'r';
        unit = '[rad/s]';
    case 7
        out = 'theta';
        unit = '[deg]';
    case 8
        out = 'phi';
        unit = '[deg]';
end

figure
tmp1 = squeeze(mean_err_ad_par(cmd,output,:));
tmp2 = squeeze(mean_err_un_par(cmd,output,:));
qtmp1 = squeeze(Qmean_err_ad(cmd,output,:));
qtmp2 = squeeze(Qmean_err_un(cmd,output,:));

yl(1) = 0;
c1 = max(tmp1);
c2 = max(tmp2);
yl(2) = max(c1,c2) + max(c1,c2)/100;

subplot(1,2,2);
h = boxplot(tmp2,'whisker',1e10,'labels','Adaptation OFF');
% set(h(5,1), 'YData', [qtmp2(1) qtmp2(2) qtmp2(2) qtmp2(1) qtmp2(1)]);
% upWhisker = get(h(1,1), 'YData');
% set(h(1,1), 'YData', [qtmp2(2) upWhisker(2)]);
% dwWhisker = get(h(2,1), 'YData');
% set(h(2,1), 'YData', [ dwWhisker(1) qtmp2(1)])
% xlabel('Adaptation OFF');
ylim(yl);

subplot(1,2,1);
h = boxplot(tmp1,'whisker',1e10,'labels','Adaptation ON');
% set(h(5,1), 'YData', [qtmp1(1) qtmp1(2) qtmp1(2) qtmp1(1) qtmp1(1)]);
% upWhisker = get(h(1,1), 'YData');
% set(h(1,1), 'YData', [qtmp1(2) upWhisker(2)]);
% dwWhisker = get(h(2,1), 'YData');
% set(h(2,1), 'YData', [ dwWhisker(1) qtmp1(1)])
% xlabel('Adaptation ON');
ylim(yl);
ylabel(strcat({'Mean error for input'},{' '},{in},...
    {' and output'},{' '},{out}));

```

```

saveas(gcf, strcat...
('D:\Uni\Tesi Magistrale\Latex\thesis\images\montecarlo_results\boxplot_mean-',...
'input', num2str(cmd), 'output', num2str(output)), 'png');

%       figure
%       tmp1 = squeeze(max_err_ad_par(cmd,output,:));
%       tmp2 = squeeze(max_err_un_par(cmd,output,:));
%       qtmp1 = squeeze(Qmax_err_ad(cmd,output,:));
%       qtmp2 = squeeze(Qmax_err_un(cmd,output,:));
%
%       yl(1) = 0;
%       c1 = max(tmp1);
%       c2 = max(tmp2);
%       yl(2) = max(c1,c2) + max(c1,c2)/100;
%
%       subplot(1,2,2);
%       h = boxplot(tmp2,'whisker',1e10,'labels','Adaptation OFF');
%       set(h(5,1), 'YData', [qtmp2(1) qtmp2(2) qtmp2(2) qtmp2(1) ...
%qtmp2(1)]);
%       upWhisker = get(h(1,1), 'YData');
%       set(h(1,1), 'YData', [qtmp2(2) upWhisker(2)]);
%       dwWhisker = get(h(2,1), 'YData');
%       set(h(2,1), 'YData', [ dwWhisker(1) qtmp2(1)])
%       xlabel('Adaptation OFF')
%       ylim(yl);
%
%       subplot(1,2,1);
%       h = boxplot(tmp1,'whisker',1e10,'labels','Adaptation ON');
%       set(h(5,1), 'YData', [qtmp1(1) qtmp1(2) qtmp1(2) qtmp1(1) ...
%qtmp1(1)]);
%       upWhisker = get(h(1,1), 'YData');
%       set(h(1,1), 'YData', [qtmp1(2) upWhisker(2)]);
%       dwWhisker = get(h(2,1), 'YData');
%       set(h(2,1), 'YData', [ dwWhisker(1) qtmp1(1)])
%       xlabel('Adaptation ON');
%       ylim(yl);
%       title('Max error');
%       saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images...
%\montecarlo_results\boxplot_max-', 'input', num2str(cmd), 'output', ...
%num2str(output)), 'jpg');
%
end
end
%
reler_ad2 = rel_er_ad(3:4, :, [3,6], :);
reler_un2 = rel_er_un(3:4, :, [3,6], :);
reler_ad = rel_er_ad(1:2, :, :, :);
reler_un = rel_er_un(1:2, :, :, :);

tmp1 = squeeze(mean(squeeze(mean(reler_ad(:, :, :, 2:end), 3)), 3));
tmp2 = squeeze(mean(squeeze(mean(reler_un(:, :, :, 2:end), 3)), 3));
tmp3 = squeeze(mean(squeeze(mean(reler_ad2(:, :, :, 2:end), 3)), 3));
tmp4 = squeeze(mean(squeeze(mean(reler_un2(:, :, :, 2:end), 3)), 3));

```

```

figure
% title('Mean relative error for lateral input')
subplot(1,2,1)
boxplot(tmp1(1,:), 'labels', 'Adaptation ON')
title('Mean relative error for lateral input');
ylim([-0.1, max(tmp2(1,:))+1]);
subplot(1,2,2)
boxplot(tmp2(1,:), 'whisker', 1e5, 'labels', 'Adaptation OFF')
% title('uncertain');
ylim([-0.1, max(tmp2(1,:))+1]);
% saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images\...
%montecarlo_results\boxplot_in1'), 'jpg');

```

```

figure
% title('Mean relative error for longitudinal input')
subplot(1,2,1)
boxplot(tmp1(2,:), 'labels', 'Adaptation ON')
title('Mean relative error for longitudinal input');
ylim([-0.1, max(tmp2(1,:))+1]);
subplot(1,2,2)
boxplot(tmp2(2,:), 'whisker', 1e5, 'labels', 'Adaptation OFF')
% title('uncertain');
ylim([-0.1, max(tmp2(1,:))+1]);
% saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images...
%\montecarlo_results\boxplot_in2'), 'jpg');

```

```

figure
% title('Mean relative error for collective input')
subplot(1,2,1)
boxplot(tmp3(1,:), 'labels', 'Adaptation ON')
title('Mean relative error for collective input');
ylim([-0.1, max(tmp4(1,:))+1]);
subplot(1,2,2)
boxplot(tmp4(2,:), 'whisker', 1e5, 'labels', 'Adaptation OFF')
% title('uncertain');
ylim([-0.1, max(tmp4(1,:))+1]);
%saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images...
%\montecarlo_results\boxplot_in3'), 'jpg');

```

```

figure
% title('Mean relative error for pedal input')
subplot(1,2,1)
boxplot(tmp3(2,:), 'labels', 'Adaptation ON')
title('Mean relative error for pedal input');
ylim([-0.1, max(tmp4(2,:))+1]);
subplot(1,2,2)
boxplot(tmp4(2,:), 'whisker', 1e5, 'labels', 'Adaptation OFF')
ylim([-0.1, max(tmp4(2,:))+1]);
%saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images...
%\montecarlo_results\boxplot_in4'), 'jpg');

```

```

%% Shadow separate adON and adOFF
for cmd=1:1
    for output=1:8
        switch cmd
            case 1
                in = 'lateral stick';
            case 2
                in = 'longitudinal stick';
            case 3
                in = 'collective';
            case 4
                in = 'pedal';
        end

        switch output
            case 1
                out = 'u';
                unit = '[m/s]';
            case 2
                out = 'v';
                unit = '[m/s]';
            case 3
                out = 'w';
                unit = '[m/s]';
            case 4
                out = 'p';
                unit = '[rad/s]';
            case 5
                out = 'q';
                unit = '[rad/s]';
            case 6
                out = 'r';
                unit = '[rad/s]';
            case 7
                out = 'theta';
                unit = '[deg]';
            case 8
                out = 'phi';
                unit = '[deg]';
        end

        figure
        x = linspace(0,10,length(squeeze(sd_un(cmd,output,:))));
        % prop = fill([x,x],...
        %[[squeeze(sd_un(cmd,output,:))+squeeze(mean_un(cmd,output,:));0];
        prop = fill([x';flipud(x')],[squeeze(sd_un(cmd,output,:))...
        +squeeze(mean_un(cmd,output,:));flipud(...
        -squeeze(sd_un(cmd,output,:))+squeeze(mean_un(cmd,output,:))...
        ]),[1 0 0]);
        set(prop,'edgecolor','none');
        alpha(0.4);
        hold on
        plot(x,squeeze(mean_un(cmd,output,:)),'r',x,squeeze(nominal(...
        cmd,output,:)),'--k');
    end
end

```

```

xlabel('time, [s]');
ylabel(strcat(out, ', ', ' ', unit));
legend('std adaptation OFF', 'mean adaptation OFF', 'nominal', ...
       'Location', 'southeast')
yl = ylim;
%saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\...
%images\montecarlo_results\shadowUN', 'input', num2str(cmd), ...
%'output', num2str(output)), 'png');
%savefig(strcat('C:\Users\utente\Desktop\Tesi Magistrale\Latex...
%\thesis\images\montecarlo_results\shadowUN', 'input', ...
%num2str(cmd), 'output', num2str(output)));

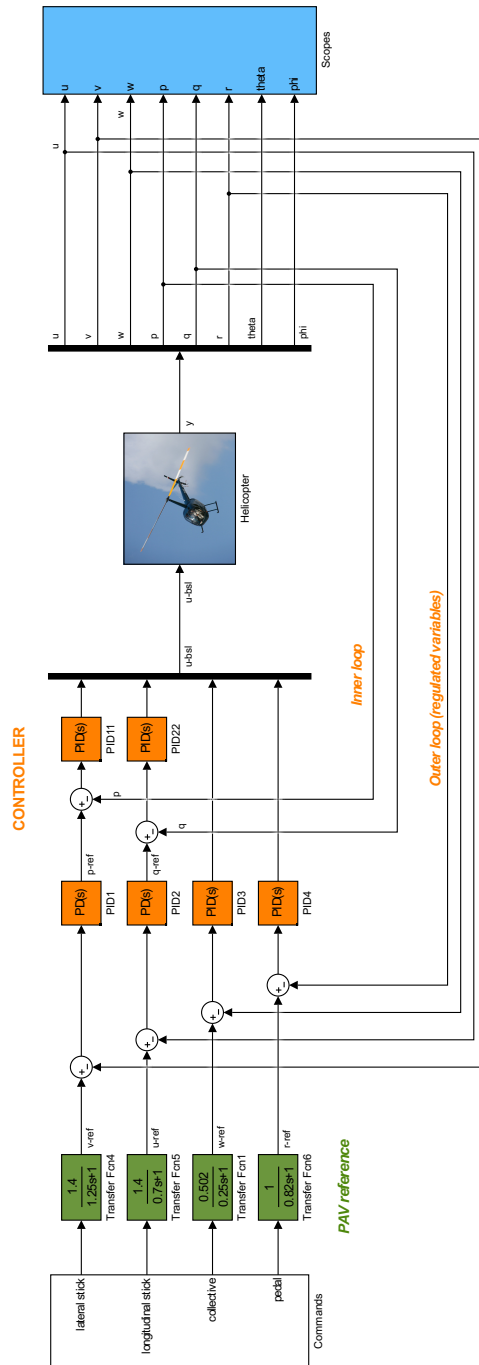
figure
x = linspace(0, 10, length(squeeze(sd_ad(cmd, output, :))));
% prop = fill([x, x], [squeeze(sd_ad(cmd, output, :))+squeeze(...
%mean_ad(cmd, output, :)); 0]; [-squeeze(sd_ad(cmd, output, :))+squeeze(...
%mean_ad(cmd, output, :)); 0]], [0 1 0]);
prop = fill([x'; flipud(x')], [squeeze(sd_ad(cmd, output, :)) ...
+squeeze(mean_ad(cmd, output, :)); flipud(-squeeze(...
sd_ad(cmd, output, :))+squeeze(mean_ad(cmd, output, :))], [0 0 1]);
set(prop, 'edgecolor', 'none');
alpha(0.4);
hold on
plot(x, squeeze(mean_ad(cmd, output, :)), 'b', x, squeeze(...
nominal(cmd, output, :)), '--k');
xlabel('time, [s]');
ylabel(strcat(out, ', ', ' ', unit));
legend('std adaptation ON', 'mean adaptation ON', 'nominal', ...
       'Location', 'southeast')
% ylim(yl);
%saveas(gcf, strcat('D:\Uni\Tesi Magistrale\Latex\thesis\images...
%\montecarlo_results\shadowAD', 'input', num2str(cmd), 'output', ...
%num2str(output)), 'png');
%savefig(strcat('C:\Users\utente\Desktop\Tesi Magistrale\Latex...
%\thesis\images\montecarlo_results\shadowAD', 'input', ...
%num2str(cmd), 'output', num2str(output)));
end
end

```

## 7.3 Appendix C

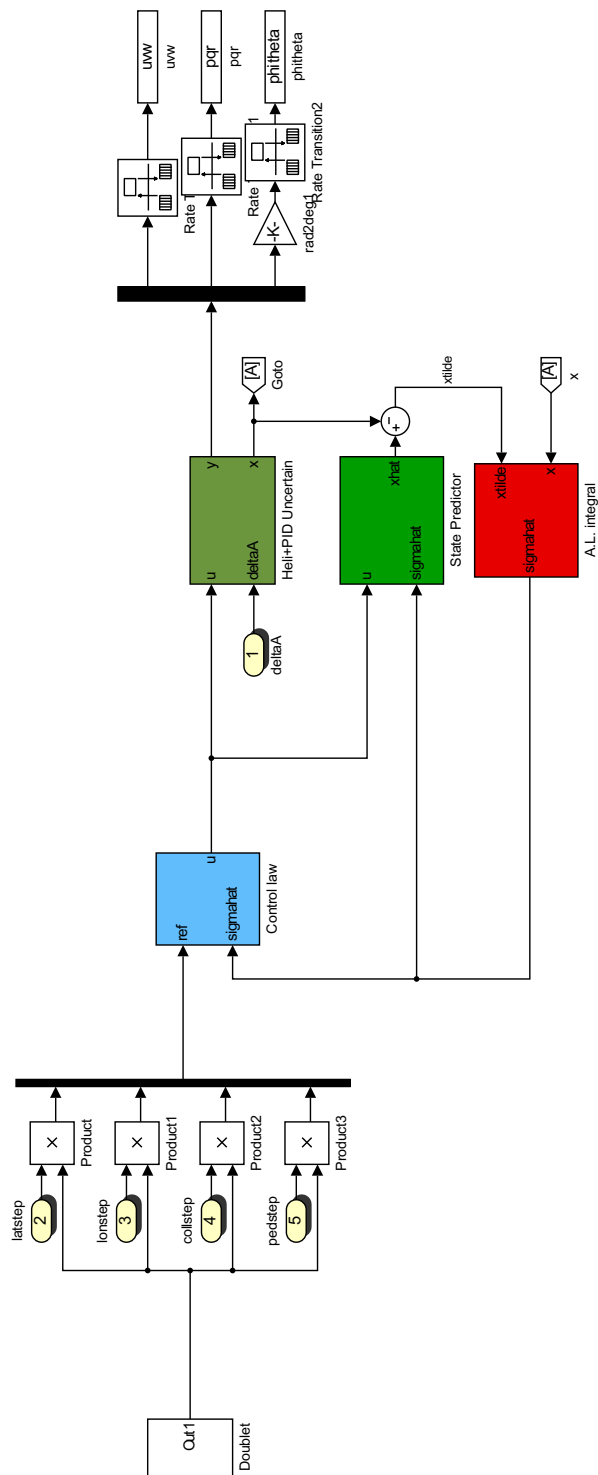
Matlab code provided makes use of the following simulink schemes.

### 7.3.1 Augmented helicopter

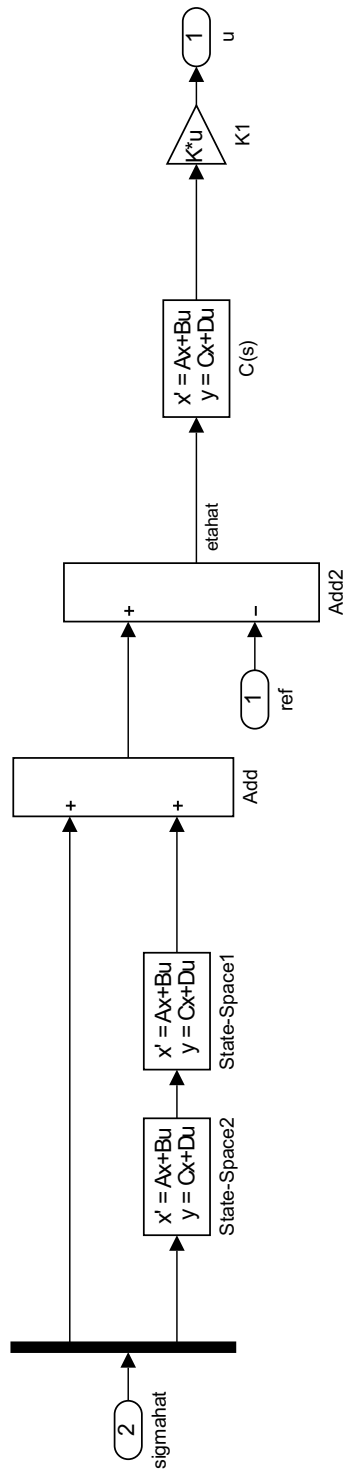




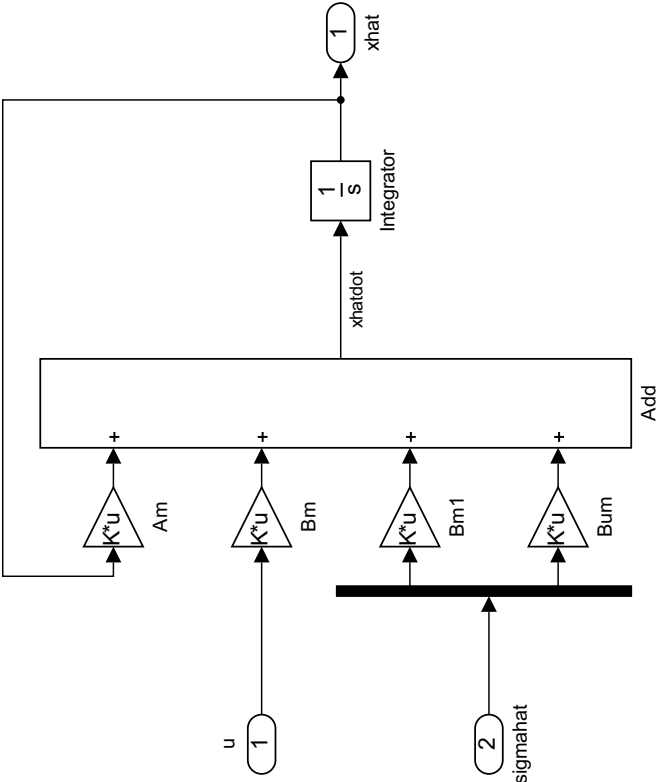
### 7.3.2 Augmented helicopter with adaptive controller



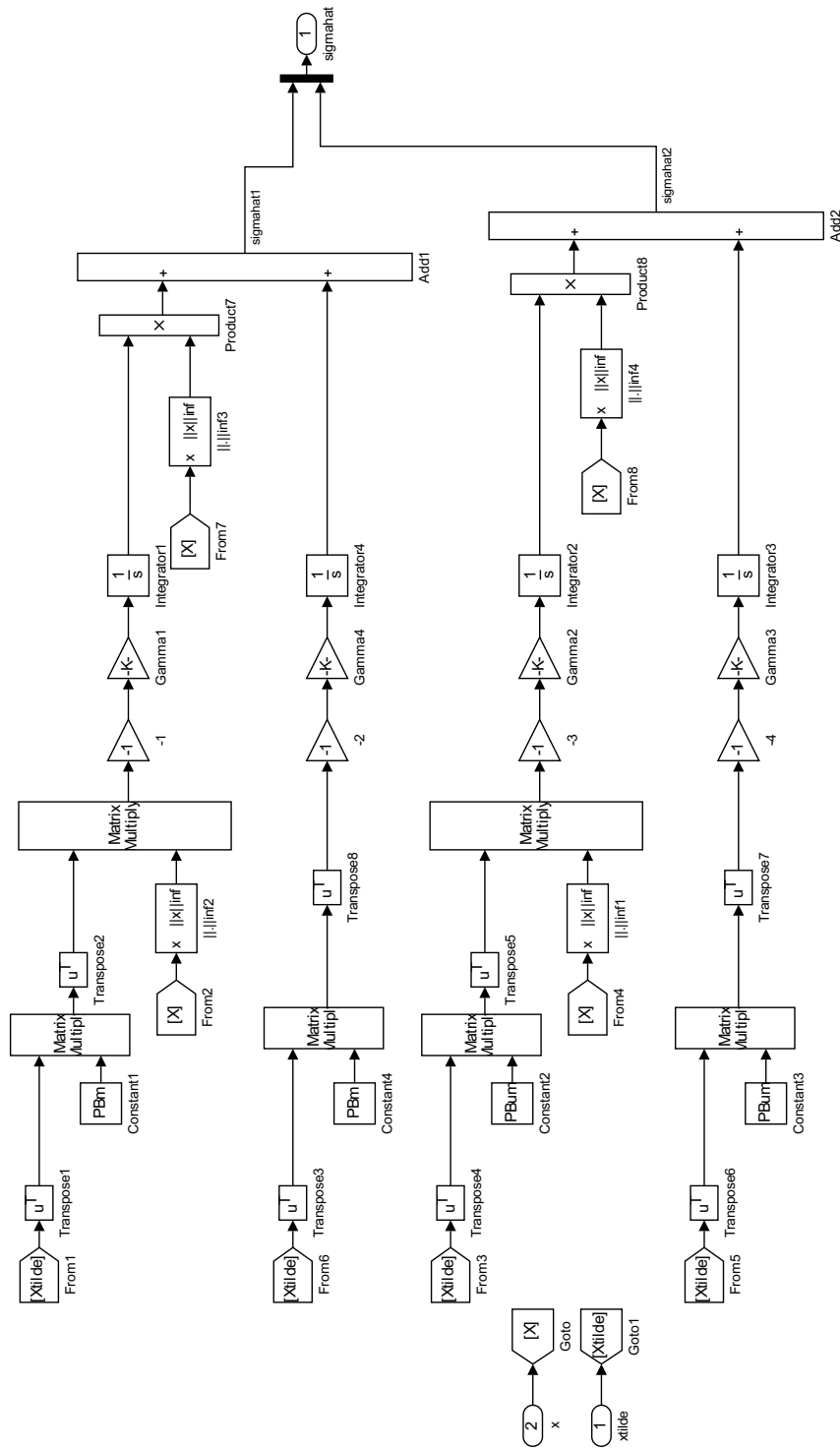
## 7.3.3 Adaptive Control Law



### 7.3.4 State Predictor



### 7.3.5 Estimation Law



# Bibliography

- [1] Perfect, Philip, Michael Jump, and Mark D. White. "Methods to Assess the Handling Qualities Requirements for Personal Aerial Vehicles." *Journal of Guidance, Control, and Dynamics* 38.11 (2015): 2161-2172.
- [2] Perfect, Philip, Michael Jump, and Mark D. White. "Handling Qualities Requirements for Future Personal Aerial Vehicles." *Journal of Guidance, Control, and Dynamics* 38.12 (2015): 2386-2398.
- [3] Cao, Chengyu, and Naira Hovakimyan. "L1 adaptive output-feedback controller for non-strictly-positive-real reference systems: missile longitudinal autopilot design." *Journal of guidance, control, and dynamics* 32.3 (2009): 717-726.
- [4] Xargay, Enric, Naira Hovakimyan, and Chengyu Cao. "L1 adaptive controller for multi-input multi-output systems in the presence of nonlinear unmatched uncertainties." *American Control Conference*. 2010.
- [5] Hovakimyan, Naira, and Chengyu Cao. *L1 adaptive control theory: guaranteed robustness with fast adaptation*. Vol. 21. Siam, 2010.
- [6] Pettersson, Anders, et al. "Analysis of linear L1 adaptive control architectures for aerospace applications." *51st IEEE Conference on Decision and Control*. IEEE, 2012.
- [7] Yue, A., and I. Postlethwaite. "Improvement of helicopter handling qualities using H infinity-optimisation." *Control Theory and Applications, IEE Proceedings D*. Vol. 137. No. 3. IET, 1990.
- [8] Lohar, Fayyaz A. "CONTROL OF HELICOPTER IN." (2000).
- [9] Bichlmeier, Magnus, et al. "L1 adaptive augmentation of a helicopter baseline controller." *AIAA Guidance, Navigation and Control Conference, Boston, AIAA-2013-4855*. 2013.
- [10] Dennehy, Cornelius J. *A Comprehensive Analysis of the X-15 Flight 3-65 Accident* National Aeronautics and Space Administration, Langley Research Center. 2014.
- [11] A. J. Calise, S. Lee, and M. Sharma, *Development of a reconfigurable flight control law for the X-36 tailless fighter aircraft*, Proceedings of AIAA Guidance, Navigation and Control Conference, Denver, Colorado, USA, vol. 2000.

- [12] M. Sharma, A. J. Calise, and J. Corban, *Application of an Adaptive Autopilot Design to a Family of Guided Munitions*, Proceedings of AIAA Guidance, Navigation and Control Conference, Denver, Colorado, USA, 2000.
- [13] K. A. Wise and E. Lavretsky, *Robust and Adaptive Control of X-45A J-UCAS: A Design Trade Study*, 18th IFAC World Congress, Milano, Italy, 2011.
- [14] V. V. Patel, C. Cao, N. Hovakimyan, K. A. Wise, and E. Lavretsky, *L1 Adaptive Control for Tailless Unstable Aircraft in the Presence of Unknown Actuator Failures*, Proceedings of AIAA Guidance
- [15] J.J Burken, P. Williams-Hayes, J. T. Kaneshige, and S. J. Stachowiak, *Adaptive Control Using Neural Network Augmentation for a Modified F-15 Aircraft*, 14th Mediterranean Conference on Control and Automation, Ancona, Italy, 2006.
- [16] Geluardi, S., Nieuwenhuizen F.M., Pollini L., Bühlhoff, H.H., *Augmented Systems for a Personal Aerial Vehicle Using a Civil Light Helicopter Model*, 71st American Helicopter Society International Annual Forum, Virginia Beach, Virginia, May 5-7, 2015.
- [17] Geluardi, S., Nieuwenhuizen F.M., Pollini L., Bühlhoff, H.H., *Frequency Domain System Identification of a Light Helicopter in Hover*, 70th American Helicopter Society International Annual Forum, Montreal, Canada, May 20-22, 2014.
- [18] Baskett, B.J., *Aeronautical Design Standard Performance Specification Handling Qualities Requirements for Military Rotorcraft*, ADS-33E-PRF, USAAMC, March 2000.
- [19] Lavretsky, E., Wise, K., A., *Robust and Adaptive Control with Aerospace Applications* Springer, 2013.
- [20] Guerreiro, B. J., Silvestre, C., Cunha, R., Cao, C., Hovakimyan, N.  *$\mathcal{L}_1$  adaptive control for autonomous rotorcraft*. 2009 American Control Conference, St. Louis, Missouri, June 10-12, 2009.
- [21] Bichlmeier, M., Holzapfel, F., Xargay, E., Hovakimyan, N.  *$\mathcal{L}_1$  adaptive augmentation of a helicopter baseline controller*. 2013 AIAA Guidance, Navigation and Control Conference, Boston, Massachusetts, August 19-22, 2013.
- [22] Gregory, I. M., Cao, C., Xargay, E., Hovakimyan, N., Zou, X. ,  *$\mathcal{L}_1$  adaptive control design for NASA AirSTAR flight test vehicle*. 2009 AIAA Guidance, Navigation, and Control Conference, Pasadena, California, September 15-17, 2009.
- [23] Perfect, P., White, M. D., and Jump, M., "Towards Handling Qualities Requirements for Future Personal Aerial Vehicles", Proceedings of the American Helicopter Society 69th Annual Forum, Phoenix, Arizona, May 21-23, 2013.
- [24] Tischler, M. B. and Remple, R. K., *Aircraft and Rotorcraft System Identification Engineering Methods with Flight Test Examples*, AIAA Education Series, 2012.

- [25] Hovakimyan, N. and Cao, C., 2010.  *$\mathcal{L}_1$  adaptive control theory: guaranteed robustness with fast adaptation* Vol. 21. Siam.
- [26] Xargay, E., Hovakimyan, N., and Cao, C.,  *$\mathcal{L}_1$  Adaptive Controller for Multi-Input Multi-Output Systems in the Presence of Nonlinear Unmatched Uncertainties*. 2010 American Control Conference, Marriott Waterfront, Baltimore, Maryland, June 30 - July 02, 2010.
- [27] Tischler, M. B., Ivler, C. M., Mansur, M. H., Cheung, K. K., Berger, T., Berrios, M.. *Handling-qualities optimization and trade-offs in rotorcraft flight control design*. In AHS Specialists Meeting on Rotorcraft Handling-Qualities, November, 2008, Liverpool, UK (pp. 4-6).
- [28] Pettersson, A., Åström, K. J., Robertsson, A., Johansson, R.. *Analysis of linear  $L1$  adaptive control architectures for aerospace applications*. 51st IEEE Conference on Decision and Control, Grand Wailea, Maui, Hawaii, December 10-13, 2012.