UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA

MASTER THESIS

Control Strategies of a Tilt-rotor UAV for Load Transportation

Marcelino Mendes de Almeida Neto

Belo Horizonte
2014

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA

MASTER THESIS

Control Strategies of a Tilt-rotor UAV for Load Transportation

Thesis submitted to the Programa de Pós-Graduação em Engenharia Elétrica, Escola de Engenharia,
in partial fulfillment of the requirements for the degree of
Master in Electrical Engineering
at the
Universidade Federal de Minas Gerais.

By

Marcelino Mendes de Almeida Neto

Belo Horizonte, August 2014

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA

MASTER THESIS

Control Strategies of a Tilt-rotor UAV for Load Transportation

Author: Marcelino Mendes de Almeida Neto
Advisor: Guilherme Vianna Raffo

Acknowledgements

First of all, I would like to thank my parents *Americo Almeida* and *Simone Teixeira*, who always supported me and were source of pride, admiration and reference for my personal e professional growth.

To my brothers *Anna Laura Almeida*, *Americo Almeida* and *Pedro Arthur Almeida*: friends for an entire life.

To *Ana Christine*, for cheering up even more my day-to-day.

Special thanks to Prof. *Guilherme Raffo* for guiding and assisting me on this extraordinary experience for the achievement of the Master Degree.

I thank Prof. *Bruno Adorno*, my informal co-advisor.

Thanks to Prof. *Patrícia Pena*, who helped me with my first steps in the university. She guided (and still guides) me both in personal and professional life. Not only a professor: a second mother.

I am very much thankful to Prof. *Ronaldo Pena*. He inspired me right at the beginning of my undergraduate engineering course and paved the way for my past and future steps.

To the members of the ProVant team, who were always on my side through this journey. Special thanks to *Werner Andrade*, who helped me designing some figures for my text.

To the members and cohabitants of LACSED: *Luis Henrique*, *Ramon Durães*, *Gustavo Vieira*, *Pedro Braga*, *Hugo Bravo*, among other passers-by.

To the *Programa de Pós Graduação em Engenharia Elétrica UFMG* and all its faculty for their teaching excellency. They are able to expand one's conceptions about engineering, life, the universe and everything rather than just providing *42* as an answer.

(...) Nobody is gonna hit as hard as life.
But it ain't about how hard you hit:
It's about how hard you can get hit and keep moving forward.
That's how winning is done!

Sylvester Stallone

Abstract of the Thesis submitted to the Programa de Pós-Graduação em Engenharia Elétrica, Escola de Engenharia, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering at the Universidade Federal de Minas Gerais.

Control Strategies of a Tilt-rotor UAV for Load Transportation

Marcelino Mendes de Almeida Neto

August / 2014

Advisors: Guilherme Vianna Raffo

Area of Concentration: Systems Engineering and Automation

Keywords: Unmanned Aerial Vehicles, Tilt-rotor, Load Transportation, $\mathcal{H}_\infty$ Control, Input-Output Feedback Linearization, Path Tracking, Kalman Filter

This dissertation presents control strategies to solve the problem of suspended load transportation by a Tilt-rotor Unmanned Air Vehicle (UAV) passing through a desired trajectory. For the present study, it is important for the aircraft to maintain itself and the load stable even in the presence of external disturbances, parametric uncertainties and measurement errors.

In general, a precise dynamic model of a system is needed in order to design advanced control strategies to it. Therefore, a rigorous dynamic model is derived for the Tilt-rotor UAV with suspended load using Euler-Lagrange formulation. After obtaining the model, it is then possible to design control laws that satisfy the desired specifications. Consequently, linear and nonlinear control laws are designed.

In order to design linear control laws, the system is linearized around its operation point. Two linear control laws are designed: one using D-stability control design and the second using simultaneous D-stability and minimization of the $\mathcal{H}_\infty$ norm.

As for the nonlinear control design, a three-level cascade strategy is proposed. Each level of the cascade system executes a control law through the method of input-output feedback linearization. Each one of these levels controls a different group of the system's state variables until the aircraft becomes fully stable. Two path tracking controllers are specified for this strategy. The first considers the load only as a disturbance and does not actuate to avoid its swinging. The second controller, on the other hand, seeks to find a compromise between path tracking and reducing the load's swing. At last, as proof of concept, the nonlinear strategy is modified so that the aircraft is able to stabilize an inverted pendulum.

For all the described control laws, it is considered that the physical measurements of the aircraft are precisely known in all time instants. However, every physical measure is subject to errors and uncertainties and one cannot always obtain a high sampling frequency when measuring process variables. Therefore, part of this work is dedicated to the study of uncertainties when measuring the position of the aircraft in a situation where the controller has a higher sampling frequency than the GPS. In face of this problem, the aircraft's position must be estimated while no new measurements are available taking also into the consideration the existence of disturbance inputs on the system. This whole problem is solved by using the Kalman Filter with Unknown Inputs.

Resumo da dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Escola de Engenharia, como um dos requisitos necessários para obtenção do título de Master in Electrical Engineering na Universidade Federal de Minas Gerais.

Control Strategies of a Tilt-rotor UAV for Load Transportation

Marcelino Mendes de Almeida Neto

August / 2014

Orientador:	Guilherme Vianna Raffo
Área de Concentração:	Engenharia Elétrica
Palavras-chave:	Veículos Aéreos Não Tripulados, Tilt-rotor, Transporte de Carga, Controle $\mathcal{H}_\infty$, Linearização por Realimentação de Saída, Seguimento de Trajetórias, Filtro de Kalman

Nessa dissertação são apresentadas estratégias de controle para solucionar o problema de transporte de carga suspensa ao longo de uma trajetória desejada por um Veículo Aéreo Não Tripulado (VANT) na configuração Tilt-rotor. Para o presente estudo, é importante que a aeronave seja capaz de manter tanto a si mesma quanto a carga transportada estável mesmo na presença de perturbações externas, incertezas paramétricas e erros de medição.

Em geral, é importante que se tenha um modelo dinâmico preciso do sistema para que se possa projetar estratégias de controle avançadas para o mesmo. Dessa forma, um modelo dinâmico para o VANT Tilt-rotor com carga suspensa é rigorosamente derivado usando a formulação de Euler-Lagrange. Com o modelo obtido, é possível então projetar leis de controle que satisfaçam as especificações desejadas. Para tanto, leis de controle lineares e não lineares são projetadas.

Para projetar leis de controle lineares, lineariza-se o sistema em torno do seu ponto de operação. Com o sistema linearizado, duas leis de controle são projetadas: uma por D-estabilidade e outra que leva em consideração D-estabilidade e a norma $\mathcal{H}_\infty$ simultaneamente.

Já para o projeto do sistema de controle não linear, uma estratégia em cascata é proposta. Cada nível do sistema em cascata executa uma lei de controle através do método de linearização por realimentação de saída, sendo considerados três níveis. Cada um desses níveis controla um grupo diferente de variáveis do sistema até que a aeronave esteja estável por completo. Para essa estratégia, dois controladores para seguimento de trajetória são especificados. O primeiro controlador considera a carga apenas como uma perturbação e não atua para impedi-la de balançar, preocupando-se apenas com o seguimento de trajetória. O segundo controlador, por sua vez, busca encontrar um compromisso entre seguir a trajetória e reduzir o balanço da carga. Por fim, como prova de conceito, a estratégia não linear é modificada de forma a fazer com que a aeronave estabilize um pêndulo invertido.

Para os projetos de controle descritos, considera-se que todas as medições físicas da aeronave são precisamente conhecidas em todos instantes de tempo. Entretanto, é sabido

que toda medição física é sujeita a erros e imprecisões e nem sempre é possível obter alta frequência de amostragem ao medir alguma variável de processo. Portanto, parte desse trabalho é destinada ao estudo de incertezas em medições de posição da aeronave em uma situação na qual o controlador possua frequência de amostragem superior à frequência de amostragem do GPS. Diante desse problema, deve-se estimar a posição da aeronave enquanto não se tem novas medições levando em consideração que existem entradas de perturbação no sistema. Esse problema é solucionado usando o Filtro de Kalman com Entradas Desconhecidas.

Contents

Contents i

List of Figures iv

List of Tables vii

Acronyms ix

Notation x

1 Introduction 1

1.1 Motivation . . . 1

1.1.1 Historical Background . . . 2

1.2 State of the Art . . . 8

1.2.1 Tilt-rotor UAV control . . . 8

1.2.2 Load Transportation . . . 10

1.3 Justification and Objectives . . . 11

1.4 Structure of the Text . . . 13

1.5 List of Publications . . . 14

2 Modelling 15

2.1 Introduction . . . 15

2.2 Generalized Coordinates . . . 16

2.3 Kinematics . . . 18

2.4 Dynamics using Euler-Lagrange Formulation . . . 18

2.4.1 Inertia Matrix . . . 19

2.4.2 Coriolis and Centripetal Matrix . . . 23

2.4.3 Gravity Force Vector . . . 23

2.4.4 Input Force Vector . . . 24

2.4.5 Drag Force Vector . . . 26

2.5 State-Space Representation of the System . . . 26

2.6 System Design Parameters . . . 27

2.7 Conclusions . . . 28

3 Linear Control Strategies **29**
3.1 Introduction . . . 29
3.2 Linear Control Systems Theory . . . 30
3.2.1 Controllability of Linear Systems . . . 30
3.2.2 Stability of Linear Systems . . . 31
3.2.3 Linear Matrix Inequalities . . . 32
3.2.4 D-stability . . . 32
3.2.5 Linear $\mathcal{H}_\infty$ Controllers . . . 35
3.3 Tilt-Rotor Linear Control . . . 37
3.3.1 Equilibrium Point and Linear Model . . . 38
3.3.2 Controllability of the Tilt-rotor UAV . . . 39
3.3.3 D-stable Controller Design . . . 39
3.3.4 D-Stable $\mathcal{H}_\infty$ Controller Design . . . 40
3.4 Simulation and Results . . . 41
3.5 Conclusions . . . 43

4 Nonlinear Control Strategies **47**
4.1 Introduction . . . 47
4.2 Nonlinear Control Theory . . . 48
4.2.1 Preliminary Theory on Nonlinear Systems . . . 49
4.2.2 Input-Output Feedback Linearization . . . 50
4.3 Tilt-rotor Nonlinear Control . . . 54
4.3.1 First-level Feedback Linearization . . . 54
4.3.2 Second-level Feedback Linearization . . . 58
4.3.3 Third-level Feedback Linearization . . . 61
4.3.3.1 Performance Improvement of the Load's Swing . . . 68
4.3.4 Inverted Pendulum Control Strategy . . . 69
4.4 Simulation Results . . . 71
4.4.1 Comparison between NLPT and NLLS . . . 71
4.4.2 Comparison between NLLS and Linear $\mathcal{H}_\infty$ . . . 72
4.4.3 Inverted Pendulum . . . 76
4.5 Conclusions . . . 78

5 State Estimation **89**
5.1 Introduction . . . 89
5.2 Technical Preliminaries and Notation . . . 90

5.3 System Modelling for State Estimation 90
5.4 Kalman Filter with Unknown Inputs 94
5.5 Simulation Results and Analysis 96
5.6 Conclusion 98

6 Conclusions **105**
6.1 Future Works 107

Bibliography **109**

A Theory on Robotics **113**
A.1 Rotation Matrices 113
A.2 Skew Symmetric Matrices 115
A.3 Euler-Lagrange Equations 116

List of Figures

1.1 Focke Achgelis FA-269 (Courtesy of Aviastar). 3
1.2 Transcendental 1-G (Courtesy of Aviastar). 3
1.3 McDonnell Aircraft Co. XV-1 compound helicopter (Courtesy of Aviastar). 4
1.4 Bell Helicopter Company XV-3 Tilt-rotor aircraft (Courtesy of Aviastar). . 4
1.5 Bell Helicopter XV-15 Tilt-rotor aircraft (Courtesy of Wikipedia). 5
1.6 Osprey V-22 Tilt-rotor (Courtesy of Wikipedia). 5
1.7 Augusta Westland AW609 Tilt-rotor (Courtesy of Wikipedia). 6
1.8 Cruise Speeds of Various Helicopters (Courtesy of FindTheBest). 6
1.9 Cruise Speeds of Various Airplanes (Courtesy of FindTheBest). 7
1.10 Bell Eagle Eye TR911X (Courtesy of BlueSkyRotor). 7
1.11 Lockheed-Martin's ARES concept (Courtesy of Lockheed Martin). 8
1.12 ProVant Model 1.0. 8
1.13 ProVant Model 2.0. 9
1.14 ProVant Model 2.0 with Suspended Load. 12

2.1 Tilt-rotor UAV. 16
2.2 Tilt-rotor UAV frames and variables definition. 17

3.1 $\mathcal{H}_\infty$ diagram block. 36
3.2 Allocation of poles via D-Stability. 40
3.3 Allocation of poles via D-Stability with minimization of the $\mathcal{H}_\infty$ norm. . . 41
3.4 System disturbances in function of time. 42
3.5 Path tracking of the aircraft for both D-stable and $\mathcal{H}_\infty$ controllers. 43
3.6 Tracking errors for D-stable and $\mathcal{H}_\infty$ controllers. 44
3.7 Body and Load angles for D-stable and $\mathcal{H}_\infty$ controllers. 45
3.8 Inputs of the system for D-stable and $\mathcal{H}_\infty$ controllers. 46

4.1 Nonlinear Feedback Linearization Cascade Control Strategy. 48
4.2 Inverted Pendulum Projections. 70
4.3 Path tracking of the aircraft for NLPT and NLLS. 72
4.4 Tracking error on NLPT and NLLS. 73
4.5 Body and Load angles on NLPT and NLLS. 74

4.6 Inputs of the system using NLPT and NLLS. 75
4.7 Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$. 76
4.8 Tracking error on NLLS and $\mathcal{H}_\infty$. 77
4.9 Body and Load angles on NLLS and $\mathcal{H}_\infty$. 78
4.10 Inputs of the system using NLLS and $\mathcal{H}_\infty$. 79
4.11 Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$ for a faster trajectory. . . . 80
4.12 Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$ for harsher disturbances. . 81
4.13 Tracking error on NLLS and $\mathcal{H}_\infty$ for Nonzero Initial Conditions. 82
4.14 Body and Load angles on NLLS and $\mathcal{H}_\infty$ for Nonzero Initial Conditions. . 83
4.15 Path tracking of the aircraft with inverted pendulum. 84
4.16 Aircraft's position and attitude for the inverted pendulum. 85
4.17 $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ in function of time for the inverted pendulum. 86
4.18 Inputs for the inverted pendulum. 87

5.1 System disturbances for the LKFUI simulation. 98
5.2 Path tracking of the aircraft for the LKFUI simulation. 99
5.3 Position estimation error in function of time for the LKFUI simulation. . . 100
5.4 Tracking error for the LKFUI simulation. 101
5.5 Body and Load angles for the LKFUI simulation. 102
5.6 Inputs of the system for the LKFUI simulation. 103
5.7 Disturbance estimation in function of time for the LKFUI simulation. . . . 104

List of Tables

2.1 System Parameters. 28

3.1 Mean-square-error comparison between D-stable and $\mathcal{H}_\infty$ controller. 42
3.2 IAVU Index comparison between D-stable and $\mathcal{H}_\infty$ controller. 42

4.1 Controllers parameters for the two first Feedback-Linearizations. 61
4.2 Controllers parameters for the whole cascade system. 68
4.3 Controllers parameters for load control. 69
4.4 Controllers parameters for the inverted pendulum. 71
4.5 Mean-square-error comparison between NLPT and NLLS. 72
4.6 IAVU Index comparison between NLPT and NLLS. 73
4.7 IAVU Index comparison between NLLS and $\mathcal{H}_\infty$. 76
4.8 Mean-square-error comparison between NLLS and $\mathcal{H}_\infty$. 77
4.9 IAVU Index for the Inverted Pendulum. 79

5.1 Translational controllers parameters for use with LKFUI. 98
5.2 Mean-square-error of the LKFUI simulation. 98

Acronyms

VTOL	Vertical Take-off and Landing
DH	Denavit-Hartenberg Convention
UAV	Unmanned Aerial Vehicle
DOF	Degrees of Freedom
FKM	Forward Kinematic Model
LMI	Linear Matrix Inequalities
MIMO	Multiple Input Multiple Output
FL	Feedback Linearization
ISFL	Input-State Feedback Linearization
IOFL	Input-Output Feedback Linearization
PFL	Partial Feedback Linearization
PD	Proportional-Derivative
PID	Proportional-Integral-Derivative
NLPT	Nonlinear Controller for Path Tracking
NLLS	Nonlinear Controller for Reduced Load Swing
GPS	Global Positioning System
LKFUI	Linear Kalman Filter with Unknown Inputs

Notation

Notation

a	Italic lower case letters denote scalars
$\boldsymbol{a}$	Boldface italic lower case letters denote vectors
$\boldsymbol{A}$	Boldface italic upper case letters denote matrices

Symbols

$\mathbf{0}_{n\times m}$	Zero matrix with n lines and m columns
$\boldsymbol{I}_n$	Identity matrix of dimension n
k	Number of samples
$\boldsymbol{x}$	Vector of n^{th} order, where x_i, $i = 1 \dots n$, $\boldsymbol{x} \in \Re^n$
$\boldsymbol{x}^T$	Transpose vector of $\boldsymbol{x}$
$\dot{\boldsymbol{x}}$	Time-derivative of $\boldsymbol{x}$
$\hat{\boldsymbol{x}}_{k\|j}$	Prediction of the state vector $\boldsymbol{x}$ at instant k using measured information up to instant j
$\boldsymbol{x_0}$	Initial condition of $\boldsymbol{x}$
$\boldsymbol{x_r}$	Reference vector of the variable $\boldsymbol{x}$
ε_q	Tracking error for the generalized variable q

Model Notation

f_R	Right rotor's thrust
f_L	Left rotor's thrust
τ_{α_R}	Tilting torque for the right rotor
τ_{α_L}	Tilting torque for the left rotor
$\mathcal{I}$	Fixed inertial frame
$\mathcal{B}$	Moving body frame
$\mathcal{C}_1$	Frame rigidly attached to the main body's center of mass
$\mathcal{C}_2$	Frame rigidly attached to the right rotor's center of mass
$\mathcal{C}_3$	Frame rigidly attached to the left rotor's center of mass
$\mathcal{C}_4$	Frame rigidly attached to the suspended load's center of mass
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\boldsymbol{\xi} = [x^{\mathcal{I}}\ y^{\mathcal{I}}\ z^{\mathcal{I}}]^T$	Translation between the origins of frames $\mathcal{I}$ and $\mathcal{B}$
$\boldsymbol{\eta} = [\phi\ \theta\ \psi]^T$	The main body attitude with respect to frame $\mathcal{I}$
α_R, α_L	Tilt angles of the right and left rotors, respectively
γ_1, γ_2	Degrees of freedom of the suspended load
β	Constant rotation of right and left rotors around $x_{\mathcal{C}_i}$, for $i = 1, 2$
$\mathbf{d}_i^{\mathcal{B}} = [d_{xi}^{\mathcal{B}}\ d_{yi}^{\mathcal{B}}\ d_{zi}^{\mathcal{B}}]^T$	Translation between the origins of frames $\mathcal{B}$ and $\mathcal{C}_i$, for $i = 1, 2, 3, 4$
l	Distance between frame $\mathcal{B}$ and the suspended load's center of mass
$\boldsymbol{q} \in \Re^{10}$	Generalized coordinates vector

$\boldsymbol{R}_A^B$	Rotation matrix from generic frame A to generic B
$\boldsymbol{p}_i^A$	point rigidly attached to frame $\mathcal{C}_i$ represented in generic frame A
$\boldsymbol{M}(\boldsymbol{q})$	Euler-Lagrange's inertia matrix
$\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$	Euler-Lagrange's coriolis and centrifugal forces matrix
$\boldsymbol{G}(\boldsymbol{q})$	Euler-Lagrange's gravitational force vector
$\boldsymbol{F}(\boldsymbol{q})$	Euler-Lagrange's independent input force vector
$\boldsymbol{F}_{ext}$	Euler-Lagrange's external unknown force vector
$\boldsymbol{F}_{drag}$	Drag force vector
$\boldsymbol{\mu}$	Drag coefficients matrix
K	Kinetic energy of the whole system
K_i	Kinetic energy of the i^{th} body
$\boldsymbol{v}_i^{\mathcal{I}}$	Velocity of the center of mass of body i with respect to frame $\mathcal{I}$
$\boldsymbol{\omega}_{BA}^B$	Angular velocity of generic frame B with respect to a generic frame A represented in frame B
$\boldsymbol{S}(\cdot)$	Skew symmetric matrix
ρ_i	Mass density of body i
m	Total mass of the helicopter with suspended load
m_i	Mass of the i^{th} body
$\boldsymbol{I}_i$	Inertia tensor of body i
$\boldsymbol{J}_i$	Inertia tensor of a rotated and displaced body
$\boldsymbol{W}_{\boldsymbol{\eta}}$	Euler matrix expressed in the body-fixed frame
P	Potential energy of the whole system
P_i	Potential energy of the i^{th} body
$\boldsymbol{g} = [0 \;\; 0 \;\; -gz]^T$	Gravity vector
gz	Gravity acceleration
b	Thrust coefficient of the rotors
k_τ	Drag coefficient of the propellers
$\boldsymbol{T}^{\mathcal{I}}$	Translational Force Vector
$\boldsymbol{\tau}$	Attitude Force Vector
$\boldsymbol{\Gamma}$	Control action vector
$\boldsymbol{B}(\boldsymbol{q})$	Force matrix (input coupling matrix)

Controllers Notation

$\boldsymbol{d}$	Exogenous disturbance vector
$\boldsymbol{u}$	Control effort vector
$\boldsymbol{A}$	Linear state matrix
$\boldsymbol{B}_u$	Linear input matrix
$\boldsymbol{B}_g$	Linear external input matrix
$\boldsymbol{K}$	Control matrix
$V(\boldsymbol{x})$	Lyapunov function
γ	Attenuation level of the $\mathcal{H}_\infty$ problem
$H(s)$	Transfer function between the input signal and the output signal z
$\|H(s)\|_\infty$	$\mathcal{H}_\infty$-norm of the transfer function $H(s)$
$\boldsymbol{C}_z,\ \boldsymbol{D}_{uz},\ \boldsymbol{D}_{wz}$	Weighting matrices of the linear $\mathcal{H}_\infty$ controller

$\boldsymbol{f}(\boldsymbol{x})$	Nonlinear drift vector field
$\boldsymbol{g}_u(\boldsymbol{x})$	Nonlinear steering vector field
$\boldsymbol{g}_d(\boldsymbol{x})$	Nonlinear external steering vector field
$\boldsymbol{h}(\boldsymbol{x})$	Nonlinear output vector field
$\boldsymbol{D}_{\boldsymbol{f}}$,	Jacobian matrix associated to the vector field $\boldsymbol{f}(\boldsymbol{x})$
r_i	Relative degree of the i^{th} output
$\boldsymbol{\Phi}(\boldsymbol{x})$	Diffeomorphism function
$\boldsymbol{z}$	Transformed state-space vector
$\boldsymbol{v}$	Transformed input vector
$\boldsymbol{\vartheta}$	Internal dynamics state-space vector
$\boldsymbol{q}_c$	Controllable variables vector
$\boldsymbol{q}_u$	Uncontrollable variables vector
$\sigma(p)$	Saturation function

State Estimation Notation

T_s	GPS's sampling time
τ_s	Controller's sampling time
$\boldsymbol{w}_k$	Process noise
$\boldsymbol{Q}$	Process noise's covariance matrix
$\boldsymbol{v}_k$	Measurement noise
$\boldsymbol{R}_k$	Measurement noise's covariance matrix
$\mathcal{E}[\cdot]$	Expected value operation
$\boldsymbol{L}_k$	Kalman Filter's gain matrix
$\boldsymbol{e}_{k\|k-1}$	Forecast error
$\boldsymbol{\nu}_{k\|k-1}$	Innovation
$\boldsymbol{e}_{k\|k}$	Data-assimilation error
$\boldsymbol{P}^{xx}_{k\|k-1}$	Forecast error covariance
$\boldsymbol{P}^{yy}_{k\|k-1}$	Innovation covariance
$\boldsymbol{P}^{xy}_{k\|k-1}$	Cross covariance
$\boldsymbol{P}^{xx}_{k\|k}$	Data-assimilation error covariance

Chapter 1

Introduction

Sumário

1.1 Motivation 1

1.1.1 Historical Background 2

1.2 State of the Art 8

1.2.1 Tilt-rotor UAV control 8

1.2.2 Load Transportation 10

1.3 Justification and Objectives 11

1.4 Structure of the Text 13

1.5 List of Publications 14

1.1 Motivation

It is a moment of great development on Unmanned Aerial Vehicles (UAV) research. Many fields of control and robotics, such as sensor fusion, computer vision, rapid prototyping, state estimation and control methodologies improved the performance of this kind of systems. This allowed UAV's to be produced with low cost, becoming accessible to numerous research facilities and hobbyists all around the world.

As for the applications, there is a vast list of UAV uses: cargo delivery, surveillance, field recognition, cave exploration, cinematographic filming, military purposes, water and spray over plantations, 3-D mapping, search and rescue, wildlife research, among many others. In short, UAVs are being increasingly used for diverse civil and military purposes.

Presently, the most commonly studied UAVs are helicopters, quadcopters and fixed-wing airplanes. Usually, helicopter-like aircrafts have the advantage over the airplanes for performing Vertical Take-off and Landing (VTOL). On the other hand, airplanes are able to fly at higher speeds than helicopters. In order to combine the advantages of these kind of aircrafts, present research on UAVs is looking into the direction of the Tilt-rotor UAV, a hybrid copter-plane aircraft. The Tilt-rotor is a type of aircraft that combines the vertical lift capacity of helicopters with the range and speeds of fixed-wing airplanes. As its name suggests, the Tilt-rotor UAV uses tiltable rotating propellers for lift and propulsion.

When compared to Quadrotor and Helicopter UAVs, the Tilt-rotor present the following advantages:

- Birotor helicopters are driven by two rotors, which reduces each rotor's size keeping the same payload when compared to a helicopter with main propeller. In comparison with quadcopters, Tilt-rotors are smaller and consume less energy.
- Tilt-rotors do not need mechanical coupling actuating on its propellers. This simplifies the project, reduces maintenance, and, consequently, decreases the vehicle's cost.
- The simplicity of the mechanical design provides movement control through direct transmission to the rotors, varying their speed and tilting them. In a standard helicopter, the angular velocities of the propellers are usually constant and its movement is controlled by changing the angle of attack of the blades. This requires transmission between the rotors as well as precision mechanical devices to change these angles.

From the control systems perspective, the construction of this kind of UAV is far from simplifying the problem. The torques and forces necessary to control the system are applied not only by aerodynamic effects, but also through the coupling effect that occurs between the dynamics of the rotors and the aircraft's body. This fact, together with the uncertainties of the model, especially in high frequency bands, makes this system even more complex to be controlled than a standard helicopter, specially when using classic control techniques.

It is worth pointing out that the Tilt-rotor UAV is an underactuated mechanical system, i.e. it possesses less inputs than degrees of freedom. This is result of a trade-off made between electromechanical design, where it is desired to minimize the number of actuators - causing weight reduction, lower cost, less energy consumption - of the system, and control design, where more actuators simplify the project. Consequently, techniques that are commonly used for fully actuated systems cannot be applied to this kind of systems, given that most underactuated systems are not fully linearizable. Therefore, it is common to employ nonlinear modelling and control techniques to underactuated systems (e.g. aircrafts) so as to reach high-performance flight in specific conditions such as: hover, take off, land, etc (Castillo et al., 2005c).

1.1.1 Historical Background

Historically (Maisel et al., 2000), the first attempt to develop a functional Tilt-rotor started in Germany in 1942 with the Focke Achgelis FA-269 (figure 1.1). However, this project was abandoned after a full-scale mockup was destroyed during a bombing in the Second World War. The idea of a Tilt-rotor intrigued two american enterprising engineers; Dr. Wynn Laurence LePage and Haviland Hull Platt (Platt-LePage Aircraft Company of Eddystone, Pennsylvania) decided to produce the Platt-LePage XR-1A. Again, the design was never developed, but granted a patent for the concept in July, 1950. While Platt and Lepage were patenting their ideas, the Transcendental Aircraft Corporation initiated work on the Model 1-G in 1947. The Model 1-G was the world's first Tilt-rotor-like aircraft to fly and premiered in June, 1954 (figure 1.2). However, after more than 100 successful flights, the Model 1-G had reached partial conversions to within 10° of the airplane mode but crashed on July, 1955. A Transcendental Model 2-G was subsequently developed,

but the American Air Force withdrew funding support to the enterprise and the program terminated in 1957.

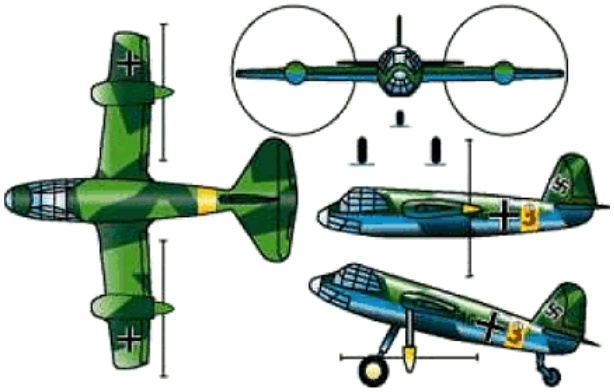

Figure 1.1: Focke Achgelis FA-269 (Courtesy of Aviastar).

Figure 1.2: Transcendental 1-G (Courtesy of Aviastar).

Going back in time, in 1950 the United States started the Convertiplan Program (Maisel et al., 2000). The program required someone to build an aircraft that could maintain significant hover duration, low speed manoeuvring and agility, and higher speeds than conventional helicopters. Three designs were selected from a design competition: the *XV-1* compound helicopter (proposed by McDonnell Aircraft Co.), the *XV-2* stoppable rotor aircraft (proposed by Sikorsky Aircraft) and the *XV-3* Tilt-rotor aircraft (submitted by the Bell Helicopter Company). The *XV-2* did not survive the initial evaluation phase and the *XV-1* (figure 1.3) experienced severe oscillatory load conditions when performing its initial tests in 1955. The *XV-3* (figure 1.4) initially had some instability problems that caused a hard landing with minor damage in 1955 and a serious crash in 1956. After a big effort of the engineering crew, in December 1958 the *XV-3* accomplished the goal of completing a dynamically stable full conversion to the airplane mode.

The *XV-3* flew throughout thirteen years of restless tests. However, the aircraft did not reach high speeds as expected for a Tilt-rotor; it could reach only 212 km/h, a common speed for helicopters of the time. Moreover, the aircraft had stability problems and it was feared that it could be destroyed during flight. As a result, the Convertiplan Program closed and gave space for the XV-15 program, which was launched in 1971 at NASA Ames Research Center with the objective of research and development of prototype designs of an improved Tilt-rotor aircraft. Two contracts were issued in 1972: one to Bell Helicopter

Figure 1.3: McDonnell Aircraft Co. XV-1 compound helicopter (Courtesy of Aviastar).

Figure 1.4: Bell Helicopter Company XV-3 Tilt-rotor aircraft (Courtesy of Aviastar).

(Bell Model 301) and the other to Boeing-Vertol (Model 222). In 1973 NASA chose the Bell Model 301 for further development and the *Bell XV-15* (figure 1.5) first flew in 1977. The *XV-15* reached 456 km/h, proving its concept.

With the success of the *XV-15*, the United States Department of Defense started the Joint Advanced Lift Aircraft program in 1982[1]. The aim of the program was to develop a military aircraft that could take off and land vertically and also carry combat troops at speed. Bell Helicopter teamed with Boeing Vertol submitting a proposal for an enlarged version of the *Bell XV-15*. In 1986, the so-called *Osprey V-22* (figure 1.6) started to be produced in full-scale. The *Osprey V-22* was target of controversy on next years mainly due to its production cost and safety incertitude. Therefore, it was only in 2000 that the United States Marine Corps began crew training for the *Osprey V-22*, fielding it in 2007.

In 1998 Bell Helicopters started a joint venture with AugustaWestland, establishing the Bell/Augusta Aerospace Company (BAAC). Their objective was to design the *BA609*, a civil version of the *Osprey V-22*. The *BA609* first flew in 2003 and its first conversion from helicopter to airplane took place in 2005. In 2011 AugustaWestland assumed full

[1]Further information is provided on http://www.globalsecurity.org/military/systems/aircraft/v-22-history.htm

Figure 1.5: Bell Helicopter XV-15 Tilt-rotor aircraft (Courtesy of Wikipedia).

Figure 1.6: Osprey V-22 Tilt-rotor (Courtesy of Wikipedia).

ownership of the program, renaming the aircraft to *AW609* (figure 1.7). AugustaWestland is now working with the International Civil Aviation Organization to ensure that the regulatory framework is in place before the *AW609* starts operating in commercial marketplace in 2017[2].

Figure 1.8 compare the speeds of the *Osprey V-22* and the *AW609* with some other helicopters [3]. It can be seen that the cruise speed of Tilt-rotors exceed by far the cruise speed of other helicopters. On the other side, Tilt-rotors are not as fast as most airplanes, as shown on figure 1.9[4].

After the successful development of the *Osprey V-22*, Bell Helicopters started the Eagle Eye program in 1993: the objective was to develop a small scale Vertical Takeoff Unmanned Aerial Vehicle. In 1998 the *TR911X* (figure 1.10) was tested and approved in

[2]Further information is provided on http://www.agustawestland.com/news/agustawestland-completes-first-customer-demonstration-aw609

[3]The equivalence between *knots* and km/h is: $1\,knot = 1.852\,km/h$.

[4]Figures extracted from http://www.findthebest.com/

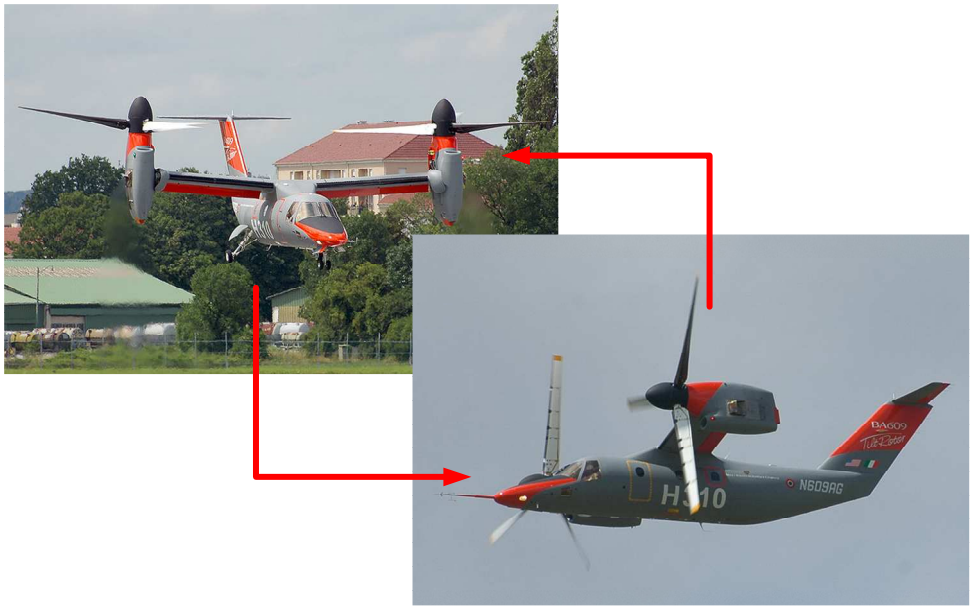

Figure 1.7: Augusta Westland AW609 Tilt-rotor (Courtesy of Wikipedia).

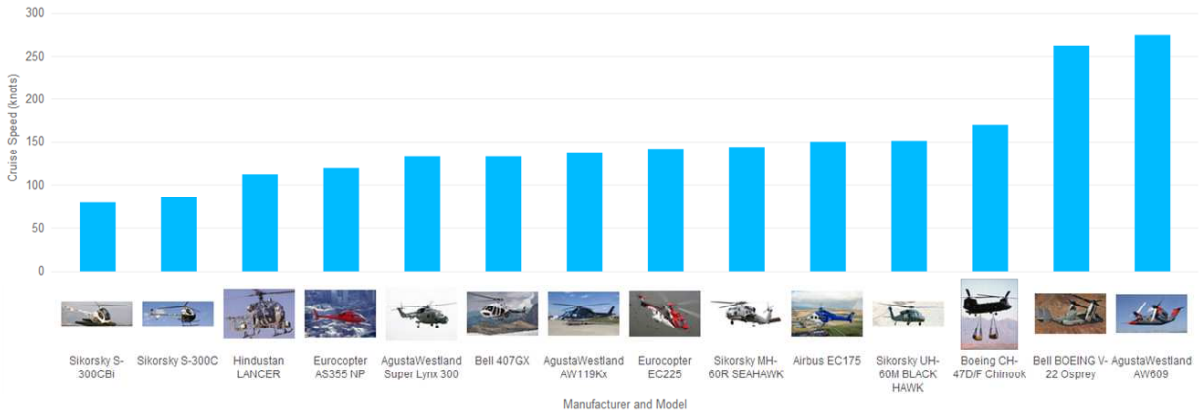

Figure 1.8: Cruise Speeds of Various Helicopters (Courtesy of FindTheBest).

land operations, reaching 370 km/h with an endurance of approximately 6 $hours$ with a 90.7 kg payload[5]. The *TR911X* is then the first Tilt-rotor UAV ever created.

In 2010, the American's Defense Advanced Research Projects Agency (DARPA) initiated the DARPA TX program, also known as the Transformer program. In 2013, DARPA selected the Lockheed Martin's Aerial Reconfigurable Embedded System (ARES - figure 1.11) design concept to move forward. The ARES[6] is a Tilt-rotor UAV capable of attaching to vehicles, providing flexible, terrain-independent transportation for logistics, personnel transportation and tactical support missions for small ground units.

At the beggining of last decade, Tilt-rotor UAVs became subject of study in many universities around the world, and some of them even designed small-scale prototypes for test. The first one found in literature was developed at the Universite de Technologie de

[5]Further information is provided on http://www.naval-technology.com/projects/belleagleeyeuav/
[6]Further information is provided on http://www.lockheedmartin.com/us/products/ares.html/

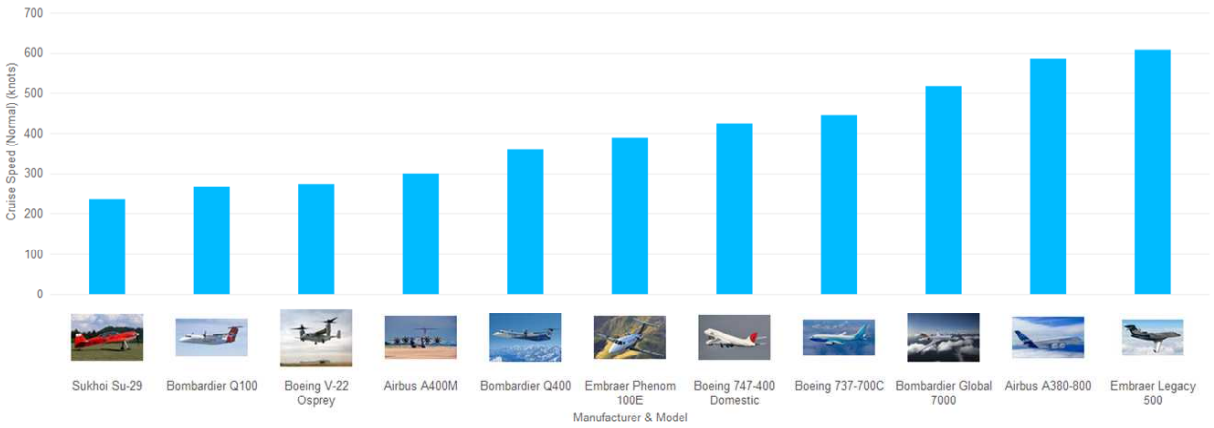

Figure 1.9: Cruise Speeds of Various Airplanes (Courtesy of FindTheBest).

Figure 1.10: Bell Eagle Eye TR911X (Courtesy of BlueSkyRotor).

Compiegne, named BIROTAN (BI-ROtors with tilting propellers in TANdem) (Kendoul et al., 2005). Since then, many other designs appeared, namely: the Arizona State University's Tilt-wing HARVee (High-Speed Autonomous Rotorcraft Vehicle) (Dickeson et al., 2007); the T-Phoenix (Sanchez et al., 2008); the Tilt-rotor of the Korea Aerospace Research Institute (Lee et al., 2007); the Tilt-rotor of the Nanjing University of Aeronautics and Astronautics - China - (Yanguo and Huanjin, 2009); the UPAT Tilt-rotor (Papachristos et al., 2011b), among others.

Even though many universities proposed to develop and control Tilt-rotor-like aircrafts, the literature is still very poor on experimental results. It is in this context that the Brazilian universities Universidade Federal de Minas Gerais (UFMG) and Universidade Federal de Santa Catarina (UFSC) started the ProVant project (Gonçalves et al., 2013). The aim of the project is to develop a fully open-source small scale Tilt-rotor aircraft capable of performing autonomous flights and waypoint navigation. In 2013 UFSC's ProVant team designed the Tilt-rotor 1.0, which is now on phase of assemblage (figure 1.12). In 2014 the UFMG's ProVant team came up with the design of the Tilt-rotor 2.0 (figure 1.13), which is also being assembled.

Figure 1.11: Lockheed-Martin's ARES concept (Courtesy of Lockheed Martin).

Figure 1.12: ProVant Model 1.0.

1.2 State of the Art

This section presents some literature review about control of Tilt-rotor UAVs and the problem of load transportation. To the best knowledge of the author, there are no articles dealing with load transportation using a Tilt-rotor.

1.2.1 Tilt-rotor UAV control

Until the year of 2005, only a small quantity of research works were published assessing control of Tilt-rotor aircrafts. Most of the previous research and published work related to Tilt-rotors derived from issues and problems during aircraft development (Kleinhesselink, 2007).

Then, one of the first approaches to design and control a small-scale Tilt-rotor for research purposes came with Kendoul et al. (2005). This work based on the ideas of Gress (2002) and intended to design the BIROTAN, a Tilt-rotor where each rotor has

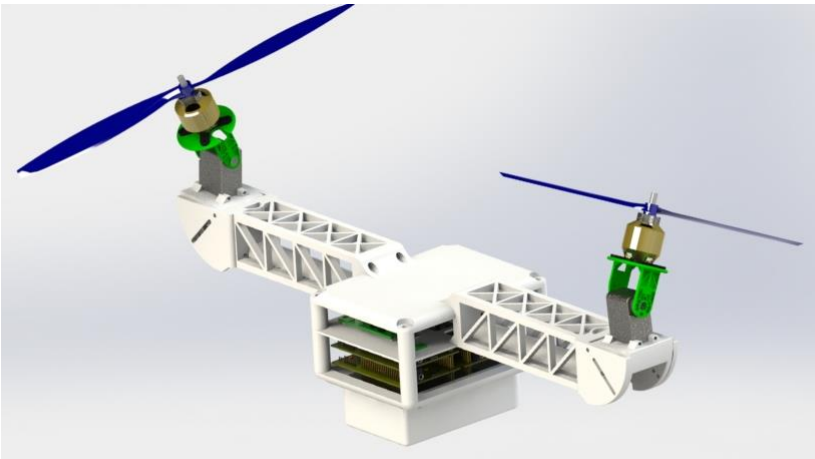

Figure 1.13: ProVant Model 2.0.

two degrees of freedom, instead of only one as used on the *Osprey V-22* and *AW609*. However, the BIROTAN project was abandoned due to the difficulty of implementing the two degrees of freedom on each rotor of the real aircraft. A few years later, the researchers designed a new Tilt-rotor called T-Phoenix (Sanchez et al., 2008), with only one degree of freedom in each rotor. They designed a control law using nonlinear control in the vicinity of the equilibrium point and experimental results were successful on maintaining the aircraft in hovering. In order to design the controller, a simplified dynamic model was derived for the system.

Lee et al. (2007) explored the use of gain-scheduling to control a Tilt-rotor's roll and pitch by choosing a vast number of linearization points. This result is part of a Korean initiative to develop a Tilt-rotor UAV at the Korea Airspace Research Institute - KARI. A linear controller was designed for each of the linearization points with use of an optimization method called 'particle swarm optimization'. In each controller, the objective function was to maximize gain and phase margin while allocating poles within a defined region of the complex plane. Simulation results showed that the system follows desired trajectories but unfortunately the authors do not provide any information regarding the system's model used for controller design and simulation. KARI's Tilt-rotor UAV flew in 2013, with results being presented by Kang et al. (2013).

Dickeson et al. (2007) also proposed a gain-scheduling controller to command a Tilt-wing aircraft. [7] However, each controller maximized disturbance rejection ($\mathcal{H}_\infty$ control) instead of phase and gain margins. Again, the system model used for simulation was not provided on the paper, but it was referenced as obtained from Kinder and Whitcraft (2000) and Mix and Seitz (2000).[8]

Yanguo and Huanjin (2009) propose a full control of a Tilt-rotor UAV by means of cascade control. The inner loop controls the attitude of the aircraft using feedback linearization, while the outer loop controls the position of the aircraft. However, the

[7] Tilt-wing aircrafts are similar to Tilt-rotors with the difference that Tilt-wings tilt both the rotors and the wings, while a Tilt-rotor tilt only its rotors.

[8] These two papers cannot be accessed from the Periodicos CAPES - both papers were published on the American Institute of Aeronautics and Astronautics - AIAA.

feedback linearization was done using a numerical model obtained from tests in a wind-tunnel, which requires identification of the aircraft's model for each different Tilt-rotor. Simulation results look promising, but the authors do not provide the experimental results allegedly obtained.

The work of Dhaliwal and Ramirez-Serrano (2009) presented an approach to control a Tilt-rotor UAV using Fuzzy Logic Control. First, a controller was designed using a trial-and-error method, whose simulations showed that it was possible to maintain the aircraft stable in hovering position but it could become unstable in some other configurations. In order to solve that, the authors designed a different controller using an optimization module that could satisfy some constraints such as stability or path tracking. Although intuitive and effective, it is hard to guarantee stability when using approaches like Fuzzy Logic Control.

Papachristos et al. (2011a) studies a Model Predictive Control for the attitude of a Tilt-rotor aircraft. Simulation results showed that the controller was successful on controlling the attitude of the aircraft, while rejecting disturbance inputs. However, the researchers used a simplified dynamic model of the Tilt-rotor both for simulations and controller design.

Papachristos et al. (2011b) made an adaptation from the previously mentioned work of Sanchez et al. (2008). The model was improved so as to include coupling gyroscopic body effects. The control design is similar but the experimental results seem to be less stable. However, given that the mechanical and hardware architectures of both works are very different, it is not possible to compare their controller designs solely from experimental results.

Bhanja Chowdhury et al. (2012) derived a simplified Euler-Lagrange model for the Tilt-rotor UAV and used it to design a Back-stepping control strategy for the aircraft. Simulation results showed that the aircraft could stabilize and respond to step references even when starting far from the equilibrium point.

A similar approach using Back-stepping control was done by Amiri et al. (2013), using the model derived at Amiri et al. (2011). This work considered that the tilting propellers have two tilting degrees of freedom.

1.2.2 Load Transportation

When comparing to the literature of Tilt-rotor, there is a wider group of publications that explore the problem of load transportation. If one is able to control the attitude of an aircraft, then the system aircraft with load can be approximated to a crane system where it is possible to actuate with forces on three Cartesian axes so as to regulate both position and load swing. This subsection presents some recent works assessing both control of crane systems and load transportation using aircrafts.

Moon et al. (2012) and Lee et al. (2013) both introduce nonlinear controllers that stabilize overhead cranes performing the task of load transportation. The first work used a Sliding Mode Controller which adapts along with time while estimating some unknown parameters. The latter work, on the other hand, used a Partial Feedback Linearization

approach so as to obtain a stable closed-loop system, supported by experimental results. Both controllers were designed based on a detailed Euler-Lagrange model of the system.

Faust et al. (2013) used a Quadrotor UAV to transport a suspended load from one desired point to another. In order to avoid load swing, the researchers used a machine learning approach without making any assumption about the dynamics of the system. Their experiment is separated in two phases: in the learning phase, the system learn the value function approximation for a particular load. Once the value function is learned, it is used to generate the trajectories for the aircraft to accomplish the goal while minimizing load swing on its final position.

Dai et al. (2014) also uses a Quadrotor UAV to stabilize the swing of a suspended load. However, in this work the nonlinear model of the whole system is considered. A load with unknown mass is attached to the aircraft by a chain of n links with unknown masses. The authors compare three control approaches: one using classic PD controllers, another using PID and a third one combining PD with Retrospective Cost Adaptive Control (RCAC). The idea of applying the RCAC was to adapt the controller while estimating the masses of the load and the chain links. Simulation results show that RCAC+PD responded faster than PID in the presence of mass uncertainties. Although more rigorous, the use of many links approach might have some implementation drawbacks, given that each link's position should be constantly measured, instead of measuring only the position of the load.

Sreenath et al. (2013) generates trajectories to a Quadrotor UAV so that a suspended load passes through a desired trajectory. This work proves that a Quadrotor with suspended load is differentiably flat and explores this property to propose controllers for the system that can either track the Quadrotor attitude, the load attitude or the load position. They derive equations to control the load position and minimize its sixth derivative, insuring minimum snap motion for the Quadrotor. Experimental results are presented where the load follows the desired path while the aircraft minimizes its motion along with time.

If one considers that the load is attached to the aircraft by a rigid string and its initial position is right above the aircraft, the system becomes an inverted pendulum. Controlling an inverted pendulum, because of its inherent instability, is more complicated than controlling a swinging load. However, both systems are similar and so are their control designs. Therefore, its worthwhile mentioning the work of Hehn and D'Andrea (2011), who used a Quadrotor UAV to stabilize an inverted pendulum[9].

1.3 Justification and Objectives

The present work focus on the development of control laws for the Tilt-rotor UAV with the further requirement that it should pass through a desired trajectory carrying a suspended load, as illustrated in figure 1.14. For the present study, it is important for the aircraft to maintain itself and the load stable even in the presence of external disturbances, parametric uncertainties, unmodelled dynamics and measurement errors. Only the helicopter

[9]Experimental results are shown on http://www.idsc.ethz.ch/people/staff/hehn-m

mode of the Tilt-rotor is assessed and results are presented via simulations, given that the Tilt-rotor UAV is not yet ready for flight in the ProVant project.

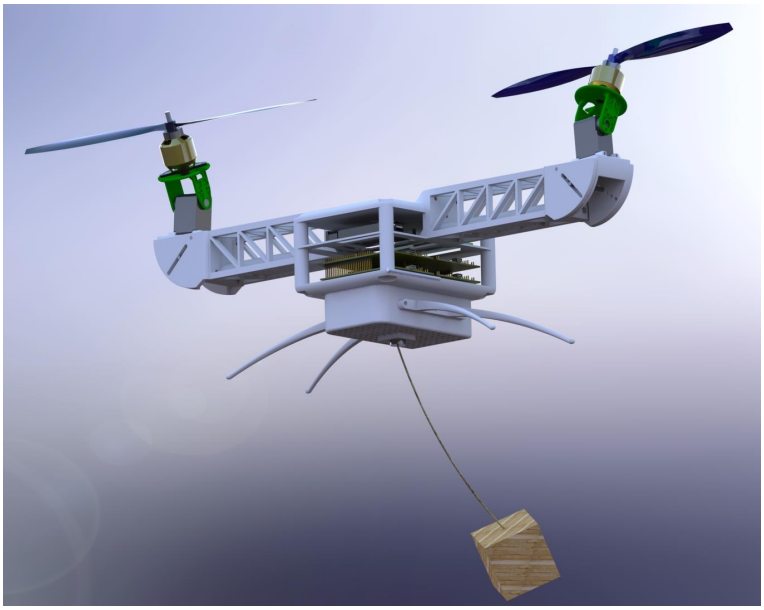

Figure 1.14: ProVant Model 2.0 with Suspended Load.

Among all cited works on the previous section [10], only Sanchez et al. (2008), Papachristos et al. (2011b) and Kang et al. (2013) presented experimental results to the control of a Tilt-rotor UAV. The first two achieved the goal of keeping the Tilt-rotor in hovering while using nonlinear controllers designed to work in the vicinity of their equilibrium point (hover); the third work presented results for all flight envelope, but did not provide much information about its model or controller design. All other papers that evaluated closed-loop control of the Tilt-rotor used a simplified—or did not provide any—system model, which means that their simulation results might not be consistent with reality.

Therefore, in the present work, a rigorous dynamic model for the aircraft is obtained using Euler-Lagrange formulation. The tilting angles are considered as state variables of the system, instead of inputs. These angles, in turn, are actuated by input torques applied by a pair of servomotors, obtaining a system affine in the inputs. The dynamic equations for the suspended load are also introduced in the same model. Dynamic and gyroscopic coupling between position motion, attitude motion, load swinging and tilting angles variation are considered. Thus, one should expect more trustworthy simulation results than previously published works.

As for the control system, linear and nonlinear control laws are designed. In order to design linear control laws, the system is linearized around its operation point. Two linear control laws are designed: one using D-stability control design and the second using simultaneous D-stability and minimization of the $\mathcal{H}_\infty$ norm. These controllers provide stabilization of all state variables (including the load) and their performances are evaluated and compared.

[10] Given that the ProVant project aims to design a Tilt-rotor with only one degree-of-freedom on its tilting propellers, further analysis solely considers works with similar designs. The introduction of a second degree of freedom widely simplifies the problem of controlling the Tilt-rotor UAV.

When dealing with nonlinear control design, a three-level cascade strategy is proposed. Each level of the cascade system executes a control law through the method of input-output feedback linearization. The two innermost levels are responsible for controlling attitude and altitude of the aircraft. The third level uses theory of load transportation to stabilize the rest of the system. Two path tracking controllers are specified for this strategy. The first considers the load only as a disturbance and does not actuate to avoid its swinging. The second controller, on the other hand, seeks to find a compromise between path tracking and reducing the load's swing, based on the approach of Lee et al. (2013). Performance is evaluated and compared for linear and nonlinear controller designs.

As proof of concept, the nonlinear strategy is slightly modified so that the aircraft is able to stabilize an inverted pendulum.

Furthermore, the analysis of the Tilt-rotor UAV's performance in the presence of measurement uncertainties with low sampling frequency is studied, a common problem when using GPS measurements. In face of this problem, the aircraft's position must be estimated while no new measurements are available taking also into the consideration the existence of disturbance inputs on the system. With use of the Kalman Filter with Unknown Inputs, it is possible to estimate the aircraft's position with higher precision, helping the aircraft to accomplish the task of path tracking with low tracking error.

The objectives of this work is summarized as follows:

- Model the Tilt-rotor UAV with suspended load using Euler-Lagrange formulation.
- Development of linear and nonlinear control strategies for path tracking of the Tilt-rotor aircraft with suspended load. The control laws should be robust to input disturbances, parametric uncertainties and unmodelled dynamics.
- Investigation of the system's performance in presence of measurement uncertainties with low sampling rate.

1.4 Structure of the Text

The thesis is organized as follows:

- **Chapter 2** derives the equations of motion for the Tilt-rotor UAV with suspended load using Euler-Lagrange formulation. From these equations, an state-space representation for the system is introduced. Parameters of the Tilt-rotor UAV used in this dissertation are presented.
- **Chapter 3** deals with closed-loop linear control for the Tilt-rotor UAV with suspended load. Two linear control strategies are derived in the vicinity of the aircraft's equilibrium point: D-stability and D-stability with minimization of the $\mathcal{H}_\infty$ norm. Simulation results are provided so as to show the effectiveness of each designed strategy, comparing them.

- **Chapter 4** deals with closed-loop nonlinear control for the Tilt-rotor UAV with suspended load. A nonlinear control strategy is developed based on a cascade scheme with three input-output feedback linearization blocks, where each block stabilizes a given quantity of state variables until the whole system is stable. Three control solutions are provided: the first considers the load only as a disturbance and does not actuate to avoid its swinging. The second solution, on the other hand, seeks to find a compromise between path tracking and reducing the load's swing. The third solution is used for stabilization and path tracking of an inverted pendulum. Simulation results are provided so as to show the effectiveness of the designed strategies, also comparing them with the linear approaches of Chapter 3.

- **Chapter 5** presents a strategy for position estimation of the aircraft in the presence of measurement uncertainties with low sampling rate. A simplified linear model is derived for the aircraft's translational motion so that it can be used to estimate the Tilt-rotor UAV's position using the Linear Kalman Filter with Unknown Inputs algorithm. An interesting advantage of this algorithm is that it estimates disturbance inputs on the system, incorporating this information on the position estimation itself.

- **Chapter 6** summarizes the contributions and results presented in this dissertation and suggests possible future research lines.

1.5 List of Publications

The following scientific works were accepted for publication during the elaboration of this dissertation:

Conference papers:

1. (Almeida et al., 2014a) M. M. de Almeida Neto, R. Donadel, G. V. Raffo, and L. B. Becker. Full control of a tiltrotor uav for load transportation. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014. To be published.

2. (Almeida et al., 2014b) M. M. de Almeida Neto, L. Schreiber, and G. V. Raffo. Robust state estimation for uavs - a comparison study among a deterministic and a stochastic approach. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014. To be published.

3. (Donadel et al., 2014a) R. Donadel, M. M. de Almeida Neto, G. V. Raffo, and L. B. Becker. Path tracking control of a small scale tiltrotor unmmaned aerial vehicle. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014. To be published.

Chapter 2

Modelling

Sumário

2.1 Introduction 15
2.2 Generalized Coordinates 16
2.3 Kinematics 18
2.4 Dynamics using Euler-Lagrange Formulation 18
2.4.1 Inertia Matrix 19
2.4.2 Coriolis and Centripetal Matrix 23
2.4.3 Gravity Force Vector 23
2.4.4 Input Force Vector 24
2.4.5 Drag Force Vector 26
2.5 State-Space Representation of the System 26
2.6 System Design Parameters 27
2.7 Conclusions 28

2.1 Introduction

This chapter focuses on modeling the Tilt-Rotor UAV with suspended load. The equations of motion are adapted from Donadel et al. (2014b), who uses Euler-Lagrange formulation to obtain the differential equations of the aircraft (without the suspended load). Their derivation is adapted so as to include the degrees of freedom of the hanging weight.

The Tilt-Rotor UAV with suspended load is a multibody system composed of four rigid bodies (see Figure 2.1):

- Main body - composed of a carbon-fiber structure, a landing gear, a battery, and a group of electronic devices;
- Two thrusters groups - one on each side of the aircraft (servomotors with rotors), interconnected to the main body by actuated revolute joints;
- Suspended load - load attached to the main body via a rigid pole with negligible mass.

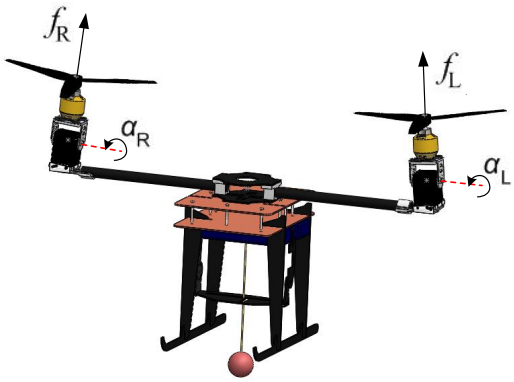

Figure 2.1: Tilt-rotor UAV.

The movements on the aircraft result from the thrusts f_R and f_L generated by the rotors and from the amplitude of the rotation angles α_R and α_L of the servomotors. Pitch motions can be achieved by equally varying α_R and α_L, given that the thrusts f_R and f_L are non-zero. Lateral displacements (i.e. roll motions) can be obtained by applying a thrust f_R different from f_L. Yaw movements are performed by tilting α_R in the opposite direction of α_L (again, given that the thrusts f_R and f_L are non-zero). Vertical motion is obtained by equally varying f_R and f_L, given that α_R and α_L angles are lower than $\frac{\pi}{2}$rad.

It should be noted that the two propellers rotate in opposite directions in relation to each other. This solution helps on reducing yaw motion caused by propellers' drag. This statement will be clarified when deriving the equations of motion in the next subsections.

Section 2.2 presents the generalized coordinates of the system. Section 2.3 derives the Forward Kinematic Model for each of the UAV's bodies with respect to the inertial frame. Section 2.4 derives the dynamic model using Euler-Lagrange formulation. Section 2.5 presents the system's dynamic representation on the form of nonlinear state-space equations. Finally, section 2.7 brings a conclusion about what was presented in the whole chapter. Further details on robotics theory can be seen in Appendix A.

2.2 Generalized Coordinates

This section defines the frames and generalized coordinates for the Tilt-Rotor UAV.

Consider the frames shown in Figure 2.2. There is a fixed inertial frame $\mathcal{I}$, a moving frame $\mathcal{B}$ rigidly attached to the main body, a frame $\mathcal{C}_1$ rigidly attached to the main body's center of mass, a frame $\mathcal{C}_2$ rigidly attached to the rotation axis of the right servomotor, a frame $\mathcal{C}_3$ rigidly attached to the rotation axis of the left servomotor, and a frame $\mathcal{C}_4$ rigidly attached to the center of mass of the suspended load. It is assumed that the rotation axes of both rotors coincide with their respective center of masses.

Moreover, $\boldsymbol{\xi} = [x^{\mathcal{I}} \quad y^{\mathcal{I}} \quad z^{\mathcal{I}}]^T$ is defined as the translation between the origins of frames $\mathcal{I}$ and $\mathcal{B}$, and $\mathbf{d}_i^{\mathcal{B}} = [d_{xi}^{\mathcal{B}} \quad d_{yi}^{\mathcal{B}} \quad d_{zi}^{\mathcal{B}}]^T$ is the translation between the origins of frames $\mathcal{B}$ and $\mathcal{C}_i$, for $i = 1, 2, 3, 4$. It should be noted that $\mathbf{d}_1^{\mathcal{B}}$, $\mathbf{d}_2^{\mathcal{B}}$ and $\mathbf{d}_3^{\mathcal{B}}$ are all constants, while $\mathbf{d}_4^{\mathcal{B}}$ varies due to the degrees of freedom of the suspended load.

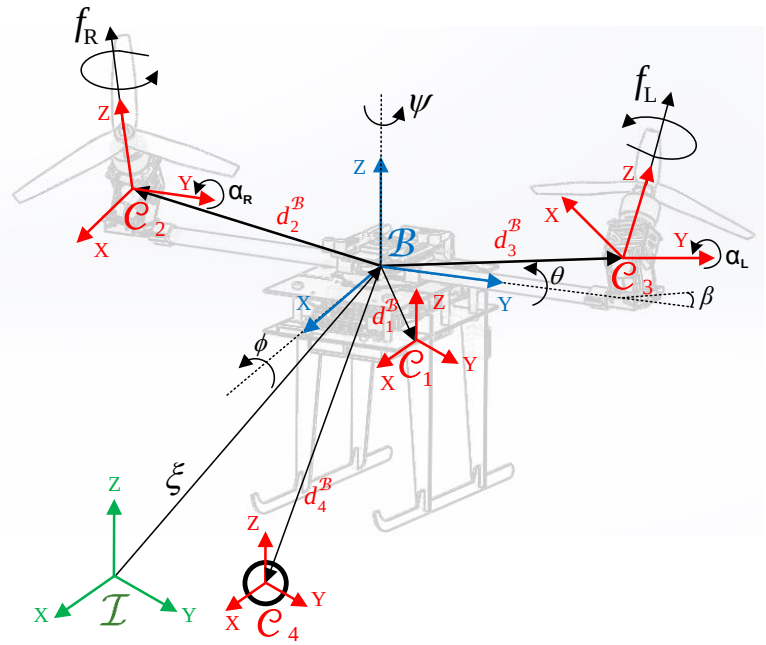

Figure 2.2: Tilt-rotor UAV frames and variables definition.

In order to calculate the vector $\boldsymbol{d}_4^{\mathcal{B}}$, it is used a parametrization that considers the load as a pendulum with a massless rigid rod of length l and two degrees of freedom represented by γ_1 and γ_2 (rotations around $x^{\mathcal{B}}$ and $y^{\mathcal{B}}$, respectively). The Forward Kinematic Model (FKM) of the pendulum subsystem with respect to the aircraft's body is given by:

$$\boldsymbol{d}_4^{\mathcal{B}} = \boldsymbol{R}_{y,\gamma_2}\,\boldsymbol{R}_{x,\gamma_1}\begin{bmatrix}0\\0\\-l\end{bmatrix} = l\begin{bmatrix}-c_{\gamma_1}s_{\gamma_2}\\ s_{\gamma_1}\\ -c_{\gamma_1}c_{\gamma_2}\end{bmatrix}, \tag{2.1}$$

where $c_\theta = cos(\theta)$ and $s_\theta = sin(\theta)$.

The load's attitude is represented with respect to frame $\mathcal{B}$ and its rotation matrix is $R_{\mathcal{C}_4}^{\mathcal{B}} = \boldsymbol{R}_{y,\gamma_2}\,\boldsymbol{R}_{x,\gamma_1}$. The main body attitude in relation with frame $\mathcal{I}$ is described by $\boldsymbol{\eta} = [\phi\ \ \theta\ \ \psi]^T$ (Euler angles with the roll-pitch-yaw convention)[1].

The attitude of the rotors in relation to the main body ($R_{\mathcal{C}_i}^{\mathcal{B}}$, for $i = 2, 3$) is also described using Euler angles. However, it is assumed that there is no rotation around axis $z^{\mathcal{C}_i}$, and the rotation around axis $x^{\mathcal{C}_i}$ is constant and defined by $-\beta$ for $i = 2$ and β for $i = 3$, where β is a small angle. The angle of rotation around axis $y^{\mathcal{C}_i}$, on the other hand, is variable and is denoted by α_R for the frame $\mathcal{C}_2$ and α_L for the frame $\mathcal{C}_3$.

Therefore, the generalized coordinates vector $\boldsymbol{q} \in \Re^{10}$ is defined as follows:

[1]The used convention is shown on Appendix A.1.

$$\boldsymbol{q} = \begin{bmatrix} \boldsymbol{\xi} \\ \boldsymbol{\eta} \\ \boldsymbol{\alpha} \\ \boldsymbol{\gamma} \end{bmatrix}, \tag{2.2}$$

where $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_R & \alpha_L \end{bmatrix}^T$ and $\boldsymbol{\gamma} = \begin{bmatrix} \gamma_1 & \gamma_2 \end{bmatrix}^T$.

2.3 Kinematics

The relation of a point rigidly attached to the body frame $\boldsymbol{p}^{\mathcal{B}}$ with respect to the inertial frame $\mathbf{p}^{\mathcal{I}}$ is given by:

$$\boldsymbol{p}^{\mathcal{I}} = \boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}} \boldsymbol{p}^{\mathcal{B}} + \boldsymbol{\xi}, \tag{2.3}$$

where $\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}$ is the rotation matrix from frame $\mathcal{B}$ to $\mathcal{I}$. This matrix is derived using the roll-pitch-yaw convention and is given by:

$$\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}} = \begin{bmatrix} c_\psi c_\theta & c_\psi c_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \tag{2.4}$$

Moreover, the relation between a point rigidly attached to frame $\mathcal{C}_i$ in relation to the body frame $\mathcal{B}$ is obtained as follows:

$$\boldsymbol{p}^{\mathcal{B}}_i = \boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i} \boldsymbol{p}^{\mathcal{C}_i}_i + \boldsymbol{d}^{\mathcal{B}}_i, \quad i = 1, 2, 3, 4. \tag{2.5}$$

Thus, by replacing equation (2.5) into (2.3) the rigid motion with respect to $\mathcal{I}$ is computed by:

$$\boldsymbol{p}^{\mathcal{I}}_i = \boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}} (\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i} \boldsymbol{p}^{\mathcal{C}_i}_i + \boldsymbol{d}^{\mathcal{B}}_i) + \boldsymbol{\xi}, \quad i = 1, 2, 3, 4. \tag{2.6}$$

2.4 Dynamics using Euler-Lagrange Formulation

This section derives the Euler-Lagrange's equations of motion[2].

[2]Further details on Euler-Lagrange's formulation can be seen in Appendix A.3

A slight variation from the common Euler-Lagrange equations is introduced so as to separate the external forces into known and unknown forces. This model considers that dissipative viscous friction forces are known forces that can be experimentally obtained. The insertion of these friction components are mathematically useful because it provides stabilization of some elements of the system. Thus, Euler-Lagrange equations can be written as:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{F}_{ext} + \boldsymbol{F}_{drag}, \tag{2.7}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{10\times 10}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \Re^{10\times 10}$ is the Coriolis and centrifugal forces matrix, $\boldsymbol{G}(\boldsymbol{q}) \in \Re^{10}$ is the gravitational force vector, $\boldsymbol{F}(\boldsymbol{q}) \in \Re^{10}$ is the independent generalized input force vector, $\boldsymbol{F}_{ext} \in \Re^{10}$ represents external disturbances on the system and the vector $\boldsymbol{F}_{drag} \in \Re^{10}$ is the generalized *drag force* vector. Assuming that drag forces are proportional to the generalized velocity, they are given by:

$$\boldsymbol{F}_{drag} = \boldsymbol{\mu}\dot{\boldsymbol{q}}, \tag{2.8}$$

where $\boldsymbol{\mu} \in \Re^{10\times 10}$ is a constant matrix. Consequently, equation (2.7) can be rewritten as follows:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{F}_{ext}. \tag{2.9}$$

2.4.1 Inertia Matrix

The inertia matrix is obtained by calculating the system's kinetic energy and expressing it in the form $K = \frac{1}{2}\dot{\boldsymbol{q}}^T\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}$. Since the Tilt-Rotor UAV is considered a multibody system, the kinetic energy of the whole system is given by the sum of the individual kinetic energies K_i of each body (Shabana, 2013):

$$K = \sum_{i=1}^{4} K_i, \tag{2.10}$$

where the kinetic energy of the i^{th} body can be obtained from the volume integral:

$$K_i = \frac{1}{2}\int_{V_i} \rho_i(\boldsymbol{v}_i^{\mathcal{I}})^T(\boldsymbol{v}_i^{\mathcal{I}})\mathrm{d}V_i, \tag{2.11}$$

and ρ_i is the mass density at body i. The vector $\boldsymbol{v}_i^{\mathcal{I}}$ is the velocity of a point of body i with respect to frame $\mathcal{I}$ and is given by the time derivative of equation (2.6):

$$\boldsymbol{v}_i^{\mathcal{I}} = \dot{\boldsymbol{p}}_i^{\mathcal{I}} = \dot{\boldsymbol{R}}_{\mathcal{B}}^{\mathcal{I}}(\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{p}_i^{\mathcal{C}_i} + \boldsymbol{d}_i^{\mathcal{B}}) + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}(\dot{\boldsymbol{R}}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{p}_i^{\mathcal{C}_i} + \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\dot{\boldsymbol{p}}_i^{\mathcal{C}_i} + \dot{\boldsymbol{d}}_i^{\mathcal{B}}) + \dot{\boldsymbol{\xi}}. \tag{2.12}$$

As stated before, the points $\boldsymbol{p}_i^{\mathcal{C}_i}$ are rigidly attached to their respective frames, which leads to $\dot{\boldsymbol{p}}_i^{\mathcal{C}_i} = \boldsymbol{0}$ for $i = 1, 2, 3, 4$. Translations $\boldsymbol{d}_i^{\mathcal{B}}$ for $i = 1, 2, 3$ are constant, resulting in $\dot{\boldsymbol{d}}_1^{\mathcal{B}} = \dot{\boldsymbol{d}}_2^{\mathcal{B}} = \dot{\boldsymbol{d}}_3^{\mathcal{B}} = \boldsymbol{0}$. Since $\boldsymbol{R}_{\mathcal{C}_1}^{\mathcal{B}}$ is also constant (the body's frame $\mathcal{B}$ is fixed with respect to the frame of the UAV's center of mass $\mathcal{C}_1$), then $\dot{\boldsymbol{R}}_{\mathcal{C}_1}^{\mathcal{B}} = \boldsymbol{0}_{3\times3}$.

Moreover, it is now possible to use the property of skew symmetric matrices[3] given by $\dot{\boldsymbol{R}}_B^A(t) = \boldsymbol{R}_B^A(t)\,\boldsymbol{S}(\boldsymbol{\omega}_{BA}^B(t))$, being $\boldsymbol{\omega}_{BA}^B(t) \in \Re^3$ the angular velocity of frame B with respect to frame A represented in frame B. Thus, equation (2.12) can be rewritten for each body in the form:

$$\dot{\boldsymbol{p}}_1^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})(\boldsymbol{R}_{\mathcal{C}_1}^{\mathcal{B}}\boldsymbol{p}_1^{\mathcal{C}_1} + \boldsymbol{d}_1^{\mathcal{B}}) + \dot{\boldsymbol{\xi}} \tag{2.13}$$

$$\dot{\boldsymbol{p}}_i^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{p}_i^{\mathcal{C}_i} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})\boldsymbol{d}_i^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i})\boldsymbol{p}_i^{\mathcal{C}_i} + \dot{\boldsymbol{\xi}}, \quad i = 2, 3 \tag{2.14}$$

$$\dot{\boldsymbol{p}}_4^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}}\boldsymbol{p}_4^{\mathcal{C}_4} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})\boldsymbol{d}_4^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{\omega}_{\mathcal{C}_4\mathcal{B}}^{\mathcal{C}_4})\boldsymbol{p}_4^{\mathcal{C}_4} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\dot{\boldsymbol{d}}_4^{\mathcal{B}} + \dot{\boldsymbol{\xi}}. \tag{2.15}$$

With use of the skew symmetric matrices properties $\boldsymbol{S}(\boldsymbol{p})\boldsymbol{q} = \boldsymbol{S}(\boldsymbol{q})^T\boldsymbol{p}$, $\boldsymbol{S}(a\boldsymbol{p} + b\boldsymbol{q}) = a\boldsymbol{S}(\boldsymbol{p}) + b\boldsymbol{S}(\boldsymbol{q})$ and $\boldsymbol{S}(\boldsymbol{R}\boldsymbol{p}) = \boldsymbol{R}\boldsymbol{S}(\boldsymbol{p})\boldsymbol{R}^T$, equations (2.13) – (2.15) can be rewritten as:

$$\dot{\boldsymbol{p}}_1^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_1}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_1^{\mathcal{C}_1})^T(\boldsymbol{R}_{\mathcal{C}_1}^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{d}_1^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \dot{\boldsymbol{\xi}} \tag{2.16}$$

$$\dot{\boldsymbol{p}}_i^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T(\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i} + \dot{\boldsymbol{\xi}}, \quad i = 2, 3 \tag{2.17}$$

$$\dot{\boldsymbol{p}}_4^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_4^{\mathcal{C}_4})^T(\boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{S}(\boldsymbol{d}_4^{\mathcal{B}})^T\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_4^{\mathcal{C}_4})^T\boldsymbol{\omega}_{\mathcal{C}_4\mathcal{B}}^{\mathcal{C}_4} + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\dot{\boldsymbol{d}}_4^{\mathcal{B}} + \dot{\boldsymbol{\xi}}. \tag{2.18}$$

The product $(\boldsymbol{v}_i^{\mathcal{I}})^T(\boldsymbol{v}_i^{\mathcal{I}})$ is then given by:

$$(\boldsymbol{v}_1^{\mathcal{I}})^T(\boldsymbol{v}_1^{\mathcal{I}}) = X_1 \tag{2.19}$$

$$(\boldsymbol{v}_2^{\mathcal{I}})^T(\boldsymbol{v}_2^{\mathcal{I}}) = X_2 + Y_2 \tag{2.20}$$

$$(\boldsymbol{v}_3^{\mathcal{I}})^T(\boldsymbol{v}_3^{\mathcal{I}}) = X_3 + Y_3 \tag{2.21}$$

$$(\boldsymbol{v}_4^{\mathcal{I}})^T(\boldsymbol{v}_4^{\mathcal{I}}) = X_4 + Y_4 + Z_4. \tag{2.22}$$

where X_i, Y_i and Z_i are given by:

[3] Further details on skew symmetric matrices can be seen on Appendix A.2

$$
\begin{aligned}
X_i &= \dot{\boldsymbol{\xi}}^T\dot{\boldsymbol{\xi}} + 2\,\dot{\boldsymbol{\xi}}^T\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T(\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i})^T\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}} + 2\,\dot{\boldsymbol{\xi}}^T\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}} \\
&\quad + (\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T\Big[\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T(\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i})^T + 2\,\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})(\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i})^T\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T \\
&\quad + \boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T\Big]\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}} \qquad (2.23)\\
Y_i &= 2\,\dot{\boldsymbol{\xi}}^T\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}} + (\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}})^T\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}} \\
&\quad + (\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T\Big[2\,\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T + 2\,\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\Big]\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}} \qquad (2.24)\\
Z_i &= 2\,\dot{\boldsymbol{\xi}}^T\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\dot{\boldsymbol{d}}_i^{\mathcal{B}} + (\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T\Big[2\,\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})(\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i})^T + 2\,\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\Big]\dot{\boldsymbol{d}}_i^{\mathcal{B}} \\
&\quad + 2\,\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}}\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\dot{\boldsymbol{d}}_i^{\mathcal{B}} + (\dot{\boldsymbol{d}}_i^{\mathcal{B}})^T\dot{\boldsymbol{d}}_i^{\mathcal{B}}. \qquad (2.25)
\end{aligned}
$$

Assuming that all the system's bodies are symmetric and that each frame $\mathcal{C}_i$ coincides with the center of mass of the i^{th} body, the following holds (Shabana, 2013, p. 147):

$$
\int_{V_i}\rho_i\boldsymbol{p}_i^{\mathcal{C}_i}\mathrm{d}V_i = \boldsymbol{0}_{3\times 1}. \qquad (2.26)
$$

Thus, substituting equations (2.19)-(2.22) into (2.11) and taking into account property (2.26), the kinetic energies of the system's bodies yields to:

$$
\begin{aligned}
K_1 &= X_1' \qquad (2.27)\\
K_2 &= X_2' + Y_2' \qquad (2.28)\\
K_3 &= X_3' + Y_3' \qquad (2.29)\\
K_4 &= X_4' + Y_4' + Z_4'. \qquad (2.30)
\end{aligned}
$$

where X_i', Y_i' and Z_i' are given by:

$$
\begin{aligned}
X_i' &= \frac{1}{2}m_i\dot{\boldsymbol{\xi}}^T\dot{\boldsymbol{\xi}} - m_i\dot{\boldsymbol{\xi}}^T\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}} \\
&\quad + \frac{1}{2}(\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T\left[\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\left[\int\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\mathrm{d}m\right](\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i})^T + m_i\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\right]\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}} \qquad (2.31)\\
Y_i' &= (\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\left[\int\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\mathrm{d}m\right]\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}} + \frac{1}{2}(\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}})^T\left[\int\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T\boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})\mathrm{d}m\right]\boldsymbol{\omega}^{\mathcal{C}_i}_{\mathcal{C}_i\mathcal{B}} \qquad (2.32)\\
Z_i' &= (\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{I}})^T m_i\boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})\dot{\boldsymbol{d}}_i^{\mathcal{B}} + \frac{1}{2}(\dot{\boldsymbol{d}}_i^{\mathcal{B}})^T m_i\dot{\boldsymbol{d}}_i^{\mathcal{B}} + \dot{\boldsymbol{\xi}}^T m_i\boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\dot{\boldsymbol{d}}_i^{\mathcal{B}}, \qquad (2.33)
\end{aligned}
$$

being m_i the mass of body i. Moreover, the inertia tensor of body i with respect to frame $\mathcal{C}_i$ is given by (Shabana, 2013):

$$\boldsymbol{I}_i = \int \boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i})^T \boldsymbol{S}(\boldsymbol{p}_i^{\mathcal{C}_i}) \mathrm{d}m = \begin{bmatrix} I_{xx}^i & I_{xy}^i & I_{xz}^i \\ I_{yx}^i & I_{yy}^i & I_{yz}^i \\ I_{zx}^i & I_{zy}^i & I_{zz}^i \end{bmatrix}. \tag{2.34}$$

In addition, the inertia tensor of body i for a rotation around an axis displaced by a distance $\boldsymbol{d}_i$ (Steiner's theorem for parallel axis) is given by (Shabana, 2013):

$$\boldsymbol{J}_i = \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{I}_i (\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}})^T + m_i \boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}})^T \boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}}). \tag{2.35}$$

Thus, substituting (2.34) and (2.35) into equations (2.31)-(2.33), X_i', Y_i' and Z_i' can be simplified to the form:

$$X_i' = \frac{1}{2} m_i \dot{\boldsymbol{\xi}}^T \dot{\boldsymbol{\xi}} - m_i \dot{\boldsymbol{\xi}}^T \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}}) \boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} + \frac{1}{2} (\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})^T \boldsymbol{J}_i \boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} \tag{2.36}$$

$$Y_i' = (\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})^T \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{I}_i \boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i} + \frac{1}{2} (\boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i})^T \boldsymbol{I}_i \boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i} \tag{2.37}$$

$$Z_i' = (\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}})^T m_i \boldsymbol{S}(\boldsymbol{d}_i^{\mathcal{B}}) \dot{\boldsymbol{d}}_i^{\mathcal{B}} + \frac{1}{2} (\dot{\boldsymbol{d}}_i^{\mathcal{B}})^T m_i \dot{\boldsymbol{d}}_i^{\mathcal{B}} + \dot{\boldsymbol{\xi}}^T m_i \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \dot{\boldsymbol{d}}_i^{\mathcal{B}}. \tag{2.38}$$

In order to write the kinetic energy as a function of the generalized coordinates, the following mappings are applied:

$$\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{W}_\eta \dot{\boldsymbol{\eta}}, \qquad \text{(Raffo, 2011)} \tag{2.39}$$

$$\boldsymbol{\omega}_{\mathcal{C}_2\mathcal{B}}^{\mathcal{C}_2} = \dot{\alpha}_R [0 \;\; 1 \;\; 0]^T = \dot{\alpha}_R \boldsymbol{a}, \tag{2.40}$$

$$\boldsymbol{\omega}_{\mathcal{C}_3\mathcal{B}}^{\mathcal{C}_3} = \dot{\alpha}_L [0 \;\; 1 \;\; 0]^T = \dot{\alpha}_L \boldsymbol{a}, \tag{2.41}$$

$$\boldsymbol{\omega}_{\mathcal{C}_4\mathcal{B}}^{\mathcal{C}_4} = \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} = \boldsymbol{P} \dot{\boldsymbol{\gamma}}, \tag{2.42}$$

$$\dot{\boldsymbol{d}}_4^{\mathcal{B}} = \begin{bmatrix} l s_{\gamma_1} s_{\gamma_2} \dot{\gamma}_1 - l c_{\gamma_1} c_{\gamma_2} \dot{\gamma}_2 \\ l c_{\gamma_1} \dot{\gamma}_1 l s_{\gamma_1} c_{\gamma_2} \dot{\gamma}_1 + l c_{\gamma_1} s_{\gamma_2} \dot{\gamma}_2 \end{bmatrix} = \begin{bmatrix} l s_{\gamma_1} s_{\gamma_2} & -l c_{\gamma_1} c_{\gamma_2} \\ l c_{\gamma_1} & 0 \\ l s_{\gamma_1} c_{\gamma_2} & l c_{\gamma_1} s_{\gamma_2} \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} = \boldsymbol{L} \dot{\boldsymbol{\gamma}}. \tag{2.43}$$

Therefore, summing the kinetic energies through equation (2.10) and representing it in the form $K = \frac{1}{2} \dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q}) \dot{\boldsymbol{q}}$, the inertia matrix is given by:

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{m}_{12} & \boldsymbol{m}_{13} & \boldsymbol{m}_{14} & \boldsymbol{m}_{15} \\ \boldsymbol{m}_{12}^T & \boldsymbol{W}_\eta^T \boldsymbol{J} \boldsymbol{W}_\eta & \boldsymbol{m}_{23} & \boldsymbol{m}_{24} & \boldsymbol{m}_{25} \\ \boldsymbol{m}_{13}^T & \boldsymbol{m}_{23}^T & \boldsymbol{a}^T \boldsymbol{I}_2 \boldsymbol{a} & \boldsymbol{m}_{34} & \boldsymbol{m}_{35} \\ \boldsymbol{m}_{14}^T & \boldsymbol{m}_{24}^T & \boldsymbol{m}_{34}^T & \boldsymbol{a}^T \boldsymbol{I}_3 \boldsymbol{a} & \boldsymbol{m}_{45} \\ \boldsymbol{m}_{15}^T & \boldsymbol{m}_{25}^T & \boldsymbol{m}_{35}^T & \boldsymbol{m}_{45}^T & m_4 \boldsymbol{L}^T \boldsymbol{L} + \mathbf{P}^T \mathbf{I}_4 \mathbf{P} \end{bmatrix}, \tag{2.44}$$

where

$$\begin{aligned} &\boldsymbol{m}_{12} = -\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{H} \boldsymbol{W}_\eta, \quad \boldsymbol{m}_{13} = \mathbf{0}_{3\times1}, \quad \boldsymbol{m}_{14} = \mathbf{0}_{3\times1}, \quad \boldsymbol{m}_{15} = m_4 \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{L}, \\ &\boldsymbol{m}_{23} = \boldsymbol{W}_\eta^T \boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}} \boldsymbol{I}_2 \boldsymbol{a}, \quad \boldsymbol{m}_{24} = \boldsymbol{W}_\eta^T \boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}} \boldsymbol{I}_3 a, \quad \boldsymbol{m}_{25} = \boldsymbol{W}_\eta^T \boldsymbol{R}_{\mathcal{C}_4}^{\mathcal{B}} \boldsymbol{I}_4 \boldsymbol{a} + m_4 \boldsymbol{W}_\eta^T \boldsymbol{S}(\boldsymbol{d}_4^{\mathcal{B}}) \boldsymbol{P}, \\ &\boldsymbol{m}_{34} = 0, \quad \boldsymbol{m}_{35} = \mathbf{0}_{1\times2}, \quad \boldsymbol{m}_{45} = \mathbf{0}_{1\times2}. \end{aligned} \tag{2.45}$$

with $m = \sum m_i$, $\boldsymbol{J} = \sum \boldsymbol{J}_i$ and $\boldsymbol{H} = \boldsymbol{S}\left(\sum m_i \boldsymbol{d}_i^{\mathcal{B}}\right)$.

2.4.2 Coriolis and Centripetal Matrix

The Coriolis and Centripetal matrix is obtained from the Inertia Matrix $\boldsymbol{M}(\boldsymbol{q})$ using Christoffel symbols of the first kind. Thus, the $(k,j)^{th}$ element of the matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is defined as (Spong et al., 2005):

$$c_{kj} = \sum_{i=1}^{10} \frac{1}{2} \left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right] \dot{q}_i, \tag{2.46}$$

where m_{ij} is the $(i,j)^{th}$ element of $\boldsymbol{M}(\boldsymbol{q})$.

2.4.3 Gravity Force Vector

In a multibody system, the potential energy of the whole system is given by the sum of the potential energies of the individual bodies P_i (Shabana, 2013):

$$P = \sum_{i=1}^{4} P_i, \tag{2.47}$$

and

$$P_i = \int_{V_i} \rho_i (\boldsymbol{g}^{\mathcal{I}})^T \boldsymbol{p}_i^{\mathcal{I}} \mathrm{d}V_i, \tag{2.48}$$

is the volume integral on a body i with mass density ρ_i. $\boldsymbol{g}^{\mathcal{I}} = [0\ 0\ -g_z]^T$ is the gravity vector with respect to the inertial frame and V_i is the volume of the body.

By substituting $\boldsymbol{p}_i^{\mathcal{I}}$ (as given in equation (2.6)) into (2.48), one obtains:

$$P_i = (\boldsymbol{g}^{\mathcal{I}})^T \int_{V_i} \rho_i [\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} (\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{p}_i^{\mathcal{C}_i} + \boldsymbol{d}_i^{\mathcal{B}}) + \boldsymbol{\xi}] \mathrm{d}V_i. \tag{2.49}$$

Moreover, taking into account the assumption of equation (2.26), the potential energy of the whole system is given by:

$$P = (\boldsymbol{g}^{\mathcal{I}})^T \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \left(\textstyle\sum_{i=1}^{4} m_i \boldsymbol{d}_i^{\mathcal{B}} \right) + (\boldsymbol{g}^{\mathcal{I}})^T m \boldsymbol{\xi}, \tag{2.50}$$

where m_i is the mass of body i and $m = \sum m_i$. The vector $\boldsymbol{G}(\boldsymbol{q})$ can then be found using:

$$\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial P}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial P}{\partial q_1} \\ \frac{\partial P}{\partial q_2} \\ \vdots \\ \frac{\partial P}{\partial q_{10}} \end{bmatrix}. \tag{2.51}$$

2.4.4 Input Force Vector

The force vector $\boldsymbol{F}(\boldsymbol{q})$ shown in this work was adapted from Gonçalves et al. (2013) and it is given by:

$$\boldsymbol{F}(\boldsymbol{q}) = [T_x \quad T_y \quad T_z \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi \quad \tau_{\alpha_R} \quad \tau_{\alpha_L} \quad \tau_{\gamma_1} \quad \tau_{\gamma_2}]^T, \tag{2.52}$$

where T_i represent translational forces along an axis i and τ_k represent rotational torques actuating around an axis so as to change angle k.

The force provided by each propeller can be decomposed along frame $\mathcal{B}$ such as follows:

$$\boldsymbol{F}_R^{\mathcal{B}} = \begin{bmatrix} f_{Rx}^{\mathcal{B}} \\ f_{Ry}^{\mathcal{B}} \\ f_{Rz}^{\mathcal{B}} \end{bmatrix} = \boldsymbol{R}_{x,-\beta}\, \boldsymbol{R}_{y,\alpha_R} \begin{bmatrix} 0 \\ 0 \\ f_R \end{bmatrix} = \begin{bmatrix} s_{\alpha_R} \\ c_{\alpha_R} s_\beta \\ c_{\alpha_R} c_\beta \end{bmatrix} f_R = \boldsymbol{r}_R f_R \tag{2.53}$$

$$\boldsymbol{F}_L^{\mathcal{B}} = \begin{bmatrix} f_{Lx}^{\mathcal{B}} \\ f_{Ly}^{\mathcal{B}} \\ f_{Lz}^{\mathcal{B}} \end{bmatrix} = \boldsymbol{R}_{x,\beta}\, \boldsymbol{R}_{y,\alpha_L} \begin{bmatrix} 0 \\ 0 \\ f_L \end{bmatrix} = \begin{bmatrix} s_{\alpha_L} \\ -c_{\alpha_L} s_\beta \\ c_{\alpha_L} c_\beta \end{bmatrix} f_L = \boldsymbol{r}_L f_L, \tag{2.54}$$

where f_R and f_L are the right and left propeller thrusts, respectively. The translational forces expressed in the inertial frame are given by:

$$\boldsymbol{T}^{\mathcal{I}} = \begin{bmatrix} T_x^{\mathcal{I}} \\ T_y^{\mathcal{I}} \\ T_z^{\mathcal{I}} \end{bmatrix} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}(\boldsymbol{F}_R^{\mathcal{B}} + \boldsymbol{F}_L^{\mathcal{B}}) = \begin{bmatrix} \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{r}_R & \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{r}_L \end{bmatrix} \begin{bmatrix} f_R \\ f_L \end{bmatrix}. \tag{2.55}$$

The rotational torques are obtained by adding the torque generated by the thrust forces of the propellers to the torque caused by the drag of the propellers. The drag torque generated by each propeller is assumed in steady-state and given by (Castillo et al., 2005b):

$$\tau_{drag} = \frac{k_\tau}{b} f, \tag{2.56}$$

where k_τ and b are aerodynamic constants obtained experimentally and f is the vertical thrust of the given propeller. Thus, the main body's rotational torques are written as follows:

$$\boldsymbol{\tau}^{\mathcal{I}} = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \boldsymbol{W}_\eta^T \begin{bmatrix} (f_{Lz}^{\mathcal{B}} - f_{Rz}^{\mathcal{B}})d_y + \frac{k_\tau}{b}(f_{Lx}^{\mathcal{B}} - f_{Rx}^{\mathcal{B}}) \\ (f_{Rx}^{\mathcal{B}} + f_{Lx}^{\mathcal{B}})d_z + \frac{k_\tau}{b}(f_{Ry}^{\mathcal{B}} + f_{Ly}^{\mathcal{B}}) \\ (f_{Rx}^{\mathcal{B}} - f_{Lx}^{\mathcal{B}})d_y + \frac{k_\tau}{b}(f_{Rz}^{\mathcal{B}} + f_{Lz}^{\mathcal{B}}) \end{bmatrix}, \tag{2.57}$$

where d_y and d_z are given by:

$$d_z = d_{z2}^{\mathcal{B}} = d_{z3}^{\mathcal{B}}, \quad d_y = |d_{y2}^{\mathcal{B}}| = d_{y3}^{\mathcal{B}}. \tag{2.58}$$

It is possible to rewrite (2.57) in the following form:

$$\boldsymbol{\tau}^{\mathcal{I}} = \boldsymbol{W}_\eta^T \begin{bmatrix} -c_{\alpha_R}c_\beta d_y - \frac{k_\tau}{b}s_{\alpha_R} & c_{\alpha_L}c_\beta d_y + \frac{k_\tau}{b}s_{\alpha_L} \\ s_{\alpha_R}d_z + \frac{k_\tau}{b}c_{\alpha_R}s_\beta & s_{\alpha_L}d_z - \frac{k_\tau}{b}c_{\alpha_L}s_\beta \\ s_{\alpha_R}d_y + \frac{k_\tau}{b}c_{\alpha_R}c_\beta & -s_{\alpha_L}d_y - \frac{k_\tau}{b}c_{\alpha_L}c_\beta \end{bmatrix} \begin{bmatrix} f_R \\ f_L \end{bmatrix} = \boldsymbol{W}_\eta^T \begin{bmatrix} \boldsymbol{\tau_R} & \boldsymbol{\tau_L} \end{bmatrix} \begin{bmatrix} f_R \\ f_L \end{bmatrix}. \tag{2.59}$$

The torques τ_{α_R} and τ_{α_L} are direct inputs of the system. The input torques τ_{γ_1} and τ_{γ_2} are always zero, since there is no input that actuates directly over γ_1 and γ_2.

Finally, the input force vector can be expressed in a decoupled form:

$$\boldsymbol{F}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{T}^{\mathcal{I}} \\ \boldsymbol{\tau}^{\mathcal{I}} \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \\ \tau_{\gamma_1} \\ \tau_{\gamma_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{r}_R & \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{r}_L & 0 & 0 \\ \boldsymbol{W}_{\eta}^{T} \boldsymbol{\tau_R} & \boldsymbol{W}_{\eta}^{T} \boldsymbol{\tau_L} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_R \\ f_L \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \end{bmatrix} = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\Gamma}, \tag{2.60}$$

where $\boldsymbol{\Gamma} = \begin{bmatrix} f_R & f_L & \tau_{\alpha_R} & \tau_{\alpha_L} \end{bmatrix}^T$ is the system's input vector. Given that the force input vector can be decoupled in the form $\boldsymbol{F}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{\Gamma}$, this system's state space representation is classified as *affine in the inputs*, as will be shown on next subsection. Equation 2.60 also evidences the fact that this system is underactuated, i.e., it has less inputs than degrees of freedom.

2.4.5 Drag Force Vector

The drag force is assumed proportional to the generalized velocity and is given by:

$$\boldsymbol{F}_{drag} = \boldsymbol{\mu}\dot{\boldsymbol{q}}. \tag{2.61}$$

The matrix $\boldsymbol{\mu}$ is considered to be a diagonal matrix, where its $(i,i)^{th}$ element represents the drag coefficient associated with the i^{th} generalized coordinate:

$$\boldsymbol{\mu} = diag(\mu_x,\, \mu_y,\, \mu_z,\, \mu_\phi,\, \mu_\theta,\, \mu_\psi,\, \mu_{\alpha_R},\, \mu_{\alpha_L},\, \mu_{\gamma_1},\, \mu_{\gamma_2}). \tag{2.62}$$

This work assumes that the coefficients of matrix $\boldsymbol{\mu}$ can be obtained experimentally.

2.5 State-Space Representation of the System

A total of twenty state-space variables are needed to represent the system's dynamics:

$$\boldsymbol{x}(t) = \begin{bmatrix} x_1 \\ \vdots \\ x_{20} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}. \tag{2.63}$$

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_u(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{g}_d(\boldsymbol{x})\boldsymbol{d}, \tag{2.64}$$

where $\boldsymbol{u} = \boldsymbol{\Gamma}$ and $\boldsymbol{d} = \boldsymbol{F}_{ext}$. The vector $\boldsymbol{f}(\boldsymbol{x})$ is called *drift vector field*, $\boldsymbol{g}_u(\boldsymbol{x})$ is the *steering vector field* and $\boldsymbol{g}_d(\boldsymbol{x})$ is the *external steering vector field*.

Euler-Lagrange's dynamic equations (2.9) can be rewritten in the following form:

$$\begin{aligned}\ddot{\boldsymbol{q}} &= \boldsymbol{M}^{-1}\{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{\Gamma} + \boldsymbol{F}_{ext} - [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})\} \\ &= \boldsymbol{M}^{-1}\{\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{d} - [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})\}. \end{aligned} \tag{2.65}$$

Therefore, the nonlinear state-space representation is found to be:

$$\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{d} - [\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})]\end{bmatrix}, \tag{2.66}$$

from where it is possible to extract the nonlinear vector fields:

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix}\dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}[-[\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})]\end{bmatrix}, \quad \boldsymbol{g}_u(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{M}^{-1}\boldsymbol{B}(\boldsymbol{q})\end{bmatrix}, \quad \boldsymbol{g}_d(\boldsymbol{x}) = \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{M}^{-1}\end{bmatrix}. \tag{2.67}$$

2.6 System Design Parameters

Table 2.1 show the nominal parameters of the Tilt-rotor under design. The mass and inertia moments parameters were estimated using Computer Aided Design (CAD) tools. The parameter g_z is a known natural constant, while k_τ and b can be obtained experimentally (Castillo et al., 2005a). The angle β and the string length l can be changed, but their nominal values are the ones shown on the table.

As for the drag force matrix $\boldsymbol{\mu}$, nonzero friction are considered only for μ_{γ_1} and μ_{γ_2}. The introduction of friction for these variables is useful because it makes the variables γ_1 and γ_2 to be stable, as will be discussed on Chapter 4. These values were chosen by trial and error through observations from simulations and are given by $\mu_{\gamma_1} = \mu_{\gamma_2} = 0.005\frac{N.m}{rad/s}$. Therefore, this matrix is modeled as:

$$\boldsymbol{\mu} = diag(0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0.005,\ 0.005). \tag{2.68}$$

The actuators are considered to respond instantaneously, e.g., if a given thrust is required on the propeller, it readily delivers this thrust. However, the actuators are all modelled with saturations: the propellers can deliver thrusts in the region $0 < \{f_R, f_L\} < 17$ N and the servomotors are able to apply torques $-1 < \{\tau_{\alpha_R}, \tau_{\alpha_L}\} < 1$ N·m. These are the values of the instruments that will be used on the ProVant project.

Table 2.1: System Parameters.

Parameter	Value
m_1	1.243 Kg
m_2, m_3	0.150 Kg
m_4	0.050 Kg
$\boldsymbol{d}_1^{\mathcal{B}}$	$[-6.5 \cdot 10^{-4} \quad -7.2 \cdot 10^{-3} \quad -4.6 \cdot 10^{-2}]^T$ m
$\boldsymbol{d}_2^{\mathcal{B}}$	$[1.7 \cdot 10^{-2} \quad -0.27 \quad 5.1 \cdot 10^{-2}]^T$ m
$\boldsymbol{d}_3^{\mathcal{B}}$	$[1.7 \cdot 10^{-2} \quad 0.27 \quad 5.1 \cdot 10^{-2}]^T$ m
I_{xx}^1	0.018891956 Kg·m^2
I_{yy}^1	0.005237518 Kg·m^2
I_{zz}^1	0.018027985 Kg·m^2
I_{xx}^2, I_{xx}^3	0.000077509 Kg·m^2
I_{yy}^2, I_{yy}^3	0.000069700 Kg·m^2
I_{zz}^2, I_{zz}^3	0.000076109 Kg·m^2
$I_{xx}^4, I_{yy}^4, I_{zz}^4$	0.000002645 Kg·m^2
g_z	9.81 m/s^2
k_τ	1.7×10^{-7} N·m·s^2
b	9.5×10^{-6} N·s^2
β	5°
l	0.5 m

2.7 Conclusions

This chapter presented the dynamic equations of motion of the Tilt-rotor UAV with suspended load using Euler-Lagrange formulation. The developed model considered a system with ten degrees of freedom:

- Three translational coordinates (x, y and z);
- Three attitude coordinates (ϕ, θ and ψ) described by the Roll-Pitch-Yaw convention;
- Two coordinates describing the orientation of both tilting rotors (α_R and α_L);
- Two coordinates describing the attitude of the load with respect to the aircraft (γ_1 and γ_2).

Then, a state-space model affine in the inputs was derived for the system. This representation is used in the next chapters so as to design linear and nonlinear control laws for the Tilt-rotor UAV with suspended load.

Chapter 3

Linear Control Strategies

Sumário

3.1 Introduction 29
3.2 Linear Control Systems Theory 30
3.2.1 Controllability of Linear Systems 30
3.2.2 Stability of Linear Systems 31
3.2.3 Linear Matrix Inequalities 32
3.2.4 D-stability 32
3.2.5 Linear $\mathcal{H}_\infty$ Controllers 35
3.3 Tilt-Rotor Linear Control 37
3.3.1 Equilibrium Point and Linear Model 38
3.3.2 Controllability of the Tilt-rotor UAV 39
3.3.3 D-stable Controller Design 39
3.3.4 D-Stable $\mathcal{H}_\infty$ Controller Design 40
3.4 Simulation and Results 41
3.5 Conclusions 43

3.1 Introduction

The aim of this chapter is to provide some solutions for the problem of linear closed-loop control of the Tilt-rotor UAV with suspended load. The objective of the proposed control strategies is to transport a payload using a Tilt-rotor UAV along a predefined trajectory using a unique linear control law, which should be able to stabilize the whole system, even when exogenous disturbances affect the system, and in presence of unmodelled dynamics and parametric uncertainties. The controller must provide path tracking for x, y, z (position coordinates) and ψ (yaw angle) whilst stabilizing all other generalized coordinates. In this chapter it is assumed that all twenty states of the UAV are precisely known so that state-feedback control strategies are designed.

The linear controller is designed in the vicinity of the aircraft's equilibrium point using Linear Matrix Inequalities (LMI) and two linear control laws are developed. The first controller is based on regional pole allocation (D-Stability): a region on the complex plane is defined and the solver finds a state-feedback control law $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$ such that the poles in closed-loop are all inside this region. This first solution can be satisfactory if the

requirement is to stabilize the rotorcraft while following some trajectories. However, it is still possible to optimize the D-Stable solution by finding some pole allocation inside the chosen region that also minimizes some given cost function. Therefore, a second linear controller is designed such that it keeps the pole allocation within the region of the first controller but also maximizing disturbance rejection. The group of controllers that satisfy the constraint of maximal disturbance rejection for the worst disturbance scenario is known as $\mathcal{H}_\infty$ controllers.

Section 3.2 introduces theory on closed loop control using linear state-space representations. Besides, D-stability and $\mathcal{H}_\infty$ strategies using LMI design are also introduced in this section. Section 3.3 presents the linearization of the Tilt-rotor around an equilibrium point so that the linear methods can be applied. Section 3.4 shows simulation results of both methods and compare them.

3.2 Linear Control Systems Theory

The dynamics of linear continuous multivariable systems can be represented according to the following equation (Ogata, 2001):

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \tag{3.1}$$

where $\boldsymbol{x}(t) \in \Re^n$ is the *state vector* of the system, $\boldsymbol{u}(t) \in \Re^m$ is the *input vector*, $\boldsymbol{A} \in \Re^{n\times n}$ is the *state matrix* and $\boldsymbol{B} \in \Re^{n\times m}$ is the *input matrix*.

3.2.1 Controllability of Linear Systems

An important definition on linear systems is the notion of controllability: when a system is controllable, then it is possible to steer it—by admissible inputs—from any initial value to any final value within some finite time window. A system represented in the form of equation (3.1) is said to be controllable if and only if (Ogata, 2001):

$$rank\left(\begin{bmatrix}\boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2\boldsymbol{B} & \dots & \boldsymbol{A}^{n-1}\boldsymbol{B}\end{bmatrix}\right) = n. \tag{3.2}$$

Alternatively, controllability can also be defined by using the Popov-Belevitch-Hautus test (Hautus, 1970), which says that a system is controllable if and only if:

$$rank\left(\begin{bmatrix}\lambda\boldsymbol{I}_n - \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\right) = n, \quad \forall\, \lambda \in \boldsymbol{\Lambda} = eig(\boldsymbol{A}), \tag{3.3}$$

where $eig(\cdot)$ is the operation that returns all eigenvalues of matrix $\cdot$ and $\boldsymbol{I}_n$ is the $n\times n$ identity matrix. It is preferable to use (3.3) rather than (3.2) when the controllability matrix $\begin{bmatrix}\boldsymbol{B} & \boldsymbol{A}\boldsymbol{B} & \boldsymbol{A}^2\boldsymbol{B} & \dots & \boldsymbol{A}^{n-1}\boldsymbol{B}\end{bmatrix}$ is ill-conditioned or sensitive to round-off errors.

3.2.2 Stability of Linear Systems

The stability and natural response characteristics of a linear system can be studied by looking at its poles. In state-space, the poles of the system are equal to the eigenvalues of matrix $\boldsymbol{A}$. Therefore, one can affirm that a system is stable if and only if (Ogata, 2001):

$$Re(\lambda) < 0, \quad \forall\, \lambda \in \boldsymbol{\Lambda} = eig(\boldsymbol{A}). \tag{3.4}$$

Choosing a control law given by $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$, where $\boldsymbol{K} \in \Re^{m \times n}$ is defined as the *control matrix*, then the closed-loop state-space representation becomes:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{K}\boldsymbol{x} \\ &= (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\boldsymbol{x} \\ &= \boldsymbol{A}_f\boldsymbol{x}, \end{aligned} \tag{3.5}$$

where $\boldsymbol{A}_f$ is then defined as the *closed loop state matrix* of the system. Thus, in order to evaluate the stability of the closed loop system, one needs to verify if:

$$Re(\lambda_f) < 0, \quad \forall\, \lambda_f \in \boldsymbol{\Lambda_f} = eig(\boldsymbol{A_f}). \tag{3.6}$$

Therefore, if the system is controllable, then it is possible to find a matrix $\boldsymbol{K}$ such that all poles of $\boldsymbol{A}_f$ are stable.

Alternatively, another way of analysing the stability of a system is using Lyapunov's stability theory (Lyapunov, 1892). According to Lyapunov's method, it is possible to say that a system is asymptotically stable in the sense of Lyapunov if it possible to find a function $V(\boldsymbol{x}) : \Re^n \rightarrow \Re$ such that the following constraints hold:

LYAPUNOV-1: $V(\boldsymbol{x} = \boldsymbol{0}) = 0$;

LYAPUNOV-2: $V(\boldsymbol{x}) > 0, \quad \forall\, \boldsymbol{x} \neq \boldsymbol{0}$;

LYAPUNOV-3: $\dot{V}(\boldsymbol{x}) < 0, \quad \forall\, \boldsymbol{x} \neq \boldsymbol{0}$.

In linear systems, it is usual to use the candidate function $V(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x}$, where $\boldsymbol{P} \in \Re^{n \times n}$ is a symmetric matrix. The conditions for satisfying the above constraints are:

- The constraint LYAPUNOV-1 is always satisfied for this chosen $V(\boldsymbol{x})$, given that $\boldsymbol{0}^T\boldsymbol{P}\boldsymbol{0} = 0, \forall\, \boldsymbol{P}$.
- The constraint LYAPUNOV-2 is only satisfied if $\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} > 0, \forall\, \boldsymbol{x} \neq \boldsymbol{0}$. This is true if and only if $\boldsymbol{P}$ is positive definite, or likewise:

$$\boldsymbol{P} > 0. \tag{3.7}$$

- As for constraint LYAPUNOV-3, one needs to analyse $\dot{V}(\boldsymbol{x})$:

$$\dot{V}(\boldsymbol{x}) = \dot{\boldsymbol{x}}^T \boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{P}\dot{\boldsymbol{x}} = (\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u})^T \boldsymbol{P}\boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{P}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}). \tag{3.8}$$

  By choosing $\boldsymbol{u} = \boldsymbol{0}$, then it is possible to analyse the stability of the open-loop system:

$$\dot{V}(\boldsymbol{x}) = \boldsymbol{x}^T [\boldsymbol{A}^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}]\boldsymbol{x} < 0. \tag{3.9}$$

  Thus, LYAPUNOV-3 is satisfied if and only if the following inequality holds:

$$\boldsymbol{A}^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} < 0. \tag{3.10}$$

  If, in turn, the input $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$ is chosen, then LYAPUNOV-3 is satisfied if and only if the following inequality holds:

$$\boldsymbol{A}_f^T \boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}_f < 0. \tag{3.11}$$

Therefore, in short, it is possible to state that an open loop system is stable if it is possible to find a symmetric definite positive matrix $\boldsymbol{P}$ such that the inequality (3.10) holds. Equally, a closed-loop system is stable if it is possible find a symmetric definite positive matrix $\boldsymbol{P}$ such that the inequality given by (3.11) holds. However, if one cannot find $\boldsymbol{P}$ that satisfies these constraints, nothing can be said about the stability of the system.

3.2.3 Linear Matrix Inequalities

The inequalities shown at (3.7), (3.10) and (3.11) are LMIs, i.e. Linear Matrix Inequalities (Boyd et al., 1994). In general, LMIs are hard to solve. Fortunately, there are some techniques to solve these problems using optimization algorithms. Thus, there are many toolboxes that provide solutions to this kind of problems, such as Yalmip (Löfberg, 2004), CXV (Grant and Boyd, 2014) or LMILab (Gahinet and Nemirovskii, 1993), just to name a few. In this work, it was adopted solutions using Yalmip with Sedumi solver (Labit and Peaucelle, 2002).

3.2.4 D-stability

A system is defined as D-stable if all of its poles are within a defined convex region in the complex plane (Trofino et al., 2003). A LMI region is defined as a subset $\mathbb{D}$ of the complex plane $\mathbb{C}$, where $\mathbb{D}$ is given by (Chilali and Gahinet, 1996):

$$\mathbb{D} = \left\{ s \in \mathbb{C} : \boldsymbol{L} + s\boldsymbol{M} + s^* \boldsymbol{M}^T < 0 \right\}, \tag{3.12}$$

where $\boldsymbol{L} = \boldsymbol{L}^T$ and $\boldsymbol{M}$ are matrices that define region $\mathbb{D}$, $s = \sigma + j\omega$ and s^* is the complex conjugate of s. The most commonly used regions are:

Region 1 $\mathrm{Re}(s) < -\alpha$:

This region can be easily defined by the expression $s+s^* < -2\alpha$, given that $s+s^* = \sigma+j\omega+\sigma-j\omega = 2\sigma$. Therefore, it is possible to rewrite this expression in the form $2\alpha + s + s^T < 0$ and then $\boldsymbol{L} = 2\alpha$ and $\boldsymbol{M} = 1$.

Region 2 Disk centred at $(c, 0)$ with radius r:

This region can be defined by the expression $|s-c| < r$. It can be developed as:

$$\begin{aligned} |\sigma + j\omega - c| &< r \\ \sqrt{(\sigma-c)^2+\omega^2} &< r \\ r^2 - [(\sigma-c)+j\omega][(\sigma-c)-j\omega] &> 0 \\ r^2 - (s-c)(s^*-c) &> 0 \\ det\left(\begin{bmatrix} r & s-c \\ s^*-c & r \end{bmatrix}\right) &> 0. \end{aligned}$$

Given that the determinant of a definite positive matrix is always positive, then the following holds:

$$\begin{bmatrix} r & s-c \\ s^*-c & r \end{bmatrix} > 0.$$

Changing the sign of the inequality and then applying a similarity transformation:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -r & -(s-c) \\ -(s^*-c) & -r \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} < 0$$

$$\begin{bmatrix} -r & s-c \\ s^*-c & -r \end{bmatrix} < 0$$

$$\begin{bmatrix} -r & -c \\ -c & -r \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} s + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} s^* < 0.$$

Therefore:

$$\boldsymbol{L} = \begin{bmatrix} -r & -c \\ -c & -r \end{bmatrix} \quad \boldsymbol{M} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Region 3 Cone defined by $|\mathrm{Im}(s)| < \tan(\varphi)|\mathrm{Re}(s)|$

This region can be defined by the expression $\omega\cos(\varphi) < -\sigma\sin(\varphi)$. It is possible to

obtain the matrices $\boldsymbol{L}$ and $\boldsymbol{M}$ by the following procedure:

$$\begin{gathered}
\sigma^2 \sin^2(\varphi) < \omega^2 \cos^2(\varphi) \\
\sigma^2 \sin^2(\varphi) < (j\omega)(-j\omega) \cos^2(\varphi) \\
\frac{1}{4}(s+s^*)^2 \sin^2(\varphi) < \frac{1}{2}(s-s^*)\frac{1}{2}(s^*-s) \cos^2(\varphi) \\
(s+s^*)^2 \sin^2(\varphi) - (s-s^*)(s^*-s) \cos^2(\varphi) < 0 \\
\begin{bmatrix} (s+s^*)\sin(\varphi) & (s-s^*)\cos(\varphi) \\ (s^*-s)\cos(\varphi) & (s+s^*)\sin(\varphi) \end{bmatrix} < 0 \\
\begin{bmatrix} \sin(\varphi) & \cos(\varphi) \\ -\cos(\varphi) & \sin(\varphi) \end{bmatrix} s + \begin{bmatrix} \sin(\varphi) & -\cos(\varphi) \\ \cos(\varphi) & \sin(\varphi) \end{bmatrix} s^* < 0.
\end{gathered} \tag{3.13}$$

Therefore, the following holds:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \boldsymbol{M} = \begin{bmatrix} \sin(\varphi) & \cos(\varphi) \\ -\cos(\varphi) & \sin(\varphi) \end{bmatrix}.$$

Region 1 is useful so that it guarantees that the system is faster than a minimum requirement. Region 3 is important because it limits the system's maximum percentage overshoot. Region 2, in turn, is usually used with $c = 0$ and is relevant because it avoids the poles from being allocated too far away from the origin of the plane, avoiding too large gains in the controller.

According to Chilali and Gahinet (1996), a state matrix $\boldsymbol{A}$ is D-stable if and only if there is a real definite positive symmetric matrix $\boldsymbol{Q} \in \Re^{n \times n}$ such that:

$$\boldsymbol{L} \otimes \boldsymbol{Q} + \boldsymbol{M} \otimes \boldsymbol{AQ} + \boldsymbol{M}^T \otimes (\boldsymbol{AQ})^T < 0, \tag{3.14}$$

where $\otimes$ is the Kronecker product operator. This product operation is defined by:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{l1} & \dots & a_{lk} \end{bmatrix} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \dots & a_{1k}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{l1}\boldsymbol{B} & \dots & a_{lk}\boldsymbol{B} \end{bmatrix}. \tag{3.15}$$

Thus, the LMIs that verify whether the eigenvalues of $\boldsymbol{A}$ are D-stable are given by (Trofino et al., 2003):

Region 1 For $Re(s) < -\alpha$, then:

$$2\alpha\boldsymbol{Q} + \boldsymbol{AQ} + \boldsymbol{QA}^T < 0, \quad \boldsymbol{Q} > 0. \tag{3.16}$$

Region 2 For the disk centered at $(c, 0)$ with radius r, then:

$$\begin{bmatrix} -r & -c \\ -c & -r \end{bmatrix} \otimes \boldsymbol{Q} + \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \otimes \boldsymbol{AQ} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \otimes (\boldsymbol{AQ})^T < 0.$$

Therefore, the LMIs are:

$$\begin{bmatrix} -r\boldsymbol{Q} & -c\boldsymbol{Q}+\boldsymbol{A}\boldsymbol{Q} \\ -c\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T & -r\boldsymbol{Q} \end{bmatrix} < 0, \quad \boldsymbol{Q} > 0. \tag{3.17}$$

Region 3 For the cone defined by $|Im(s)| < \tan(\varphi)|Re(s)|$, the LMIs are:

$$\begin{bmatrix} \sin(\varphi)(\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T) & \cos(\varphi)(\boldsymbol{A}\boldsymbol{Q}-\boldsymbol{Q}\boldsymbol{A}^T) \\ \cos(\varphi)(-\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T) & \sin(\varphi)(\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T) \end{bmatrix} < 0, \quad \boldsymbol{Q} > 0. \tag{3.18}$$

In order to analyse the system in closed loop, it is possible to substitute the matrix $\boldsymbol{A}$ by $\boldsymbol{A}+\boldsymbol{B}\boldsymbol{K}$ on equations (3.16)-(3.18), obtaining the following LMIs for $\boldsymbol{Q} > 0$:

$$2\alpha\boldsymbol{Q}+\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T+\boldsymbol{B}\boldsymbol{Y}+\boldsymbol{Y}^T\boldsymbol{B}^T < 0, \tag{3.19}$$

$$\begin{bmatrix} -r\boldsymbol{Q} & -c\boldsymbol{Q}+\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{B}\boldsymbol{Y} \\ -c\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T+\boldsymbol{Y}^T\boldsymbol{B}^T & -r\boldsymbol{Q} \end{bmatrix} < 0, \tag{3.20}$$

$$\begin{bmatrix} s_\varphi(\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T+\boldsymbol{B}\boldsymbol{Y}+\boldsymbol{Y}^T\boldsymbol{B}^T) & c_\varphi(\boldsymbol{A}\boldsymbol{Q}-\boldsymbol{Q}\boldsymbol{A}^T+\boldsymbol{B}\boldsymbol{Y}-\boldsymbol{Y}^T\boldsymbol{B}^T) \\ c_\varphi(-\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T-\boldsymbol{B}\boldsymbol{Y}+\boldsymbol{Y}^T\boldsymbol{B}^T) & s_\varphi(\boldsymbol{A}\boldsymbol{Q}+\boldsymbol{Q}\boldsymbol{A}^T+\boldsymbol{B}\boldsymbol{Y}+\boldsymbol{Y}^T\boldsymbol{B}^T) \end{bmatrix} < 0, \tag{3.21}$$

where $\boldsymbol{Y} = \boldsymbol{K}\boldsymbol{Q}$. This solution is very interesting because it is able to provide two different functionalities. First, it can verify if the system is D-stable for a given designed controller $\boldsymbol{K}$. On the other hand, this solution can also be inserted in the algorithm to find both matrices $\boldsymbol{Q}$ and $\boldsymbol{Y}$ that satisfy the given constraints. As a result, one can obtain a controller that D-stabilizes the system by using the relation $\boldsymbol{K} = \boldsymbol{Y}\boldsymbol{Q}^{-1}$.

3.2.5 Linear $\mathcal{H}_\infty$ Controllers

When dealing with linear $\mathcal{H}_\infty$ controllers with state feedback, the following extended linear system is considered:

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}_u\boldsymbol{u}+\boldsymbol{B}_w\boldsymbol{d} \\ \boldsymbol{z} &= \boldsymbol{C}_z\boldsymbol{x}+\boldsymbol{D}_{uz}\boldsymbol{u}+\boldsymbol{D}_{wz}\boldsymbol{d}, \end{aligned} \tag{3.22}$$

where $\boldsymbol{z}$ is the error signal to be minimized; $\boldsymbol{C}_z$, $\boldsymbol{D}_{wz}$ and $\boldsymbol{D}_{uz}$ are constant weighting matrices that are determined on the design of the controller. This extended system can be illustrated by the block diagram presented in Figure 3.1, where $\boldsymbol{P}(s)$ is the generalized process and $\boldsymbol{K}(s)$ is the controller.

The linear $\mathcal{H}_\infty$ controller belongs to the class of optimal controllers, which exponentially stabilizes the system while also minimizing the $\mathcal{H}_\infty$ norm. In SISO systems, the $\mathcal{H}_\infty$ norm is defined as the maximum magnitude of its frequency response to disturbance inputs (equation (3.23)). On the other hand, MIMO systems are described by multiple frequency response mappings and one should use the frequency response obtained from

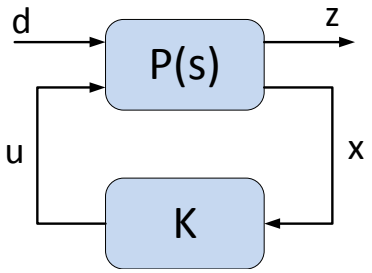

Figure 3.1: $\mathcal{H}_\infty$ diagram block.

the singular value decomposition of the system's transfer function matrix. Then, the $\mathcal{H}_\infty$ norm is defined as the maximum of the highest singular value for given input frequencies, as shown in equation (3.24).

$$||\boldsymbol{H}(s)||_\infty = \max_\omega |\boldsymbol{H}(j\omega)| \tag{3.23}$$

$$||\boldsymbol{H}(s)||_\infty = \sup_\omega(\max_{\bar{\sigma}}\{\boldsymbol{H}(j\omega)\}). \tag{3.24}$$

The $\mathcal{H}_\infty$ norm can also be interpreted as the highest gain in terms of energy of the response signal for a given input and an alternative definition for the $\mathcal{H}_\infty$ norm is obtained from Parseval's Theorem, which states that the energy E_z of the signal $z(t)$ is given by (Lathi, 2009):

$$E_z = ||z(t)||_2^2 = \int_0^\infty |z(t)|^2 \,\mathrm{d}t = \frac{1}{2\pi}\int_0^\infty \boldsymbol{Z}(j\omega)^* \cdot \boldsymbol{Z}(j\omega)\,\mathrm{d}\omega, \tag{3.25}$$

where $\boldsymbol{Z}(j\omega)$ is the Fourier transform of $z(t)$. Similarly, it is also possible to calculate the energy of the input disturbance signal:

$$E_d = ||d(t)||_2^2 = \frac{1}{2\pi}\int_0^\infty \boldsymbol{D}(j\omega)^* \cdot \boldsymbol{D}(j\omega)\,\mathrm{d}\omega. \tag{3.26}$$

Since $\boldsymbol{Z}(j\omega) = \boldsymbol{H}(j\omega)\boldsymbol{D}(j\omega)$, then it is possible to obtain the following:

$$\begin{aligned} ||z(t)||_2^2 &= \frac{1}{2\pi}\int_0^\infty \boldsymbol{D}(j\omega)^* \cdot \boldsymbol{H}(j\omega)^* \cdot \boldsymbol{H}(j\omega) \cdot \boldsymbol{D}(j\omega)\,\mathrm{d}\omega \\ &\leq \frac{1}{2\pi}\int_0^\infty (\bar{\sigma}\{\boldsymbol{H}(j\omega)\})^2\, \boldsymbol{D}(j\omega)^* \cdot \boldsymbol{D}(j\omega)\,\mathrm{d}\omega \\ &\leq (\sup_\omega \bar{\sigma}\{\boldsymbol{H}(j\omega)\})^2 \frac{1}{2\pi}\int_0^\infty \boldsymbol{D}(j\omega)^* \cdot \boldsymbol{D}(j\omega)\,\mathrm{d}\omega. \end{aligned} \tag{3.27}$$

which yields:

$$||z(t)||_2 \leq ||\boldsymbol{H}(s)||_\infty \, ||d(t)||_2. \tag{3.28}$$

This way, if the energy of the disturbance input is nonzero and bounded, then the $\mathcal{H}_\infty$ norm is defined in the time domain as (Skogestad and Postlethwaite, 2005):

$$||\boldsymbol{H}(s)||_\infty = \sup_{E_d \neq 0} \frac{||z(t)||_2}{||d(t)||_2}. \tag{3.29}$$

In other words, the $\mathcal{H}_\infty$ norm can be described as the maximum energy amplification for E_z due to an input with energy E_d.

The linear optimal $\mathcal{H}_\infty$ control problem consists in computing the controller that minimizes
$||\boldsymbol{H}(s)||_\infty$. This optimal problem hard to solve, but it is possible to find a suboptimal solution (Raffo, 2011), where a numerical upper bound γ is found such that:

$$||\boldsymbol{H}(s)||_\infty < \gamma \tag{3.30}$$

and it is common to search for the minimum upper bound γ through an iterative process. This can be solved by using the LMI in (3.31) (Dullerud and Paganini, 2000). Following this approach one can find a level attenuation $\tilde{\gamma}$ which guarantees that $||\boldsymbol{H}(s)||_\infty < \sqrt{\tilde{\gamma}}$. In order to combine this solution with the D-stable controllers, inequality (3.31) is considered altogether with inequalities (3.19) to (3.21).

$$\min_{\boldsymbol{Q},\boldsymbol{Y}} \tilde{\gamma} \left\{ \begin{bmatrix} \boldsymbol{AQ} + \boldsymbol{B}_u\boldsymbol{Y} + \boldsymbol{Q}\boldsymbol{A}^T + \boldsymbol{Y}^T\boldsymbol{B}_u^T & \boldsymbol{B}_w & \boldsymbol{Q}\boldsymbol{C}_z^T + \boldsymbol{Y}^T\boldsymbol{D}_{uz}^T \\ \boldsymbol{B}_w^T & -\tilde{\gamma}\mathbb{I}_{nw} & \boldsymbol{D}_{wz}^T \\ \boldsymbol{C}_z\boldsymbol{Q} + \boldsymbol{D}_{uz}\boldsymbol{Y} & \boldsymbol{D}_{wz} & -\tilde{\gamma}\mathbb{I}_{nz} \end{bmatrix} < 0. \right. \tag{3.31}$$

3.3 Tilt-Rotor Linear Control

First, a linear state space model of the system is obtained in which the operation point is chosen to be an equilibrium point of the UAV. Then, a pure D-stable controller is designed by finding the matrices $\boldsymbol{Q} > 0$ and $\boldsymbol{Y}$ that satisfy the inequalities (3.19)-(3.21). After this, a second linear controller is designed using both D-stable and $\mathcal{H}_\infty$ requisites, being necessary to find $\boldsymbol{Q} > 0$ and $\boldsymbol{Y}$ that satisfy the inequalities (3.19)-(3.21) and (3.31) simultaneously[1].

When designing the controllers, the linear state space model will be extended so as to avoid steady state errors in closed loop in the presence of constant disturbances. An

[1]The approach presented in this section was submitted and accepted to be published on CBA - 2014 (Almeida et al., 2014a).

integral action is added to the trajectory errors in x, y, z and ψ and the new state vector $\boldsymbol{x} \in \Re^{24}$ is defined as follows:

$$\boldsymbol{x} = [x_1 \quad \dots \quad x_{24}]^T = [\boldsymbol{q}^T \quad \dot{\boldsymbol{q}}^T \quad \int \Delta x \quad \int \Delta y \quad \int \Delta z \quad \int \Delta \psi]^T. \tag{3.32}$$

3.3.1 Equilibrium Point and Linear Model

In order to linearize the system, first the equilibrium point is computed as the state space point where all generalized coordinates' velocities and accelerations are zero:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} \dot{\boldsymbol{q}}(t) \\ \ddot{\boldsymbol{q}}(t) \end{bmatrix} = 0. \tag{3.33}$$

Thus, the equilibrium values of the inputs $\boldsymbol{u}$ and generalized coordinates $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ which ensure hovering of the Tilt-rotor UAV are obtained by setting:

$$\dot{\boldsymbol{q}} = 0 \tag{3.34}$$

$$\boldsymbol{M}^{-1}[\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{d} - [\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}(\boldsymbol{q})] = 0. \tag{3.35}$$

Assuming a scenario with no external disturbances ($\boldsymbol{d} = \boldsymbol{0}$), the following relation is obtained:

$$\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} - \boldsymbol{G}(\boldsymbol{q}) = 0. \tag{3.36}$$

Since this system has ten equations and fourteen variables, in order to avoid an infinite number of solutions, four variables are freely chosen and the other ten are computed from (3.36). The state variables x, y and z do not appear on the above equations, what would lead to an infinite number of equilibrium points. The fourth chosen variable is ψ, given that the UAV can be found in equilibrium independently of its yaw angle. Therefore, these four state variables are fixed as x_R, y_R, z_R and ψ_R. Thus, a system defined around an equilibrium point $\boldsymbol{x}_R = [\boldsymbol{q}_R \quad \boldsymbol{0}]^T$ and reference forces $\boldsymbol{u}_R = \boldsymbol{\Gamma}_R$ is obtained by solving this system of nonlinear equations using the parameters shown in Table 2.1. When solving equations 3.36, only the solution with $f_R > 0$ and $f_L > 0$ were regarded. The equilibrium point was found to be at the following point:

$$\boldsymbol{q}_R = \begin{bmatrix} x_R \\ y_R \\ z_R \\ \phi_R \\ \theta_R \\ \psi_R \\ \alpha_{RR} \\ \alpha_{LR} \\ \gamma_{1R} \\ \gamma_{2R} \end{bmatrix} = \begin{bmatrix} x_R \\ y_R \\ z_R \\ 0.0018162 \\ 0.0348383 \\ \psi_R \\ -0.0353406 \\ -0.0340442 \\ -0.0018151 \\ -0.0348383 \end{bmatrix}, \qquad \boldsymbol{u}_R = \begin{bmatrix} f_{RR} \\ f_{LR} \\ \tau_{\alpha_R R} \\ \tau_{\alpha_L R} \end{bmatrix} = \begin{bmatrix} 8.0100 \\ 7.6839 \\ 0 \\ 0 \end{bmatrix}. \tag{3.37}$$

It can be seen that the values for ϕ_R, θ_R, α_{RR}, α_{LR}, γ_{1R} and γ_{2R} are not zero. In addition, $f_{RR} \neq f_{LR}$. This happened due to the fact that the center of mass of the tilt-rotor system is dislocated with respect to frame $\mathcal{B}$.

It is then possible to obtain a linearized error model given by:

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{B}_u\Delta\boldsymbol{u} + \boldsymbol{B}_d\boldsymbol{d}, \tag{3.38}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_r$ and $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_r$ with

$$\boldsymbol{A} = \frac{\partial \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial \boldsymbol{x}} \Big|_{\boldsymbol{x}=\boldsymbol{x}_R}^{\boldsymbol{u}=\boldsymbol{u}_R}, \quad \boldsymbol{B}_u = \frac{\partial \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial \boldsymbol{u}} \Big|_{\boldsymbol{x}=\boldsymbol{x}_R}^{\boldsymbol{u}=\boldsymbol{u}_R}, \quad \boldsymbol{B}_d = \frac{\partial \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial \boldsymbol{d}} \Big|_{\boldsymbol{x}=\boldsymbol{x}_R}^{\boldsymbol{u}=\boldsymbol{u}_R}.$$

3.3.2 Controllability of the Tilt-rotor UAV

By evaluating the controllability of the linearized system using the standard controllability test (equation (3.2)), the rank of the controllability matrix was found numerically to be equal to six instead of twenty-four, as desired. However, what really happens is that there is some ill-conditioning on the linear matrices. In fact, there are some big numbers together with small numbers and it is difficult to find a proper tolerance to compute numerically the *rank* of the controllability matrix.

On the other hand, by applying the PBH test of (3.3), it numerically easier to show that the system is indeed controllable. Moreover, when numerically computing the rank of the reduced controllability matrix given by $\begin{bmatrix} \boldsymbol{B} & \boldsymbol{AB} & \boldsymbol{A}^2\boldsymbol{B} & \boldsymbol{A}^3\boldsymbol{B} & \boldsymbol{A}^4\boldsymbol{B} & \boldsymbol{A}^5\boldsymbol{B} & \boldsymbol{A}^6\boldsymbol{B} \end{bmatrix}$, one finds twenty-four as result, what indicates that ill-conditioning appears when the *rank* of the full controllability matrix is calculated.

3.3.3 D-stable Controller Design

The controller that D-stabilizes the Tilt-rotor UAV with suspended load is obtained by finding matrices $\boldsymbol{Q} > 0$ and $\boldsymbol{Y}$ that simultaneously satisfy inequalities (3.19)-(3.21). It

was desired to obtain a system as fast as possible (σ as big as possible) with very low percentage overshoot ($\varphi << \pi/2$), and the actuators should not saturate when performing path track. The input parameters were then adjusted as $\sigma = -2.4$, $\varphi = \pi/10$, $c = 0$ and $r = 80$. Figure 3.2 show the poles in open loop along with its allocation after closed loop.

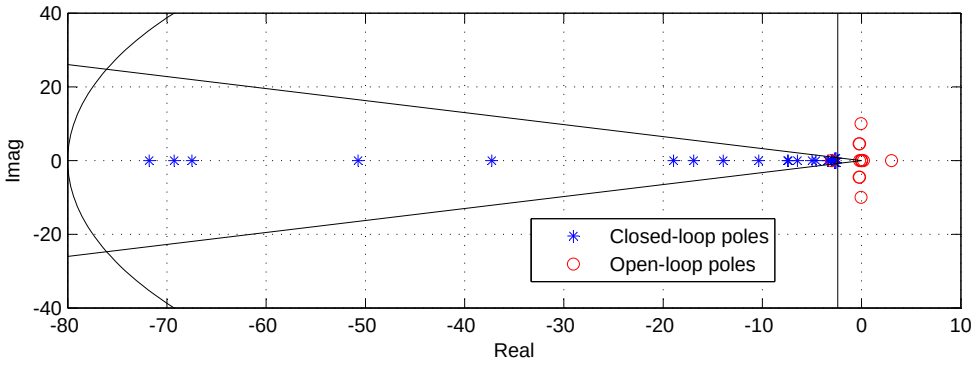

Figure 3.2: Allocation of poles via D-Stability.

3.3.4 D-Stable $\mathcal{H}_\infty$ Controller Design

The controller that D-stabilizes the Tilt-rotor UAV with suspended load maximizing disturbance rejection is obtained by finding matrices $\boldsymbol{Q} > 0$ and $\boldsymbol{Y}$ that simultaneously satisfy the inequalities (3.19)-(3.21) and (3.31), also minimizing $\tilde{\gamma}$. The following controller matrices were adjusted heuristically, taking as starting point Bryson's inverse-square method (Johnson and Grimble, 1987):

$$\begin{aligned}\boldsymbol{C}_z = diag(&3, 4, 5, 2, 2, 3, 0.2, 0.2, 3, 3, 2, 2,\\ &3, 1/3, 1/3, 3.5, 1/30, 1/30, 0.8, 0.8, 10.5, 10.5, 10, 10),\end{aligned} \tag{3.39}$$

$$\boldsymbol{D}_{wz} = \begin{bmatrix} 0_{10,3} & 0_{10,3} & 0_{10,2} & 0_{10,2} \\ \mathbb{I}_{3\times3} & 0_{3,3} & 0_{3,2} & 0_{3,2} \\ 0_{3,3} & \mathbb{I}_{3\times3} & 0_{3,2} & 0_{3,2} \\ 0_{2,3} & 0_{2,3} & 0_{2,2} & 0_{2,2} \\ 0_{2,3} & 0_{2,3} & 0_{2,2} & \mathbb{I}_{2\times2} \\ \mathbb{I}_{3\times3} & 0_{3,3} & 0_{3,2} & 0_{3,2} \\ 0_{1,3} & \boldsymbol{b} & 0_{1,2} & 0_{1,2} \end{bmatrix}, \qquad \boldsymbol{D}_{uz} = \begin{bmatrix} 0_{10,2} & 0_{10,2} \\ 0.5 * \mathbf{1}_{3\times2} & 0_{3,2} \\ 0_{2,2} & \mathbf{1}_{2\times2} \\ 0_{1,2} & 5 * \mathbf{1}_{1\times2} \\ 0_{2,2} & 5 * \mathbb{I}_{2\times2} \\ 0_{2,2} & 4 * \mathbf{1}_{2\times2} \end{bmatrix}. \tag{3.40}$$

with $\boldsymbol{b} = [0 \quad 0 \quad 1]$ and $\mathbf{1}_{n\times m}$ is a matrix with n lines and m columns filled with $1's$.

As a result, a solution with $||\boldsymbol{H}(s)||_\infty < 34.11$ was found. The pole allocation can be seen in Figure 3.3.

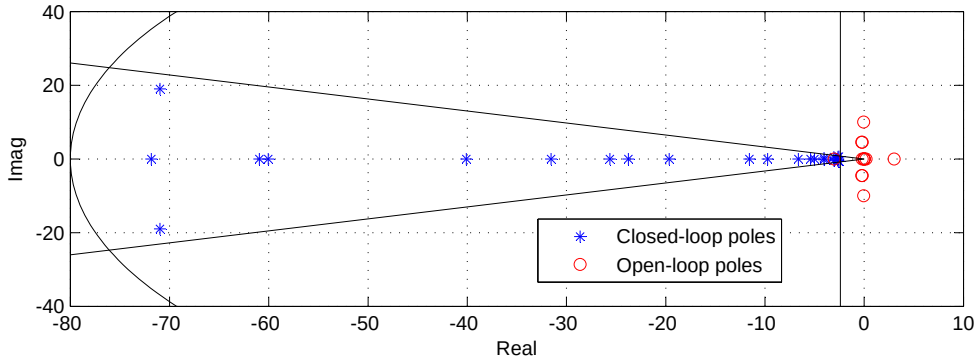

Figure 3.3: Allocation of poles via D-Stability with minimization of the $\mathcal{H}_\infty$ norm.

3.4 Simulation and Results

The proposed control systems were simulated in Matlab Simulink 2012a with the model parameters shown in Table 2.1 and with initial conditions starting already on the desired reference point. In order to show robustness of the designed controllers, the UAV tracked a predefined trajectory $x_{Ref}(t)$, $y_{Ref}(t)$, $z_{Ref}(t)$ and $\psi_{Ref}(t)$ while some disturbance forces ($Fx_{ext}(t)$, $Fy_{ext}(t)$ and $Fz_{ext}(t)$) affected the vehicle as forces applied at its geometric center (Figure 3.4). Moreover, the simulation considered that the model's masses m_i and inertia tensors I_i, for $i = 1, 2, 3, 4$, had all uncertainties ranging from -30% to 30% of their nominal values. In addition, a linear feed-forward term was added to the control action (Raffo et al., 2010):

$$\boldsymbol{u}_R = \boldsymbol{B}^+(\boldsymbol{q}_R)\{\boldsymbol{M}(\boldsymbol{q}_R)\ddot{\boldsymbol{x}}_{Ref} + [\boldsymbol{C}(\boldsymbol{q}_R, \dot{\boldsymbol{q}_R}) - \boldsymbol{\mu}]\dot{\boldsymbol{x}}_{Ref} + \boldsymbol{G}(\boldsymbol{q}_R)\}, \tag{3.41}$$

where $\boldsymbol{B}^+ = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T$ is the left pseudo-inverse of matrix $\boldsymbol{B}$. Thus, the system's inputs are given by the control law: $\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x} + \boldsymbol{u}_R$.

Figure 3.5 shows a 3D view of the trajectory followed by the aircraft. The set point ψ_{Ref} was kept constantly equal to zero. Figure 3.6 shows the tracking error of the generalized coordinates x, y, z and ψ defined as: $\varepsilon_q = q_{Ref} - q$. It can be seen from this image that the $\mathcal{H}_\infty$ controller presented better results on tracking reference and disturbance rejection when compared to the D-stable controller. The time evolution of the remaining generalized coordinates are shown in figure 3.7. It is possible to see that the designed control laws maintain all variables stabilized.

Figure 3.8 shows the system's control inputs along with time. It is possible to verify that the control effort computed by the $\mathcal{H}_\infty$ controller showed to be more aggressive than the pure D-stable controller.

Table 3.1 shows the mean square error of the trajectory of the aircraft in relation with its set point. The $\mathcal{H}_\infty$ controller presented 23% of the mean square error on the direction x with respect to the D-Stable controller, 24% in the direction y, 2.1% in ψ and it was 926% worse than the D-Stable controller on the z direction. Even though z worsened this much, the order of magnitude of its mean square error is still very small. We believe that

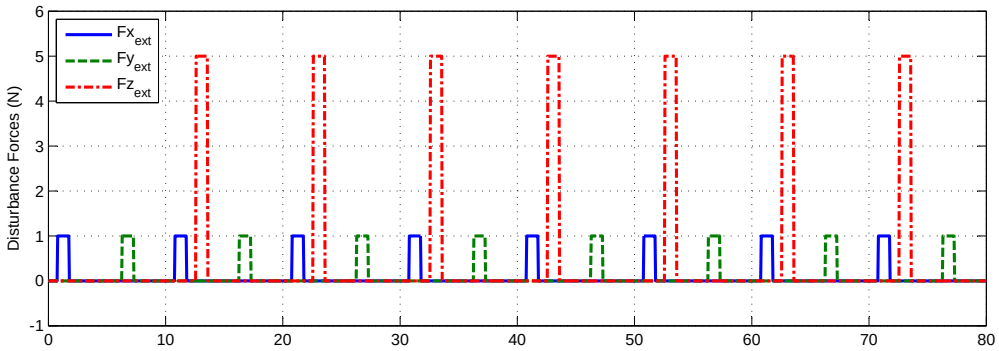

Figure 3.4: System disturbances in function of time.

Table 3.1: Mean-square-error comparison between D-stable and $\mathcal{H}_\infty$ controller.

	$D-stable$	H_∞	$\frac{MSE_{H_\infty}}{MSE_{D-Stable}}$
MSE_x	$3,85.10^{-4}$	$8,88.10^{-5}$	$0,2306$
MSE_y	$2,24.10^{-4}$	$5,56.10^{-5}$	$0,2491$
MSE_z	$4,90.10^{-6}$	$5,03.10^{-5}$	$10,263$
MSE_ψ	$6,03.10^{-2}$	$1,31.10^{-3}$	$0,0218$

the performance along z axis worsened due to the process of obtaining the $\mathcal{H}_\infty$ controller, since this approach tries to minimize only the maximum of the highest singular value, which means that other singular values may increase while minimizing the biggest one.

Table 3.2 presents the Integrated Absolute Variation of the Control signal (IAVU) index for both controllers. This index evaluates the control effort and is given by:

$$IAVU_i = \int_0^{t_f} \left| \frac{\mathrm{d}u_i}{\mathrm{d}t} \right| \mathrm{d}t. \tag{3.42}$$

It can be seen that the inputs for the $\mathcal{H}_\infty$, as observed in Figure 3.8, are more aggressive than the pure D-stable controller, which are smoother.

The linear controllers proved to be effective on stabilizing the Tilt-rotor UAV with suspended load when accomplishing the task of path tracking for the given trajectory. However, there are some drawbacks on using linear controllers. Given that the system should always be nearby the operating point, a large deviation from this neighborhood may destabilize the UAV. As consequence, the following occurrences showed to result on

Table 3.2: IAVU Index comparison between D-stable and $\mathcal{H}_\infty$ controller.

	$D-stable$	H_∞
$IAVUf_R$	$125,1231$	$6,4251 \cdot 10^3$
$IAVUf_L$	$128,6739$	$4,4195 \cdot 10^3$
$IAVU\tau_{\alpha_R}$	$0,1391$	$102,2888$
$IAVU\tau_{\alpha_L}$	$0,4368$	$209,8082$

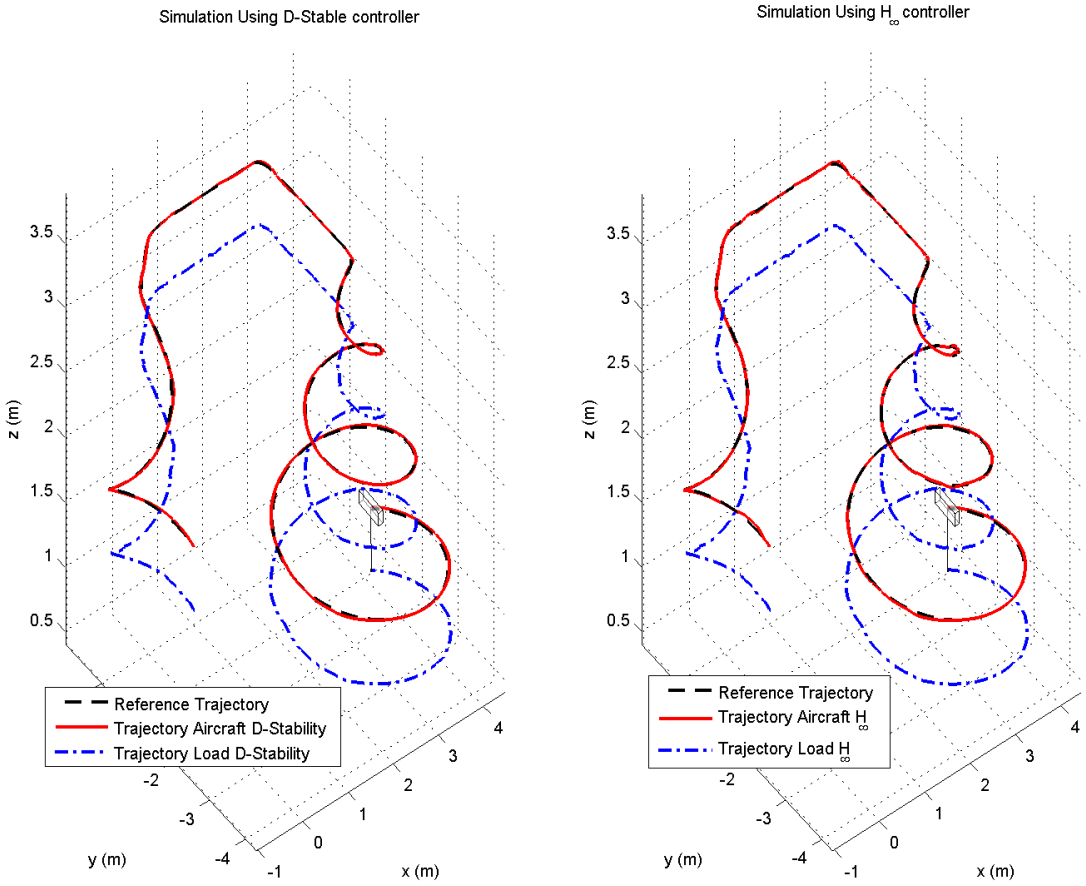

Figure 3.5: Path tracking of the aircraft for both D-stable and $\mathcal{H}_\infty$ controllers.

destabilization of the system:

- Non-smooth trajectories or trajectories that demanded high accelerations of the system;
- Harsher disturbances than the ones used on the simulations;
- When the system starts distant or gets distant from its reference.

Therefore, these disadvantages of linear controllers appeal for the design of nonlinear control strategies, which is the subject of the next chapter.

3.5 Conclusions

This chapter provided solutions for the problem of closed-loop linear control of the Tilt-rotor UAV with suspended load. Two strategies were proposed and both were able to stabilize the system while tracking a desired reference in the presence of exogenous disturbances, parametric uncertainties and unmodelled dynamics.

Basically, two linear controllers were designed using LMIs solvers: a D-stable and a D-stable with maximal disturbance rejection ($\mathcal{H}_\infty$ norm). Both controllers are expected

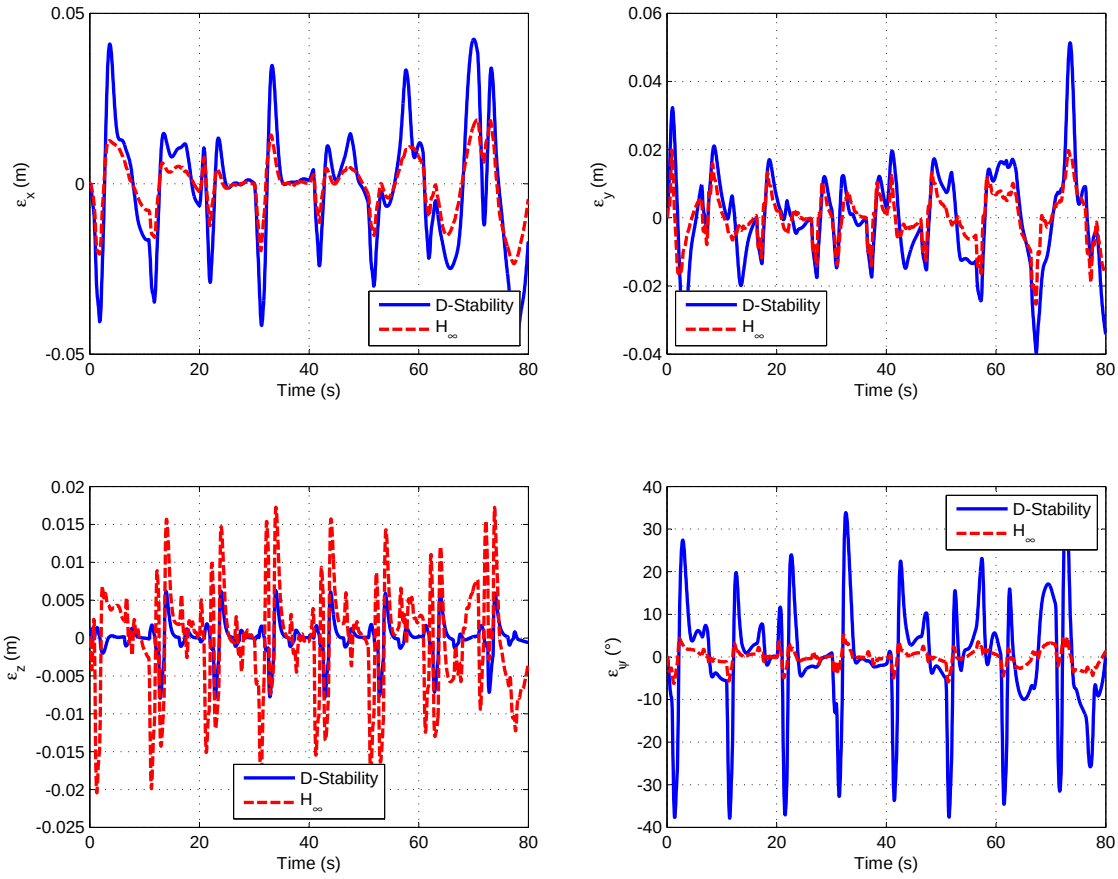

Figure 3.6: Tracking errors for D-stable and $\mathcal{H}_\infty$ controllers.

to work in the vicinity of the equilibrium point of the aircraft. They accomplished the task of path tracking, but the $\mathcal{H}_\infty$ was better on the presence of disturbances. On the other hand, the D-stable controller presented control inputs that are smoother with time than the inputs of the $\mathcal{H}_\infty$ controller. However, both linear controllers were not able to stabilize the aircraft when it deviated too much from its equilibrium point.

Next chapter will deal with this problem by exploring the development of nonlinear controllers so as to enlarge the domain of attraction of the control system.

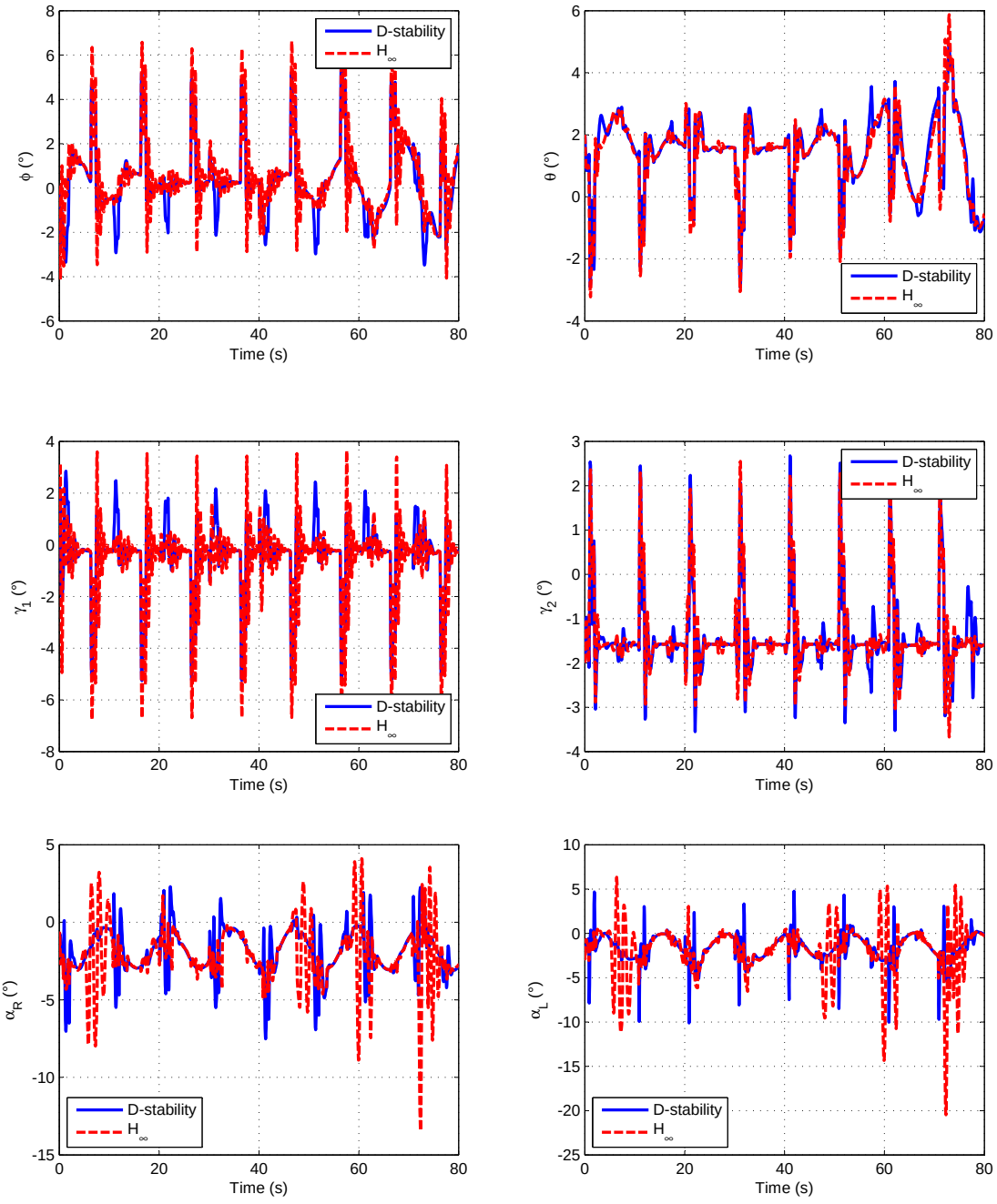

Figure 3.7: Body and Load angles for D-stable and $\mathcal{H}_\infty$ controllers.

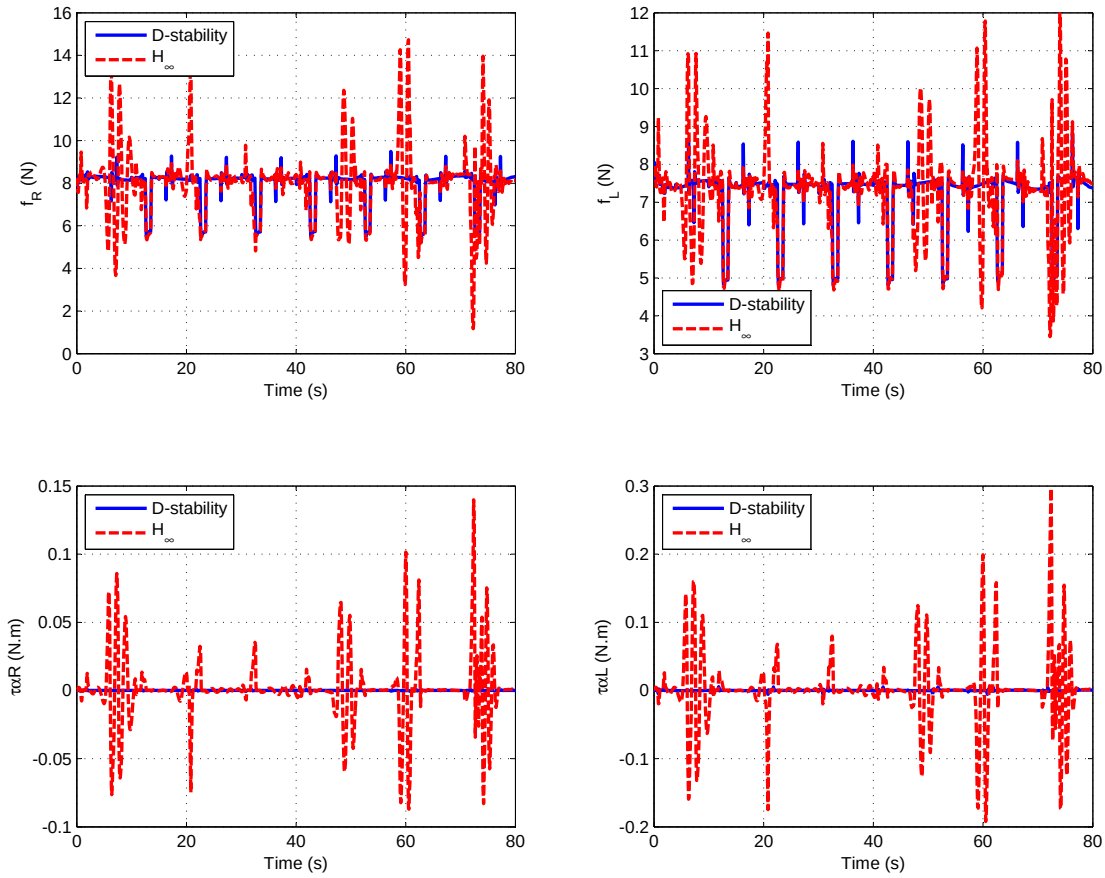

Figure 3.8: Inputs of the system for D-stable and $\mathcal{H}_\infty$ controllers.

Chapter 4

Nonlinear Control Strategies

Sumário

4.1 Introduction 47
4.2 Nonlinear Control Theory 48
4.2.1 Preliminary Theory on Nonlinear Systems 49
4.2.2 Input-Output Feedback Linearization 50
4.3 Tilt-rotor Nonlinear Control 54
4.3.1 First-level Feedback Linearization 54
4.3.2 Second-level Feedback Linearization 58
4.3.3 Third-level Feedback Linearization 61
4.3.4 Inverted Pendulum Control Strategy 69
4.4 Simulation Results 71
4.4.1 Comparison between NLPT and NLLS 71
4.4.2 Comparison between NLLS and Linear $\mathcal{H}_\infty$ 72
4.4.3 Inverted Pendulum 76
4.5 Conclusions 78

4.1 Introduction

The aim of this chapter is to provide some solutions for the problem of nonlinear closed-loop control of the Tilt-rotor UAV with suspended load. The objective of the proposed control strategies is to transport a payload using a Tilt-rotor UAV along a predefined trajectory using a control strategy, which should be able to stabilize the whole system, even when exogenous disturbances affect the system, and in presence of unmodelled dynamics and parametric uncertainties. The controller must provide path tracking for x, y, z and ψ whilst stabilizing all other generalized coordinates. In this chapter it is assumed that all twenty states of the UAV are precisely known such that state-feedback control strategies are designed.

Last chapter introduced some solutions for the control of the Tilt-rotor UAV using linear control theory. However, since the Tilt-rotor UAV with suspended load is a nonlinear system, it is expected that linear controllers provide limited performance when stabilizing the system, since they only work in the vicinity of its operation point. Therefore, in order to enlarge the domain of attraction of the control system, a nonlinear control strategy is

developed. This strategy is based on a cascade scheme with three input-output feedback linearization blocks, as shown in Figure 4.1. The rightmost feedback linearization block stabilizes the variables z, ϕ, α_R and α_L by actuating into f_R, f_L, τ_{α_R} and τ_{α_L}. The middle block stabilizes θ and ψ by actuating into the references of α_R and α_L. As for the leftmost block, it actuates on the references of ϕ and θ and there are two strategies on its design:

- The first one controls x and y but considers that the movement of γ_1 and γ_2 are both disturbances for the system. This strategy provides good path tracking of the aircraft, but the load strongly swings;
- The second strategy tries to control x and y at the same time that it stabilizes γ_1 and γ_2. This approach allows the load to swing less, but the aircraft deviates from set point in order to avoid the load's swing.

Finally, in order to show the robustness of the designed nonlinear strategy, some modifications are made in the controller so that the Tilt-rotor is able to stabilize an inverted pendulum.

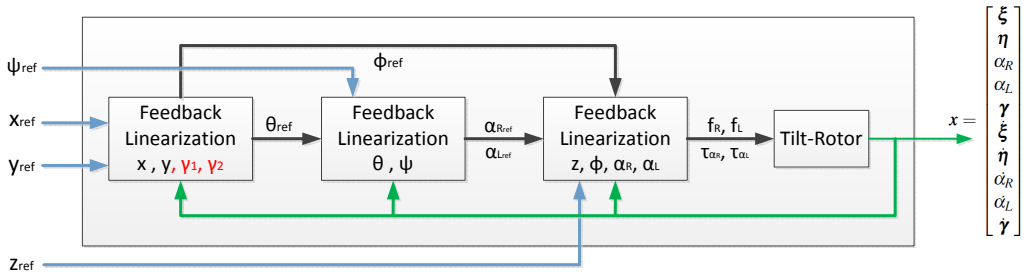

Figure 4.1: Nonlinear Feedback Linearization Cascade Control Strategy.

Section 4.2 introduces some important notions on nonlinear control theory, also presenting the *Input-Output Feedback Linearization* technique. Section 4.3 presents the nonlinear cascade control strategies for the aircraft. Section 4.4 shows simulation results and their analysis.

4.2 Nonlinear Control Theory

A nonlinear affine in the inputs system is described as follows:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_u(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{g}_d(\boldsymbol{x})\boldsymbol{d} \\ \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \end{cases}, \tag{4.1}$$

where $\boldsymbol{y} \in \Re^p$ is the output vector of the system and is function of the state space vector $\boldsymbol{x}$. The dynamic of $\dot{\boldsymbol{x}}$ was already obtained in section 2.5.

When designing a nonlinear controller, there are several methods for accomplishing such task. As examples, Slotine and Li (1991) mention the methods of *Trial and Error*,

Feedback Linearization, *Robust Control*, *Adaptive Control* and *Gain-scheduling*. This work focuses only on using the method of *Feedback Linearization* (FL).

The FL method depends on having a precise mathematical model of the system. Its basic idea is to transform the nonlinear system into a (full or partial) linear system so that linear controller design techniques can be applied. A disadvantage of this approach is that it does not guarantee robustness in the face of parameter uncertainty or disturbances.

When it is possible to apply a full FL to a system (all states' dynamics are linearized), then it is said that an Input-State Feedback Linearization (ISFL) is performed and the system can be fully controlled using a single control law. If, however, one can only linearize the outputs of the system, then it is said that an Input-Output Feedback Linearization (IOFL) is performed and only the linearized portion is controlled, while the rest is denoted *Internal Dynamics*. Extra caution must be taken when using IOFL because the designed control law may destabilize the Internal Dynamics. Therefore, one must prove that the internal dynamics are not unstable when using this approach (Spong et al., 2005).

Basically, the challenge of Feedback Linearization is to find a coordinate transformation $\boldsymbol{z} = \boldsymbol{\Phi}(\boldsymbol{x})$ and an input transformation based on feedback $\boldsymbol{v} = \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{u})$ such that the relation between the input ($\boldsymbol{v}$) and the linearized variables ($\boldsymbol{x}$ for ISFL and $\boldsymbol{y}$ for IOFL) is linear.

For the Tilt-rotor UAV it was not possible to verify conditions for ISFL[1]. Given that the linearized system was proven to be controllable on section 3.3.2, it is possible to say that the nonlinear system is at least locally controllable, but this is not a sufficient condition for the linearization of the nonlinear system (Respondek and Tall, 2002). Moreover, even if ISFL could be proven to be feasible for this system, it would still be necessary to obtain a diffeomorphism that transforms the system into a linear one. In general, this is a hard task (Hedrick and Girard, 2005), specially for systems with complicated equations of motion like the Tilt-rotor.

Consequently, only the IOFL methodology is evaluated on the present work. Subsection 4.2.1 provides some mathematical foundations that are further used on subsection 4.2.2, which shows the methodology to obtain IOFL.

4.2.1 Preliminary Theory on Nonlinear Systems

This section provides some important mathematical foundations used in geometric control. First, two important operations are defined: *Lie Derivative* and *Lie Bracket*. Then, the concept of *Diffeomorphism* is introduced. All the theory presented in this section is found in Slotine and Li (1991).

The *Lie Derivative* is defined as the derivative of a smooth scalar field $h(\boldsymbol{x})$ in the direction of a smooth vector field $\boldsymbol{f}(\boldsymbol{x})$, where $h(\boldsymbol{x}) : \Re^n \rightarrow \Re$ and $\boldsymbol{f}(\boldsymbol{x}) : \Re^n \rightarrow \Re^n$. The Lie Derivative results in a third scalar field and it is given by:

[1]The calculations to verify whether a system is Input-State Linearizable require the vector fields $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_u(\boldsymbol{x})$ as shown on equation (2.67). However, due to computational costs, it was not possible to obtain $\boldsymbol{M}^{-1}$, precluding the computation of these vector fields.

$$\mathcal{L}_{\boldsymbol{f}} h(\boldsymbol{x}) = \nabla h(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}), \tag{4.2}$$

where $\nabla h(\boldsymbol{x}) = \begin{bmatrix} \frac{\partial h(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial h(\boldsymbol{x})}{\partial x_n} \end{bmatrix}$ is the gradient of $h(\boldsymbol{x})$.

The *Lie Bracket* is defined for two smooth vector fields $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$, where $\boldsymbol{f}, \boldsymbol{g} : \Re^n \to \Re^n$. The Lie Bracket results in a third vector field and is given by:

$$\big[\boldsymbol{f}, \boldsymbol{g}\big] = ad_{\boldsymbol{f}}\boldsymbol{g} = \boldsymbol{D}_g \boldsymbol{f} - \boldsymbol{D}_f \boldsymbol{g}, \tag{4.3}$$

where $\boldsymbol{D}_g$ and $\boldsymbol{D}_f$ are, respectively, the Jacobian matrices associated to $\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) \\ \vdots \\ g_n(\boldsymbol{x}) \end{bmatrix}$ and $\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{bmatrix}$, and are given by:

$$\boldsymbol{D}_g = \begin{bmatrix} \frac{\partial g_1(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial g_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_n(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial g_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix}, \quad \boldsymbol{D}_f = \begin{bmatrix} \frac{\partial f_1(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial f_1(\boldsymbol{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(\boldsymbol{x})}{\partial x_1} & \dots & \frac{\partial f_n(\boldsymbol{x})}{\partial x_n} \end{bmatrix}. \tag{4.4}$$

A *Diffeomorphism* is determined as a smooth function $\Phi : \Re^n \to \Re^n$ defined in a region $\boldsymbol{\Omega}$ if its inverse Φ^{-1} uniquely exists and is also smooth. If the region $\boldsymbol{\Omega}$ is the whole state space $\Re^n$, then Φ is called a *global diffeomorphism*. If the Jacobian matrix $\boldsymbol{D}_\Phi$ is non-singular at a point $\boldsymbol{x} = \boldsymbol{x}_0$ of $\boldsymbol{\Omega}$, then $\boldsymbol{\Phi}(\boldsymbol{x})$ defines a *local diffeomorphism* in a subregion of $\boldsymbol{\Omega}$.

4.2.2 Input-Output Feedback Linearization

First, the notion of *relative degree* related to an output should be introduced. Thus, for the nonlinear system of the form:

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1(\boldsymbol{x})u_1 + \boldsymbol{g}_2(\boldsymbol{x})u_2 + \ldots + \boldsymbol{g}_m(\boldsymbol{x})u_m \\ \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} h_1(\boldsymbol{x}) \\ \vdots \\ h_m(\boldsymbol{x}) \end{bmatrix} \end{cases}, \tag{4.5}$$

where $\boldsymbol{x} \in \Re^n$ and $\boldsymbol{y} \in \Re^m$ (it should be noted that the dimension of $\boldsymbol{y}$ is equal to the dimension of $\boldsymbol{u}$, which is a prerequisite to IOFL). The relative degree r_i of output y_i is

defined as the number of times the output y_i needs to be differentiated until an input appears. Therefore, by differentiating y_i, the following is obtained:

$$\dot{y}_i = \mathcal{L}_{\boldsymbol{f}} h_i(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}_1} h_i(\boldsymbol{x}) u_1 + \ldots + \mathcal{L}_{\boldsymbol{g}_m} h_i(\boldsymbol{x}) u_m. \tag{4.6}$$

If $\mathcal{L}_{\boldsymbol{g}_j} h_i(\boldsymbol{x}) \neq 0$ for any $1 < j < m$, then the relative degree of y_i is 1. Otherwise, it has to be differentiated again:

$$\ddot{y}_i = \mathcal{L}_{\boldsymbol{f}}^2 h_i(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}_1} \mathcal{L}_{\boldsymbol{f}} h_i(\boldsymbol{x}) u_1 + \ldots + \mathcal{L}_{\boldsymbol{g}_m} \mathcal{L}_{\boldsymbol{f}} h_i(\boldsymbol{x}) u_m. \tag{4.7}$$

One should differentiate again and again until for some integer r_i it follows that $\mathcal{L}_{\boldsymbol{g}_j} \mathcal{L}_{\boldsymbol{f}}^{r_i - 1} h_i(\boldsymbol{x}) \neq 0$ for some $1 < j < m$ and $\boldsymbol{x} = \boldsymbol{x}_0$ in $\boldsymbol{\Omega}$. Then, the relative degree of output y_i is r_i at $\boldsymbol{x}_0$.

The relative degree of the system is given by the sum of all relative degrees:

$$r = \sum_{i=1}^{m} r_i. \tag{4.8}$$

Therefore, if $r = n$ then one can say that the system is fully feedback linearisable. If, otherwise, $r < n$, then only r states in new coordinates are controllable and the other $r - n$ states are denoted as the *Internal Dynamics* of the system. Extra caution must be taken when using IOFL since the designed control law may destabilize the Internal Dynamics. Therefore, one must check whether the internal dynamics are not unstable when using this approach.

In order to perform IOFL, the following matrices are defined:

$$\boldsymbol{\Delta}(\boldsymbol{x}) = \begin{bmatrix} \mathcal{L}_{\boldsymbol{g}_1} \mathcal{L}_{\boldsymbol{f}}^{r_1 - 1} h_1(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2} \mathcal{L}_{\boldsymbol{f}}^{r_1 - 1} h_1(\boldsymbol{x}) & \ldots & \mathcal{L}_{\boldsymbol{g}_m} \mathcal{L}_{\boldsymbol{f}}^{r_1 - 1} h_1(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{g}_1} \mathcal{L}_{\boldsymbol{f}}^{r_2 - 1} h_2(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2} \mathcal{L}_{\boldsymbol{f}}^{r_2 - 1} h_2(\boldsymbol{x}) & \ldots & \mathcal{L}_{\boldsymbol{g}_m} \mathcal{L}_{\boldsymbol{f}}^{r_2 - 1} h_2(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{L}_{\boldsymbol{g}_1} \mathcal{L}_{\boldsymbol{f}}^{r_m - 1} h_m(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2} \mathcal{L}_{\boldsymbol{f}}^{r_m - 1} h_m(\boldsymbol{x}) & \ldots & \mathcal{L}_{\boldsymbol{g}_m} \mathcal{L}_{\boldsymbol{f}}^{r_m - 1} h_m(\boldsymbol{x}) \end{bmatrix} \tag{4.9}$$

$$\boldsymbol{b} = \begin{bmatrix} \mathcal{L}_{\boldsymbol{f}}^{r_1} h_1(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{f}}^{r_2} h_2(\boldsymbol{x}) \\ \vdots \\ \mathcal{L}_{\boldsymbol{f}}^{r_m} h_m(\boldsymbol{x}) \end{bmatrix}. \tag{4.10}$$

Then, the following control law can be applied:

$$\boldsymbol{u} = \boldsymbol{\Delta}^{-1}(\boldsymbol{v} - \boldsymbol{b}), \tag{4.11}$$

where $\boldsymbol{v} \in \Re^m$ is the new control input. It should be noted that $\boldsymbol{\Delta}$ must be a nonsingular matrix. This way, the new state-space $\boldsymbol{z} \in \Re^r$ and its derivative $\dot{\boldsymbol{z}}$ look like the following:

$$\begin{cases} z_1 = h_1(\boldsymbol{x}) \\ z_2 = \mathcal{L}_{\boldsymbol{f}} h_1(\boldsymbol{x}) \\ \vdots \\ z_{r_1} = \mathcal{L}_{\boldsymbol{f}}^{r_1-1} h_1(\boldsymbol{x}) \\ z_{r_1+1} = h_2(\boldsymbol{x}) \\ z_{r_1+2} = \mathcal{L}_{\boldsymbol{f}} h_2(\boldsymbol{x}) \\ \vdots \\ z_{r_1+r_2} = \mathcal{L}_{\boldsymbol{f}}^{r_2-1} h_2(\boldsymbol{x}) \\ \vdots \\ z_{r_1+r_2+\ldots+r_{m-1}+1} = h_m(\boldsymbol{x}) \\ z_{r_1+r_2+\ldots+r_{m-1}+2} = \mathcal{L}_{\boldsymbol{f}} h_m(\boldsymbol{x}) \\ \vdots \\ z_r = \mathcal{L}_{\boldsymbol{f}}^{r_m-1} h_m(\boldsymbol{x}) \end{cases} \Longrightarrow \begin{cases} \dot{z}_1 = z_2 \\ \dot{z}_2 = z_3 \\ \vdots \\ \dot{z}_{r_1} = v_1 \\ \dot{z}_{r_1+1} = z_{r_1+2} \\ \dot{z}_{r_1+2} = z_{r_1+3} \\ \vdots \\ \dot{z}_{r_1+r_2} = v_2 \\ \vdots \\ \dot{z}_{r_1+r_2+\ldots+r_{m-1}+1} = z_{r_1+r_2+\ldots+r_{m-1}+2} \\ \dot{z}_{r_1+r_2+\ldots+r_{m-1}+2} = z_{r_1+r_2+\ldots+r_{m-1}+3} \\ \vdots \\ \dot{z}_r = v_m \end{cases} . \tag{4.12}$$

The complete state-space representation of the system becomes:

$$\dot{\boldsymbol{\zeta}} = \begin{bmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{\vartheta}} \end{bmatrix} \Longrightarrow \begin{cases} \dot{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{v} \\ \dot{\boldsymbol{\vartheta}} = \mathcal{L}_{\boldsymbol{f}} \boldsymbol{\vartheta}(\boldsymbol{x}) + \mathcal{L}_{\boldsymbol{g}} \boldsymbol{\vartheta}(\boldsymbol{x}) \boldsymbol{v} \end{cases} , \tag{4.13}$$

where the equations for $\dot{\boldsymbol{\vartheta}} \in \Re^{n-r}$ represent the internal dynamics. The variables $\boldsymbol{\vartheta}$ of the internal dynamics are functions of $\boldsymbol{x}$ and they should be chosen such that they complete the diffeomorphism, i.e., $rank(\boldsymbol{D}_\zeta) = n$, where $\boldsymbol{D}_\zeta$ is the Jacobian matrix of the nonlinear change of coordinates:

$$\boldsymbol{\zeta} = \begin{bmatrix} h_1(\boldsymbol{x}) \\ h_2(\boldsymbol{x}) \\ \vdots \\ h_m(\boldsymbol{x}) \\ \vartheta_1(\boldsymbol{x}) \\ \vartheta_2(\boldsymbol{x}) \\ \vdots \\ \vartheta_{n-m}(\boldsymbol{x}) \end{bmatrix} \tag{4.14}$$

According to Slotine and Li (1991), it is always possible find $n-r$ functions $\boldsymbol{\vartheta}(\boldsymbol{x})$ such that:

$$\mathcal{L}_{\boldsymbol{g}}\vartheta_k(\boldsymbol{x}) = 0, \quad \forall\, 1 < k < n - r, \tag{4.15}$$

what implies that the second term on equation (4.13) is equal to zero. However, it is not always an easy task to find $\boldsymbol{\vartheta}(\boldsymbol{x})$ that satisfies the constraints imposed by (4.15).

It is possible to obtain the same results of equation (4.13) without the need to use Lie algebra. This is possible by using the Partial Feedback Linearization (PFL) formulation (Olfati-Saber, 2001). This approach is useful when dealing with mechanical systems whose dynamics are modelled using the Euler-Lagrange formulation, as shown in equation (2.9). For the sake of simplicity, the following is defined: $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = [\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{\mu}]\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$. Then, considering null external disturbances, the following Euler-Lagrange representation is obtained:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{F}(\boldsymbol{q}). \tag{4.16}$$

Then, separating the controlled part of the system (denoted with subscript c) from the uncontrolled part of the system (denoted by subscript u) yields:

$$\begin{bmatrix} \boldsymbol{M}_{uu}(\boldsymbol{q}) & \boldsymbol{M}_{uc}(\boldsymbol{q}) \\ \boldsymbol{M}_{cu}(\boldsymbol{q}) & \boldsymbol{M}_{cc}(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}}_u \\ \ddot{\boldsymbol{q}}_c \end{bmatrix} + \begin{bmatrix} \boldsymbol{h}_u(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{h}_c(\boldsymbol{q}, \dot{\boldsymbol{q}}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{B}_u(\boldsymbol{q}) \\ \boldsymbol{B}_c(\boldsymbol{q}) \end{bmatrix} \boldsymbol{\Gamma}. \tag{4.17}$$

It is possible to rewrite (4.17) in the form:

$$\boldsymbol{M}_{uu}\ddot{\boldsymbol{q}}_u + \boldsymbol{M}_{uc}\ddot{\boldsymbol{q}}_c + \boldsymbol{h}_u = \boldsymbol{B}_u\boldsymbol{\Gamma} \tag{4.18}$$

$$\boldsymbol{M}_{cu}\ddot{\boldsymbol{q}}_u + \boldsymbol{M}_{cc}\ddot{\boldsymbol{q}}_c + \boldsymbol{h}_c = \boldsymbol{B}_c\boldsymbol{\Gamma}. \tag{4.19}$$

If $\boldsymbol{M}_{uu}$ is invertible, then it is possible to rewrite equation (4.18) as $\ddot{\boldsymbol{q}}_u = \boldsymbol{M}_{uu}^{-1}(\boldsymbol{B}_u\boldsymbol{\Gamma} - \boldsymbol{M}_{uc}\ddot{\boldsymbol{q}}_c - \boldsymbol{h}_u)$ and substituting it into (4.19) leads to:

$$(\boldsymbol{M}_{cc} - \boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{M}_{uc})\ddot{\boldsymbol{q}}_c = (\boldsymbol{B}_c - \boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{B}_u)\boldsymbol{\Gamma} + (\boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{h}_u - \boldsymbol{h}_c). \tag{4.20}$$

Then, by defining:

$$\boldsymbol{\Delta} = (\boldsymbol{B}_c - \boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{B}_u), \tag{4.21}$$

and if $\boldsymbol{\Delta}$ is proven to be nonsingular, the following can also be defined:

$$\boldsymbol{\alpha} = \boldsymbol{\Delta}^{-1}(\boldsymbol{M}_{cc} - \boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{M}_{uc}) \tag{4.22}$$
$$\boldsymbol{\beta} = \boldsymbol{\Delta}^{-1}(-\boldsymbol{M}_{cu}\boldsymbol{M}_{uu}^{-1}\boldsymbol{h}_u + \boldsymbol{h}_c). \tag{4.23}$$

It is then possible to choose a control law as given by:

$$\boldsymbol{\Gamma} = \boldsymbol{\alpha}\boldsymbol{v} + \boldsymbol{\beta}, \tag{4.24}$$

with $\boldsymbol{v} \in \Re^m$ being the new control input. This leads to a linear relation between the controlled part of the system $\ddot{\boldsymbol{q}}_c$ and the new input $\boldsymbol{v}$:

$$\ddot{\boldsymbol{q}}_c = \boldsymbol{v}. \tag{4.25}$$

By substituting (4.24) into (4.18), it is possible to obtain the behaviour of the internal dynamics:

$$\ddot{\boldsymbol{q}}_u = \boldsymbol{M}_{uu}^{-1}(\boldsymbol{B}_u\boldsymbol{\beta} - \boldsymbol{h}_u) + [\boldsymbol{M}_{uu}^{-1}(\boldsymbol{B}_u\boldsymbol{\alpha} - \boldsymbol{M}_{uc})]\boldsymbol{v}. \tag{4.26}$$

4.3 Tilt-rotor Nonlinear Control

This section deals with the development of nonlinear control strategies for the Tilt-rotor UAV with suspended load. As already shown in Figure 4.1, the strategy uses a three-level cascade control techhnique based on IOFL. Subsection 4.3.1 shows the calculations for the innermost IOFL, which controls z, ϕ, α_R and α_L, while actuating on f_R, f_L, τ_{α_R} and τ_{α_L}. Subsection 4.3.2 takes care of the middle IOFL, which controls θ and ψ by actuating on $\alpha_{R_{ref}}$ and $\alpha_{L_{ref}}$. Subsection 4.3.3 discusses strategies to control the remaining variables (x, y, γ_1 and γ_2) by actuating on ϕ_{ref} and θ_{ref}. Finally, subsection 4.3.4 suggests some modification to what was designed on subsection 4.3.3 so that the Tilt-rotor is able to stabilize an inverted pendulum.

4.3.1 First-level Feedback Linearization

This section presents the designed control strategy to stabilize the outputs z, ϕ, α_R and α_L at their operation points, while actuating on the system's inputs $\boldsymbol{u} = \begin{bmatrix} f_R & f_L & \tau_{\alpha_R} & \tau_{\alpha_L} \end{bmatrix}^T$.

This choice of outputs $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} z & \phi & \alpha_R & \alpha_L \end{bmatrix}^T$ may seem peculiar at a first glance but this is the most natural choice given the system's inputs. First of all, matrix $\boldsymbol{\Delta}(\boldsymbol{x})$ of equation (4.9) has to be full rank, what forces us to use α_R and α_L as outputs so that the inputs τ_{α_R} and τ_{α_L} are used to drive any output. As for the other two variables,

it is interesting to choose outputs that are directly affected by the inputs f_R and f_L and their combinations. Thus, the choice of z and ϕ as remaining outputs is natural.

Thus, using the defined inputs and outputs of the system, its nonlinear model can be written as (assuming null external disturbances):

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_1(\boldsymbol{x})f_R + \boldsymbol{g}_2(\boldsymbol{x})f_L + \boldsymbol{g}_3(\boldsymbol{x})\tau_{\alpha_R} + \boldsymbol{g}_4(\boldsymbol{x})\tau_{\alpha_L} \\ \boldsymbol{y} = \begin{bmatrix} h_1(\boldsymbol{x}) \\ h_2(\boldsymbol{x}) \\ h_3(\boldsymbol{x}) \\ h_4(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} z \\ \phi \\ \alpha_R \\ \alpha_L \end{bmatrix} \end{cases}, \tag{4.27}$$

where

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}) &= \begin{bmatrix} \boldsymbol{f}(1) \\ \vdots \\ \boldsymbol{f}(20) \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \boldsymbol{M}^{-1}[-[\boldsymbol{C} - \boldsymbol{\mu}]\dot{\boldsymbol{q}} - \boldsymbol{G}] \end{bmatrix}, \\ \boldsymbol{g}_u(\boldsymbol{x}) &= \begin{bmatrix} \boldsymbol{g}_1(1) & \boldsymbol{g}_2(1) & \boldsymbol{g}_3(1) & \boldsymbol{g}_4(1) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{g}_1(20) & \boldsymbol{g}_2(20) & \boldsymbol{g}_3(20) & \boldsymbol{g}_4(20) \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}_{10\times 4} \\ \boldsymbol{M}^{-1}\boldsymbol{B} \end{bmatrix}. \end{aligned} \tag{4.28}$$

Calculating the relative degree of output $h_1(\boldsymbol{x}) = z$, the following is obtained:

$$\mathcal{L}_{\boldsymbol{g}_u} h_1(\boldsymbol{x}) = \begin{bmatrix} 0 & 0 & 1 & \boldsymbol{0}_{1\times 17} \end{bmatrix} \begin{bmatrix} \boldsymbol{0}_{10\times 4} \\ \boldsymbol{M}^{-1}\boldsymbol{B} \end{bmatrix} = 0 \tag{4.29}$$

$$\mathcal{L}_{\boldsymbol{g}_u}\mathcal{L}_{\boldsymbol{f}} h_1(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{0_{1\times 12}} & 1 & \boldsymbol{0_{1\times 7}} \end{bmatrix} \begin{bmatrix} \boldsymbol{0}_{10\times 4} \\ \boldsymbol{M}^{-1}\boldsymbol{B} \end{bmatrix} \neq 0. \tag{4.30}$$

Therefore, the relative degree $r_1 = 2$. The same result is obtained for the other three outputs. Therefore, the system's relative degree is $r = \sum r_i = 8$. The matrices $\boldsymbol{\Delta}(\boldsymbol{x})$ and $\boldsymbol{b}$ are given by:

$$\boldsymbol{\Delta}(\boldsymbol{x}) = \begin{bmatrix} \mathcal{L}_{\boldsymbol{g}_1}\mathcal{L}_{\boldsymbol{f}}h_1(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2}\mathcal{L}_{\boldsymbol{f}}h_1(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_3}\mathcal{L}_{\boldsymbol{f}}h_1(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_4}\mathcal{L}_{\boldsymbol{f}}h_1(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{g}_1}\mathcal{L}_{\boldsymbol{f}}h_2(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2}\mathcal{L}_{\boldsymbol{f}}h_2(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_3}\mathcal{L}_{\boldsymbol{f}}h_2(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_4}\mathcal{L}_{\boldsymbol{f}}h_2(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{g}_1}\mathcal{L}_{\boldsymbol{f}}h_3(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2}\mathcal{L}_{\boldsymbol{f}}h_3(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_3}\mathcal{L}_{\boldsymbol{f}}h_3(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_4}\mathcal{L}_{\boldsymbol{f}}h_3(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{g}_1}\mathcal{L}_{\boldsymbol{f}}h_4(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_2}\mathcal{L}_{\boldsymbol{f}}h_4(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_3}\mathcal{L}_{\boldsymbol{f}}h_4(\boldsymbol{x}) & \mathcal{L}_{\boldsymbol{g}_4}\mathcal{L}_{\boldsymbol{f}}h_4(\boldsymbol{x}) \end{bmatrix} \tag{4.31}$$

$$= \begin{bmatrix} \boldsymbol{g}_1(13) & \boldsymbol{g}_2(13) & \boldsymbol{g}_3(13) & \boldsymbol{g}_4(13) \\ \boldsymbol{g}_1(14) & \boldsymbol{g}_2(14) & \boldsymbol{g}_3(14) & \boldsymbol{g}_4(14) \\ \boldsymbol{g}_1(17) & \boldsymbol{g}_2(17) & \boldsymbol{g}_3(17) & \boldsymbol{g}_4(17) \\ \boldsymbol{g}_1(18) & \boldsymbol{g}_2(18) & \boldsymbol{g}_3(18) & \boldsymbol{g}_4(18) \end{bmatrix} \tag{4.32}$$

$$\boldsymbol{b} = \begin{bmatrix} \mathcal{L}_{\boldsymbol{f}}^2 h_1(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{f}}^2 h_2(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{f}}^2 h_3(\boldsymbol{x}) \\ \mathcal{L}_{\boldsymbol{f}}^2 h_4(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}(13) \\ \boldsymbol{f}(14) \\ \boldsymbol{f}(17) \\ \boldsymbol{f}(18) \end{bmatrix}. \tag{4.33}$$

In order to analyse $rank(\boldsymbol{\Delta}(\boldsymbol{x}))$, it was chosen to calculate it in the equilibrium point found at section 3.3.1. Therefore, it is possible to obtain $rank(\boldsymbol{\Delta}(\boldsymbol{x}_R))$ by calculating first $\boldsymbol{g}_u(\boldsymbol{x}_R) = \begin{bmatrix} \boldsymbol{0}_{10\times 4} \\ \boldsymbol{M}^{-1}(\boldsymbol{x}_R)\boldsymbol{B}(\boldsymbol{x}_R) \end{bmatrix}$ and then evaluating the rank given by the thirteenth, fourteenth, seventeenth and eighteenth lines of $\boldsymbol{g}_u(\boldsymbol{x}_R)$. It was observed that indeed $rank(\boldsymbol{\Delta}(\boldsymbol{x}_R)) = 4$.

As a result, the new inputs $\boldsymbol{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_4 \end{bmatrix}$ are related to the system's states as follows:

$$\ddot{z} = v_1, \quad \ddot{\phi} = v_2, \quad \ddot{\alpha}_R = v_3, \quad \ddot{\alpha}_L = v_4. \tag{4.34}$$

The inputs v_2, v_3 and v_4 were designed to perform a Proportional-Derivative (PD) control, while v_1 performs a Proportional-Integral-Derivative (PID) control. This choice was made due to the fact that ϕ, α_R and α_L need only to be stabilized, while z should track desired trajectories and reject constant disturbances. Moreover, a feed-forward term is also included in all of them, and their behaviour are given by:

$$v_1 = \ddot{x}_{3_{Ref}} + K_d\dot{e}_3 + K_p e_3 + K_I \int e_3 \, \mathrm{d}t \tag{4.35}$$

$$v_i = \ddot{x}_{i_{Ref}} + K_d\dot{e}_i + K_p e_i, \quad i = 2, 3, 4. \tag{4.36}$$

where $e_i = (x_i - x_{i_{Ref}})$ and x_i is the i^{th} state space variable. The K_d, K_p and K_I parameters were chosen so that the closed-loop poles were inside D-stable regions. One D-stable region was tuned for each of these variables according to desired closed-loop characteristics for them. As a result, the following associated characteristic equations were obtained:

$$\begin{aligned} z &: (s+9.695)(s+6.405)(s+5.170) = 0 \\ \phi &: (s+19.606)(s+15.692) = 0 \\ \alpha_R, \alpha_L &: (s+92.977)(s+43.012) = 0. \end{aligned} \tag{4.37}$$

As for the internal dynamics, a base completion was performed using the trivial solution:

$$\begin{cases} \vartheta_1 = x \\ \vartheta_2 = y \\ \vartheta_3 = \theta \\ \vartheta_4 = \psi \\ \vartheta_5 = \gamma_1 \\ \vartheta_6 = \gamma_2 \\ \vartheta_7 = \dot{x} \\ \vartheta_8 = \dot{y} \\ \vartheta_9 = \dot{\theta} \\ \vartheta_{10} = \dot{\psi} \\ \vartheta_{11} = \dot{\gamma}_1 \\ \vartheta_{12} = \dot{\gamma}_2 \end{cases} \Longrightarrow \begin{cases} \dot{\vartheta}_1 = \mathcal{L}_{\boldsymbol{f}}\vartheta_1 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_1\boldsymbol{u} \\ \dot{\vartheta}_2 = \mathcal{L}_{\boldsymbol{f}}\vartheta_2 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_2\boldsymbol{u} \\ \dot{\vartheta}_3 = \mathcal{L}_{\boldsymbol{f}}\vartheta_3 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_3\boldsymbol{u} \\ \dot{\vartheta}_4 = \mathcal{L}_{\boldsymbol{f}}\vartheta_4 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_4\boldsymbol{u} \\ \dot{\vartheta}_5 = \mathcal{L}_{\boldsymbol{f}}\vartheta_5 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_5\boldsymbol{u} \\ \dot{\vartheta}_6 = \mathcal{L}_{\boldsymbol{f}}\vartheta_6 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_6\boldsymbol{u} \\ \dot{\vartheta}_7 = \mathcal{L}_{\boldsymbol{f}}\vartheta_7 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_7\boldsymbol{u} \\ \dot{\vartheta}_8 = \mathcal{L}_{\boldsymbol{f}}\vartheta_8 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_8\boldsymbol{u} \\ \dot{\vartheta}_9 = \mathcal{L}_{\boldsymbol{f}}\vartheta_9 + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_9\boldsymbol{u} \\ \dot{\vartheta}_{10} = \mathcal{L}_{\boldsymbol{f}}\vartheta_{10} + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_{10}\boldsymbol{u} \\ \dot{\vartheta}_{11} = \mathcal{L}_{\boldsymbol{f}}\vartheta_{11} + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_{11}\boldsymbol{u} \\ \dot{\vartheta}_{12} = \mathcal{L}_{\boldsymbol{f}}\vartheta_{12} + \mathcal{L}_{\boldsymbol{g}_u}\vartheta_{12}\boldsymbol{u} \end{cases} . \tag{4.38}$$

In order to analyse the stability of the internal dynamics, it is necessary to calculate the Lie Derivatives as shown in (4.38). However, these derivatives can only be calculated if the nonlinear vector fields $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_u(\boldsymbol{x})$ are known. As shown in equation (4.28), $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{g}_u(\boldsymbol{x})$ are functions of $\boldsymbol{M}^{-1}$, what means that the nonlinear inverse of the inertia matrix has to be calculated so as to obtain these vector fields. Due to the computational complexity, this task was not accomplished. Alternatively, an attempt to evaluate the behaviour of the internal dynamics using PFL (equation (4.26)) was also performed, but it was also too expensive computationally.

An alternative solution is to try to obtain a linearized state space representation of the whole system in closed loop, where now the inputs are z_{Ref}, ϕ_{Ref}, $\alpha_{R_{Ref}}$ and $\alpha_{L_{Ref}}$. It is possible to numerically obtain an approximate state space linearization by using the *numerical perturbation method*[2] in the vicinity of the equilibrium point. This method finds a linear representation of a nonlinear system by applying small variations in the system's inputs and states, so that the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ can be obtained. This procedure was done in the neighborhood of the equilibrium point and found a characteristic equation with two unstable and six marginally stable poles. When simulating the system, it was possible to verify that the variables θ and $\dot{\theta}$ are unstable, while x, y, ψ and their derivatives are marginally stable. The remaining variables are stable.

[2]This method is already implemented in Matlab and further information can be found at http://www.mathworks.com/help/slcontrol/ug/exact-linearization-algorithm.html.

Next subsection shows that when the variables θ and ψ are controlled by actuating on $\alpha_{R_{ref}}$ and $\alpha_{L_{ref}}$, the unstable poles disappear and the marginally stable poles are reduced to 4.

4.3.2 Second-level Feedback Linearization

This section presents the performed strategy to stabilize the outputs θ and ψ around an equilibrium point, while actuating on the references $\alpha_{R_{Ref}}$ and $\alpha_{L_{Ref}}$.

In the previous section, it was not possible to calculate the internal dynamics of the system. Therefore, one cannot obtain the precise model of the system that maps from the inputs $\alpha_{R_{ref}}$ and $\alpha_{L_{ref}}$ to the outputs θ and ψ. Thus, in order to circumvent this problem it was used a reduced Euler-Lagrange model for the system, neglecting some coupling effects. The neglected dynamic couplings are then considered modelling errors on the control design. First, it is assumed that there is no coupling between the attitude variables and the remaining generalized coordinates. This way, the Euler-Lagrange equations can be written for the attitude variables in the form:

$$\left[\boldsymbol{W}_\eta^T \boldsymbol{J} \boldsymbol{W}_\eta\right] \ddot{\eta} + \begin{bmatrix} c_{44} & c_{45} & c_{46} \\ c_{54} & c_{55} & c_{56} \\ c_{64} & c_{65} & c_{66} \end{bmatrix} \dot{\eta} + \begin{bmatrix} g_4 \\ g_5 \\ g_6 \end{bmatrix} = \boldsymbol{W}_\eta^T \begin{bmatrix} \boldsymbol{\tau}_R & \boldsymbol{\tau}_L & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \begin{bmatrix} f_r \\ f_L \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \end{bmatrix}, \quad (4.39)$$

where c_{ij} is the element of the coriolis and centripetal matrix located at the i^{th} line and j^{th} column. The same applies g_i which represent terms for the gravity vector.

Assuming that $\boldsymbol{W}_\eta^T$ is invertible (this is true for $|\theta| \neq \pi/2$), equation (4.39) can be rewritten as:

$$\left[\boldsymbol{J} \boldsymbol{W}_\eta\right] \ddot{\eta} + \begin{bmatrix} \bar{c}_{44} & \bar{c}_{45} & \bar{c}_{46} \\ \bar{c}_{54} & \bar{c}_{55} & \bar{c}_{56} \\ \bar{c}_{64} & \bar{c}_{65} & \bar{c}_{66} \end{bmatrix} \dot{\eta} + \begin{bmatrix} \bar{g}_4 \\ \bar{g}_5 \\ \bar{g}_6 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_R & \boldsymbol{\tau}_L & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \begin{bmatrix} f_r \\ f_L \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \end{bmatrix}. \quad (4.40)$$

Since the roll angle ϕ was already controlled with the first feedback-linearization block, its dynamics and coupling will be neglected from now on. In addition, substituting the values of $\boldsymbol{\tau}_R$ and $\boldsymbol{\tau}_L$, the following is obtained:

$$\begin{bmatrix} \bar{m}_{55} & \bar{m}_{56} \\ \bar{m}_{65} & \bar{m}_{66} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \bar{c}_{55} & \bar{c}_{56} \\ \bar{c}_{65} & \bar{c}_{66} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \bar{g}_5 \\ \bar{g}_6 \end{bmatrix} = \begin{bmatrix} s_{\alpha_R} d_z f_R + s_{\alpha_L} d_z f_L \\ s_{\alpha_R} d_y f_R - s_{\alpha_L} d_y f_L \end{bmatrix} - \frac{k_\tau}{b} \begin{bmatrix} s_\beta(-c_{\alpha_R} f_R + c_{\alpha_L} f_L) \\ c_\beta(-c_{\alpha_R} f_R + c_{\alpha_L} f_L) \end{bmatrix}. \quad (4.41)$$

It should be noticed that there are some input torques that are generated by the thrust

forces of the propellers, while other are result from the drag effect due to the propellers' rotation. For simplicity, the drag terms are going to be considered as known disturbances which should be compensated by the controller. Therefore, by defining:

$$\begin{bmatrix} h_5 \\ h_6 \end{bmatrix} = \begin{bmatrix} \bar{c}_{55} & \bar{c}_{56} \\ \bar{c}_{65} & \bar{c}_{66} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \bar{g}_5 \\ \bar{g}_6 \end{bmatrix} + \frac{k_\tau}{b} \begin{bmatrix} s_\beta(-c_{\alpha_R} f_R + c_{\alpha_L} f_L) \\ c_\beta(-c_{\alpha_R} f_R + c_{\alpha_L} f_L) \end{bmatrix}. \tag{4.42}$$

Assuming that the dynamics of α_R and α_L are much faster than the dynamics of θ and ψ, then it is possible to approximate $\alpha_R = \alpha_{R_{Ref}}$ and $\alpha_L = \alpha_{L_{Ref}}$ and equation (4.41) can be written as follows:

$$\begin{bmatrix} \bar{m}_{55} & \bar{m}_{56} \\ \bar{m}_{65} & \bar{m}_{66} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} h_5 \\ h_6 \end{bmatrix} = \begin{bmatrix} d_z f_R & d_z f_L \\ d_y f_R & -d_y f_L \end{bmatrix} \begin{bmatrix} s_{\alpha_{R_{Ref}}} \\ s_{\alpha_{L_{Ref}}} \end{bmatrix}. \tag{4.43}$$

Therefore, a new system representation is obtained with inputs $\sin(\alpha_{R_{Ref}})$ and $\sin(\alpha_{L_{Ref}})$. As consequence, one can regulate θ and ψ by demanding references on the angles α_R and α_L. In order to eliminate the *sine* operations in (4.43), the following transformation is applied:

$$\alpha_{R_{Ref}} = \arcsin\left[\frac{1}{2f_R}\left(\frac{\Upsilon_1 + h_5}{d_z} + \frac{\Upsilon_2 + h_6}{d_y}\right)\right] \tag{4.44}$$

$$\alpha_{L_{Ref}} = \arcsin\left[\frac{1}{2f_L}\left(\frac{\Upsilon_1 + h_5}{d_z} - \frac{\Upsilon_2 + h_6}{d_y}\right)\right]. \tag{4.45}$$

Substituting equations (4.44)-(4.45) into (4.43), the following is obtained:

$$\begin{bmatrix} \bar{m}_{55} & \bar{m}_{56} \\ \bar{m}_{65} & \bar{m}_{66} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \end{bmatrix}. \tag{4.46}$$

It should be noticed that one must be careful when using the relation given by equations (4.44)-(4.45). Given that the operation $\arcsin(w)$ is defined only for $-1 < w < 1$, it is important to observe whether this operation is being calculated within the permitted region. This work overcomes this problem by using the same solution proposed by Bhanja Chowdhury et al. (2012), who uses a saturation function $\sigma(p)$ as follows:

$$\sigma(p) = \begin{Bmatrix} 1 & if & p > 1 \\ p & if & -1 < p < 1 \\ -1 & if & p < -1 \end{Bmatrix}. \tag{4.47}$$

Thus, the real transformation for $\alpha_{R_{Ref}}$ and $\alpha_{L_{Ref}}$ is given by:

$$\alpha_{R_{Ref}} = \arcsin\left[\sigma\left(\frac{1}{2f_R}\left(\frac{\Upsilon_1+h_5}{d_z} + \frac{\Upsilon_2+h_6}{d_y}\right)\right)\right] \tag{4.48}$$

$$\alpha_{L_{Ref}} = \arcsin\left[\sigma\left(\frac{1}{2f_L}\left(\frac{\Upsilon_1+h_5}{d_z} - \frac{\Upsilon_2+h_6}{d_y}\right)\right)\right]. \tag{4.49}$$

Then, the following relation for the inputs Υ_1 and Υ_2 is used:

$$\begin{bmatrix}\Upsilon_1\\ \Upsilon_2\end{bmatrix} = \begin{bmatrix}\bar{m}_{55} & \bar{m}_{56}\\ \bar{m}_{65} & \bar{m}_{66}\end{bmatrix}\begin{bmatrix}v_5\\ v_6\end{bmatrix} \tag{4.50}$$

$$\boldsymbol{\Upsilon}_{\theta,\psi} = \boldsymbol{M}_{\theta,\psi}\boldsymbol{v}_{\theta,\psi}. \tag{4.51}$$

Given that $\boldsymbol{M}_{\theta,\psi}$ is nonsingular around the equilibrium point, the following holds:

$$\begin{bmatrix}\ddot{\theta}\\ \ddot{\psi}\end{bmatrix} = \begin{bmatrix}v_5\\ v_6\end{bmatrix}. \tag{4.52}$$

The input v_5 was designed to perform a PD control with feed-forward, while v_6 performs a PID control with feed-forward. This choice was made due to the fact that θ needs only to be stabilized, while ψ should track desired trajectories and reject sustained disturbances. The behaviours of v_5 and v_6 are similar to the ones showed in equations (4.35) (PID controller) and (4.36) (PD controller).

This section assumed that some dynamic coupling are negligible. However, this is not true, since the whole system is coupled. Therefore, when allocating poles for θ and ψ, the inner loops of the cascade system also change the position of their poles. Consequently, it is important to look at the closed loop poles of the linearized system when tuning the controllers for v_5 and v_6 so that the coupling does not provide undesired results.

The same controller parameters designed for the controllers of the previous section were kept, while the parameters for θ and ψ were tuned avoiding unstable poles on the linearized system. The tuning was performed using D-stability algorithms. The controllers' parameters can be seen in Table 4.1.

In order to analyse the whole system's stability, it was used again the *numerical perturbation method*. This procedure was done in the neighborhood of the equilibrium point and found a characteristic equation with four marginally stable poles and no unstable poles. When simulating the system, it was possible to verify that the variables x, y, $\dot{x}$ and $\dot{y}$ had marginally stable behaviours.

Therefore, this IOFL block was able to stabilize both θ and ψ. The control of x and y is accomplished by using the third IOFL block. The stability of γ_1 and γ_2 are also taken into account in the next subsection.

Table 4.1: Controllers parameters for the two first Feedback-Linearizations.

Variable	K_p	K_I	K_d
z	-145.3474	-321.0882	-21.2709
ϕ	-307.6781	-	-35.2992
θ	-52.2404	-	-14.9975
ψ	-210.9180	-524.9711	-26.4364
α_R	-3999.1715	-	-135.9896
α_L	-3999.1715	-	-135.9896

4.3.3 Third-level Feedback Linearization

This section presents the strategy performed to stabilize the outputs x, y, γ_1 and γ_2 around an equilibrium point while actuating on the references ϕ_{Ref} and θ_{Ref}.

Again, a reduced Euler-Lagrange model is used for the system's controller design, assuming that x, y, γ_1 and γ_2 do not affect the other variables, while the other variables do not affect x, y, γ_1 and γ_2 either. Again, the neglected dynamic coupling are then considered as modelling errors on the control design. Consequently, the Euler-Lagrange equations can be written for these variables in the form:

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,9} & m_{1,10} \\ m_{2,1} & m_{2,2} & m_{2,9} & m_{2,10} \\ m_{9,1} & m_{9,2} & m_{9,9} & m_{9,10} \\ m_{10,1} & m_{10,2} & m_{10,9} & m_{10,10} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} c_{1,1} & c_{1,2} & c_{1,9} & c_{1,10} \\ c_{2,1} & c_{2,2} & c_{2,9} & c_{2,10} \\ c_{9,1} & c_{9,2} & c_{9,9} & c_{9,10} \\ c_{10,1} & c_{10,2} & c_{10,9} & c_{10,10} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \\ g_9 \\ g_{10} \end{bmatrix} = \begin{bmatrix} T_x^{\mathcal{I}} \\ T_y^{\mathcal{I}} \\ 0 \\ 0 \end{bmatrix}, \tag{4.53}$$

where $m_{i,j}$ is the element of the inertia matrix located at the i^{th} line and j^{th} column. The same applies to c_{ij} and g_i, which represent terms for the coriolis and centripetal matrix and gravity vector, respectively. The variables $T_x^{\mathcal{I}}$ and $T_y^{\mathcal{I}}$ are, respectively, the translational forces along x and y expressed in the inertial frame:

$$T_x^{\mathcal{I}} = c_\psi c_\theta f_x^{\mathcal{B}} + (c_\psi s_\theta s_\phi - s_\psi c_\phi) f_y^{\mathcal{B}} + (s_\psi s_\phi + c_\psi s_\theta c_\phi) f_z^{\mathcal{B}} \tag{4.54}$$

$$T_y^{\mathcal{I}} = s_\psi c_\theta f_x^{\mathcal{B}} + (s_\psi s_\theta s_\phi + c_\psi c_\phi) f_y^{\mathcal{B}} + (s_\psi s_\theta c_\phi - c_\psi s_\phi) f_z^{\mathcal{B}}, \tag{4.55}$$

where

$$\begin{bmatrix} f_x^{\mathcal{B}} \\ f_y^{\mathcal{B}} \\ f_z^{\mathcal{B}} \end{bmatrix} = \boldsymbol{F}_R^{\mathcal{B}} + \boldsymbol{F}_L^{\mathcal{B}}, \tag{4.56}$$

and $\boldsymbol{F}_R^{\mathcal{B}}$, $\boldsymbol{F}_L^{\mathcal{B}}$ are defined on equations (2.53) and (2.54).

An important observation about this approach is that the actuation on the system

should be done by changing the projection of $f_z^{\mathcal{B}}$ along $x^{\mathcal{I}}$ and $y^{\mathcal{I}}$. In other words, the roll (ϕ) and pitch (θ) angles need to be changed so as to obtain the desired projections of $f_z^{\mathcal{B}}$ along $x^{\mathcal{I}}$ and $y^{\mathcal{I}}$. The projections of $f_x^{\mathcal{B}}$ and $f_y^{\mathcal{B}}$ along the inertial frame are going to be considered as known disturbances which should be compensated by the controller.

For further simplifications when needed, equation (4.53) is also conveniently denoted as:

$$\boldsymbol{M}_{x,y,\gamma_1,\gamma_2}\ddot{\boldsymbol{q}}_{x,y,\gamma_1,\gamma_2} + \boldsymbol{C}_{x,y,\gamma_1,\gamma_2}\dot{\boldsymbol{q}}_{x,y,\gamma_1,\gamma_2} + \boldsymbol{G}_{x,y,\gamma_1,\gamma_2} = \boldsymbol{T}_{x,y,\gamma_1,\gamma_2}. \tag{4.57}$$

By inspecting equations (4.54) and (4.55), it is not trivial to obtain a nonlinear state-space representation affine in the inputs where the inputs are given by ϕ_{Ref} and θ_{Ref}. Therefore, some mathematical manipulation should be made so that IOFL theory can be used. As a proposal, assume that it is possible to obtain a system representation such as follows:

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,9} & m_{1,10} \\ m_{2,1} & m_{2,2} & m_{2,9} & m_{2,10} \\ m_{9,1} & m_{9,2} & m_{9,9} & m_{9,10} \\ m_{10,1} & m_{10,2} & m_{10,9} & m_{10,10} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ h_9 \\ h_{10} \end{bmatrix} = \begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \\ 0 \\ 0 \end{bmatrix}. \tag{4.58}$$

Then it would be possible to use IOFL techniques to linearize this portion of the system. In what follows, two solutions are provided so that the relation in equation (4.58) is obtained.

Solution 1 This first solution is simple, but it requires the variable ψ to be always on the vicinity of $\psi = 0$. Then, it is possible to actuate on s_ϕ so that the UAV moves along y and actuate on s_θ so that the Tilt-rotor moves along x. Thus, defining the following variables:

$$\begin{bmatrix} h_1 \\ h_2 \\ h_9 \\ h_{10} \end{bmatrix} = \boldsymbol{C}_{x,y,\gamma_1,\gamma_2}\dot{\boldsymbol{q}}_{x,y,\gamma_1,\gamma_2} + \boldsymbol{G}_{x,y,\gamma_1,\gamma_2} - \begin{bmatrix} c_\psi c_\theta f_x^{\mathcal{B}} + (c_\psi s_\theta s_\phi - s_\psi c_\phi) f_y^{\mathcal{B}} + s_\psi s_\phi f_z^{\mathcal{B}} \\ s_\psi c_\theta f_x^{\mathcal{B}} + (s_\psi s_\theta s_\phi + c_\psi c_\phi) f_y^{\mathcal{B}} + s_\psi s_\theta c_\phi f_z^{\mathcal{B}} \\ 0 \\ 0 \end{bmatrix}. \tag{4.59}$$

Equation (4.53) can then be rewritten as:

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,9} & m_{1,10} \\ m_{2,1} & m_{2,2} & m_{2,9} & m_{2,10} \\ m_{9,1} & m_{9,2} & m_{9,9} & m_{9,10} \\ m_{10,1} & m_{10,2} & m_{10,9} & m_{10,10} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_9 \\ h_{10} \end{bmatrix} = \begin{bmatrix} (c_\psi c_\phi f_z^{\mathcal{B}}) s_\theta \\ (-c_\psi f_z^{\mathcal{B}}) s_\phi \\ 0 \\ 0 \end{bmatrix}. \tag{4.60}$$

Therefore, by choosing:

$$\theta_{Ref} = \arcsin\left[\sigma\left(\frac{\Upsilon_1+h_1}{c_\psi c_\phi f_z^{\mathcal{B}}}\right)\right], \quad \phi_{Ref} = \arcsin\left[\sigma\left(\frac{\Upsilon_2+h_2}{-c_\psi f_z^{\mathcal{B}}}\right)\right], \tag{4.61}$$

then it is possible to obtain the relation given in (4.58). It should be noted that this solution is undefined when $c_\psi = 0$, $c_\phi = 0$ or $f_z = 0$. Thus, this solution is valid only if $-\pi/2 < \psi < \pi/2$ (this is satisfied if ψ is in the vicinity of $\psi = 0$), $-\pi/2 < \phi < \pi/2$ (this is usually satisfied, given that abrupt manoeuvres are not applied on the aircraft) and if $f_z > 0$ (this is also usually satisfied, given that the aircraft needs to be lifted by f_z).

Solution 2 This solution does not require the variable ψ to be in the vicinity of $\psi = 0$. With this solution, ϕ and θ are adjusted for any given ψ, so that the Tilt-rotor moves along x and y according to the desired trajectory. However, some caution must be taken when using this solution, as discussed afterwards. First, the following is defined:

$$\begin{bmatrix} h_1 \\ h_2 \\ h_9 \\ h_{10} \end{bmatrix} = \boldsymbol{C}_{x,y,\gamma_1,\gamma_2}\dot{\boldsymbol{q}}_{x,y,\gamma_1,\gamma_2} + \boldsymbol{G}_{x,y,\gamma_1,\gamma_2} - \begin{bmatrix} c_\psi c_\theta f_x^{\mathcal{B}} + (c_\psi s_\theta s_\phi - s_\psi c_\phi) f_y^{\mathcal{B}} \\ s_\psi c_\theta f_x^{\mathcal{B}} + (s_\psi s_\theta s_\phi + c_\psi c_\phi) f_y^{\mathcal{B}} \\ 0 \\ 0 \end{bmatrix}. \tag{4.62}$$

Equation (4.53) can then be rewritten as:

$$\begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,9} & m_{1,10} \\ m_{2,1} & m_{2,2} & m_{2,9} & m_{2,10} \\ m_{9,1} & m_{9,2} & m_{9,9} & m_{9,10} \\ m_{10,1} & m_{10,2} & m_{10,9} & m_{10,10} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_9 \\ h_{10} \end{bmatrix} = \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi) f_z^{\mathcal{B}} \\ (s_\psi s_\theta c_\phi - c_\psi s_\phi) f_z^{\mathcal{B}} \\ 0 \\ 0 \end{bmatrix}. \tag{4.63}$$

Therefore, in order to obtain a system representation such as the one given by equation (4.58), then one needs to find ϕ and θ that satisfies the following system of nonlinear equations:

$$\begin{cases} s_\psi s_\phi + c_\psi s_\theta c_\phi = \frac{\Upsilon_1+h_1}{f_z^{\mathcal{B}}} \\ s_\psi s_\theta c_\phi - c_\psi s_\phi = \frac{\Upsilon_2+h_2}{f_z^{\mathcal{B}}} \end{cases} \implies \begin{cases} s_\psi s_\phi + c_\psi s_\theta c_\phi = c_1 \\ s_\psi s_\theta c_\phi - c_\psi s_\phi = c_2 \end{cases}, \tag{4.64}$$

where c_1 and c_2 are obviously defined. The solution of this system of nonlinear equations was obtained using numerical methods. This method finds multiple solutions for the nonlinear system of equations 4.64. Each solution was analysed and the one that fitted the most for $-\pi/2 < \phi_{Ref} < \pi/2$ and $-\pi/2 < \theta_{Ref} < \pi/2$ was chosen. The solution is given by:

$$\phi_{Ref} = \arctan\left[\frac{a^2+1}{b} - \left[c\left(\frac{2a^2+a^4-c_2^2-4c_1a^2+2c_2^2a^2-c_2^2a^4+4c_1c_2a-4c_1c_2a^3+1}{b}\right)\right]\right] \tag{4.65}$$

$$\theta_{Ref} = \arcsin\left[c\left(c_1 + 2c_2a - c_1a^2\right)\right], \tag{4.66}$$

where:

$$a = \arctan\left(\frac{\psi}{2}\right), \quad b = 2c_1a - c_2 + c_2a^2, \quad c = \sqrt{\frac{1}{(b+a^2+1)(-b+a^2+1)}}. \tag{4.67}$$

There are two issues that require caution when using this solution: first, the function σ of equation (4.47) should be used in equation (4.66) so that it does not become undefined. The second, and most troublesome problem, is that $\phi, \theta \in \Re$ but the variable c may assume imaginary values depending on the values within the square root operator. In order to solve the problem when c is imaginary, two solutions are suggested:

1. Given that in this case $Re(c) = 0$, then $c = 0$ can be assumed. Good simulation results were obtained when using this approach. Extensive tests were performed varying the simulated values of ϕ, θ and ψ, but the system did not present any ill behaved response when using this approach.
2. When it happens that $c \notin \Re$, the controller may switch to an alternative way of solving the system of equations (4.64) using some optimization algorithm instead of using the solution provided by equations (4.65).

The second solution imposes the necessity of implementing optimization algorithms on the hardware of the UAV, what may be undesired given the computational cost that are usually expected for these algorithms. Therefore, the first solution was used when simulating the system.

Independently of the specific solution (*Solution 1* or *Solution 2*) used, the system representation in the form given by (4.58) is obtained, and now it is possible to apply input-output feedback linearization using the outputs:

$$\boldsymbol{y}(\boldsymbol{x}) = \begin{bmatrix} h_1(\boldsymbol{x}) \\ h_2(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}. \tag{4.68}$$

The relative degree of both outputs is $r_i = 2$, $i = 1, 2$. With this, it is possible to drive the variables x, $\dot{x}$, y and $\dot{y}$. The dynamics of the variables γ_1, $\dot{\gamma}_1$, γ_2 and $\dot{\gamma}_2$ can then be evaluated by looking at the internal dynamics.

The state-space representation obtained from equation (4.58) is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\gamma}_1 \\ \dot{\gamma}_2 \\ \ddot{x} \\ \ddot{y} \\ \ddot{\gamma}_1 \\ \ddot{\gamma}_2 \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{x}}_{x,y,\gamma_1,\gamma_2} \\ \ddot{\boldsymbol{x}}_{x,y,\gamma_1,\gamma_2} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{x}}_{x,y,\gamma_1,\gamma_2} \\ \boldsymbol{F}_{x,y,\gamma_1,\gamma_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0}_{4\times 2} \\ \boldsymbol{g}_{x,y,\gamma_1,\gamma_2} \end{bmatrix} \begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \end{bmatrix}, \tag{4.69}$$

where:

$$\boldsymbol{H} = \begin{bmatrix} 0 \\ 0 \\ h_9 \\ h_{10} \end{bmatrix}, \quad \boldsymbol{F}_{x,y,\gamma_1,\gamma_2} = -\boldsymbol{M}^{-1}_{x,y,\gamma_1,\gamma_2}\boldsymbol{H}, \quad \boldsymbol{g}_{x,y,\gamma_1,\gamma_2} = \boldsymbol{M}^{-1}_{x,y,\gamma_1,\gamma_2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.70}$$

Representing the matrix $\boldsymbol{M}^{-1}_{x,y,\gamma_1,\gamma_2}$ as:

$$\boldsymbol{M}^{-1}_{x,y,\gamma_1,\gamma_2} = \begin{bmatrix} M^{-1}_{11} & M^{-1}_{12} & M^{-1}_{13} & M^{-1}_{14} \\ M^{-1}_{21} & M^{-1}_{22} & M^{-1}_{23} & M^{-1}_{24} \\ M^{-1}_{31} & M^{-1}_{32} & M^{-1}_{33} & M^{-1}_{34} \\ M^{-1}_{41} & M^{-1}_{42} & M^{-1}_{43} & M^{-1}_{44} \end{bmatrix}, \tag{4.71}$$

then the IOFL matrices $\boldsymbol{\Delta}(\boldsymbol{x})$ and $\boldsymbol{b}$ are given by:

$$\boldsymbol{\Delta}(\boldsymbol{x}) = \begin{bmatrix} M^{-1}_{11} & M^{-1}_{12} \\ M^{-1}_{21} & M^{-1}_{22} \end{bmatrix}, \quad \boldsymbol{b}(\boldsymbol{x}) = \begin{bmatrix} -M^{-1}_{13}h_9 - M^{-1}_{14}h_{10} \\ -M^{-1}_{23}h_9 - M^{-1}_{24}h_{10} \end{bmatrix}. \tag{4.72}$$

In order to analyse the internal dynamics a first simplification is made, which is to substitute into $\boldsymbol{M}_{x,y,\gamma_1,\gamma_2}$ the equilibrium values of the state variables that were controlled in the inner IOFL blocks. That is, ϕ, θ ψ, α_R and α_L are all substituted into $\boldsymbol{M}_{x,y,\gamma_1,\gamma_2}$ by their respective equilibrium values. The following simplified matrix is obtained:

$$\bar{\boldsymbol{M}}_{x,y,\gamma_1,\gamma_2} = \begin{bmatrix} \bar{m}_{1,1} & 0 & 0 & \bar{m}_{1,10} \\ 0 & \bar{m}_{2,2} & \bar{m}_{2,9} & 0 \\ 0 & \bar{m}_{9,2} & \bar{m}_{9,9} & 0 \\ \bar{m}_{10,1} & 0 & 0 & \bar{m}_{10,10} \end{bmatrix} \implies \bar{\boldsymbol{M}}^{-1}_{x,y,\gamma_1,\gamma_2} = \begin{bmatrix} \frac{\bar{m}_{10,10}}{den_1} & 0 & 0 & -\frac{\bar{m}_{1,10}}{den_1} \\ 0 & \frac{\bar{m}_{9,9}}{den_2} & -\frac{\bar{m}_{2,9}}{den_2} & 0 \\ 0 & -\frac{\bar{m}_{2,9}}{den_2} & \frac{\bar{m}_{2,2}}{den_2} & 0 \\ -\frac{\bar{m}_{1,10}}{den_1} & 0 & 0 & \frac{\bar{m}_{1,1}}{den_1} \end{bmatrix}. \tag{4.73}$$

where:

$$den_1 = \bar{m}_{1,1}\bar{m}_{10,10} - \bar{m}_{1,10}^2, \quad den_2 = \bar{m}_{2,2}\bar{m}_{9,9} - \bar{m}_{2,9}^2, \tag{4.74}$$

In this case, matrices $\boldsymbol{F}_{x,y,\gamma_1,\gamma_2}$ and $\boldsymbol{g}_{x,y,\gamma_1,\gamma_2}$ are given by:

$$\boldsymbol{F}_{x,y,\gamma_1,\gamma_2} = \begin{bmatrix} \frac{\bar{m}_{1,10}}{den_1}h_{10} \\ \frac{\bar{m}_{2,9}}{den_2}h_9 \\ -\frac{\bar{m}_{2,2}}{den_2}h_9 \\ -\frac{\bar{m}_{1,1}}{den_1}h_{10} \end{bmatrix}, \quad \boldsymbol{g}_{x,y,\gamma_1,\gamma_2} = \begin{bmatrix} \frac{\bar{m}_{10,10}}{den_1} & 0 \\ 0 & \frac{\bar{m}_{9,9}}{den_2} \\ 0 & -\frac{\bar{m}_{2,9}}{den_2} \\ -\frac{\bar{m}_{1,10}}{den_1} & 0 \end{bmatrix}. \tag{4.75}$$

Therefore, choosing the following variables for the internal dynamics:

$$\vartheta_1 = \gamma_1, \quad \vartheta_2 = \gamma_2, \quad \vartheta_3 = \frac{\dot{y}}{\bar{m}_{9,9}} + \frac{\dot{\gamma}_1}{\bar{m}_{2,9}}, \quad \vartheta_4 = \frac{\dot{x}}{\bar{m}_{10,10}} + \frac{\dot{\gamma}_2}{\bar{m}_{1,10}}, \tag{4.76}$$

the diffeomorphism is completed and it satisfies the constraint given by equation (4.15), which demands that $\mathcal{L}_{\boldsymbol{g}}\vartheta_k(\boldsymbol{x}) = 0,\ \forall\ 1 < k < 4$, given $\boldsymbol{g} = \begin{bmatrix} \boldsymbol{0}_{4\times 2} \\ \boldsymbol{g}_{x,y,\gamma_1,\gamma_2} \end{bmatrix}$. Therefore, the internal dynamics can be written as:

$$\boldsymbol{f}_\vartheta = \begin{bmatrix} \dot{\vartheta}_1 \\ \dot{\vartheta}_2 \\ \dot{\vartheta}_3 \\ \dot{\vartheta}_4 \end{bmatrix} \implies \begin{cases} \dot{\vartheta}_1 = \mathcal{L}_{\boldsymbol{f}}\vartheta_1 \\ \dot{\vartheta}_2 = \mathcal{L}_{\boldsymbol{f}}\vartheta_2 \\ \dot{\vartheta}_3 = \mathcal{L}_{\boldsymbol{f}}\vartheta_3 \\ \dot{\vartheta}_4 = \mathcal{L}_{\boldsymbol{f}}\vartheta_4 \end{cases}, \tag{4.77}$$

where $\boldsymbol{f} = \begin{bmatrix} \dot{\boldsymbol{x}}_{x,y,\gamma_1,\gamma_2} \\ \boldsymbol{F}_{x,y,\gamma_1,\gamma_2} \end{bmatrix}$. By linearizing the internal dynamics in the neighborhood of the equilibrium point, the following state matrix $\boldsymbol{A}_\vartheta$ is obtained:

$$\boldsymbol{A}_\vartheta = \frac{\partial \boldsymbol{f}_\vartheta}{\partial \boldsymbol{\vartheta}}\Big|_{\boldsymbol{\vartheta}=\boldsymbol{\vartheta}_{Eq}} = \begin{bmatrix} 0 & 0 & 0.025 & 0 \\ 0 & 0 & 0 & -0.025 \\ -785 & 0 & -80\mu_{\gamma_1} & 0 \\ 0 & 785 & 0 & -80\mu_{\gamma_2} \end{bmatrix}, \tag{4.78}$$

where μ_{γ_1} and μ_{γ_2} are drag coefficients as described on section 2.4.5. These coefficients were left unsubstituted on equation (4.78) so that it is possible to see that if they are neglected, i.e. $\mu_{\gamma_1} = \mu_{\gamma_2} = 0$ N.m/(rad/s), then the eigenvalues of $\boldsymbol{A}_\vartheta$ would be $\lambda_\vartheta = 0 \pm 4.43j$. On the other hand, considering $\mu_{\gamma_1} = \mu_{\gamma_2} = 0.005$ N.m/(rad/s), the eigenvalues of γ_1 and γ_2 are $\lambda_\vartheta = -0.2 \pm 4.43j$. This makes sense, since a pendulum may swing forever in a frictionless environment, while it eventually stops swinging in the presence of

friction.

Therefore, it is possible to conclude that, in the presence of friction, the internal dynamics for $\boldsymbol{\gamma}$ is stable in the vicinity of the equilibrium point regardless of the designed controller. It is then possible to design the controllers for this IOFL so that variables x and y are able to perform path tracking, rejecting sustained disturbances. This is accomplished by using PID controllers with feed-forward:

$$v_x = \ddot{x}_{Ref} + K_{d_x}\dot{e}_x + K_{p_x}e_x + K_{I_x}\int e_x \,\mathrm{d}t \tag{4.79}$$

$$v_y = \ddot{y}_{Ref} + K_{d_y}\dot{e}_y + K_{p_x}e_y + K_{I_y}\int e_y \,\mathrm{d}t. \tag{4.80}$$

It was assumed at the beginning of this subsection that the dynamics of x, y, γ_1 and γ_2 do not interact with the remaining variables' dynamics neither on the other way around. However, this is not true, since the whole system is coupled. Therefore, when allocating poles for x and y, the inner loops of the cascade system also change the position of their poles.

Therefore, the controller was designed such that all closed-loop poles are stable on the linearized system. Again, the controllers' parameters were tuned using D-stability algorithms and they are presented in table 4.2.

In order to analyse the whole system's stability, it was used again the *numerical perturbation method*. This procedure was done in the neighborhood of the equilibrium point and found the following closed-loop characteristic equation:

$$\begin{aligned} det(\boldsymbol{A} - \lambda \boldsymbol{I}) =& (s + 96.34)(s + 25.528)(s + 30.01 + 105.16j)(s + 30.01 - 105.16j) \\ & (s + 19.87)(s + 9.69)(s + 11.46 + 14.33j)(s + 11.46 - 14.33j) \\ & (s + 6.40)(s + 5.17)(s + 4.75 + 0.99j)(s + 4.75 - 0.99j) \\ & (s + 2.92 + 0.86j)(s + 2.92 - 0.86j)(s + 1.29 + 3.73j)(s + 1.29 - 3.73j) \\ & (s + 0.69 + 0.16j)(s + 0.69 - 0.16j)(s + 0.57 + 0.23j)(s + 0.57 - 0.23j) \\ & (s + 0.33 + 3.93j)(s + 0.33 - 3.93j)(s + 0.23 + 4.40j)(s + 0.23 - 4.40j). \end{aligned} \tag{4.81}$$

Given that all the poles on the characteristic equation are on the left complex half-plane, then it is possible to say that this system is stable on the vicinity of the equilibrium point. However, it is not possible to say how far from the equilibrium point this system becomes unstable. It is shown on section 4.4 that this control strategy is valid in a wider region than the one obtained by relying on the linear control strategies presented in Chapter 3.

Table 4.2: Controllers parameters for the whole cascade system.

Variable	K_p	K_I	K_d
x	-2.8353	-0.8820	-2.9561
y	-2.8353	-0.8820	-2.9561
z	-145.3474	-321.0882	-21.2709
ϕ	-307.6781	-	-35.2992
θ	-119.6486	-	-21.9717
ψ	-69.4846	-101.4744	-15.0122
α_R	-3310.6447	-	-125.4249
α_L	-3310.6447	-	-125.4249

4.3.3.1 Performance Improvement of the Load's Swing

The previous control solution accomplishes the task of path tracking for the aircraft with suspended load, but no improvements are made to reduce the load's swing. If it is desired to reduce the load's swing when following a desired path, then a different control law must be designed. In what follows, this problem is solved by adapting the solution of Lee et al. (2013), who used a feedback linearization approach for controlling an overhead crane. They considered that the crane could change the distance l between the trolley and the suspended load, but this degree of freedom will not be considered in this work.

In order to improve load stabilization, first the following change of variables are applied:

$$\boldsymbol{d}^{\mathcal{I}}_{\mathcal{BC}_4} = \begin{bmatrix} d^{\mathcal{I}}_x \\ d^{\mathcal{I}}_y \\ d^{\mathcal{I}}_z \end{bmatrix} = \boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}}\, \mathbf{d}^{\mathcal{B}}_4. \tag{4.82}$$

The vector $\boldsymbol{d}^{\mathcal{I}}_{\mathcal{BC}_4}$ represents the distance between the aircraft's base and the load expressed in the inertial frame. In order to reduce the load's swing, the variables $d^{\mathcal{I}}_x$ and $d^{\mathcal{I}}_y$ are regulated. The use of variables expressed in the inertial frame is done because easy relations can be employed to stabilize the load. For instance, $d^{\mathcal{I}}_x$ can be regulated by simply actuating on $\ddot{x}$; the same applies to $d^{\mathcal{I}}_y$.

Now, the reference for the load is $\begin{bmatrix} \dot{d}^{\mathcal{I}}_x \\ \dot{d}^{\mathcal{I}}_y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. That is, the objective is to avoid load swing, regardless of its position with respect to the aircraft. In order to accomplish this, equations (4.79) and (4.80) are adapted so as to add a PI controller to regulate the load's speed with respect to the Tilt-rotor (Lee et al., 2013):

$$v_x = \ddot{x}_{Ref} + K_{d_x}\dot{e}_x + K_{p_x}e_x + K_{I_x}\int e_x\,\mathrm{d}t + K_{p_{d_x}}\dot{d}^{\mathcal{I}}_x + K_{I_{d_x}}\int \dot{d}^{\mathcal{I}}_x\,\mathrm{d}t \tag{4.83}$$

$$v_y = \ddot{y}_{Ref} + K_{d_y}\dot{e}_y + K_{p_x}e_y + K_{I_y}\int e_y\,\mathrm{d}t + K_{p_{d_y}}\dot{d}^{\mathcal{I}}_y + K_{I_{d_y}}\int \dot{d}^{\mathcal{I}}_y\,\mathrm{d}t. \tag{4.84}$$

With this approach, it is possible to reduce the load's swing. In contrast, the path-

Table 4.3: Controllers parameters for load control.

Variable	K_p	K_I	K_d
x	-4.5271	-1.8246	-3.7051
y	-4.5271	-1.8246	-3.7051
$\dot{d}_x^{\mathcal{I}}$	-2.4808	-1.5201	-
$\dot{d}_y^{\mathcal{I}}$	-2.4808	-1.5201	-

tracking performance deteriorates when the system tries to avoid load's motion. The parameters for the controllers of x, y, $\dot{d}_x^{\mathcal{I}}$ and $\dot{d}_y^{\mathcal{I}}$ were tuned using D-stability algorithms and are shown in Table 4.3. The other parameters remain the same as shown in Table 4.2.

4.3.4 Inverted Pendulum Control Strategy

In this section, some modifications to the previous design are suggested so that the Tilt-rotor UAV is able to stabilize an inverted pendulum. Given that the model obtained in the last section considered that the load is attached to the aircraft by a rigid rod, then the same model can be used to express the dynamics of an inverted pendulum. In order to do that, the initial values of $\boldsymbol{\gamma}$ are $\gamma_1(0) = \pi\, rad$ and $\gamma_2(0) = 0\, rad$.

The same linearization performed at the previous section is used. The modification resides on equations (4.83) and (4.84). Now the controller tries to control only $d_x^{\mathcal{I}}$ and $d_y^{\mathcal{I}}$ by using a PID controller to regulate it:

$$v_{d_x} = K_{d_{d_x}} \dot{e}_{d_x} + K_{p_{d_x}} e_{d_x} + K_{I_{d_x}} \int e_{d_x}\, \mathrm{d}t \tag{4.85}$$

$$v_{d_y} = K_{d_{d_y}} \dot{e}_{d_y} + K_{p_{d_y}} e_{d_y} + K_{I_{d_y}} \int e_{d_y}\, \mathrm{d}t. \tag{4.86}$$

where $e_{d_x} = d_x^{\mathcal{I}} - d_{x_{Ref}}$ and $e_{d_y} = d_y^{\mathcal{I}} - d_{y_{Ref}}$. Consequently, it is then possible to regulate the inverted pendulum above the aircraft. However, the position of the aircraft is not regulated. This can be solved by looking at the decomposition of the gravitational force over the pendulum as shown in figure 4.2 (this figure shows only decompositions on $z-y$ plane, but the same applies to $z-x$). It is possible to see that when $d_y^{\mathcal{I}}$ is non-zero, then a component of the gravitational force W is decomposed into a tractive force between the load and the rod, which, in turn, decomposes into F_{d_y}, which accelerates the load on the direction of $d_y^{\mathcal{I}}$. The value of this component is given by:

$$F_{d_y} = W s_\alpha c_\alpha = m_4 g_z \frac{d_y^{\mathcal{I}}}{l} \frac{d_z^{\mathcal{I}}}{l} = \frac{m_4 g_z d_y^{\mathcal{I}} d_z^{\mathcal{I}}}{l^2}. \tag{4.87}$$

The acceleration along $d_y^{\mathcal{I}}$ is given by:

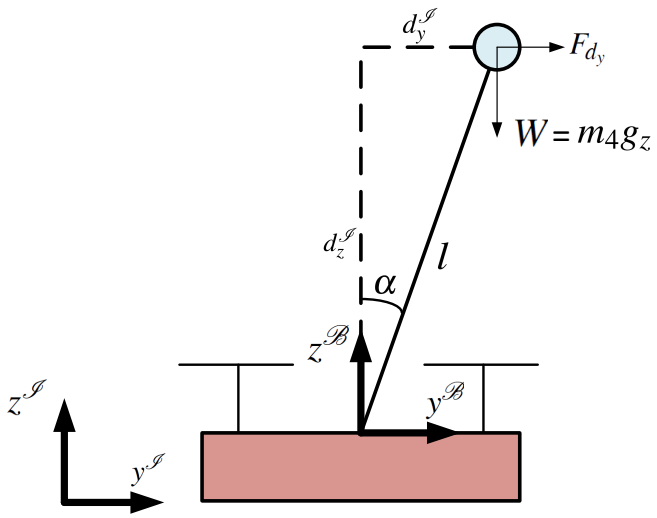

Figure 4.2: Inverted Pendulum Projections.

$$\ddot{d}_y^{\mathcal{I}} = \frac{g_z d_y^{\mathcal{I}} d_z^{\mathcal{I}}}{l^2}. \tag{4.88}$$

If, however, the acceleration $\ddot{y} = \ddot{d}_y^{\mathcal{I}}$ is applied on the aircraft, then the load remains balanced with no variation in $d_y^{\mathcal{I}}$. This means that by choosing set points $d_{y_{Ref}}$, the aircraft tries to balance it around this reference. Consequently, the aircraft accelerates in this direction. Therefore, in order to obtain an acceleration a_y for the aircraft along direction y, then proper $d_{y_{Ref}}$ should be chosen so that it goes in the desired direction. This reference is given by:

$$d_{y_{Ref}} = \frac{a_y l^2}{g_z d_z}. \tag{4.89}$$

Thus, in order to regulate the aircraft in the directions x and y, their desired accelerations can be obtained using a proportional controller:

$$a_x = K_{p_x} e_x, \quad a_y = K_{p_y} e_y. \tag{4.90}$$

In order to make this strategy work in simulations, it was necessary to increase the length of the rigid rod. This had to be done due to the fact that inverted pendulums with larger l have larger margin of stability, given that their frequency of oscillation decrease. Therefore, the simulation results for the inverted pendulum that are shown on section 4.4.3 takes into account $l = 2\,m$. Table 4.4 shows the controller parameters used for the control of the inverted pendulum.

Table 4.4: Controllers parameters for the inverted pendulum.

Variable	K_p	K_I	K_d
x	-0.0315	-	-
y	-0.0315	-	-
z	-145.3474	-321.0882	-21.2709
ϕ	-307.6781	-	-35.2992
θ	-119.6486	-	-21.9717
ψ	-69.4846	-101.474	-15.0122
α_R	-3310.6447	-	-125.42496
α_L	-3310.6447	-	-125.42496
$d_x^{\mathcal{I}}$	-15.0041	-11.2467	-6.6966
$d_y^{\mathcal{I}}$	-15.0041	-11.2467	-6.6966

4.4 Simulation Results

This section shows simulation results carried out to analyse the performance of the nonlinear control strategies designed for the Tilt-rotor UAV with suspended load. First, a comparison is made regarding the nonlinear controller for path tracking (NLPT, given by equations (4.79) and (4.80)) with respect to the nonlinear controller that reduces the load's swing (NLLS, given by equation (4.83) and (4.84)). Then, the NLLS controller is compared to the linear $\mathcal{H}_\infty$ controller designed for the linearized system (presented at section 3.3.4). At last, simulation results are shown for the strategy designed for the inverted pendulum.

4.4.1 Comparison between NLPT and NLLS

A simulation was performed using both NLPT and NLLS strategies to perform a squared trajectory, where the aircraft should displace $2.5m$ in 5 seconds for each of the square's sides. There were no disturbance inputs, the set point for height z was constant and ψ was always oriented in the direction of the trajectory. Figure 4.3 shows the results of the trajectories performed by observing from the top. The aircraft started on the point $x = 0$, $y = 0$ and its initial values for γ_1 and γ_2 were both $\pi/10$. The NLLS reduced the load's swing, but its path tracking error for x and y increased, as it can be seen in Figure 4.4. The time evolution of the remaining generalized coordinates are shown in Figure 4.5; special attention should be given to the dynamics of γ_1 and γ_2, which shows that their motion presented reduced swing when using the NLLS controller. Figure 4.6 shows the system's control inputs.

Table 4.5 shows the mean square error of the trajectory of the aircraft in relation with its reference. The NLPT controller presented 62% of the mean square error on the direction x with respect to the NLLS controller and 71% in the direction y. The tracking errors for z and ψ were similar on both controllers. Even though NLPT performed better path tracking than NLLS, one should be careful when using it, since too much swing of the load, specially for heavy loads, may destabilize the aircraft.

Table 4.6 compares the IAVU indexes for inputs on NLPT and NLLS. Given that

Table 4.5: Mean-square-error comparison between NLPT and NLLS.

	$NLPT$	$NLLS$	$\frac{MSE_{NLPT}}{MSE_{NLLS}}$
MSE_x	$5,18 \cdot 10^{-3}$	$8,32 \cdot 10^{-3}$	$0,6230$
MSE_y	$5,55 \cdot 10^{-3}$	$7,76 \cdot 10^{-3}$	$0,7152$
MSE_z	$2,83 \cdot 10^{-9}$	$2,70 \cdot 10^{-9}$	$1,0460$
MSE_ψ	$3,95 \cdot 10^{-2}$	$3,94 \cdot 10^{-2}$	$1,0013$

both controllers use similar control laws, there are only some slight variations on this index from one approach to the other.

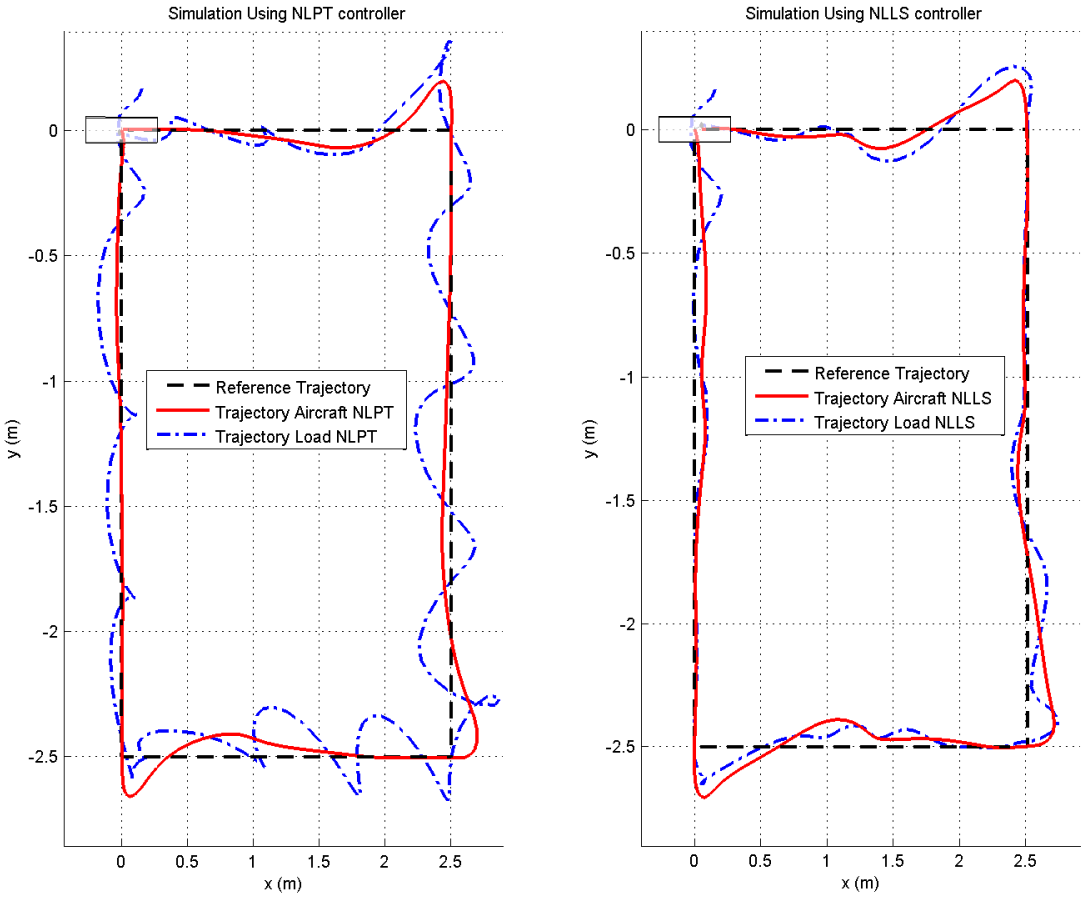

Figure 4.3: Path tracking of the aircraft for NLPT and NLLS.

4.4.2 Comparison between NLLS and Linear $\mathcal{H}_\infty$

Now, figures 4.7-4.10 compare the results obtained using the NLLS and $\mathcal{H}_\infty$ controllers. The simulation shown in these figures were performed with the same configuration as done for the linear simulations: the UAV tracks a predefined trajectory while some disturbance forces ($Fx_{ext}(t)$, $Fy_{ext}(t)$ and $Fz_{ext}(t)$) affect the vehicle on its geometric center (same disturbances as shown in Figure 3.4). Moreover, the simulation considered that the model's masses m_i and inertia tensors I_i, for $i = 1, 2, 3, 4$, had all uncertainties ranging from -30% to 30% of their nominal values. Besides, $\psi_{Ref}(t) = 0$ for all $0 < t < 80$.

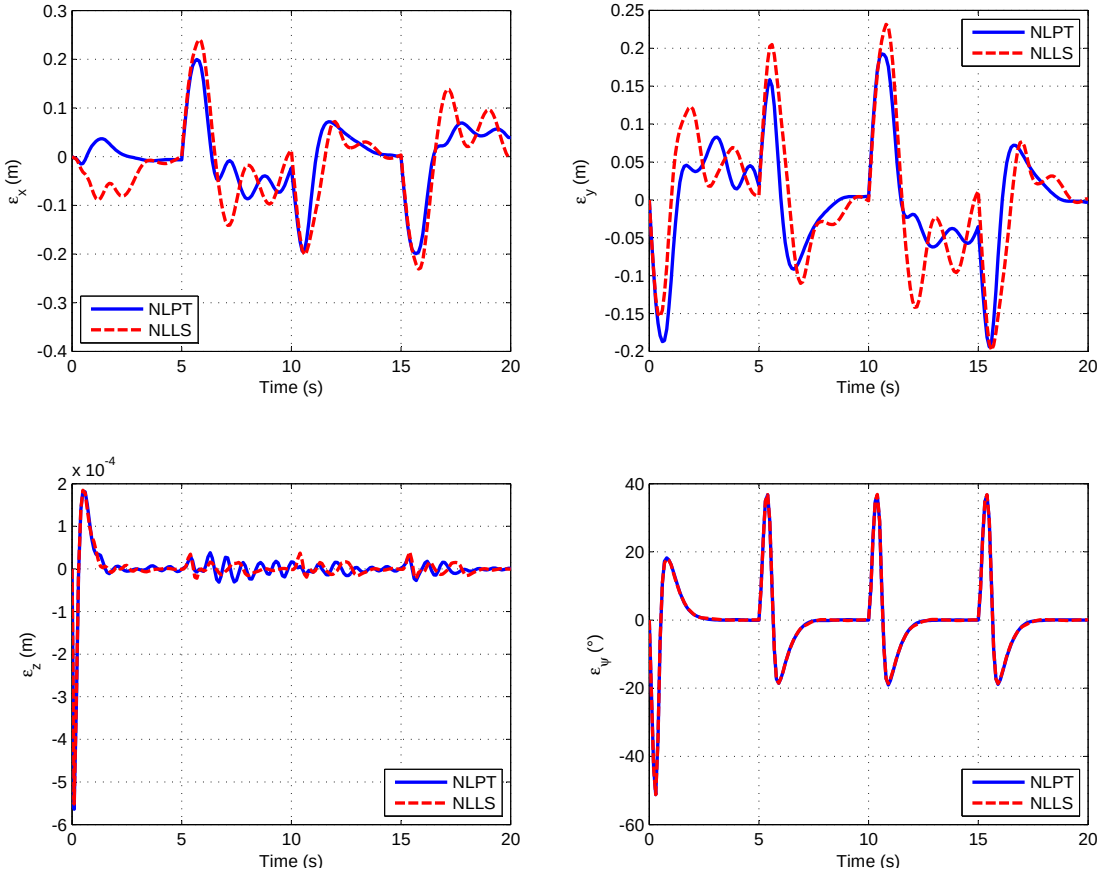

Figure 4.4: Tracking error on NLPT and NLLS.

Table 4.6: IAVU Index comparison between NLPT and NLLS.

	$NLPT$	$NLLS$
$IAVUf_R$	$105,1756$	$95,1902$
$IAVUf_L$	$101,4633$	$91,3149$
$IAVU\tau_{\alpha_R}$	$4,9287$	$4,9930$
$IAVU\tau_{\alpha_L}$	$5,1383$	$5,2141$

Figures 4.7 and 4.8 show that the $\mathcal{H}_\infty$ controller provided better path tracking in the presence of disturbances. Furthermore, Figure 4.9 shows that the system had a response more oscillatory when using NLLS than when using $\mathcal{H}_\infty$. Figure 4.10 shows the system's control inputs. It is possible to verify that the inputs for the $\mathcal{H}_\infty$ controller are more abrupt than for the NLLS case. This can be verified by looking at table 4.7, which shows that the IAVU index was much larger for the $\mathcal{H}_\infty$ controller than for the NLLS.

Table 4.8 shows the mean square error of the trajectory of the aircraft in relation with its reference. The $\mathcal{H}_\infty$ controller presented 5.74% of the mean square error on the direction x with respect to the NLLS controller and 3.57% in the direction y. The NLLS controller presented 39.7% of the mean square error on the direction z with respect to the $\mathcal{H}_\infty$ controller and 5.88% in the direction of ψ.

According to Slotine and Li (1991), feedback linearization techniques does not guar-

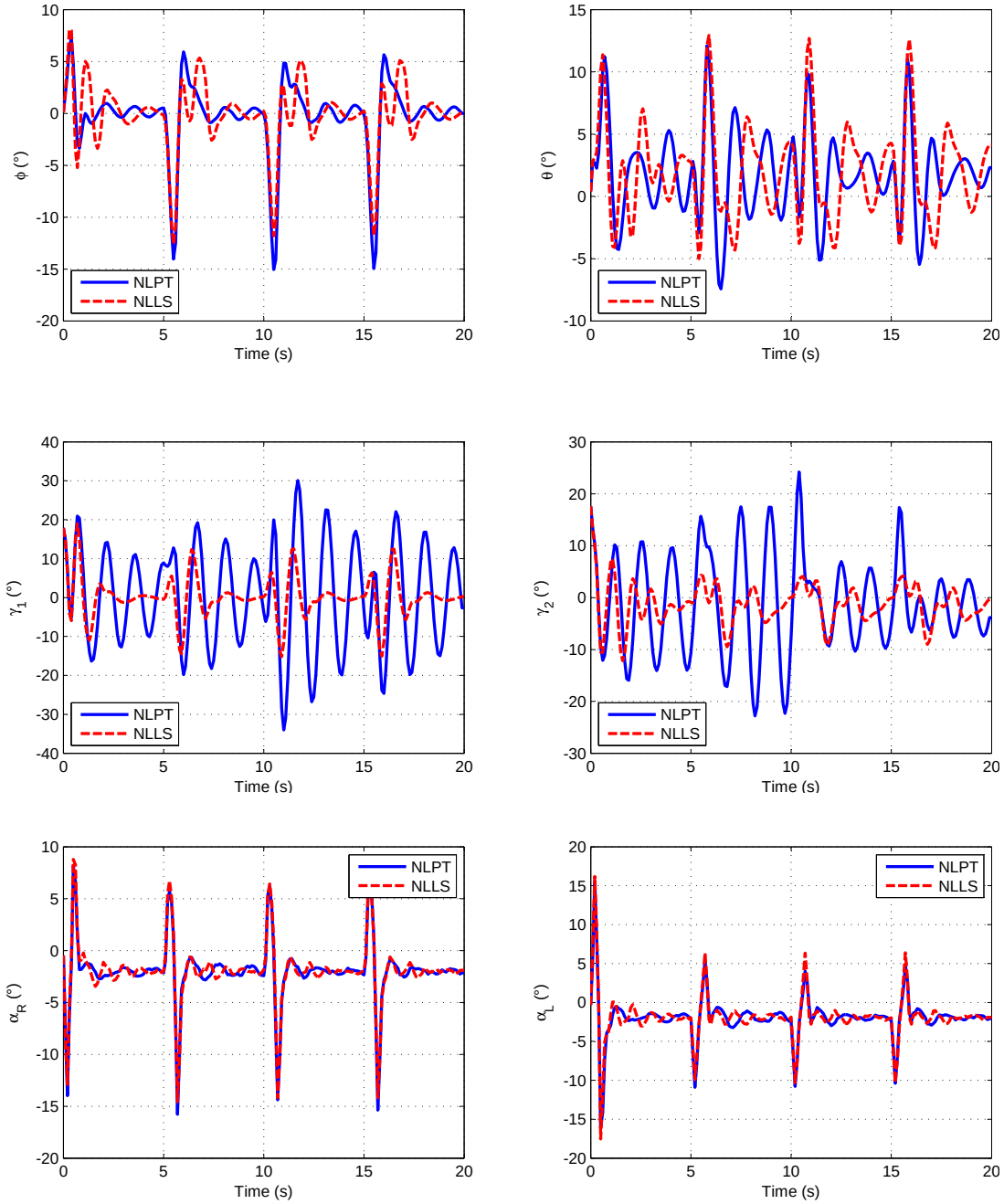

Figure 4.5: Body and Load angles on NLPT and NLLS.

antee robustness in the face of parameter uncertainties or disturbances. This, along with all the design simplifications assumptions, might explain the worse results for path tracking of x and y using the nonlinear controller. However, it might be possible that, with some careful tuning, the NLLS approach may present similar or even better results when compared to the $\mathcal{H}_\infty$ controller. Besides, even though the nonlinear controller does not provide better results than the $\mathcal{H}_\infty$ controller when tracking a defined trajectory in the presence of disturbances, the nonlinear approach is advantageous when the system goes far from the equilibrium point. In section 3.4 it was described some situations in which nonlinear controllers might provide better results than linear ones. It then follows that

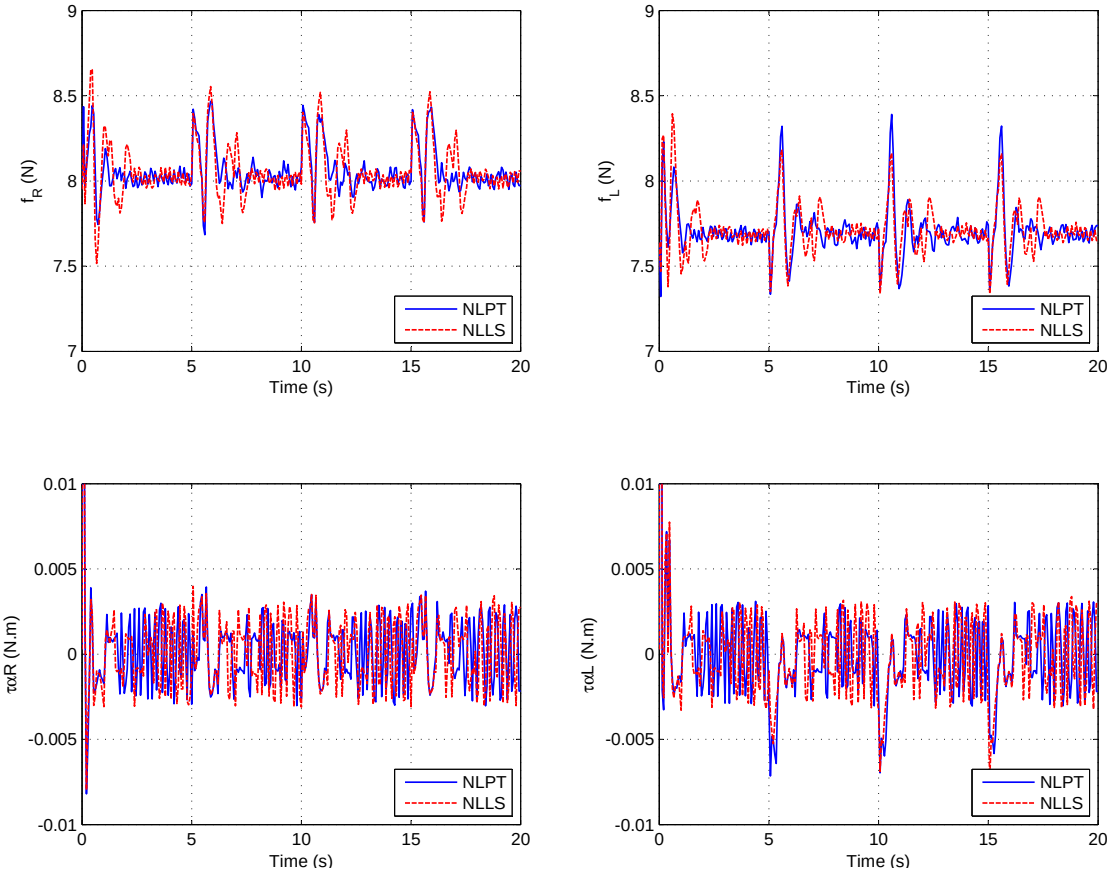

Figure 4.6: Inputs of the system using NLPT and NLLS.

simulation results demonstrate that the following is true:

- The nonlinear controller is capable of making the aircraft follow non-smooth trajectories or trajectories that demands higher accelerations of the system. Figure 4.11 shows a simulation where the aircraft should have followed the same path as the previous simulation, but in sixty seconds (instead of in eighty seconds). It shows that the linear controller destabilized right away at the beginning of the simulation, while the nonlinear controller could stabilize the aircraft through all the path;

- The nonlinear controller is able to deal with harsher disturbances than the linear controller. This can be seen in Figure 4.12, which shows the same simulation of Figure 4.7 but with harsher disturbances (four times higher disturbances on x and y, and two times higher disturbances on z). It can be seen that the linear controller could not stabilize the aircraft right when the first disturbance step was applied, while the nonlinear controller could still track the desired reference. Even though the $\mathcal{H}_\infty$ controller is designed for disturbance rejection, this controller wasn't able to keep the aircraft stabilized because the disturbance displaced the aircraft too much from its reference signal;

- The nonlinear controller is capable of stabilizing the aircraft even with the system's state variables start far from the equilibrium point. Figures 4.13 and 4.14 show the

Table 4.7: IAVU Index comparison between NLLS and $\mathcal{H}_\infty$.

	$NLLS$	H_∞
$IAVUf_R$	$424,2282$	$6,4251 \cdot 10^3$
$IAVUf_L$	$388,9417$	$4,4195 \cdot 10^3$
$IAVU\tau_{\alpha_R}$	$17,3053$	$102,2888$
$IAVU\tau_{\alpha_L}$	$20,8913$	$209,8082$

response of the system using the controllers NLLS and $\mathcal{H}_\infty$ where the initial generalized coordinates are $\boldsymbol{q}(0) = \begin{bmatrix}0.5 & 0.5 & 0.5 & \pi/4 & \pi/4 & \pi/4 & \pi/4 & \pi/4 & -\pi/4 & -\pi/4\end{bmatrix}$ and the set point is $x_{Ref}(t) = y_{Ref}(t) = z_{Ref}(t) = \psi_{Ref}(t) = 0$. It is possible to see that the nonlinear controller is able to stabilize the system with these conditions, while the linear one cannot.

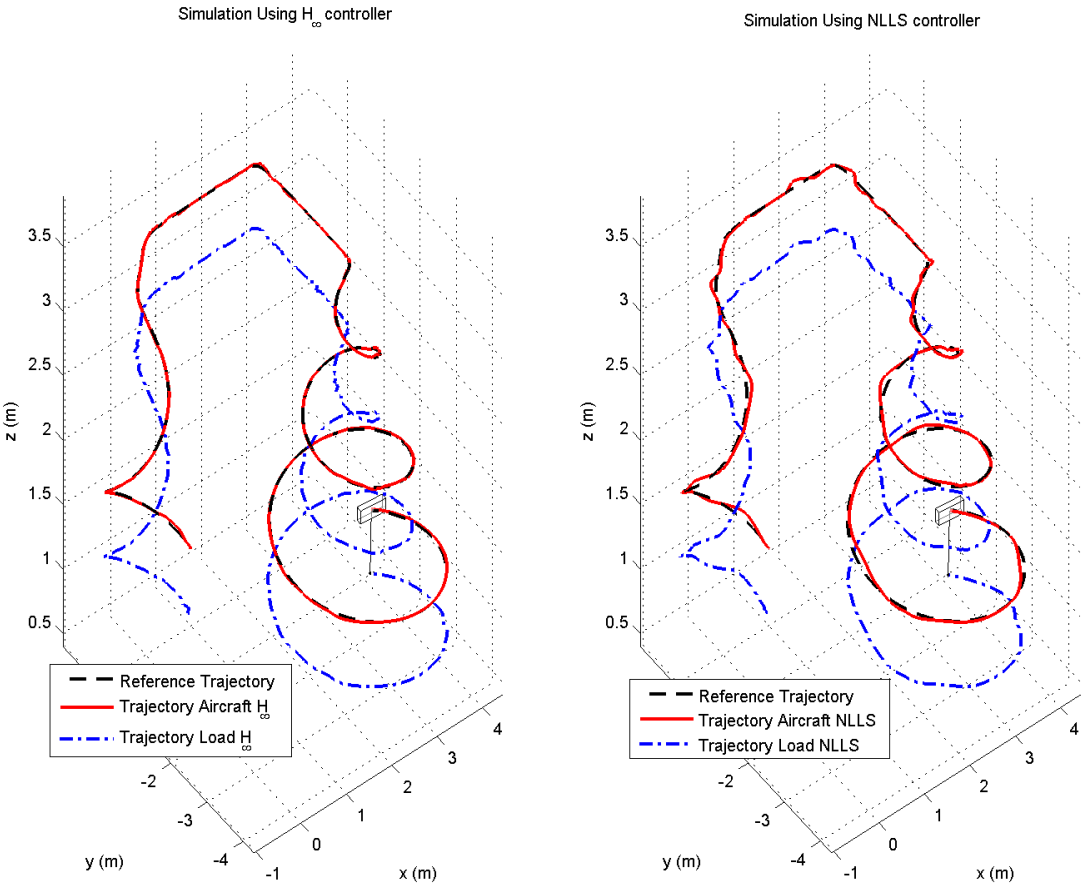

Figure 4.7: Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$.

4.4.3 Inverted Pendulum

Figure 4.15 shows the path followed by the system while tracking a desired reference stabilizing an inverted pendulum. The aircraft's reference demanded tracking a straight line where $\dot{x}_{ref} = \dot{y}_{ref} = \dot{z}_{ref} = 5\,cm/s$ and $\psi_{ref} = 0$. Figures 4.16-4.17 shows the

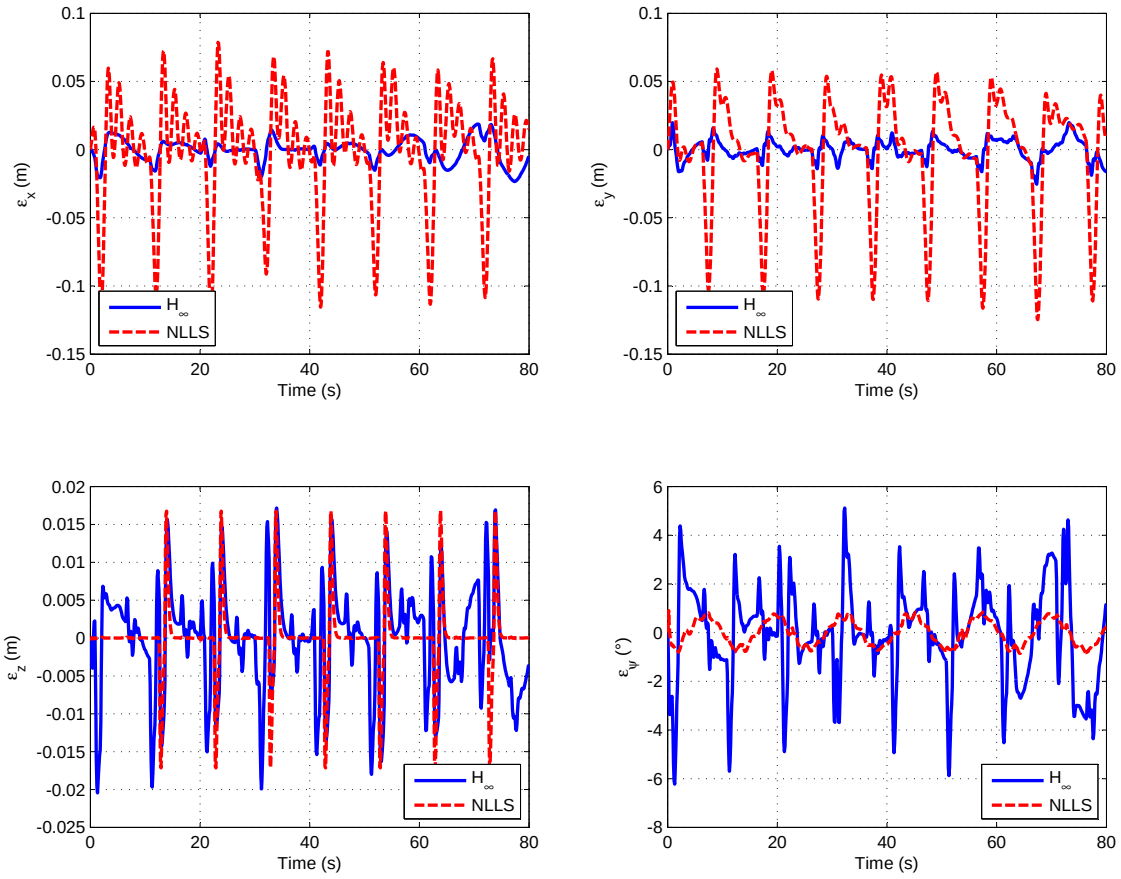

Figure 4.8: Tracking error on NLLS and $\mathcal{H}_\infty$.

Table 4.8: Mean-square-error comparison between NLLS and $\mathcal{H}_\infty$.

	$NLLS$	H_∞	$\frac{MSE_{H_\infty}}{MSE_{NLLS}}$
MSE_x	$1,54.10^{-3}$	$8,88.10^{-5}$	0.0574
MSE_y	$1,56.10^{-3}$	$5.56.10^{-5}$	0.0357
MSE_z	$2,00.10^{-5}$	$5,03.10^{-5}$	2.5188
MSE_ψ	$7,67.10^{-5}$	$1,31.10^{-3}$	17.124

generalized coordinates of the aircraft. It is possible to observe that even though the response was oscillatory, γ_1 remained in the vicinity of $\gamma_1 = \pi\, rad$ and the position of the aircraft also stood near its references. It is possible to in Figure 4.18 that the control effort was quite high, as verified in Table 4.9. The IAVU index presented to be very high for τ_{α_R} and τ_{α_L}, which could be a problem when using this approach on a real experiment.

Even though it was shown that the Tilt-rotor stabilized an inverted pendulum, its results were not as good as solutions using Quad-rotor UAVs (Hehn and D'Andrea, 2011). This happens because the Tilt-rotor is not able to actuate fast enough on its pitch angle as Quad-rotors can. Consequently, for $\psi = 0$, the Tilt-rotor is able to quickly stabilize d_y (actuating on roll angle), but not so fast when stabilizing d_x (actuating on pitch angle). This can be verified in figure 4.17, where γ_1 deviated $\pm 1°$ from its equilibrium, while γ_2 deviated around $\pm 5°$.

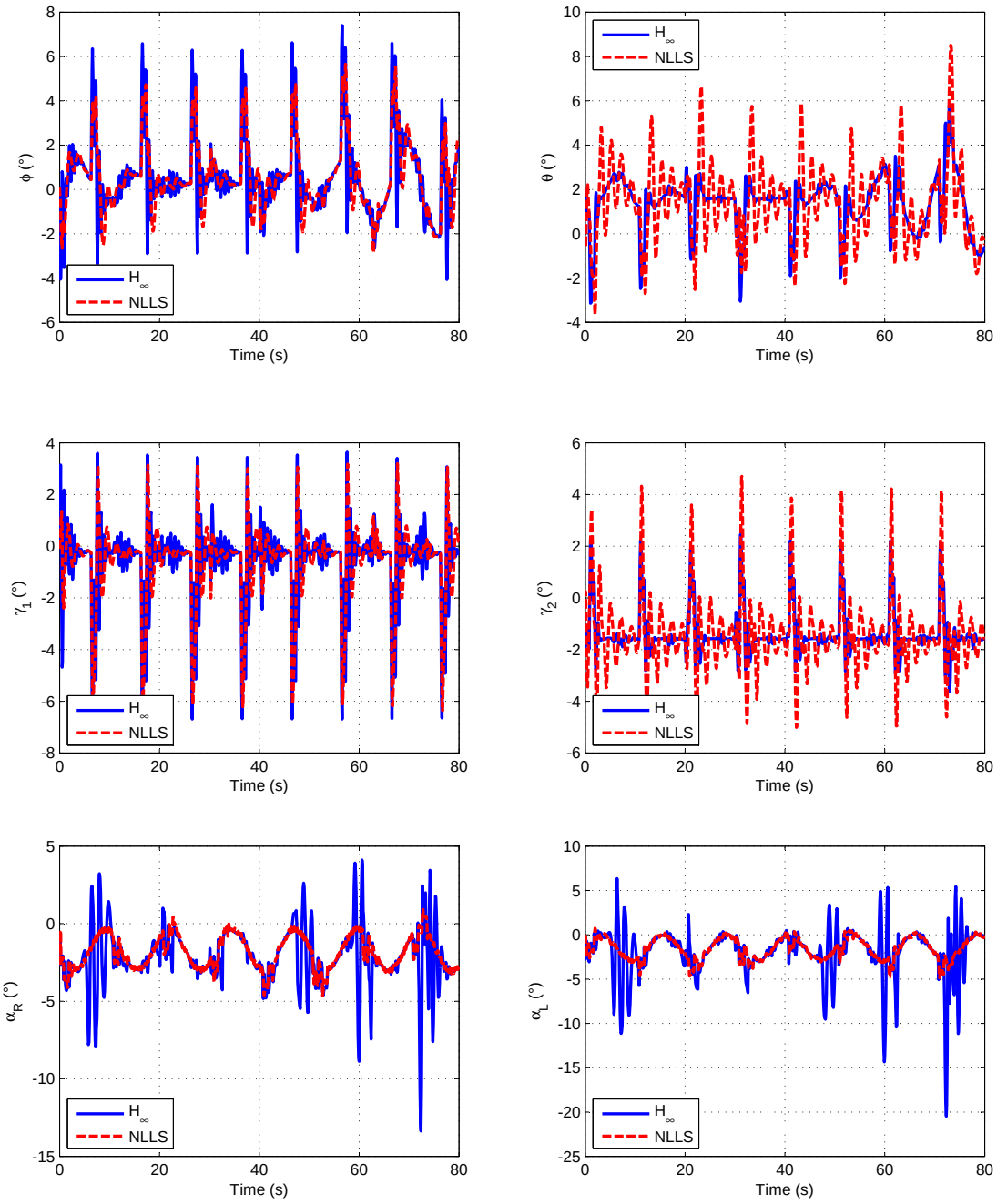

Figure 4.9: Body and Load angles on NLLS and $\mathcal{H}_\infty$.

4.5 Conclusions

This chapter provided solutions for the problem of nonlinear closed-loop control of the Tilt-rotor UAV with suspended load. The proposed strategies were able to stabilize the whole system while tracking a desired reference in the presence of exogenous disturbances, parametric uncertainties and unmodelled dynamics.

Two nonlinear controllers were designed to accomplish the task of path tracking: NLPT (Nonlinear for path tracking) and NLLS (Nonlinear for reduced load swing). NLLS

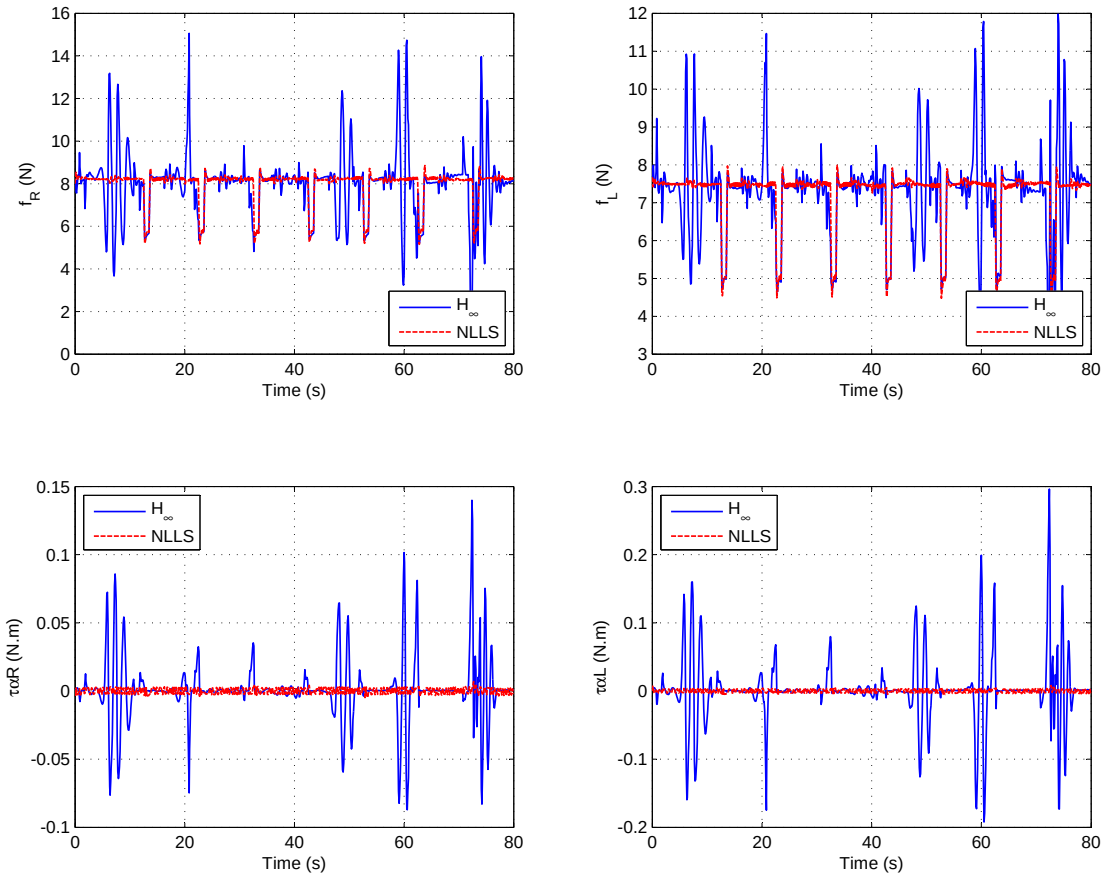

Figure 4.10: Inputs of the system using NLLS and $\mathcal{H}_\infty$.

Table 4.9: IAVU Index for the Inverted Pendulum.

	f_R	f_R	τ_{α_R}	τ_{α_L}
$IAVU$	$1,0065 \cdot 10^3$	$954,0050$	$1,2230 \cdot 10^3$	$1,2217 \cdot 10^3$

reduce the load's swing with respect to NLPT at the cost of worsening its tracking error. However, even though NLPT performed better path tracking than NLLS, one should be careful when using it, since too much swing of the load, specially for heavy loads, may destabilize the aircraft.

When compared to the $\mathcal{H}_\infty$ controller (introduced in chapter 3), the NLLS was worse both on path tracking and disturbance rejection for a pre-defined trajectory. However, NLLS is able to maintain the system stable when it deviates from the equilibrium point. NLLS was shown to be better than linear controllers when there are complex paths, harsher disturbances or when the aircraft starts far from its reference. Besides, the control effort obtained from NLLS (IAVU index) is around ten times smaller when compared to the $\mathcal{H}_\infty$ controller.

Apart from the previous controllers, one last nonlinear controller was designed enabling the Tilt-rotor to stabilize an inverted pendulum. Conceptually, it was possible to attain stabilization of the inverted pendulum, but one cannot expect results as good as the ones provided by Quad-rotor UAVs due to their additional two rotors.

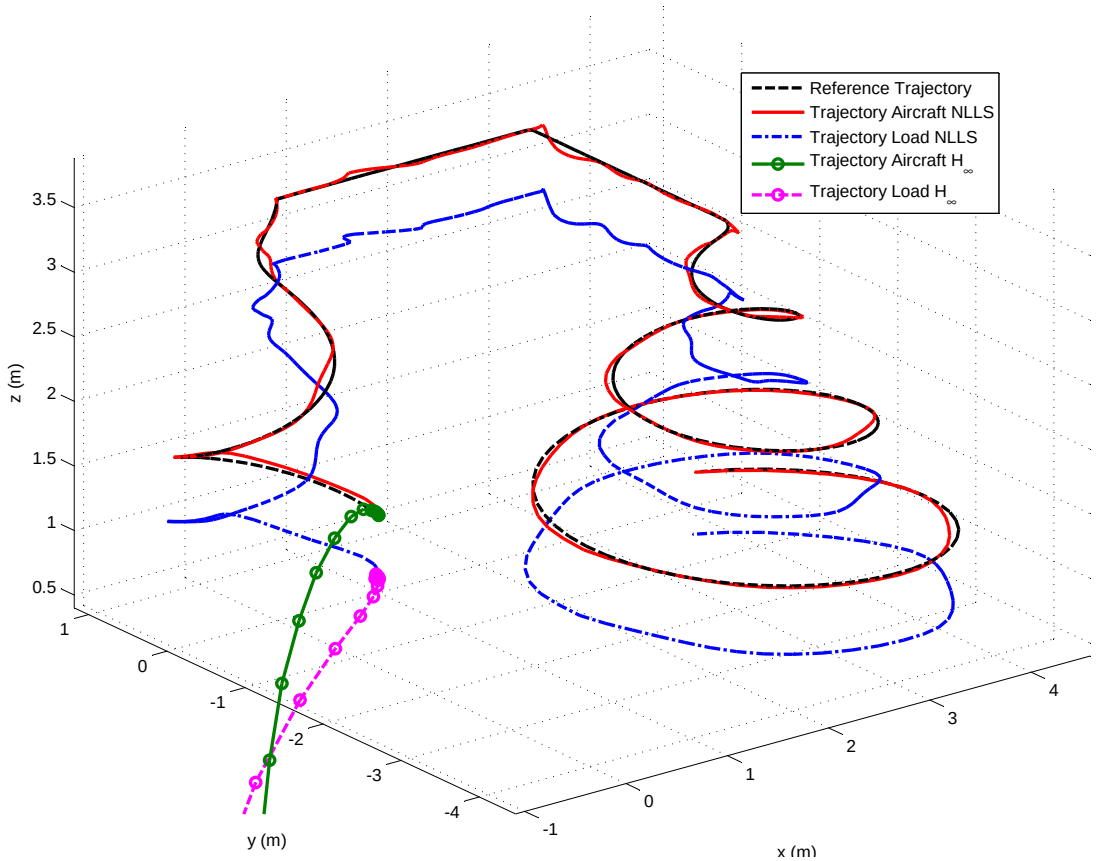

Figure 4.11: Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$ for a faster trajectory.

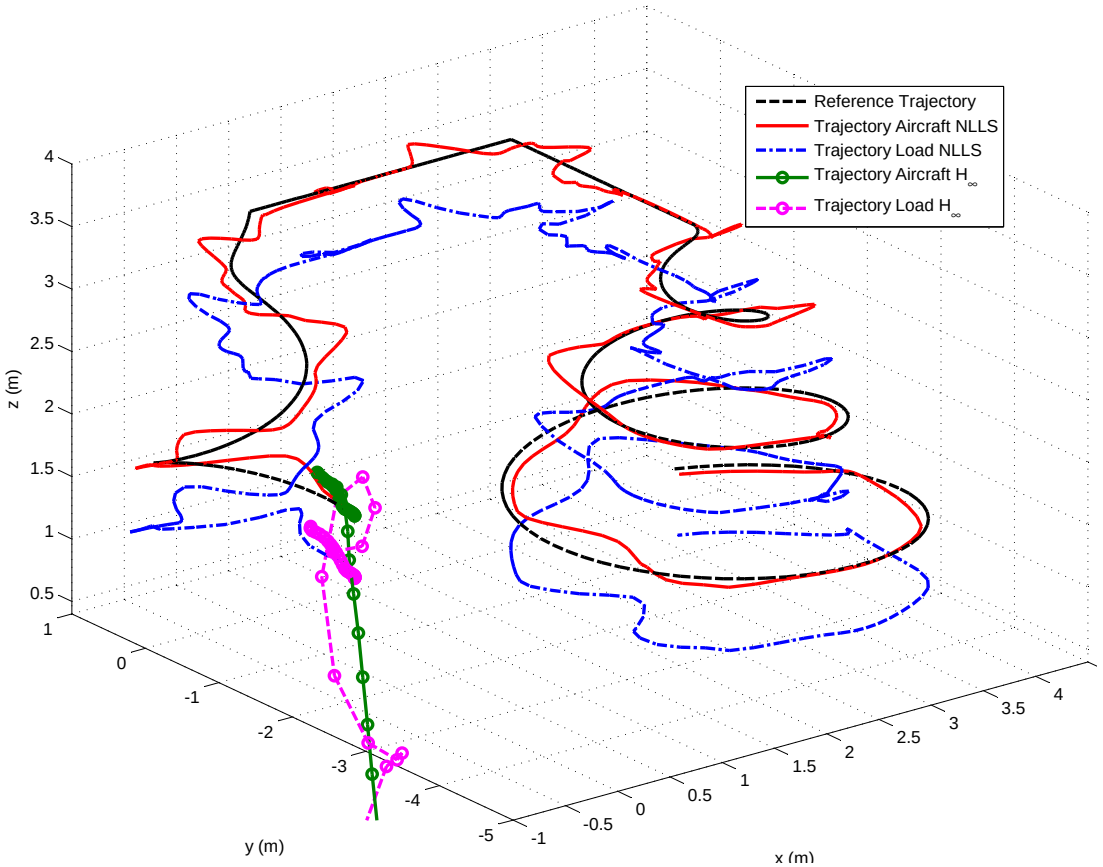

Figure 4.12: Path tracking of the aircraft for NLLS and $\mathcal{H}_\infty$ for harsher disturbances.

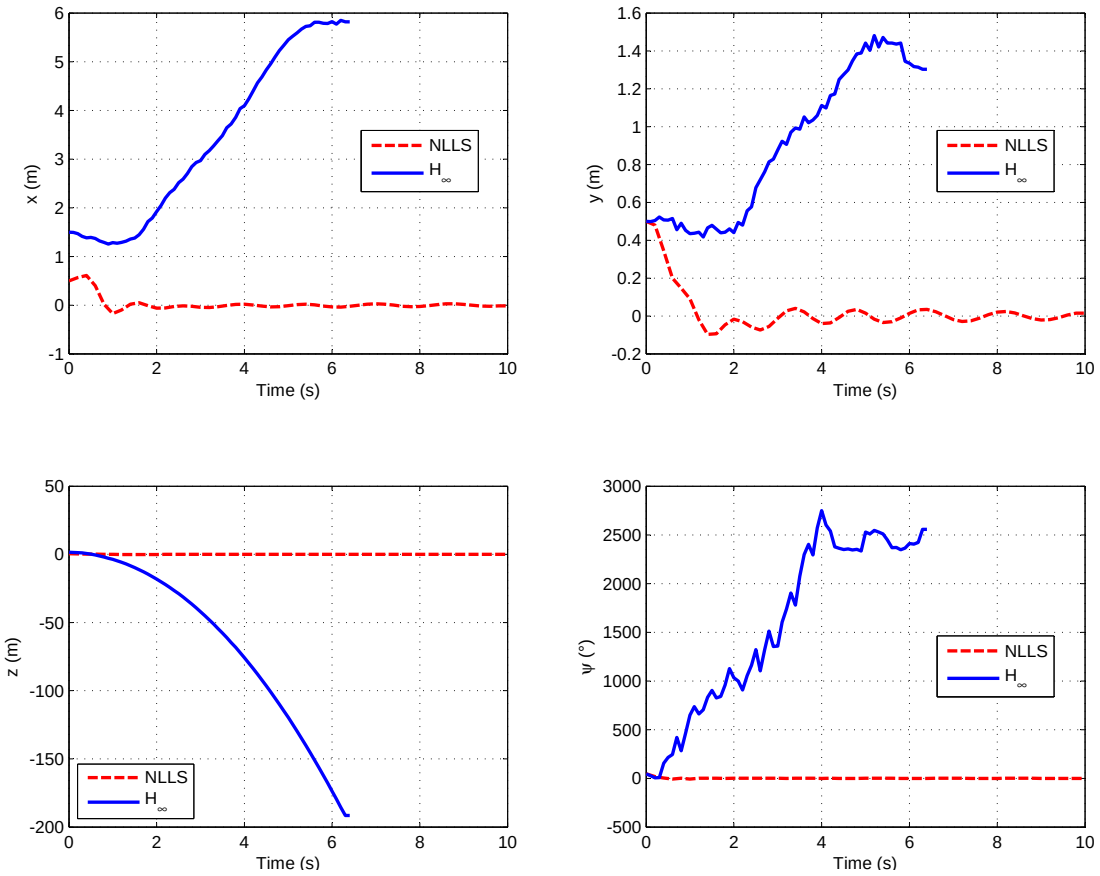

Figure 4.13: Tracking error on NLLS and $\mathcal{H}_\infty$ for Nonzero Initial Conditions.

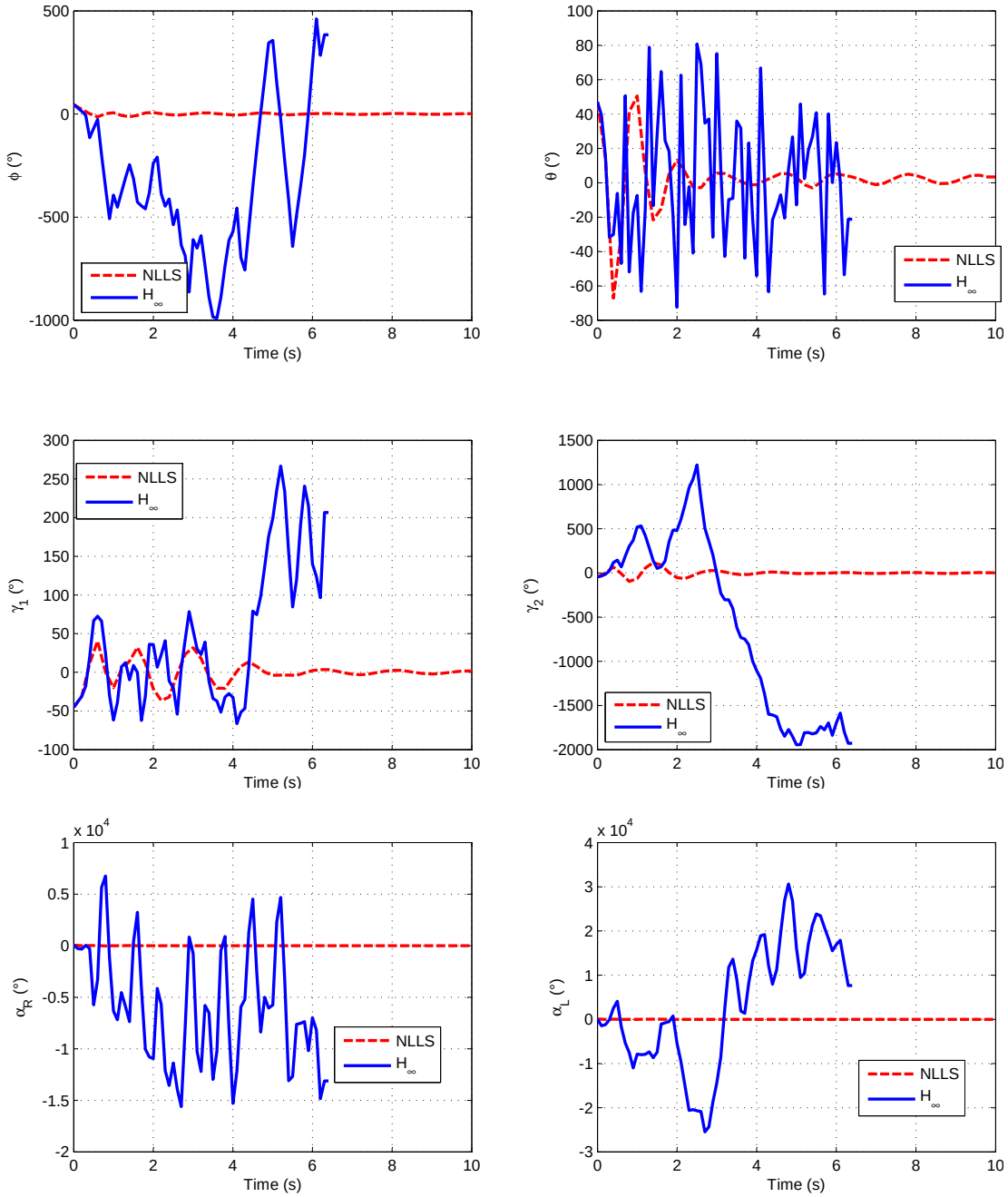

Figure 4.14: Body and Load angles on NLLS and $\mathcal{H}_\infty$ for Nonzero Initial Conditions.

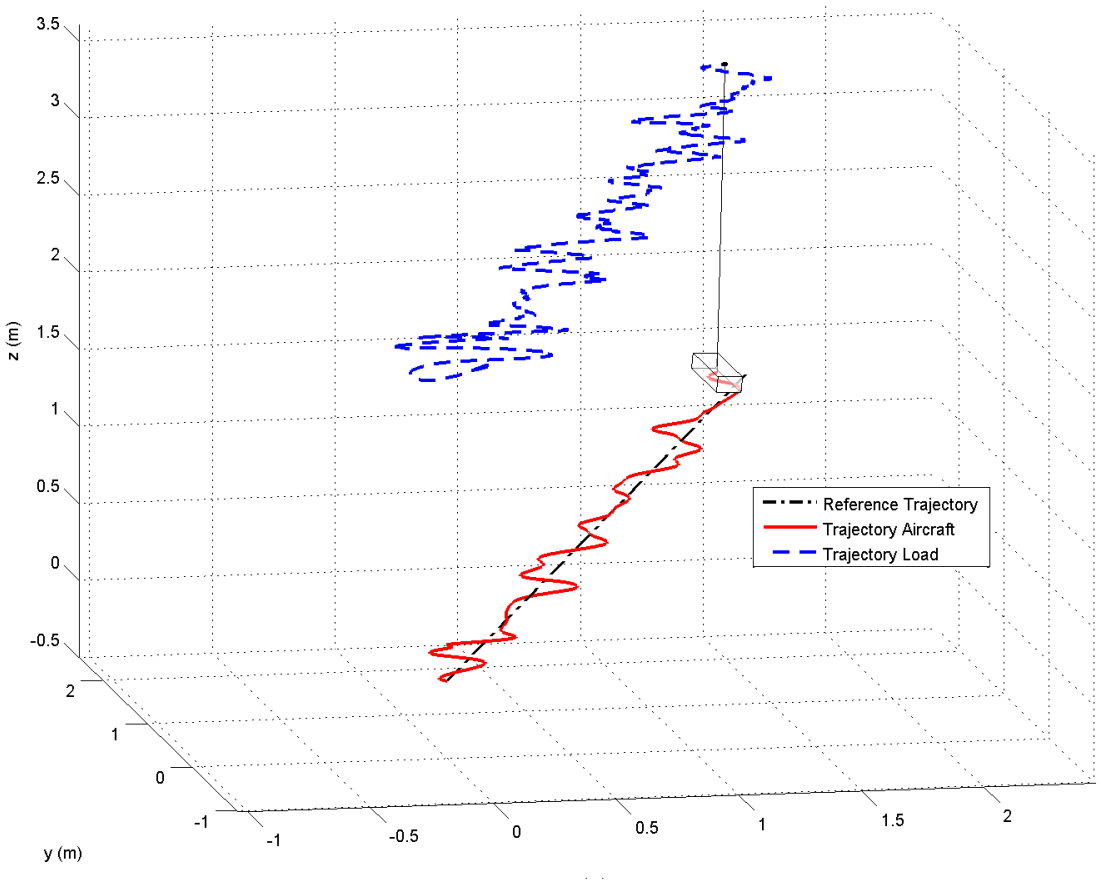

Figure 4.15: Path tracking of the aircraft with inverted pendulum.

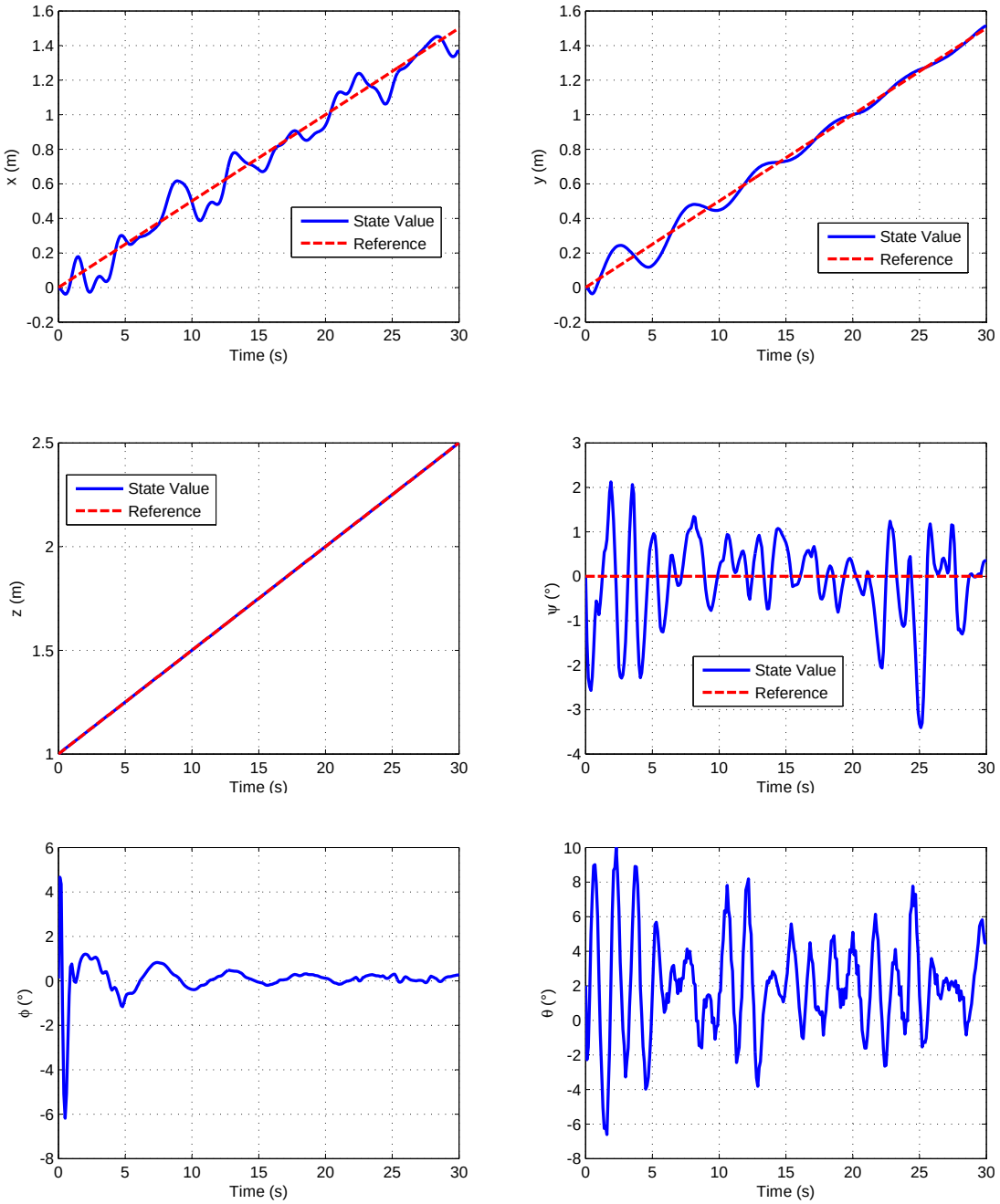

Figure 4.16: Aircraft's position and attitude for the inverted pendulum.

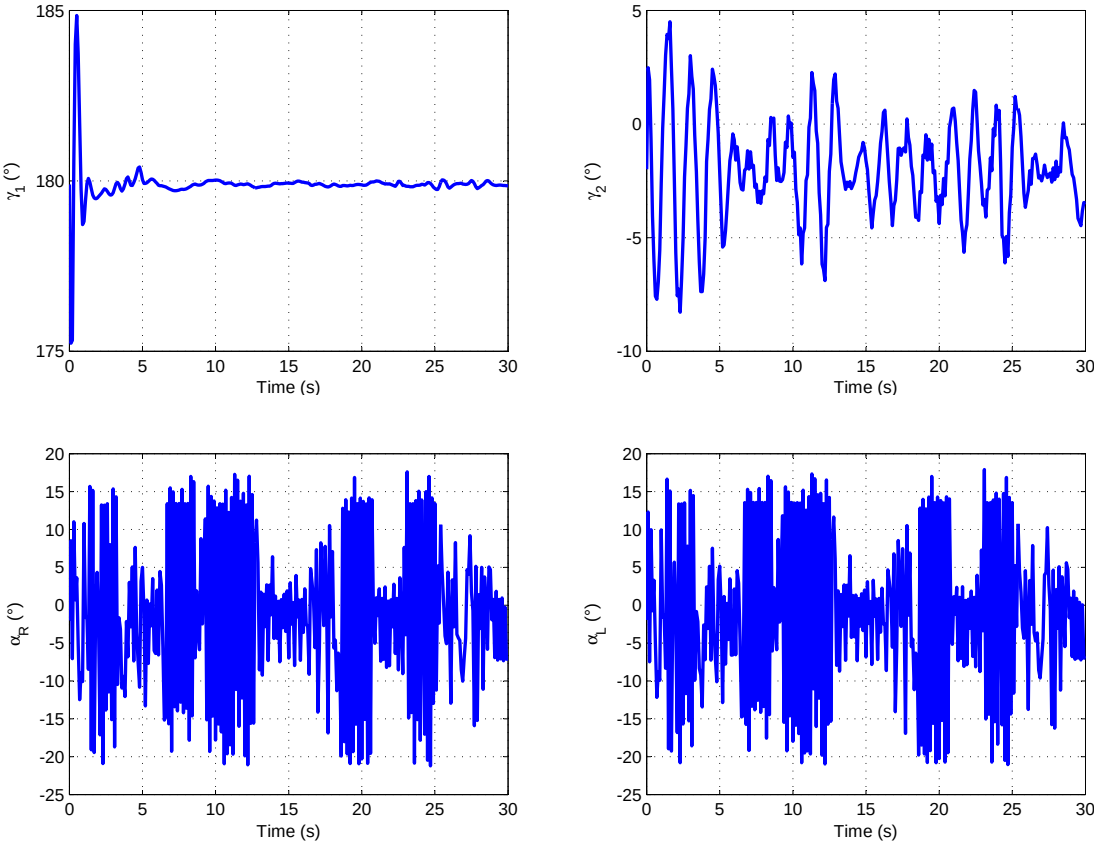

Figure 4.17: $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ in function of time for the inverted pendulum.

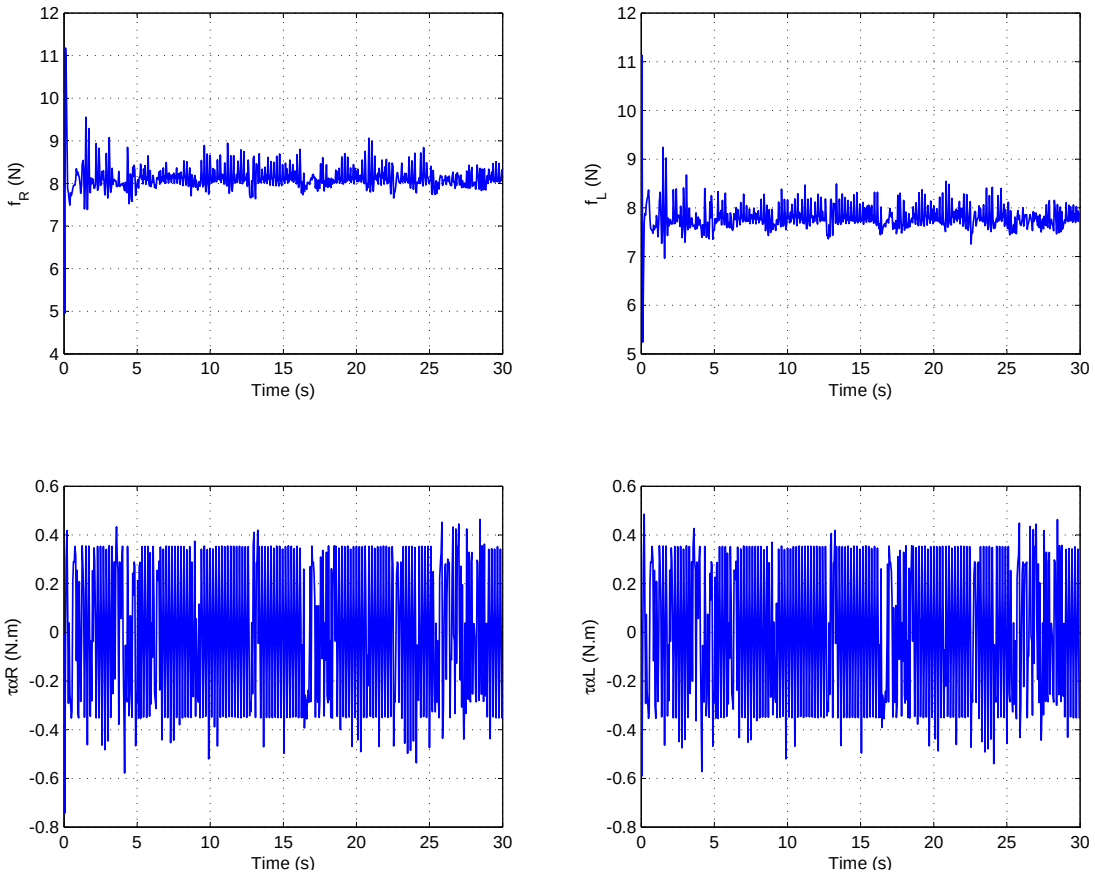

Figure 4.18: Inputs for the inverted pendulum.

Chapter 5

State Estimation

Sumário

5.1 Introduction 89
5.2 Technical Preliminaries and Notation 90
5.3 System Modelling for State Estimation 90
5.4 Kalman Filter with Unknown Inputs 94
5.5 Simulation Results and Analysis 96
5.6 Conclusion 98

5.1 Introduction

In the previous chapters it was assumed that all states were precisely known in all time instants. However, this assumption may not be true when dealing with real experiments. Usually, built-in sensors provide noisy measurements with their own sampling frequency, which may be different from the controller's frequency. A relevant problem in the development of UAVs is how to manage these difficulties, which is usually circumvented by using *state estimation* techniques.

This chapter focuses on the problem of state estimation for the *position and speed of the aircraft*, assuming though that the remaining state space variables are known. It assumes that the position is actually measured by a Global Positioning System (GPS) equipment with sampling time T_s, while the controller has a sampling time τ_s, with $\tau_s < T_s$ and assuming $T_s/\tau_s \in \mathbb{N}$. Therefore, the estimator must evaluate the position of the aircraft when no new measurements are available from the GPS receiver, also taking into account its measurement uncertainty. To further increase the challenge, the estimator must consider that the aircraft's motion may be affected by disturbances (e.g. wind gusts), which are not usually measured.

On state estimation problems it is important to know all the inputs that affect a system. Otherwise the estimates become biased due to the unknown input parameters. In order to solve this whole problem, the technique *Linear Kalman Filter with Unknown Inputs* (LKFUI) is used. Unlike classic *Linear Kalman Filter* (LKF) (Kalman, 1960), the LKFUI (Darouach et al., 2003) can deal with problems involving unknown inputs. In the present work, the formulation proposed by Teixeira et al. (2008) is used, which presents a generalized approach to Kalman Filters called *Gain-Constrained Kalman Filtering* in which both LKFUI and LKF are particular solutions.

The remainder of this chapter is organized as follows: section 5.2 presents some technical preliminaries and notation that is used all throughout the chapter; section 5.3 deals with the Tilt-rotor's state space model for state estimation; section 5.4 presents the equations used on LKFUI and section 5.4 provides simulation results [1].

5.2 Technical Preliminaries and Notation

For a certain state variable x, x_k denotes the value of the state at sampling instant k, being $\hat{x}_k$ its estimated value. $\boldsymbol{y}_k$ contains the values measured at instant k, while $\hat{\boldsymbol{y}}_k$ represents the estimation of the same variable. $\hat{\boldsymbol{x}}_{k|k-1}$ denotes that $\boldsymbol{x}$ at instant k has been predicted using measured information up to instant $k-1$. The notation used for the disturbance vector is $\boldsymbol{d}$ and the disturbance at instant $k-1$ is calculated at instant k, that is, one sampling time after it really happened.

Since this work deals with two different sampling times, $\hat{\boldsymbol{x}}_{i,k}$ $(i = 0, 1, 2, ..., T_s/\tau_s)$ denotes the estimated states of the system $i \cdot \tau_s$ seconds after the instant given at $\hat{\boldsymbol{x}}_{k|k}$. Index i represents increments in the controller's cycles, while index k represents sampling instants of the GPS. Every time that the predictor assimilates new measurements, $\hat{\boldsymbol{x}}_{i=0,k}$ is initialized as shown below:

$$\hat{\boldsymbol{x}}_{i=0,k} = \hat{\boldsymbol{x}}_{k|k}. \tag{5.1}$$

The following steps are used to obtain $\hat{\boldsymbol{x}}_{i,k}$:

Initialization

It is assumed initial estimates for $\hat{\boldsymbol{x}}_{0|0}$, $\hat{\boldsymbol{d}}_{0|0}$ and their uncertainties.

Prediction

- Calculate $\hat{\boldsymbol{x}}_{i|k}$ for $i = 0, 1, 2, ..., T_s/\tau_s$
- Calculate $\hat{\boldsymbol{x}}_{k+1|k}$

Correction

- Calculate $\hat{\boldsymbol{x}}_{k+1|k+1}$ given $\hat{\boldsymbol{x}}_{k+1|k}$ and the measurement $\boldsymbol{y}_{k+1}$
- Calculate $\hat{\boldsymbol{d}}_{k|k+1}$
- Increment k and then go back to the *Prediction* step.

5.3 System Modelling for State Estimation

Given that the system has two different sampling times, then two state-space equations are formulated distinguishing both sampling times. For the controller's sampling time τ_s

[1] The approach presented in this chapter was submitted and accepted to be published on CBA - 2014 (Almeida et al., 2014b).

it is considered a stochastic linear discrete-time dynamic system of the form:

$$\boldsymbol{x}_{i,k} = \boldsymbol{A}\boldsymbol{x}_{i-1,k} + \boldsymbol{B}\boldsymbol{u}_{i-1,k} + \boldsymbol{G}\boldsymbol{d}_{i-1,k} + \boldsymbol{w}_{i-1,k}, \tag{5.2}$$

where $\boldsymbol{A} \in \Re^{n\times n}$, $\boldsymbol{B} \in \Re^{n\times p}$ and $\boldsymbol{G} \in \Re^{n\times s}$ are known matrices in which n is the number of states of the system, p is the number of known inputs and s is the number of unknown inputs. It is assumed that for all $i, k \geq 1$ the input $\boldsymbol{u}_{i-1,k} \in \Re^p$ is known. The process noise $\boldsymbol{w}_{i-1,k} \in \Re^n$ is assumed to be white, Gaussian, zero-mean and mutually independent with known diagonal covariance matrix $\boldsymbol{Q}$. $\boldsymbol{x}_{i,k} \in \Re^n$ is the state vector, which should be estimated. The initial state vector $\boldsymbol{x}_{0,0} \in \Re^n$ is supposed to have Gaussian distribution with initial estimate $\hat{\boldsymbol{x}}_{0,0}$ and error covariance $\boldsymbol{P}_{0,0}^{xx} \triangleq \mathcal{E}[(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{0,0})^T(\boldsymbol{x}_0 - \hat{\boldsymbol{x}}_{0,0})]$, where $\mathcal{E}[\cdot]$ denotes expected value.

Since the model in equation (5.2) uses sampling time τ_s, these equations can be used iteratively to predict the system's states for sampling time T_s. Thus, to predict $\hat{\boldsymbol{x}}_{k|k-1}$ given $\hat{\boldsymbol{x}}_{k-1|k-1}$ and all the inputs $\boldsymbol{u}_{i,k-1}$ and $\boldsymbol{d}_{i,k-1}$ for $i = 1, \ldots, T_s/\tau_s$, this prediction can be calculated by finding $\hat{\boldsymbol{x}}_{i=T_s/\tau_s,k}$. By defining $h = T_s/\tau_s$, the prediction h samples ahead of $\hat{\boldsymbol{x}}_{k-1|k-1}$ can be calculated as shown below:

$$\begin{aligned}
\boldsymbol{x}_{1,k} =& \boldsymbol{A}\boldsymbol{x}_{0,k} + \boldsymbol{B}\boldsymbol{u}_{0,k} + \boldsymbol{G}\boldsymbol{d}_{0,k} \\
\boldsymbol{x}_{2,k} =& \boldsymbol{A}\boldsymbol{x}_{1,k} + \boldsymbol{B}\boldsymbol{u}_{1,k} \\
=& \boldsymbol{A}(\boldsymbol{A}\boldsymbol{x}_{0,k} + \boldsymbol{B}\boldsymbol{u}_{0,k} + \boldsymbol{G}\boldsymbol{d}_{0,k}) + \boldsymbol{B}\boldsymbol{u}_{1,k} + \boldsymbol{G}\boldsymbol{d}_{1,k} \\
=& \boldsymbol{A}^2\boldsymbol{x}_{0,k} + \boldsymbol{A}\boldsymbol{B}\boldsymbol{u}_{0,k} + \boldsymbol{A}\boldsymbol{G}\boldsymbol{d}_{0,k} + \boldsymbol{B}\boldsymbol{u}_{1,k} + \boldsymbol{G}\boldsymbol{d}_{1,k} \\
& \vdots \\
\boldsymbol{x}_{h,k} =& \boldsymbol{A}^h\boldsymbol{x}_{0,k} + \sum_{j=0}^{h-1} \boldsymbol{A}^{h-1-j}(\boldsymbol{B}\boldsymbol{u}_{j,k} + \boldsymbol{G}\boldsymbol{d}_{j,k}) \\
=& \boldsymbol{A}^h\boldsymbol{x}_{0,k} + [\boldsymbol{A}^{h-1}\boldsymbol{B} \;\; \boldsymbol{A}^{h-2}\boldsymbol{B} \;\; \ldots \;\; \boldsymbol{A}\boldsymbol{B} \;\; \boldsymbol{B}] \begin{bmatrix} \boldsymbol{u}_{0,k} \\ \boldsymbol{u}_{1,k} \\ \vdots \\ \boldsymbol{u}_{h-1,k} \end{bmatrix} + [\boldsymbol{A}^{h-1}\boldsymbol{G} \;\; \boldsymbol{A}^{h-2}\boldsymbol{G} \;\; \ldots \;\; \boldsymbol{A}\boldsymbol{G} \;\; \boldsymbol{G}] \begin{bmatrix} \boldsymbol{d}_{0,k} \\ \boldsymbol{d}_{1,k} \\ \vdots \\ \boldsymbol{d}_{h-1,k} \end{bmatrix} \\
=& \boldsymbol{A}_{T_s}\boldsymbol{x}_{0,k} + \boldsymbol{B}_{T_s}\vec{\boldsymbol{u}}_k + \boldsymbol{G}_{T_s}\vec{\boldsymbol{g}}_k,
\end{aligned}$$

where $\boldsymbol{A}_{T_s} = \boldsymbol{A}^h$, $\boldsymbol{B}_{T_s} = [\boldsymbol{A}^{h-1}\boldsymbol{B} \;\; \ldots \;\; \boldsymbol{B}]$, $\boldsymbol{G}_{T_s} = [\boldsymbol{A}^{h-1}\boldsymbol{G} \;\; \ldots \;\; \boldsymbol{G}]$, $\vec{\boldsymbol{u}}_k = [\boldsymbol{u}_{0,k} \;\; \ldots \;\; \boldsymbol{u}_{h-1,k}]^T$ and $\vec{\boldsymbol{d}}_k = [\boldsymbol{d}_{0,k} \;\; \ldots \;\; \boldsymbol{d}_{h-1,k}]^T$.

Therefore, when estimating the states at an instant k given information up to time $k-1$ the following state space equations should be used:

$$\begin{cases} \hat{\boldsymbol{x}}_{k|k-1} &= \boldsymbol{A}_{T_s}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_{T_s}\vec{\boldsymbol{u}}_{k-1} + \boldsymbol{G}_{T_s}\vec{\boldsymbol{d}}_{k-1} \\ \hat{\boldsymbol{y}}_k &= \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} \end{cases}, \tag{5.3}$$

where $\hat{\boldsymbol{y}}_k \in \Re^m$ is the estimated output of the system and is calculated at each time

instant k; $\boldsymbol{C} \in \Re^{m \times n}$ is a known matrix and m is the number of outputs. On the other hand, these same outputs are measured by the GPS receiver according to the following equation:

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k, \tag{5.4}$$

where the measurement noise $\boldsymbol{v}_k \in \Re^m$ is assumed to be white, Gaussian, zero-mean and mutually independent with known diagonal covariance matrix $\boldsymbol{R}_k$.

The equations of motion of a UAV on a coordinate system defined by the axes x, y and z can be obtained by using Newton's second law. It can be assumed that all known exerted forces along an axis are grouped into a single component (for example, on the z axis the gravity force can be grouped with the UAV vertical thrust into a single component). For instance, on the x axis it is possible to use the expression $\ddot{x} = \frac{T_x^{\mathcal{I}}}{m} + \frac{T_{dx}^{\mathcal{I}}}{m}$, where $T_x^{\mathcal{I}}$ is the known translational force along the axis x expressed in the inertial frame and $T_{dx}^{\mathcal{I}}$ is the unknown net disturbance force along the same axis and m is the body's mass. The expressions for the axis y and z can be easily obtained by simply exchanging x for y and z on the aforementioned expression.

Therefore, choosing the state variables $x_1 = x$, $x_2 = y$, $x_3 = z$, $x_4 = \dot{x}$, $x_5 = \dot{y}$ and $x_6 = \dot{z}$, the continuous-time state-space representation for this system is then given by:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \frac{T_x^{\mathcal{I}}}{m} \\ \frac{T_y^{\mathcal{I}}}{m} \\ \frac{T_z^{\mathcal{I}}}{m} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{T_{dx}^{\mathcal{I}}}{m} \\ \frac{T_{dy}^{\mathcal{I}}}{m} \\ \frac{T_{dz}^{\mathcal{I}}}{m} \end{bmatrix}. \tag{5.5}$$

Equation (5.5) can be approximated in the discrete-time domain with the system's sampling time τ_s as follows:

$$\begin{bmatrix} x_{1_{i+1}} \\ x_{2_{i+1}} \\ x_{3_{i+1}} \\ x_{4_{i+1}} \\ x_{5_{i+1}} \\ x_{6_{i+1}} \end{bmatrix} = \begin{bmatrix} x_{1_i} + \tau_s \cdot x_{4_i} \\ x_{2_i} + \tau_s \cdot x_{5_i} \\ x_{3_i} + \tau_s \cdot x_{6_i} \\ x_{4_i} \\ x_{5_i} \\ x_{6_i} \end{bmatrix} + \begin{bmatrix} \frac{\tau_s^2}{2} \cdot \frac{T_x^{\mathcal{I}}}{m} \\ \frac{\tau_s^2}{2} \cdot \frac{T_y^{\mathcal{I}}}{m} \\ \frac{\tau_s^2}{2} \cdot \frac{T_z^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_x^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_y^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_z^{\mathcal{I}}}{m} \end{bmatrix} + \begin{bmatrix} \frac{\tau_s^2}{2} \cdot \frac{T_{dx}^{\mathcal{I}}}{m} \\ \frac{\tau_s^2}{2} \cdot \frac{T_{dy}^{\mathcal{I}}}{m} \\ \frac{\tau_s^2}{2} \cdot \frac{T_{dz}^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_{dx}^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_{dy}^{\mathcal{I}}}{m} \\ \tau_s \cdot \frac{T_{dz}^{\mathcal{I}}}{m} \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}. \tag{5.6}$$

Thus, the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{G}$ of equation (5.2) are function of the sampling time τ_s and are given by:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \tau_s & 0 & 0 \\ 0 & 1 & 0 & 0 & \tau_s & 0 \\ 0 & 0 & 1 & 0 & 0 & \tau_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \frac{\tau_s^2}{2m} & 0 & 0 \\ 0 & \frac{\tau_s^2}{2m} & 0 \\ 0 & 0 & \frac{\tau_s^2}{2m} \\ \frac{\tau_s}{m} & 0 & 0 \\ 0 & \frac{\tau_s}{m} & 0 \\ 0 & 0 & \frac{\tau_s}{m} \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \frac{\tau_s^2}{2m} & 0 & 0 \\ 0 & \frac{\tau_s^2}{2m} & 0 \\ 0 & 0 & \frac{\tau_s^2}{2m} \\ \frac{\tau_s}{m} & 0 & 0 \\ 0 & \frac{\tau_s}{m} & 0 \\ 0 & 0 & \frac{\tau_s}{m} \end{bmatrix}. \tag{5.7}$$

Matrices $\boldsymbol{A}_{T_s}$, $\boldsymbol{B}_{T_s}$ and $\boldsymbol{G}_{T_s}$ are easily obtained from (5.7) and, by assumption, $\boldsymbol{C} = \mathbb{I}_{6\times 6}$.

In order to use equation (5.6), the forces $T_x^{\mathcal{I}}$, $T_y^{\mathcal{I}}$ and $T_z^{\mathcal{I}}$ have to be known. For the Tilt-rotor, these variables are obtained from equation (2.55) and are functions of the aircraft's attitude. Since attitude values are obtained by measurements, it is fair to say that $T_x^{\mathcal{I}}$, $T_y^{\mathcal{I}}$ and $T_z^{\mathcal{I}}$ are not precisely known. Therefore, the uncertainties of these forces along with uncertainties of the estimated disturbance force and modelling errors can be expressed by the process noise vector $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix}^T$.

In addition, it should be mentioned that for the LKFUI the unknown inputs are estimated once at each instant of the GPS's measurement and are denoted by $d_{k-1|k}$. When using equation (5.3) to estimate $\hat{\boldsymbol{x}}_{k|k-1}$, the vector of unknown inputs $\vec{\boldsymbol{d}}_{k-1}$ possesses its terms equal to $\boldsymbol{d}_{k-1|k}$ for all $i = 1, \ldots, h$. Therefore, when calculating $\boldsymbol{G}_{T_s}\vec{\boldsymbol{d}}_{k-1}$ it can be proven that for the given matrices of equation (5.7):

$$\boldsymbol{G}_{T_s}\vec{\boldsymbol{d}}_{k-1} = \left(\sum_{j=0}^{h-1} \boldsymbol{A}^j \boldsymbol{G}\right) d_{k-1|k} = \begin{bmatrix} \frac{T_s^2}{2m} & 0 & 0 \\ 0 & \frac{T_s^2}{2m} & 0 \\ 0 & 0 & \frac{T_s^2}{2m} \\ \frac{T_s}{m} & 0 & 0 \\ 0 & \frac{T_s}{m} & 0 \\ 0 & 0 & \frac{T_s}{m} \end{bmatrix} d_{k-1|k}. \tag{5.8}$$

Therefore, it can be said that $\boldsymbol{G}_{T_s}\vec{\boldsymbol{d}}_{k-1} = \bar{\boldsymbol{G}}_{T_s} d_{k-1|k}$ where the matrix $\bar{\boldsymbol{G}}_{T_s}$ is equal to matrix $\boldsymbol{G}$ of equation (5.7) except that τ_s is substituted by T_s.

Equation (5.8) is easily demonstrated for a system that moves along a single axis, but the idea can be easily extended for the three-dimensional system. Thus, considering a system that moves only along axis x this can be represented by a discrete state space model with two state variables: x and $\dot{x}$. Then, matrices $\boldsymbol{A}$ and $\boldsymbol{G}$ can be adapted from equation (5.7) with the reduced form:

$$\boldsymbol{A}_{red} = \begin{bmatrix} 1 & \tau_s \\ 0 & 1 \end{bmatrix}, \boldsymbol{G}_{red} = \begin{bmatrix} \frac{\tau_s^2}{2m} \\ \frac{\tau_s}{m} \end{bmatrix}. \tag{5.9}$$

It can be seen by direct calculation that:

$$\boldsymbol{A}_{red}^{j} = \begin{bmatrix} 1 & j.\tau_s \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_{red}^{j}.\boldsymbol{G}_{red} = \begin{bmatrix} (2j+1)\frac{\tau_s^2}{2m} \\ \frac{\tau_s}{m} \end{bmatrix}. \tag{5.10}$$

Thus, $\sum_{j=0}^{h-1} \boldsymbol{A}_{red}^{j}\boldsymbol{G}_{red}$ can be obtained by calculating the summations $\sum_{j=0}^{h-1}(2j+1)\frac{\tau_s^2}{2m}$ and $\sum_{j=0}^{h-1}\frac{\tau_s}{m}$ separately. Therefore, noting that $\sum_{j=0}^{h-1}(2j+1)$ is the sum of the first h odd numbers:

$$\sum_{j=0}^{h-1}(2j+1)\frac{\tau_s^2}{2m} = \frac{\tau_s^2}{2m}\sum_{j=0}^{h-1}(2j+1) = \frac{(h\tau_s)^2}{2m}$$
$$\sum_{j=0}^{h-1}\frac{\tau_s}{m} = \frac{h\tau_s}{m}.$$

Since $h = T_s/\tau_s$ then $h\tau_s = T_s$ and $\sum_{j=0}^{h-1} \boldsymbol{A}_{red}^{j}\boldsymbol{G}_{red} = \bar{\boldsymbol{G}}_{red}$, where $\bar{\boldsymbol{G}}_{red}$ is equal to $\boldsymbol{G}$ except that τ_s is substituted by T_s.

5.4 Kalman Filter with Unknown Inputs

Considering the model presented in Section 5.3, the LKFUI uses the three-step algorithm shown on section 5.2. On its first step it only initializes its variables. On the second step, it predicts the system's states and outputs given previous measurements (equations (5.11)-(5.12)). In the third step it assimilates new incoming data updating the states estimation (equation (5.13)).

$$\hat{\boldsymbol{x}}_{k|k-1} = \boldsymbol{A}_{T_s}\hat{\boldsymbol{x}}_{k-1|k-1} + \boldsymbol{B}_{T_s}\vec{\boldsymbol{u}}_{k-1} \tag{5.11}$$
$$\hat{\boldsymbol{y}}_{k|k-1} = \boldsymbol{C}\hat{\boldsymbol{x}}_{k|k-1} \tag{5.12}$$
$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{L}_k(\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}). \tag{5.13}$$

The filter gain $\boldsymbol{L}_k \in \Re^{n\times m}$ must be chosen such that it minimizes the cost function:

$$\boldsymbol{J}_k(\boldsymbol{L}_k) \triangleq \mathcal{E}[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})^T(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k})]. \tag{5.14}$$

subject to the constraint:

$$\boldsymbol{L}_k\boldsymbol{E}_k = \boldsymbol{F}_k. \tag{5.15}$$

where $\mathbf{E}_k$ and $\mathbf{F}_k$ solve the problem of the LKFUI when they are given by (Teixeira et al., 2008):

$$\boldsymbol{E}_k = \boldsymbol{C}\boldsymbol{G} \qquad \boldsymbol{F}_k = \boldsymbol{G}. \tag{5.16}$$

The *forecast error* $\boldsymbol{e}_{k|k-1}$, the *innovation* $\boldsymbol{\nu}_{k|k-1}$ and the *data-assimilation error* $\boldsymbol{e}_{k|k}$ are defined as:

$$\boldsymbol{e}_{k|k-1} \triangleq \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1} \tag{5.17}$$
$$\boldsymbol{\nu}_{k|k-1} \triangleq \boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1} \tag{5.18}$$
$$\boldsymbol{e}_{k|k} \triangleq \boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k}. \tag{5.19}$$

Similarly, the *forecast error covariance* $\boldsymbol{P}^{xx}_{k|k-1}$, the *innovation covariance* $\boldsymbol{P}^{yy}_{k|k-1}$, the *cross covariance* $\boldsymbol{P}^{xy}_{k|k-1}$ and the *data-assimilation error covariance* $\boldsymbol{P}^{xx}_{k|k}$ are defined as:

$$\boldsymbol{P}^{xx}_{k|k-1} \triangleq \mathcal{E}[\boldsymbol{e}_{k|k-1}\boldsymbol{e}^T_{k|k-1}] \tag{5.20}$$
$$\boldsymbol{P}^{yy}_{k|k-1} \triangleq \mathcal{E}[\boldsymbol{\nu}_{k|k-1}\boldsymbol{\nu}^T_{k|k-1}] \tag{5.21}$$
$$\boldsymbol{P}^{xy}_{k|k-1} \triangleq \mathcal{E}[\boldsymbol{e}_{k|k-1}\boldsymbol{\nu}^T_{k|k-1}] \tag{5.22}$$
$$\boldsymbol{P}^{xx}_{k|k} \triangleq \mathcal{E}[\boldsymbol{e}_{k|k}\boldsymbol{e}^T_{k|k}]. \tag{5.23}$$

From the filter of equations (5.11)-(5.13), the data assimilation error covariance $\boldsymbol{P}^{xx}_{k|k}$ is given by:

$$\boldsymbol{P}^{xx}_{k|k} = \boldsymbol{P}^{xx}_{k|k-1} - \boldsymbol{L}_k(\boldsymbol{P}^{xy}_{k|k-1})^T - \boldsymbol{P}^{xy}_{k|k-1}\boldsymbol{L}^T_k + \boldsymbol{L}_k\boldsymbol{P}^{yy}_{k|k-1}\boldsymbol{L}^T_k, \tag{5.24}$$

where:

$$\boldsymbol{P}^{xx}_{k|k-1} = \boldsymbol{A}_{T_s}\boldsymbol{P}^{xx}_{k-1|k-1}\boldsymbol{A}^T_{T_s} + \boldsymbol{Q} \tag{5.25}$$
$$\boldsymbol{P}^{yy}_{k|k-1} = \boldsymbol{C}\boldsymbol{P}^{xx}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{R}_k \tag{5.26}$$
$$\boldsymbol{P}^{xy}_{k|k-1} = \boldsymbol{P}^{xx}_{k|k-1}\boldsymbol{C}^T. \tag{5.27}$$

Therefore, using (5.19) and (5.23) into (5.14), the cost function can be expressed by:

$$\boldsymbol{J}_k(\boldsymbol{L}_k) = tr(\boldsymbol{P}^{xx}_{k|k}). \tag{5.28}$$

For convenience, the following equations are defined:

$$\begin{aligned}
\boldsymbol{E}_k^L &\triangleq (\boldsymbol{E}_k^T \boldsymbol{E}_K)^{-1} \boldsymbol{E}_k^T, && (5.29)\\
\boldsymbol{\Omega}_k &\triangleq \boldsymbol{E}_k [\boldsymbol{E}_k^T (\boldsymbol{P}_{k|k-1}^{yy})^{-1} \boldsymbol{E}_k]^{-1} \boldsymbol{E}_k^T (\boldsymbol{P}_{k|k-1}^{yy})^{-1}, && (5.30)\\
\boldsymbol{\Omega}_{k\perp} &\triangleq \mathbb{I}_{m\times m} - \boldsymbol{\Omega}_k, && (5.31)\\
\boldsymbol{K}_k &\triangleq \boldsymbol{P}_{k|k-1}^{xy} (\boldsymbol{P}_{k|k-1}^{yy})^{-1}. && (5.32)
\end{aligned}$$

It is interesting to note that K_k is the classical Kalman gain.

The gain $\boldsymbol{L}_k$ that minimizes (5.28) and satisfies (5.15) is given by:

$$\boldsymbol{L}_k = \boldsymbol{K}_k \boldsymbol{\Omega}_{k\perp} + \boldsymbol{F}_k \boldsymbol{E}_k^L \boldsymbol{\Omega}_k. \tag{5.33}$$

The *data-assimilation error covariance* is then given by the Riccati equation:

$$\begin{aligned}
\boldsymbol{P}_{k|k}^{xx} &= \boldsymbol{P}_{k|k-1}^{xx} - \boldsymbol{P}_{k|k-1}^{xy} (\boldsymbol{P}_{k|k-1}^{yy})^{-1} (\boldsymbol{P}_{k|k-1}^{xy})^T + (\boldsymbol{F}_k \boldsymbol{E}_k^L \boldsymbol{\Omega}_k) \boldsymbol{P}_{k|k-1}^{yy} (\boldsymbol{F}_k \boldsymbol{E}_k^L \boldsymbol{\Omega}_k)^T \\
&\quad + [\boldsymbol{P}_{k|k-1}^{xy} (\boldsymbol{P}_{k|k-1}^{yy})^{-1} \boldsymbol{\Omega}_k] \boldsymbol{P}_{k|k-1}^{yy} [\boldsymbol{P}_{k|k-1}^{xy} (\boldsymbol{P}_{k|k-1}^{yy})^{-1} \boldsymbol{\Omega}_k]^T - (\boldsymbol{\Delta}_{1,k} + \boldsymbol{\Delta}_{1,k}^T), \quad (5.34)
\end{aligned}$$

where:

$$\boldsymbol{\Delta}_{1,k} \triangleq \boldsymbol{P}_{k|k-1}^{xy} (\boldsymbol{P}_{k|k-1}^{yy})^{-1} \boldsymbol{\Omega}_k \boldsymbol{P}_{k|k-1}^{yy} \boldsymbol{\Omega}_k^T (\boldsymbol{F}_k \boldsymbol{E}_k^L)^T. \tag{5.35}$$

The estimation of the disturbance $\hat{\boldsymbol{d}}_{k-1|k}$ can be obtained as follows:

$$\hat{\boldsymbol{d}}_{k-1|k} = (\boldsymbol{G}_{k-1}^T \boldsymbol{G}_{k-1})^{-1} \boldsymbol{G}_{k-1}^T \boldsymbol{L}_k (\boldsymbol{y}_k - \hat{\boldsymbol{y}}_{k|k-1}). \tag{5.36}$$

Therefore, the states of the aircraft can be estimated using the controller's sampling frequency τ_s. The estimate $\hat{\boldsymbol{x}}_{i,k}$ is initialized ($i = 0$) as shown on equation (5.1) and for $i = 1, 2, ..., T_s/\tau_s$, it is given by:

$$\hat{\boldsymbol{x}}_{i,k} = \boldsymbol{A} \hat{x}_{i-1,k} + \boldsymbol{B} \boldsymbol{u}_{i-1,k} + \boldsymbol{G} \hat{\boldsymbol{d}}_{k-1|k}. \tag{5.37}$$

It is important to notice that the control input used is $\boldsymbol{u}_{i-1,k}$, which represents the last input value applied to the system on the previous controller cycle.

5.5 Simulation Results and Analysis

This section shows simulation results carried out to analyse the performance of the designed controllers when subject to position measurement uncertainties. The simulated

system has the GPS receiver with sampling rate equal to $T_s = 0.1$s[2] and the controller's sampling rate is equal to $\tau_s = 0.01$s. The simulation tool used a fixed-step simulation with period τ_s and the GPS measures passes through a zero-order-hold with period T_s. Furthermore, it is assumed GPS measurements with Gaussian accuracy of $3\sigma = \pm 1.5m$ for position measurements and a speed sensor with accuracy of $3\sigma = \pm 0.1m/s$, where σ is the standard deviation. The translational forces $T_x^{\mathcal{I}}$, $T_y^{\mathcal{I}}$ and $T_z^{\mathcal{I}}$ are estimated using the Tilt-rotor's model and the applied control input, and Gaussian noise with $3\sigma = \pm 1N$ is added to their values. The matrix $\boldsymbol{Q}$ was tuned empirically using iterative trial-and-error method and its value is given by:

$$\boldsymbol{Q} = diag(1 \times 10^{-3}, 1 \times 10^{-3}, 1 \times 10^{-3}, 2 \times 10^{-3}, 2 \times 10^{-3}, 2 \times 10^{-3}). \tag{5.38}$$

The system was simulated using the same strategy as the Nonlinear Controller for Reduced Load Swing (NLLS), except that the parameters for the controllers of x, y and z were modified to the ones shown in table 5.1. These modifications were necessary because the previous chapter assumed noiseless measurements, while now the measurements are noisy and the state estimation is not a smooth curve. Then, in order to maintain the system stable, the translational controllers were tuned more conservatively.

The external disturbance profile is shown in Figure 5.1. The system had to track the same path as the one used in previous simulations (the only difference is the disturbance). Figure 5.2 shows that the aircraft did not follow the reference as good as it followed in the previous chapters. In order to explain this worsening, Figure 5.3 shows the position estimation error for the ten first seconds of the simulation. It can be seen that although the LKFUI was able to converge the position estimation, its estimation error is bounded to be $\pm 38cm$ with 99.8% of confidence or, likewise, the estimation error is bounded to be $\pm 25.3cm$ with 95% of confidence.

Therefore, it is possible to note in Figure 5.4 that the tracking error is mostly bounded between $\pm 30cm$. The curve *Set Point Error* shows how the set point deviated from its desired value as consequence of the uncertain position estimation. The time evolution of the remaining generalized coordinates are shown in Figure 5.5 and the system's inputs are shown in Figure 5.6.

Figure 5.7 shows the disturbance estimation. It can be seen that the LKFUI algorithm was capable of following the tendency of the disturbance, while estimating with reasonable variance.

At last, Table 5.2 shows the mean-square-error of the variables for the LKFUI simulation. Even though these values cannot be literally compared to the MSEs of D-stability, $\mathcal{H}_\infty$ or NLLS (the simulation was not equal, given that the disturbances profiles were different), it can be seen that its magnitude is higher than what was seen in Tables 3.1 and 4.8.

[2]This value is equivalent to the specifications of the Novatel OEMStar GPS receiver operating on the DGPS mode used at the ProVant project.

Table 5.1: Translational controllers parameters for use with LKFUI.

Variable	K_p	K_I	K_d
x	-2.9256	-0.9479	-2.9782
y	-2.9256	-0.9479	-2.9782
y	-2.9256	-0.9479	-2.9782

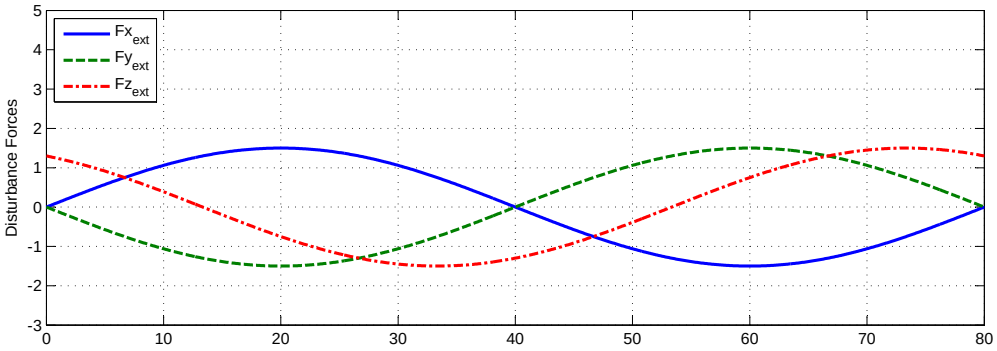

Figure 5.1: System disturbances for the LKFUI simulation.

5.6 Conclusion

This chapter dealt with the problem of position state estimation in the presence of disturbances, and with noisy measurements with lower sampling frequency than the controller. In order to solve this problem, it was used the Linear Kalman Filter with Unknown Inputs (LKFUI).

The state space model that was used for the LKFUI algorithm simplifies the dynamics of the tilt-rotor by deriving its equations from Newton's second law. This way, there is no need to use a nonlinear state estimation algorithm. However, an interesting future work would be to implement an Unscented Kalman Filter with Unknown Inputs (Teixeira et al., 2008) so as to use the Tilt-rotor's nonlinear equations for estimation and compare the results.

Simulation results were presented where the LKFUI algorithm was able to estimate the disturbances and use its information along with the aircraft translational forces so as to estimate the system's position while the GPS receiver does not provide new measurements. At the moment that the GPS receiver provides a new measurement, a correction step is taken to merge the information from the measurement along with the previous estimation. Consequently, the LKFUI was able to reduce estimation error variability from $\pm 1.5m$ (GPS receiver's measurement error) to approximately $\pm 0.38m$.

Table 5.2: Mean-square-error of the LKFUI simulation.

MSE_x	MSE_y	MSE_z	MSE_ψ
$9,03.10^{-3}$	$9,59.10^{-3}$	$1,95.10^{-2}$	$7,83.10^{-5}$

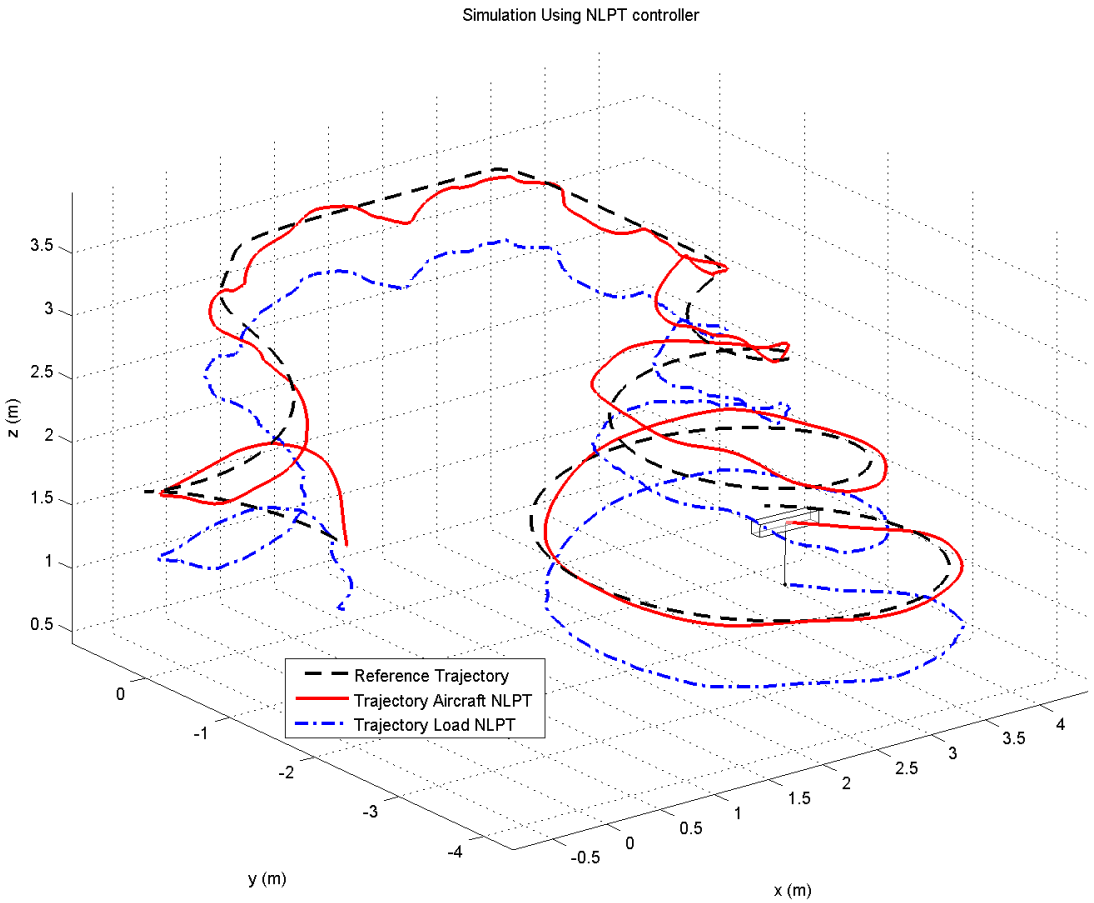

Figure 5.2: Path tracking of the aircraft for the LKFUI simulation.

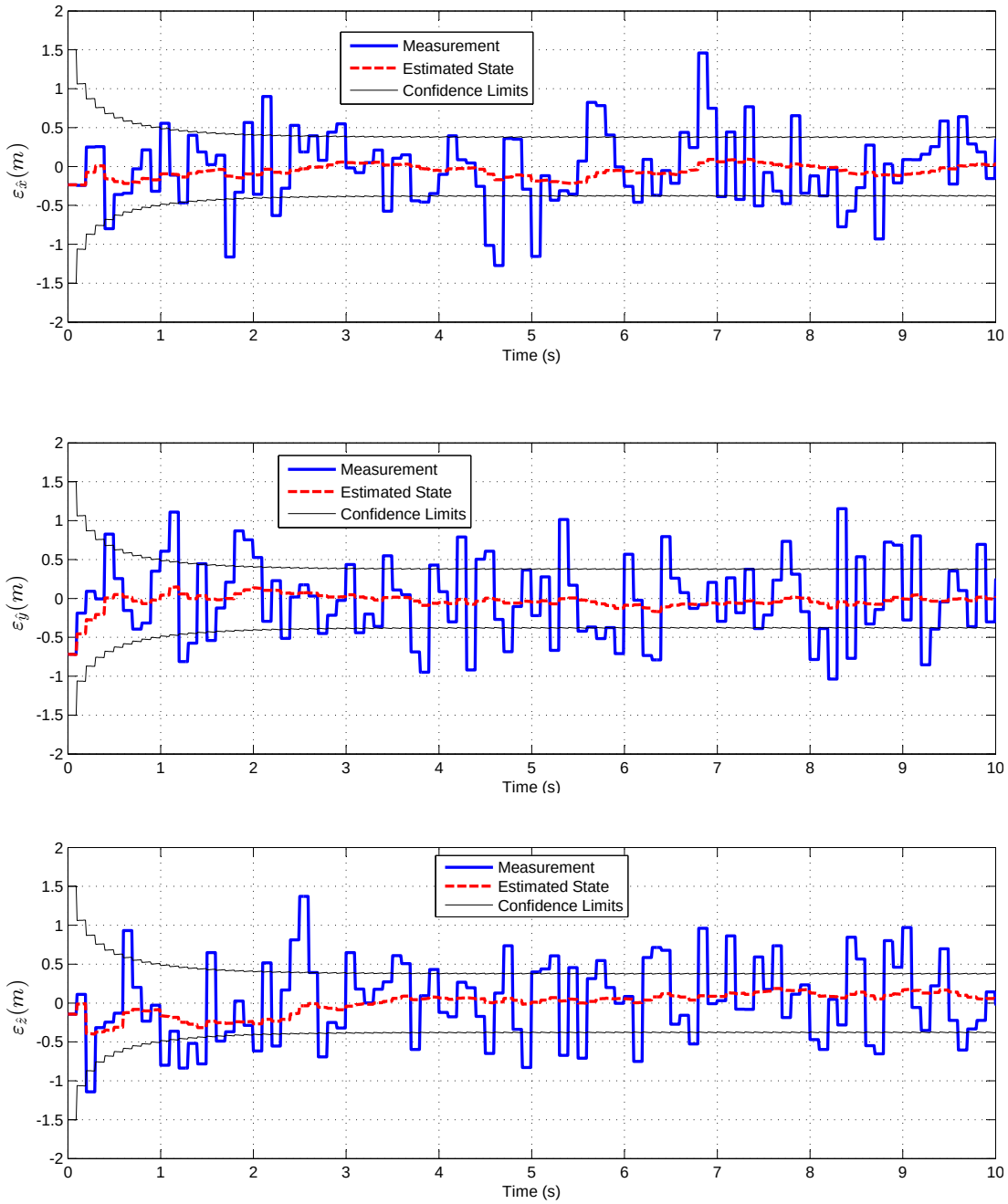

Figure 5.3: Position estimation error in function of time for the LKFUI simulation.

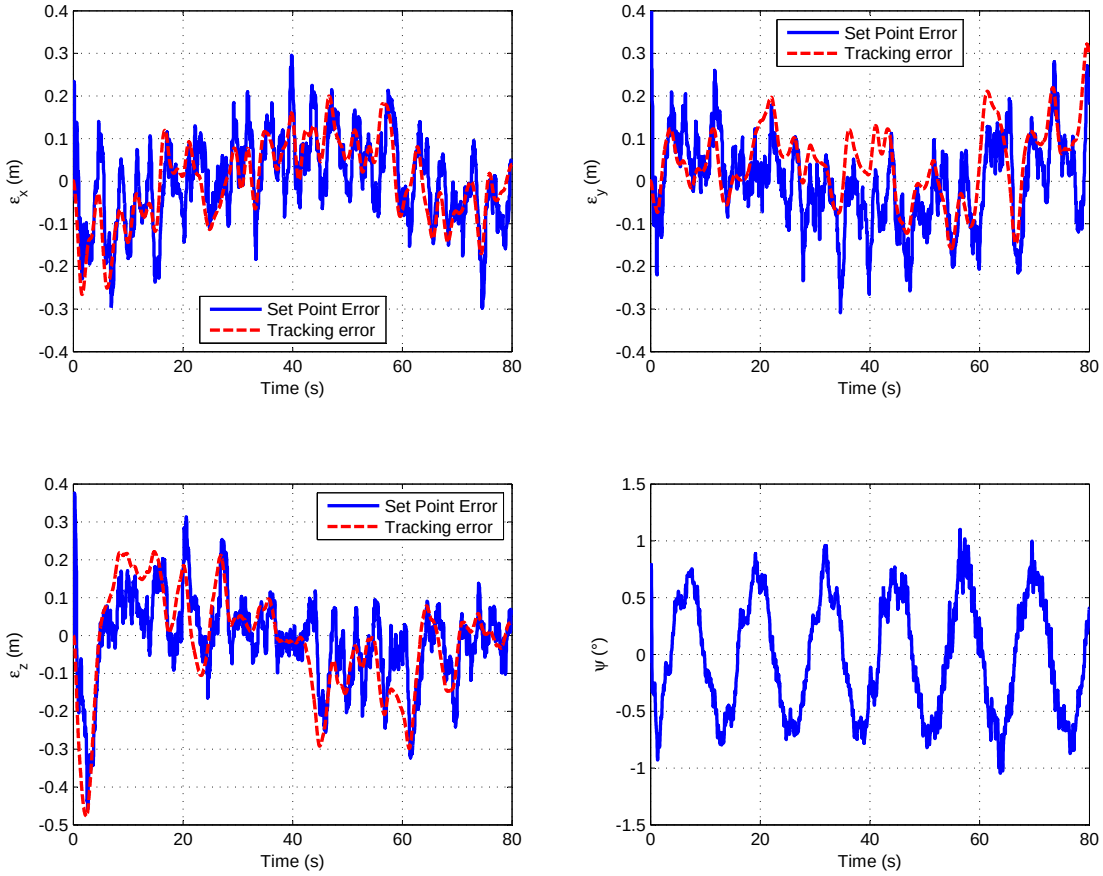

Figure 5.4: Tracking error for the LKFUI simulation.

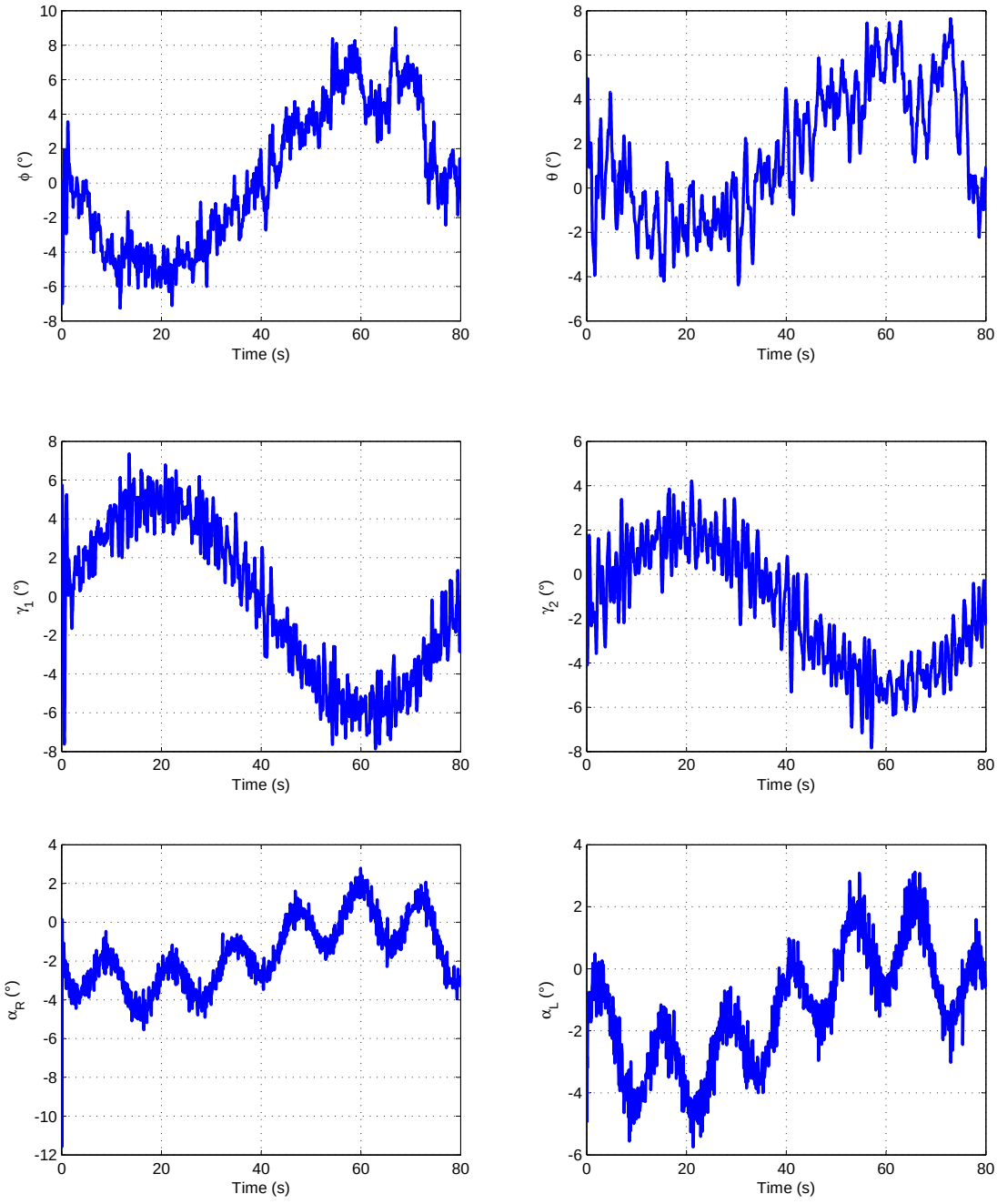

Figure 5.5: Body and Load angles for the LKFUI simulation.

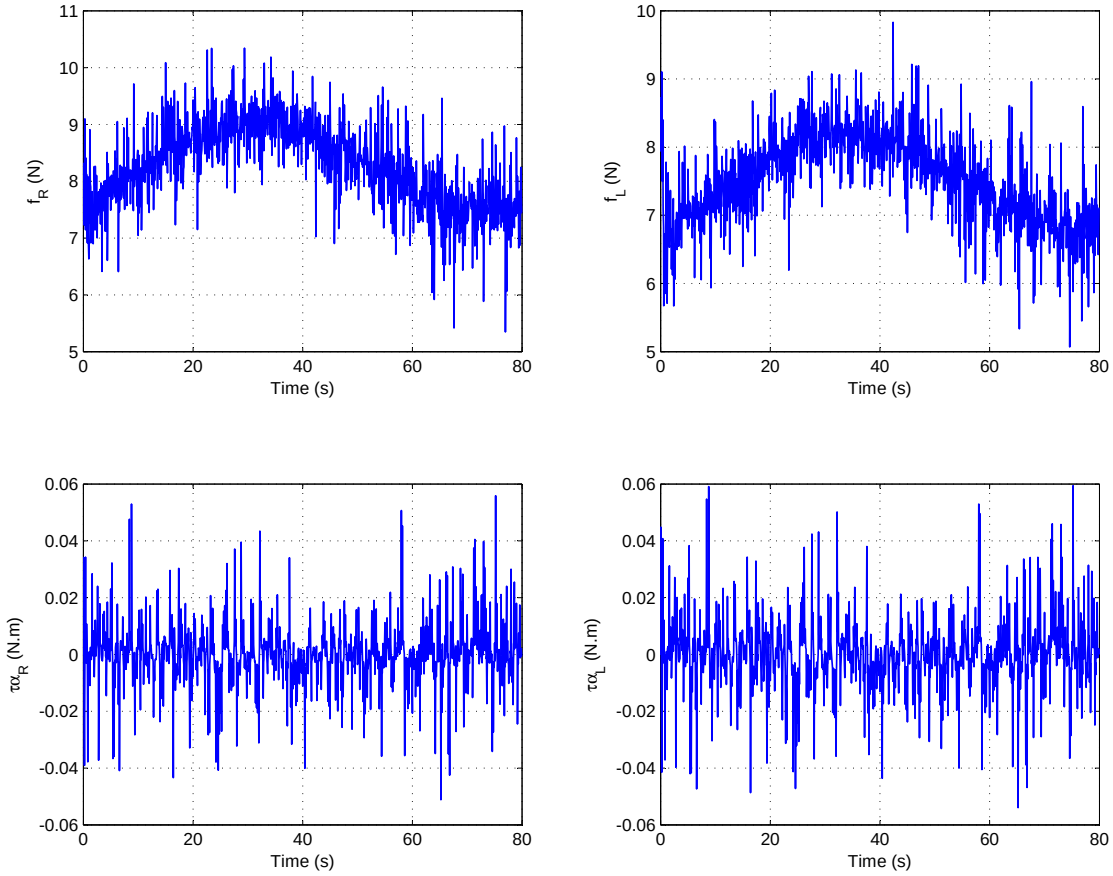

Figure 5.6: Inputs of the system for the LKFUI simulation.

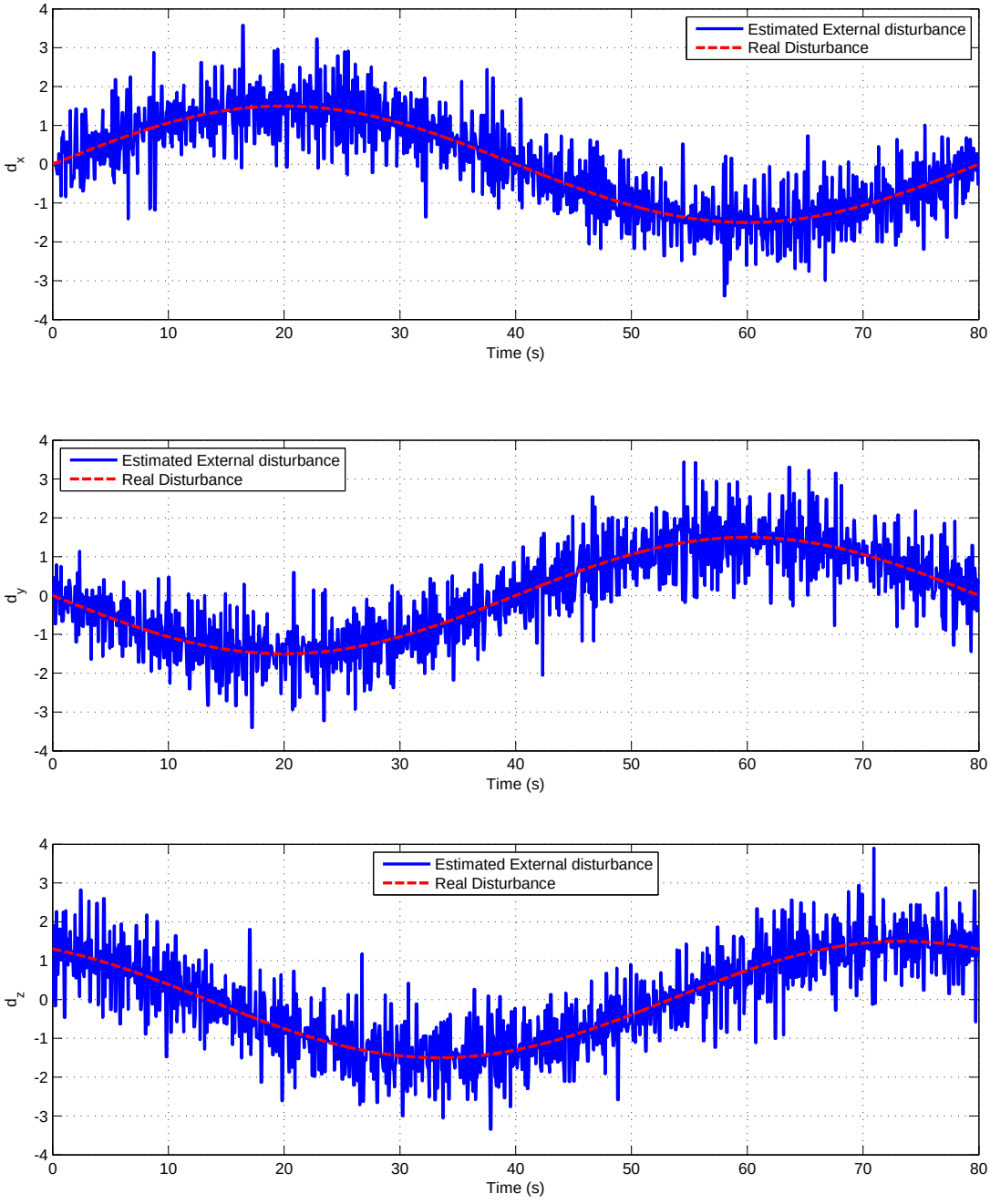

Figure 5.7: Disturbance estimation in function of time for the LKFUI simulation.

Chapter 6

Conclusions

Sumário

6.1 Future Works 107

This dissertation presented control strategies to solve the problem of suspended load transportation by a Tilt-rotor Unmanned Aerial Vehicle passing through a desired trajectory. These strategies were developed taking into account the presence of external disturbances, parametric uncertainties and measurement errors. The controllers provided path tracking for x, y, z and ψ whilst stabilizing all other generalized coordinates.

In order to design advanced control laws, an accurate dynamic model of the system is needed. Therefore, a rigorous dynamic model for the Tilt-rotor UAV with suspended load was derived using Euler-Lagrange formulation. This model considers the system as composed of four bodies, resulting in a highly coupled nonlinear dynamic model. The developed model considered a system with ten degrees of freedom:

- Three translational coordinates (x, y and z);
- Three attitude coordinates (ϕ, θ and ψ) described by the Roll-Pitch-Yaw convention;
- Two coordinates describing the orientation of the two tilting rotors (α_R and α_L);
- Two coordinates describing the position of the load with respect to the aircraft (γ_1 and γ_2).

Then, a state-space model affine in the inputs was derived to the system. This representation is used in the posterior chapters to design linear and nonlinear control laws for the Tilt-rotor UAV with suspended load.

The problem of designing linear controllers for the Tilt-rotor UAV with suspended load in the vicinity of its equilibrium point was initially studied. Two linear controllers were designed with use of LMIs solvers: a D-stable and a D-stable with disturbance rejection ($\mathcal{H}_\infty$). Both accomplished the task of path tracking but the $\mathcal{H}_\infty$ controller presented better performance on path tracking in presence of external disturbances. However, neither of them were able to stabilize the aircraft when it deviated too much from its equilibrium point.

Given that linear controllers present the drawback of working only in the vicinity of their equilibrium points, nonlinear control strategies for the system were developed to overcome that. These strategies relied on a cascade scheme with three input-output

feedback linearization blocks. Each one of these levels controls a different group of the system's state variables until the aircraft becomes fully stable:

1. The innermost level stabilizes the variables z, ϕ, α_R and α_L by actuating in f_R, f_L, τ_{α_R} and τ_{α_L};

2. The second level stabilizes θ and ψ by actuating in the references of α_R and α_L. With this and the first level, the attitude of the aircraft is fully stabilized;

3. The third level stabilizes x and y by actuating on the references of ϕ and θ.

Two path tracking controllers were specified for the strategy described above. The first one, the Nonlinear Controller for Path Tracking (NLPT), considers the load only as a disturbance and follows path references without actuating to avoid the load's swing. The second controller, the Nonlinear Controller for Reduced Load Swing (NLLS), seeks to find a compromise between path tracking and reducing the load's swing. Simulation results showed that NLLS widely reduced the load's swing with respect to NLPT at the cost of worsening its tracking error. However, even though NLPT performed better path tracking than NLLS, one should be careful when using it, since high swing of the load, specially for heavy loads, may destabilize the aircraft.

Simulation results comparing NLLS to the linear $\mathcal{H}_\infty$ controller were also carried out. As result, the NLLS was worse both on path tracking and disturbance rejection. This might be explained by the design simplification assumptions and the fact that the feedback linearization techniques does not guarantee robustness in the face of parameter uncertainties or disturbances. However, it might be possible that, with some careful tuning for the NLLS controller, it may present improved performance when compared to the results of the $\mathcal{H}_\infty$ approach. On the other hand, NLLS is able to maintain the system stable when it deviates from the equilibrium point, while it may destabilize if using the $\mathcal{H}_\infty$ controller. NLLS was shown to be better than linear controllers when there are complex paths, harsher disturbances or when the aircraft gets far from its reference.

As proof of concept, modifications on the previous approaches were introduced so that the Tilt-rotor is able to stabilize an inverted pendulum. Conceptually, it was possible to attain stabilization of the inverted pendulum, but one cannot expect results as good as the ones provided by Quad-rotor UAVs, due to their additional two rotors.

The control strategies presented in chapters 3 and 4 assume that all states of the Tilt-rotor UAV with suspended load are precisely known. However, this assumption may not be true when dealing with real experiments. Usually, built-in sensors provide noisy measurements with their own sampling frequency, which may be different from the controller's frequency. Problems like these are common in UAV applications, where the aircraft's position is measured using a GPS with these characteristics. This kind of problem may be circumvented by using state estimators.

Consequently, a situation where the aircraft's position is measured using a GPS equipment with sampling time T_s, while the controller has a sampling time τ_s, with $\tau_s < T_s$ is analysed. In this case, it is important to estimate the system's states while no new

measurements are available from the GPS, also taking into account its measurement uncertainty. To further increase the challenge, the estimator must consider that the aircraft's motion may be affected by disturbances (e.g. wind gusts), which are not usually measured. This was solved using the technique of Linear Kalman Filter with Unknown Inputs (LKFUI).

The state space model that was used for the LKFUI algorithm simplifies the dynamics of the Tilt-rotor by deriving its equations from Newton's second law. This way, there is no need to use a nonlinear state estimation algorithm. Simulation results were presented where the LKFUI algorithm was able to estimate the disturbances and use its information along with the aircraft translational forces so as to estimate the system's position while the GPS does not provide new measurements. At the moment that the GPS provides a new measure, a correction step is taken to merge the informations of the measurement along with the previous estimation. Consequently, the LKFUI was able to reduce estimation errors from some value higher than $\pm 1.5m$ (GPS's measurement error) to approximately $\pm 0.38\,m$ with 99.8% of confidence. Simulation results showed that, within most of the time, the aircraft's tracking error for position bounded between $\pm 0.30\,m$ away from the reference along the three cartesian axes.

6.1 Future Works

This section presents some possible directions of future researches as continuation of the present work.

- *Implementation of the proposed control strategies.* As commented in Chapter 1, a Tilt-rotor UAV is being developed by the ProVant project. Therefore, the next step is to implement the designed control strategies on the real aircraft.

- *Reduce attitude control of the Tilt-rotor UAV from two IOFL blocks into a single IOFL block.* This approach is being developed, in which the authors noticed that if some dynamic coupling are neglected on the system's model, then it would be possible to apply dynamic extension by integrating two times the inputs f_R and f_L. This way it would be possible to stabilize z, ϕ, θ, ψ, α_R and α_L using a single Input-Output Feedback Linearization control loop.

- *Study methodologies that provide path tracking for the load's motion.* The present work only tried to avoid the load from swinging, but in some cases it may be interesting to perform path tracking of the load.

- *Mathematically analyse the nonlinear design for the controller.* This work presented some simplifications that were applied when designing the nonlinear control strategy. However, the validity range of such simplifications within the space state were not mathematically studied. Therefore, an important work might be the evaluation of this issue.

- *Evaluate nonlinear state estimation methodologies.* For the sake of simplification, the present work used a simplified model of the system to estimate its position

along with time. However, an interesting future work would be to implement a nonlinear state estimation technique, such as the Unscented Kalman Filter with Unknown Inputs (Teixeira et al., 2008) and compare the results. It is likely that the estimations might provide better results but it might be computationally expensive, given that it would have to calculate forty sigma points using the nonlinear model of the system, instead of only one calculation used on the linear model.

- *Model the system using Newton-Euler formulation.* The present work used Euler-Lagrange formulation when deriving the equations of motion for the Tilt-rotor UAV. A problem that emerged was the inversion of the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$. If the Newton-Euler formulation is used, different methods can be used in order to obtain the state-space representation without the need for matrix inversion.

- *Further exploit linear strategies by using gain-scheduling approach.* An interesting suggestion of future work might be the use of gain-scheduling such as proposed by Lee et al. (2007) and Dickeson et al. (2007). This way, a group of interpolated linear controllers may stabilize the aircraft in different operation points, instead of a single one.

- *Study adaptive control methodologies for dealing with unknown masses.* This dissertation considered the load's mass to be known with bounded uncertainty of 30%. However, it can also use an adaptive approach where no assumptions are made about the mass of the suspended load. An adaptive controller may be designed so as to change the system's actuation while estimating the load's mass.

Bibliography

M. M. Almeida, R. Donadel, G. V. Raffo, and L. B. Becker. Full control of a tiltrotor uav for load transportation. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014a. To be published.

M. M. Almeida, L. Schreiber, and G. V. Raffo. Robust state estimation for uavs - a comparison study among a deterministic and a stochastic approach. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014b. To be published.

N. Amiri, A. Serrano, and R. Davies. Modelling of opposed lateral and longitudinal tilting dual-fan unmanned aerial vehicle. In *18th IFAC World Congress*, volume 28, pages 2054–2059, 2011.

N. Amiri, A. Ramirez-Serrano, and R. J. Davies. Integral backstepping control of an unconventional dual-fan unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1-4):147–159, 2013.

A. Bhanja Chowdhury, A. Kulhare, and G. Raina. Back-stepping control strategy for stabilization of a tilt-rotor uav. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 3475–3480, May 2012.

S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.

P. Castillo, R. Lozano, and A. Dzul. Stabilization of a Mini Rotorcraft with Four Rotors. *IEEE Control Systems Magazine*, 22(6):45–55, Dec. 2005a.

P. Castillo, R. Lozano, and A. Dzul. Stabilization of a mini rotorcraft with four rotors. *Control Systems, IEEE*, 25(6):45 – 55, dec 2005b. ISSN 1066-033X.

P. Castillo, R. Lozano, and A. E. Dzul. *Modelling and Control of Mini-Flying Machines*. Springer-Verlag, London, UK, 2005c.

M. Chilali and P. Gahinet. $\mathcal{H}_\infty$ design with pole placement constraints: an lmi approach. *Automatic Control, IEEE Transactions on*, 41, 1996.

S. Dai, T. Lee, and D. S. Bernstein. Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass. In *Proc. of 45th IEEE Conference on Decision and Control*, 2014. To be published.

M. Darouach, M. Zasadzinski, and M. Boutayeb. Extension of minimum variance estimation for systems with unknown inputs. *Automatica*, 39(5):867 – 876, 2003.

S. S. Dhaliwal and A. Ramirez-Serrano. Optimization architecture of a modular architecture for robotic control: Marc control structure applied to a vtol uav. In *Autonomic and Autonomous Systems, 2009. ICAS '09. Fifth International Conference on*, pages 238–244, April 2009.

J. Dickeson, D. Miles, O. Cifdaloz, V. Wells, and A. Rodriguez. Robust lpv $\mathcal{H}_\infty$ gain-scheduled hover-to-cruise conversion for a tilt-wing rotorcraft in the presence of cg variations. In *46th IEEE Conference on Decision and Control*, pages 2773–2778, Dec 2007.

R. Donadel, M. M. Almeida, G. V. Raffo, and L. B. Becker. Path tracking control of a small scale tiltrotor unmmaned aerial vehicle. In *Proc. of XXth Congresso Brasileiro de Automatica, CBA 2014*, 2014a. To be published.

R. Donadel, G. V. Raffo, and L. B. Becker. Modeling and control of a tiltrotor uav for path tracking. *19th World Congress, IFAC*, 2014b. To be published.

G. E. Dullerud and F. G. Paganini. *A course in robust control theory : a convex approach*. Texts in applied mathematics. Springer, New York, 2000.

A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia. Learning swing-free trajectories for uavs with a suspended load. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4902–4909. IEEE, 2013.

P. Gahinet and A. Nemirovskii. LMI Lab: A Package for Manipulating and Solving LMIs. In *INRIA*, 1993.

F. Gonçalves, J. Bodanese, R. Donadel, G. Raffo, J. Normey-Rico, and L. Becker. Small scale uav with birotor configuration. *Procedings of the 2013 International Conference on Unmanned Aircraft Systems*, 2013.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, Mar. 2014.

G. R. Gress. Using dual propellers as gyroscopes for tilt-prop hover control. In *Proc. AIAA Biennial Int. Powered Lift Conf. Exhibit, Williamsburg, VA*, 2002.

M. Hautus. Stabilization controllability and observability of linear autonomous systems. *Indagationes Mathematicae (Proceedings)*, 73(0):448 – 455, 1970. ISSN 1385-7258.

J. K. Hedrick and A. Girard. Feedback linearization, 2005.

M. Hehn and R. D'Andrea. A flying inverted pendulum. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 763–770. IEEE, 2011.

M. Johnson and M. Grimble. Recent trends in linear optimal quadratic multivariable control system design. In *Control Theory and Applications, IEE Proceedings D*, pages 53–71. IET, 1987.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Y.-s. Kang, B.-J. Park, A. Cho, C.-S. Yoo, and S.-w. Choi. Envelop expansion flight test of flight control systems for tr-60 tilt-rotor uav. In *13th International Conference on Control, Automation and Systems (ICCAS)*, pages 1866–1871. IEEE, 2013.

F. Kendoul, I. Fantoni, and R. Lozano. Modeling and control of a small autonomous aircraft having two tilting rotors. In *44th IEEE European Control Conference on Decision and Control*, pages 8144–8149, Dec 2005.

D. Kinder and G. Whitcraft. Design and development of a tilt-wing aircraft. In *AIAA Student Conference, San Diego, CA*, 2000.

K. Kleinhesselink. *Stability and Control Modeling of Tiltrotor Aircraft*. University of Maryland, College Park, 2007.

Y. Labit and D. Peaucelle, D. ; Henrion. Sedumi interface 1.02: a tool for solving lmi problems with sedumi. In *Proceedings, IEEE International Symposium on Computer Aided Control System Design*, pages 272–277, 2002.

B. P. Lathi. *Linear systems and signals*. Oxford University Press, 2009.

J.-H. Lee, B.-M. Min, and E.-T. Kim. Autopilot design of tilt-rotor uav using particle swarm optimization method. In *International Conference on Control, Automation and Systems*, pages 1629–1633. IEEE, 2007.

S.-G. Lee, V.-H. Dang, S. Moon, B. Kim, et al. Partial feedback linearization control of a three-dimensional overhead crane. *International Journal of Control, Automation and Systems*, 11(4):718–727, 2013.

J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *CCA/ISIC/CACSD*, Sept. 2004.

A. M. Lyapunov. The general problem of motion stability. *Kharkovskoye Matematicheskoe Obshchestvo*, 11, 1892.

M. Maisel, D. Giulianetti, D. Dugan, U. S. N. Aeronautics, and S. A. H. Office. *The history of the XV-15 tilt rotor research aircraft: from concept to flight*. NASA history series. National Aeronautics and Space Administration, Office of Policy and Plans, NASA History Division, 2000.

D. Mix and D. Seitz. Dynamics of control system design for a tilt-wing vehicle. In *AIAA Student Conference, San Diego, CA*, 2000.

S.-C. Moon, D.-H. Kim, S.-G. Lee, et al. Adaptive sliding mode control of three dimensional overhead cranes. In *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems*, pages 354–359. IEEE, 2012.

K. Ogata. *Modern Control Engineering*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 4th edition, 2001. ISBN 0130609072.

R. Olfati-Saber. *Nonlinear Control of Underactuated Mechanical Systems with Application to Robotics and Aerospace Vehicles*. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2001.

C. Papachristos, K. Alexis, G. Nikolakopoulos, and A. Tzes. Model predictive attitude control of an unmanned tilt-rotor aircraft. In *Industrial Electronics (ISIE), 2011 IEEE International Symposium on*, pages 922–927. IEEE, 2011a.

C. Papachristos, K. Alexis, and A. Tzes. Design and experimental attitude control of an unmanned tilt-rotor aerial vehicle. In *15th International Conference on Advanced Robotics*, pages 465–470. IEEE, 2011b.

G. V. Raffo. *Robust Control Strategies for a Quadrotor Helicopter. An underactuated mechanical system.* Universidad de Sevilla, Escuela Tecnica Superior de Ingenieria, 2011.

G. V. Raffo, M. G. Ortega, and F. R. Rubio. An integral predictive/nonlinear $\mathcal{H}_\infty$ control structure for a quadrotor helicopter. *Automatica*, 46:29–39, 2010.

W. Respondek and I. A. Tall. Nonlinearizable single-input control systems do not admit stationary symmetries. *Systems & control letters*, 46(1):1–16, 2002.

A. Sanchez, J. Escareno, O. Garcia, and R. Lozano. Autonomous hovering of a noncyclic tiltrotor uav: Modeling, control and implementation. In *Proc. of the 17th IFAC Wold Congress*, pages 803–808, 2008.

A. A. Shabana. *Dynamics of multibody systems.* Cambridge university press, 2013.

S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design.* John Wiley & Sons, 2005. ISBN 0470011688.

J.-J. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, Oct. 1991. ISBN 0130408905.

M. W. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control.* John Wiley and Sons, 2005.

K. Sreenath, N. Michael, and V. Kumar. Trajectory generation and control of a quadrotor with a cable-suspended load-a differentially-flat hybrid system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4888–4895. IEEE, 2013.

B. Teixeira, J. Chandrasekar, H. Palanthandalam-Madapusi, L. Torres, L. Aguirre, and D. Bernstein. Gain-constrained kalman filtering for linear and nonlinear systems. *Trans. Sig. Proc.*, 56(9), Sept. 2008.

A. Trofino, D. Coutinho, and K. A. Barbosa. Sistemas multivariaveis: Uma aborgadem via lmis (versão preliminar), Aug. 2003.

S. Yanguo and W. Huanjin. Design of flight control system for a small unmanned tilt rotor aircraft. *Chinese Journal of Aeronautics*, 22(3):250–256, 2009.

Appendix A

Theory on Robotics

A.1 Rotation Matrices

A point "p" in a three-dimensional space can be represented in relation with a coordinate frame "a" using three parameters. These parameters represent the projections of the vector which goes from the origin $\boldsymbol{o}^a$ to point p. From now on the notation $\boldsymbol{p}^a \in \Re^3$ is adopted to denote that the point p is a three-dimensional vector represented in respect with frame a (Spong et al., 2005).

In order to represent this same point $\boldsymbol{p}^a$ with respect to a different frame b, whose origin $\boldsymbol{o}^b$ is at the same place of $\boldsymbol{o}^a$, but rotated with respect to frame a, the transformation of a frame to another can be given by:

$$\boldsymbol{p}^b = \boldsymbol{R}_a^b \, \boldsymbol{p}^a, \tag{A.1}$$

where $\boldsymbol{R}_a^b$ is a rotation matrix, which carries information on the projections of frame a into b. Three-dimensional rotation matrices belongs to a group called **Special Orthogonal** of dimension three, or simply $SO(3)$. A rotation matrix $\boldsymbol{R} \in SO(3)$ holds some very important properties, as shown below:

1. $\boldsymbol{R}^T = \boldsymbol{R}^{-1} \in SO(3)$;
2. The columns (and also the rows) of $\boldsymbol{R}$ are mutually orthogonal;
3. Each column (and also each row) of $\boldsymbol{R}$ is a unit vector;
4. $det(\boldsymbol{R}) = 1$.

Properties 2 and 4 have an importance of telling us that there is always an inverse $\boldsymbol{R}^{-1}$ of a rotation matrix and that its inverse is also a rotation matrix. The property 1 goes even further stating how easy it is to obtain the inverse of this matrix. Moreover, these properties altogether are very useful when deriving many concepts in robotics.

Therefore, given (A.1) and property 1 it is possible to say that:

$$\boldsymbol{p}^a = (\boldsymbol{R}_a^b)^{-1} \, \boldsymbol{p}^b = (\boldsymbol{R}_a^b)^T \, \boldsymbol{p}^b = \boldsymbol{R}_b^a \, \boldsymbol{p}^b. \tag{A.2}$$

Some convenient rotation matrices in robotics are the ones for rotations around the axes x, y and z. For example, the rotation of θ degrees around the axis x of a frame can be denoted as $\boldsymbol{R}_{x,\theta}$ and given by:

$$\boldsymbol{R}_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix}, \tag{A.3}$$

where c_θ and s_θ are, respectively, shorter representations for $\cos(\theta)$ and $\sin(\theta)$.

Similarly, rotation matrices $\boldsymbol{R}_{y,\theta}$ and $\boldsymbol{R}_{z,\theta}$ are given by:

$$\boldsymbol{R}_{z,\theta} = \begin{bmatrix} c_\theta & -s_\theta & 0 \\ s_\theta & c_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{R}_{y,\theta} = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}. \tag{A.4}$$

Given a coordinate system a, which is rotated once according to $\boldsymbol{R}_a^b$ obtaining a new axis b and then coordinate system b is rotated according to $\boldsymbol{R}_b^c$ in order to obtain axis c, it is possible to obtain a single rotation matrix that transforms from a to c given by the following relation:

$$\boldsymbol{R}_a^c = \boldsymbol{R}_b^c \, \boldsymbol{R}_a^b. \tag{A.5}$$

The relation shown on (A.5) is known as the composition law for rotational transformations. It states that consecutive rotations can be represented by a single rotation matrix whose terms are equal to the successive multiplication of the individual rotation matrices.

An intuitive property that can be proven using (A.5) is that two consecutive rotations around a same arbitrary axis results in a single rotation around this axis, but with the rotation angles added up. For example, a rotation $\boldsymbol{R}_{z,\theta}$ followed by a rotation $\boldsymbol{R}_{z,\phi}$ is given by $\boldsymbol{R}_{z,\theta} \, \boldsymbol{R}_{z,\phi} = \boldsymbol{R}_{z,\theta+\phi}$.

Another important concept in robotics is that, given an initial frame a, it is always possible to rotate it in order to obtain any final frame b only by combining at most three consecutive rotations around the axes x, y and z. There are 12 possible combinations of rotations of this kind that can result on the same final rotated frame b:

$$\begin{gathered} X-Y-Z, \quad X-Z-Y, \quad X-Y-X, \quad X-Z-X, \quad Z-Y-X, \quad Z-X-Y, \\ Z-Y-Z, \quad Z-X-Z, \quad Y-X-Z, \quad Y-Z-X, \quad Y-X-Y, \quad Y-Z-Y. \end{gathered} \tag{A.6}$$

A combination shown at (A.6) that is very commonly used in aeronautical research is the $Z-Y-X$, also known as the *Roll-Pitch-Yaw* convention. The resulting transformation

matrix is given by:

$$\begin{aligned}\boldsymbol{R} &= \boldsymbol{R}_{z,\psi}\,\boldsymbol{R}_{y,\theta}\,\boldsymbol{R}_{x,\phi} \\ &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \\ &= \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}. \end{aligned} \tag{A.7}$$

A.2 Skew Symmetric Matrices

This section presents skew symmetric matrices and its properties. This group of matrices holds some properties that are very useful when deriving the equations of motion of the Tilt-Rotor UAV (Spong et al., 2005).

An $n \times n$ matrix $\boldsymbol{S}$ is said to be **skew symmetric** if and only if:

$$\boldsymbol{S}^T + \boldsymbol{S} = \boldsymbol{0}_{n\times n}. \tag{A.8}$$

This means that, for $i, j = 1, 2, ..., n$, the following is true:

$$s_{ij} + s_{ji} = 0, \tag{A.9}$$

where s_{ij} is the term situated at the i^{th} row and j^{th} column of $\boldsymbol{S}$.

It is then easy to see that:

$$s_{ij} = 0 \quad \text{, if } i = j \tag{A.10}$$

$$s_{ij} = -s_{ji}, \text{ if } i \neq j. \tag{A.11}$$

Focusing only on 3×3 skew symmetric matrices, $\boldsymbol{S}$ has the form:

$$\boldsymbol{S} = \begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}. \tag{A.12}$$

As definition, the skew symmetric matrix related to a three-dimensional vector $\boldsymbol{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ is denoted as $\boldsymbol{S}(\boldsymbol{p})$ and given by:

$$\boldsymbol{S}(\boldsymbol{p}) = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix}. \tag{A.13}$$

Thus, given this preceding definition, the following properties hold:

1. For any three-dimensional vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ and scalars a and b:

$$\boldsymbol{S}(a\boldsymbol{p} + b\boldsymbol{q}) = a\boldsymbol{S}(\boldsymbol{p}) + b\boldsymbol{S}(\boldsymbol{q}). \tag{A.14}$$

2. For any three-dimensional vectors $\boldsymbol{p}$ and $\boldsymbol{q}$:

$$\boldsymbol{S}(\boldsymbol{p})\boldsymbol{q} = \boldsymbol{S}(\boldsymbol{q})^T\boldsymbol{p} = \boldsymbol{p} \times \boldsymbol{q}. \tag{A.15}$$

   where $\boldsymbol{p} \times \boldsymbol{q}$ denotes the vector cross product.

3. For $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{p} \in \Re^3$:

$$\boldsymbol{R}\boldsymbol{S}(\boldsymbol{p})\boldsymbol{R}^T = \boldsymbol{S}(\boldsymbol{R}\boldsymbol{p}). \tag{A.16}$$

4. For any $\boldsymbol{p}, \boldsymbol{q} \in \Re^3$:

$$\boldsymbol{p}^T\boldsymbol{S}(\boldsymbol{q})\boldsymbol{p} = 0. \tag{A.17}$$

5. For a rotation matrix that is function of time $\boldsymbol{R}_B^A(t) \in SO(3)$:

$$\dot{\boldsymbol{R}}_B^A(t) = \boldsymbol{R}_B^A(t)\, \boldsymbol{S}(\boldsymbol{\omega}_{BA}^B(t)). \tag{A.18}$$

   where $\boldsymbol{\omega}_{BA}^B(t) \in \Re^3$ is the angular velocity of frame B with respect to frame A represented in frame B.

A.3 Euler-Lagrange Equations

The Euler-Lagrange equations are used in order to obtain the dynamic equations of motion of a system. Defining $\boldsymbol{q} \in \Re^n, \boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & \dots & q_n \end{bmatrix}^T$ as the vector containing the system's n generalized coordinates, the equations of motion of a dynamic system can be described by:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{F}_{ext}. \tag{A.19}$$

where $\boldsymbol{M}(\boldsymbol{q}) \in \Re^{n \times n}$ is called the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \Re^{n \times n}$ is the Coriolis and centrifugal forces matrix, $\boldsymbol{G}(\boldsymbol{q}) \in \Re^n$ is the gravitational force vector, $\boldsymbol{F}(\boldsymbol{q}) \in \Re^n$ is the independent input force vector and $\boldsymbol{F}_{ext} \in \Re^n$ represents external disturbances on the system.

The inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ can be obtained by calculating the system's kinetic energy and expressing it in the form:

$$K = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{A.20}$$

The Coriolis and centrifugal forces matrix can be obtained from the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ using Christoffel symbols of first kind. This way, the $(k,j)^{th}$ element of the matrix $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ is defined as:

$$c_{kj} = \sum_{i=1}^{n} \frac{1}{2}\left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right]\dot{q}_i, \tag{A.21}$$

where m_{kj} is the $(k,j)^{th}$ element of $\boldsymbol{M}(\boldsymbol{q})$.

The gravitational force vector $\boldsymbol{G}(\boldsymbol{q})$ is calculated as follows:

$$\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial P}{\partial \boldsymbol{q}}. \tag{A.22}$$

where P is the potential energy of the system.

The form of the independent input force vector $\boldsymbol{F}(\boldsymbol{q})$ is shaped depending on how the inputs of the studied system actuate over the generalized coordinates. The same applies on the external disturbances force vector $\boldsymbol{F}_{ext}$.