

Richard Alfonso Andrade Alfaro

**PREDICTIVE CONTROL STRATEGIES FOR
UNMANNED AERIAL VEHICLES IN CARGO
TRANSPORTATION TASKS**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.

Orientador: Prof. Dr. Julio Elias Normey Rico - PGEAS - UFSC

Co-orientador: Prof. Dr. Guilherme Vianna Raffo - PPGEE - UFMG

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Alfaro, Richard Alfonso Andrade
Predictive control strategies for unmanned aerial
vehicles in cargo transportation tasks / Richard Alfonso
Andrade Alfaro ; orientador, Julio Normey-Rico ;
coorientador, Guilherme Raffo. - Florianópolis, SC, 2016.
145 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Engenharia de Automação e Sistemas.

Inclui referências

1. Engenharia de Automação e Sistemas. 2. Controle
preditivo baseado em modelo. 3. Tilt-rotor. 4. Veículo
aéreo não tripulado. 5. Transporte de carga. I. Normey
Rico, Julio. II. Raffo, Guilherme. III. Universidade
Federal de Santa Catarina. Programa de Pós-Graduação em
Engenharia de Automação e Sistemas. IV. Título.

Richard Alfonso Andrade Alfaro

**PREDICTIVE CONTROL STRATEGIES FOR
UNMANNED AERIAL VEHICLES IN CARGO
TRANSPORTATION TASKS**

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Engenharia de Automação e Sistemas”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

Florianópolis, 28 de março 2016.

Prof. Dr. Rômulo Silva de Oliveira
PGEAS - UFSC

Coordenador do Programa de Pós-Graduação em Engenharia de
Automação e Sistemas

Prof. Dr. Julio Elias Normey Rico - PGEAS - UFSC
Orientador

Prof. Dr. Guilherme Vianna Raffo - PPGEE - UFMG
Co-orientador

Banca Examinadora:

Prof. Julio Elias Normey Rico, Dr.
PGEAS - UFSC
Presidente

Prof. Antonio Ferramosca, Dr.
FRRq-UTN

Prof. Leandro Buss Becker, Dr.
PGEAS - UFSC

Prof. Douglas Wildgrube Bertol, Dr.
DEE-UDESC

To my caring parents, Fabian and
Fanny, and to my love, Nathaly.

ACKNOWLEDGMENTS

I thank my parents for their support throughout my life, who have accompanied me in my victories and my defeats, who have always encouraged me to continue and fulfill my dreams and desires. Them with their daily efforts and teachings have allowed me to grow as a person and as a professional. I give special thanks to Nathaly who has led me do things I never imagined. Thank you for being by my side while developing this project and for helping me.

I would also like to thank my advisors, Julio Normey Rico, and Guilherme Vianna Raffo, for the opportunity given to me and for supporting me with great disposition throughout my thesis. I appreciate all you have done for me.

I am grateful for all the support from the members of Project ProVANT: Patrick, Fernando, Davinder, Gabriel.

I like to thank The Agreement-Graduate Student Program (PEC-PG) of the Brazilian research agencies Capes/CNPq for their support in this dissertation.

The mind, once stretched by a new idea,
never returns to its original dimensions.
(Ralph Waldo Emerson)

RESUMO

O desenvolvimento de veículos aéreos não tripulados (VANTs) vem despertando um grande interesse tanto no meio acadêmico quanto na indústria nas últimas décadas. Muitos campos da robótica e da teoria de controle vem sendo explorados visando melhorar o desempenho destes sistemas. Existem vários cenários onde estas aeronaves são utilizadas, tais como monitoramento de ambientes, agricultura de precisão, busca e resgate, entre outras. Dentre as diferentes aplicações destas aeronaves temos o transporte de carga suspensa por cabo, o qual tem promovido várias pesquisas relacionadas com transporte de alimentos, medicamentos e suprimentos em geral, para zonas de risco. Neste sentido, este trabalho tem como foco o uso de VANTs em tarefas de transporte de carga, considerando perturbações externas e incertezas paramétricas. A aeronave utilizada é um birotor na configuração Tilt-rotor que carrega uma carga suspensa. Um Tilt-rotor é um veículo movimentado por dois rotores inclináveis, os quais geram e direcionam forças de impulso para sustentar a aeronave. Neste estudo, é importante que a aeronave seja capaz de seguir uma trajetória predefinida enquanto estabiliza a carga suspensa mesmo quando afetada por perturbações externas ou incertezas paramétricas. Além disso, um modelo não linear multicorpo é obtido via formulação Euler-Lagrange para o VANT Tilt-rotor considerando a carga suspensa. Neste modelo foi considerado que a aeronave é composta por quatro corpos rígidos e tem dez graus de liberdade. O problema de controle é solucionado com um controlador preditivo (MPC) incremental e um não incremental, baseados no modelo linear do erro do sistema, o qual é linearizado em torno a uma trajetória genérica. Além disso, os MPCs consideram custo terminal, com o objetivo de garantir estabilidade e por consequência reduzir o horizonte de predição. Devido ao fato do sistema linear ser variante no tempo (LVT), o custo terminal é calculado mediante desigualdades matriciais lineares (LMI). Por outro lado, restrições são impostas na formulação do MPC, relacionadas com as limitações físicas dos atuadores e considerando que o VANT está confinado numa área específica. Finalmente, simulações foram realizadas para avaliar o desempenho dos controladores propostos, considerando perturbações constantes em diferentes instantes de tempo, e levando em conta incertezas paramétricas.

Palavras-chave: VANT, Tilt-rotor, Carga Suspensa, Transporte de

Carga, MPC, Custo Terminal, Seguimento de Trajetória

ABSTRACT

The development of unmanned aerial vehicles (UAVs) has aroused great interest in both academia and industry in the recent decades. Many areas of robotics and control theory have been exploited to improve the performance of these systems. There are several scenarios where these aerial vehicles are used, like monitoring environment, precision agriculture, construction, search and rescue. Transportation of cable-suspended loads with UAVs is another application. This has promoted research related to load transportation of food, medicine, and supplies in general for unsafe areas. This research is focused on this topic, where it is necessary that the UAV follows a predefined trajectory while stabilizing the suspended load, even if it is affected by external disturbances. In this dissertation, two model predictive controllers (MPCs) are used to solve the path tracking problem of a small scale Tilt-rotor Unmanned Aerial Vehicle (UAV) while carrying a suspended load. A Tilt-rotor is a vehicle lifted and propelled by two tiltable rotors, in order to control the direction of thrust forces. In the present study, it is important that the aircraft able to follow a predefined trajectory while maintaining the suspended load stable even in the presence of external disturbances and parametric uncertainties. Moreover, a rigorous multibody non-linear dynamic model is obtained via Euler-Lagrange formulation for the Tilt-rotor UAV with suspended load, assuming four rigid bodies and ten degrees of freedom (DOF) of the vehicle. The control problem is solved with incremental and non-incremental model predictive controllers, based on the linear error model of the system, which is linearized around a generic trajectory. Furthermore, the MPCs consider a terminal cost in order to ensure stability, allowing the prediction horizon reduction. As the linear model is a linear time-varying (LTV) system, the terminal cost is calculated via linear matrix inequalities (LMI). In addition, some constraints are imposed on the formulation, related to physical limitations of the actuators and assuming that the aircraft is confined to a particular area. Finally, numerical simulations are performed in order to evaluate the controllers, considering constant disturbances at different instants of time, and modeling errors.

Keywords: UAV, Tilt-rotor, Suspended Load, Cargo Transportation, MPC, Terminal Cost, Path Tracking

LIST OF FIGURES

Figure 1	Bell-Boeing V-22 Osprey in cargo transportation task . .	2
Figure 2	Bell Eagle Eye TiltRotor UAV (UAVGLOBAL, 2008)	2
Figure 3	(a) TiltRotor UAV developed by KARI (SUNG-KI, 2011) ; (b) Tilt-rotor Panther UAV (OPLI, 2012); (c) American Dynamics AD-150 UAV (GINGICHASHVILI, 2009)	3
Figure 4	Tilt-rotor flight modes (HOWARD, 2014; OSBORN, 2015)	5
Figure 5	Tilt-rotor aircraft with suspended load	11
Figure 6	Tilt-rotor UAV	13
Figure 7	Simulation structure	60
Figure 8	Reference trajectory of the Tilt-rotor with suspended load	61
Figure 9	Reference trajectory of the Tilt-rotor with suspended load made by I-SSMPC by LQR controller	62
Figure 10	Time evolution of translational and rotational positions performed by the Tilt-rotor UAV applying the proposed I-SSMPC and LQR controller	63
Figure 11	Time evolution of α_r and α_l performed by Tilt-rotor applying the proposed I-SSMPC and LQR controller	64
Figure 12	Time evolution of γ_1 and γ_2 , performed by Tilt-rotor applying the proposed I-SSMPC and LQR controller	65
Figure 13	Time evolution of thrust forces and torques applied to Tilt-rotor UAV by the proposed I-SSMPC and LQR controller . . .	66
Figure 14	Reference Trajectory of the Tilt-rotor with suspended load made by I-SSMPC and LQR controller	67
Figure 15	Time evolution of translational and rotational positions performed by the Tilt-rotor UAV applying the proposed NI-SSMPC and LQR controller	68
Figure 16	Time evolution of rotors and suspended load angular position, performed by the Tilt-rotor UAV applying the proposed NI-SSMPC and LQR controller	69
Figure 17	Time evolution of thrust forces and torques applied to the Tilt-rotor UAV by the proposed NI-SSMPC and LQR controller . .	70
Figure 18	Time response of translational and rotational positions with parametric uncertainties in Tilt-rotor model	72

Figure 19 Time response of rotors and suspended load angular position with parametric uncertainties in Tilt-rotor model. 73

Figure 20 Control signal of thrust forces and torques applied to Tilt-rotor UAV with parametric uncertainties. 74

Figure 21 Tilt-rotor UAV second prototype assembly 78

Figure 22 Tilt-rotor UAV general description. 79

Figure 23 Electronic devices attached to the Tilt-rotor 80

Figure 24 AADL representation of the software on the Sensor and Actuator Processing subsystem. 81

Figure 25 Flowchart of DataProcessing execution thread 83

Figure 26 Curve of behavioral for the AXI 2826/12 GOLD motor 85

Figure 27 Flowchart of Communication execution thread. 87

Figure 28 AADL representation of the software on the Control and Estimation subsystem. 89

Figure 29 Flowchart of CommLowLevel execution thread. 90

Figure 30 Flowchart of DistcreteControl execution thread. 92

Figure 31 Flowchart of MPC calculation 94

Figure 32 Structure of the test bench for the CMUCAM2 camera 96

LIST OF TABLES

Table 1	Tilt-rotor system parameters	26
Table 2	Tilt-rotor equilibrium point.....	26
Table 3	Output constraints applied to Tilt-rotor with suspended load.....	57
Table 4	Disturbance parameters	60
Table 5	TV index performance analysis	75
Table 6	MSE index performance analysis.....	75
Table 7	Single board computers specifications.....	80
Table 8	Setup configuration of MPU6050 sensor.....	84

LIST OF ACRONYMS

DAS	Automation and Systems Department
DELT	Electronic Engineering Department
I-SSPMC	Incremental MPC Based on State Space System
LMI	Linear Matrix Inequality.
LTV	Linear Time Varying
LTI	Linear Time Invariant
MPC	Model Predictive Control
MSE	Mean Square Error
NI-SSMPC	Non-Incremental MPC based on State Space System
PVTOL	Planar Vertical Take-Off and Landing
QP	Quadratic Programming
TV	Total Variation
UAV	Unnamed Arial Vehicles
UFSC	Federal University of Santa Catarina
UFMG	Federal University of Minas Gerais
VTOL	Vertical Take-Off and Landing

LIST OF SYMBOLS

Notation

- a Italic lower case letters denote scalars.
 \mathbf{a} Boldface italic lower case letters denote vectors.
 \mathbf{A} Boldface italic upper case letters denote matrices.

Symbols

- t Continuous time.
 k Discrete time.
 $\dot{\mathbf{a}}$ Time-derivative of vector \mathbf{a} .
 $\hat{\mathbf{a}}$ Prediction of vector \mathbf{a} .
 $\bar{\mathbf{a}}$ Augmented vector \mathbf{a} .
 $\tilde{\mathbf{a}}$ Error vector of the variable \mathbf{a} , $\tilde{\mathbf{a}} = \mathbf{a} - \mathbf{a}_r$.
 \mathbf{a}^T Transpose of vector \mathbf{a} .
 \mathbf{A}^{-1} Inverse operator of matrix \mathbf{A} .
 \mathbf{A}^+ Pseudo inverse operator of matrix \mathbf{A} .
 $\mathbf{S}(\cdot)$ Skew symmetric matrix.
 \mathbf{I}_n Identity matrix of size n .
 $\mathbf{0}_{n \times m}$ Zero matrix of n row and m columns.

Model Notation

- \mathcal{I} Fixed inertial frame.
 \mathcal{B} Moving body frame.
 \mathcal{C}_i Frame rigidly attached to i^{th} body's center of mass.
 \mathbf{q} Generalized coordinate vector.
 $\boldsymbol{\xi}$ Position of the origin of \mathcal{B} w.r.t \mathcal{I} .
 $\boldsymbol{\eta}$ Orientation of main body w.r.t \mathcal{I} .
 ϕ Roll angle.
 θ Pitch angle.
 ψ Yaw angle.
 α_r, α_l Rotation angles of the right and left rotors, respectively.
 γ_1, γ_2 Rotation angles of the suspended load.
 β Fixed rotation of servomotors around $x_{\mathcal{C}_i}$ for $i = 2, 3$.

- $\mathbf{d}_{C_i}^{\mathcal{B}}$ Translation between the origin of frame \mathcal{B} and C_i for $i = 1, \dots, 4$.
- $\mathbf{p}_i^{C_i}$ Point rigidly attached to reference frame C_i for $i = 1, \dots, 4$.
- $\mathbf{R}_{C_i}^{\mathcal{B}}$ Rotation matrix of frame C_i to \mathcal{B} for $i = 1, \dots, 4$.
- $\mathbf{R}_{\mathcal{B}}^{\mathcal{I}}$ Rotation matrix of frame \mathcal{B} to \mathcal{I} .
- $\mathbf{M}(\mathbf{q})$ Inertia matrix.
- $\mathbf{C}(\dot{\mathbf{q}}, \mathbf{q})$ Coriolis and centripetal matrix.
- $\mathbf{G}(\mathbf{q})$ Gravitational force vector.
- $\mathbf{F}(\mathbf{q})$ Generalized force/torque vector.
- \mathbf{F}_{ext} External disturbance force vector.
- \mathbf{F}_{drag} Drag force vector.
- $\boldsymbol{\mu}$ Viscous coefficient matrix.
- K Kinetic energy.
- $\mathbf{v}_i^{\mathcal{I}}$ Velocity of the point i expressed in frame \mathcal{I} .
- \mathbf{I}_i Inertia tensor of the body i .
- \mathbf{J}_i Inertia tensor of the body i for a rotation around an axis displaced \mathbf{d}_i .
- $\boldsymbol{\omega}_{\mathcal{B}\mathcal{A}}^{\mathcal{B}}$ Angular velocity of frame \mathcal{B} with respect to \mathcal{A} represented in frame \mathcal{B} .
- \mathcal{P} Potential energy of whole system.
- ρ_i Mass density of body i .
- \mathbf{g} Gravity Vector.
- \mathbf{T} Thrust force vector.
- $\boldsymbol{\tau}$ Rotational torque vector.
- f_r Force generated by right propeller.
- f_l Force generated by left propeller.
- τ_r Torque generated by right servomotor.
- τ_l Torque generated by left servomotor.
- \mathbf{x}_s State space vector.
- \mathbf{x}_r State space reference vector.
- $\mathbf{A}(t)$ Linear state matrix.
- \mathbf{B} Linear input matrix.
- \mathbf{C} Linear output matrix.
- $\tilde{\mathbf{x}}_s$ Error state vector.
- $\tilde{\mathbf{u}}$ Error control vector.

$\mathbf{A}_a(t)$	Linear state augmented matrix.
\mathbf{B}_a	Linear input augmented matrix.
$\tilde{\mathbf{x}}_s$	Error state augmented vector.
$\mathbf{A}_z(k)$	Discrete time linear state matrix.
\mathbf{B}_z	Discrete time linear input matrix.
t_s	Sample time.

Control Notation

N	Prediction horizon.
M	Control horizon.
$\Delta \mathbf{u}(k)$	Increment of control signal.
\mathbf{P}	Predicted states matrix.
\mathbf{Q}	Predicted input matrix.
$\hat{\mathbf{x}}$	Predicted output vector.
$\hat{\mathbf{u}}$	Predicted control vector.
$\hat{\mathbf{u}}_r$	Future control reference vector.
\mathbf{W}_y	Output weighting matrix.
\mathbf{W}_u	Input weighting matrix.
Σ_ρ	Diagonal matrix of outputs weight.
Σ_λ	Diagonal matrix of inputs weight.
$\hat{\mathbf{x}}_r$	Future predefined reference trajectory vector.
$\mathbf{A}_r, \mathbf{b}_r$	Constraints matrices.
\mathbf{T}_m	Lower triangular matrix of entities.
\mathbf{I}_m	Diagonal matrix of entities.
$\hat{\mathbf{u}}_{max}$	Vectors with M copies of maximum values.
$\hat{\mathbf{u}}_{min}$	Vectors with M copies of minimum values.
$\bar{\mathbf{u}}(k-1)$	Vectors with M copies of previous control signal.
$\hat{\mathbf{x}}_{min}$	Vectors with N copies of minimum states values.
$\hat{\mathbf{x}}_{max}$	Vectors with N copies of maximum states values.
\mathbf{C}_z	Output weight matrix of LMI.
\mathbf{D}_z	Input weight matrix of LMI.
\mathbf{L}	Terminal value which is a Lyapunov matrix.
$\hat{\hat{\mathbf{u}}}_{max}$	Vectors with M copies of maximum error control values.
$\hat{\hat{\mathbf{u}}}_{min}$	Vectors with M copies of minimum error control values.

$\tilde{\mathbf{u}}(k-1)$ Vectors with M copies of previous error control signal.

$\hat{\mathbf{x}}_{min}$ Vectors with N copies of minimum error states values.

$\hat{\mathbf{x}}_{max}$ Vectors with N copies of maximum error states values.

CONTENTS

1	INTRODUCTION	1
1.1	MOTIVATION	3
1.2	STATE OF THE ART	6
1.2.1	UAV For Load Transportation	6
1.2.2	Model Predictive Control	7
1.3	OBJECTIVES	8
1.4	OUTLINE	9
1.5	LIST OF PUBLICATIONS	10
2	TILT-ROTOR UAV WITH SUSPENDED LOAD DYNAMIC MODEL	11
2.1	GENERALIZED COORDINATES	12
2.2	FORWARD KINEMATICS FORMULATION OF THE TILT-ROTOR UAV	14
2.3	EQUATION OF MOTION OF THE TILT-ROTOR UAV WITH SUSPENDED LOAD	15
2.3.1	Inertia Matrix of the Tilt-rotor UAV with Suspended Load	16
2.3.2	Coriolis and Centripetal Matrix	20
2.3.3	Gravity Forces Vector	21
2.3.4	External Forces Vector	22
2.4	FRICITION FORCE VECTOR	24
2.5	STATE-SPACE REPRESENTATION OF THE SYSTEM	25
2.6	SYSTEM DESIGN PARAMETERS	25
2.7	LINEARIZATION OF THE TILT-ROTOR UAV WITH SUSPENDED LOAD MODEL	27
2.8	SUMMARY	29
3	LINEAR TIME VARYING MODEL PREDICTIVE CONTROL	31
3.1	INCREMENTAL MODEL PREDICTIVE CONTROL (I-SSMPC)	31
3.1.1	Linear Prediction Model	31
3.1.2	Objective Function	33
3.1.3	Computing the control law	35
3.1.3.1	Quadratic Programing Formulation	35
3.1.3.2	Unconstrained Problem	37
3.2	NON INCREMENTAL MODEL PREDICTIVE CONTROL (NI-SSMPC)	38

3.2.1	Linear Predictive Model	38
3.2.2	Objective Function	38
3.2.3	Computing the control law	39
3.2.3.1	Quadratic Programming Formulation	39
3.2.3.2	Unconstrained Formulation	40
3.3	TERMINAL COST	41
3.3.1	Terminal Stage	41
3.4	SUMMARY	44
4	MPC BASED ON THE ERROR MODEL OF A TILT-ROTOR UAV WITH SUSPENDED LOAD	45
4.1	I-SSMPC BASED ON THE ERROR MODEL OF A TILT-ROTOR	45
4.2	NI-SSMPC BASED ON THE ERROR MODEL OF A TILT-ROTOR	50
4.3	TUNING OF MODEL PREDICTIVE CONTROLLERS .	54
4.3.1	State and Input Weight Matrices	55
4.3.2	Terminal Cost	56
4.3.3	Prediction and Control Horizon	56
4.3.4	Constraints Parametrization	57
4.4	SUMMARY	58
5	SIMULATION RESULTS	59
5.1	SIMULATION PROTOCOL	59
5.2	I-SSMPC	62
5.3	NI-SSMPC	65
5.4	SIMULATION WITH PARAMETRIC UNCERTAINTIES	71
5.5	SIMULATION ANALYSIS	71
5.6	SUMMARY	76
6	PRACTICAL IMPLEMENTATION	77
6.1	PROVANT PROJECT	77
6.2	GENERAL DESCRIPTION	78
6.3	SENSOR AND ACTUATOR PROCESSING SUBSYSTEM	79
6.3.1	DataProcessing execution thread	81
6.3.1.1	Attitude Estimation	82
6.3.1.2	Altitude Estimation	84
6.3.1.3	Servomotor States Reading	84
6.3.1.4	Control Actuation	85
6.3.2	Communication execution thread	86
6.4	CONTROL AND ESTIMATION SUBSYSTEM	86
6.4.1	CommLowLevel execution thread	88
6.4.2	DataProcessing execution thread	91
6.4.3	DiscreteControl execution thread	91

6.4.3.1	Control Calculation	93
6.4.3.2	Load Position Estimation	93
6.5	PRELIMINARY RESULTS	95
6.6	SUMMARY	97
7	CONCLUSIONS	99
7.1	FUTURE WORKS	100
	Bibliography	103
	APPENDIX A – Control Design Matrices	109
	APPENDIX B – Symbols Used in the Flowcharts	117

1 INTRODUCTION

In recent years, the use of unnamed aerial vehicles (UAVs) has grown tremendously, mainly because of the technological innovation in fields like design of controllers and estimators, availability of highly-accurate on-chip sensors and GPS at low cost and extensive development of single board computers with increasingly powerful features (PAPACHRISTOS et al., 2011).

Initially, these vehicles were widely used in military applications, due to its flexibility and ability to transmit real-time intelligence, surveillance, and reconnaissance information from hostile areas (KEANE; CARR, 2013). However, in recent years, UAVs began to be used in civilian applications, promoting much research. These vehicles have shown potential for missions like remote sensing, cargo transportation, search and rescue.

Many researches have been produced for cargo transportation tasks due to the great need to carry loads such as food, medicine and general supplies to disaster and war zones, or areas of biological and chemical risk. In this field, for example, quad-rotors UAVs have demonstrated the ability to carry loads up to 100% of their own weight (FAUST et al., 2014).

It is possible to identify two big groups of UAV architectures in the literature: fixed wing and rotary wing. Fixed wings are characterized by high autonomy and high speed; on the other hand, rotary wings, namely helicopters, are characterized by high maneuverability and vertical takeoff and landing. However, lately there has been considered the Tilt-rotor aircraft, which is an aerial vehicle in the middle of these two architectures, propelled by two tilttable rotors. This type of aircraft has been developed since 1930 and is currently used in military and civilian applications. One of the most notable is the Bell-Boeing V-22 Osprey, which is used by US military to perform several kinds of missions and transportation of troop or military equipment, as shown in Figure 1. However, the use of this class of vehicles as UAVs has been studied only in the last decade. Nowadays, there are some Tilt-rotor UAVs already developed, such as TR918 Eagle Eye shown in Figure 2, which began to be built in 1993 and was released in its final version in 1998. This was designed and built for Bell by the research company Scaled Composites.



Figure 1 – Bell-Boeing V-22 Osprey in cargo transportation task



Figure 2 – Bell Eagle Eye TiltRotor UAV (UAVGLOBAL, 2008)

Other current examples of such vehicles are shown in Figure 3. The UAV developed by Korea Aerospace Research Institute (KARI) (Figure 3a) can be used in civil and military applications. In Figure 3b, it is shown the Panther Tilt-rotor UAV, developed by Israel Aerospace Industries (IAI), which differs from others by having the motors displaced forward with respect to the fixed wing. Finally, Figure 3c shows one of the last developments of American Dynamics Flight Systems: The Tilt-rotor AD-150 UAV, characterized by the use of a ducted fan to increase control during hover and transition to forward flight.



(a)



(b)



(c)

Figure 3 – (a) TiltRotor UAV developed by KARI (SUNG-KI, 2011) ; (b) Tilt-rotor Panther UAV (OPLI, 2012); (c) American Dynamics AD-150 UAV (GINGICHASHVILI, 2009)

1.1 MOTIVATION

The development of UAVs involves multiple design challenges, especially in terms of algorithms development for control and localization, which have strict real-time and fast calculation requirements. Commonly, a control cascade structure is applied to UAVs, where it is used at least two control loops with different time scales i.e. translation and rotation subsystems controlled with input-output feedback linearization. For the system control complies with the design specifications, the embedded system must ensure that the signals from each control loop will be calculated within the specified time intervals.

It should be considered that due to the electromechanical design of the UAVs, most of these vehicles are underactuated mechanical systems, inherently unstable system with complex nonlinear dynamics. Furthermore, the addition of a suspended load further complicates the system's dynamics (FAUST et al., 2014).

It is noted, however, that the development of control systems for these vehicles is not trivial, increasing the difficulty when the stabilization of the suspended load is considered, which can cause instability of the UAV. Moreover, the UAVs have a highly nonlinear and time-varying behavior and are constantly affected by aerodynamic disturbances. In addition, they are subjected to error modeling and parametric uncertainties. This means that classical and linear control laws may have certain limitations with respect to its attraction domain, causing instability when the system does not operate near of the equilibrium point. Therefore, advanced control strategies are needed to achieve good performance in autonomous flights, or at least to assist in piloting the vehicle with high maneuverability and robustness with respect to external disturbances (RAFFO, 2011).

Underactuated systems generate complexity and extra challenge in the control area. In this way, techniques developed for fully actuated robots cannot be directly applied to these types of mechanical systems, since most underactuated systems are not fully linearizable by feedback and present non-holonomic constraints. This is the reason why techniques of nonlinear modeling and modern control theories are usually employed to achieve autonomous flight with high performance and under specific conditions, such as hovering, landing, take off, etc (RAFFO, 2011).

However, the problem with nonlinear control methods is that none of them are capable of dealing explicitly with operational constraints prevalent in a control system. Recently, the model predictive control, which is the most advanced control used in the industry apart from PID, could be implemented into faster dynamic systems, due to the development of fast MPC algorithms. The advantage of this formulation is the capability to explicit deal with the constraints of the system and calculate an optimal control action (FAUST et al., 2014).

In some missions, it is necessary to reach higher speed to cover long distances in a short period of time while keeping the high maneuverability provided by rotary-wing aircrafts. From these requirements, a proper aircraft is based on the Tilt-rotor configuration, which can switch between two modes: rotary-wing and fixed-wing, as presented in Figure 4 .

This is the most advantageous characteristic in this UAV configuration, since in the airplane mode, it can reach more velocity comparing with standard helicopters and quad-rotors providing greater autonomy. Also, the helicopter mode presents some advantageous

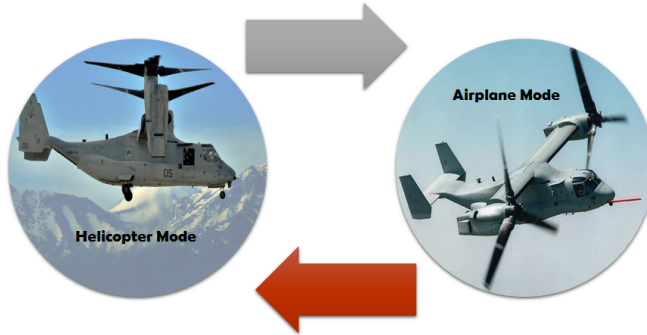


Figure 4 – Tilt-rotor flight modes (HOWARD, 2014; OSBORN, 2015)

characteristics as follows:

1. As this aircraft is lifted and propelled by two rotors, it is possible to reduce the size of each rotor and keep the load capacity, compared to a helicopter having a main rotor.
2. Due to the reduction of propellers, it is possible to reduce energy consumption when compared with the quadrotor helicopter.
3. The simplicity of the mechanical design by using electrical motors with fixed angles of attack propellers provides movement control through the direct transmission of the rotors, varying the speed and tilting them. In a standard helicopter, the angular velocity of the blades is generally constant, where the movement is controlled by varying the angle of attack of the blades (cyclic and collective). This requires transmission between the rotors, as well as precise mechanical devices, in order to change the mentioned angles.
4. Electrical UAVs are an interesting type of vehicle to use inside buildings, due to the use of electric motors instead of combustion engines since they do not pollute the air with combustion residues.
5. Compared to a fixed wing aircraft, this UAV maintains the characteristics of vertical takeoff and landing.

In this context, some researches began to be developed around this kind of vehicle. One of these researches is the ProVANT project, which is a brazilian project conformed by the Automation and Systems Department (DAS) of the Federal University of Santa Catarina (UFSC)

and the Electronic Engineering Department (DELTA) of the Federal University of Minas Gerais (UFMG). The main objective of this research is to develop a small-scale Tilt-rotor aircraft. The study presented in this work is part of one research target, which is to use the UAV for load transportation tasks.

1.2 STATE OF THE ART

In recent years, the autonomy and flexibility of unmanned aerial vehicles have rapidly increased, allowing the use of this kind of aircraft in many dangerous missions, such as search and rescue. These missions often require cargo transportation of sensors, medicines, supplies, food, among other loads, in which multi-rotors UAVs are well-suitable. It is due to their high maneuverability, vertical take-off and landing (VTOL) in small areas and hovering capacities, added to the extra power generated by multi-propellers.

1.2.1 UAV For Load Transportation

Suspended load systems have been extensively studied in the literature. However, researches on suspended load transportation using UAVs are recent. In Palunko, Cruz & Fierro (2012) and Sreenath, Michael & Kumar (2013) quadrotors are used to carry suspended loads, in which path planning algorithms are proposed in order to minimize the load swing avoiding that the quadrotor becomes unstable.

Additionally, some works consider the suspended load dynamics into the mathematical model. In Sadr, Moosavian & Zarafshan (2014) the equations of motion are obtained through the Newton-Euler formulation, for which a feedback linearization controller is designed to control only the attitude and position of the quadrotor. In Pizetta, Brandao & Sarcinelli-Filho (2015) the model of a planar vertical take-off and landing (PVTOL) aircraft-like quadrotor is obtained based on the Euler-Lagrange approach for the XZ motion, and a feedback linearization controller is proposed. In Sreenath, Lee & Kumar (2013) and Cruz, Oishi & Fierro (2015) the tracking problem of the suspended load is handled considering the differential flatness properties of the dynamic model. In Sreenath, Lee & Kumar (2013) a nonlinear geometric controller is designed to stabilize the attitude of the quadrotor and the load, while performing path tracking of the

suspended load. In Cruz, Oishi & Fierro (2015) a linear geometric control law is proposed for the load path tracking problem.

Researches on small tilt-rotor UAVs have started in the last decade. In Sanchez et al. (2008) a back-stepping control law based on a dynamic model obtained through the Newton-Euler formulation is proposed. Furthermore, the proposed model does not consider the dynamics generated by the rotor when tilting. However, regarding this kind of aircraft the coupling between the main body and the rotors body is not negligible. For this reason, in Donadel, Raffo & Becker (2014), a multibody dynamic model, obtained via Euler-Lagrange formulation, considering the rotor's dynamics is developed. A linear mixed H_2/H_∞ controller based on the linearized model is designed in order to perform path tracking, considering robustness requirements.

In Almeida et al. (2014) the equations of motion of the tilt-rotor UAV are extended considering the suspended load dynamics. A linear control law is designed based on the D-Stability and H_∞ requirements to solve the path tracking problem of the tilt-rotor UAV while carrying a suspended load.

In Machado & Raffo (2015), a suspended load stabilization with a quad-rotor using visual feedback is proposed. In this paper, it is presented an integrated platform with an AR.Drone quad-rotor implementing an algorithm for navigation using a front camera, and position estimation using a bottom camera. Two controllers are proposed for load transportation. Firstly, a PID to control the XY motion. Secondly, PI controllers were increased to control load velocities.

In Almeida & Raffo (2015), a non-linear control for Tilt-rotor UAV carrying a suspended load is presented. It proposes a three-level cascade control using input-output feedback linearization; the first two levels are responsible for controlling the attitude and altitude of the aircraft, and the third level for performing the path tracking while the suspended load is stabilized.

1.2.2 Model Predictive Control

Additionally, model predictive control strategies have been also studied to control UAVs and to perform load transportation. Raffo, Ortega & Rubio (2010) proposes a model predictive controller (MPC) to track a reference trajectory. This MPC works in cascade with a nonlinear H_∞ controller to stabilize the rotational movements of a

quadrotor helicopter.

Papachristos et al. (2011) design a model predictive controller for controlling the attitude of a tilt-rotor UAV. Nevertheless, this controller is acting only on the angular position, while a feedforward controller is applied in order to compute the necessary lift to the Tilt-rotor. The prediction model is based on a Newton-Euler formulation linearized around an operation point.

Jansen & Ramirez-Serrano (2011) propose a local obstacle avoidance methodology that is expanded to the MPC approach. The objective of this work was to generate a trajectory that satisfies the UAV constraints, including the geometry of the vehicle, in the obstacle avoidance task. This enables to perform complex maneuvers and navigate through highly confined environments

Jain (2015) designs a linear time invariant (LTV) MPC controller to solve the path tracking problem for transporting a suspended load using a quadrotor. The MPC is executed in a ground station, in which the control action is sent to the quadrotor. The dynamic model used for the controller is based on the Euler-Lagrange formulation.

1.3 OBJECTIVES

The main objective of this master thesis is to design predictive control strategies for unmanned aerial vehicles, in Tilt-rotor configuration, to transport cable-suspended loads.

In order to achieve the main objective, some specific objectives are proposed:

1. Study and implement a mathematical model of a Tilt-rotor UAV with suspended load to be used in the model predictive controller.
2. Design model based predictive control strategies for the path tracking problem of Tilt-rotor UAV with suspended load, considering external disturbances and parametric uncertainties.
3. Explore a formulation in the MPC theory in order to guarantee stability in the system.
4. Implement the designed controllers, compare them to similar controllers and evaluate their performance through simulations.
5. Employ the proposed controllers in an embedded system for the Tilt-rotor UAV assembled by ProVANT project to execute experimental flights.

1.4 OUTLINE

In this chapter, the problem of a Tilt-rotor UAV carrying a transport suspended load was introduced and motivated. State of the art methods were briefly reviewed and the problem statement was formulated for the dissertation. The next chapters are organized as follows:

- **Chapter 2** presents the equations of motion for the Tilt-rotor with suspended load. These equations are based on Euler-Lagrange formulation and the state space representation of these equations is presented. Then, the state space system is linearized around a predefined trajectory. Finally, the system is parametrized.
- **Chapter 3** introduces the control strategies that will be applied to the Tilt-rotor UAV. First, an incremental predictive controller for time varying systems is presented. Secondly, a non-incremental predictive controller is designed using a time varying model. Finally, a terminal cost formulation for time varying systems is presented based on linear matrix inequality (LMI).
- **Chapter 4** shows the predictive control strategies synthesized in Chapter 3, applied to the Tilt-rotor UAV model. The model predictive controllers are extended to the predictive formulation based on the error model. Then, the controllers are tuned in order to be implemented in simulations.
- **Chapter 5** introduces the simulation scenarios, then, simulations are performed aiming to analyze the system behavior and the controllers performance. Finally, some discussion are presented.
- **Chapter 6** presents the practical implementation made in this dissertation. It is given a brief description of the second version of the Tilt-rotor assembled by proVANT project at UFSC.
- **Chapter 7** summarizes the contributions and results presented in this dissertation and suggests possible future research lines.

1.5 LIST OF PUBLICATIONS

The following papers were accepted for publication during the Master's course:

- ANDRADE, R.; RAFFO, G. V.; NORMEY, J. E. Model Predictive Control for Path Tracking of a Tilt-rotor UAV. **XII Simpósio Brasileiro de Automação Inteligente (SBAI)**, p. 1507-1512, 2015.
- ANDRADE, R.; RAFFO, G. V.; NORMEY, J. E. Model Predictive Control of a Tilt-Rotor UAV for Load Transportation. **European Control Conference**, 2016 (accepted).

2 TILT-ROTOR UAV WITH SUSPENDED LOAD DYNAMIC MODEL

In this chapter, the mathematical model of a Tilt-Rotor UAV in helicopter flight-mode with suspended load is presented. The dynamic model of the Tilt-rotor UAV was proposed firstly in Donadel, Raffo & Becker (2014), which was extended in Almeida (2014) to the load transportation problem. In this dissertation, this model is used applying the modifications presented in Rego (2015). Additionally, some parameters are included in this model in order to consider a better approximation of the actuator behavior. The mathematical model presented in this chapter is obtained from the Euler-Lagrange formulation.

The Tilt-rotor UAV, presented in Figure 5, is composed by a main body, where embedded computer, sensor devices, and batteries are supported. Two tiltable rotors are fixed to the main body, controlled by servomotors assumed as revolute joints. Moreover, the suspended load is attached to the main body with a cable, considered as a mass-less and rigid rod. The movements of the Tilt-rotor UAV are



Figure 5 – Tilt-rotor aircraft with suspended load

performed by combining the two thrust forces, (f_r and f_l), generated by the propellers, with the angular inclination of rotors (α_r and α_l).

Additionally, in order to improve controllability in the Y axis, a lateral fixed inclination is considered in the servomotors (RAFFO; ORTEGA; RUBIO, 2011a; DONADEL; RAFFO; BECKER, 2014). It is important to remark that the propellers rotate with opposite direction from each other, with the objective to reduce the drag forces generated by the propellers.

This chapter is structured as follows, Section 2.1 introduces the generalized coordinates used in the mathematical model of the aircraft. Section 2.2 describes each body frames placed in the UAV, with respect to the inertial frame. Section 2.3 drives into the equation of motion of the Tilt-rotor UAV with suspended load based on the Euler-Lagrange formulation. Section 2.5 presents the non-linear state space representation of the system. Section 2.6 shows the system parameters of the aircraft, which are obtained via software CAD and experimentally. Finally, section 2.7 presents the linearization of the non-linear state space model from which a discrete linear model is obtained to be used in the model predictive control design.

2.1 GENERALIZED COORDINATES

In this section, the generalized coordinates for the Tilt-rotor UAV with suspended load are presented. According to the Figure 6, the UAV is composed by four rigid bodies. Then, for modeling purposes, six reference frames are defined. Firstly, reference frame \mathcal{B} is rigidly attached to the rotational axis of the Tilt-rotor, and frame \mathcal{C}_1 to the center of mass of the main body. Then, in the right rotor the reference frame \mathcal{C}_2 is rigidly attached to its center of mass. On the same way, the reference frame \mathcal{C}_3 is rigidly attached to the left rotor's center of mass and \mathcal{C}_4 to the center of mass of the suspended load. Finally, a fixed inertial reference frame \mathcal{I} is considered.

In the Tilt-rotor UAV with suspended load system, it is possible to identify four types of dynamics given by the translation and rotation of the main body, the attitude of the rotors and the orientation of the suspended load. In this way, the translational position of the origin of frame \mathcal{B} with respect to \mathcal{I} , is described by $\boldsymbol{\xi} = [x \ y \ z]^T$.

The orientation of frame \mathcal{B} with respect to \mathcal{I} is obtained through the Euler angles by using the ZYX convention around the local axes. From this, the angular position of the main body is given by $\boldsymbol{\eta} = [\phi \ \theta \ \psi]^T$, which are the well-known roll, pitch and yaw angles, respectively. In the same way, the angular positions of the tiltable

mechanisms, are represented by α_r and α_l , where the subscripts r and l mean right and left.

Furthermore, the angular position of the suspended load is given by $\gamma = [\gamma_1 \ \gamma_2]^T$, around the reference axis x_b and y_b , respectively. Finally, the generalized coordinates vector is described by:

$$\begin{aligned} \mathbf{q} &= [\xi \ \eta \ \alpha_r \ \alpha_l \ \gamma]' \\ &= [x \ y \ z \ \phi \ \theta \ \psi \ \alpha_r \ \alpha_l \ \gamma_1 \ \gamma_2]'. \end{aligned} \quad (2.1)$$

On the other hand, as the servomotors are fixed to the main body, it is assumed that there is no rotation around the reference axis z_{C_2} and z_{C_3} . However, a fixed inclination around the axis x_{C_2} and x_{C_3} is given by $-\beta$ and β angles, respectively. A similar situation occurs with the suspended load, where is assumed that there is no rotation around the axis z_{C_4} .

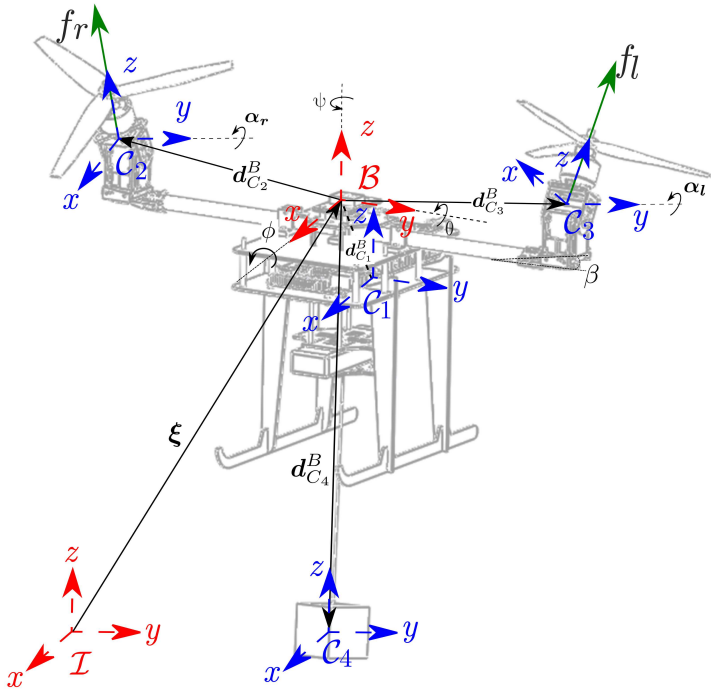


Figure 6 – Tilt-rotor UAV.

2.2 FORWARD KINEMATICS FORMULATION OF THE TILT-ROTOR UAV

In order to represent the relationship of the rotors and the suspended load with the main body, the forward kinematic model is given by:

$$\mathbf{p}_i^B = \mathbf{R}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{d}_{C_i}^B, \quad (2.2)$$

where the rigidly attached point $\mathbf{p}_i^{C_i}$ in the reference C_i is represented with respect to the body frame \mathcal{B} .

In equation (2.2), matrix $\mathbf{R}_{C_i}^B$ is the rotation of reference frame C_i to \mathcal{B} and is given for the bodies frames C_2, C_3, C_4 , as follows:

$$\mathbf{R}_{C_2}^B = \begin{bmatrix} C\alpha_R(t) & 0 & S\alpha_R(t) \\ -S\alpha_R(t)S\beta & C\beta & C\alpha_R(t)S\beta \\ -S\alpha_R(t)C\beta & -S\beta & C\alpha_R(t)C\beta \end{bmatrix}, \quad (2.3)$$

$$\mathbf{R}_{C_3}^B = \begin{bmatrix} C\alpha_L(t) & 0 & S\alpha_L(t) \\ S\alpha_L(t)S\beta & C\beta & -C\alpha_L(t)S\beta \\ -S\alpha_L(t)C\beta & S\beta & C\alpha_L(t)C\beta \end{bmatrix}, \quad (2.4)$$

$$\mathbf{R}_{C_4}^B = \begin{bmatrix} C\gamma_2(t) & S\gamma_1(t)S\gamma_2(t) & C\gamma_1(t)S\gamma_2(t) \\ 0 & C\gamma_1(t) & -S\gamma_1(t) \\ -S\gamma_2(t) & C\gamma_2(t)S\gamma_1(t) & C\gamma_1(t)C\gamma_2(t) \end{bmatrix}. \quad (2.5)$$

Also, rotation matrix $\mathbf{R}_{C_1}^B$ is assumed as constant, since frame C_1 is rigidly attached to the main body of the Tilt-rotor UAV. Additionally, vector $\mathbf{d}_{C_i}^B$ is the translation between the origin of frames \mathcal{B} and C_i . Vectors $\mathbf{d}_{C_1}^B$, $\mathbf{d}_{C_2}^B$ and $\mathbf{d}_{C_3}^B$ are considered constants and are expressed as follows:

$$\mathbf{d}_{C_1}^B = [d_{1x} \quad d_{1y} \quad d_{1z}]^T, \quad (2.6)$$

$$\mathbf{d}_{C_2}^B = [dx \quad -dy \quad dz]^T, \quad (2.7)$$

$$\mathbf{d}_{C_3}^B = [dx \quad dy \quad dz]^T. \quad (2.8)$$

On the other hand, vector $\mathbf{d}_{C_4}^B$ does not have the same behavior, since this distance varies with the load rotation angles. According with Almeida (2014), it is possible to parametrize the suspended load as a pendulum with a massless rigid rod of length l and two degrees of freedom, as commented in Section 2.1. In this way, this vector takes

the following form:

$$\begin{aligned} \mathbf{d}_{C_4}^B &= \mathbf{R}_{C_4}^B \begin{bmatrix} 0 \\ 0 \\ -l \end{bmatrix} \\ &= \begin{bmatrix} -l C\gamma_1 S\gamma_2 \\ l S\gamma_1 \\ -l C\gamma_1 C\gamma_2 \end{bmatrix}. \end{aligned} \quad (2.9)$$

Moreover, the point $\mathbf{p}_i^{\mathcal{I}}$, rigidly attached to the moving reference frame \mathcal{B} and expressed in the inertial frame \mathcal{I} , is given by:

$$\mathbf{p}_i^{\mathcal{I}} = \mathbf{R}_B^{\mathcal{I}} \mathbf{p}_i^B + \boldsymbol{\xi}, \quad (2.10)$$

where $\mathbf{R}_B^{\mathcal{I}}$ is the rotation from frame \mathcal{B} to the inertial frame \mathcal{I} , and $\boldsymbol{\xi}$ is the translation position between origin of both references frames. Matrix $\mathbf{R}_B^{\mathcal{I}}$ is calculated using the roll-pitch-yaw convention (SPONG; HUTCHINSON; VIDYASAGAR, 2005) and is expressed by:

$$\mathbf{R}_B^{\mathcal{I}} = \begin{bmatrix} C\psi C\theta & S\psi C\psi S\theta - C\phi S\psi & C\phi C\psi S\theta + S\phi S\psi \\ S\psi C\theta & S\phi S\psi S\theta + C\phi C\psi & C\phi S\psi S\theta - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix}. \quad (2.11)$$

Finally, using equation (2.2) into (2.10), the rigid motion of all bodies with respect to the inertial frame \mathcal{I} is computed as follows:

$$\mathbf{p}_i^{\mathcal{I}} = \mathbf{R}_B^{\mathcal{I}} (\mathbf{R}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{d}_{C_i}^B) + \boldsymbol{\xi}. \quad (2.12)$$

2.3 EQUATION OF MOTION OF THE TILT-ROTOR UAV WITH SUSPENDED LOAD

In order to obtain the dynamic model of the Tilt-rotor UAV with suspended load, the Euler-Lagrange formulation is used, which is represented in its canonical form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{F}(\mathbf{q}) + \mathbf{F}_{ext} + \mathbf{F}_{drag}, \quad (2.13)$$

where $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centripetal force matrix, $\mathbf{G}(\mathbf{q})$ is the gravitational force vector, $\mathbf{F}(\mathbf{q})$ is the generalized forces/torques vector, \mathbf{F}_{ext} is the external perturbation of

the system, and \mathbf{F}_{drag} , known as drag force, is expressed by:

$$\mathbf{F}_{drag} = -\boldsymbol{\mu}\dot{\mathbf{q}}, \quad (2.14)$$

where $\boldsymbol{\mu}$ is the viscous coefficient matrix. The $\mathbf{F}(\mathbf{q})$ term can be rewritten as:

$$\mathbf{F}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u}(t), \quad (2.15)$$

where the input control $\mathbf{u}(t) = [f_r \ f_l \ \tau_r \ \tau_l]^T$. Later, all these terms will be detailed in the following subsections. Consequently, equation (2.13) can be rewritten as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + [\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\mu}]\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{F}_{ext}. \quad (2.16)$$

2.3.1 Inertia Matrix of the Tilt-rotor UAV with Suspended Load

According to Spong, Hutchinson & Vidyasagar (2005), the inertia matrix $\mathbf{M}(\mathbf{q})$ can be calculated through the kinetic energy for a rigid body expressed as a quadratic function of generalized velocities vector $\dot{\mathbf{q}}$ in the form of $\mathbf{K} = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}}$, where the matrix $\mathbf{M}(\mathbf{q})$ is symmetric and positive definite. However, taking into account that the Tilt-rotor UAV is considered as a multibody system, according to Shabana (2005), the kinetic energy of the whole body is calculated by the sum of the individual energies \mathbf{K}_i of each body, given by:

$$\mathbf{K} = \sum_{i=1}^4 \mathbf{K}_i, \quad (2.17)$$

where the kinetic energy of each body is expressed as follows:

$$\mathbf{K}_i = \frac{1}{2} \int_{V_i} \rho_i(\mathbf{v}_i^{\mathcal{I}})^T (\mathbf{v}_i^{\mathcal{I}}) dV_i. \quad (2.18)$$

In equation (2.18), the term ρ_i is the mass density of the body i and $\mathbf{v}_i^{\mathcal{I}}$ is the time derivative of equation (2.12), resulting in the velocity of the point \mathbf{p}_i with respect to inertial frame \mathcal{I} . This velocity is expressed by:

$$\mathbf{v}_i^{\mathcal{I}} = \dot{\mathbf{R}}_B^{\mathcal{I}}(\mathbf{R}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{d}_{C_i}^B) + \mathbf{R}_B^{\mathcal{I}}(\dot{\mathbf{R}}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{R}_{C_i}^B \dot{\mathbf{p}}_i^{C_i} + \dot{\mathbf{d}}_{C_i}^B) + \dot{\boldsymbol{\xi}}, \quad (2.19)$$

Moreover, it is assumed that, as the point \mathbf{p}_i is rigidly attached

to the frame \mathcal{C}_i , the term $\dot{\mathbf{p}}_i^{C_i} = 0$ for $i = 1$ to 4. In section 2.2, it was mentioned that the translation vectors $\mathbf{d}_{C_i}^B$ for $i = 1$ to 3 are constants, which leads to $\dot{\mathbf{d}}_{C_i}^B = 0$ for $i = 1$ to 3. Besides, since the rotation matrix $\mathbf{R}_{C_1}^B$ is constant, $\dot{\mathbf{R}}_{C_1}^B = 0_{3 \times 3}$.

Furthermore, expression (2.19) can be rewritten, for all bodies of the system, using the property of skew symmetric matrices give by $\dot{\mathbf{R}}_B^A = \mathbf{R}_B^A \mathbf{S}(\mathbf{w}_{B,A}^B)$, defining $\mathbf{w}_{B,A}^B \in R^3$ as the angular velocity of frame B with respect to A represented in frame B , as follows:

$$\begin{aligned} \mathbf{v}_1^I &= \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{R}_{C_1}^B \mathbf{p}_1^{C_1} + \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{d}_{C_1}^B + \dot{\boldsymbol{\xi}} & (2.20) \\ \mathbf{v}_i^I &= \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{R}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{d}_{C_i}^B + \mathbf{R}_B^I \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{w}_{C_i,B}^{C_i}) \mathbf{p}_i^{C_i} + \dot{\boldsymbol{\xi}} \\ &\text{for } i = 2, 3 \\ \mathbf{v}_4^I &= \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{R}_{C_4}^B \mathbf{p}_4^{C_4} + \mathbf{R}_B^I \mathbf{S}(\mathbf{w}_{B,I}^B) \mathbf{d}_{C_4}^B + \mathbf{R}_B^I \mathbf{R}_{C_4}^B \mathbf{S}(\mathbf{w}_{C_4,B}^{C_4}) \mathbf{p}_4^{C_4} \\ &\quad + \mathbf{R}_B^I \mathbf{d}_{C_4}^B + \dot{\boldsymbol{\xi}}. \end{aligned}$$

Now, by using the properties of symmetric matrices $\mathbf{S}(\mathbf{p})\mathbf{q} = \mathbf{S}(\mathbf{q})^T \mathbf{p}$ and $\mathbf{S}(\mathbf{R}\mathbf{p}) = \mathbf{R}\mathbf{S}(\mathbf{p})\mathbf{R}^T$, equations (2.20) are rewritten as:

$$\begin{aligned} \mathbf{v}_1^I &= \mathbf{R}_B^I \mathbf{R}_{C_1}^B \mathbf{S}(\mathbf{p}_1^{C_1})^T (\mathbf{R}_{C_1}^B)^T \mathbf{w}_{B,I}^B + \mathbf{R}_B^I \mathbf{S}(\mathbf{d}_{C_1}^B)^T \mathbf{w}_{B,I}^B + \dot{\boldsymbol{\xi}} & (2.21) \\ \mathbf{v}_i^I &= \mathbf{R}_B^I \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i})^T (\mathbf{R}_{C_i}^B)^T \mathbf{w}_{B,I}^B + \mathbf{R}_B^I \mathbf{S}(\mathbf{d}_{C_i}^B)^T \mathbf{w}_{B,I}^B + \mathbf{R}_B^I \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{w}_{C_i,B}^{C_i} + \dot{\boldsymbol{\xi}} \\ &\text{for } i = 2, 3 \\ \mathbf{v}_4^I &= \mathbf{R}_B^I \mathbf{R}_{C_4}^B \mathbf{S}(\mathbf{p}_4^{C_4})^T (\mathbf{R}_{C_4}^B)^T \mathbf{w}_{B,I}^B + \mathbf{R}_B^I \mathbf{S}(\mathbf{d}_{C_4}^B)^T \mathbf{w}_{B,I}^B + \mathbf{R}_B^I \mathbf{R}_{C_4}^B \mathbf{S}(\mathbf{p}_4^{C_4})^T \mathbf{w}_{C_4,B}^{C_4} \\ &\quad + \mathbf{R}_B^I \mathbf{d}_{C_4}^B + \dot{\boldsymbol{\xi}}. \end{aligned}$$

According to (ALMEIDA, 2014), the quadratic products of the velocities can be expressed as:

$$\begin{aligned} (\mathbf{v}_1^I)^T (\mathbf{v}_1^I) &= X_1, \\ (\mathbf{v}_2^I)^T (\mathbf{v}_2^I) &= X_2 + Y_2, \\ (\mathbf{v}_3^I)^T (\mathbf{v}_3^I) &= X_3 + Y_3, \\ (\mathbf{v}_4^I)^T (\mathbf{v}_4^I) &= X_4 + Y_4 + Z_4. \end{aligned} \tag{2.22}$$

where X_i , Y_i and Z_i are given by:

$$\begin{aligned}
X_i &= \dot{\boldsymbol{\xi}}^T \boldsymbol{\xi} + 2\dot{\boldsymbol{\xi}}^T \mathbf{R}_B^T \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i})^T (\mathbf{R}_{C_i}^B)^T \mathbf{w}_{BL}^B + 2\dot{\boldsymbol{\xi}}^T \mathbf{R}_B^T \mathbf{S}(\mathbf{d}_{C_i}^B)^T \mathbf{w}_{BL}^B \\
&\quad + (\mathbf{w}_{BL}^B)^T \left[\mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i}) \mathbf{S}(\mathbf{p}_i^{C_i})^T (\mathbf{R}_{C_i}^B)^T + 2\mathbf{S}(\mathbf{p}_i^{C_i}) (\mathbf{R}_{C_i}^B)^T \mathbf{S}(\mathbf{d}_{C_i}^B)^T \right. \\
&\quad \left. + \mathbf{S}(\mathbf{d}_{C_i}^B) \mathbf{S}(\mathbf{d}_{C_i}^B)^T \right] \mathbf{w}_{BL}^B, \\
Y_i &= 2\dot{\boldsymbol{\xi}}^T \mathbf{R}_B^T \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{w}_{C_i B}^{C_i} + (\mathbf{w}_{C_i B}^{C_i})^T \mathbf{S}(\mathbf{p}_i^{C_i}) \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{w}_{C_i B}^{C_i} \quad (2.23) \\
&\quad + (\mathbf{w}_{BL}^B)^T \left[2\mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i}) \mathbf{S}(\mathbf{p}_i^{C_i})^T + 2\mathbf{S}(\mathbf{d}_{C_i}^B) \mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i})^T \right] \mathbf{w}_{C_i B}^{C_i}, \\
Z_i &= 2\dot{\boldsymbol{\xi}}^T \mathbf{R}_B^T \mathbf{d}_{C_i}^B + (\mathbf{w}_{BL}^B)^T \left[2\mathbf{R}_{C_i}^B \mathbf{S}(\mathbf{p}_i^{C_i}) (\mathbf{R}_{C_i}^B)^T + 2\mathbf{S}(\mathbf{d}_{C_i}^B) \right] \mathbf{d}_{C_i}^B \\
&\quad + 2\mathbf{w}_{C_i B}^{C_i} \mathbf{S}(\mathbf{p}_i^{C_i}) \mathbf{R}_{C_i}^B \mathbf{d}_{C_i}^B + (\mathbf{d}_{C_i}^B)^T \mathbf{d}_{C_i}^B.
\end{aligned}$$

From Shabana (2005, p. 147), by assuming the origin of the reference frame \mathcal{C}_i attached to the center of mass of the body i yields to:

$$\int_{V_i} \rho_i \mathbf{S}(\mathbf{p}_i^{C_i}) dV_i = \mathbf{0}_{3 \times 1}. \quad (2.24)$$

Thereby, applying equation (2.24) into equation (2.23) the kinetic energy for all bodies are expressed as follows:

$$\begin{aligned}
K_1 &= X'_1, \\
K_2 &= X'_2 + Y'_2, \\
K_3 &= X'_3 + Y'_3, \\
K_4 &= X'_4 + Y'_4 + Z'_4,
\end{aligned} \quad (2.25)$$

where X'_i , Y'_i and Z'_i are given by:

$$\begin{aligned}
X'_i &= \frac{1}{2} m_i \dot{\boldsymbol{\xi}}^T \boldsymbol{\xi} + m_i \dot{\boldsymbol{\xi}}^T \mathbf{R}_B^T \mathbf{S}(\mathbf{d}_{C_i}^B)^T \mathbf{w}_{BL}^B \\
&\quad + \frac{1}{2} (\mathbf{w}_{BL}^B)^T \left[\mathbf{R}_{C_i}^B \left[\int_{V_i} \rho_i \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{S}(\mathbf{p}_i^{C_i}) dV_i \right] (\mathbf{R}_{C_i}^B)^T \right. \\
&\quad \left. + m_i \mathbf{S}(\mathbf{d}_{C_i}^B)^T \mathbf{S}(\mathbf{d}_{C_i}^B) \right] \mathbf{w}_{BL}^B, \\
Y'_i &= (\mathbf{w}_{BL}^B)^T \mathbf{R}_{C_i}^B \left[\int_{V_i} \rho_i \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{S}(\mathbf{p}_i^{C_i}) dV_i \right] \mathbf{w}_{C_i B}^{C_i} \\
&\quad + (\mathbf{w}_{C_i B}^{C_i})^T \left[\int_{V_i} \rho_i \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{S}(\mathbf{p}_i^{C_i}) dV_i \right] \mathbf{w}_{C_i B}^{C_i}, \\
Z'_i &= \dot{\boldsymbol{\xi}}^T m_i \mathbf{R}_B^T \mathbf{d}_{C_i}^B + (\mathbf{w}_{BL}^B)^T m_i \mathbf{S}(\mathbf{d}_{C_i}^B) \mathbf{d}_{C_i}^B + \frac{1}{2} (\mathbf{d}_{C_i}^B)^T m_i \mathbf{d}_{C_i}^B,
\end{aligned} \quad (2.26)$$

in which m_i is the mass of body i .

From Shabana (2005), the inertia tensor of the body i with respect to the frame \mathcal{C}_i is given by:

$$\mathbf{I}_i = \int_{V_i} \rho_i \mathbf{S}(\mathbf{p}_i^{C_i})^T \mathbf{S}(\mathbf{p}_i^{C_i}) dV_i = \begin{bmatrix} I_{xx}^i & I_{xy}^i & I_{xz}^i \\ I_{yx}^i & I_{yy}^i & I_{yz}^i \\ I_{zx}^i & I_{zy}^i & I_{zz}^i \end{bmatrix}, \quad (2.27)$$

and, considering the Steiner's theorem for parallel axis, the inertia tensor of the body i for a rotation around an axis displaced d_i , given by:

$$\mathbf{J}_i = \mathbf{R}_{C_i}^B \mathbf{I}_i (\mathbf{R}_{C_i}^B)^T + m_i \mathbf{S}(\mathbf{d}_{C_i}^B)^T \mathbf{S}(\mathbf{d}_{C_i}^B). \quad (2.28)$$

The terms X'_i , Y'_i and Z'_i from the expression (2.26) can be simplified using equations (2.27) and (2.28), as follows:

$$\begin{aligned} X'_i &= \frac{1}{2} m_i \dot{\boldsymbol{\xi}}^T \boldsymbol{\xi} - m_i \dot{\boldsymbol{\xi}}^T \mathbf{R}_B^I \mathbf{S}(\mathbf{d}_{C_i}^B) \mathbf{w}_{BL}^B + \frac{1}{2} (\mathbf{w}_{BL}^B)^T \mathbf{J}_i \mathbf{w}_{BL}^B, \\ Y'_i &= (\mathbf{w}_{BL}^B)^T \mathbf{R}_{C_i}^B \mathbf{I}_i \mathbf{w}_{C_i B}^{C_i} + \frac{1}{2} (\mathbf{w}_{C_i B}^{C_i})^T \mathbf{I}_i \mathbf{w}_{C_i B}^{C_i}, \\ Z'_i &= \dot{\boldsymbol{\xi}}^T m_i \mathbf{R}_B^I \mathbf{d}_{C_i}^B + (\mathbf{w}_{BL}^B)^T m_i \mathbf{S}(\mathbf{d}_{C_i}^B) \mathbf{d}_{C_i}^B + \frac{1}{2} (\mathbf{d}_{C_i}^B)^T m_i \mathbf{d}_{C_i}^B. \end{aligned} \quad (2.29)$$

With the aim to rewrite the kinetics energies as a function of the generalized coordinates, the angular velocity \mathbf{w}_{BL}^B can be expressed as (RAFFO, 2011) :

$$\mathbf{w}_{BL}^B = \begin{bmatrix} 1 & 0 & -S\theta \\ 0 & C\phi & S\phi C\theta \\ 0 & -S\phi & C\phi C\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \mathbf{W}_n \dot{\boldsymbol{\eta}} \quad (2.30)$$

Moreover, the angular velocities of the tiltable mechanism can be computed as:

$$\mathbf{w}_{C_2 B}^{C_2} = \dot{\boldsymbol{\alpha}}_r [0 \ 1 \ 0]^T = \dot{\boldsymbol{\alpha}}_r \mathbf{a}, \quad (2.31)$$

$$\mathbf{w}_{C_3 B}^{C_3} = \dot{\boldsymbol{\alpha}}_l [0 \ 1 \ 0]^T = \dot{\boldsymbol{\alpha}}_l \mathbf{a}. \quad (2.32)$$

In addition, the angular velocities of the suspended load can be

calculated solving $\mathbf{S}(\mathbf{w}_{C_4B}^{C_4}) = [\mathbf{R}_{C_4}^B]^T \dot{\mathbf{R}}_{C_4}^B$, resulting in:

$$\mathbf{w}_{C_4B}^{C_4} = \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 C \gamma_1 \\ -\dot{\gamma}_2 S \gamma_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & C \gamma_1 \\ 0 & -S \gamma_1 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} = \mathbf{P} \dot{\boldsymbol{\gamma}} \quad (2.23)$$

Additionally, the time derivative of equation (2.9) is given by:

$$\dot{\mathbf{d}}_{C_4}^B = \begin{bmatrix} lS\gamma_1 S\gamma_2 \dot{\gamma}_1 - lC\gamma_1 C\gamma_2 \dot{\gamma}_2 & \\ & lC\gamma_1 \dot{\gamma}_1 \\ lS\gamma_1 C\gamma_2 \dot{\gamma}_1 + lC\gamma_1 S\gamma_2 \dot{\gamma}_2 & \end{bmatrix} = \begin{bmatrix} lS\gamma_1 S\gamma_2 & -lC\gamma_1 C\gamma_2 \\ lC\gamma_1 & 0 \\ lS\gamma_1 C\gamma_2 & lC\gamma_1 S\gamma_2 \end{bmatrix} \begin{bmatrix} \dot{\gamma}_1 \\ \dot{\gamma}_2 \end{bmatrix} = \mathbf{L} \dot{\boldsymbol{\gamma}} \quad (2.24)$$

Finally, replacing equations (2.29) into equations (2.22) considering (2.30) – (2.34), the total kinetic energy of the Tilt-rotor UAV with suspended load is computed through equation (2.17). Then, the inertia matrix is obtained as follows:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} m\mathbf{I}_{3 \times 3} & -\mathbf{R}_B^T \mathbf{H} \mathbf{W}_\eta & \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 1} & m_4 \mathbf{R}_B^T \mathbf{L} \\ * & (\mathbf{W}_\eta)^T \mathbf{J} \mathbf{W}_\eta & \mathbf{W}_\eta \mathbf{R}_B^T \mathbf{I}_2 \mathbf{a} & \mathbf{W}_\eta \mathbf{R}_B^T \mathbf{I}_3 \mathbf{a} & \mathbf{m}_{25} \\ * & * & \mathbf{a}^T \mathbf{I}_2 \mathbf{a} & 0 & \mathbf{0}_{1 \times 2} \\ * & * & * & \mathbf{a}^T \mathbf{I}_3 \mathbf{a} & \mathbf{0}_{1 \times 2} \\ * & * & * & * & m_4 \mathbf{L}^T \mathbf{L} + \mathbf{P}^T \mathbf{I}_4 \mathbf{P} \end{bmatrix}, \quad (2.25)$$

where $\mathbf{m}_{25} = \mathbf{W}_\eta \mathbf{R}_{C_4}^B \mathbf{I}_4 \mathbf{P} + \mathbf{W}_\eta \mathbf{H}_4 \mathbf{P}$, $m = \sum m_i$, $\mathbf{J} = \sum \mathbf{J}_i$, $\mathbf{H}_i = \mathbf{S}(\sum m_i \mathbf{d}_{C_i}^B)$ and $\mathbf{H} = \sum \mathbf{H}_i$.

2.3.2 Coriolis and Centripetal Matrix

The Coriolis and centripetal matrix $\mathbf{C}(\mathbf{q})$ is derived using the Christoffel symbol of the first kind from the inertia matrix $\mathbf{M}(\mathbf{q})$, where the $c(k, j)$ element of the matrix $\mathbf{C}(\mathbf{q})$ is calculated as follows (SPONG; HUTCHINSON; VIDYASAGAR, 2005):

$$c(k, j) = \frac{1}{2} \sum_{i=1}^{10} \left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k} \right], \quad (2.26)$$

whit $m_{i,j}$ an element of the inertia matrix $\mathbf{M}(\mathbf{q})$.

2.3.3 Gravity Forces Vector

The gravitational forces vector is calculated through the potential energy. As mentioned before, the total energy of the whole body is the sum of the energies of each body; accordingly, the total potential energy of the Tilt-rotor UAV with suspended load can be calculated by (SHABANA, 2005):

$$P = \sum_{i=1}^4 P_i. \quad (2.37)$$

Moreover, the potential energy of each body is obtained as:

$$P_i = - \int_{V_i} \rho_i(\mathbf{g}^{\mathcal{I}})^T \mathbf{p}_i^{\mathcal{I}} dV_i, \quad (2.38)$$

where the gravity vector with respect to the inertial frame \mathcal{I} is given by $\mathbf{g}^{\mathcal{I}} = [0 \ 0 \ -g]$.

Thus, the equation (2.12) is replaced into (2.38) yielding:

$$P_i = - \int_{V_i} \rho_i(\mathbf{g}^{\mathcal{I}})^T [\mathbf{R}_B^{\mathcal{I}}(\mathbf{R}_{C_i}^B \mathbf{p}_i^{C_i} + \mathbf{d}_{C_i}^B) + \boldsymbol{\xi}] dV_i. \quad (2.39)$$

Assuming that all bodies are symmetric, setting $\int_{V_i} \rho_i \mathbf{p}_i^{C_i} dV_i = 0$, and taking into account that the mass of each body is given by $\int_{V_i} \rho_i dV_i = m_i$, equation (2.39) can be rewritten as:

$$P_i = -(\mathbf{g}^{\mathcal{I}})^T (\mathbf{R}_B^{\mathcal{I}} m_i \mathbf{d}_{C_i}^B + m_i \boldsymbol{\xi}). \quad (2.40)$$

Finally, the total potential energy of the system is computed as follows:

$$P = -(\mathbf{g}^{\mathcal{I}})^T \mathbf{R}_B^{\mathcal{I}} \left(\sum_{i=1}^4 m_i \mathbf{d}_{C_i}^B \right) + (\mathbf{g}^{\mathcal{I}})^T m \boldsymbol{\xi}, \quad (2.41)$$

which is given by the sum of each potential energy, as presented in equation (2.37).

Therefore, the gravity forces vector can be obtained by:

$$\mathbf{G}(\mathbf{q}) = \frac{\partial P}{\partial \mathbf{q}} = \begin{bmatrix} \frac{\partial P}{\partial q_1} \\ \frac{\partial P}{\partial q_2} \\ \vdots \\ \frac{\partial P}{\partial q_{10}} \end{bmatrix}. \quad (2.42)$$

2.3.4 External Forces Vector

The external forces vector is given by:

$$\mathbf{F}(\mathbf{q}) = [\mathbf{T}^{\mathcal{I}} \quad \boldsymbol{\tau}^{\mathcal{I}} \quad \tau_{\alpha_r}^{\mathcal{I}} \quad \tau_{\alpha_l}^{\mathcal{I}} \quad \tau_{\gamma_1}^{\mathcal{I}} \quad \tau_{\gamma_2}^{\mathcal{I}}]^T, \quad (2.43)$$

where $\mathbf{T}^{\mathcal{I}}$ are the thrust forces generated by the propellers expressed in the inertial frame \mathcal{I} , $\boldsymbol{\tau}^{\mathcal{I}}$ are the rotational torques calculated by adding the torques generated by the thrust and drag forces of the propellers and $\tau_{\alpha}^{\mathcal{I}}$ are the torques generated by the tiltable mechanisms. Moreover, $\tau_{\gamma}^{\mathcal{I}}$ are the torques applied in the load, that are assumed zero, because there is no directly actuation over this variable.

As reference frames \mathcal{C}_2 and \mathcal{C}_3 are rigidly attached to their bodies, the forces f_r and f_l are always in the components \mathcal{C}_{zi} of the reference bodies. Then, representing this forces with respect to the moving frame \mathcal{B} , yields to:

$$\mathbf{F}_R^B = \begin{bmatrix} f_{R_x}^B \\ f_{R_y}^B \\ f_{R_z}^B \end{bmatrix} = \mathbf{R}_{\mathcal{C}_2}^B \begin{bmatrix} 0 \\ 0 \\ f_r \end{bmatrix} = \begin{bmatrix} S\alpha_r \\ C\alpha_r S\beta \\ C\alpha_r C\beta \end{bmatrix} f_r = \mathbf{r}_R f_r, \quad (2.44)$$

$$\mathbf{F}_L^B = \begin{bmatrix} f_{L_x}^B \\ f_{L_y}^B \\ f_{L_z}^B \end{bmatrix} = \mathbf{R}_{\mathcal{C}_3}^B \begin{bmatrix} 0 \\ 0 \\ f_l \end{bmatrix} = \begin{bmatrix} S\alpha_l \\ -C\alpha_l S\beta \\ C\alpha_l C\beta \end{bmatrix} f_l = \mathbf{r}_L f_l. \quad (2.45)$$

The total thrust forces generated in the Tilt-rotor UAV is the sum of the forces generated by each rotor with respect to the frame B. The thrust forces $\mathbf{T}^{\mathcal{I}}$, which represents the forces \mathbf{F}_R^B and \mathbf{F}_L^B with respect to the \mathcal{I} , are given by:

$$\mathbf{T}^{\mathcal{I}} = \begin{bmatrix} T_x^{\mathcal{I}} \\ T_y^{\mathcal{I}} \\ T_z^{\mathcal{I}} \end{bmatrix} = \mathbf{R}_B^{\mathcal{I}} (\mathbf{F}_R^B + \mathbf{F}_L^B) = [\mathbf{R}_B^{\mathcal{I}} \mathbf{r}_R^B \quad \mathbf{R}_B^{\mathcal{I}} \mathbf{r}_L^B] \begin{bmatrix} f_r \\ f_l \end{bmatrix}. \quad (2.46)$$

On the other hand, the torques generated by the thrust forces can be computed by the cross product of the thrusts, represented with respect to the frame \mathcal{B} , and the translation vector between the origins of frames \mathcal{B} and \mathcal{C}_i , as:

$$\begin{aligned}
 \boldsymbol{\tau}_f^B &= \mathbf{d}_{C_2}^B \times \mathbf{F}_R^B + \mathbf{d}_{C_3}^B \times \mathbf{F}_L^B \\
 &= \begin{bmatrix} -C\alpha_r C\beta d_y - C\alpha_r S\beta d_z & C\alpha_l C\beta d_y + C\alpha_l S\beta d_z \\ S\alpha_r d_z - C\alpha_r C\beta d_x & S\alpha_l d_z - C\alpha_l C\beta d_x \\ S\alpha_r d_y + C\alpha_r S\beta d_x & -S\alpha_l d_y - C\alpha_l S\beta d_x \end{bmatrix} \begin{bmatrix} f_r \\ f_l \end{bmatrix} \\
 &= [\boldsymbol{\tau}_{f_r}^B \quad \boldsymbol{\tau}_{f_l}^B] \begin{bmatrix} f_r \\ f_l \end{bmatrix}. \tag{2.47}
 \end{aligned}$$

According with Raffo (2011) and Castillo, Lozano & Dzul (2004), the drag torque is computed by $\tau_{drag} = k_\tau \Omega^2$, where k_τ is a constant and Ω is the angular velocity of the rotor; however, the thrust force can be approximated by $f = b\Omega^2$, where b is the thrust coefficient of the rotor. In this context, the drag torque can be rewritten as follows:

$$\tau_{drag} = \frac{k_\tau}{b} f. \tag{2.48}$$

Also, the sign of the force f is taken according to the direction of the propeller's rotation, being positive for counter-clockwise rotation and negative for clockwise rotation. In this context, assuming that the left rotor rotates clockwise and expressing this torque with respect to the moving frame \mathcal{B} , the total drag torque is given by:

$$\begin{aligned}
 \boldsymbol{\tau}_{drag}^B &= \begin{bmatrix} \frac{k_\tau}{b} (f_{R_x}^B - f_{L_x}^B) \\ \frac{k_\tau}{b} (f_{R_y}^B - f_{L_y}^B) \\ \frac{k_\tau}{b} (f_{R_z}^B - f_{L_z}^B) \end{bmatrix} \\
 &= \begin{bmatrix} \frac{k_\tau}{b} S\alpha_r & -\frac{k_\tau}{b} S\alpha_l \\ \frac{k_\tau}{b} C\alpha_r S\beta & \frac{k_\tau}{b} C\alpha_l S\beta \\ \frac{k_\tau}{b} C\alpha_r C\beta & -\frac{k_\tau}{b} C\alpha_l C\beta \end{bmatrix} \begin{bmatrix} f_r \\ f_l \end{bmatrix} \\
 &= [\boldsymbol{\tau}_{d_r}^B \quad \boldsymbol{\tau}_{d_l}^B] \begin{bmatrix} f_r \\ f_l \end{bmatrix}. \tag{2.49}
 \end{aligned}$$

Therefore, the external total torque applied to the vehicle is expressed as the sum of the torques generated by the thrust forces

and the drag torques of the propellers. Thus, it is given by:

$$\boldsymbol{\tau}^B = [\boldsymbol{\tau}_{f_r}^B + \boldsymbol{\tau}_{d_r}^B \quad \boldsymbol{\tau}_{f_l}^B + \boldsymbol{\tau}_{d_l}^B] \begin{bmatrix} f_r \\ f_l \end{bmatrix} = [\boldsymbol{\tau}_R^B \quad \boldsymbol{\tau}_L^B] \begin{bmatrix} f_r \\ f_l \end{bmatrix}. \quad (2.50)$$

Finally, the external forces vector can take the form of $F(\boldsymbol{q}) = B(\boldsymbol{q})\boldsymbol{u}$ expressed as:

$$\boldsymbol{F}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{T}^{\mathcal{I}} \\ \boldsymbol{\tau}^{\mathcal{I}} \\ \tau_{\alpha_r}^{\mathcal{I}} \\ \tau_{\alpha_l}^{\mathcal{I}} \\ \tau_{\gamma_1}^{\mathcal{I}} \\ \tau_{\gamma_2}^{\mathcal{I}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_B^{\mathcal{I}} \boldsymbol{r}_R & \boldsymbol{R}_B^{\mathcal{I}} \boldsymbol{r}_L & 0 & 0 \\ \boldsymbol{W}_\eta^{\mathcal{I}} \boldsymbol{\tau}_R^B & \boldsymbol{W}_\eta^{\mathcal{I}} \boldsymbol{\tau}_L^B & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_r \\ f_l \\ \tau_r \\ \tau_l \end{bmatrix} \quad (2.51)$$

2.4 FRICTION FORCE VECTOR

In this dissertation, the friction forces vector is assumed proportional to the generalized velocities. In equation (2.14), $\boldsymbol{\mu}$ is considered a semidefinite constant diagonal matrix, where each element of the minor diagonal represents the viscous coefficient associated with their respective generalized velocity. Thus, the matrix $\boldsymbol{\mu}$ is given by:

$$\boldsymbol{\mu} = \text{diag}(\mu_x \quad \mu_y \quad \mu_z \quad \mu_\phi \quad \mu_\theta \quad \mu_\psi \quad \mu_{\alpha_r} \quad \mu_{\alpha_l} \quad \mu_{\gamma_1} \quad \mu_{\gamma_2}). \quad (2.52)$$

It is worth to say, that the viscous coefficients for this matrix are chosen experimentally. The load's coefficients were computed through observation in simulations. On the other hand, the coefficients for the servomotor are calculated through experiments made with real devices. Furthermore, the rest of the coefficients are assumed to be null.

2.5 STATE-SPACE REPRESENTATION OF THE SYSTEM

In order to represent the dynamic model of the Tilt-rotor UAV with suspended load in the state space form, twenty states are defined:

$$\mathbf{x}_s(t) \triangleq \begin{bmatrix} x_1 \\ \vdots \\ x_{20} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix}. \quad (2.53)$$

By isolating $\ddot{\mathbf{q}}$ in (2.16), the system can be rewritten in the following form:

$$\ddot{\mathbf{q}} = \mathbf{M}(\mathbf{q})^{-1}[\mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{F}_{ext} - (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\mu})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})] \quad (2.54)$$

Thus, system (2.54) can be represented in the state-space form $\dot{\mathbf{x}}_s(t) = f(\mathbf{x}_s(t), \mathbf{u}(t))$. As a result, the non-linear state space representation is given by:

$$\dot{\mathbf{x}}_s(t) = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{M}(\mathbf{q})^{-1}[\mathbf{B}(\mathbf{q})\mathbf{u} + \mathbf{F}_{ext} - (\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \boldsymbol{\mu})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q})] \end{bmatrix} \quad (2.55)$$

Assuming $\dot{\mathbf{x}}_s(t) = 0$, the operating point $(\mathbf{u}^*, \mathbf{q}^*, \dot{\mathbf{q}}^*)$ can be calculated through the following equality:

$$\begin{bmatrix} \dot{\mathbf{q}}^* \\ \mathbf{M}(\mathbf{q}^*)^{-1}[\mathbf{B}(\mathbf{q}^*)\mathbf{u}^* - (\mathbf{C}(\mathbf{q}^*, \dot{\mathbf{q}}^*) + \boldsymbol{\mu})\dot{\mathbf{q}}^* - \mathbf{G}(\mathbf{q}^*)] \end{bmatrix} = 0. \quad (2.56)$$

2.6 SYSTEM DESIGN PARAMETERS

The system parameters of the Tilt-rotor UAV with suspended load are shown in Table 1. Most of the parameters were obtained from a real implementation of the UAV developed into the ProVANT project, being, the inertia tensor of the bodies only calculated by the software *SolidWorks*[®], using a computer aided design (CAD) of the Tilt-rotor UAV. On the other hand, considering the parameters presented in Table 1, the equilibrium point of the system is calculated with the equation (2.56), where the the yaw angle is assumed as $\psi_r(t) = 0$. These values are presented in the Table 2.

For the viscous coefficient matrix $\boldsymbol{\mu}$, it is considered a viscous friction in the load $(\mu_{\gamma_1}, \mu_{\gamma_1})$ and servos $(\mu_{\alpha_r}, \mu_{\alpha_l})$. In the case of

Table 1 – Tilt-rotor system parameters

Parameter	Value
m_1	1.54727 K_g
m_2, m_3	0.22637 K_g
m_4	0.088 K_g
$d_{C_1}^B$	$[0.00139 \quad -0.00033 \quad -0.083]^T m$
$d_{C_2}^B$	$[0 \quad -0.24673 \quad 0.0069]^T m$
$d_{C_3}^B$	$[0 \quad 0.24673 \quad 0.0069]^T m$
I_{xx}^1	0.01739447464 $K_g \cdot m^2$
I_{yy}^1	0.00572626786 $K_g \cdot m^2$
I_{zz}^1	0.02734745713 $K_g \cdot m^2$
I_{xx}^2, I_{xx}^3	0.00011099565 $K_g \cdot m^2$
I_{yy}^2, I_{yy}^3	0.021978 $K_g \cdot m^2$
I_{zz}^2, I_{zz}^3	0.00084170622 $K_g \cdot m^2$
I_{xx}^4	0.00003666441 $K_g \cdot m^2$
I_{yy}^4	0.00003666441 $K_g \cdot m^2$
I_{zz}^4	0.00003667706 $K_g \cdot m^2$
k_τ	1.54727 $N \cdot m \cdot s^2$
b	0.22637 $N \cdot s^2$
g_z	9.8 m/s^2
β	5°
l	0.5 m

Table 2 – Tilt-rotor equilibrium point

State	Value
$\phi^*(t)$	0.0000847216 rad
$\theta^*(t)$	0.0153948 rad
$\alpha_r^*(t)$	-0.0153214 rad
$\alpha_l^*(t)$	-0.015351 rad
$\gamma_1^*(t)$	-0.0000847015 rad
$\gamma_2^*(t)$	-0.0153948 rad
$f_r^*(t)$	10288.3 N
$f_l^*(t)$	10268.4 N
$\tau_r^*(t)$	0 $N \cdot m$
$\tau_l^*(t)$	0 $N \cdot m$

the load, the parameters were chosen through the observation of the simulation. In the case of servomotors, some experiments were executed to compute the coefficients, considering the local dynamic of them into the Tilt-rotor UAV model.

Taking into account the right rotor dynamic in equation (2.35), it is possible to say that it is coupled only with the attitude dynamic in the Tilt-rotor UAV model. Consequently, the Coriolis and centripetal matrix $\mathbf{C}(\mathbf{q})$ will present the same behavior since this matrix is derived from inertia matrix $\mathbf{M}(\mathbf{q})$. In this context, with the objective to isolate the local dynamics of right servomotor, the coupled dynamics of the attitude can be neglected, giving as a result:

$$\mathbf{a}^T \mathbf{I}_2 \mathbf{a} \ddot{\alpha}_r + \mu_{\alpha_r} \dot{\alpha}_r = \tau_r. \quad (2.57)$$

Then, applying the Laplace transform to this system, yields to:

$$\frac{\alpha_r(s)}{u(s)} = \frac{\frac{1}{\mu_{\alpha_r}}}{\frac{I_{yy}^2}{\mu_{\alpha_r}} s + 1} = \frac{K_r}{t_o s + 1}, \quad (2.58)$$

where t_o is the time when the system reaches 63.2% of its steady-state value and K_r is the static gain. From the experiments performed with the servomotor, considering the load of the rotors, the parameter t_o was identified as $0.072s$ and the parameter $K_r = 3.2803 \frac{rad}{s}$. Thus, considering equation (2.58), the value of the inertia tensor (I_{yy}^2 , I_{yy}^3) and viscous friction are calculated. The drag forces matrix is given by:

$$\boldsymbol{\mu} = \text{diag}(0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0.30485 \ 0.30485 \ 0.005 \ 0.005). \quad (2.59)$$

2.7 LINEARIZATION OF THE TILT-ROTOR UAV WITH SUSPENDED LOAD MODEL

In this section the equations of motion are linearized around a generic desired trajectory in order to obtain a linear model of the tilt-rotor UAV with suspended load. This system is underactuated, since it has only four control signals and ten degrees of freedom. Hence, it is only possible to track reference trajectories of four degrees of freedom, and the remaining DOFs must be stabilized around the equilibrium point calculated in (2.56). Thus, the generic reference

vector is given by:

$$\mathbf{x}_{sr}(t) = [x_r(t) \quad y_r(t) \quad z_r(t) \quad \phi^* \quad \theta^* \quad \psi_r(t) \quad \alpha_r^* \quad \alpha_l^* \quad \gamma_1^* \quad \gamma_2^* \quad \dot{x}_r(t) \quad \dot{y}_r(t) \quad \dot{z}_r(t) \quad 0 \quad 0 \quad \dot{\psi}_r(t) \quad 0 \quad 0 \quad 0 \quad 0]^T. \quad (2.60)$$

By using the reference state vector (2.60) and isolating \mathbf{u} from the equation (2.16), assuming $\mathbf{F}_{ext} = 0$ the reference control vector $\mathbf{u}_r(t) = [f_r^* \quad f_l^* \quad \tau_r^* \quad \tau_l^*]$ is computed as follows:

$$\mathbf{u}_r(t) = \mathbf{B}(\mathbf{q}_r)^+ (\mathbf{M}(\mathbf{q}_r) \ddot{\mathbf{q}}_r + [\mathbf{C}(\mathbf{q}_r, \dot{\mathbf{q}}_r) + \boldsymbol{\mu}] \dot{\mathbf{q}}_r + \mathbf{G}(\mathbf{q}_r)).^1 \quad (2.61)$$

By defining the error state vector as $\tilde{\mathbf{x}}_s(t) = \mathbf{x}_s(t) - \mathbf{x}_{sr}(t)$, the error control vector as $\tilde{\mathbf{u}}(t) = \mathbf{u}(t) - \mathbf{u}_r(t)$ and calculating the terms of first order of the Taylor's series from equation (2.55), it is possible to obtain the error linear model:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_s(t) &= \mathbf{A}(t) \tilde{\mathbf{x}}_s(t) + \mathbf{B} \tilde{\mathbf{u}}(t) \\ \tilde{\mathbf{y}}_s(t) &= \mathbf{C} \tilde{\mathbf{x}}_s(t) \end{aligned} \quad (2.62)$$

where:

$$\mathbf{A}(t) = \left. \frac{\partial f(\mathbf{x}_s, \mathbf{u})}{\partial \mathbf{x}_s} \right|_{\substack{\mathbf{x}_s = \mathbf{x}_{sr} \\ \mathbf{u} = \mathbf{u}_r}}, \quad \mathbf{B} = \left. \frac{\partial f(\mathbf{x}_s, \mathbf{u})}{\partial \mathbf{u}} \right|_{\substack{\mathbf{x}_s = \mathbf{x}_{sr} \\ \mathbf{u} = \mathbf{u}_r}},$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}.$$

It should be noted that matrix $\mathbf{A}(t)$ in the linear error model varies with time (see appendix A.1). Therefore, the error state space system (2.62) is considered a linear time varying model (LTV).

With the objective of performing the path tracking and rejecting constant disturbances, the error state space vector is extended with the integral action of the error states x , y , z and the yaw angle. Thus, the new augmented system is given by:

¹+ means the pseudo inverse operator of a matrix.

$$\dot{\tilde{\mathbf{x}}}_s(t) = \mathbf{A}_a(t)\tilde{\mathbf{x}}_s(t) + \mathbf{B}_a\tilde{\mathbf{u}}(t) \quad (2.63)$$

with:

$$\mathbf{A}_a(t) = \begin{bmatrix} \mathbf{A}(t) & \mathbf{0} \\ \mathbf{C} & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

Consequently, the augmented error states vector is:

$$\tilde{\mathbf{x}}_s(t) = \begin{bmatrix} \mathbf{x}_s(t) - \mathbf{x}_{sr}(t) \\ \int(x(t) - x_r(t)) \\ \int(y(t) - y_r(t)) \\ \int(z(t) - z_r(t)) \\ \int(\psi(t) - \psi_r(t)) \end{bmatrix}. \quad (2.64)$$

In order to use this model in the control predictive formulation presented in Chapter 4, it is discretized through Euler's approximation:

$$\tilde{\mathbf{x}}_s(k+1) = \mathbf{A}_z(k)\tilde{\mathbf{x}}_s(k) + \mathbf{B}_z\tilde{\mathbf{u}}(k) \quad (2.65)$$

where $\mathbf{A}_z(k) = \mathbf{I} - \mathbf{A}_a(t) t_s$, $\mathbf{B}_z = \mathbf{B}_a \cdot t_s$, and t_s is the sampling time chosen twenty times faster than the fastest dynamic of the Tilt-rotor UAV. It should be note, that tilttable servomotors' dynamics are the fastest ones.

2.8 SUMMARY

In this chapter the dynamic model with ten degrees of freedom of the Tilt-rotor UAV with suspended load was presented. The model considers the dynamics generated by the rotor when tilting. Additionally, this non-linear model was linearized around a predefined trajectory reference, yielding to a LTV model. In the next chapter, considering that the linear model of the Tilt-rotor UAV is a LTV system, it is introduced two approaches of model predictive control design based on a generic LTV system. Then, the specific case for the Tilt-rotor UAV with suspended load is developed.

3 LINEAR TIME VARYING MODEL PREDICTIVE CONTROL

In the previous chapter, a dynamic model was presented for the Tilt-rotor UAV with a suspended load. The modeling procedure and the linearization of this system was outlined. In this chapter, two different strategies of linear predictive controller are described in order to perform path tracking of the Tilt-rotor aircraft. As the linearized model obtained in Section 2.7 is a time varying linear error model, the controllers must be based on this kind of systems.

However, before proceeding towards control methods for the specific case of the Tilt-rotor UAV with suspended load, the predictive control formulations based on a linear time variant model are described. The MPC strategy, presented in Section 3.1, is based on an incremental prediction model, which uses the last control signal applied to the system for calculating the new control action. Section 3.2 presents the second strategy used in this work, which is based on a non-incremental prediction model. In addition, it is described the objective function used for this strategy, and the constraint formulation is depicted.

Finally, in order to guaranty stability and reduce the prediction and control horizon, in Section 3.3, a terminal cost function is presented for the two controllers.

3.1 INCREMENTAL MODEL PREDICTIVE CONTROL (I-SSMPC)

In this section, an incremental prediction model based on a state space system (I-SSMPC) is presented. This model is based on the LTV model presented in Raffo (2011), which was used to obtain a non incremental prediction model. Furthermore, as presented in Camacho & Alba (2013), an incremental prediction model can be obtained considering the variation of the control action. Then, this variation is calculated via an optimization problem, where a cost function is solved assuming input and output constraints.

3.1.1 Linear Prediction Model

Assuming a known reference trajectory at any time in the future, and linearizing the system throughout this reference, it is possible to obtain a linear time varying system that will be used in the prediction

model. Moreover, assuming that states can be measured at any instant k , the system can be discretized and expressed as:

$$\mathbf{x}(k+1) = \mathbf{A}_z(k)\mathbf{x}(k) + \mathbf{B}_z(k)\mathbf{u}(k). \quad (3.1)$$

This section deals with the incremental model predictive control formulation presented in Camacho & Alba (2013). The incremental prediction model is obtained by defining the increment of the control signal as $\Delta\mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$ and augmenting the space state vector of the linear discrete model (3.1) as $\mathbf{x}_a(k) = [\mathbf{x}^T(k) \quad \mathbf{u}^T(k-1)]^T$, where $\mathbf{u}(k-1)$ is the past control signal. In this context, the system takes the following form:

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_z(k) & \mathbf{B}_z(k) \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_z(k) \\ \mathbf{I} \end{bmatrix} \Delta\mathbf{u}(k), \quad (3.2)$$

which, in its compact form, is given by:

$$\mathbf{x}_a(k+1) = \mathbf{A}_a(k)\mathbf{x}_a(k) + \mathbf{B}_a(k)\Delta\mathbf{u}(k). \quad (3.3)$$

Therefore, the predicted state of the model in any instant j could be calculated recursively with $\mathbf{x}_a(k+j)$, being j from 1 to N , where N is the prediction horizon for all variables. Furthermore, when the prediction horizon and the control horizon M are equal, the prediction model can be expressed by:

$$\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{u}} + \mathbf{Q}\mathbf{x}_a(k), \quad (3.4)$$

where:

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{B}_a(k) & 0 & 0 & 0 \\ \mathbf{\Lambda}(k, 2, 1)\mathbf{B}_a(k) & \mathbf{B}_a(k+1) & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Lambda}(k, N-1, 1)\mathbf{B}_a(k) & \mathbf{\Lambda}(k, N-1, 2)\mathbf{B}_a(k+1) & \dots & 0 \\ \mathbf{\Lambda}(k, N, 1)\mathbf{B}_a(k) & \mathbf{\Lambda}(k, N, 2)\mathbf{B}_a(k+1) & \dots & \mathbf{B}_a(k+N) \end{bmatrix}, \quad (3.5)$$

$$\mathbf{Q}(k|k) = \begin{bmatrix} \mathbf{\Lambda}(k, 0, 0) \\ \mathbf{\Lambda}(k, 0, 2) \\ \vdots \\ \mathbf{\Lambda}(k, 0, N-2) \\ \mathbf{\Lambda}(k, 0, N-1) \end{bmatrix}. \quad (3.6)$$

On the other hand, when the control horizon is different ($M < N$), the matrix \mathbf{P} must be rewritten, adding to the last column N the terms created with the remaining prediction horizon. The term $\mathbf{\Lambda}(k, i, j)$ is defined by:

$$\mathbf{\Lambda}(k, i, j) = \begin{cases} 0 & \text{if } i \leq j \\ \prod_{l=j}^{i-1} \mathbf{A}_a(k+l) & \text{if } i > j \end{cases}. \quad (3.7)$$

The predicted state vector, $\hat{\mathbf{x}}$, and the predicted control vector $\hat{\mathbf{u}}$ are given by:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_a(k+1|k) \\ \hat{\mathbf{x}}_a(k+2|k) \\ \hat{\mathbf{x}}_a(k+3|k) \\ \vdots \\ \hat{\mathbf{x}}_a(k+N|k) \end{bmatrix}, \quad \hat{\mathbf{u}} = \begin{bmatrix} \Delta \hat{\mathbf{u}}(k|k) \\ \Delta \hat{\mathbf{u}}(k+1|k) \\ \Delta \hat{\mathbf{u}}(k+2|k) \\ \vdots \\ \Delta \hat{\mathbf{u}}(k+M-1|k) \end{bmatrix}. \quad (3.8)$$

3.1.2 Objective Function

The benefit of MPC is to calculate an optimal control action by minimizing a given cost function. Several MPC approaches propose different cost functions, but according to Rawlings & Mayne (2009), a generalized cost function can be expressed as follows:

$$V(\mathbf{x}_a(k), \mathbf{u}(k)) = \sum_{i=0}^{N-1} L[\mathbf{x}_a(k+i|k), \mathbf{u}(k+i|k)] + F[\mathbf{x}_a(k+N|k)], \quad (3.9)$$

where $L[x, u]$ is called the stage cost and $F[x]$ is known as the terminal cost. The general aim of the stage cost is to ensure that the future state follows a determined reference signal on the considered horizon with a minimum of energy spent by the control input. The terminal cost is used to guarantee stability of the system in a finite horizon and

it will be treated in Section 3.3.

The generalized stage cost function can be expressed as (CAMACHO; ALBA, 2013):

$$\begin{aligned}
 L(\mathbf{x}_a(k), \mathbf{u}(k)) &= \sum_{j=1}^N ([\mathbf{x}_a(k+j|k) - \mathbf{x}_r(k+j|k)]^T \boldsymbol{\Sigma}_\rho [\mathbf{x}_a(k+j|k) \\
 &\quad - \mathbf{x}_r(k+j|k)]) + \sum_{j=0}^{M-1} ([\Delta \mathbf{u}(k+j|k)]^T \boldsymbol{\Sigma}_\lambda [\Delta \mathbf{u}(k+j|k)]),
 \end{aligned} \tag{3.10}$$

where $\mathbf{x}_r(k+j)$ is the known future reference of the aircraft at instant $k+j$. Expressing this function in a matrix form the following objective function is obtained:

$$J = [\hat{\mathbf{x}} - \hat{\mathbf{x}}_r]^T \mathbf{W}_y [\hat{\mathbf{x}} - \hat{\mathbf{x}}_r] + \hat{\mathbf{u}}^T \mathbf{W}_u \hat{\mathbf{u}}, \tag{3.11}$$

where the state weighting matrix \mathbf{W}_y is a diagonal matrix with N blocks, where each block $\boldsymbol{\Sigma}_\rho$ is given by:

$$\boldsymbol{\Sigma}_\rho = \begin{bmatrix} \rho_1 & 0 & \dots & 0 \\ 0 & \rho_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \rho_i \end{bmatrix}. \tag{3.12}$$

In this equation, ρ_i is the weight for each state and the number of states is represented by i .

In the same way, the input weighting matrix \mathbf{W}_u is a diagonal matrix with M blocks, where each block $\boldsymbol{\Sigma}_\lambda$ is given by:

$$\boldsymbol{\Sigma}_\lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_p \end{bmatrix}, \tag{3.13}$$

where λ_i is the weight for each input and the number of system inputs is represented by p . In equation (3.10), the pre-defined reference trajectory vector is expressed as:

$$\hat{\mathbf{x}}_r = \begin{bmatrix} \hat{\mathbf{x}}_r(k+1|k) \\ \hat{\mathbf{x}}_r(k+2|k) \\ \vdots \\ \hat{\mathbf{x}}_r(k+N|k) \end{bmatrix}. \quad (3.14)$$

3.1.3 Computing the control law

As mentioned before, in the standard MPC, the control action is computed by optimizing a cost function. However, it must be considered the prediction model of the system to be controlled. Thus, replacing the prediction model (3.4) into the objective function (3.11), the cost function to be minimized is written as follows:

$$J = \hat{\mathbf{u}}^T [\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u] \hat{\mathbf{u}} + 2[(\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y \mathbf{P}] \hat{\mathbf{u}} \quad (3.15) \\ + (\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y (\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r).$$

3.1.3.1 Quadratic Programming Formulation

When constraints are considered, in a standard MPC, the control action is computed at each sample time solving a quadratic programming (QP) problem. Therefore, the cost function (3.15) must be rewritten in the following form:

$$J = \frac{1}{2} \hat{\mathbf{u}}^T \mathcal{G} \hat{\mathbf{u}} + f' \hat{\mathbf{u}} + c \quad (3.16) \\ \text{Subject to : } \mathbf{A} \mathbf{r} \cdot \hat{\mathbf{u}} < \mathbf{b} \mathbf{r} \text{ ,}$$

where:

$$\frac{1}{2} \mathcal{G} = \mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u, \\ f' = 2(\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y \mathbf{P}, \\ c = (\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y (\mathbf{Q} \mathbf{x}_a(k) - \hat{\mathbf{x}}_r), \\ \mathbf{A} \mathbf{r}, \mathbf{b} \mathbf{r} = \text{Constraints matrices.}$$

Generally, the constraints are chosen in function of the admissible values that the system can perform. In the case of input constraints, saturation in the control signal can be considered with the maximum and minimum values of the constraint function. Thereby,

the general expression for an input constraint is presented as:

$$\mathbf{u}_{min} \leq \mathbf{u}(k + j|k) \leq \mathbf{u}_{max} \quad \text{where } j = 0 \cdots M - 1, \quad (3.17)$$

where \mathbf{u}_{min} and \mathbf{u}_{max} are the minimum and maximum values for the control signal that can be applied to the actuator, respectively. However, as the prediction model is in the incremental form, the control action applied to the system is the increment control signal at the instant k , added to the previous control action. In this way, the equation (3.17) can be rewritten with the expression $\mathbf{u}(k + j|k) = \mathbf{u}(k + j - 1|k) + \Delta\mathbf{u}(k + j|k)$ as presented below:

$$\mathbf{u}_{min} \leq \mathbf{u}(k + j - 1|k) + \Delta\mathbf{u}(k + j|k) \leq \mathbf{u}_{max}. \quad (3.18)$$

For applying this constraint in the QP formulation, it is necessary to expand it recursively throughout the control horizon and transform in a matrix form. Therefore, the constraint is rewritten as:

$$\hat{\mathbf{u}}_{min} - \bar{\mathbf{u}}(k - 1) \leq \mathbf{T}_m \hat{\mathbf{u}} \leq \hat{\mathbf{u}}_{max} - \bar{\mathbf{u}}(k - 1), \quad (3.19)$$

where \mathbf{T}_m is a lower triangular matrix of identities, $\hat{\mathbf{u}}_{min}$ and $\hat{\mathbf{u}}_{max}$ are vectors with M copies of minimum and maximum values of the control signal. Similarly, $\bar{\mathbf{u}}(k - 1)$ is a vector with M copies of the previous control action applied to the system. Equation (3.19) can be rewritten in the matrix form as:

$$\begin{bmatrix} \mathbf{T}_m \\ -\mathbf{T}_m \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\mathbf{u}}_{max} - \bar{\mathbf{u}}(k - 1) \\ \bar{\mathbf{u}}(k - 1) - \hat{\mathbf{u}}_{min} \end{bmatrix}. \quad (3.20)$$

It is also assumed that the UAV must track the trajectory into a confined environment, which can be represented by constraints in the states. The general form of this kind of constraint is expressed by:

$$\hat{\mathbf{x}}_{amin} \leq \hat{\mathbf{x}}_a(k + i|k) \leq \hat{\mathbf{x}}_{amax} \quad , \quad \text{where } i = 1 \cdots N. \quad (3.21)$$

In the same way as presented for the input constraints, the equation (3.21) must be expanded throughout the prediction horizon and rewritten in a matrix form to be included in the QP formulation. However, as explained before, the expansion of $\mathbf{x}_a(k + i|k)$ results in the predictive model presented in (3.4). Considering this, the equation takes the following form:

$$\hat{\mathbf{x}}_{min} \leq \mathbf{P}\hat{\mathbf{u}} + \mathbf{Q}\mathbf{x}_a(k) \leq \hat{\mathbf{x}}_{max} \quad (3.22)$$

where $\hat{\mathbf{x}}_{min}$ and $\hat{\mathbf{x}}_{max}$ are vectors with N entries of maximum and minimum values of the states in the system. By rewriting this inequality in the matrix form, yields to:

$$\begin{bmatrix} \mathbf{P} \\ -\mathbf{P} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\mathbf{x}}_{max} - \mathbf{Q}\mathbf{x}_a(k) \\ \mathbf{Q}\mathbf{x}_a(k) - \hat{\mathbf{x}}_{min} \end{bmatrix}, \quad (3.23)$$

which can be used in the QP formulation.

To solve the QP problem subject to input and state constraints presented in equation (3.16), it is necessary to create the matrices A_r and B_r using the inequalities (3.20) and (3.23) as follows:

$$\mathbf{A}_r = \begin{bmatrix} \mathbf{T}_m \\ -\mathbf{T}_m \\ \mathbf{P} \\ -\mathbf{P} \end{bmatrix}, \quad \mathbf{b}_r = \begin{bmatrix} \hat{\mathbf{u}}_{max} - \bar{\mathbf{u}}(k-1) \\ \bar{\mathbf{u}}(k-1) - \hat{\mathbf{u}}_{min} \\ \hat{\mathbf{x}}_{max} - \mathbf{Q}\mathbf{x}_a(k) \\ \mathbf{Q}\mathbf{x}_a(k) - \hat{\mathbf{x}}_{min} \end{bmatrix}.$$

3.1.3.2 Unconstrained Problem

In this section, the constraint free case is considered. As mentioned in Rossiter (2004) and Maciejowski (2002), the control variation could be calculated analytically by finding the gradient of equation (3.16), which is calculated as:

$$\frac{dJ}{d\hat{\mathbf{u}}} = 2\mathcal{G}\hat{\mathbf{u}} + 2f' = 0 \quad (3.24)$$

Thus, isolating $\hat{\mathbf{u}}$ from equation (3.24), the control action could be calculated analytically by:

$$\hat{\mathbf{u}} = (\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u)^{-1} \mathbf{P}^T \mathbf{W}_y (\hat{\mathbf{x}}_r - \mathbf{Q}\mathbf{x}_a(k)). \quad (3.25)$$

Although this expression contains the future control actions for the whole control horizon, only the first p elements corresponding to the control increment for the instant k are used. Since the predictive control law is in the incremental form, it is necessary to add the previous control signal $\mathbf{u}(k-1)$ as follows:

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta\hat{\mathbf{u}}(k). \quad (3.26)$$

3.2 NON INCREMENTAL MODEL PREDICTIVE CONTROL (NI-SSMPC)

In this section, another predictive formulation based on Raffo, Ortega & Rubio (2011b) is presented, where it considers a prediction model in a non-incremental form. Furthermore, a new objective function is presented for this type of MPC formulation, assuming both unconstrained and constrained formulations.

3.2.1 Linear Predictive Model

The main feature of this formulation is that it does not use an augmented system with the last control action, as presented before. The linear time varying system used in this formulation is presented in Equation (3.1).

The predicted states for an instant j is defined by $\mathbf{x}(k+j|k)$, with $j = 1, 2, \dots, N$, where N is the prediction horizon for all state variables. Therefore, if this definition is expanded recursively throughout the prediction horizon, the predictive model can be computed as:

$$\hat{\mathbf{x}} = \mathbf{P}\hat{\mathbf{u}} + \mathbf{Q}\mathbf{x}(k). \quad (3.27)$$

Matrices \mathbf{P} and \mathbf{Q} are obtained similar to matrices (3.5) and (3.6), but now using system (3.1). The predicted state vector, $\hat{\mathbf{x}}$, and the predicted control vector, $\hat{\mathbf{u}}$, are represented by:

$$\hat{\mathbf{x}} = \begin{bmatrix} \mathbf{x}(k+1|k) \\ \mathbf{x}(k+2|k) \\ \mathbf{x}(k+3|k) \\ \vdots \\ \mathbf{x}(k+N|k) \end{bmatrix}, \quad \hat{\mathbf{u}} = \begin{bmatrix} \mathbf{u}(k|k) \\ \mathbf{u}(k+1|k) \\ \mathbf{u}(k+2|k) \\ \vdots \\ \mathbf{u}(k+M-1|k) \end{bmatrix}. \quad (3.28)$$

3.2.2 Objective Function

Assuming that the reference states and reference control are known, the generalized stage cost function for this kind of predictive controller is given by:

$$J = [\hat{\mathbf{x}} - \hat{\mathbf{x}}_r]^T \mathbf{W}_y [\hat{\mathbf{x}} - \hat{\mathbf{x}}_r] + [\hat{\mathbf{u}} - \hat{\mathbf{u}}_r]^T \mathbf{W}_u [\hat{\mathbf{u}} - \hat{\mathbf{u}}_r], \quad (3.29)$$

where the input weighting matrix \mathbf{W}_y and \mathbf{W}_u are the same matrices presented in (3.13) and (3.12), respectively. Additionally, the pre-defined reference trajectory vector and the future control reference are:

$$\hat{\mathbf{x}}_r = \begin{bmatrix} \mathbf{x}_r(k+1|k) \\ \mathbf{x}_r(k+2|k) \\ \vdots \\ \mathbf{x}_r(k+N|k) \end{bmatrix}, \quad \hat{\mathbf{u}}_r = \begin{bmatrix} \mathbf{u}_r(k|k) \\ \mathbf{u}_r(k+1|k) \\ \vdots \\ \mathbf{u}_r(k+M-1|k) \end{bmatrix}. \quad (3.30)$$

According with Rossiter (2004), this equation differs from (3.9) because the weight of the inputs optimizes the distance from steady state rather than incremental changes of the control signal.

3.2.3 Computing the control law

Considering the prediction model (3.27), and replacing it into the objective function (3.29), the cost function used to calculate the control action is written as follows:

$$\begin{aligned} J = & \hat{\mathbf{u}}^T [\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u] \hat{\mathbf{u}} + 2[(\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y \mathbf{P} - \hat{\mathbf{u}}_r^T \mathbf{W}_u] \hat{\mathbf{u}} \\ & + (\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y (\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r) + \hat{\mathbf{u}}_r^T \mathbf{W}_u \hat{\mathbf{u}}_r. \end{aligned} \quad (3.31)$$

3.2.3.1 Quadratic Programming Formulation

Assuming the constrained problem, the predictive control law is computed by an optimization algorithm, which in this work is solved through the quadratic programming formulation. Therefore, the cost function (??) must be rewritten in a proper form as follows:

$$\begin{aligned} J = & \frac{1}{2} \hat{\mathbf{u}}^T \mathcal{G} \hat{\mathbf{u}} + f' \hat{\mathbf{u}} + c \\ \text{Subjected to : } & \mathbf{A}_r \hat{\mathbf{u}} \leq \mathbf{b}_r, \end{aligned} \quad (3.32)$$

where:

$$\begin{aligned}\frac{1}{2}\mathcal{G} &= \mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u, \\ f' &= 2((\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y \mathbf{P} - \hat{\mathbf{u}}_r^T \mathbf{W}_u), \\ c &= (\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r)^T \mathbf{W}_y (\mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_r) + \hat{\mathbf{u}}_r^T \mathbf{W}_u \hat{\mathbf{u}}_r, \\ \mathbf{A}_r, \mathbf{b}_r &= \text{Constraints matrices.}\end{aligned}$$

The constraint, for the QP formulation, was chosen in function of physical limitations. In the case of input constraints, it is considered saturation in the input control signals, representing the maximum and minimum values applied to the actuators of the Tilt-rotor UAV. Thereby, the general expression for a input constraint is presented as:

$$\mathbf{u}_{min} \leq \mathbf{u}(k + j|k) \leq \mathbf{u}_{max} \quad \text{where} \quad j = 0 \cdots M - 1, \quad (3.33)$$

where u_{min} and u_{max} are the minimum and maximum value for the control signal that can be applied to the actuator, respectively. In this case, the prediction model is in the non-incremental form, thus the control action is applied directly to the system.

In the case of state constraints, the constraint has the same form as presented in section 3.1. To solve the QP formulation subjected to input and state constraints presented in equation (3.32), it is necessary to create matrices A_r and B_r as follows:

$$\mathbf{A}_r = \begin{bmatrix} \mathbf{I}_m \\ -\mathbf{I}_m \\ \mathbf{P} \\ -\mathbf{P} \end{bmatrix}, \quad \mathbf{b}_r = \begin{bmatrix} \hat{\mathbf{u}}_{max} \\ -\hat{\mathbf{u}}_{min} \\ \hat{\mathbf{x}}_{max} - \mathbf{Q}\mathbf{x}(k) \\ \mathbf{Q}\mathbf{x}(k) - \hat{\mathbf{x}}_{max} \end{bmatrix}.$$

3.2.3.2 Unconstrained Formulation

As mentioned before, in the absence of constraints, the control variation could be computed using equation (3.32), when the control action can be computed by solving the equation $2\mathcal{G}\hat{\mathbf{u}} + 2f' = 0$ for $\hat{\mathbf{u}}$ with the terms \mathcal{G} and f' presented in (3.32). Thus, $\hat{\mathbf{u}}$ is given by:

$$\hat{\mathbf{u}} = (\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u)^{-1} (\mathbf{P}^T \mathbf{W}_y (\hat{\mathbf{x}}_r - \mathbf{Q}\mathbf{x}(k)) + \mathbf{W}_u \hat{\mathbf{u}}_r), \quad (3.34)$$

Taking into account that the discrete model is in a non-incremental form, the applied control input $\mathbf{u}(k)$ to the system is given directly by the first p terms of the vector $\hat{\mathbf{u}}$ as presented below:

$$\mathbf{u}(k) = \hat{\mathbf{u}}(k|k). \quad (3.35)$$

3.3 TERMINAL COST

In order to guarantee stability, there are several approaches in the literature. According to Maciejowski (2002), the most simple way to ensure stability is making the prediction horizon large enough or even infinite. Another kind of approach to ensure stability is using the optimal control function as a Lyapunov function.

By adding a terminal constraint, which can be used with any length of prediction horizon, the state is forced to take a particular value at end of the prediction horizon, but it has some drawbacks. One of those assumes that the optimization problem and the global optimum solution have a solution at each step. Another drawback is with respect to the hard complexity to solve the constraint optimization problem and adding a terminal constraint may transform it in an infeasible problem.

However, there exists the Terminal Constraint set formulation, which is a relaxation of the terminal constraint, where the MPC drives the state into a set X_0 and then switch to some other control law that guarantee stability for an initial condition within X_0 . On the other hand, it is possible to guarantee stability with finite horizon even without explicit terminal constraint. The predictive control problem is associated with the Riccati equation, which is intimately related to the optimal value of the cost function. In this way, according to Rawlings & Mayne (2009), it is possible to ensure closed-loop stability by adding a terminal cost function without the use of a terminal constraint.

In this section, a proposal to guarantee stability in a finite horizon MPC and to decrease the prediction horizon for a time varying system is presented.

3.3.1 Terminal Stage

Equation (3.9) presented a general cost function for a model predictive control. This cost function has two principal parts: the stage cost that was presented in (3.15) and (3.31) for a incremental and non

incremental predictive controller, respectively; and, the second part is the terminal cost, which is calculated here, based on Rawlings & Mayne (2009), but extended for time varying systems. The general expression for a terminal cost is expressed as follows:

$$F[\mathbf{x}(k+N)] = [\delta\mathbf{x}(k+N|k)]^T \mathbf{L} [\delta\mathbf{x}(k+N|k)] \quad (3.36)$$

where $\delta\mathbf{x}(k+N|k)$ is the error between the state vector and its desired value at step $k+N$ which is given as $\delta\mathbf{x}(k+N|k) = \mathbf{x}_a(k+N) - \mathbf{x}_r(k+N)$ for the incremental prediction model, and $\delta\mathbf{x}(k+N|k) = \mathbf{x}(k+N) - \mathbf{x}_r(k+N)$ for the non-incremental predictive model. The matrix \mathbf{L} is the terminal value, which is calculated as a Lyapunov matrix solving the Riccati equation for a time-invariant systems:

$$(\mathbf{A}_z + \mathbf{B}_z \mathbf{K}) \mathbf{L} (\mathbf{A}_z + \mathbf{B}_z \mathbf{K})^T - \mathbf{L} + \mathbf{W}_y + \mathbf{K}^T \mathbf{W}_u \mathbf{K} \leq 0 \quad (3.37)$$

However, in this work the discrete model used for a model prediction formulation is a time varying model and, for this reason, equation (3.37) is not applicable. In this way, according to Trofino, Coutinho & Barbosa (2003), it is possible to create a similar equation to the LQR control problem using a LMI formulation, where the stability condition is satisfied for all linear models inside the polytope, that is defined by the uncertainty model. In the other hand, as the approach presented by Trofino, Coutinho & Barbosa (2003) is developed for continuous systems, an extension for discrete systems is presented in what follows.

First, consider the discrete system:

$$\mathbf{x}(k+1) = \mathbf{A}(\alpha)\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \quad (3.38)$$

$$\mathbf{z}(k) = \mathbf{C}_z\mathbf{x}(k) + \mathbf{D}_z\mathbf{u}(k) \quad (3.39)$$

$$\mathbf{u}(k) = \mathbf{K}\mathbf{x}(k), \quad (3.40)$$

where α is the uncertainty of the system, $\mathbf{z}(k)$ is an auxiliary vector with the energy to be minimized, and \mathbf{C}_z and \mathbf{D}_z are constant weight matrices arbitrarily chosen with an appropriate size. It should be mentioned, for the case of the Tilt-rotor UAV, the uncertainties are given by the variation of trajectory accelerations ($\ddot{x}_r, \ddot{y}_r, \ddot{z}_r$).

Then, consider a cost function associated to the auxiliary vector as follows:

$$V(z) = \min_{\mathbf{u}(k)} \sum_{k=0}^{\infty} \mathbf{z}(k)^T \mathbf{z}(k), \quad (3.41)$$

where $\mathbf{z}(k)^T \mathbf{z}(k)$ will be always positive for any $\mathbf{z}(k) \neq 0$. Using the Lyapunov function $V(\mathbf{x}(k)) = \mathbf{x}(k)^T \mathbf{L} \mathbf{x}(k)$ being $\mathbf{L} = \mathbf{L}^T > 0$ a Lyapunov matrix that stabilizes asymptotically the system. The Lyapunov inequality can be defined as follows:

$$\Delta V(\mathbf{x}(k)) + \mathbf{z}^T \mathbf{z} < 0 \quad (3.42)$$

where $\Delta V(\mathbf{x}(k))$ is the increment of the quadratic Lyapunov function. Furthermore, considering the closed loop system given by $\mathbf{x}(k+1) = (\mathbf{A}(\alpha) + \mathbf{BK})\mathbf{x}(k)$ and $\mathbf{z}(k) = (\mathbf{C}_z + \mathbf{D}_z \mathbf{K})\mathbf{x}(k)$, the term $\mathbf{z}(k)^T \mathbf{z}(k)$ and $\Delta V(\mathbf{x}(k))$ can be expressed as follows:

$$\mathbf{z}^T \mathbf{z} = \mathbf{C}_z^T \mathbf{C}_z + \mathbf{C}_z^T \mathbf{D}_z \mathbf{K} + \mathbf{K}^T \mathbf{D}_z^T \mathbf{C}_z + \mathbf{K}^T \mathbf{D}_z^T \mathbf{D}_z \mathbf{K}, \quad (3.43)$$

$$\Delta V(\mathbf{x}(k)) = (\mathbf{A}(\alpha) + \mathbf{BK})^T \mathbf{L} (\mathbf{A}(\alpha) + \mathbf{BK}) - \mathbf{L}. \quad (3.44)$$

Now, replacing the equation (3.43) and (3.44) into $\Delta V(\mathbf{x}(k)) + \mathbf{z}^T \mathbf{z} < 0$, the discrete Riccati equation for the closed loop system is obtained, assuming that matrices \mathbf{C}_z and \mathbf{D}_z satisfy the condition of $\mathbf{C}_z > 0$, $\mathbf{D}_z > 0$ and $\mathbf{C}_z^T \mathbf{D}_z = 0$,

$$(\mathbf{A}(\alpha) + \mathbf{BK})^T \mathbf{L} (\mathbf{A}(\alpha) + \mathbf{BK}) - \mathbf{L} + \mathbf{C}_z^T \mathbf{C}_z + \mathbf{K}^T \mathbf{D}_z^T \mathbf{D}_z \mathbf{K} < 0. \quad (3.45)$$

According to Trofino, Coutinho & Barbosa (2003), the cost function (3.41) can be minimized defining an upper bound as follows:

$$\sum_{k=0}^{\infty} \mathbf{z}(k)^T \mathbf{z}(k) < \mathbf{x}(0)^T \mathbf{L} \mathbf{x}(0), \quad (3.46)$$

where minimizing the term $\mathbf{x}(0)^T \mathbf{L} \mathbf{x}(0)$ the cost function $V(\mathbf{z})$ is minimized.

Additionally, defining $\kappa - \mathbf{x}_o^T \mathbf{M}^{-1} \mathbf{x}_o < 0$, being κ the objective to be minimized and $\mathbf{M} = \mathbf{L}^{-1}$, the complete conditions for the LMI formulation are obtained. Thus, rewritten these conditions with the Schur's complement, the LMI formulation for a discrete system is given by:

$$\min \kappa \tag{3.47}$$

subjected to:

$$\begin{bmatrix} \kappa & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} > 0$$

$$\begin{bmatrix} -\mathbf{M} & \mathbf{M} \cdot \mathbf{C}_z^T + \mathbf{Y}^T \cdot \mathbf{D}_z^T & \mathbf{M} \cdot \mathbf{A}_i^T + \mathbf{Y}^T \cdot \mathbf{B}^T \\ \mathbf{C}_z \cdot \mathbf{M} + \mathbf{D}_z \cdot \mathbf{Y} & -\mathbf{I} & \mathbf{0} \\ \mathbf{A}_i \cdot \mathbf{M} + \mathbf{B} \cdot \mathbf{Y} & \mathbf{0} & -\mathbf{M} \end{bmatrix} < 0$$

$$\mathbf{M} > 0,$$

where the matrix $\mathbf{M} = \mathbf{M}^T$, and $\mathbf{K} = \mathbf{Y} \mathbf{M}^{-1}$.

Finally, solving these LMIs with a semidefinite programming, the terminal value is obtained, which is a Lyapunov matrix, which ensures the stability of the system from the N -th step.

3.4 SUMMARY

In this chapter, two MPC approaches for linear time varying systems were presented. The main difference between them is the prediction model. The I-SSMPC uses the last control action in an augmented state vector to compute the prediction model. On the other hand, the NI-SSMPC uses the state space system without modification. Additionally, a formulation to calculate a terminal value for a linear time varying system was presented in order to implement a terminal cost in the MPC cost function. The aim of this terminal cost is to guarantee stability for the control system and to reduce the prediction horizon.

In the next chapter, these formulations are applied to the Tilt-rotor UAV with suspended load to solve the path tracking problem.

4 MPC BASED ON THE ERROR MODEL OF A TILT-ROTOR UAV WITH SUSPENDED LOAD

The aim of this chapter is to solve the path tracking problem of the Tilt-rotor UAV in load transportation tasks. In the previous chapter the theory of a model predictive controller for time-varying linear systems was introduced. This theory is applied to the error linear model of the Tilt-rotor UAV with suspended load, presented in Chapter 2, to synthesize two types of controllers, each one with a different prediction model. The predictive controllers use the linear error model (2.63). Consequently, the model predictive controllers based on state space systems (SSMPC) presented in Chapter 3 are extended to an error based predictive control approach (E-SSMPC).

In section 4.1, the incremental predictive controller based on the linear error model is proposed. Section 4.2 presents the non-incremental predictive controller based on the linear error model of the aircraft. Finally, in section 4.3, both controllers are tuned. Simulation results are presented in the next chapter in order to validate the controllers.

4.1 I-SSMPC BASED ON THE ERROR MODEL OF A TILT-ROTOR

In this section, the Tilt-rotor UAV with suspended load model is used in the incremental model predictive formulation, presented in Section 3.1, to solve the path tracking problem. Consider the linear error model equation (2.63). This model is based on the error control signal between the current and reference control inputs. Therefore, the incremental error action is defined as $\Delta \mathbf{u}(k) = \tilde{\mathbf{u}}(k) - \tilde{\mathbf{u}}(k-1)$, where $\tilde{\mathbf{u}}(k-1)$ is the last error control signal. Thus, as mentioned in Section 3.1.1, to improve the incremental prediction model, the state space is augmented with the last control error action, defining the new incremental error state vector as $\tilde{\mathbf{x}}_{\alpha}(k) = [\tilde{\mathbf{x}}_s(k) \quad \tilde{\mathbf{u}}(k-1)]$. Accordingly, the augmented system takes the following form:

$$\begin{bmatrix} \tilde{\mathbf{x}}_s(k+1) \\ \tilde{\mathbf{u}}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_z(k) & \mathbf{B}_z \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_s(k) \\ \tilde{\mathbf{u}}(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_z \\ \mathbf{I} \end{bmatrix} \Delta \mathbf{u}(k). \quad (4.1)$$

For better visualization, the equation can be rewritten in a compact form, as:

$$\tilde{\mathbf{x}}_a(k+1) = \bar{\mathbf{A}}_z(k) \tilde{\mathbf{x}}_a(k) + \bar{\mathbf{B}}_z \Delta \mathbf{u}(k). \quad (4.2)$$

Then, expanding the LTV system (4.2) throughout the prediction horizon N and the control horizon M , the predicted output model is computed:

$$\hat{\mathbf{x}} = \mathbf{P}(k|k) \hat{\mathbf{u}} + \mathbf{Q}(k|k) \tilde{\mathbf{x}}_a(k), \quad (4.3)$$

where matrix $\mathbf{P}(k|k)$ is quite different from the one presented in (3.5), because in this case only matrix $\mathbf{A}_z(k+j)$ is time varying, as described below:

$$\mathbf{P}(k|k) = \begin{bmatrix} \bar{\mathbf{B}}_z & 0 & 0 & 0 \\ \hat{\mathbf{\Lambda}}(k, 2, 1) \bar{\mathbf{B}}_z & \bar{\mathbf{B}}_z & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{\Lambda}}(k, M, 1) \bar{\mathbf{B}}_z & \hat{\mathbf{\Lambda}}(k, M, 2) \bar{\mathbf{B}}_z & \dots & \bar{\mathbf{B}}_z \\ \hat{\mathbf{\Lambda}}(k, M+1, 1) \bar{\mathbf{B}}_z & \hat{\mathbf{\Lambda}}(k, M+1, 2) \bar{\mathbf{B}}_z & \dots & \hat{\mathbf{\Lambda}}(k, M+1, M) \bar{\mathbf{B}}_z + \boldsymbol{\varrho}(k, M+1, M) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{\mathbf{\Lambda}}(k, N, 1) \bar{\mathbf{B}}_z & \hat{\mathbf{\Lambda}}(k, N, 2) \bar{\mathbf{B}}_z & \dots & \hat{\mathbf{\Lambda}}(k, N, M) \bar{\mathbf{B}}_z + \boldsymbol{\varrho}(k, N, M) \end{bmatrix} \quad (4.4)$$

$$\mathbf{Q}(k|k) = \begin{bmatrix} \hat{\mathbf{\Lambda}}(k, 0, 1) \\ \hat{\mathbf{\Lambda}}(k, 0, 2) \\ \vdots \\ \hat{\mathbf{\Lambda}}(k, 0, N-1) \\ \hat{\mathbf{\Lambda}}(k, 0, N) \end{bmatrix}. \quad (4.5)$$

where:

$$\boldsymbol{\varrho}(k, i, j) = \left(\sum_{l=j+1}^{i-1} \hat{\mathbf{\Lambda}}(k, i, l) \right) \bar{\mathbf{B}}_z + \bar{\mathbf{B}}_z, \quad (4.6)$$

and the term $\hat{\mathbf{\Lambda}}$ defined in (3.7), is the product of the variations of the matrix $\bar{\mathbf{A}}_z(k+j)$, according to the accelerations of the predefined reference trajectory. However, to simplify the problem of calculating these matrices, it is possible to assume that the prediction model takes into account only the system for instant k , defining the future matrix $\bar{\mathbf{A}}_z(k+j)$ as $\bar{\mathbf{A}}_z(k)$. When the matrix $\bar{\mathbf{A}}_z(k)$ is assumed constant throughout the prediction horizon, a model uncertainty is added to the system; however, this is possible consider that the terminal cost function is calculated in order to guarantee the stability of the entire family of models within the trajectory accelerations ranges.

In this context the matrices \mathbf{P} and \mathbf{Q} are redefined as follows:

$$\mathbf{P}(k|k) = \begin{bmatrix} \bar{\mathbf{B}}_z & 0 & \cdots & 0 \\ \bar{\mathbf{A}}_z(k|k)\bar{\mathbf{B}}_z & \bar{\mathbf{B}}_z & \cdots & 0 \\ \bar{\mathbf{A}}_z(k|k)^2\bar{\mathbf{B}}_z & \bar{\mathbf{A}}_z(k|k)\bar{\mathbf{B}}_z & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{A}}_z(k|k)^M\bar{\mathbf{B}}_z & \bar{\mathbf{A}}_z(k|k)^{M-1}\bar{\mathbf{B}}_z & \cdots & \bar{\mathbf{B}}_z + \varepsilon(\mathbf{k}, \mathbf{i}) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\mathbf{A}}_z(k|k)^{N-1}\bar{\mathbf{B}}_z & \bar{\mathbf{A}}_z^{N-2}(k|k)\bar{\mathbf{B}}_z & \cdots & \bar{\mathbf{A}}_z^{N-M}(k|k)\bar{\mathbf{B}}_z + \varepsilon(\mathbf{k}, \mathbf{i}) \end{bmatrix}, \quad (4.7)$$

$$\mathbf{Q}(k|k) = \begin{bmatrix} \bar{\mathbf{A}}_z(k|k) \\ \bar{\mathbf{A}}_z(k|k)^2 \\ \vdots \\ \bar{\mathbf{A}}_z(k|k)^N \end{bmatrix}, \quad (4.8)$$

where:

$$\varepsilon(\mathbf{k}, \mathbf{i}) \begin{cases} 0 & \text{if } M = N \\ \sum_{i=1}^{N-M} \bar{\mathbf{A}}_z^{N-M-i}\bar{\mathbf{B}}_z & \text{if } M < N \end{cases}. \quad (4.9)$$

Nevertheless, the state prediction vector $\hat{\mathbf{x}}$ in equation (3.8) is redefined as the error prediction vector, considering the future variation of the states, and $\hat{\mathbf{u}}$ is redefined as the increment vector of the future error of the control, as expressed by:

$$\hat{\mathbf{x}} = \begin{bmatrix} \tilde{\tilde{\mathbf{x}}}_a(k+1|k) \\ \tilde{\tilde{\mathbf{x}}}_a(k+2|k) \\ \tilde{\tilde{\mathbf{x}}}_a(k+3|k) \\ \vdots \\ \tilde{\tilde{\mathbf{x}}}_a(k+N|k) \end{bmatrix}, \quad \hat{\mathbf{u}} = \begin{bmatrix} \Delta \mathbf{u}(k|k) \\ \Delta \mathbf{u}(k+1|k) \\ \Delta \mathbf{u}(k+2|k) \\ \vdots \\ \Delta \mathbf{u}(k+M-1|k) \end{bmatrix}. \quad (4.10)$$

With the goal of having the entire formulation based on the error model, the objective function compounded by the stage cost presented in (3.11) and the terminal cost in (3.36), must be rewritten taking into account the error model as follows:

$$\begin{aligned}
V(\bar{\mathbf{x}}_a(k), \Delta \mathbf{u}(k)) &= \sum_{i=1}^{N-1} [\delta \tilde{\mathbf{x}}_a(k+i|k)]' \Sigma_\rho [\delta \tilde{\mathbf{x}}_a(k+i|k)] \\
&+ \sum_{j=0}^{M-1} [\Delta \mathbf{u}(k+j|k)]' \Sigma_\lambda [\Delta \mathbf{u}(k+j|k)] + [\delta \tilde{\mathbf{x}}_a(k+N|k)]' \mathbf{L} [\delta \tilde{\mathbf{x}}_a(k+N|k)],
\end{aligned} \tag{4.11}$$

where $\delta \tilde{\mathbf{x}}_a(k+j|k) = \tilde{\mathbf{x}}_a(k+j|k) - \tilde{\mathbf{x}}_r(k+j|k)$. Besides, this objective function can be represented in the matrix form, as presented in (3.11). However, to include the terminal cost in this function, the terminal value \mathbf{L} must be included in the last term of the main diagonal of the weighting state matrix \mathbf{W}_y , that represents the prediction horizon $k+N$, as:

$$\mathbf{W}_y = \begin{bmatrix} \Sigma_\rho & 0 & \dots & 0 & 0 \\ 0 & \Sigma_\rho & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \Sigma_\rho & 0 \\ 0 & 0 & \dots & 0 & \mathbf{L} \end{bmatrix}. \tag{4.12}$$

Additionally, the state weighting matrix Σ_ρ has the form expressed in (3.12), and \mathbf{W}_u is a diagonal matrix with M blocks of Σ_λ , which is represented in equation (3.13). In the objective function (4.11), the predefined reference trajectory vector (3.14) takes the form of the future reference error vector as:

$$\hat{\tilde{\mathbf{x}}}_r = \begin{bmatrix} \bar{\mathbf{x}}_r(k+1|k) - \bar{\mathbf{x}}_r(k|k) \\ \vdots \\ \bar{\mathbf{x}}_r(k+N|k) - \bar{\mathbf{x}}_r(k+N-1|k) \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{x}}_r(k+1|k) \\ \vdots \\ \tilde{\mathbf{x}}_r(k+N|k) \end{bmatrix} \tag{4.13}$$

In order to calculate the control law when constraints are not considered, equation in (??) is reformulated yielding to:

$$\hat{\mathbf{u}} = (\mathbf{P}' \mathbf{W}_y \mathbf{P} + \mathbf{W}_u)^{-1} \mathbf{P}' \mathbf{W}_y (\hat{\tilde{\mathbf{x}}}_r - \mathbf{Q} \tilde{\mathbf{x}}_a(k)) \tag{4.14}$$

Nevertheless, as the predictive controller is based on the error model, the result of the last equation is the increment of the control error action instead of the increment control action that was presented in section 3.1.3. Therefore, considering that the controller has an incremental

prediction model, the error control action at instant k considers the incremental error action calculated at this instant and the last control error action as:

$$\tilde{\mathbf{u}}(k) = \tilde{\mathbf{u}}(k-1) + \Delta\hat{\mathbf{u}}(k|k). \quad (4.15)$$

Thus, the control action applied to the aircraft is obtained adding the control reference to the error signal as follows:

$$\mathbf{u}(k) = \tilde{\mathbf{u}}(k) + \mathbf{u}_r(k). \quad (4.16)$$

Remembering that the aim of this work is to obtain the control action subject to constraints, the terms in the equation (3.16), must be rewritten in the following form:

$$\begin{aligned} \mathcal{G} &= 2 [\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u], \\ \mathbf{f}' &= 2(\mathbf{Q} \tilde{\mathbf{x}}_a(k) - \hat{\tilde{\mathbf{x}}}_r)^T \mathbf{W}_y \mathbf{P}, \\ \mathbf{c} &= (\mathbf{Q} \tilde{\mathbf{x}}_a(k) - \hat{\tilde{\mathbf{x}}}_r)^T \mathbf{W}_y (\mathbf{Q} \tilde{\mathbf{x}}_a(k) - \hat{\tilde{\mathbf{x}}}_r), \\ \mathbf{A}\mathbf{r}, \mathbf{b}\mathbf{r} &= \text{Constraints matrices.} \end{aligned} \quad (4.17)$$

With the objective of reformulating the input constraint presented in section 3.1.3.1, as a function of the error model, the inequality $\tilde{\mathbf{u}}_{min} \leq \tilde{\mathbf{u}}(k+j-1|k) \leq \tilde{\mathbf{u}}_{max}$ is used to express the limits of the error control signal. However, this error must take into account the maximum and minimum admissible values of the actuators. Therefore, it is possible to extend the inequality with the control error definition as follows:

$$\begin{aligned} \tilde{\mathbf{u}}_{max}(k+j-1|k) &= \mathbf{u}_{max} - \mathbf{u}_r(k+j-1|k), \\ \tilde{\mathbf{u}}_{min}(k+j-1|k) &= \mathbf{u}_{min} - \mathbf{u}_r(k+j-1|k), \end{aligned} \quad (4.18)$$

where \mathbf{u}_{max} and \mathbf{u}_{min} , constant for the entire prediction horizon, are the maximum and minimum admissible values that can be applied by the actuators. Thus, the constraint equation can be written as:

$$\begin{bmatrix} \mathbf{T}\mathbf{m} \\ -\mathbf{T}\mathbf{m} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\tilde{\mathbf{u}}}_{max} - \tilde{\mathbf{u}}(k-1) \\ \tilde{\mathbf{u}}(k-1) - \hat{\tilde{\mathbf{u}}}_{min} \end{bmatrix}, \quad (4.19)$$

where $\hat{\tilde{\mathbf{u}}}_{max}$ and $\hat{\tilde{\mathbf{u}}}_{min}$ are the maximum and minimum control error vector. They are obtained expanding equations (4.18) throughout the prediction control. Considering the state constraints, the inequality (3.21) is rewritten as $\tilde{\mathbf{x}}_{amin}(k+i|k) \leq \tilde{\mathbf{x}}_a(k+i|k) \leq \tilde{\mathbf{x}}_{amax}(k+i|k)$,

where the maximum and minimum error states values are defined as:

$$\begin{aligned}\tilde{\tilde{\mathbf{x}}}_{\mathbf{a}min}(k+i|k) &= \tilde{\mathbf{x}}_{\mathbf{a}min} - \tilde{\mathbf{x}}_{\mathbf{r}}(k+i|k), \\ \tilde{\tilde{\mathbf{x}}}_{\mathbf{a}max}(k+i|k) &= \tilde{\mathbf{x}}_{\mathbf{a}max} - \tilde{\mathbf{x}}_{\mathbf{r}}(k+i|k),\end{aligned}\quad (4.20)$$

where $\tilde{\mathbf{x}}_{\mathbf{a}max}$ and $\tilde{\mathbf{x}}_{\mathbf{a}min}$ are the maximum and minimum values that the states can take assuming that the aircraft is in a confined environment. Thus, the matrix form of the constraint given by:

$$\begin{bmatrix} \mathbf{P} \\ -\mathbf{P} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}max} - \mathbf{Q}\tilde{\tilde{\mathbf{x}}}_{\mathbf{a}}(k|k) \\ \mathbf{Q}\tilde{\tilde{\mathbf{x}}}_{\mathbf{a}}(k|k) - \hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}min} \end{bmatrix}, \quad (4.21)$$

where $\hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}max}$ and $\hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}min}$ are the maximum and minimum error state vector obtained expanding equation (4.20) throughout the prediction horizon. In this context, the constraint matrices $\mathbf{A}_{\mathbf{r}}$ and $\mathbf{b}_{\mathbf{r}}$ are defined using (4.19) and (4.21) as:

$$\mathbf{A}_{\mathbf{r}} = \begin{bmatrix} \mathbf{T}_m \\ -\mathbf{T}_m \\ \mathbf{P} \\ -\mathbf{P} \end{bmatrix}, \quad \mathbf{b}_{\mathbf{r}} = \begin{bmatrix} \hat{\tilde{\tilde{\mathbf{u}}}}_{max} - \tilde{\tilde{\mathbf{u}}}(k-1) \\ \tilde{\tilde{\mathbf{u}}}(k-1) - \hat{\tilde{\tilde{\mathbf{u}}}}_{min} \\ \hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}max} - \mathbf{Q}\tilde{\tilde{\mathbf{x}}}_{\mathbf{a}}(k) \\ \mathbf{Q}\tilde{\tilde{\mathbf{x}}}_{\mathbf{a}}(k) - \hat{\tilde{\tilde{\mathbf{x}}}}_{\mathbf{a}min} \end{bmatrix}. \quad (4.22)$$

4.2 NI-SSMPC BASED ON THE ERROR MODEL OF A TILT-ROTOR

In this section, the non-incremental predictive control formulation presented in (3.2) is applied to the linear error model of the Tilt-rotor UAV, obtained in Section 2.7, to predict the system's behavior. Rewriting the discrete linear error model presented in the equation (2.65) in the following form:

$$\tilde{\tilde{\mathbf{x}}}_{\mathbf{s}}(k+1+i|k) = \mathbf{A}_{\mathbf{z}}(k+i|k)\tilde{\tilde{\mathbf{x}}}_{\mathbf{s}}(k+i|k) + \mathbf{B}_{\mathbf{z}}\tilde{\tilde{\mathbf{u}}}(k+i|k), \quad (4.23)$$

it is possible to find the prediction model extending this equation to the future, where $i = 1, 2, \dots, N$. By doing this, the prediction error model is written as follows:

$$\hat{\tilde{\tilde{\mathbf{x}}}} = \mathbf{P}(k|k)\tilde{\tilde{\mathbf{u}}} + \mathbf{Q}(k|k)\tilde{\tilde{\mathbf{x}}}_{\mathbf{s}}(k). \quad (4.24)$$

As mentioned before, in the linear error model (2.7), only the

matrix $\mathbf{A}_z(k)$ is time varying. For this reason, the matrices \mathbf{P} and \mathbf{Q} are modified as:

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{B}_z & 0 & 0 & 0 \\ \mathbf{\Lambda}(k, 2, 1)\mathbf{B}_z & \mathbf{B}_z & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Lambda}(k, M, 1)\mathbf{B}_z & \mathbf{\Lambda}(k, M, 2)\mathbf{B}_z & \dots & \mathbf{B}_z \\ \mathbf{\Lambda}(k, M+1, 1)\mathbf{B}_z & \mathbf{\Lambda}(k, M+1, 2)\mathbf{B}_z & \dots & \mathbf{\Lambda}(k, M+1, M)\mathbf{B}_z + \boldsymbol{\varrho}(k, M+1, M) \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{\Lambda}(k, N, 1)\mathbf{B}_z & \mathbf{\Lambda}(k, N, 2)\mathbf{B}_z & \dots & \mathbf{\Lambda}(k, N, M)\mathbf{B}_z + \boldsymbol{\varrho}(k, N, M) \end{bmatrix}, \quad (4.25)$$

$$\mathbf{Q}(k|k) = \begin{bmatrix} \mathbf{\Lambda}(k, 0, 1) \\ \mathbf{\Lambda}(k, 0, 2) \\ \vdots \\ \mathbf{\Lambda}(k, 0, N-1) \\ \mathbf{\Lambda}(k, 0, N) \end{bmatrix}, \quad (4.26)$$

where:

$$\boldsymbol{\varrho}(k, i, j) = \left(\sum_{l=j+1}^{i-1} \hat{\mathbf{\Lambda}}(k, i, l) \right) \mathbf{B}_z + \mathbf{B}_z, \quad (4.27)$$

and the product variation $\mathbf{\Lambda}$ of matrix $\mathbf{A}_z(k+j)$ is given by expression (3.7). Nevertheless, as mentioned in last section, it is possible use $\mathbf{A}(k+j|k) = \mathbf{A}(k|k)$, in order to reduce the computational cost. Thereby, matrices \mathbf{P} and \mathbf{Q} are simplified and expressed by:

$$\mathbf{P}(k|k) = \begin{bmatrix} \mathbf{B}_z & 0 & 0 & \dots & 0 \\ \mathbf{A}_z(k|k)\mathbf{B}_z & \mathbf{B}_z & 0 & \dots & 0 \\ \mathbf{A}_z(k|k)^2\mathbf{B}_z & \mathbf{A}_z(k|k)\mathbf{B}_z & \mathbf{B}_z & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_z(k|k)^M\mathbf{B}_z & \mathbf{A}_z(k|k)^{M-1}\mathbf{B}_z & \mathbf{A}_z(k|k)^{M-2}\mathbf{B}_z & \dots & \mathbf{B}_z + \boldsymbol{\varepsilon}(k, j) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{A}_z(k|k)^{N-1}\mathbf{B}_z & \mathbf{A}_z(k|k)^{N-2}\mathbf{B}_z & \mathbf{A}_z(k|k)^{N-3}\mathbf{B}_z & \dots & \mathbf{A}_z(k|k)^{N-M}\mathbf{B}_z + \boldsymbol{\varepsilon}(k, j) \end{bmatrix} \quad (4.28)$$

$$\mathbf{Q}(k|k) = \begin{bmatrix} \mathbf{A}_z(k|k) \\ \mathbf{A}_z(k|k)^2 \\ \vdots \\ \mathbf{A}_z(k|k)^N \end{bmatrix}, \quad (4.29)$$

where:

$$\varepsilon(k, j) \begin{cases} 0 & \text{if } M = N \\ \sum_{i=1}^{N-M} \mathbf{A}_z^{N-M-i} \mathbf{B}_z & \text{if } M < N. \end{cases} \quad (4.30)$$

The error state prediction vector $\hat{\mathbf{x}}$ (resulting from Equation (4.24)), and the incremental control error vector $\hat{\mathbf{u}}$ are expressed, respectively, as:

$$\hat{\mathbf{x}} = \begin{bmatrix} \tilde{\tilde{\mathbf{x}}}_s(k+1|k) \\ \tilde{\tilde{\mathbf{x}}}_s(k+2|k) \\ \tilde{\tilde{\mathbf{x}}}_s(k+3|k) \\ \vdots \\ \tilde{\tilde{\mathbf{x}}}_s(k+N|k) \end{bmatrix}, \quad \hat{\mathbf{u}} = \begin{bmatrix} \tilde{\tilde{\mathbf{u}}}(k|k) \\ \tilde{\tilde{\mathbf{u}}}(k+1|k) \\ \tilde{\tilde{\mathbf{u}}}(k+2|k) \\ \vdots \\ \tilde{\tilde{\mathbf{u}}}(k+M-1|k) \end{bmatrix}. \quad (4.31)$$

In order to calculate the future error action, the objective function, formed by the stage cost presented in (3.29) and the terminal cost in (3.36) applied to the linear error model, is expressed by:

$$\begin{aligned} V(\mathbf{x}(k), \mathbf{u}(k)) &= \sum_{j=1}^{N-1} [\delta \tilde{\tilde{\mathbf{x}}}_s(k+j)]^T \boldsymbol{\Sigma}_\rho [\delta \tilde{\tilde{\mathbf{x}}}_s(k+j)] \\ &+ \sum_{j=0}^{M-1} [\tilde{\tilde{\mathbf{u}}}(k+j)]^T \boldsymbol{\Sigma}_\lambda [\tilde{\tilde{\mathbf{u}}}(k+j)] + [\delta \tilde{\tilde{\mathbf{x}}}_s(k+N)]^T \mathbf{L} [\delta \tilde{\tilde{\mathbf{x}}}_s(k+N)], \end{aligned} \quad (4.32)$$

where $\delta \tilde{\tilde{\mathbf{x}}}_s(k+i|k) = \tilde{\tilde{\mathbf{x}}}_s(k+i|k) - \tilde{\tilde{\mathbf{x}}}_r(k+i|k)$. In order to express this function in a matrix form, as presented in equation (3.29), the terminal value is added to the last term of the main diagonal in the state weighting matrix \mathbf{W}_y , as presented in (4.12). The predefined reference error vector $\tilde{\tilde{\mathbf{x}}}_r$ is expressed in (4.13), and the error control reference vector $\hat{\mathbf{u}}_r$ takes the form:

$$\hat{\mathbf{u}}_r = \begin{bmatrix} \mathbf{u}_r(k|k) - \mathbf{u}_r(k-1|k) \\ \vdots \\ \mathbf{u}_r(k+M-1|k) - \mathbf{u}_r(k+M-2|k) \end{bmatrix} = \begin{bmatrix} \tilde{\tilde{\mathbf{u}}}_r(k|k) \\ \vdots \\ \tilde{\tilde{\mathbf{u}}}_r(k+M-1|k) \end{bmatrix}. \quad (4.33)$$

In the absence of constraints, the control variation could be computed as:

$$\hat{\mathbf{u}} = (\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u)^{-1} [\mathbf{P}^T \mathbf{W}_y (\hat{\tilde{\mathbf{x}}}_r - \mathbf{Q} \tilde{\mathbf{x}}_s(k)) + \mathbf{W}_u \hat{\mathbf{u}}_r]. \quad (4.34)$$

As mentioned in Section 3.2.3.2, the error control action at the instant k is composed by the first p terms of the vector $\hat{\mathbf{u}}$. Otherwise, as the predictive model is based on the error model, the control action applied to the system is calculated by:

$$\mathbf{u}(k) = \tilde{\mathbf{u}}(k|k) + \mathbf{u}_r(k). \quad (4.35)$$

In order to calculate the control considering the constrained problem, the QP formulation presented in 3.2.3.1 is reformulated as a function of the error system, where the terms \mathcal{G} , f' and c are given by:

$$\begin{aligned} \mathcal{G} &= 2[\mathbf{P}^T \mathbf{W}_y \mathbf{P} + \mathbf{W}_u], \\ f' &= 2(\mathbf{Q} \tilde{\mathbf{x}}_s(k) - \hat{\tilde{\mathbf{x}}}_r)^T \mathbf{W}_y \mathbf{P} - \mathbf{W}_u \hat{\mathbf{u}}_r, \\ c &= (\mathbf{Q} \tilde{\mathbf{x}}_s(k) - \hat{\tilde{\mathbf{x}}}_r)^T \mathbf{W}_y (\mathbf{Q} \tilde{\mathbf{x}}_s(k) - \hat{\tilde{\mathbf{x}}}_r) + \hat{\mathbf{u}}_r^T \mathbf{W}_u \hat{\mathbf{u}}_r, \\ \mathbf{A}_r, \mathbf{b}_r &= \text{Constraints matrices.} \end{aligned} \quad (4.36)$$

As mentioned before, the constraint matrices consider the physical limitation of the actuator as input constraints and assume that the aircraft is operating in a confined environment as output constraints.

Therefore, the input constraints, formulated in section 3.2.3.1, must be applied to the error model of the Tilt-rotor UAV. Considering that the error control signal is the difference between the control action and the reference control action, the input constraints take the form:

$$\tilde{\mathbf{u}}_{min}(k+i|k) \leq \tilde{\mathbf{u}}(k+i|k) \leq \tilde{\mathbf{u}}_{max}(k+i|k), \quad (4.37)$$

where the maximum and minimum values of the error signal are calculated by equation (4.18).

With the objective to apply these constraints to the QP formulation, equation (4.37) is expanded throughout the control horizon and expressed in a matrix form as:

$$\begin{bmatrix} \mathbf{I} \mathbf{m} \\ -\mathbf{I} \mathbf{m} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\tilde{\mathbf{u}}}_{max} \\ -\hat{\tilde{\mathbf{u}}}_{min} \end{bmatrix}. \quad (4.38)$$

On the other hand, the state constraints, presented in section 3.2.3.1, considering that the error states are the difference between the states and the predefined reference trajectory, can be rewritten based on the error model as follows:

$$\tilde{\mathbf{x}}_{smin}(k+i|k) \leq \tilde{\mathbf{x}}_s(k+i|k) \leq \tilde{\mathbf{x}}_{smax}(k+i|k), \quad (4.39)$$

where the maximum and minimum error states values are defined as the difference between the admissible values of the estates with the predefined trajectory of the aircraft:

$$\begin{aligned} \tilde{\mathbf{x}}_{smin}(k+i|k) &= \bar{\mathbf{x}}_{smin} - \bar{\mathbf{x}}_r(k+i) \\ \tilde{\mathbf{x}}_{smax}(k+i|k) &= \bar{\mathbf{x}}_{smax} - \bar{\mathbf{x}}_r(k+i). \end{aligned} \quad (4.40)$$

In order to apply these constraints to the QP formulation, equations (4.40) can be expressed in a matrix form expanding throughout the prediction horizon, $i = 1, 2, \dots, N$, as shown below:

$$\begin{bmatrix} P \\ -P \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\tilde{\mathbf{x}}}_{smax} - Q\tilde{\mathbf{x}}_s \\ Q\tilde{\mathbf{x}}_s - \hat{\tilde{\mathbf{x}}}_{smin} \end{bmatrix}, \quad (4.41)$$

where $\hat{\tilde{\mathbf{x}}}_{smax}$ and $\hat{\tilde{\mathbf{x}}}_{smin}$ are the maximum and minimum error estate vectors. Finally, using inequalities (4.38) and (4.41), it is possible to build the constraint matrices \mathbf{A}_r and \mathbf{b}_r :

$$\mathbf{A}_r = \begin{bmatrix} I_m \\ -I_m \\ P \\ -P \end{bmatrix}, \quad \mathbf{b}_r = \begin{bmatrix} \hat{\mathbf{u}}_{max} \\ -\hat{\mathbf{u}}_{min} \\ \hat{\tilde{\mathbf{x}}}_{smax} - Q\tilde{\mathbf{x}}_s(k) \\ Q\tilde{\mathbf{x}}_s(k) - \hat{\tilde{\mathbf{x}}}_{smin} \end{bmatrix}. \quad (4.42)$$

4.3 TUNING OF MODEL PREDICTIVE CONTROLLERS

In this section, the MPC tuning parameters used in the simulations are discussed. Considering that the LQR controllers are equivalent to infinite horizon, unconstrained Model Predictive Controllers, the input and state weighting matrices of the MPC will be used in the LQR to compare the performance between these controllers in simulations.

4.3.1 State and Input Weight Matrices

The first set of tuning parameters, which are common to both LQR controller and MPC, are the weights on the states Σ_{ρ} and control inputs Σ_{λ} . In this work, both matrices are chosen to be diagonal, reducing the number of tunable entries inside the matrix and making possible to manage the states and the control inputs independently.

According to Rawlings & Mayne (2009), high values of Σ_{ρ} in comparison to Σ_{λ} drive the state to the origin quickly, resulting in a strong control action. On the other hand, by penalizing the control action through high values of Σ_{λ} relative to Σ_{ρ} , reduces the control action and slow down the rate at which the state approaches the origin.

Choosing appropriate values of Σ_{ρ} and Σ_{λ} is not always trivial, and is one of the challenges faced in practice. For this reason, these matrices were tuned heuristically with a starting point given by the Bryson's inverse square method (JOHNSON; GRIMBLE, 1987). The input and state weigh matrices are given by:

$$\Sigma_{\lambda} = \text{diag} \left(\frac{1}{(17000 - \text{freq})^2}, \frac{1}{(17000 - \text{freq})^2}, \frac{1}{2000^2}, \frac{1}{2000^2} \right) \quad (4.43)$$

$$\Sigma_{\rho} = \text{diag} \left(2, 2, 2, \frac{3}{\pi^2}, \frac{4}{\pi^2}, \frac{5}{\pi^2}, \frac{1}{\pi^2}, \frac{1}{\pi^2}, \frac{5}{\pi^2}, \frac{5}{\pi^2}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{9\pi^2}, \frac{1}{9\pi^2}, \frac{1}{9\pi^2}, \frac{1}{9\pi^2}, \frac{1}{100\pi^2}, \frac{1}{100\pi^2}, \frac{1}{9\pi^2}, \frac{1}{9\pi^2} \right). \quad (4.44)$$

For MPC and LQR control with integral action, three times higher penalties were placed on the states to be integrated (translational position x , y , z and the yaw angle ψ). These penalties are added as final terms of equation (4.44) with the following values:

$$\Sigma_{\rho_i} = \left[6, 6, 6, \frac{150}{\pi^2} \right]. \quad (4.45)$$

4.3.2 Terminal Cost

The terminal value is one of the most important terms in this work, because with this term it is possible to reduce the prediction horizon, as mentioned in Section 3.3. To calculate this value, the LMI presented in equation (3.47) is used, where the matrices \mathbf{D}_z and \mathbf{C}_z must be chosen to satisfy the condition of $\mathbf{C}_z^T \mathbf{D}_z = 0$. The weights of both matrices are chosen in order to obtain the same weights presented in equations (4.43) and (4.44) when $\mathbf{D}_z^T \mathbf{D}_z = \Sigma_{0\lambda}$ and $\mathbf{C}_z^T \mathbf{C}_z = \Sigma_\rho$ are considered. These matrices are given by:

$$\mathbf{D}_z = \begin{bmatrix} 0_{14,1} & 0_{14,1} & 0_{14,1} & 0_{14,1} \\ \frac{1}{(17000-freq)} & 0 & 0 & 0 \\ 0 & \frac{1}{(17000-freq)} & 0 & 0 \\ 0 & 0 & \frac{1}{2000} & 0 \\ 0 & 0 & 0 & \frac{1}{2000} \end{bmatrix} \quad (4.46)$$

$$\mathbf{C}_z = \text{diag}(\sqrt{2}, \sqrt{2}, \sqrt{2}, \frac{\sqrt{3}}{\pi}, \frac{\sqrt{4}}{\pi}, \frac{\sqrt{5}}{\pi}, \frac{\sqrt{1}}{\pi}, \frac{\sqrt{1}}{\pi}, \frac{\sqrt{5}}{\pi}, \frac{\sqrt{5}}{\pi}), \quad (4.47)$$

$$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{3\pi}, \frac{1}{3\pi}, \frac{1}{3\pi}, \frac{1}{10\pi}, \frac{1}{10\pi}, \frac{1}{3\pi}, \frac{1}{3\pi} \right).$$

In addition, another parameter to be defined to calculate this LMI is the maximum and minimum trajectory accelerations of the states x , y , z . As the system model is time varying, these values define the polytope for the LMI formulation. The predefined trajectory is presented in the next chapter. The maximum and minimum accelerations for this trajectory are:

$$\begin{aligned} \ddot{\mathbf{x}} &= [-0.5 \ 0.5], \\ \ddot{\mathbf{y}} &= [-0.5 \ 0.5], \\ \ddot{\mathbf{z}} &= [-0.5 \ 0.5]. \end{aligned} \quad (4.48)$$

4.3.3 Prediction and Control Horizon

The prediction horizon N and control horizon M are parameters that have to be tuned in an MPC. As mentioned before, one of the ways

to guarantee stability in MPC is to have a larger prediction horizon, but this approach is not suitable in practical terms because it demands a very high computational cost. Nevertheless, with a sampling time of 12 *ms*, and due to the presence of the terminal cost presented in section 3.3, a prediction horizon of $N = 5$ was chosen. In the other hand, the minimum possible control horizon that could be used was $M = 1$ for I-SSMPC and $M = 2$ for NI-SSMPC, and this value was chosen for the simulations.

4.3.4 Constraints Parametrization

As mentioned before, input constraints are assumed as physical limitations of the actuators, and state constraints are defined assuming UAV is in a confined environment. Thus, the control signals' values are constrained to $\{f_r, f_l\} \in [0, 17] N$ and $\tau_r, \tau_l \in \{-2, 2\} N.m$, which correspond to the actuators' specification.

Furthermore, the system states constraints are settled as presented in Table 3.

Table 3 – Output constraints applied to Tilt-rotor with suspended load.

State	Maximum	Minimum	Unit
x	-4	4	<i>m</i>
y	-4	4	<i>m</i>
z	-0	5.5	<i>m</i>
ϕ	-0.5236	0.5236	<i>rad</i>
θ	-0.5236	0.5236	<i>rad</i>
ψ	-0.5236	0.5236	<i>rad</i>
α_r	-0.5236	0.5236	<i>rad</i>
α_l	-0.5236	0.5236	<i>rad</i>
γ_1	-0.5236	0.5236	<i>rad</i>
γ_2	-0.5236	0.5236	<i>rad</i>

In addition, the interval for the translational (*m/s*) and rotational(*rad/s*) velocities are $[-3, 3]$, and the constraint for the integral action of the error states is $[-0.5, 0.5]$.

4.4 SUMMARY

In this chapter, the MPC controllers based on the linear error model of Tilt-rotor UAV with suspended load was presented. These controllers are designed to improve the path tracking problem of the aircraft while stabilizing the suspended load. Two strategies were synthesized. Additionally, tuning parameters were presented in order to compare the performance of MPC against LQR controller.

The next chapter presents several simulation in different scenarios with the objective to demonstrate the effectiveness of the proposed controllers.

5 SIMULATION RESULTS

In this chapter, simulation results are presented in order to evaluate the performance of the proposed controllers. These simulations are obtained in scenarios designed to test different capabilities of the proposed MPCs. With the aim of better appreciate the differences presented by these controllers, a smooth trajectory was chosen. The model parameters of the Tilt-rotor used in the simulations are given in Table 1, which are taken from a real implementation made by the project ProVant. Besides, the tuning parameters of the controllers were presented in Section 4.3.

As mentioned before, the unconstrained MPC formulation is very similar with an LQR controller. For this reason, some simulations compare the performance between the designed controllers and the LQR controller, in order to prove the benefit of using the presented approach.

Section 5.1 describes the scenarios where the controllers, given in Chapter 4, are tested. Also, the predefined trajectory used for all simulations is presented. Additionally, the disturbances affecting the Tilt-rotor UAV are defined. Furthermore, simulation results considering the first scenario are presented in Sections 5.2 and 5.3 for incremental and non-incremental MPCs, respectively. Simulation results for parametric uncertainties are presented in Section 5.4. In section 5.5 quantitative results derived from the simulations are presented.

5.1 SIMULATION PROTOCOL

In this section, the scenarios used to simulate the controllers proposed throughout this dissertation are presented. The purpose of these scenarios is to ensure the same test conditions for the controllers, in order to compare them.

The simulation are executed with the proposed MPC strategies, in order to demonstrate the capabilities for solving the path tracking problem and stabilizing the suspended load when persistent disturbances affect the whole system. The simulations have the structure presented in Figure 7, where it is considered the non-linear model (2.55), and saturation of the actuators in order to achieve a more realistic simulation of the Tilt-rotor aircraft.

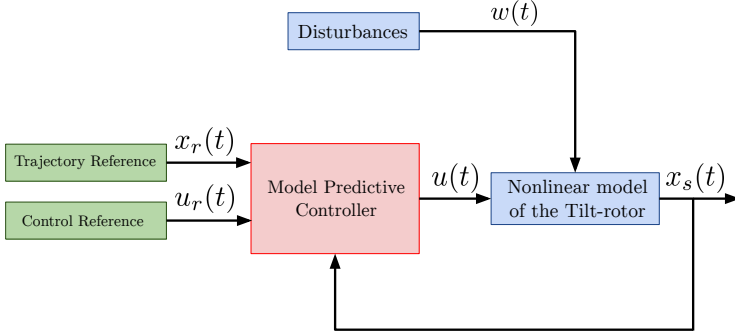


Figure 7 – Simulation structure.

Furthermore, with the aim of testing the robustness of the controllers, an amount of $\pm 30\%$ was applied to the inertial tensors, suspended load and body mass of the Tilt-rotor UAV. Additionally, they were considered constant disturbances affecting all degrees of freedom of the Tilt-rotor UAV with suspended load, where aerodynamic forces and moments are applied in different instants of time with the objective to test the disturbance rejection ability of the proposed controllers. The value of the disturbances used in the simulations are presented in Table 4.

Table 4 – Disturbance parameters

Disturbance	Value	Time	Unit
F_{extx}	1	15	N
F_{exty}	1	25	N
F_{extz}	2	50	N
$F_{ext\phi}$	0.2	30	$N.m$
$F_{ext\theta}$	0.2	35	$N.m$
$F_{ext\psi}$	0.2	40	$N.m$
$F_{ext\alpha_r}$	0.1	45	$N.m$
$F_{ext\alpha_1}$	0.2	60	$N.m$
$F_{ext\gamma_1}$	0.1	65	$N.m$
$F_{ext\gamma_2}$	0.1	75	$N.m$

On the other hand, simulations results are used to compare the proposed controllers with a LQR LTV controller in order to show the improvement obtained with the proposed approach. Finally, the simulation scenarios consider the pre-defined reference trajectory

presented in Figure 8, which is a circular trajectory defined by:

$$x_r(t) = 3 \cos\left(\frac{\pi}{50}t\right) [m], \quad y_r(t) = 3 \sin\left(\frac{\pi}{50}t\right) [m], \quad (5.1)$$

$$z_r(t) = 3 - 2 \cos\left(\frac{\pi}{50}t\right) [m], \quad \psi_r(t) = 0 [rad]. \quad (5.2)$$

Also, the take-off position of the aircraft is considered as: $q(0) = [3 \ 0 \ 1 \ 0_{1 \times 7}] [m]$.

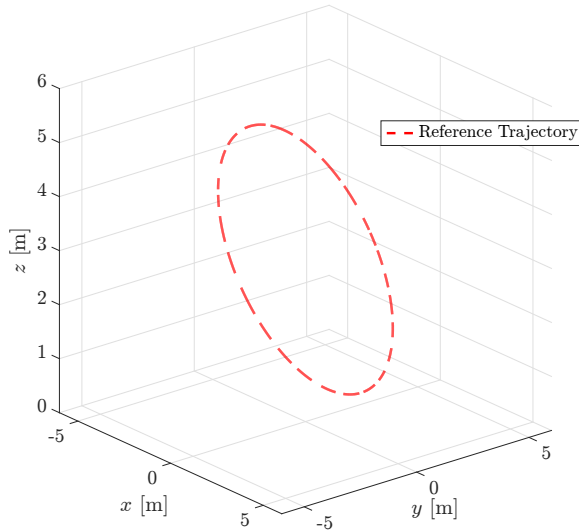


Figure 8 – Reference trajectory of the Tilt-rotor with suspended load

In this context, in sections 5.2 and 5.3 the first scenario is presented for the incremental and non-incremental MPC, respectively. These simulations consider nominal parameters in the inertial tensors and body mass. Additionally, the disturbances presented in Table 4 are applied to the vehicle. In section 5.4 the second and third scenarios are presented taking into account parametric uncertainties of $+30$ and -30 , respectively. These uncertainties are considered in inertial tensors, body mass of the aircraft and the mass of the suspended load. Also, disturbances are considered.

Additionally, in order to have a quantitative comparison of the results obtained in the simulation, some performance indexes are used. The first index is the Mean Square Error (MSE) of the error states. Secondly, the Total Variation index is calculated for all control signals.

5.2 I-SSMPC

In this section, it is presented the first simulation scenario for a I-SSMPC, where the Tilt-rotor UAV with suspended load is tracking a trajectory while affected by constant disturbances.

The controller parameters used in this simulation were presented in Section 4. The trajectory performed by the Tilt-rotor with suspended load while tracking the reference is illustrated in Figure 9; this trajectory is compared to the one one obtaining by using the LQR controller. It is possible to notice that both controllers are able to track the trajectory. Furthermore, the load is stabilized and performs a similar path than the aircraft. Broadly, the I-SSMPC has slightly lower overshoots than the LQR controller in the aircraft trajectory, but, the load path is quite similar.

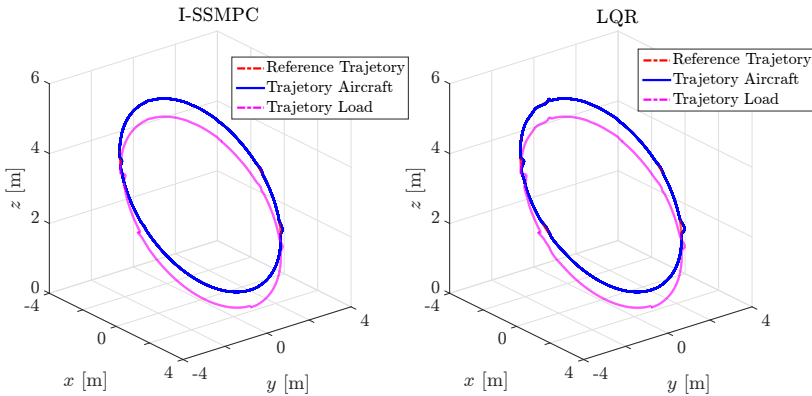
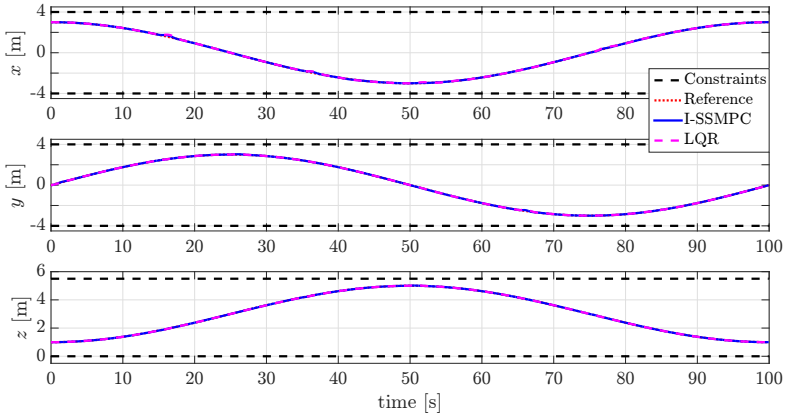


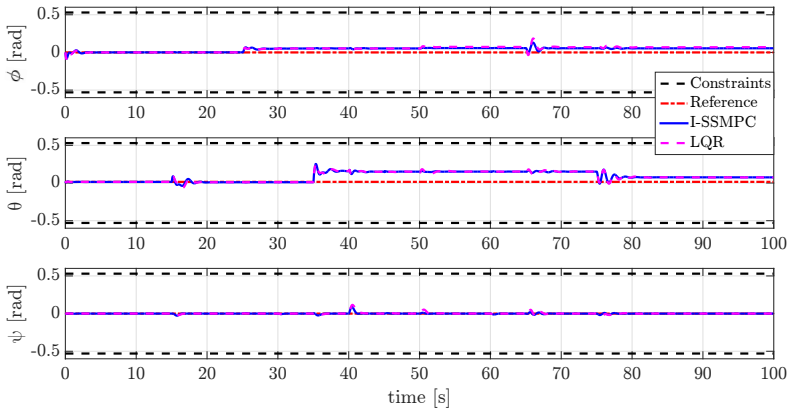
Figure 9 – Reference trajectory of the Tilt-rotor with suspended load made by I-SSMPC by LQR controller.

In Figure 10 translational and rotational states are presented. Specifically, in Figure 10a the time evolution of the translational states shows how the estates reach the reference trajectory given for them. Also, it can be seen that the states x , y and z of I-SSMPC have a slightly better performance than the LQR controller. These states present lower overshoot with the same time response.

In Figure 10b, it can be noticed that the roll and pitch angles reach equilibrium point while there are no disturbances. However, when a disturbance affects the system, these angles change the equilibrium point in order to compensate the states that the controller is tracking.



(a) Translational position (x, y, z) of the Tilt-rotor UAV with respect to the inertial frame.



(b) Rotational position (ϕ, θ, ψ) of the Tilt-rotor UAV around the local axis.

Figure 10 – Time evolution of translational and rotational positions performed by the Tilt-rotor UAV applying the proposed I-SSMPC and LQR controller

In this context, rate change of the LQR controller is greater than I-SSMPC. In contrast, yaw angle is able to track the reference for this state. As presented in Table 4, the disturbances affecting the load angular position appear in time 65 [s] and 75 [s] for γ_1 and γ_2 , respectively. Note that at this time instants, in Figure 10b, the states present more oscillatory response compared to other perturbations, due to the fact that suspended load is the slowest dynamic in the system. Finally, comparing Tilt-rotor attitude given by the I-SSMPC with LQR controller, it presents better performance.

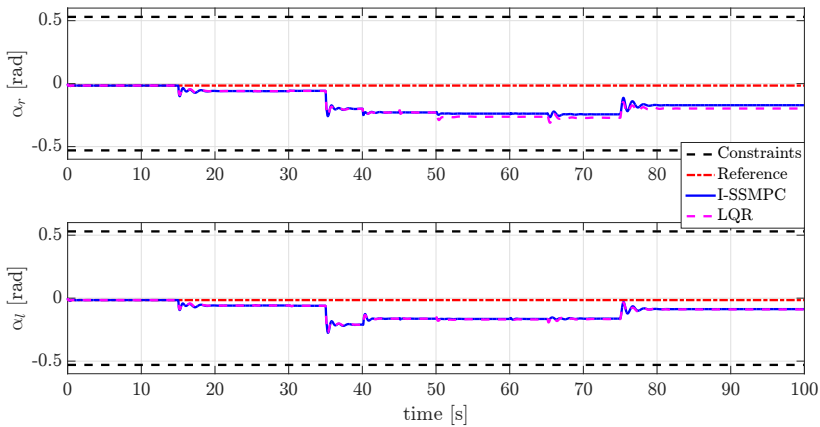


Figure 11 – Time evolution of α_r and α_l performed by Tilt-rotor applying the proposed I-SSMPC and LQR controller.

Figures 11 and 12 show the time responses of the servomotor and suspended load angular position. Servomotors angular position, illustrated in Figure 11, have the same situation as the attitude states, where the equilibrium point of the α_r and α_l angles is achieved in the absence of disturbances. Then, this equilibrium point changes to compensate the other states. On the other hand, the load position response, represented in Figure 12 shows that the LQR controller has better performance in these states than I-SSMPC, since they have a less oscillatory response and the overshoots are lower. It should be noticed that the oscillation amplitude of I-SSMPC is greater than LQR control, because load stabilization needs an aggressive control action, and as I-SSMPC takes into account the physical limitations of actuators, it produces a smoother control action than LQR controller. The time

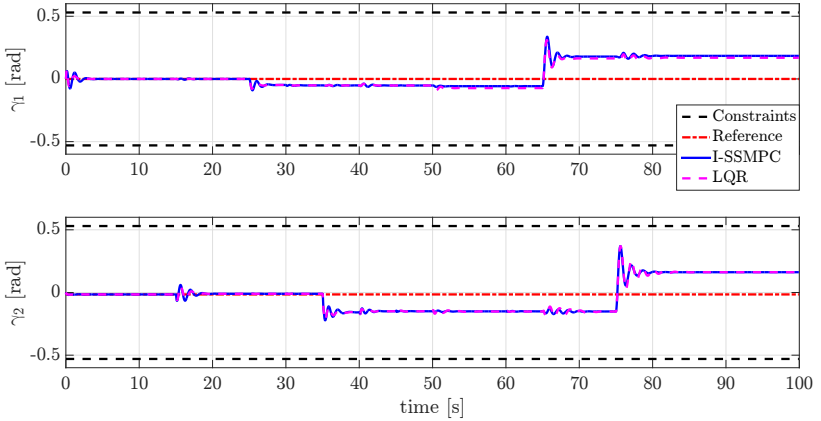


Figure 12 – Time evolution of γ_1 and γ_2 , performed by Tilt-rotor applying the proposed I-SSMPC and LQR controller

response of thrust forces and torques of servomotors are presented in Figure 13. Considering only the trust forces in Figure 13a, the variation of the control action, presented by the I-SSMPC is lower than the one obtained by the LQR controller, which means that the I-SSMPC generates a less aggressive control action. Also, it can be noticed that the left force is slightly lower than the right force. This situation is due to the position of the center of mass which is shifted to the right side. In contrast, the right torque applied to the servomotor is lower than the left torque when a direct disturbance affects it.

I-SSMPC presented better performance compared to the LQR controller since the states have less overshoot and the stationary state changes are lower. However, the suspended load has slightly more swing with I-SSMPC. In the case of the control signal, it was shown that I-SSMPC generates a smoother control action since the increments are smaller when compared with the LQR controller.

5.3 NI-SSMPC

In this section simulation results for the NI-SSMPC are presented, where the tuning parameters used are shown in Section 4.3 and Tilt-rotor UAV parameters are given in Table 1. The main difference of this controller lies in the non-use of integral action in

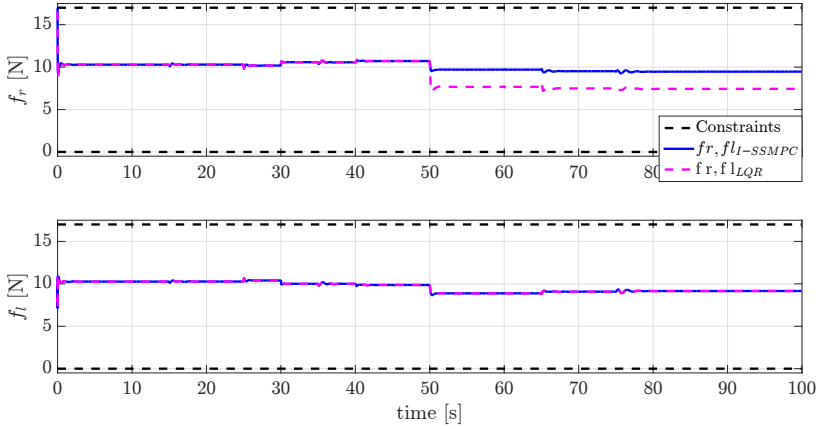
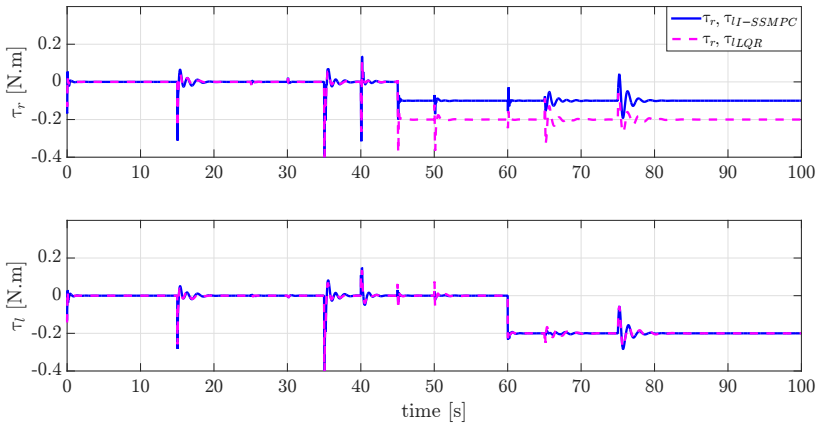
(a) Thrust Forces (f_r, f_l) generated by the Tilt-rotor UAV.(b) Torques (τ_r, τ_l) applied to the servomotors

Figure 13 – Time evolution of thrust forces and torques applied to Tilt-rotor UAV by the proposed I-SSMPC and LQR controller.

the control signal, resulting in a more aggressive control action. For this reason, the last control horizon, $M = 1$, was modified to $M = 2$, because with the previous value the controller generates a very aggressive control action, which destabilizes the system. The simulation results are depicted in Figures 14 to 17.

In Figure 14 the path tracking performed by the Tilt-rotor UAV is presented for both controllers, NI-SSMPC and LQR. It shows very similar behavior than the last approach, where the path made by LQR controller has higher overshoots compared to the proposed one.

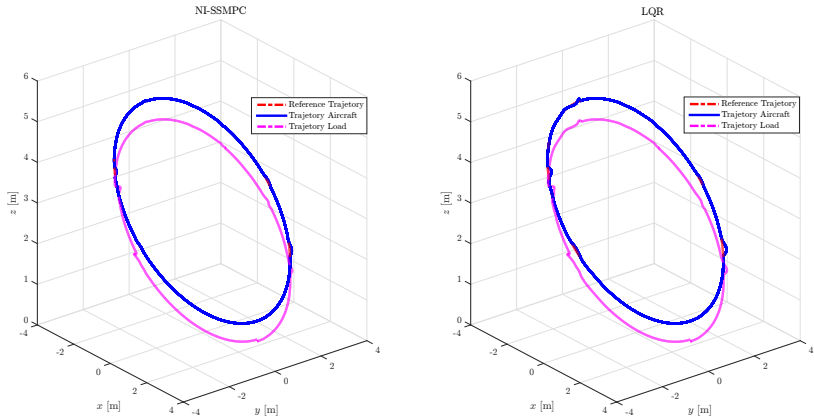
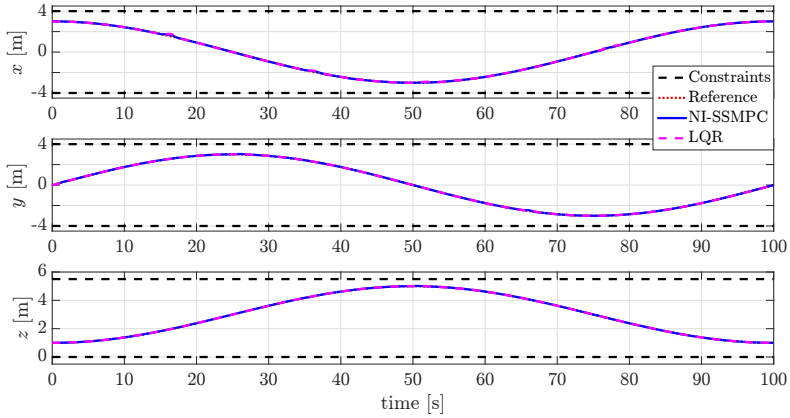


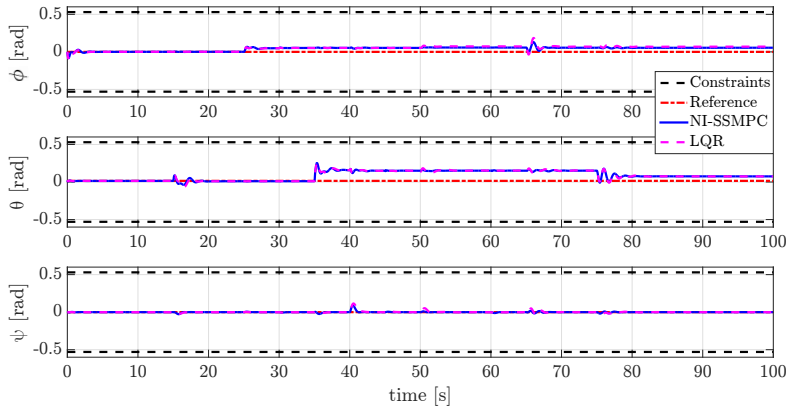
Figure 14 – Reference Trajectory of the Tilt-rotor with suspended load made by I-SSMPC and LQR controller.

In Figure 15a, corroborate that I-SSMPC has a good trajectory tracking since the translational motion response converges quickly to the reference, when a perturbation affects the system and the overshoot of LQR response is higher than the NI-SSMPC. Otherwise, the attitude states are presented in Figure 15b, where the steady-state value reached by the LQR controller is slightly bigger than the NI-SSMPC in the case of roll angle, but the others values have the same behavior. Additionally, figures 15 and 16 show oscillatory responses for both controllers when load disturbances affects the system, being the oscillation amplitude greater with the NI-SSMPC, because, as presented in Figure 17b, the control action in this controller is smoother.

Figure 16b shows the same situation presented in the last controller, where the overshoots of load states are bigger in the



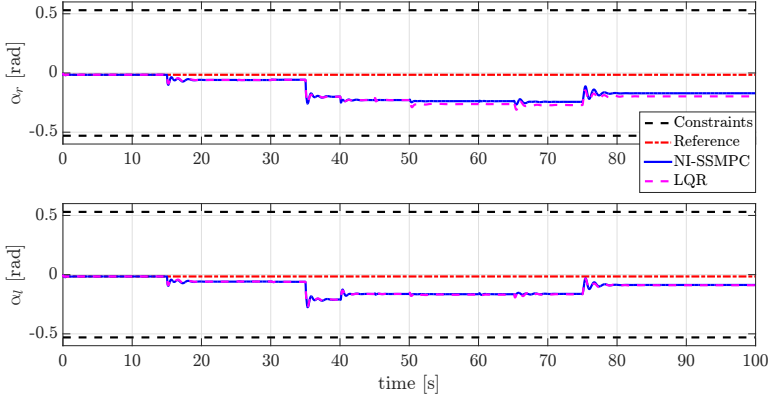
(a) Translational position (x, y, z) of the Tilt-rotor UAV with respect to the inertial frame.



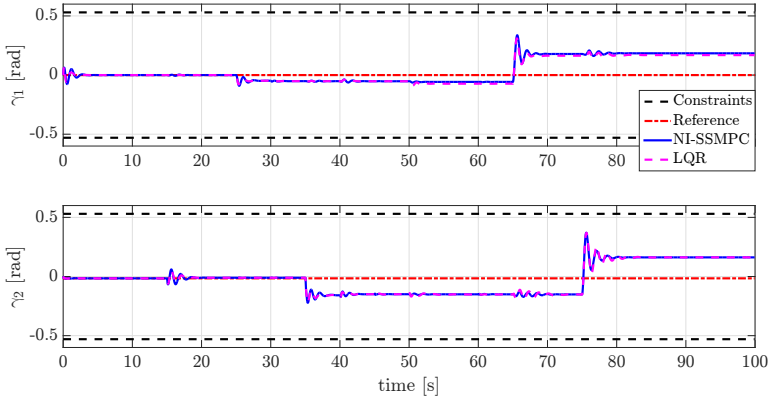
(b) Rotational position (ϕ, θ, ψ) of the Tilt-rotor UAV around the local axis.

Figure 15 – Time evolution of translational and rotational positions performed by the Tilt-rotor UAV applying the proposed NI-SSMPC and LQR controller.

NI-SSMPC. This demonstrates that both controllers, I-SSMPC and NI-SSMPC, produce more oscillations in the suspended load.



(a) Servomotor angular position (α_r , α_t) of the Tilt-rotor UAV.



(b) Suspended Load angular position (γ_1 , γ_2) of the Tilt-rotor UAV.

Figure 16 – Time evolution of rotors and suspended load angular position, performed by the Tilt-rotor UAV applying the proposed NI-SSMPC and LQR controller

Considering Figure 17, it is noticed that variations of the thrust forces and torques are smaller in the NI-SSMPC. This fact reflects that the proposed MPC produces smoothest control actions than LQR controller.

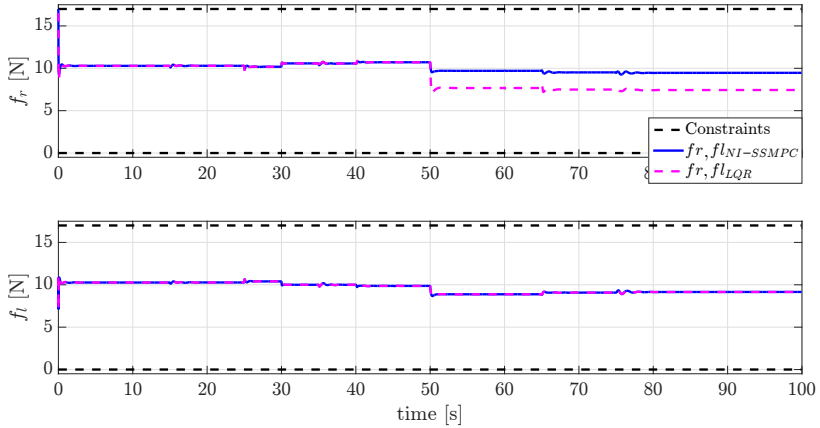
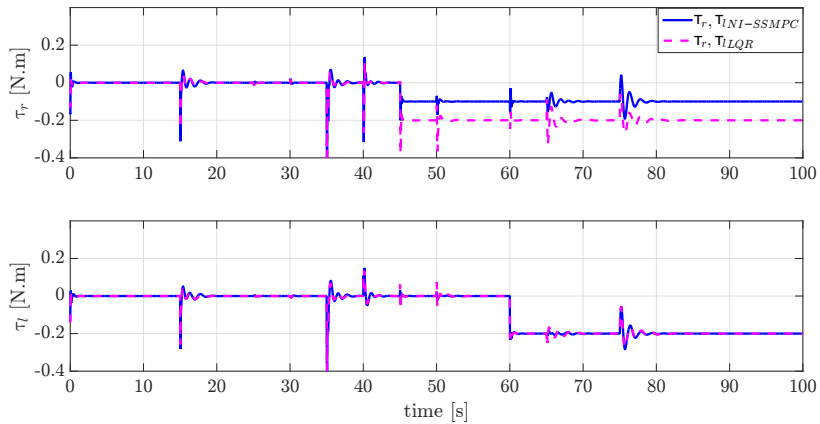
(a) Thrust Forces (f_r , f_i)(b) Servomotors Torques (τ_r , τ_l)

Figure 17 – Time evolution of thrust forces and torques applied to the Tilt-rotor UAV by the proposed NI-SSMPC and LQR controller.

5.4 SIMULATION WITH PARAMETRIC UNCERTAINTIES

This section presents the second and third simulation scenarios, where parametric uncertainty are considered in the inertia tensor and body mass of the Tilt-rotor. The parametric uncertainties values are -30% and 30% in the non-linear model. These simulations are used to verify the robustness of proposed controllers.

It is possible to notice in Figure 18 that the proposed controllers are able to track the reference trajectory of states x , y , z and ψ , with parametric uncertainties and constant disturbances. It can be seen that when uncertainty is $+30\%$, the system has fewer oscillations in the states since it is heavier. Additionally, it can be noticed in the time response of the angular position α_r , in Figure 19, when a disturbance affects the system, applied the NI-SSMPC controller with the uncertainty of -30% , the transient response almost reaches the low limit of the constraint.

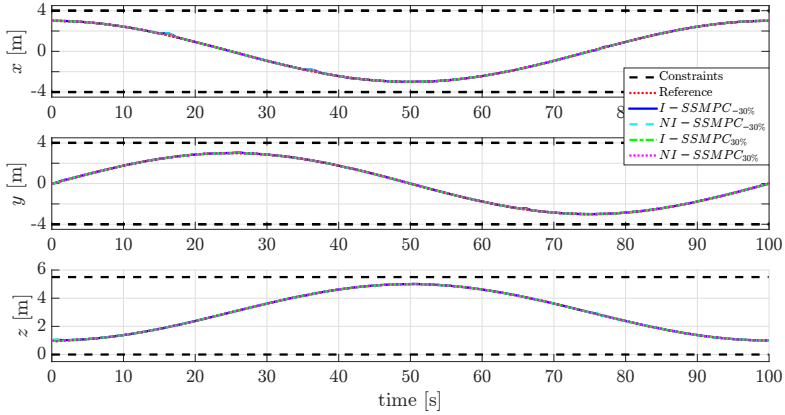
The last situation reflects the robustness of the controller is not only subject to its tuning parameters, but also depends on the constraints that are imposed on the system, i.e., from Figure 19b, it is possible to see that the state reached the restriction, and when the controller enters in any constraint, this generates a very aggressive control action, which can destabilize the control system, as shown in Figure 20 in time 75 [s].

On the other hand, when an uncertainty of $+30\%$ is applied to Tilt Rotor UAV, the transient responses of the states are not too far from the equilibrium point. Therefore, the problem with the limits of the constraints is not present.

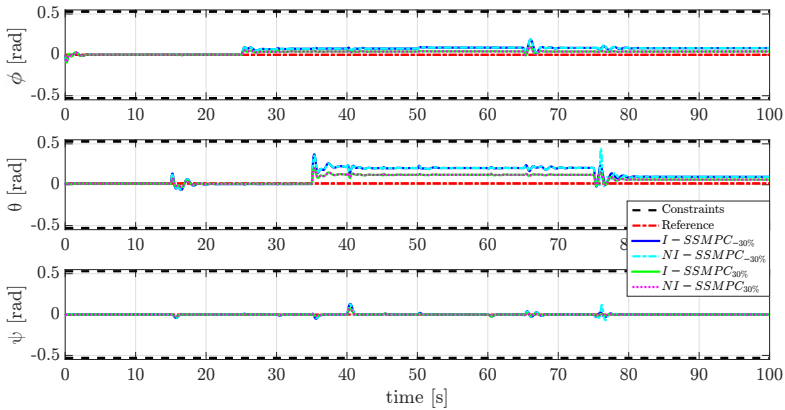
However, another situation must be considered in this case. As seen in Figure 20a, when the system has a parametric uncertainty of $+30\%$, the thrust forces have greater steady state value than calculated in Table 2, because the aircraft mass was increased by the uncertainty. This results in a reduction of capability to reject disturbances by the controller, and loss of lift.

5.5 SIMULATION ANALYSIS

In this section, some numerical results derived from simulations are presented in order to perform quantitative comparisons between controllers. Some performance indexes have been computed. On one hand, the mean square error (MSE) is used to assess the difference



(a) Translational position (x , y , z) of the Tilt-rotor UAV with respect to the inertial frame.



(b) Rotational states position (ϕ , θ , ψ) of the Tilt-rotor UAV around local axis.

Figure 18 – Time response of translational and rotational positions with parametric uncertainties in Tilt-rotor model.

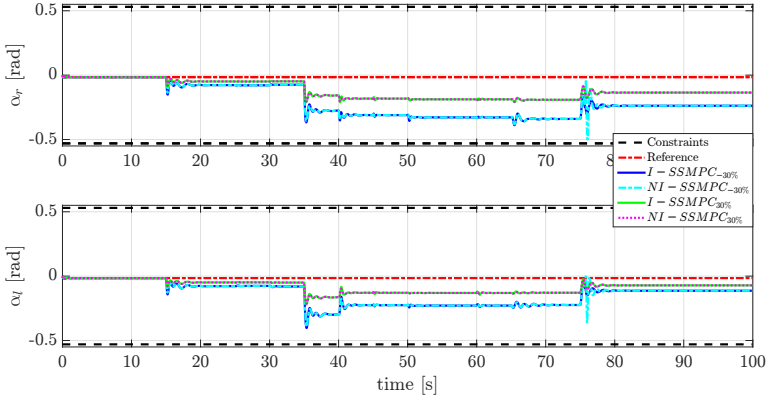
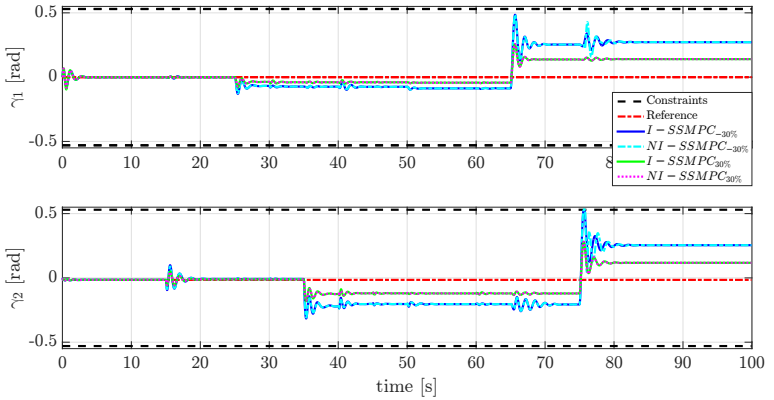
(a) Servomotor angular position (α_r , α_l) of the Tilt-rotor UAV.(b) Suspended Load angular position (γ_1 , γ_2) of the Tilt-rotor UAV.

Figure 19 – Time response of rotors and suspended load angular position with parametric uncertainties in Tilt-rotor model.

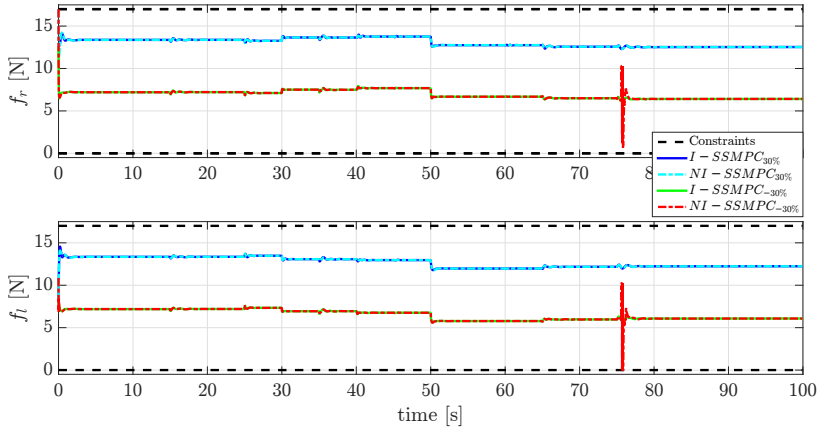
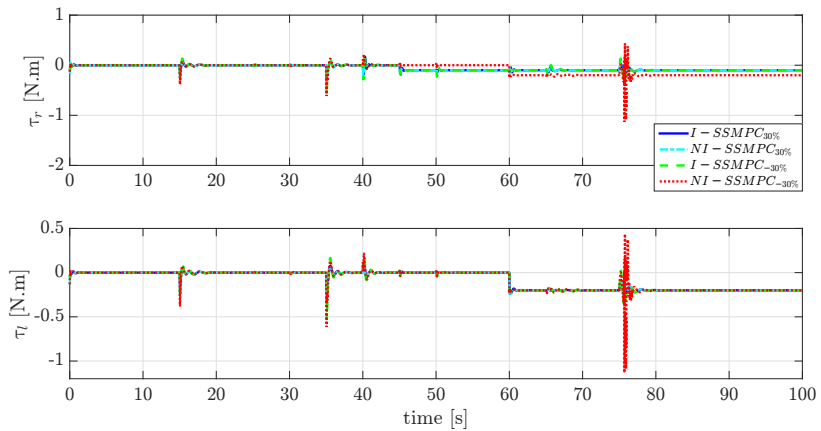
(a) Thrust Forces (f_r , f_l)(b) Servomotors Torques (τ_r , τ_l)

Figure 20 – Control signal of thrust forces and torques applied to Tilt-rotor UAV with parametric uncertainties.

between the linear and angular position and the reference trajectory. On the other hand, the total variation index is used to measure the total control action used by each controller.

The total variation index is presented in Table 5, where it can be noticed that the proposed controllers have smoother control effort than LQR controller, because MPC considers constraints in the control signal and this one is calculated with the future information of the reference. However, the best performance is presented by I-SSMPC; this controller uses the incremental approach considering the last applied control signal to calculate the new one, producing a smoother control action.

Table 5 – TV index performance analysis

States	LQR	I-SSMPC	NI-SSMPC
f_r	24.681662	14.171920	17.427798
f_l	22.775579	13.044563	16.526173
τ_l	6.005928	5.260038	6.226658
τ_r	4.594553	3.891857	4.362787

The MSE indexes obtained for each controller are shown in Table 6. It can be observed that proposed controllers obtained better performance than LQR, except for the load states. This situation occurs because, in order to avoid a large swing of the load, it is necessary to stabilize the Tiltrotor UAV quickly, which requires an aggressive control action. As presented in the Table 5, the control signal is smoother in the proposed controllers. Furthermore, the indexes obtained for the position states shown a good performance obtained.

Table 6 – MSE index performance analysis

States	LQR	I-SSMPC	NI-SSMPC
x	0.001327	0.000326	0.000293
y	0.000267	0.000075	0.000068
z	0.000043	0.000005	0.000006
ϕ	0.002986	0.002293	0.002290
θ	0.009667	0.008594	0.008568
ψ	0.000148	0.000059	0.000050
α_r	0.030251	0.025467	0.025455
α_l	0.012984	0.011369	0.011358
γ_1	0.008005	0.013034	0.013028
γ_2	0.012706	0.015684	0.015669

On the other hand, the NI-SSMPC approach reached better performance than the I-SSMPC with respect to the MSE index. This better states performance is obtained with a more aggressive control action provided by the NI-SSMPC (see table 5). Additionally, the presented simulations showed that LQR and MPC controllers have very similar behaviors, since MPC cost function is similar to the LQR controller when restrictions are not considered and there is an infinite prediction horizon. However, MPC formulation is most appropriate and presents better performance than LQR controller when considering restrictions and finite prediction horizon.

5.6 SUMMARY

In this chapter the simulation results for the Tilt-rotor UAV with suspended load were presented. Three scenarios were considered to determine the performance of the proposed controllers. First scenario considers nominal parameters, a smooth trajectory and constant disturbances applied to all degrees of freedom. Then, the second and third scenarios consider parametric uncertainties in the inertia tensor and body mass. Two model predictive controllers were tested. An incremental predictive control is simulated considering the first scenario. This controller presents good performance solving the path tracking problem, being able to reject sustained disturbances with a smooth control action. On the other hand, the non-incremental predictive controller has presented better performance when performing path tracking than the previous one, with good capabilities to reject constant disturbances, although, the control action generated is more aggressive. In order to evaluate the inherent robustness of the proposed controllers against small perturbations, they were subjected to the second and third scenarios. Both controllers have good path tracking capabilities in presence of parametric uncertainties and constant disturbances. However, this robustness could be lost if the perturbations send the states too far from the operation point or if the state is very close to the restriction. Finally, some quantitative performance indexes were calculated to demonstrate the effectiveness of these controllers. These indexes were compared to a LQR controller which is similar to the proposed ones when no constraints are considered. In these results, it was corroborated the smoothness of the control action compared to the LQR controller and the good performance in the output states.

6 PRACTICAL IMPLEMENTATION

In order to test the proposed controller in a real situation, a Tilt-rotor UAV was developed by the ProVANT project. In this chapter it is described the Tilt-rotor UAV, including its hardware configuration and software implementation. Then, some discussion is performed about preliminary results obtained with the UAV.

6.1 PROVANT PROJECT

The ProVANT project aims the assembly of a small scale Tilt-rotor UAV. Additionally, this project has the interest in developing an open platform in order to test controllers, estimators and embedded systems designed for UAVs. Into this project, the first version of the UAV was developed in Donadel (2015), a second version is being assembled by the UFMG ProVANT team, and a third version assembled in this dissertation. It is worth to say that the third version of this aircraft was developed from scratch.

The first version of the Tilt-rotor UAV was based on a mathematical model that did not consider the dynamics generated by the servomotors. For this reason, the control signals used in that approach were thrust forces and reference position for tiltable mechanisms (servomotors). Moreover, this version does not consider the inclination β in the servomotors. On the other hand, it has only one computer to read sensors, estimate states and control actuators. Additionally, the thrust forces generated by the rotors of this version are almost equal to the weight of the aircraft, reducing maneuverability of the aircraft. In the second version, it is considered other body structure and material in order to reduce the weight of the aircraft, other servomotors controlled by torque reference were used in order to consider the dynamics generated by the servomotors.

In the third version, assembled in this master thesis, the structure of the first version and the servomotors used in the second version were considered. Therefore, rotors more powerful were used in order to increase the payload of the aircraft. With the aim of implementing the controllers developed in this dissertation, two computers were considered, one computer is used in the same way as in the first and second versions, and the other computer is used to implement the model predictive controller.

6.2 GENERAL DESCRIPTION

The physical structure of the aircraft is depicted in Figure 21. It is similar to the first version, where carbon fiber is used in the arm to connect the rotors with the main body. However, in this version these arms have a support that gives five degrees of inclination to the servomotors, equal to the assumption made in the mathematical model. The main body structure is composed by glass fiber that supports the embedded system and the electronic devices. Also, it was included a landing gear made with aluminum, in order to give support to the aircraft. Additionally, it was designed a support for the camera and the suspended load, made in glass fiber. The smaller parts of the aircraft were designed in the software *SolidWorks*® and printed using 3D printers available at the Automation and Systems Department (DAS) at UFSC and at the Electronic Engineering Department (DELTA) at UFMG.



Figure 21 – Tilt-rotor UAV second prototype assembly

In Figure 22, it is presented how the embedded system is structured with respect to the tasks. The Sensor and Actuator Processing subsystem is responsible for managing the electro-mechanical devices of the Tilt-rotor UAV, and the Control and Estimation subsystem for calculating the control action and execute the load position estimation. Therefore, Sensor and Actuator Processing subsystem is responsible for managing sensors and actuators, estimating the attitude and altitude, reading radio control

data and communicating with the other subsystem. Moreover, Control and Estimation subsystem is responsible for computing the control action, communicating with the base station in order to obtain a telemetry of the system, estimating load position and communicating with the other subsystem.

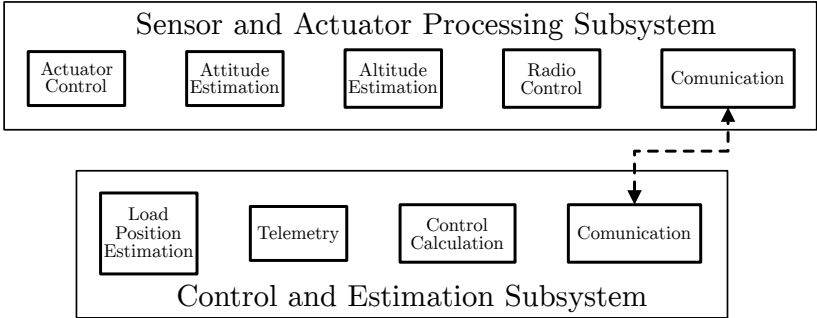


Figure 22 – Tilt-rotor UAV general description

In this dissertation, as presented in Figure 23, other embedded computer is used in order to calculate the control action due to the high computational cost of the proposed MPC formulation. The Sensor and Actuator Processing subsystem is supported by Discovery STM32f4 board computer, and the Control and Estimation subsystem by Beagle Bone Black computer. The two systems are exchanging information through a serial interface using the MultiWii protocol. The data sent is related to the estimated states, system status and control actions.

The specifications of each computer are presented in Table 7. As mentioned before, the Beagle Bone Black presents better processing capability than the Discovery board computer, consequently, this single computer was chosen to calculate control algorithms. Specifically, this computer is very used in image processing due to its graphic engine which can be used for matrix calculation.

6.3 SENSOR AND ACTUATOR PROCESSING SUBSYSTEM

In this section is described the Sensor and Actuator Processing subsystem. As presented in Figure 22, this subsystem is responsible for managing sensors and actuators of the Tilt-rotor UAV and uses as computational system the Discovery STM32F4 board computer, which is connected to other devices (Figure 23) throughout a shield developed

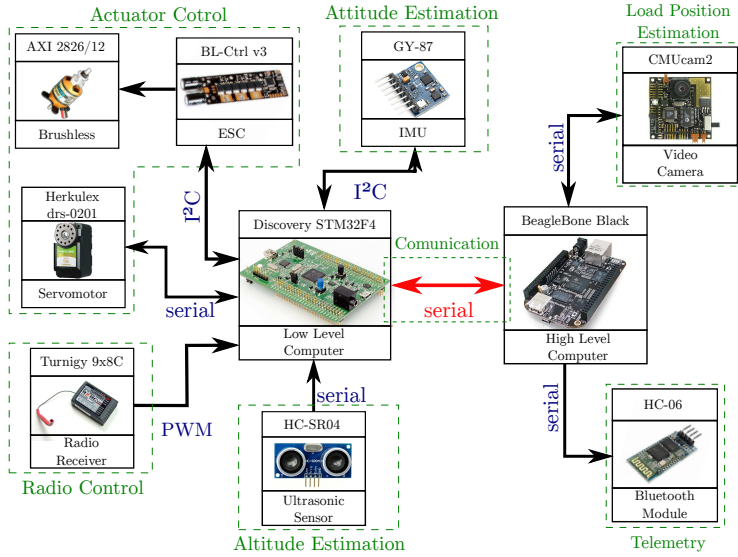


Figure 23 – Electronic devices attached to the Tilt-rotor

Item	BeagleBone Black	Discovery STM32F4
Processor	Sitara AM3358BZCZ 1GHz Cortex-A8	STM32F407VGT 168MHz Cortex-M4
SDRAM Memory	512 MB	192KB
On-board Flash	4GB 8-bits eMMC	1MB
Floating Point	NEON	FPU
Graphics Engine	SGX530 3D, 20M	Not Available
Power	5V@210-460 mA	5V@100-500 mA

Table 7 – Single board computers specifications

by ProVANT team¹. Most of the software used in the Discovery computer is based on the first version presented in Donadel (2015). The software was structured with a single thread of execution, which was responsible for estimating states, calculating control signals, reading sensors, controlling actuators, and sending telemetry data to the base station.

Moreover, as this dissertation uses two computers that need to exchange information between them, it was necessary to implement another task responsible for communication. In this way, the software was restructured into two threads, as presented in Figure 24.

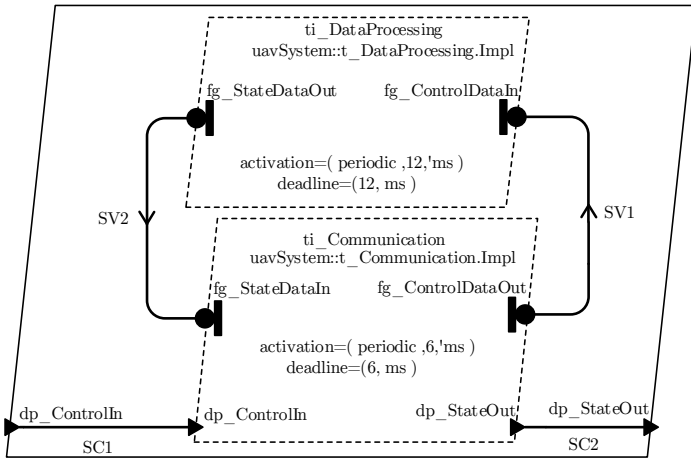


Figure 24 – AADL representation of the software on the Sensor and Actuator Processing subsystem.

In this figure, the suffixes *In* and *Out* represent input and output data of the execution thread, respectively. The term *SV_i* represents the name of the connectors between the shared variables of the execution threads. Nevertheless, these connectors represent only data interchange between threads and not a variable defined within the program.

6.3.1 DataProcessing execution thread

In order to represent the algorithm of the Tilt-rotor UAV software, it is used the flowchart representation based on the notation

¹Further information can be found in: <https://github.com/proVANTbr/provant-hardware/t>

presented in Ashok (2003) and is described in the Appendix B.1.

In these graphics the modules developed in Donadel (2015) were marked with the symbol *. The thread of `DataProcessing` is described in Figure 25. This thread is responsible for reading the sensors, estimating the states and writing the control values to the actuators. The execution period was set with respect to the servomotors sampling time, which update the angular position and velocities every 12 ms. The algorithm begins with the initialization of variables and communication interfaces, for example, I²C interface for reading the IMU device or USART interface for controlling the servomotors. The attitude estimation process reads the linear acceleration, angular velocities and the magnetic field direction of the Earth from the IMU device, and estimates the states. This subprocess is detailed in Subsection 6.3.1.1. Radio data reading is a pulse-width modulation (PWM) interpreter which separates 6-channel information into six variables into the software. This information is received via a Turnigy 9x radio receiver of 2.4 GHz from the radio control. Following, the altitude estimates are estimated, and the position and velocities of the servomotors are read. Then, all shared variables are updated. The shared variables are used to exchange information between threads. In this case, the output shared variables (`fg_StateDataOut`) are the states estimated and the status of the system. Besides, the input shared variable contains the control action values (`fg_ControlDataIn`). Finally, the torque and forces, calculated by the controller, are transformed into PWM values and applied to the actuators devices and the thread goes to sleep for the remaining time of the execution period.

6.3.1.1 Attitude Estimation

The Attitude Estimation subprocess, in Figure 25, estimates the attitude of Tilt-rotor UAV. The formulation used is described in Donadel (2015) and is based on Mahony, Kumar & Corke (2012). This algorithm uses the measurements of linear acceleration, angular velocity, and the magnetic field direction of the earth to estimate angular position and velocity. These informations are provided by the GY-87 inertial measurement unit (IMU), which communicates with the Discovery board through I²C interface communication. This device is composed by three sensors: MPU6050 measures linear acceleration and angular velocities; HMC5883L is a magnetometer; and the BMP180 is a barometer. As this device was not used in the first version assembled in

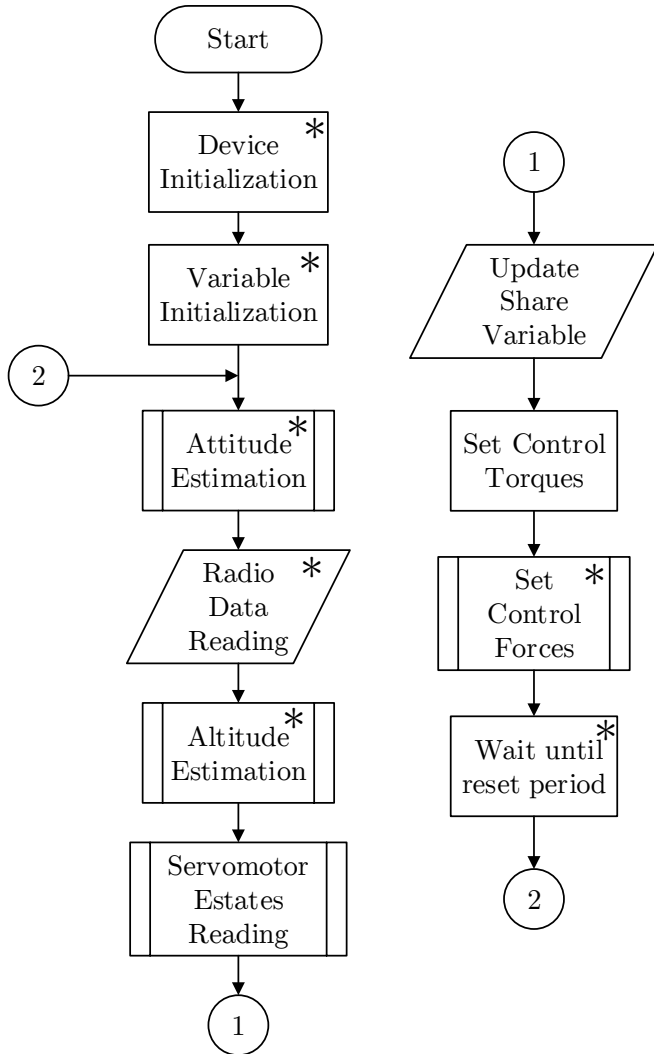


Figure 25 – Flowchart of DataProcessing execution thread

Donadel (2015), the libraries of each sensor were developed and can be found in the software repository of the ProVANT project². The sensors' configuration setup are presented in Table 8, where full-scale range is the upper and lower limits of what the accelerometer or gyroscope measurements, and LSB sensitivity is the smallest amount of change that can be detected by the sensor. This value is used to compute the physic value given by the raw data of the sensor.

Item	Full Scale Range	LSB Sensitivity
Accelerometer	$\pm 4 [g]$	8192 <i>LSB/g</i>
Gyroscope	$\pm 2000 [H_z]$	16.4 <i>LSB/°/s</i>

Table 8 – Setup configuration of MPU6050 sensor

6.3.1.2 Altitude Estimation

The altitude estimation of the Tilt-Rotor UAV is performed using the HC-SR04 ultrasonic ranging module. This sensor, managed by an Arduino mini, measures the altitude value with respect to the floor. Then, this data is sent to the Discovery board through USART communication interface at 115.2 Kbps. The vertical velocity is estimated using the altitude estimation as presented in Donadel (2015). The firmware used in the Arduino mini is available in ProVANT devices repository³.

6.3.1.3 Servomotor States Reading

This process is responsible for reading the angular position and velocities of each servomotor. The servomotors used to control the inclination of the rotors are Herkulex DRS-0201, which are controlled by torque reference. This servomotor provides the actual position and velocity, but the velocity values need to be filtered, since this signal is noisy. A low pass filtered is used with cutoff frequency at 5 Hz. The values given by the servomotor are updated every 12 ms. Moreover, it uses a serial communication with transmission speed of 666 KBps. The

²Further information can be found in: <https://github.com/proVANTbr/provant-software/tree/development>

³ProVant devices repository available in: <https://github.com/proVANTbr/provant-devices>

software library for these servomotors was implemented by ProVant team of UFMG and in this dissertation was added to the Tilt-rotor embedded software.⁴

6.3.1.4 Control Actuation

The servomotors are controlled by torque reference, through PWM values. These values are calculated in direct relationship to the reference torque, where the maximum value configured in the servomotor is 1023, representing the maximum torque value of 2 N.m. In the case of thrust forces, two BL-CTR V3 electronic speed controllers (ESC) controls the brushless motors, which are the actuator that generate the thrust forces. The ESC device is controlled by a PWM value which is sent through I²C communication from Discovery board. The brushless motors used in this version are AXI 2826/12 GOLD, which can generate 18.5 N of thrust force. As the control action is given in Newton, the relation between the force generated by the motor and the PWM set in the ESC was obtained from a experimental test. The test was performed increasing the motor velocity with intervals of 5 units until reaching the value of 255. Then, it was possible to approximate the behavior of the motor to a fifth-order polynomial, as presented in Figure 26.

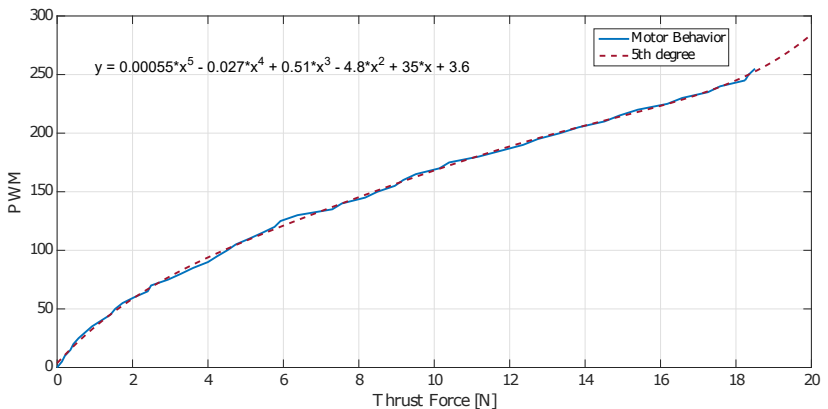


Figure 26 – Curve of behavioral for the AXI 2826/12 GOLD motor

⁴Further information can be found in: <https://github.com/proVANTbr/provant-software/tr>

6.3.2 Communication execution thread

The thread Communication is responsible to communicate with the other embedded computer. This communication is performed through a serial interface, using the MultiWii protocol. This protocol was used in the first version of the Tiltrotor UAV for communicating unidirectionally with the base station. In this dissertation, the protocol was extended in order to become bidirectional with the Beaglebone Black computer. The algorithm implemented in the Communication thread is presented in Figure 27. First, the communication interface is initialized, then, the input shared variables (`fg_StateDataIn`), containing the states of the Tilt-rotor and the system status, are updated. Next, it sends the states values (`dp_StateOut`) to the other embedded computer and receives back the control action (`dp_ControlIn`). When the state information is sent, the algorithm waits for reply of control data. If these data do not arrive within 3 ms, it is considered as lost communication and restarted. On the other hand, when the control data are received, this information is evaluated in order to determine if there are any errors. If there is an error, the last control data are used, otherwise, the shared variables (`fg_ControlDataOut`) are updated. Finally, the thread goes to sleep for the remaining time.

6.4 CONTROL AND ESTIMATION SUBSYSTEM

In this section, the Control and Estimation subsystem, supported by the Beaglebone Black single board computer, is described. This level is responsible for calculating the control action, estimating the angular position of the suspended load, sending telemetry information to the base station and communicating with the Sensor and Actuator Processing subsystem. Considering that Beaglebone Black computer must execute the proposed MPC in a short period of time, the operating system Debian Jessie 8.2 compiled with hardware floating point was considered. This operating system provides support for two main features of Beaglebone Black computer, with high acceleration when working with vectors or matrices. The first feature is the NEON technology. This technology consist of an accelerator processor SIMD (Single Instruction Multiple Data), in which during the execution of one instruction, the same operation will occur on up to 16 data sets in parallel. The second feature is the hardware floating-point, which

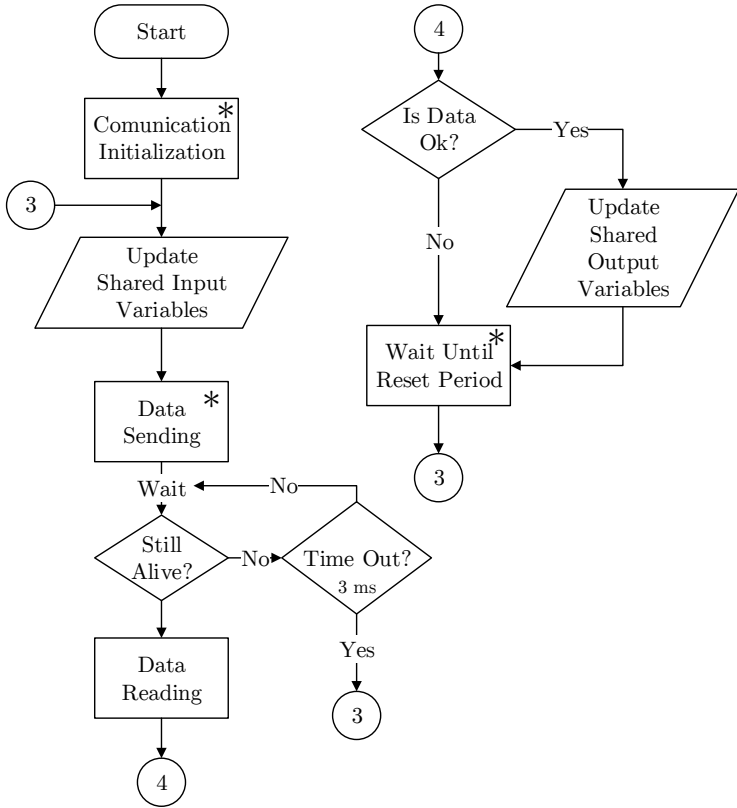


Figure 27 – Flowchart of Communication execution thread.

accelerates the operations performed with a single precision floating point.

Moreover, the formulation proposed in Chapter 3 uses operations with matrices and a quadratic programming solver. In this way, two libraries were chosen to develop the MPC formulation. Eigen library, with NEON technology capabilities, used for computing matrices operations, and qpOASES library used for fast calculation of the QP formulation.

As presented in Figure 28, the software, in the BeagleBone Black computer, is structured with three execution threads, corresponding to the tasks of communication with Discovery board, control calculation, and sending telemetry information to the base station. Thread `CommLowLevel` is responsible for receiving the states of the system (`dp_InputStates`) and sending the control values (`dp_ControlOut`) throughout the serial communication, `ContinuousControl` calculates the control action and `DataProcessing` sends the telemetry information to the base station. In this figure, the suffixes *in* and *out* represent input and output data of the execution thread, respectively. The term `SVi` represents the name of the connectors between the shared variables of the execution threads. Nevertheless, these connectors represent only data interchange between threads and not a variable defined within the program.

6.4.1 `CommLowLevel` execution thread

This thread is responsible for receiving the states from the Sensor and Actuator Processing subsystem, and sending the control data. As presented in Figure 29, this program begins with the initialization of serial interface, setting the transmission velocity in 921.6 Kbps. Then, the input shared variables are updated; in this case the input variable contains the control actions that will be sent to the Sensor and Actuator Processing subsystem. Next, the program enters a wait state for the data sent from the Sensor and Actuator Processing subsystem. When this information arrives, a validation is performed in order to detect errors in the data. If the data is correct the output shared variables are updated, otherwise the last values are maintained. Finally, the thread goes to sleep for the remaining time of 6 ms. On the other hand, if the data information does not arrive in 3 ms the communication is restarted.

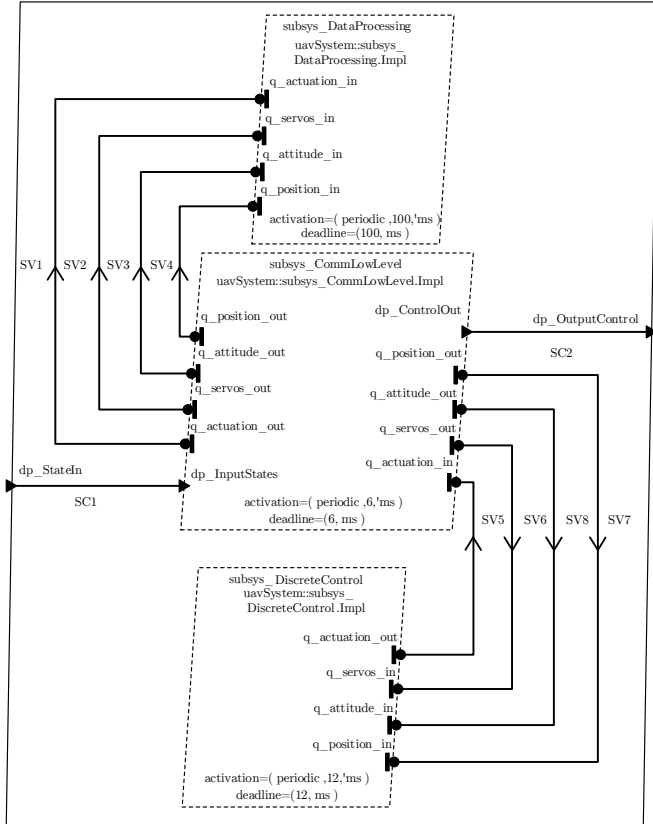


Figure 28 – AADL representation of the software on the Control and Estimation subsystem.

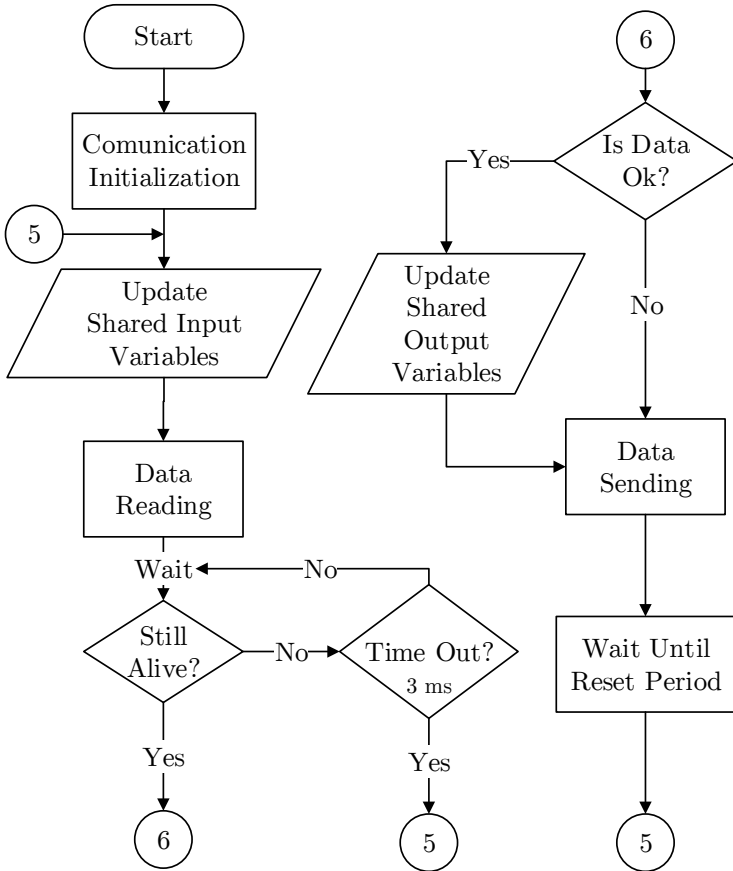


Figure 29 – Flowchart of CommLowLevel execution thread.

6.4.2 DataProcessing execution thread

This thread is responsible for sending telemetry information to the base station. These data are sent through HC-06 Bluetooth module, which communicates with the BeagleBone Black through serial interface. As mentioned before, the MultiWii protocol is used in the communication between the Tilt-rotor and the base station. The data are the following:

- Angular position of the Tilt-rotor expressed in degrees.
- The translational position and velocities of the Tilt-rotor UAV in meters and meters per second, respectively.
- The angular position and velocities of the Tilt-rotor UAV in radian and radians per second, respectively.
- The angular position and velocities of the rotors in radian and radians per second, respectively.
- The calculated control action in newtons and newtons meter.

The algorithm of this thread begins with the initialization of the serial interface. Then, the shared variable are updated and sent to the base station. The sampling time was configured with 100 ms.

6.4.3 DiscreteControl execution thread

This thread is responsible to calculate the control action of the Tilt-rotor UAV. This program was developed with the ability to use the controllers as objects that can be called by the software. This object receives as parameters the states of the system and the status of software, then it returns the control action. As presented in Figure 30, the code starts by initializing the control variable. In this process if the controller used is an LQR, the feedback gain matrix is initialized. Moreover, if MPC controller is considered, the tilt-rotor model and the MPC tuning parameters are initialized. Next, the input shared variables are updated and the states are sent to the controller's object to calculate the control action. Finally, the values received from the controller are updated into the output shared variables. The execution period of this thread was setup at 12 ms, like the DataProcessing.

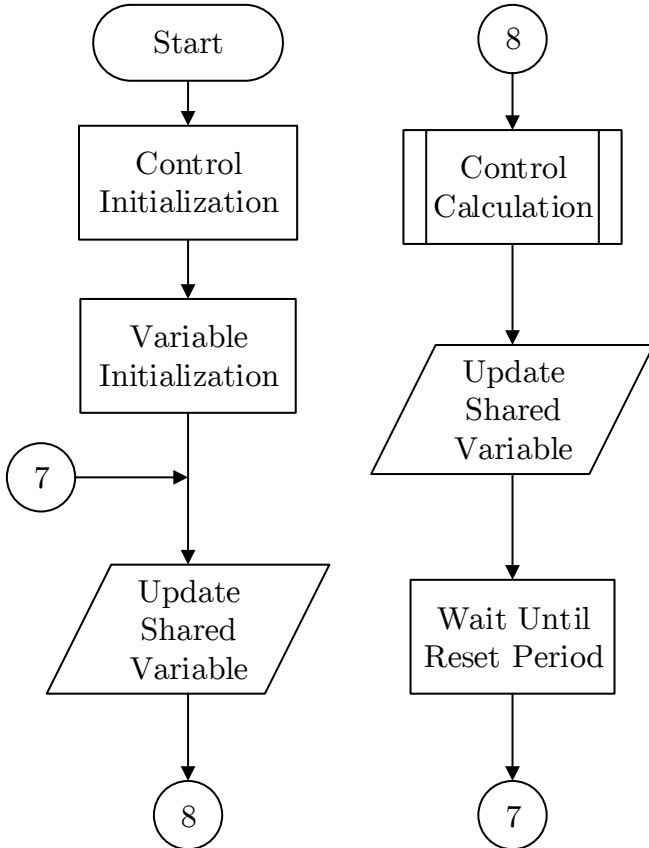


Figure 30 – Flowchart of DistcreteControl execution thread.

6.4.3.1 Control Calculation

In this section is described the generalized algorithm of the proposed controllers presented in Figure 31. This algorithm uses two libraries to calculate the control action. First, the Eigen library⁵ is used for matrix operations. It can support all matrix sizes, any numeric type, and has explicit vectorization, allowing to work with NEON technology. The second library used to solve the QP problem is qpOASES⁶. It is an implementation of the online active set strategy. It has several theoretical features that make it suited for model predictive control (MPC) applications. Furthermore, it can deal with semi-definite, ill-posed or degenerated QP problems.

In order to use this library, some modifications in the constraints are performed in the proposed formulation, since the QP problem in this library is quite different, as follows:

$$\begin{aligned} J(x) &= \frac{1}{2}x'\mathcal{H}x + x'g \\ lb &\leq x \leq ub \\ lbA &\leq Ax \leq ubA. \end{aligned}$$

In this context, the constraint's inequalities are rewritten in the following form for the I-SSMPC:

$$\begin{bmatrix} \hat{\mathbf{u}}_{min} - \tilde{\mathbf{u}}(k-1) \\ \hat{\tilde{\mathbf{x}}}_{amin} - \mathbf{Q}\tilde{\mathbf{x}}_a(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{I}_m \\ \mathbf{P} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\mathbf{u}}_{max} - \tilde{\mathbf{u}}(k-1) \\ \hat{\tilde{\mathbf{x}}}_{amax} - \mathbf{Q}\tilde{\mathbf{x}}_a(k) \end{bmatrix} \quad (6.1)$$

and for the NI-SSMPC as presented below:

$$\begin{bmatrix} \hat{\mathbf{u}}_{min} \\ \hat{\tilde{\mathbf{x}}}_{smin} - \mathbf{Q}\tilde{\mathbf{x}}_s(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{I}_m \\ \mathbf{P} \end{bmatrix} \hat{\mathbf{u}} \leq \begin{bmatrix} \hat{\mathbf{u}}_{max} \\ \hat{\tilde{\mathbf{x}}}_{smax} - \mathbf{Q}\tilde{\mathbf{x}}_s(k) \end{bmatrix}. \quad (6.2)$$

6.4.3.2 Load Position Estimation

In order to solve the problem of measuring the load position, the test bench shown in Figure 32 was implemented. The load has

⁵Eigen library is available at: http://eigen.tuxfamily.org/index.php?title=Main_Page

⁶qpOases library is available at: <https://projects.coin-or.org/qpOASES>

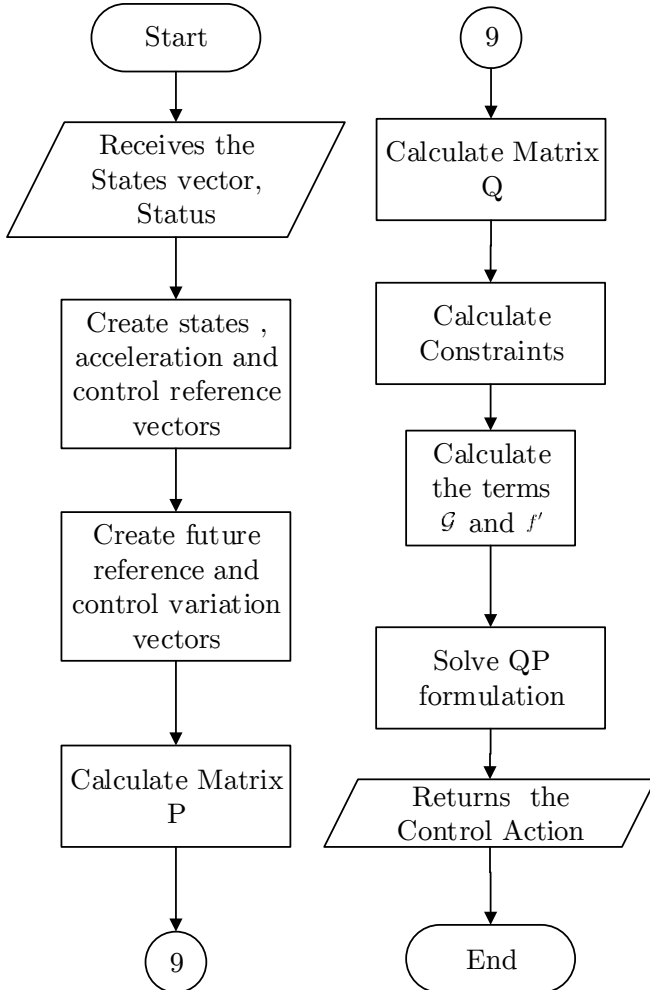


Figure 31 – Flowchart of MPC calculation

a mass of 80 grams, which is hanging by four nylon cords. These cords are supported by a glass fiber base. A camera is fixed on this base through an anti-vibration support. The load position is measured using image processing with the CMUCAM2 camera, which provides simple vision capabilities to small-embedded systems in the form of an intelligent sensor. This camera is able to track any color previously configured, then it sends a packet with some information through serial interface. This packet has the x and y coordinates of the object center with respect to the upper left corner of the frame, the bounding box, the number of pixels tracked and a confidence value. A software was developed to test the camera. This program sends the color of the load to be tracked by the camera. Then, the coordinates of the center of the object are read and the position with respect to the camera is calculated with the following equations:

$$\begin{aligned}x_l &= \frac{x_f}{2} - x_o \text{ pixels,} \\y_l &= \frac{y_f}{2} - y_o \text{ pixels,}\end{aligned}\tag{6.3}$$

where x_l and y_l are the position of the load with respect to the camera, x_o and y_o are the coordinates of the object center, x_f and y_f are the size of the frame.

6.5 PRELIMINARY RESULTS

Some preliminary results obtained with the system were obtained. However, during the preliminary tests some issues were identified which did not allow to obtain flight results. The first identified problem is related to the communication between computers. According to tests performed, sometimes, the package information is lost because the MultiWii protocol does not guarantee package reception. However, it is able to detect errors in the submitted information. Considering this capability, as first solution, it was implemented an acknowledge message in the Multiwii protocol. First, the low-level subsystem sends the states to the high-level subsystem, then, the low-level subsystem waits for the acknowledge message of the other subsystem, if the acknowledge message is not given in 3 ms, the low-level subsystem sends again the states. In the case of the high-level subsystem, it begins waiting for the states from the low-level subsystem; when they arrive, if the data is correct, it sends an

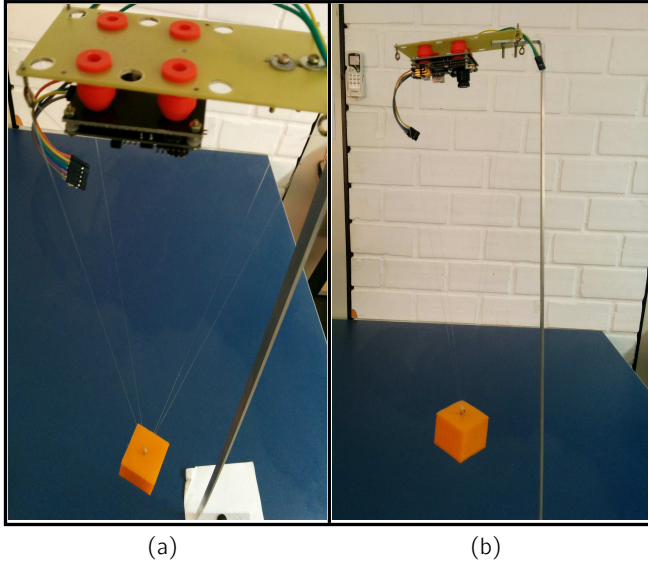


Figure 32 – Structure of the test bench for the CMUCAM2 camera

acknowledge message, on the opposite it waits again for the package. The problem with this solution is that the Multiwii protocol works in phases. On the one hand, it sends the package, and on the other hand, it receives the package. If one thread lost any phase, the program locks up because the computer does not know when the other one is writing or reading.

In this context, it was developed a partial solution when an error is detected. This solution used the last value that was received correctly and reduce the period of the communication threads on each computer to 6 ms, in order to have an exchange of information twice faster than the control calculation.

Other problem is related to the tuning parametrization of the system. This is not trivial and it is one of the biggest challenges of these systems. Considering the explanation in Section 2.6, the dynamics of the servomotors was set in the mathematical model as close as possible to the actual model, in order to reduce the difficulty of the system parameterization. However, it was not possible to stabilize the system.

Nevertheless, some preliminary results were obtained with the system. The first result considers communication time of the system. This is an important parameter, as the control calculation is executed in

the BeagleBone Black computer and this value are applied by Discovery board, the time between reading and actuating must be less than 12 ms, which is the sample time of the system. In this context, the mean time measured between sending the states and obtaining the control signal was 2 ms. Considering the MPC implementation in the BeagleBone Black, it was measured the time for solving the QP problem and the error between the control calculated in simulation and in the BeagleBone Black. The results were a mean time of 8 ms and an error of 2%. Therefore, the system is able to read the sensors, calculate the control action and act in 10 ms, which is less than the time sample of the system.

6.6 SUMMARY

In this chapter, the practical implementation of the Tilt-rotor UAV was described. The general structure was based on the first version of this aircraft; however, some improvements were applied in the Tilt-rotor assembled in this work, like the use of servomotors controlled by torque reference, the addition of a second computer in order to implement the MPC formulation derived in this work, the addition of a IMU with barometer, and the use of camera to estimate the load position. Furthermore, a brief explanation of the whole system was provided, starting from the physical structure, passing by hardware implementation and ending with the description of the software. Also, some preliminary results were discussed, with the aim to prove that is possible to apply this approach in a real application.

7 CONCLUSIONS

In this Master dissertation two MPC strategies for the path tracking problem of a Tilt-rotor UAV with a swing-free suspended load were developed. These controllers were able to track a reference in x , y , z , and ψ while stabilizing the other states, considering constant disturbances affecting all degrees of freedom and parametric uncertainties in the inertia tensor and body mass of the Tilt-rotor UAV.

An accurate non-linear model of the Tilt-rotor UAV with suspended load was presented and implemented considering dynamics generated by the rotors when they tilt and by the suspended load, modeled as a simple pendulum. The non-linear model is composed by ten degrees of freedom comprising three translational and three rotational DOF for Tilt-rotor UAV, two rotational DOF for servomotors, and two rotational DOF for the suspended load.

Considering the proposed control strategies, a linear error state space system was obtained when the non-linear model was linearized throughout a generic trajectory. From this linearization procedure a time-varying linear model based on the states and control error was obtained. In this model, the state matrix varies as a function of the trajectory acceleration. In order to perform the path tracking of the Tilt-rotor UAV with suspended load, the state vector was augmented with an integral action in the translational position and yaw angle, providing constant disturbances and parametric uncertainties rejection capabilities to the system.

The predictive controllers proposed in this work used the incremental and non-incremental formulation. These controllers are time variant and are based on the error state space model of the Tilt-rotor UAV. With the aim to consider physical limitations of the actuators and the fact that the aircraft is flying in a confined environment, there were included inputs and state constraints in the MPC formulation.

On the other hand, a terminal cost was added in the cost function of the MPCs in order to guarantee stability. However, as the model is time varying, the terminal value was calculated with an LMI approach, where the vertices of the polytope are given by the system evaluated at the maximum and minimum acceleration values of the trajectory.

The proposed controllers were simulated with the non-linear model applying constant disturbances. Furthermore, the simulation results were compared to a LQR controller. The results showed

a successful accompaniment to the reference trajectory and a good disturbance rejection. Furthermore, both controllers presented better performance than the LQR controller. However, the I-SSMPC used smoother control action but has higher states error, compared with NI-SSMPC. It should be noticed, that both controllers have higher error in the suspended load states. Both controllers presented smoother control action than the LQR controller, also, as confirmed with the MSE value of the states x , y , z and yaw, the controllers had a good path tracking capabilities. It is worth to say that, in the simulation results both controllers were able to respect the constraints.

About the simulation with parametric uncertainties, both controllers presented an inherent robustness against $\pm 30\%$ variation in the values of inertia tensors and body mass. However, a very limited restriction could reduce the robustness of the system.

7.1 FUTURE WORKS

In the following a list of possible future works related to this dissertation, is proposed:

- Formulate the non-linear model of the Tilt-rotor with suspended load with Newton-Euler approach. In the mathematical model presented in this dissertation, the inertia matrix inversion was needed to obtain the non linear model. This procedure is complex, considering that the system is higher coupled and non-linear. In this case, it could be interesting to formulate the model with this approach.
- Implement soft constraints in the states that, in this work, were only stabilized. The operating point of those states is always changing, therefore, considering soft constraints could give more freedom to the system.
- Demonstrate stability of the MPC formulation presented in this dissertation.
- Execute flight test with the Tilt-rotor developed by ProVANT project using the formulation implemented in this dissertation.
- Extend the proposed LMI formulation in order to consider the system uncertainties in the terminal cost.

- Explore a communication protocol for embedded systems that guarantee package reception. The control action needs to be applied as quickly as possible. Therefore in this work it was calculated in a dedicate computational board, which requires a proper communication protocol capable of exchanging information faster and without data loss when updating variables.
- Develop and implement a load position estimator. The frame rate of the camera is not fast enough to provide the position of the load. Therefore, it is necessary to implement an estimator for these states.
- Study control tuning strategies for such systems. One of the challenges in practical implementation of this control system is the control tuning. Thus, it could be helpful if some methodologies are used.

BIBLIOGRAPHY

ALMEIDA, M. M. **Control Strategies of Tilt-rotor UAV for Load Transportation**. Thesis (Master Degree Thesis) — Federal University of Minas Gerais, Belo Horizonte, 2014.

ALMEIDA, M. M. et al. Full control of a tiltrotor UAV for load transportation. **Anais do XX Congresso Brasileiro de Automática**, p. 2097–2104, September 2014.

ALMEIDA, M. M.; RAFFO, G. V. Nonlinear control of a tiltrotor UAV for load transportation. **11th IFAC Symposium on Robot Control**, p. 234–239, 2015.

ASHOK, K. **Programming And Data Structures (For Anna University)**. [S.l.]: Pearson Education India, 2003.

CAMACHO, E.; ALBA, C. **Model Predictive Control**. Second edition. London: Springer, 2013. 405 p.

CASTILLO, P.; LOZANO, R.; DZUL, A. Stabilization of a mini-rotorcraft with four rotors. **Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on**, vol. 3, p. 2693–2698, 2004.

CRUZ, P.; OISHI, M.; FIERRO, R. Lift of a cable-suspended load by a quadrotor: A hybrid system approach. **American Control Conference (ACC), 2015**, p. 1887–1892, July 2015.

DONADEL, R. **Modeling and Control of a Tiltrotor Unmanned Aerial Vehicule for Path Tracking**. Thesis (Master Degree Thesis) — Federal University of Santa Catarina, Florianópolis, 2015.

DONADEL, R.; RAFFO, G.; BECKER, L. Modeling and control of a tiltrotor UAV for path tracking. **19th IFAC World Congress**, p. 3839–3844, 2014.

FAUST, A. et al. Automated aerial suspended cargo delivery through reinforcement learning. **Artificial Intelligence**, p. 1–18, 2014.

GINGICHASHVILI, S. **American Dynamics AD-150 UAV**. 2009. Available from Internet:

<<http://thefutureofthings.com/6403-american-dynamics-ad-150-uav/>>.

HOWARD, C. **U.S. to sell Bell Boeing V-22 Osprey tiltrotor aircraft, engines, and advanced electronics to Israel.** 2014. Available from Internet: <http://www.militaryaerospace.com/articles/2014/01/israel-v22.html>.

JAIN, R. P. K. **Transportation of Cable Suspended Load using Unmanned Aerial Vehicles: A Real-time Model Predictive Control approach.** Thesis (Master Degree of Science and Control) — Delft University of Technology, August 2015.

JANSEN, F.; RAMIREZ-SERRANO, A. Agile unmanned vehicle navigation in highly confined environments. **Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on**, p. 2381–2386, October 2011.

JOHNSON, M.; GRIMBLE, M. Recent trends in linear optimal quadratic multivariable control system design. **IEE Proceedings D - Control Theory and Applications**, vol. 134, n. 1, p. 53–71, January 1987.

KEANE, J. F.; CARR, S. S. A brief history of early unmanned aircraft. **The Johns Hopkins APL Technical Digest**, vol. 32, n. 3, p. 558–571, 2013.

MACHADO, P. H.; RAFFO, G. V. A quad-rotor platform for load transportation using visual feedback. **XII Simpósio Brasileiro de Automação Inteligente**, p. 1895–1900, 2015.

MACIEJOWSKI, J. **Predictive Control: With Constraints.** [S.l.]: Prentice Hall, 2002. (Pearson Education).

MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. **IEEE Robotics Automation Magazine**, vol. 19, n. 3, p. 20–32, Sept 2012.

OPLI. **Panther UAS was successfully demonstrated: Israel Aerospace Industries Presents CStrike-Integrated Tactical See & Strike System.** 2012. Available from Internet: <http://www.opli.net/magazine/eo/2012/news/iaic2s>.

OSBORN, K. **Marines Plan New High-Tech Osprey Variant.** 2015. Available from Internet: <http://www.scout.com/military/warrior/story/1585088-marines-plan-new-high-tech->

PALUNKO, I.; CRUZ, P.; FIERRO, R. Agile load transportation : Safe and efficient load manipulation with aerial robots. **Robotics Automation Magazine, IEEE**, vol. 19, n. 3, p. 69–79, Sept 2012.

PAPACHRISTOS, C. et al. Model predictive attitude control of an unmanned tilt-rotor aircraft. **Industrial Electronics (ISIE), 2011 IEEE International Symposium on**, p. 922–927, June 2011.

PIZETTA, I. B.; BRANDAO, A. S.; SARCINELLI-FILHO, M. Modelling and control of a PVTOL quadrotor carrying a suspended load. **Unmanned Aircraft Systems (ICUAS), 2015 International Conference on**, p. 444–450, June 2015.

RAFFO, G. V. **Robust control strategies for a QuadRotor helicopter: an underactuated mechanical system**. Thesis (Ph.D.) — Universidad de Sevilla, 2011.

RAFFO, G. V.; ORTEGA, M. G.; RUBIO, F. R. An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter. **Automatica**, vol. 46, n. 1, p. 29 – 39, 2010.

RAFFO, G. V.; ORTEGA, M. G.; RUBIO, F. R. Nonlinear h-infinity controller for quad-rotor helicopter with input coupling. **Proceeding of the 18th IFAC World Congress**, p. 13834 – 13839, 2011.

RAFFO, G. V.; ORTEGA, M. G.; RUBIO, F. R. Path tracking of a UAV via an underactuated h-infinity control strategy. **European Journal of Control**, Elsevier, vol. 17, n. 2, p. 194–213, 2011.

RAWLINGS, J.; MAYNE, D. **Model Predictive Control: Theory and Design**. [S.l.]: Nob Hill Pub., 2009.

REGO, B. S. **Changes in previously develop tiltrotor dynamic equation**. Federal University of Minas Gerais, 2015. 1-8 p.

ROSSITER, J. A. **Model-Based Predictive Control: A Practical Approach**. First. Boca Raton, FL: CRC Press LLC, 2004.

SADR, S.; MOOSAVIAN, S. A. A.; ZARAFSHAN, P. Dynamics modeling and control of a quadrotor with swing load. **Journal of Robotics**, Hindawi Publishing Corporation, vol. 2014, 2014.

SANCHEZ, A. et al. Autonomous hovering of a noncyclic tiltrotor UAV: Modeling, control and implementation. **Proceeding of the 17th IFAC Wold Congress**, p. 803–808, 2008.

SHABANA, A. A. **Dynamics of Multibody Systems**. Third edition. Chicago:University of Illinois: Cambridge, 2005. 374 p.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. First edition. New York: John Wiley & Sons, 2005. 496 p.

SREENATH, K.; LEE, T.; KUMAR, V. Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. **Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on**, p. 2269–2274, Dec 2013. ISSN 0743-1546.

SREENATH, K.; MICHAEL, N.; KUMAR, V. Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system. **Robotics and Automation (ICRA), 2013 IEEE International Conference on**, p. 4888–4895, May 2013.

SUNG-KI, J. **Unmanned tilt-rotor vertical take-off aircraft unveiled**. 2011. Available from Internet:
<<http://www.koreatimes.co.kr/www/news/biz/2011/11/12399879.html>>.

TROFINO, A.; COUTINHO, D.; BARBOSA, K. A. **Sistemas Multivariáveis: Uma abordagem via LMIs (Versão preliminar)**. [S.l.: s.n.], 2003.

UAVGLOBAL. **Bell Eagle Eye**. 2008. Available from Internet:
<<http://www.uavglobal.com/bell-eagle-eye/>>.

APPENDIX A - Control Design Matrices

A.1 TILT-ROTOR UAV LINEAR MODEL WITH SUSPENDED LOAD

The linear model matrices obtained from the non-linear model (2.62) are presented in (A.1) and (A.2) for matrices $\mathbf{A}(t)$ and \mathbf{B} , respectively.

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0}_{10 \times 3} & \mathbf{0}_{10 \times 7} & \mathbf{I}_{10} \\ \mathbf{0}_{10 \times 3} & \mathbf{\Upsilon}_{10 \times 7} & \mathbf{A8} \end{bmatrix} \quad (\text{A.1})$$

where $\mathbf{\Upsilon} = [\mathbf{A}_1 \cdot \mathbf{a} \quad \mathbf{A}_2 \cdot \mathbf{a} \quad \mathbf{A}_3 \cdot \mathbf{a} \quad \mathbf{A}_4 \cdot \mathbf{a} \quad \mathbf{A}_5 \cdot \mathbf{a} \quad \mathbf{A}_6 \cdot \mathbf{a} \quad \mathbf{A}_7 \cdot \mathbf{a}]$, with $\mathbf{a} = [\ddot{x} \quad \ddot{y} \quad \ddot{z} \quad 1]$ and \ddot{x} , \ddot{y} , and \ddot{z} the accelerations of the desired trajectory.

$$\mathbf{A}_1 = \begin{bmatrix} -0.000035346 & -0.15886 & -0.000057162 & -0.00059514 \\ -8.0193 \cdot 10^{-6} & 0.0026811 & -0.99972 & -9.8073 \\ 0.00017011 & 0.76458 & 0.000084491 & 0.0008288 \\ 1.5394 \cdot 10^{-7} & 0.0007104 & -0.00047742 & -0.0046833 \\ -0.00053207 & -2.3914 & -0.00090363 & -0.0094132 \\ 4.7409 \cdot 10^{-6} & 0.021005 & 0.0078218 & 0.076732 \\ 0.00060363 & 2.713 & 0.0015813 & 0.016059 \\ 0.00060351 & 2.7125 & 0.00021781 & 0.0026832 \\ -1.444 \cdot 10^{-7} & -0.00064612 & -0.000074445 & -0.00073044 \\ 0.00053007 & 2.3824 & 0.0007921 & 0.0082504 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 9.3366 \cdot 10^{-6} & 0.003236 & 0.99982 & 9.8082 \\ 6.0917 \cdot 10^{-7} & 0.0027382 & -7.3698 \cdot 10^{-6} & -0.000072297 \\ -1.4418 \cdot 10^{-7} & -0.0012442 & 0.015391 & 0.15099 \\ -0.000080008 & -0.3596 & 0.00013326 & 0.0013073 \\ 0.000012925 & 0.058061 & 0.00085312 & 0.0083691 \\ -0.001988 & -8.9347 & -0.0088264 & -0.086587 \\ -0.00018597 & -0.83579 & -0.0016184 & -0.015877 \\ 0.00016055 & 0.72162 & -0.000079855 & -0.00078338 \\ 4.9897 \cdot 10^{-6} & 0.022431 & -0.00011855 & -0.001163 \\ -0.000010298 & -0.046258 & -0.00074197 & -0.0072788 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} -0.00017018 & -0.76487 & 0 & 0 \\ 2.6768 \cdot 10^{-7} & 0.0012031 & 1.4844 \cdot 10^{-10} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} 0.00024004 & 1.038 & 1.055 & 10.35 \\ -3.1156 \cdot 10^{-7} & -0.0013527 & -0.0012291 & -0.012058 \\ -1.5446 \cdot 10^{-6} & -0.0066776 & -0.0068284 & -0.066987 \\ 3.0928 \cdot 10^{-6} & 0.013453 & 0.011557 & 0.11338 \\ 0.0019397 & 8.3894 & 8.4866 & 83.254 \\ 0.0010627 & 4.5999 & 4.5561 & 44.695 \\ -0.0018398 & -7.957 & -8.0577 & -79.046 \\ -0.002025 & -8.7587 & -8.8517 & -86.836 \\ -2.4688 \cdot 10^{-6} & -0.010744 & -0.0090947 & -0.089219 \\ -0.0014604 & -6.3168 & -6.3802 & -62.589 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} -0.00022313 & -1.0439 & 1.0586 & 10.385 \\ -3.5844 \cdot 10^{-7} & -0.0016709 & 0.0015454 & 0.015161 \\ 1.4276 \cdot 10^{-6} & 0.0066806 & -0.0068162 & -0.066867 \\ 3.9699 \cdot 10^{-6} & 0.018485 & -0.016576 & -0.16261 \\ -0.0018055 & -8.4453 & 8.5238 & 83.618 \\ 0.00098988 & 4.6262 & -4.5732 & -44.863 \\ 0.0018849 & 8.8161 & -8.8896 & -87.207 \\ 0.0017123 & 8.0095 & -8.0923 & -79.386 \\ -3.256 \cdot 10^{-6} & -0.015157 & 0.013499 & 0.13242 \\ 0.00136 & 6.361 & -6.4101 & -62.883 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} 9.0923 \cdot 10^{-9} & 0.00004041 & 0.000011768 & 0.00011545 \\ 4.0557 \cdot 10^{-7} & -0.00015676 & 0.051106 & 0.50135 \\ 7.1498 \cdot 10^{-6} & 0.032136 & -0.000027396 & -0.00026876 \\ -8.4459 \cdot 10^{-7} & 0.00063884 & -0.11449 & -1.1232 \\ 1.6663 \cdot 10^{-7} & 0.00074177 & 0.00018501 & 0.0018149 \\ -1.64 \cdot 10^{-8} & 0.000012809 & -0.0022337 & -0.021912 \\ -1.6743 \cdot 10^{-7} & -0.00073784 & -0.0003788 & -0.003716 \\ -1.6457 \cdot 10^{-7} & -0.00074006 & 0.000010562 & 0.00010361 \\ -0.000016025 & 0.0048518 & -1.9847 & -19.47 \\ 0.000036676 & 0.16485 & -0.0001615 & -0.0015843 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} -6.768 \cdot 10^{-7} & 0.00021883 & -0.084181 & -0.82581 \\ -2.3248 \cdot 10^{-9} & -9.9961 \cdot 10^{-6} & -0.000011694 & -0.00011472 \\ -1.1917 \cdot 10^{-8} & -0.000050919 & -0.000068182 & -0.00066886 \\ 3.4363 \cdot 10^{-8} & 0.00014721 & 0.00018671 & 0.0018316 \\ -5.168 \cdot 10^{-6} & 0.0016571 & -0.64244 & -6.3023 \\ 1.5613 \cdot 10^{-6} & 0.0069779 & 0.0010164 & 0.009971 \\ 5.2843 \cdot 10^{-6} & -0.0010432 & 0.64009 & 6.2793 \\ 5.0121 \cdot 10^{-6} & -0.0022595 & 0.6399 & 6.2774 \\ -0.000073832 & -0.33183 & -0.00016658 & -0.0016341 \\ -0.000012373 & 0.0033383 & -1.5218 & -14.929 \end{bmatrix}$$

$$A_8 = \begin{bmatrix} 2.2211 & 2.2205 & -9.215 \cdot 10^{-9} & 0.000026997 \\ 0.0013136 & -0.000039515 & -7.1926 \cdot 10^{-7} & 9.1711 \cdot 10^{-9} \\ 0.0037593 & 0.0037633 & 2.1812 \cdot 10^{-8} & 5.4179 \cdot 10^{-8} \\ -0.020078 & -0.00013921 & 0.000091155 & -1.4549 \cdot 10^{-7} \\ 35.444 & 35.437 & -1.4629 \cdot 10^{-7} & 0.00051049 \\ -0.50352 & 0.4351 & 1.6219 \cdot 10^{-6} & -4.5522 \cdot 10^{-7} \\ -58.893 & -35.264 & 2.8695 \cdot 10^{-7} & -0.00050859 \\ -35.264 & -49.228 & 4.2406 \cdot 10^{-9} & -0.0005085 \\ 0.017495 & 0.00025855 & -0.00031661 & 1.2777 \cdot 10^{-7} \\ -31.009 & -31.003 & 1.2777 \cdot 10^{-7} & -0.00068343 \end{bmatrix}$$

$$B = \begin{bmatrix}
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 \\
 5.8514 \cdot 10^{-7} & -5.8627 \cdot 10^{-7} & -0.0072859 & -0.0072837 \\
 0.000372 & -0.00037272 & -4.309 \cdot 10^{-6} & 1.2962 \cdot 10^{-7} \\
 0.00047596 & 0.00047847 & -0.000012332 & -0.000012345 \\
 -0.0052431 & 0.0052532 & 0.000065861 & 4.5665 \cdot 10^{-7} \\
 8.8973 \cdot 10^{-6} & -8.9145 \cdot 10^{-6} & -0.11627 & -0.11624 \\
 -0.00016133 & 0.00016164 & 0.0016517 & -0.0014272 \\
 -0.000022911 & 0.000022955 & 0.19319 & 0.11568 \\
 5.2109 \cdot 10^{-6} & -5.221 \cdot 10^{-6} & 0.11568 & 0.16148 \\
 0.0045003 & -0.004509 & -0.00005739 & -8.4811 \cdot 10^{-7} \\
 -7.7231 \cdot 10^{-6} & 7.738 \cdot 10^{-6} & 0.10172 & 0.1017
 \end{bmatrix}$$

(A.2)

A.2 STATE FEEDBACK GAIN MATRIX OF LQR CONTROLLER



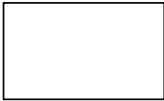

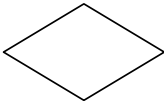
The gain matrix used in the LQR controller is given :

$$K e^T = \begin{bmatrix}
 13360.0 & -18121.0 & -28996.0 & -38357.0 \\
 -58426.0 & 59070.0 & -18.49 & -1553.4 \\
 -39041.0 & -39151.0 & 1127.5 & -826.02 \\
 56955.0 & -57609.0 & -170.29 & 322.64 \\
 26391.0 & -35717.0 & -48048.0 & -70269.0 \\
 -6124.8 & 2538.7 & -9312.2 & 14435.0 \\
 14984.0 & -14015.0 & -34278.0 & -23673.0 \\
 26375.0 & -28926.0 & -25320.0 & -65864.0 \\
 15346.0 & -15478.0 & -128.95 & 1021.5 \\
 -3669.0 & 3918.1 & 2097.8 & 6059.3 \\
 8187.3 & -11121.0 & -16366.0 & -22729.0 \\
 -26429.0 & 26728.0 & 32.427 & -469.43 \\
 -12569.0 & -12600.0 & 171.66 & -201.76 \\
 3572.2 & -3613.5 & -6.7018 & -214.39 \\
 5085.3 & -5405.6 & -7062.8 & -11001.0 \\
 -1174.9 & 1083.6 & -1628.3 & 3479.4 \\
 1603.7 & -1434.8 & -2215.9 & -2704.9 \\
 2693.4 & -2475.0 & -2924.6 & -5383.2 \\
 1953.8 & -1983.1 & -54.258 & 0.77362 \\
 1128.0 & -1450.1 & -2189.4 & -3220.8 \\
 9201.2 & -12512.0 & -21452.0 & -27213.0 \\
 -51097.0 & 51643.0 & -95.524 & -1657.5 \\
 -42590.0 & -42619.0 & 1505.7 & -891.15 \\
 -10854.0 & 3895.0 & -20999.0 & 26364.0
 \end{bmatrix} \quad (\text{A.3})$$

APPENDIX B – Symbols Used in the Flowcharts

B.1 SYMBOLS USED IN THE FLOWCHARTS

The descriptions of each symbol used in the flowcharts are the following:

Symbols	Name	Function
	Start/End	Indicates the beginning or end point of a program.
	Data	Input and output of any type of data
	Process	Indicates any processing function
	Subroutine	Indicates a predefined process, such as a subroutine or a module.
	Decision	Indicates a decision point and the action to take depending on the answer to the comparison given.