mαcrϕ@ufmg

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Research group MACRO - Mechatronics, Control and Robotics

TUBE-BASED MPC WITH ECONOMICAL CRITERIA FOR LOAD TRANSPORTATION TASKS USING TILT-ROTOR UAVS

Marcelo Alves dos Santos

Belo Horizonte, Brazil
2018

Marcelo Alves dos Santos

TUBE-BASED MPC WITH ECONOMICAL CRITERIA FOR LOAD TRANSPORTATION TASKS USING TILT-ROTOR UAVS

Thesis submitted to the Graduate Program in Electrical Engineering of Escola de Engenharia at the Universidade Federal de Minas Gerais, in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Advisor: Guilherme Vianna Raffo
Coadvisor: Antonio Ferramosca

Belo Horizonte, Brazil
2018

Santos, Marcelo Alves dos.
S237t Tube-based MPC with economical criteria for load transportation tasks using tilt-rotor UAVs [recurso eletrônico] / Marcelo Alves dos Santos. - 2018.
1 recurso online (149 f. : il., color.) : pdf.

Orientador: Guilherme Vianna Raffo.
Coorientador: Antonio Ferramosca.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 129-149.
Bibliografia: f. 121-128.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Controle preditivo - Teses. 3. Transporte de cargas - Teses. I. Raffo, Guilherme Vianna. II. Ferramosca, Antonio. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pelo bibliotecário Reginaldo César Vital dos Santos CRB/6 2165
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
Programa de Pós-Graduação em Engenharia Elétrica

ATA DA 1037ª DEFESA DE DISSERTAÇÃO DE MESTRADO
DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ATA DE DEFESA DE DISSERTAÇÃO DE MESTRADO do aluno **Marcelo Alves dos Santos** - registro de matrícula de número 2015716160. Às 14:00 horas do dia 19 do mês de fevereiro de 2018, reuniu-se na Escola de Engenharia da UFMG a Comissão Examinadora da DISSERTAÇÃO DE MESTRADO para julgar, em exame final, o trabalho intitulado **"Tube-based MPC with Economical Criteria for Load Transportation Tasks Using Tilt-rotor UAVs"** da Área de Concentração em Sinais e Sistemas, Linha de Pesquisa Controle, Automação e Robótica. O Prof. Guilherme Vianna Raffo, orientador do aluno, abriu a sessão apresentando os membros da Comissão e, dando continuidade aos trabalhos, informou aos presentes que, de acordo com o Regulamento do Programa no seu Art. 8.16, será considerado APROVADO na defesa da Dissertação de Mestrado o candidato que obtiver a aprovação unânime dos membros da Comissão Examinadora. Em seguida deu início à apresentação do trabalho pelo Candidato. Ao final da apresentação seguiu-se a arguição do candidato pelos examinadores. Logo após o término da arguição a Comissão Examinadora se reuniu, sem a presença do Candidato e do público, e elegeu o Prof Guilherme Vianna Raffo para presidir a fase de avaliação do trabalho, constituída de deliberação individual de APROVAÇÃO ou de REPROVAÇÃO e expedição do resultado final. As deliberações individuais de cada membro da Comissão Examinadora foram as seguintes:

Membro da Comissão Examinadora	Instituição de Origem	Deliberação	Assinatura
Prof. Dr. Guilherme Vianna Raffo - Orientador	DELT (UFMG)	APROVADO	
Prof. Dr. Antonio Ferramosca - Coorientador	CONICET (UTN - Argentina)	APROVADO	
Prof. Dr. Julio Elias Normey Rico	DAS (UFSC)	APROVADO	
Prof. Dr. Luciano Cunha de Araújo Pimenta	DELT (UFMG)	APROVADO	
Prof. Dr. Ricardo Hiroshi Caldeira Takahashi	DMAT (UFMG)	Aprovado	

Tendo como base as deliberações dos membros da Comissão Examinadora a Dissertação de Mestrado foi APROVADO. O resultado final de APROVAÇÃO foi comunicado publicamente ao Candidato pelo Presidente da Comissão, ressaltando que a obtenção do Grau de Mestre em ENGENHARIA ELÉTRICA fica condicionada à entrega do TEXTO FINAL da Dissertação de Mestrado.O Candidato terá um prazo máximo de 30 (trinta) dias, a partir desta data, para fazer as CORREÇÕES DE FORMA e entregar o texto final da Dissertação de Mestrado na secretaria do PPGEE/UFMG. As correções de forma exigidas pelos membros da Comissão Examinadora deverão ser registradas em um exemplar do texto da Dissertação de Mestrado, cuja verificação ficará sob a responsabilidade do Presidente da Banca Examinadora. Nada mais havendo a tratar o Presidente encerrou a reunião e lavrou a presente ATA, que será assinada pelo Presidente da Comissão Examinadora. Belo Horizonte, 19 de fevereiro de 2018.

ASSINATURA DO PRESIDENTE DA COMISSÃO EXAMINADORA

To my beloved wife and parents

Acknowledgments

Above all, I thank God for showing me His common grace at every moment of this academic journey. I am grateful for the personal and professional growth that I have experienced during this time and for given me the wisdom and strength necessary to enjoy them. Without His sustenance I would not have gotten that far.

I thank my beloved wife for her support and comprehension during the period of this master's degree. You have always supported me to pursue my dreams despite the difficulties that this could create for our lives in the short term. For this, I cannot say thank you enough. When I was hesitant, exhausted, or wanting to give up, you kept me on my toes and helped me to just keep going. I hope to always be at your side to enjoy the future we are building together. Thanks, darling!

I also thank my dear parents for encouraging me to pursue my studies. If it was not for the education you gave me, I would not be here today. Thanks for always be my safe heaven in the most difficult times and for given me valuables advices and encouragement. My achievements are also yours.

Further, I thank my parents in-laws for their support and for making my studies possible. I also thank my siblings for, despite the distance, being by my side. I will always keep you both on my heart. Moreover, I would like to thank all the colleagues from the MACRO team for making this period more productive and fun.

To my advisor, Prof. Guilherme Vianna Raffo, and coadvisor, Prof. Antonio Ferramosca, I express my sincere gratitude. Without you this work would not be possible. Guilherme, in the last three years you have shown me the beauty of doing research in control theory. Thanks for encouraging me to pursue this path and for supporting me personally and technically at every hard time. I truly hope to become a professional as capable and human as you. Antonio, thank you because your vast knowledge and insights in predictive control was essential to the conclusion of this work. I am looking forward to continuing to work with both of you.

Finally, in the person of Prof. Rodney Rezende Saldanha, I thank all the professors and staff of the Graduate Program in Electrical Engineering.

Gratiam pro gratia

Resumo

Essa dissertação de mestrado aborda o controle de um Veículo Aéreo Não Tripulado (VANT) na configuração Tilt-rotor utilizado para o transporte de carga. Propõe-se resolver esse problema através de leis de controle ótimo com horizonte deslizante, especificamente, métodos de controle preditivo baseado em modelo (MPC). O objetivo principal é obter controladores capazes de realizar o seguimento de uma trajetória desejada mantendo o sistema estável mesmo na presença de incertezas e perturbações.

Inicialmente é proposta uma estratégia de controle utilizando um MPC na formulação incremental com modelo de predição linear e invariante no tempo. Posteriormente, essa formulação é estendida para considerar modelos de predição lineares e variantes no tempo, possibilitando lidar de maneira simples com o problema de variação de parâmetros do sistema, como massa e comprimento do cabo conectando a carga suspensa ao VANT, o que permite realizar manobras de decolagem e aterrissagem com a carga ainda em contato com o solo. Além disso, essa estratégia trata o problema de seguimento de trajetória considerando diferentes pontos de equilíbrio do sistema linearizado, sendo também possível realizar seguimento de trajetória do ângulo de guinada do VANT sem instabilizar o sistema. Ademais, é desenvolvido nesse trabalho um controlador MPC com critério econômico, sendo este critério integrado ao funcional de custo do controlador através de funções potenciais de atração e repulsão. Isso possibilita ao sistema de controle lidar com o problema de desvio de obstáculos escolhendo uma trajetória econômica que evita colisão e reduz o erro entre a trajetória inicial e a executada pelo VANT. Ainda, é abordada uma classe de controladores preditivos robustos, o controlador preditivo baseado em tubos. Para garantir robustez ao sistema, esse controlador utiliza em sua formulação algumas técnicas bem conhecidas em teoria de conjuntos para definir no espaço de estados conjuntos alcançáveis e regiões de estabilidade que limitam a diferença entre a trajetória predita pelo modelo nominal e pelo modelo que leva em consideração perturbações e incertezas limitadas. Esse controlador, devido ao custo computacional das ferramentas de teoria de conjuntos, será utilizado em cascata com um controlador não linear baseado na linearização por realimentação de saída. Finalmente, as características de robustez do controlador baseado em tubos são combinadas ao controlador MPC econômico para gerar uma nova estratégia de controle robusto com critérios econômicos.

Palavras-chave: Controle Preditivo, Controle Preditivo Econômico, Controle Preditivo Robusto, VANT Tilt-rotor, Transporte de Carga.

Abstract

This Master Thesis addresses the control of an Unmanned Aerial Vehicle (UAV) in the Tilt-rotor configuration for load transportation tasks. This work proposes to solve the problem using an optimal control strategy with receding horizon, mainly, Model-based Predictive Controllers (MPC). The main objective is to obtain controllers able to perform path tracking stably even in the presence of uncertainties and disturbances.

Initially, a control strategy is proposed using an MPC based on the incremental framework with a linear time-invariant prediction model. After, this formulation is extended to consider a linear time-variant prediction model, which makes it possible to deal, in a fashion way, with time varying systems' parameters, such as mass and length of the cable connecting the suspended load to the UAV, then allowing performing take-off and landing maneuvers with the load in contact with the ground. Furthermore, this strategy deals with the path tracking control problem considering different equilibrium points of the linearized system. This also allows tracking of the UAV's yaw angle without destabilize the whole system. Moreover, an MPC controller with economic criteria is designed, in which this criteria is added to the controller's cost function through attraction and repulsion potential functions. This makes the control system able to deal with the obstacle avoidance problem by choosing an economical trajectory that avoids collision and reduces the error between the initial trajectory and the one performed by the UAV. Furthermore, a class of robust predictive controllers, the tube-based ones, is addressed. To ensure robustness to the system, this controller takes hand of some well-known techniques of set theory to design reachable sets and stability regions in the state-space in order to limit the mismatch between the trajectory predicted using the nominal model and the model that considers limited disturbances and uncertainties. Due to the computational cost of the set theory tools, the proposed tube-based controller is cascaded with a nonlinear controller based on input-output feedback linearization. Finally, the robustness features of the tube-based controller are combined to the economic MPC controller to generate a new strategy of robust control with economic criteria.

keywords: Predictive Control, Economic Predictive Control, Robust Predictive Control, Tilt-Rotor UAV, Load Transportation.

List of Figures

List of Tables

Acronyms

BIBO	Bounded-Input, Bounded-Output
CAD	Computer Aided Design
EMPC	Economic Model Predictive Controller
FIFO	First In, First Out
IADU	Integrated Absolute Derivative of Control Signal
IOFL	Input-Output Feedback Linearization
LMI	Linear Matrix Inequality
LP	Linear Programing
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTI-MPC	Linear Time-Invariant Model Predictive Controller
LTV	Linear Time-Variant
LTV-MPC	Linear Time-Variant Model Predictive Controller
MIMO	Multiple-Input, Multiple-Output
MPC	Model Predictive Controller
MSE	Mean Squared Error
PID	Proportional-Integral-Derivative
REMPC	Robust Tube-Based Economic Controller
RMPC	Robust Tube-Based Controller

RTO	Real Time Optimizer
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing
ZOH	Zero-Order Hold

Notation

General notation

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

Symbols and operators

$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$\mathbf{0}$	Zero matrix with appropriate dimension
$\mathbb{I}$	Identity matrix with appropriate dimension
$\mathbf{0}_{\mathbf{n}\times\mathbf{m}}$	Zero matrix with n lines and m columns
$\mathbb{I}_{n\times n}$	Identity matrix with n lines and n columns
$\dot{\boldsymbol{A}}$	Time derivative of $\boldsymbol{A}$
$\boldsymbol{A}^T$	Transpose of $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of $\boldsymbol{A}$
$\boldsymbol{A}^{+}$	Pseudo-inverse of $\boldsymbol{A}$
$\mathrm{Tr}(\boldsymbol{A})$	Trace of $\boldsymbol{A}$
$\oplus$	Minkowski sum operator
$\ominus$	Pontryagin difference operator
$Q(\boldsymbol{\Omega})$	One-step set operator
$\tilde{Q}(\boldsymbol{\Omega})$	Robust one-step set operator

Model Predictive Control (Chapters 3 and 4)

$\boldsymbol{x}$	State vector
$\Delta\boldsymbol{x}$	State error vector
$\Delta\bar{\boldsymbol{x}}$	Augmented state error vector for the incremental model
$\Delta\tilde{\boldsymbol{x}}$	State error vector between the aircraft and the collision-free trajectory
$\Delta\boldsymbol{x}^{nom}$	State error vector for the nominal model
$\boldsymbol{u}$	Input vector
$\Delta\boldsymbol{u}$	Input vector for linear systems
$\delta\boldsymbol{u}$	Control increment vector
$\boldsymbol{g}$	Control policy vector able to control the nominal model
$\boldsymbol{e}$	Mismatch error vector
$\boldsymbol{w}$	Addictive uncertainty vector
$\boldsymbol{\xi}^{obs}$	Vector representing an obstacle
$\boldsymbol{\xi}^{tr}$	Desired spatial trajectory vector
$\boldsymbol{\xi}^{goal}$	Collision-free spatial trajectory vector
$\mathbb{X}$	Set of admissible states
$\mathbb{U}$	Set of admissible inputs
$\mathbb{W}$	Set of bounded uncertainties
$\mathbb{U}$	Set of admissible state errors
$\mathbb{O}$	Set representing the obstacles
$\mathscr{R}$	Reachable set
$\boldsymbol{A}, \boldsymbol{B}$	Linear model Jacobian matrices
$\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}$	Linear incremental model Jacobian matrices
$\boldsymbol{\mathcal{P}}_i, \boldsymbol{\mathcal{H}}_i$	Incremental prediction model matrices
$\boldsymbol{\mathcal{P}}_{ni}, \boldsymbol{\mathcal{H}}_{ni}$	Non-incremental prediction model matrices

N_p	Prediction horizon
N_c	Control horizon
$\mathcal{J}$	Cost function
U_{att}	Attraction potential function
U_{att}	Repulsive potential function
$\boldsymbol{K}(\boldsymbol{\zeta})$	Adaptive feedback gain
$\boldsymbol{\mathcal{Q}}$	States' error weighting matrix
$\boldsymbol{\mathcal{R}}$	Control effort weighting matrix
$\boldsymbol{P}$	Lyapunov matrix
$\boldsymbol{\Omega}$	Maximal invariant
$\boldsymbol{\zeta}$	Time-varying parameters vector
Δ	Polytope containing the uncertain parameters
$V(\cdot)$	Lyapunov function
$(\cdot)^{max}$	Maximum value allowed
$(\cdot)^{min}$	Minimum value allowed
$(\cdot)^{tr}$	Trajectory values
$(\cdot)^{eq}$	Equilibrium values
$\ell(\cdot)$	Standard MPC stage cost
$\ell_e(\cdot)$	Economic stage cost
T_s	Sampling period

Modeling (Appendix A)

$(\cdot)_{\mathcal{B}}$	Variable considering the model from the UAV's perspective
$(\cdot)_{\mathcal{L}}$	Variable considering the model from the load's perspective
s_i	$sin(i)$
c_i	$cos(i)$

$\boldsymbol{q}$	Generalized coordinates vector
$\mathscr{L}$	Lagrangian
$\mathcal{K}$	Kinetic energy
$\mathcal{U}$	Potential energy
$\boldsymbol{\vartheta}$	Generalized force and torque vector
$\boldsymbol{M}$	Inertia matrix
$\boldsymbol{C}$	Coriolis and centripetal forces matrix
$\boldsymbol{G}$	Gravitational force vector
$\boldsymbol{p}_i^{\mathcal{F}}$	Point rigidly attached to a body i expressed in $\mathcal{F}$
$\boldsymbol{\omega}_{\mathcal{I}\mathcal{F}}^{\mathcal{I}}$	Angular velocity between the frames $\mathcal{I}$ and $\mathcal{F}$ expressed in $\mathcal{I}$
$\mathcal{I}$	Inertial reference frame
$\mathcal{B}$	Reference frame rigidly attached to the aircraft's geometric center
$\mathcal{C}_1$	Reference frame rigidly attached to the main body's center of mass
$\mathcal{C}_2$	Reference frame rigidly attached to the right thruster group's center of mass
$\mathcal{C}_3$	Reference frame rigidly attached to the right thruster group's center of mass
$\mathcal{L}$, $\mathcal{C}_4$	Reference frame rigidly attached to the load's center of mass
$\boldsymbol{\xi} = [x\ y\ z]^T$	Position with respect to $\mathcal{I}$
$\boldsymbol{\eta} = [\phi\ \theta\ \psi]^T$	Euler angles parametrizing the orientation with respect to $\mathcal{I}$
α_R	Tilting angle of the right thruster group
α_L	Tilting angle of the left thruster group
γ_1, γ_2	Angles parametrizing the load or aircraft with respect to each other
f_R	Magnitude of the thrust generated by the right propeller
f_L	Magnitude of the thrust generated by the left propeller
τ_R	Magnitude of the torque generated by the right servomotor
τ_L	Magnitude of the torque generated by the left servomotor

$\boldsymbol{R}_{\mathcal{F}}^{\mathcal{I}}$	Rotation matrix between the frames $\mathcal{F}$ and $\mathcal{I}$
$\boldsymbol{d}_{\mathcal{F}}^{\mathcal{I}}$	Displacement vector from $\mathcal{F}$ to $\mathcal{I}$, expressed in $\mathcal{I}$
$\boldsymbol{S}(\cdot)$	Skew-symmetric matrix operator

Nonlinear Control (Appendix B)

$\boldsymbol{f}(\boldsymbol{x})$	Nonlinear drift vector field
$\boldsymbol{g_u}(\boldsymbol{x})$	Nonlinear steering vector field
$\boldsymbol{g_d}$	Nonlinear external steering vector field
$\mathcal{L}_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x})$	Lie derivative of a scalar field $\boldsymbol{h}(\boldsymbol{x})$ in the direction of a vector field $\boldsymbol{f}(\boldsymbol{x})$
r_i	Relative degree of the i-th output
$\boldsymbol{v}$	Transformed input vector
$\boldsymbol{\Theta}(\cdot)$	Diffeomorphism function
$\tilde{\boldsymbol{e}}$	Transformed state-space vector
$\boldsymbol{\Upsilon}$	Intermediary control input
$\boldsymbol{\delta}$	Unmodeled dynamic vector
$\sigma(\cdot)$	Saturation function
$\boldsymbol{H}, \boldsymbol{D_u}, \boldsymbol{D_\pi}$	Weighting matrices of the mixed $\mathcal{H}_2/\mathcal{H}_\infty$
$\mathcal{Z}$	Z-transform
$\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)$	Discrete transfer function from $\boldsymbol{\pi}$ to $\tilde{\boldsymbol{z}}$
$\|\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)\|_2$	$\mathcal{H}_2$ norm of the matrix $\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)$
$\|\boldsymbol{\Psi}_{\boldsymbol{\pi z}}(\varsigma)\|_\infty$	$\mathcal{H}_\infty$ norm of the matrix $\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)$
ϱ	Upper-bound for the $\mathcal{H}_\infty$ norm
ε	Adjustment parameter of the first LMI region
ϖ	Adjustment parameter of the second LMI region

Contents

1 Introduction

1.1 Motivation

Unmanned Aerial Vehicles (UAVs) have experienced in the recent years a great popularization among academics, hobbyists, and, more recently, in the industry. This popularization is mainly due to the development, in the last decades, of technologies related with the design and assembly of these vehicles. For instance, the development of lighter and resistant low-cost materials, the cheapness of electronic components, and the increase processing capacity of embedded systems together with the reduction of the physical space occupied by them. This scenario allows virtually anyone with some basic engineering knowledge to build a UAV. In addition, the growing interest of the academy in these vehicles has pushed forward the state of the art knowledge, which made some problems, that once precludes the development of UAVs, to become well-known, as for instance, problems like modeling, dynamic control, state estimation, visual and navigation systems, among others.

In the earlier days, UAVs were mainly used for military purposes since these vehicles were promising platforms to perform military tasks, such as: search and rescue, surveillance, transportation, and combat (Ryan & Hedrick, 2005; Beard et al., 2006). However, with the aforementioned popularization of UAVs, new applications for these vehicles in the civil sphere arose. Nowadays, UAVs are being used in a wide range of applications, to cite a few: precision agriculture, fire detection, cargo transportation and delivery, cave exploration, cinematographic filming, 3-D mapping, and pipeline inspection (Tokekar et al., 2013; Merino et al., 2005; Palunko et al., 2012).

UAVs are often found in two main configurations: rotary-wing and fixed-wing. The first configuration has the advantage of performing Vertical Take-Off and Landing (VTOL), while the second one is able to obtain improved flights with greater forward operational range and endurance. Helicopters and quadrotors (Figures 1.1a and 1.1b) are examples of rotary-wing UAVs, while airplanes (Figure 1.1c) are examples of fixed-wing UAVs. Some hybrid aircrafts have drawn substantial attention due to their ability to combine the vertical lift capacity of helicopters with the range, endurance, and speed of fixed-wings. Amongst them, the Tilt-rotor configuration is one of the most popular, being provided with fixed-wings and rotary-wings, and capable of switching between helicopter and airplane flight-modes only by tilting its thrusters group. This kind of aircraft is particularly interesting to perform tasks that require fast deployment, access to restricted areas, and high maneuverability in slow velocities. Inspired by the capabilities of those aircrafts, recent researches are looking into the design of small-scale Tilt-rotor UAVs (Amiri et al., 2011; Park et al., 2013; Cardoso et al., 2016).

(a) Helicopter UAV.

(b) Quadrotor UAV.

(c) Airplane UAV.

Figure 1.1: Most common examples of unmanned aircrafts.

Full-scale hybrid aircrafts can be found in the aerospace industry, such as: the military Bell Boing V-22 Osprey Tilt-rotor (Figure 1.2a), which has been widely used by the U.S. and Japan military forces due to its multi-mission capability; the Augusta Westland AW609 Tilt-rotor (Figure 1.2b) developed for the civil aviation market intending to reach the offshore oil and gas operators; the Nasa Gl-10 Greased Lightning Tilt-wing prototype (Figure 1.2c) to be used in applications like long endurance surveillance and mapping; and the military Bell V-280 Valor Tilt-rotor (Figure 1.2d) designed to achieve improved forward velocity (520 Km/h) and has recently made its first flight.

(a) The Bell Boeing V-22 Osprey Tilt-rotor aircraft.

(b) The Augusta Westland AW609 Tilt-rotor aircraft.

(c) The Nasa Gl-10 Greased Lightning Tilt-wing aircraft.

(d) The Bell V-280 Valor Tilt-rotor aircraft.

Figure 1.2: Examples of full-scales hybrid aircraft.

Although the hybrid characteristics of these kind of aircrafts offer advantages over fixed-wing and rotary-wing UAVs, they also come with control design challenges since these vehicles are complex underactuated mechanical systems with highly coupled dynamics. Their underactuated behavior is due to the existence of more degrees of freedom than control inputs. In practice, work with this kind of system means that it is not possible to regulate all degrees of freedom at the same time instant. Therefore, the control system design process must be made regarding the regulation of some desired states while guaranteeing the stability of the remaining ones. Moreover, the additional tiltable mechanisms allowing the transition between flight modes increase the system's mechanical complexity when compared with others UAVs. Yet, when these mechanisms are assumed to be rigid bodies, the aircraft becomes a multi-body system with coupling between the thrusters and the main body, which made the control process design still more challenging.

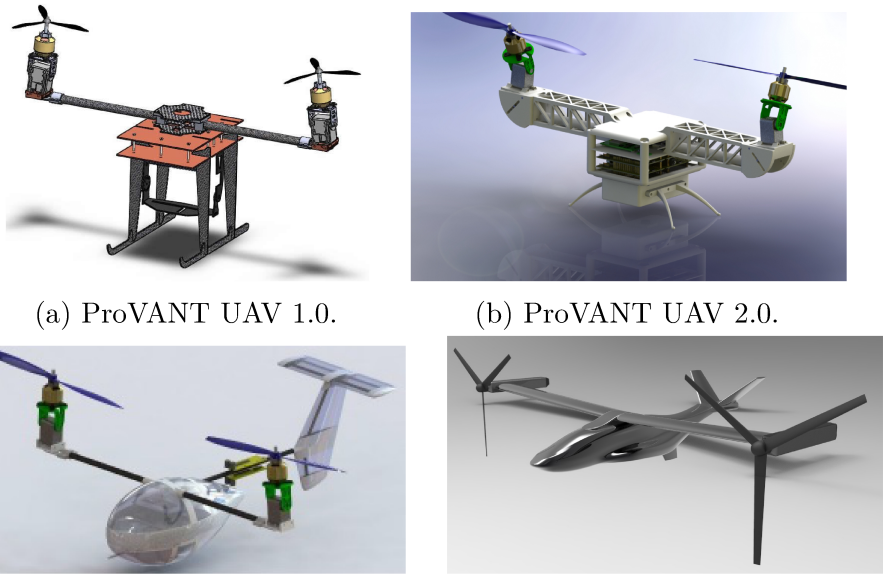
Despite being underactuated systems with coupled dynamics, these vehicles also have highly nonlinear dynamics, are affected by aerodynamic perturbations, and the models obtained for control design purpose are subject to modeling errors and parametric uncertainties. All these characteristics increase the complexity of the control design and make the use of traditional control techniques, e.g, pole placement, not suitable to be applied. In this context, advanced control techniques, for instance, adaptive, robust, model predictive, nonlinear, among others, become more appropriated in order to achieve good performance during autonomous flight.

Among several applications of UAVs, the load transportation in risky and inaccessible zones is quite important since it allows to deal with rapid deployment of supplies in search-

and-rescue missions (Bernard et al., 2011), vertical replenishment of seaborne vessels (Wang et al., 2014), and safe landmine detection (Bisgaard, 2008). However, this kind of task is also a challenging subject in terms of modeling and control. Since the payload is often connected to the UAV through a rope, the dynamic behavior of the system varies due to the load's swing, which can destabilize the whole system if it is not well attenuated. Moreover, the suspended load by a rope adds more unactuated degrees of freedom into the system, increasing its overall underactuation degree. Due to its hybrid capabilities, the Tilt-rotor UAV becomes a promising platform for such application, providing improved forward speed when compared with rotary-wing UAVs, which is a desired feature for missions requiring rapid deployment, and precise positioning of the load, which cannot be addressed by fixed-wing UAVs due to their inability to perform hover flight.

In this context the ProVANT project started as a joint effort between the Brazilians Federal University of Minas Gerais (UFMG) and Federal University of Santa Catarina (UFSC) aiming to design an open source Tilt-rotor UAV platform able to perform autonomous flight and address the load transportation problem. Figure 1.3a, shows the first Tilt-rotor UAV designed in the ProVANT project. The ProVANT 1.0 was assembled at UFSC, where in 2013 performed its first flight. The ProVANT 2.0, shown in Figure 1.3b, improved the last version by having its conceptual design made with Computer Aided Design (CAD) Software, providing a better knowledge of its physical parameters. This aircraft was assembled at UFMG using 3D printer and is currently under flight tests. The ProVANT 3.0, see Figure 1.3c, represented the evolution of the ProVANT project towards the achievement of a fully convertible Tilt-rotor aircraft. In this version, tail and fuselage surfaces were added seeking to allow improved forward flight when compared with previous versions. This aircraft is currently under assemblage process at UFMG. Later, after the University of Seville joined the project, the ProVANT 4.0 was proposed to achieve a full flight envelope, see Figure 1.3d. The inclusion of wings, besides the previous aerodynamic surfaces, will allow this aircraft to switch completely between the helicopter and airplane modes during the autonomous flight. This aircraft is under conceptual design processes and will be used to perform cargo delivery considering a 20 kilometers range.

This work is part of the ongoing effort of the ProVANT project team to investigate the problem of performing autonomous flight using a Tilt-rotor UAV applied to cargo transportation tasks. Specifically, this work aims to design model-based predictive controllers to solve the problem of transporting a load attached to the vehicle by a rope. This problem is addressed in this work from two different perspectives: (i) the control system is required to perform trajectory tracking of the suspended load while keeping the aircraft stabilized; (ii) the control system is required to perform trajectory tracking of the Tilt-rotor UAV while stabilizing the suspended load. Both approaches can be successfully used as a solution to the transport problem of slung loads using Tilt-rotor UAV. However, when precision positioning of the load is required, the first approach is shown to be more suitable.



(a) ProVANT UAV 1.0. (b) ProVANT UAV 2.0.

(c) ProVANT UAV 3.0. (d) ProVANT UAV 4.0.

Figure 1.3: Aircrafts designed in the context of the ProVANT project.

1.2 State of the art

The Tilt-rotor UAV modeling problem has been addressed in the ProVANT project through the Euler-Lagrange formulation providing an appropriate dynamic representation of the aircraft for control design purposes. Initially, the dynamic model was derived in Donadel et al. (2014b) for a Tilt-rotor UAV similar to the two first prototypes considered in the project (Figures 1.3a and 1.3b); and, after, this model was extended in Almeida et al. (2014) and Rego & Raffo (2016c) to consider a suspended load attached to the aircraft with the modeling procedure being made, respectively, from the UAV's perspective and from the load's perspective. Subsequently, in Cardoso et al. (2016) the dynamic model for a Tilt-rotor UAV with fuselage and tail surfaces was proposed. In the literature, it is possible to find other works considering the Tilt-rotor UAV modeling problem, for instance, in Sanchez et al. (2008) the dynamic equations of motion are derived through the Newton-Euler approach considering the vehicle as a single-body system despite the role that the coupling generated by the tiltable mechanisms have in this aircraft's dynamics. For the best knowledge of the author, only in Almeida et al. (2014) and in Rego & Raffo (2016c) it is possible to find the Tilt-rotor UAV with suspended load modeling problem being addressed and taking into account the coupling dynamics between the bodies composing the system.

Researches dealing with control design for Tilt-rotor UAVs are recent with few published works in the literature. Some relevant works have used the following techniques: model predictive control (Papachristos et al., 2013); inverse dynamics (Yanguo & Huanjin, 2009); backstepping (Kendoul et al., 2006; Amiri et al., 2013); and bounded smooth nonlinear function (Sanchez et al., 2008). The ProVANT group has contributed with the Tilt-rotor

UAV trajectory tracking control problem by considering in Donadel et al. (2014a) and Rego & Raffo (2016a), respectively, a continuous-time and discrete-time formulation of a linear quadratic regulator, and in Donadel et al. (2014b) an $\mathcal{H}_\infty$ and a mixed $\mathcal{H}_2/\mathcal{H}_\infty$ robust controllers.

When it comes to control design of Tilt-rotor UAVs for load transportation tasks, the literature is even more limited. However, if others UAVs structures are considered, many other works dealing with the load transportation control problem can be found, such as: Raffo & Almeida (2016); Palunko et al. (2012); Dai et al. (2014); Sreenath et al. (2013b) for quadrotors, and Bisgaard (2008) for helicopters. In the literature, control objectives for aerial load transportation include: trajectory tracking of the aircraft with reduced load's swing (Bisgaard et al., 2009; Palunko et al., 2012; Faust et al., 2013), obstacle avoidance (la Cour-Harbo & Bisgaard, 2009; Tang & Kumar, 2015), transportation by multiple aircrafts (Bernard & Kondak, 2009; Lee et al., 2013), and trajectory tracking of the suspended load (Palunko et al., 2013; Sreenath et al., 2013a; Pereira & Dimarogonas, 2016; Pereira et al., 2016). In Goodarzi (2016), the aircraft trajectory tracking problem was addressed considering a suspended load connected to a quadrotor via a length-varying rope. The problem was modeled through the Euler-Lagrange approach and a nonlinear geometric control was considered to stabilize the system. In Lee (2018) the suspended load control problem is also addressed through nonlinear geometric control. However, the problem of the load's trajectory tracking is solved considering multiples quadrotors carrying the load.

The aerial load transportation problem has been addressed in the ProVANT project considering a Tilt-rotor UAV. The aircraft trajectory tracking with load stabilization control problem is solved through the following techniques: robust mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control (Almeida et al., 2014); linear model predictive control (Santos & Raffo, 2016b; Andrade et al., 2016); adaptive LMI-based control (Santos & Raffo, 2016a); nonlinear IOFL control with a three-stage cascade structure (Almeida & Raffo, 2015) and with a two-stage cascade structure (Raffo & Almeida, 2017). Furthermore, the load trajectory tracking with the UAV stabilization is solved through the techniques: discrete linear quadratic regulator (Rego & Raffo, 2016b); discrete mixed $\mathcal{H}_2/\mathcal{H}_\infty$ robust control (Rego & Raffo, 2016c); and time-variant model predictive control (Santos et al., 2017b).

Model predictive controllers have been known for their ability to control constrained multiple-input multiple-output nonlinear systems. These controllers are able to deal in a simple way with multivariate underactuated systems ensuring input-to-output stability and also internal stability. Therefore, they have been considered in different formulations in order to deal with aerial robotics control problems, to cite a few: unmanned helicopters (Castillo et al., 2007; Kunz et al., 2013), quadrotors (Raffo et al., 2010; Alexis et al., 2014), unmanned airplanes (Kang & Hedrick, 2006), Tilt-rotors (Papachristos et al., 2013; Santos & Raffo, 2016b; Andrade et al., 2016; Santos et al., 2017b).

Since this kind of systems present different sources of uncertainty, robustness should be considered into the control design process. The problem of robustness in MPC has been addressed through different strategies. Among them, the most popular are the Min-Max open-loop MPC (Skokaert & Mayne, 1998; Lee & Yu, 1997) and the Tube-based MPC (Langson et al., 2004a; Mayne et al., 2009). On one hand, in the Min-Max open-loop MPC approach the optimization problem considers the worst case of the expected disturbances and uncertainties, which may lead to an excessively conservative control policy. One the order hand, the Tube-based MPC considers a dual control scheme, in which a standard open-loop MPC with tighter constraints is used to control the nominal system and a feedback control loop is used to increase the robustness of the system. The Tube-based MPC has shown to be a computationally efficient technique to achieve robustness in MPC without being so conservative as the Min-Max formulation. However, despite the Tube-based MPC be computationally efficient when compared to others strategies for MPC's robustification, it is still costly for applications that require small sampling periods, for instance, robotics. Therefore, many works in the literature use the Tube-based MPC technique only to control systems with a few degrees of freedom, such as: mobile robots (Kayacan et al., 2015; Ke et al., 2018; Sánchez, 2011) and PVTOL aircraft (Petkar et al., 2016).

Recently, an interesting formulation for model predictive controllers has been proposed seeking to include economic oriented criteria into the quadratic open-loop stage cost used in standard MPC formulations (Rawlings et al., 2012; Ellis et al., 2017). This formulation has been mainly considered for industrial-like process in order to gather the process optimization and control problems into an unified problem (Hinojosa et al., 2017; Amrit et al., 2013; Ellis et al., 2014). Similar ideas have been proposed earlier in the context of optimal control theory by the problems of minimal fuel or minimal time (Kirk, 2004). When it comes to the use of economic MPC in robotics, the literature is limited. However, the underlying ideas of this controller is presented in many works that use optimal controllers with the optimization problem extended to consider additional objectives, such as: safe navigation, fuel consumption, minimal time, among others. In Chung (2017), a predictive controller is formulated to include in its optimization problem the minimal fuel and time spent by an autonomous vehicle during the trajectory execution. In Alexis et al. (2015), the collision-free navigation problem is solved for a Tri-Tiltrotor UAV through a sampling-based receding horizon control considering random trees in the control space to remove colliding state evolutions. In Nascimento et al. (2014) and Mac et al. (2016), artificial potential fields are considered, respectively, to perform obstacle avoidance in a multi-robot system and for a quadrotor UAV. In Perez et al. (2012) a modified Rapidly-exploring Randomized Tree is joined with a LQR algorithm to perform obstacle avoidance.

1.3 Justification

Given the various applications of UAVs together with the technological challenges related with the design of these aircrafts, develop research in this field has been shown relevant from a theoretical and practical point-of-view. More specifically, address the autonomous flight control problem considering a Tilt-rotor UAV allows the investigation of some advanced topics on control theory applied to robotics. For instance, whole-body control of underactuated systems, optimal control techniques, nonlinear control, dynamic modeling of multi-body systems, among others.

The ability of model predictive controllers to deal with multivariate interactions and constraints, made them an interesting choice to solve aerial robotic problems. However, due to the number of degrees of freedom and the high sampling rate of these systems, only a few works have considered this strategy when dealing with Tilt-rotor UAVs because of its potentially high computational cost. Further, the parametric uncertainties, modeling errors, and aerodynamic perturbations often limit the performance of linearized control strategies due to its restricted domain of attraction, which make it necessary the investigation of control techniques to provide robustness. Finally, autonomous flight is often regarded as a composition between dynamic control and navigation control, which are usually solved hierarchically through different control techniques. By using model predictive control strategies they can be uniquely addressed.

Load transportation using Tilt-rotor UAVs is a part of the research developed in the ProVANT project and has been previously addressed through different control techniques. This thesis contributes to this research by further investigating the use of predictive controllers in such problem.

1.4 Objectives

The main objective of this work is to investigate the load transportation control problem using a Tilt-rotor UAV with suspended load by means of model predictive controllers, with the trajectory tracking problem being addressed from the UAV's perspective with load stabilization and from the load's perspective while stabilizing the UAV. The aircraft considered in this work is the ProVANT 2.0 (Figure 1.3b) with a suspended load attached to its main body. Furthermore, since the considered UAV do not have aerodynamic surfaces, only helicopter flight-mode is considered.

The general requirements for all the controllers designed in this thesis are: to perform trajectory tracking while ensuring closed-loop stability throughout the trajectory; to reject constant external disturbances and parametric uncertainties; and to satisfy constraints on state deviations and control inputs. Five formulations of model predictive controllers are proposed along this work and their specific objectives can be listed as:

- Design a model predictive controller based on a linearized time-invariant model for trajectory tracking of the suspended load considering a standard quadratic stage cost together with stability regions. The control system algorithm must have suitable computational cost to be implemented in the aircraft's embedded system;

- Design a model predictive controller based on a linearized time-variant model for trajectory tracking of the suspended load considering a standard quadratic stage cost. The control system must be able to cope with yaw angle regulation and with the variation of the load's rope length during the take-off and landing maneuvers;

- Design an economic model predictive controller based on a linearized time-invariant model for trajectory tracking and obstacle avoidance of the Tilt-rotor UAV while stabilizing the suspended load. The control strategy must be formulated from a whole-body control approach by means of a standard quadratic stage cost, ensuring stability and performance, and an economic oriented stage cost, allowing obstacle deviation.

- Design a robust tube-based model predictive control strategy based on a linearized time-variant model for trajectory tracking of the Tilt-rotor UAV while stabilizing the suspended load. The proposed controller solves the load transportation problem through a cascade structure and must cope, by design concept, with the system's uncertainties.

- Design a robust tube-based economic model predictive control strategy based on a linearized time-variant model for trajectory tracking and obstacle avoidance of the Tilt-rotor UAV while stabilizing the suspended load. The control strategy must solves the load transportation problem through a cascade structure and considers the system's uncertainties rejection and planar obstacle avoidance in its formulation.

1.5 Structure of the text

This thesis is organized as follows:

- **Chapter 2:** describes some of the main mathematical tools used in this work, such as: set theory concepts and operators for model predictive control formulation; potential functions; and convex optimization principles with focus on linear matrix inequalities.

- **Chapter 3:** design two types of linear model predictive controllers to solve the load transportation problem using a Tilt-rotor UAV from the load's perspective. Initially, a linear time-invariant model is presented, and after its formulation is extended to

a linear time-variant controller. The features and drawback of both controllers are shown, and numerical simulations are used to validate the results.

- **Chapter 4:** seeks to design a robust predictive controller able to cope with obstacle avoidance. Therefore, initially an economic model predictive controller using potential functions as economic stage cost is presented to deal with the obstacle avoidance problem, and further a robust tube-based model predictive controller is designed to achieve the desired robustness feature. Both controllers are then gathered in order to obtain a tube-based model predictive controller with economic criteria. Numerical simulations are presented to validate the results of each controller individually.

- **Chapter 5:** summarizes the contributions and results of this work, and presents suggestions for future works in this line of research.

- **Appendix A:** presents the modeling process of the Tilt-rotor UAV with suspended load from both UAV's and load's perspectives. These models are used in this thesis for control design purposes.

- **Appendix B:** presents a nonlinear two-level cascade control strategy able to solve the path tracking problem of a Tilt-rotor UAV while transporting a suspended load stably. The inner controller is designed using the input-output feedback linearization technique, and for the outer-loop the controllers presented in chapter 4 are considered.

1.6 Publications

During the development of this Master's thesis, the following papers were elaborated and accepted for publication:

Conference papers:

1. (Santos & Raffo, 2016a) Santos, M. A. & Raffo, G. V. (2016). Adaptive control of a tilt-rotor UAV in load transportation tasks - a lmi based approach. In *Proc. of the XXI Congresso Brasileiro de Automática* (pp. 2461 - 2466).

2. (Santos & Raffo, 2016b) Santos, M. A. & Raffo, G. V. (2016). Path tracking model predictive control of a tilt-rotor UAV carrying a suspended load. In *Proc. of the IEEE 19th International Conference on Intelligent Transportation Systems* (pp. 1458 - 1463).

3. (Santos et al., 2017a) Santos, M. A., Cardoso, D. N., Rego, S. B., Raffo, G. V., & Esteban, S. (2017). A discrete robust adaptive control of a tilt-rotor uav for an enlarged flight envelope. In *Proc. of the 56th IEEE Conference on Decision and Control* (pp. 5208 - 5214).

Journal papers:

1. (Santos et al., 2017b) Santos, M. A., Rego, B. S., Raffo, G. V., & Ferramosca, A. (2017). Suspended load path tracking control strategy using a tilt-rotor UAV. In *Journal of Advanced Transportation* (pp. 1 - 22).

2 Mathematical Preliminaries

This chapter presents the background on the mathematical tools used throughout this master's thesis, which are: set methods in control theory, linear matrix inequalities, and potential functions.

2.1 Set methods in control theory

This section presents basic definitions about sets, their operations and applications in control theory. All definitions presented here can be found with further details on the works of Blanchini & Miani (2007), Nguyen (2014), Kerrigan & Maciejowski (2000), and Kerrigan (2000).

2.1.1 General definitions

The notation $\boldsymbol{A} \subseteq \boldsymbol{B}$ is used to denote that $\boldsymbol{A}$ is a subset of $\boldsymbol{B}$, $\boldsymbol{A} \subset \boldsymbol{B}$ denotes that $\boldsymbol{A}$ is a proper subset of $\boldsymbol{B}$, and $a \in \boldsymbol{B}$ denotes that a belongs to $\boldsymbol{B}$. Further, $\mathbb{N}$ is the set of natural numbers and $\mathbb{R}$ is the set of real numbers.

Definition 2.1 (Closed set (Nguyen, 2014))**.** A set $\mathcal{S} \subset \mathbb{R}^n$ is closed if it contains its own boundary.

Definition 2.2 (Bounded set (Nguyen, 2014))**.** A set $\mathcal{S} \subset \mathbb{R}^n$ is said to be bounded if it is contained in some ball $B_R = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\|_2 \leq \epsilon\}$ with finite radius $\epsilon > 0$.

Definition 2.3 (Compact set (Nguyen, 2014))**.** A set $\mathcal{S} \subset \mathbb{R}^n$ is compact if it is closed and bounded.

A set is said to be convex if, given two points, every point on the line segment joining these two points is also a member of the set. This geometric interpretation of a convex set can be formally stated by the follow definition.

Definition 2.4 (Convex set (Nguyen, 2014))**.** A set $\mathcal{S} \subset \mathbb{R}^n$ is said to be convex if for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{S}$ and every number $\alpha \in \mathbb{R}$, $0 < \alpha < 1$, the point $\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2 \in \mathcal{S}$.

Definition 2.5 (Linear variety (Nguyen, 2014))**.** A set $\boldsymbol{H} \subset \mathbb{R}^n$ is said to be a linear variety, if for every $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{H}$ and every $\alpha \in \mathbb{R}$, the point $\alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2 \in \boldsymbol{H}$.

Definition 2.6 (Hyperplane (Nguyen, 2014))**.** A hyperplane $\boldsymbol{H}(\boldsymbol{f}, g)$ is a set of the form

$$\boldsymbol{H}(\boldsymbol{f}, g) = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{f}^T\boldsymbol{x} = g\},$$

where $\boldsymbol{f} \in \mathbb{R}^n$ and $g \in \mathbb{R}$.

From the geometric point of view, a hyperplane in $\mathbb{R}^n$ can be defined as an $(n-1)$-dimensional linear variety. For instance, in $\mathbb{R}^3$, a plane, which is a 2-dimensional linear variety, is a hyperplane.

Definition 2.7 (Half-spaces (Nguyen, 2014))**.** From the definition of hyperplane, a closed half-space $\mathcal{H}(\boldsymbol{f}, g)$ is a set of the form

$$\mathcal{H}(\boldsymbol{f}, g) = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{f}^T\boldsymbol{x} \leq g\},$$

where $\boldsymbol{f} \in \mathbb{R}^n$ and $g \in \mathbb{R}$.

Definition 2.8 (Polyhedral set (Blanchini & Miani, 2007))**.** A convex polyhedral set $\boldsymbol{P}(\boldsymbol{F}, \boldsymbol{g})$ is a set of the form

$$\boldsymbol{P}(\boldsymbol{F}, \boldsymbol{g}) = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F}^T\boldsymbol{x} \leq \boldsymbol{g}^T\},$$

where $\boldsymbol{F} = \begin{bmatrix} \boldsymbol{f}_1^T & \boldsymbol{f}_2^T & \cdots & \boldsymbol{f}_m^T \end{bmatrix}$ and $\boldsymbol{g} = \begin{bmatrix} g_1 & g_2 & \cdots & g_m \end{bmatrix}$ with $\boldsymbol{f}_i \in \mathbb{R}^n$ $\forall i$, and $g_i \in \mathbb{R}$ $\forall i$. In other words, a convex polyhedral is expressed as the intersection of a finite number of half-spaces.

Definition 2.9 (Polytope (Blanchini & Miani, 2007))**.** A polytope is a non-empty bounded polyhedral set.

2.1.2 Set operations

Set operations are useful tools in control theory, hence three of them are defined in this subsection, while the well-known operations, such as union ($\bigcup$), intersection ($\bigcap$), difference ($-$), and Cartesian product ($\times$) are disregarded.

Definition 2.10 (Minkowski sum (Nguyen, 2014))**.** The Minkowski sum of two polytopes $\boldsymbol{P}_1, \boldsymbol{P}_2 \subset \mathbb{R}^n$ is a polytope (see Figure 2.1)

$$\boldsymbol{P}_1 \oplus \boldsymbol{P}_2 = \{\boldsymbol{x}_1 + \boldsymbol{x}_2 : \boldsymbol{x}_1 \in \boldsymbol{P}_1, \boldsymbol{x}_2 \in \boldsymbol{P}_2\}.$$

Definition 2.11 (Pontryagin difference (Nguyen, 2014))**.** The Pontryagin difference of two polytopes $\boldsymbol{P}_1, \boldsymbol{P}_2 \subset \mathbb{R}^n$ is a polytope (see Figure 2.2)

$$\boldsymbol{P}_1 \ominus \boldsymbol{P}_2 = \{\boldsymbol{x}_1 \in \boldsymbol{P}_1 : \boldsymbol{x}_1 + \boldsymbol{x}_2 \in \boldsymbol{P}_1, \forall\ \boldsymbol{x}_2 \in \boldsymbol{P}_2\}.$$

Definition 2.12 (Affine transformation (Nguyen, 2014))**.** Let $\mathcal{S} \subset \mathbb{R}^n$ be a convex set, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$. An affine transformation

$$\boldsymbol{A}\mathcal{S} + \boldsymbol{b} = \{\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b} : \boldsymbol{x} \in \mathcal{S}\}$$

is also a convex set defined in $\mathbb{R}^m$.

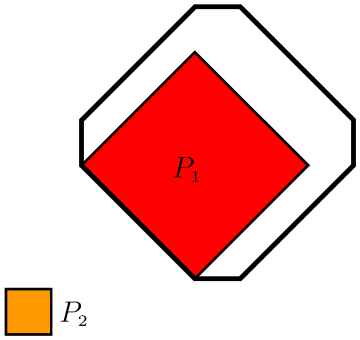


Figure 2.1: The Minkowski sum between the polytopes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.

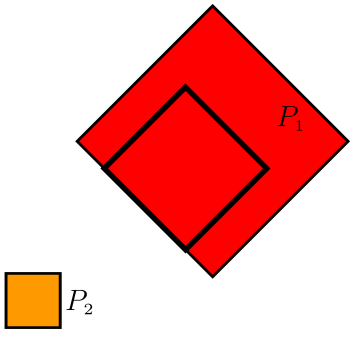


Figure 2.2: The Pontryagin difference between the polytopes $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$.

Figure 2.3 shows two affine transformations for a convex set $\mathcal{S}$. In the left part of the figure, an affine transformation is performed considering $n = m$ (see Definition 2.12), i.e., $\boldsymbol{A}$ is assumed to be a full rank square matrix. Therefore, the convex set resulting from the transformation maintains its dimension. On the other hand, in the right part, an affine transformation is performed considering $n \neq m$, i.e., $\boldsymbol{A}$ is a non-square matrix. Hence, the convex set resulting from the transformation is a transformed projection.

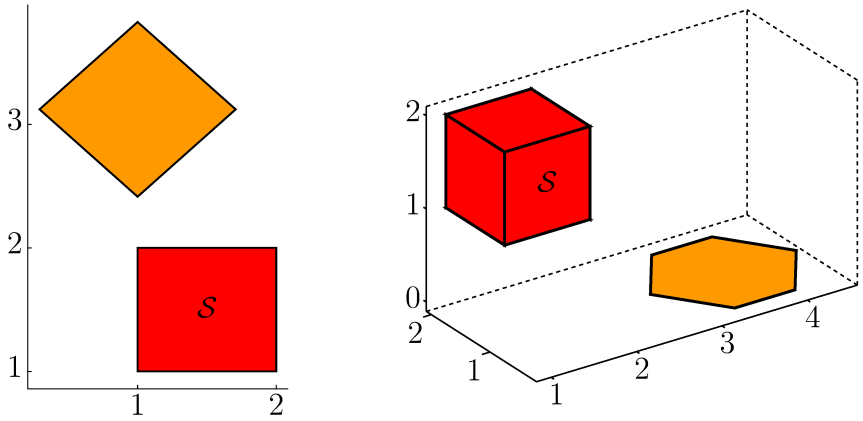


Figure 2.3: Affine transformation for a convex set $\mathcal{S}$ considering a square transformation matrix (left) and a non-square transformation matrix (right).

2.1.3 Invariant sets

The invariant set theory results are fundamental to design controllers for constrained systems and to understand some of their properties, for instance, stability and feasibility. Therefore, the following definitions describe these sets for systems without and with uncertainties. Moreover, let $\mathbb{U} \subset \mathbb{R}^m$, $\mathbb{X} \subset \mathbb{R}^n$, and $\mathbb{W} \subset \mathbb{R}^n$ denote compact sets and $\boldsymbol{\Omega}$ denotes any arbitrary subset in $\mathbb{R}^n$.

Definition 2.13 (Positively invariant set (Kerrigan & Maciejowski, 2000))**.** The set $\boldsymbol{\Omega} \subset \mathbb{R}^n$ is said to be positively invariant for the system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$ if $\forall \boldsymbol{x}_0 \in \boldsymbol{\Omega}$, the system evolution satisfies $\boldsymbol{x}_k \in \boldsymbol{\Omega}$, $\forall k \in \mathbb{N}$. In other words, if a system reaches a positively invariant set, it will stay inside this set.

Definition 2.14 (Maximal positively invariant set (Kerrigan & Maciejowski, 2000))**.** The set $\mathcal{O}_\infty(\boldsymbol{\Omega})$ is said to be the maximal positively invariant set contained in $\boldsymbol{\Omega}$ for the system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k)$ if $\mathcal{O}_\infty(\boldsymbol{\Omega})$ is positively invariant and contains all positively invariants sets contained in $\boldsymbol{\Omega}$. Therefore, if $\boldsymbol{\Phi}$ is a positively invariant set, $\boldsymbol{\Phi} \subseteq \mathcal{O}_\infty(\boldsymbol{\Omega}) \subseteq \boldsymbol{\Omega}$.

The concept of positively invariant set can be extended to closed-loop systems through the definition of the control invariant set.

Definition 2.15 (Control invariant set (Kerrigan & Maciejowski, 2000))**.** The set $\boldsymbol{\Omega} \subset \mathbb{R}^n$ is said to be a control invariant set for the system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ with $\boldsymbol{x}_k \in \mathbb{X}$ and $\boldsymbol{u}_k \in \mathbb{U}$, if for all $\boldsymbol{x}_0 \in \boldsymbol{\Omega}$ there exists $\boldsymbol{u_k} = \boldsymbol{h}(\boldsymbol{x}_k)$ such that $\boldsymbol{x}_{k+1} \in \boldsymbol{\Omega}$, $\forall k \in \mathbb{N}$.

Definition 2.16 (Maximal control invariant set (Kerrigan & Maciejowski, 2000))**.** The set $\mathcal{C}_\infty(\boldsymbol{\Omega})$ is said to be the maximal control invariant set contained in $\boldsymbol{\Omega}$ for the system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ if $\mathcal{C}_\infty(\boldsymbol{\Omega})$ is control invariant and contains all control invariant sets contained in $\boldsymbol{\Omega}$. Thus, if $\boldsymbol{\Phi}$ is a control invariant set, $\boldsymbol{\Phi} \subseteq \mathcal{C}_\infty(\boldsymbol{\Omega}) \subseteq \boldsymbol{\Omega}$.

The control invariant set can be obtained through the admissible set.

Definition 2.17 (Admissible set (Kerrigan & Maciejowski, 2000))**.** The i-th step admissible set $\mathcal{C}_i(\boldsymbol{\Omega})$ is the set of states for which exists an admissible control sequence able to keep the state's evolution inside $\boldsymbol{\Omega}$ during i steps, i.e.,

$$\mathcal{C}_i(\boldsymbol{\Omega}) = \{\boldsymbol{x}_0 \in \boldsymbol{\Omega} : \forall k = 0, \cdots, i-1, \exists \boldsymbol{u}_k \in \mathbb{U} \text{ such that } \boldsymbol{x}_{k+1} \in \boldsymbol{\Omega}\}.$$

The admissible set have some properties useful for numerical implementation of control invariant sets.

Property 2.1. The sequence $\mathcal{C}_i(\boldsymbol{\Omega})$ satisfies the properties:

(i) Each set $\mathcal{C}_{i+1}(\boldsymbol{\Omega}) \subseteq \mathcal{C}_i(\boldsymbol{\Omega})$;

(ii) Each set $\mathcal{C}_i(\boldsymbol{\Omega}) = \bigcap_{k=0}^{i} \mathcal{C}_k(\boldsymbol{\Omega})$;

(iii) $\mathcal{C}_\infty(\boldsymbol{\Omega})$ is finitely determined if $\exists i \in \mathbb{N}$ such that $\mathcal{C}_{i+1}(\boldsymbol{\Omega}) = \mathcal{C}_i(\boldsymbol{\Omega})$. Therefore, $\mathcal{C}_i(\boldsymbol{\Omega}) = \mathcal{C}_\infty(\boldsymbol{\Omega})$.

Definition 2.18 (The one-step set $Q(\boldsymbol{\Omega})$ (Kerrigan & Maciejowski, 2000))**.** The one-step set $Q(\boldsymbol{\Omega})$ is defined as the set of $\boldsymbol{x} \in \mathbb{R}^n$ for which an admissible control input exists and can drive the system to $\boldsymbol{\Omega}$ in one-step, i.e.,

$$Q(\boldsymbol{\Omega}) = \{\boldsymbol{x}_k \in \mathbb{R}^n : \exists \boldsymbol{u}_k \in \mathbb{U} \text{ such that } \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \in \boldsymbol{\Omega}, \forall k \in \mathbb{N}\}$$

Using the definition of the one-step operator it is possible to state a geometric condition for invariance.

Theorem 1 (Geometric condition for invariance (Kerrigan & Maciejowski, 2000))**.** *The set $\boldsymbol{\Omega} \in \mathbb{R}^n$ is a control invariant set if and only if $\boldsymbol{\Omega} \subseteq Q(\boldsymbol{\Omega})$.*

This theorem gives a geometric interpretation for invariant sets making the one-step set a standard tool for computation of invariants thought iterative algorithms.

Definition 2.19 (Reachable set (Kerrigan & Maciejowski, 2000))**.** The reachable set of $\mathbf{\Omega}$, $\mathscr{R}(\mathbf{\Omega})$, is the set of states for which the system can evolve from $\mathbf{\Omega}$ through an admissible input in one-step, i.e.,

$$\mathscr{R}(\mathbf{\Omega}) = \{\boldsymbol{\omega} \in \mathbb{R}^n : \exists \boldsymbol{x}_k \in \mathbf{\Omega}, \exists \boldsymbol{u}_k \in \mathbb{U} \text{ such that } \boldsymbol{\omega} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k) \forall k \in \mathbb{N}\}.$$

Definition 2.20. (Robust positively invariant set (Kerrigan, 2000)) The set $\mathbf{\Omega}$ is said to be robust positively invariant for the uncertain system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{w}_k)$ if $\forall \boldsymbol{x}_0 \in \mathbf{\Omega}$ and $\forall \boldsymbol{w}_k \in \mathbb{W}$, the system evolution satisfies $\boldsymbol{x}_k \in \mathbf{\Omega}$, $\forall k \in \mathbb{N}$. In other words, if a system reaches a robust positively invariant set, it will stay inside this set despite the uncertainties, i.e., $\boldsymbol{x}_k \in \mathbf{\Omega} \Rightarrow \boldsymbol{x}_{k+1} \in \mathbf{\Omega}, \forall \boldsymbol{w}_k \in \mathbb{W}$.

Definition 2.21. (Maximal robust positively invariant set (Kerrigan, 2000)) The set $\tilde{\mathcal{O}}_\infty(\mathbf{\Omega})$ is said to be the maximal robust positively invariant set contained in $\mathbf{\Omega}$ for the uncertain system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{w}_k)$ if $\tilde{\mathcal{O}}_\infty(\mathbf{\Omega})$ is robust positively invariant and contains all the robust positively invariant sets contained in $\mathbf{\Omega}$. Therefore, if $\mathbf{\Phi}$ is a robust positively invariant set, $\mathbf{\Phi} \subseteq \tilde{\mathcal{O}}_\infty(\mathbf{\Omega}) \subseteq \mathbf{\Omega}$.

Definition 2.22. (Robust control invariant set (Kerrigan, 2000)) The set $\mathbf{\Omega} \subset \mathbb{R}^n$ is said to be a robust control invariant set for the uncertain system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k)$ with $\boldsymbol{x}_k \in \mathbb{X}$, $\boldsymbol{u}_k \in \mathbb{U}$, and $\boldsymbol{w}_k \in \mathbb{W}$; if for all $\boldsymbol{x}_0 \in \mathbf{\Omega}$ there exists $\boldsymbol{u}_k = \boldsymbol{h}(\boldsymbol{x}_k), \forall \boldsymbol{x}_k \in \mathbf{\Omega}, \forall k \in \mathbb{N}$, such that $\boldsymbol{x}_{k+1} \in \mathbf{\Omega}$, $\forall \boldsymbol{w}_k \in \mathbb{W}$.

Definition 2.23. (Maximal robust control invariant set (Kerrigan, 2000)) The set $\tilde{\mathcal{C}}_\infty(\mathbf{\Omega})$ is said to be the maximal robust control invariant set contained in $\mathbf{\Omega}$ for the uncertain system $\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k)$ if $\tilde{\mathcal{C}}_\infty(\mathbf{\Omega})$ is robust control invariant and contains all the robust control invariant sets contained in $\mathbf{\Omega}$. Therefore, if $\mathbf{\Phi}$ is a robust control invariant set, $\mathbf{\Phi} \subseteq \tilde{\mathcal{C}}_\infty(\mathbf{\Omega}) \subseteq \mathbf{\Omega}$.

As for the case without uncertainties, the useful one-step set can be defined in order to provide an operator to deal with the set invariance for uncertain systems.

Definition 2.24. (The robust one-step set $\tilde{Q}(\mathbf{\Omega})$ (Kerrigan, 2000)) The robust one-step set $\tilde{Q}(\mathbf{\Omega})$ is defined as the set of $\boldsymbol{x} \in \mathbb{R}^n$ for which an admissible control input exists and can drive the system to $\mathbf{\Omega}$ in one step, for all considered disturbances, i.e,

$$\tilde{Q}(\mathbf{\Omega}) = \{\boldsymbol{x}_k \in \mathbb{R}^n : \exists \boldsymbol{u}_k \in \mathbb{U} \text{ such that } f(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{w}_k) \in \mathbf{\Omega}, \forall \boldsymbol{w}_k \in \mathbb{W}, \forall k \in \mathbb{N}\}.$$

Theorem 2 (Geometric condition for robust invariance (Kerrigan, 2000))**.** *The set $\mathbf{\Omega} \in \mathbb{R}^n$ is a robust control invariant set if and only if $\mathbf{\Omega} \subseteq \tilde{Q}(\mathbf{\Omega})$.*

2.2 Linear Matrix Inequalities

Convex optimization problems over Linear Matrix Inequalities (LMIs) constraints can be used to formulate many optimization problems in control theory through numerical efficient methods.

Definition 2.25 (Linear Matrix Inequality (Boyd et al., 1991))**.** A LMI is a matrix inequality of type $\boldsymbol{F}(\boldsymbol{x}) > \mathbf{0}$ in which $\boldsymbol{F}(\boldsymbol{x}) : \mathbb{R}^m \to \mathbb{R}^{n\times n}$ is symmetric and affine in the decision variables $\boldsymbol{x}$. It can be generally written in the form

$$\boldsymbol{F}(\boldsymbol{x}) = \boldsymbol{F}_0 + \sum_{i=1}^{m} x_i \boldsymbol{F}_i > \mathbf{0},$$

where $\boldsymbol{F}_i \in \mathbb{R}^{n\times n}$ are given matrices, with $i \in \{1, 2, \cdots, m\}$ and $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T$.

Definition 2.26 (Linear Matrix Inequality feasibility (Boyd et al., 1991))**.** A LMI is said to be feasible if $\exists \boldsymbol{x} \in \mathbb{R}^m$ such that $\boldsymbol{F}(\boldsymbol{x}) > \mathbf{0}$.

Often, the inequalities obtained in the control problems' design are nonlinear functions of the decision variables. A useful tool to commonly work around these situations are the Schur complement.

Definition 2.27 (Schur complement (Boyd et al., 1991))**.** Let $\boldsymbol{M}_1(\boldsymbol{x})$, $\boldsymbol{M}_2(\boldsymbol{x})$, and $\boldsymbol{M}_3(\boldsymbol{x})$ be affine functions of the decision variable $\boldsymbol{x} \in \mathbb{R}^m$ with $\boldsymbol{M}_1(\boldsymbol{x}) = \boldsymbol{M}_1(\boldsymbol{x})^T$ and $\boldsymbol{M}_2(\boldsymbol{x}) = \boldsymbol{M}_2(\boldsymbol{x})^T$. The Schur complement states that the following inequalities are equivalent.

(i) $\boldsymbol{M}_1(\boldsymbol{x}) - \boldsymbol{M}_3(\boldsymbol{x})^T \boldsymbol{M}_2(\boldsymbol{x})^{-1} \boldsymbol{M}_3(\boldsymbol{x}) > \mathbf{0}$ with $\boldsymbol{M}_2(\boldsymbol{x}) > \mathbf{0}$

(ii) $\begin{bmatrix} \boldsymbol{M}_1(\boldsymbol{x}) & \boldsymbol{M}_3(\boldsymbol{x})^T \\ \boldsymbol{M}_3(\boldsymbol{x}) & \boldsymbol{M}_2(\boldsymbol{x}) \end{bmatrix} > \mathbf{0}$

2.3 Potential functions

As stated in Choset et al. (2015), a potential function is a differentiable real-valued function $U : \mathbb{R}^n \mapsto \mathbb{R}$ that can be seen as an energy function whose gradient gives a force vector field pointing in the direction where U increases. Two types of potential functions will be used in this work: the attraction potential function, having its gradient representing an attractive force field, and the repulsive potential function, having its gradient representing a repulsive force field.

The attraction potential function is defined having its value increasing with the distance between two variables a and b. Both conic and quadratic functions satisfy this requirement, however, since the quadratic function is continuously differentiable the attractive potential function is defined from it. Therefore,

$$U_{att} = \frac{1}{2} \kappa d\left(a, b\right)^2, \tag{2.1}$$

with $d(a, b)$ being the Euclidean distance between the variables a and b, given by the 2-norm distance, and κ is a weight parameter.

On the other hand, inspired by the electric potential field, the repulsive potential function can be described as being inversely proportional to the square of the distance between the variables a and b, yielding to

$$U_{rep} = \frac{1}{2}\lambda \frac{1}{d(a, b)^2}, \tag{2.2}$$

with λ being a weight parameter.

Further, it is possible to combine the attraction and repulsive functions for describing an unified potential field as

$$U = U_{att} + U_{rep}. \tag{2.3}$$

The function (2.3) can be used, for instance, to address robot navigation problems and to describe electrical iteration between charged particles. More information about potential functions and their use in robotics can be found in Choset et al. (2015).

2.4 Final remarks

In this chapter the main mathematical concepts used along this work were formally defined aiming to provide a better understanding of this work. In the next chapters model predictive controllers will be designed with the help of the mathematical tools presented in this chapter to solve the control problem of a Tilt-rotor UAV performing load transportation tasks.

3

Linear Model Predictive Control

This chapter presents the design of two linear model predictive controllers (MPC) in order to solve the load transportation problem using a Tilt-rotor UAV. The proposed controllers must perform trajectory tracking of the suspended load in helicopter-flight mode while keeping the vehicle stabilized. The main objectives of the control system are: ensure closed-loop stability, reject constant external disturbances and parametric uncertainties, and satisfy constraints on state deviations and control inputs.

The control problem is formulated in this chapter considering a state feedback structure, i.e., all states are assumed to be known either by measurement or estimation. Since the Tilt-rotor UAV is an underactuated mechanical system[1] and aiming an improved trajectory tracking control, the load's translational position and its yaw angle are chosen to be regulated, while the others degrees of freedom are only stabilized.

For complex systems, such as the Tilt-rotor UAV, some control techniques widely used, for instance control by pole allocation, may not achieve satisfactory performance because of the system's underactuation and coupled dynamics behaviors. In this context, MPC become interesting because of their model-based nature and ability to deal with multivariable systems in a simple way (Rossiter, 2013).

In general, model predictive controllers are a class of optimal receding horizon controllers, in which the control action applied to the system is obtained by solving at each sampling time a finite horizon open-loop optimal control problem considering the current state as initial one for predictions. The optimization problem's solution gives an optimal control

[1] Those readers who want further details about the Tilt-rotor UAV modeling may refer to Appendix A.

sequence, from which the first control of the obtained sequence is applied to the system as input. Afterward, the horizon is receded and the process starts again considering the new current state as initial condition. The essence behind this procedure lies on the ability to predict N_p future states having the current information of the system and its dynamic model. Hereafter, the predicted states could be used in order to obtain N_c control actions able to drive the system optimally along some desired trajectory within the prediction horizon (Camacho & Bordons, 2004). The variables N_p and N_c are tuning parameters of the MPC algorithm and are called, respectively, prediction and control horizons. The prediction horizon defines how many system's outputs are predicted within the receding horizon time span in order to formulate the optimization problem, while the control horizon defines how many elements compose the optimal control sequence. Usually, higher prediction horizons increase the system's time-response, while smaller ones make the system's response to become oscillatory and even unstable, depending on how small the prediction horizon is regarding the system's transient time. On the other hand, large control horizons result in better closed-loop performance because more control changes are allowed during the system's transient (Rossiter, 2013). Work with both horizons allows to seek for better closed-loop response with smaller computational effort. Mainly, it allows to enlarge the system's domain of attraction without increasing the number of decision variables in the MPC algorithm.

Initially at this chapter, a linear time-invariant model predictive controller (LTI-MPC) and a linear time-variant model predictive controller (LTV-MPC) will be generally designed (Santos et al., 2017b). Later on, both controllers will be particularized for the trajectory tracking of the suspended load with stabilization of the Tilt-rotor UAV, being their features and drawbacks when used in such application highlighted and compared through numeric simulation results.

3.1 Linear time-invariant model predictive controller

This section deals with the design of the LTI-MPC.

3.1.1 Problem statement

Consider a finite-dimensional nonlinear system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{3.1}$$

where $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$ denotes the state vector and $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^m$ denotes the input vector with $\mathbb{X}$ and $\mathbb{U}$ being, respectively, the set of admissible states and inputs. Further, $\boldsymbol{f} : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{X}$ is a state-transition nonlinear map.

In this section, the control problem for the LTI-MPC is defined as the problem of designing a model predictive controller able to perform trajectory tracking of the suspended load while stabilizing the UAV. The controller considers a linear time-invariant discrete model with state-space representation as the base of its prediction process and, aiming to achieve improved performance, it works with the incremental form of the MPC. Besides, it must ensure closed-loop stability throughout the trajectory respecting the system's constraints even in the presence of constant external disturbances and parametric uncertainties.

Therefore, consider the mapping of the system (3.1) from the continuous-time to the discrete-time domain after linearized through first-order expansion in Taylor series, which yields the linear system's error model

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{A}\Delta \boldsymbol{x}_k + \boldsymbol{B}\Delta \boldsymbol{u}_k, \tag{3.2}$$

where $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{tr}$ and $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$, with the superscript $(\cdot)^{tr}$ denoting desired trajectory variables, and $\boldsymbol{A}$ and $\boldsymbol{B}$ being, respectively, the state and input Jacobians linear model matrices. Additionally, the pair $(\boldsymbol{A}, \boldsymbol{B})$ is assumed to be controllable and $\Delta \boldsymbol{x}_k \in \mathbb{E} \subset \mathbb{R}^n$, with $\mathbb{E}$ being a compact set limiting the state error.[2]

The linearized system (3.2) gives the one-step ahead prediction of $\Delta \boldsymbol{x}_k$. Thus, recursively using (3.2), the prediction of the future states of the system can be obtained by the following procedure:

1. Evaluate (3.2) at the time instant $k+2$ together with the knowledge of $\Delta \boldsymbol{x}_{k+1}$, which gives the two-step ahead prediction

$$\begin{aligned} \Delta \boldsymbol{x}_{k+2} &= \boldsymbol{A}\Delta \boldsymbol{x}_{k+1} + \boldsymbol{B}\Delta \boldsymbol{u}_{k+1} \\ &= \boldsymbol{A}\left[\boldsymbol{A}\Delta \boldsymbol{x}_k + \boldsymbol{B}\Delta \boldsymbol{u}_k\right] + \boldsymbol{B}\Delta \boldsymbol{u}_{k+1} \\ &= \boldsymbol{A}^2\Delta \boldsymbol{x}_k + \boldsymbol{A}\boldsymbol{B}\Delta \boldsymbol{u}_k + \boldsymbol{B}\Delta \boldsymbol{u}_{k+1} \end{aligned} \tag{3.3}$$

2. Consider (3.2) at $k+3$ with equation (3.3), yielding the three-step ahead prediction

$$\begin{aligned} \Delta \boldsymbol{x}_{k+3} &= \boldsymbol{A}\Delta \boldsymbol{x}_{k+2} + \boldsymbol{B}\Delta \boldsymbol{u}_{k+2} \\ &= \boldsymbol{A}\left[\boldsymbol{A}^2\Delta \boldsymbol{x}_k + \boldsymbol{A}\boldsymbol{B}\Delta \boldsymbol{u}_k + \boldsymbol{B}\Delta \boldsymbol{u}_{k+1}\right] + \boldsymbol{B}\Delta \boldsymbol{u}_{k+2} \\ &= \boldsymbol{A}^3\Delta \boldsymbol{x}_k + \boldsymbol{A}^2\boldsymbol{B}\Delta \boldsymbol{u}_k + \boldsymbol{A}\boldsymbol{B}\Delta \boldsymbol{u}_{k+1} + \boldsymbol{B}\Delta \boldsymbol{u}_{k+2} \end{aligned} \tag{3.4}$$

3. Follow this recursive procedure until the n-step ahead prediction, resulting in

$$\Delta \boldsymbol{x}_{k+n} = \boldsymbol{A}^n\Delta \boldsymbol{x}_k + \boldsymbol{A}^{n-1}\boldsymbol{B}\Delta \boldsymbol{u}_k + \cdots + \boldsymbol{B}\Delta \boldsymbol{u}_{k+n-1} \tag{3.5}$$

[2] In this work variables in continuous-time and discrete-time domains are differentiated by the time variable t and the sampling variable k.

Therefore, the sequence of future states of the system up to the time instant $k+n$ can be generally expressed by the matricial form

$$\begin{bmatrix} \Delta \boldsymbol{x}_{k+1} \\ \Delta \boldsymbol{x}_{k+2} \\ \vdots \\ \Delta \boldsymbol{x}_{k+n} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^n \end{bmatrix} \Delta \boldsymbol{x}_k + \begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{AB} & \boldsymbol{B} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{n-1}\boldsymbol{B} & \boldsymbol{A}^{n-2}\boldsymbol{B} & \cdots & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{u}_{k+1} \\ \vdots \\ \Delta \boldsymbol{u}_{k+n-1} \end{bmatrix} \tag{3.6}$$

where $\boldsymbol{0}$ denotes zero matrices with appropriated dimensions.

Equation (3.6) assumes that the prediction and control horizon are equal. However, when $N_c < N_p$ the last computed control action needs to be held since the predicted state sequence has more elements than the control input sequence, thus $\Delta \boldsymbol{u}_{k+i} = \Delta \boldsymbol{u}_{k+N_c}$, $\forall\, N_c \leq i \leq N_p$. Thereafter, when $N_c < N_p$, the previous derived prediction model (3.6) can be rewritten as

$$\underbrace{\begin{bmatrix} \Delta \boldsymbol{x}_{k+1} \\ \Delta \boldsymbol{x}_{k+2} \\ \vdots \\ \Delta \boldsymbol{x}_{k+N_p} \end{bmatrix}}_{\underrightarrow{\Delta \boldsymbol{x}}} = \underbrace{\begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A}^2 \\ \vdots \\ \boldsymbol{A}^{N_p} \end{bmatrix}}_{\boldsymbol{\mathcal{P}}_{\text{ni}}} \Delta \boldsymbol{x}_k + \underbrace{\begin{bmatrix} \boldsymbol{B} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{AB} & \boldsymbol{B} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{N_p-1}\boldsymbol{B} & \boldsymbol{A}^{N_p-2}\boldsymbol{B} & \cdots & \left(\sum_{i=0}^{N_p-N_c} \boldsymbol{A}^i\right)\boldsymbol{B} \end{bmatrix}}_{\boldsymbol{\mathcal{H}}_{\text{ni}}} \underbrace{\begin{bmatrix} \Delta \boldsymbol{u}_k \\ \Delta \boldsymbol{u}_{k+1} \\ \vdots \\ \Delta \boldsymbol{u}_{k+N_c-1} \end{bmatrix}}_{\underrightarrow{\Delta \boldsymbol{u}}}. \tag{3.7}$$

Aiming to improve the control system's performance when constant external disturbances and parametric uncertainties affect the system, the increment of control defined as $\delta \boldsymbol{u}_k = \Delta \boldsymbol{u}_k - \Delta \boldsymbol{u}_{k-1}$, $\forall\, k \in \mathbb{N}$ is taken as input variable, where $\delta \boldsymbol{u}_k \in \mathbb{V} \subset \mathbb{R}^m$ with $\mathbb{V}$ being a compact set constraining the control increment. Thus, the model described in (3.2) can be rewritten as

$$\underbrace{\begin{bmatrix} \Delta \boldsymbol{x}_{k+1} \\ \Delta \boldsymbol{u}_k \end{bmatrix}}_{\Delta \bar{\boldsymbol{x}}_{\boldsymbol{k+1}}} = \underbrace{\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{0} & \mathbb{I} \end{bmatrix}}_{\bar{\boldsymbol{A}}} \underbrace{\begin{bmatrix} \Delta \boldsymbol{x}_k \\ \Delta \boldsymbol{u}_{k-1} \end{bmatrix}}_{\Delta \bar{\boldsymbol{x}}_{\boldsymbol{k}}} + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \mathbb{I} \end{bmatrix}}_{\bar{\boldsymbol{B}}} \delta \boldsymbol{u}_k, \tag{3.8}$$

being $\mathbb{I}$ an identity matrix with appropriated dimension.

Then, using the augmented system (3.8) and making $\delta \boldsymbol{u}_{k+i} = \boldsymbol{0}$, $\forall\, i \geq N_c$, the prediction model (3.6) can be rewritten as

$$\underbrace{\begin{bmatrix} \Delta \bar{\boldsymbol{x}}_{k+1} \\ \Delta \bar{\boldsymbol{x}}_{k+2} \\ \vdots \\ \Delta \bar{\boldsymbol{x}}_{k+N_p} \end{bmatrix}}_{\underrightarrow{\Delta \bar{\boldsymbol{x}}}} = \underbrace{\begin{bmatrix} \bar{\boldsymbol{A}} \\ \bar{\boldsymbol{A}}^2 \\ \vdots \\ \bar{\boldsymbol{A}}^{N_p} \end{bmatrix}}_{\boldsymbol{\mathcal{P}}_i} \Delta \bar{\boldsymbol{x}}_k + \underbrace{\begin{bmatrix} \bar{\boldsymbol{B}} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \bar{\boldsymbol{A}}\bar{\boldsymbol{B}} & \bar{\boldsymbol{B}} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\boldsymbol{A}}^{N_p-1}\bar{\boldsymbol{B}} & \bar{\boldsymbol{A}}^{N_p-2}\bar{\boldsymbol{B}} & \cdots & \bar{\boldsymbol{A}}^{N_p-N_c}\bar{\boldsymbol{B}} \end{bmatrix}}_{\boldsymbol{\mathcal{H}}_i} \underbrace{\begin{bmatrix} \delta \boldsymbol{u}_k \\ \delta \boldsymbol{u}_{k+1} \\ \vdots \\ \delta \boldsymbol{u}_{k+N_c-1} \end{bmatrix}}_{\underrightarrow{\delta \boldsymbol{u}}}. \tag{3.9}$$

In this work, the modified model (3.8) is referred as the incremental model and (3.9) as the incremental prediction model, likewise, (3.7) is referred as the non-incremental prediction model. The main feature of working with the incremental formulation is the

addition of integrators to the system equals to the number of its control inputs (Rossiter, 2013). However, it has the drawback of increasing the dimension of the state-space which could make the controller's computational cost prohibitive and hence the use of the non-incremental model could be more suitable. On the other hand, when working with the non-incremental prediction model in a scenario where the difference between the prediction and control horizon is sufficiently large, i.e. $N_p \gg N_c$, the sums terms in the last block column of matrix $\boldsymbol{\mathcal{H}}_{ni}$ (see equation (3.7)) increase. Thus, the number of mathematical operations necessary to obtain the prediction model also increase, which could make the incremental prediction model more suitable.

The choice of which prediction model should be used is always a trade-off between good performance and small computational cost. In this work, aiming to improve the performance with the additional integrators, the incremental prediction model is used to formulate the predictive controllers. Therefore, the remaining of this chapter considers only the prediction model defined in (3.9). Note that, all definitions could be easily adapted to design non-incremental MPCs.

3.1.2 LTI-MPC optimization problem

As previously stated, the control action of an MPC is obtained through the optimization of a given performance measure. Thus, consider the standard quadratic cost function $\mathcal{J}$ for the incremental model (3.9)

$$\mathcal{J} = \sum_{i=0}^{N_p-1} \|\Delta\bar{\boldsymbol{x}}_{k+i}\|^2_{\boldsymbol{\mathcal{Q}}} + \sum_{j=0}^{N_c-1} \|\delta\boldsymbol{u}_{k+j}\|^2_{\boldsymbol{\mathcal{R}}} + \left\|\Delta\bar{\boldsymbol{x}}_{k+N_p}\right\|^2_{\boldsymbol{P}}, \tag{3.10}$$

where the matrices $\boldsymbol{\mathcal{Q}} > 0$ and $\boldsymbol{\mathcal{R}} > 0$ are, respectively, weighting matrices for the state error and the control effort, and $\boldsymbol{P} > 0$ is a matrix used to formulate a quadratic terminal cost added to the cost function to ensure closed-loop stability (Mayne et al., 2000). In addition of being positive definite, the matrix $\boldsymbol{P}$ must formulate a local Lyapunov function, e.g. $V(\Delta\bar{\boldsymbol{x}}_k) = \Delta\bar{\boldsymbol{x}}_k^T \boldsymbol{P} \Delta\bar{\boldsymbol{x}}_k$, and it must be monotonic.

Considering the prediction model (3.9), it is possible to rewrite equation (3.10) in the matricial form as

$$\mathcal{J} = \left(\boldsymbol{\mathcal{H}}_i \underrightarrow{\delta\boldsymbol{u}} + \boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k\right)^T \boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}} \left(\boldsymbol{\mathcal{H}}_i \underrightarrow{\delta\boldsymbol{u}} + \boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k\right) + \underrightarrow{\delta\boldsymbol{u}}^T \boldsymbol{\Omega}_{\boldsymbol{\mathcal{R}}} \underrightarrow{\delta\boldsymbol{u}} + \Delta\bar{\boldsymbol{x}}_{k+N_p}^T \boldsymbol{P} \Delta\bar{\boldsymbol{x}}_{k+N_p}, \tag{3.11}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}} = \mathbf{blkdiag}(\boldsymbol{\mathcal{Q}}, \cdots, \boldsymbol{\mathcal{Q}})$ and $\boldsymbol{\Omega}_{\boldsymbol{\mathcal{R}}} = \mathbf{blkdiag}(\boldsymbol{\mathcal{R}}, \cdots, \boldsymbol{\mathcal{R}})$. Finally, equation (3.11) can be rewritten in the canonical quadratic form

$$\mathcal{J} = \frac{1}{2} \underrightarrow{\delta\boldsymbol{u}}^T \boldsymbol{\Lambda} \underrightarrow{\delta\boldsymbol{u}} + \boldsymbol{f}^T \underrightarrow{\delta\boldsymbol{u}} + f_0, \tag{3.12}$$

where

$$\begin{aligned}\boldsymbol{\Lambda} &= 2\left(\boldsymbol{\mathcal{H}}_i^T \boldsymbol{\Omega}_{\mathcal{Q}} \boldsymbol{\mathcal{H}}_i + \boldsymbol{\Omega}_{\mathcal{R}}\right),\\ \boldsymbol{f}^T &= 2\left(\boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k\right)^T \boldsymbol{\Omega}_{\mathcal{Q}} \boldsymbol{\mathcal{H}}_i,\\ f_0 &= \left(\boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k\right)^T \boldsymbol{\Omega}_{\mathcal{Q}} \left(\boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k\right) + \Delta\bar{\boldsymbol{x}}_{k+N_P}^T \boldsymbol{P} \Delta\bar{\boldsymbol{x}}_{k+N_P}.\end{aligned}$$

A terminal cost able to ensure closed-loop stability can be defined through a Lyapunov function as $V(\Delta\bar{\boldsymbol{x}}_k) = \Delta\bar{\boldsymbol{x}}_k^T \boldsymbol{P} \Delta\bar{\boldsymbol{x}}_k$. Considering $V\left(\Delta\bar{\boldsymbol{x}}_k\right)$ as an upper bound for the linear quadratic regulator cost-to-go function, the following problem is considered

$$V(\Delta\bar{\boldsymbol{x}}_0) \geq \min_{\delta\boldsymbol{u}[0,\infty)} \sum_{k=0}^{\infty} \Delta\bar{\boldsymbol{x}}_k^T \boldsymbol{\mathcal{Q}} \Delta\bar{\boldsymbol{x}}_k + \delta\boldsymbol{u}_k^T \boldsymbol{\mathcal{R}} \delta\boldsymbol{u}_k. \tag{3.13}$$

In order to obtain $\boldsymbol{P}$, the necessary conditions to stability, i.e., $\boldsymbol{P} > 0$ and $V(\Delta\bar{\boldsymbol{x}}_{k+1}) - V(\Delta\bar{\boldsymbol{x}}_k) < 0$ (Boyd et al., 1991), must be considered. Therefore, it is possible to represent (3.13) as an inequality, yielding to

$$\left(\bar{\boldsymbol{A}}_f \Delta\bar{\boldsymbol{x}}_k\right)^T \boldsymbol{P} \left(\bar{\boldsymbol{A}}_f \Delta\bar{\boldsymbol{x}}_k\right) - \Delta\bar{\boldsymbol{x}}_k^T \boldsymbol{P} \Delta\bar{\boldsymbol{x}}_k \leq -\Delta\bar{\boldsymbol{x}}_k^T \left(\boldsymbol{\mathcal{Q}} + \boldsymbol{K}^T \boldsymbol{\mathcal{R}} \boldsymbol{K}\right) \Delta\bar{\boldsymbol{x}}_k, \tag{3.14}$$

where $\delta\boldsymbol{u}_k = \boldsymbol{K}\Delta\boldsymbol{x}_k$, $\bar{\boldsymbol{A}}_f = \bar{\boldsymbol{A}} + \bar{\boldsymbol{B}}\boldsymbol{K}$, and $\boldsymbol{K}$ is a stabilizing feedback gain.

The inequality (3.14) can be solved as a convex optimization problem if represented as an LMI (see Chapter 2, Section 2.2). Therefore, in order to express (3.14) as a valid LMI condition (see Definition 2.25), the inequality needs to be manipulated. Hence, rearranging (3.14) and applying the Schur complement twice (see Definition 2.27), it yields to

$$\begin{bmatrix} \boldsymbol{P} & \bar{\boldsymbol{A}}_f^T & \boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{K}\boldsymbol{\mathcal{R}}^{\frac{1}{2}} \\ \bar{\boldsymbol{A}}_f & \boldsymbol{P}^{-1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{0} & \mathbb{I} & \boldsymbol{0} \\ \boldsymbol{\mathcal{R}}^{\frac{1}{2}}\boldsymbol{K} & \boldsymbol{0} & \boldsymbol{0} & \mathbb{I} \end{bmatrix} \geq 0. \tag{3.15}$$

Seeking to remove the nonlinear terms, it is necessary to pre and post multiply (3.15) by the block diagonal matrix $\textbf{blkdiag}\left(\boldsymbol{P}^{-1}, \mathbb{I}, \mathbb{I}, \mathbb{I}\right)$, and define $\boldsymbol{S} = \boldsymbol{P}^{-1}$ and $\boldsymbol{Y} = \boldsymbol{K}\boldsymbol{P}^{-1}$. Therefore, the LMI condition is given by

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{S}^T\bar{\boldsymbol{A}}_f^T + \boldsymbol{Y}^T\bar{\boldsymbol{B}}^T & \boldsymbol{S}\boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{Y}^T\boldsymbol{\mathcal{R}}^{\frac{1}{2}} \\ \bar{\boldsymbol{A}}_f\boldsymbol{S} + \bar{\boldsymbol{B}}\boldsymbol{Y} & \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{Q}}^{\frac{1}{2}}\boldsymbol{S} & \boldsymbol{0} & \mathbb{I} & \boldsymbol{0} \\ \boldsymbol{\mathcal{R}}^{\frac{1}{2}}\boldsymbol{Y} & \boldsymbol{0} & \boldsymbol{0} & \mathbb{I} \end{bmatrix} \geq 0. \tag{3.16}$$

Finally, the matrix $\boldsymbol{P}$ can be obtained by solving the optimization problem

$$\begin{aligned} \min_{\boldsymbol{S}>0} \quad & \text{Tr}(\boldsymbol{P}), \\ \text{subject to} \quad & (3.16), \end{aligned} \tag{3.17}$$

where $\text{Tr}(\cdot)$ is the trace operator.

Further, in order to complete the LTI-MPC optimization problem (3.12) taking advantage of one of the main features of MPC methods, the following constraints are considered:

1. Input signal constraint:

   The main goal behind working with restrictions on the control signal is to avoid actuators' saturation, i.e., to allow the control algorithm to compute only admissible control inputs $\boldsymbol{u}_k \in \mathbb{U}$. Therefore, the control signal must be within the set

   $$\mathbb{U} = \{\boldsymbol{u}_k \in \mathbb{R}^m : \boldsymbol{u}^{min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}^{max}, \forall k \in \mathbb{N}\}, \tag{3.18}$$

   where $(\cdot)^{max}$ and $(\cdot)^{min}$ are, respectively, the maximum and minimum values allowed for the input variables.

   Knowing that $\Delta\boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$ and $\delta\boldsymbol{u}_k = \Delta\boldsymbol{u}_k - \Delta\boldsymbol{u}_{k-1}$, with $\boldsymbol{u}_k^{tr}$ being the control action that could drives the system towards the desired trajectory, it is possible to write the control signal at the instant k as

   $$\boldsymbol{u}_k = \boldsymbol{u}_k^{tr} + \Delta\boldsymbol{u}_{k-1} + \delta\boldsymbol{u}_k. \tag{3.19}$$

   Computing the control actions $\boldsymbol{u}_k$ up to $\boldsymbol{u}_{k+N_c-1}$, it yields to

   $$\underbrace{\begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{u}_{k+1} \\ \vdots \\ \boldsymbol{u}_{k+N_c-1} \end{bmatrix}}_{\underrightarrow{\boldsymbol{u}}} = \underbrace{\begin{bmatrix} \boldsymbol{u}_k^{tr} \\ \boldsymbol{u}_{k+1}^{tr} \\ \vdots \\ \boldsymbol{u}_{k+N_c-1}^{tr} \end{bmatrix}}_{\underrightarrow{\boldsymbol{u}^{tr}}} + \underbrace{\begin{bmatrix} \mathbb{I} \\ \mathbb{I} \\ \vdots \\ \mathbb{I} \end{bmatrix}}_{\boldsymbol{C}_1} \Delta\boldsymbol{u}_{k-1} + \underbrace{\begin{bmatrix} \mathbb{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbb{I} & \mathbb{I} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{I} & \mathbb{I} & \cdots & \mathbb{I} \end{bmatrix}}_{\boldsymbol{C}_2} \underrightarrow{\delta\boldsymbol{u}}. \tag{3.20}$$

   Then, the set (3.18) can be written in terms of the control increment through the inequalities

   $$\boldsymbol{C}_2\underrightarrow{\delta\boldsymbol{u}} \leq \boldsymbol{C}_1(\boldsymbol{u}^{max} - \Delta\boldsymbol{u}_{k-1}) - \underrightarrow{\boldsymbol{u}_k^{tr}}, \tag{3.21}$$

   $$-\boldsymbol{C}_2\underrightarrow{\delta\boldsymbol{u}} \leq -\boldsymbol{C}_1(\boldsymbol{u}^{min} - \Delta\boldsymbol{u}_{k-1}) + \underrightarrow{\boldsymbol{u}_k^{tr}}. \tag{3.22}$$

2. Maximum state error constraint:

   Since linearized models could be seen globally as error models, i.e., $\Delta\bar{\boldsymbol{x}}_k = \bar{\boldsymbol{x}}_k - \bar{\boldsymbol{x}}_k^{\text{tr}}$, it is possible to bound the maximum and minimum deviation between the states and reference. Therefore, consider $\Delta\bar{\boldsymbol{x}}_k \in \hat{\mathbb{E}} \subset \mathbb{R}^{n+m}$, where

   $$\hat{\mathbb{E}} = \{\Delta\bar{\boldsymbol{x}}_k \in \mathbb{R}^{n+m} : \Delta\bar{\boldsymbol{x}}^{min} \leq \Delta\bar{\boldsymbol{x}}_k \leq \Delta\bar{\boldsymbol{x}}^{max}, \forall k \in \mathbb{N}\}, \tag{3.23}$$

with $\Delta\bar{\boldsymbol{x}}^{min}$ and $\Delta\bar{\boldsymbol{x}}^{max}$ being, respectively, lower and upper bounds for the state error.

Using the prediction model (3.9), the state error constraints can be written as

$$\Delta\bar{\boldsymbol{x}}^{min} \leq \boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k + \boldsymbol{\mathcal{H}}_i \underrightarrow{\delta \boldsymbol{u}} \leq \Delta\bar{\boldsymbol{x}}^{max}. \tag{3.24}$$

Therefore, the constraints imposed to the state error can be represented by the following inequalities

$$\boldsymbol{\mathcal{H}}_i \underrightarrow{\delta \boldsymbol{u}} \leq \Delta\bar{\boldsymbol{x}}^{max} - \boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k, \tag{3.25}$$

$$-\boldsymbol{\mathcal{H}}_i \underrightarrow{\delta \boldsymbol{u}} \leq -\Delta\bar{\boldsymbol{x}}^{min} + \boldsymbol{\mathcal{P}}_i \Delta\bar{\boldsymbol{x}}_k. \tag{3.26}$$

3. Terminal set constraint:

A terminal set constraint $\boldsymbol{\Omega}$, which ensures that the last state error $\Delta\bar{\boldsymbol{x}}_{k+N_p}$ belongs to a control invariant set (see Definition 2.15), giving stability assurances for the system (Mayne et al., 2000), can be obtained using the concepts of invariant control sets together with the one-step set $Q(\boldsymbol{\Omega})$ (see Definition (2.18)). Thus, considering the model (3.8) and constraints (3.18) and (3.23), the one-step operator can be expressed as

$$Q(\boldsymbol{\Omega}) = \left\{ \Delta\bar{\boldsymbol{x}} \in \hat{\mathbb{E}} : \exists\, \delta\boldsymbol{u} \in \mathbb{V},\ \bar{\boldsymbol{A}}\Delta\bar{\boldsymbol{x}} + \bar{\boldsymbol{B}}\delta\boldsymbol{u} \in \boldsymbol{\Omega} \right\}. \tag{3.27}$$

Therefore, the maximal invariant set can be obtained by means of the iterative procedure:

(a) Initialization: $\boldsymbol{\Omega}_0 = \hat{\mathbb{E}} \cap \{\omega \in \mathbb{R}^{n+m} : \boldsymbol{K}\omega \in \mathbb{V}\}$.

(b) Iteration: $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k \cap Q(\boldsymbol{\Omega}_k)$.

(c) Terminal condition: stop when $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k$ or $\boldsymbol{\Omega}_{k+1} = \emptyset$. Define $\boldsymbol{\Omega} = \boldsymbol{\Omega}_\infty = \boldsymbol{\Omega}_k$.

The terminal constraint can be expressed as $\Delta\bar{\boldsymbol{x}}_{k+N_p} \in \boldsymbol{\Omega}$. However, in this form it cannot be used in the MPC optimization problem since it is not expressed as function of the decision variable $\delta\boldsymbol{u}$. Since $\boldsymbol{\Omega}$ is a bounded polyhedron, i.e., a polytope (see Definition 2.9), it is possible to obtain its H-representation, which is an unique representation constructed by means of the intersection of half-spaces (see Definition 2.7). Then, the terminal constraint can be rewritten as $\boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \Delta\bar{\boldsymbol{x}}_{k+N_p} \leq \boldsymbol{b}$. Using the prediction model (3.9) to obtain $\Delta\bar{\boldsymbol{x}}_{k+N_p}$, the terminal set constraint can be written as

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \left[\boldsymbol{p}_{i,\{N_p\}} \Delta\bar{\boldsymbol{x}}_k + \boldsymbol{h}_{i,\{N_p\}} \underrightarrow{\delta \boldsymbol{u}} \right] \leq \boldsymbol{b}, \tag{3.28}$$

where $\boldsymbol{p}_{i,\{N_p\}}$ is the last block entry of the matrix $\boldsymbol{\mathcal{P}}_i$, i.e., $\boldsymbol{p}_{i,\{N_p\}} = \bar{\boldsymbol{A}}^{N_p}$, and $\boldsymbol{h}_{i,\{N_p\}}$ is the last block line of matrix $\boldsymbol{\mathcal{H}}_i$, i.e., $\boldsymbol{h}_{i,\{N_p\}} = \begin{bmatrix} \bar{\boldsymbol{A}}^{N_p-1}\bar{\boldsymbol{B}} & \bar{\boldsymbol{A}}^{N_p-2}\bar{\boldsymbol{B}} & \cdots & \bar{\boldsymbol{A}}^{N_p-N_c}\bar{\boldsymbol{B}} \end{bmatrix}$.

Therefore, in terms of $\underset{\longrightarrow}{\delta \boldsymbol{u}}$ the terminal constraint can be expressed as

$$\boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \boldsymbol{h}_{i,\{N_p\}} \underset{\longrightarrow}{\delta \boldsymbol{u}} \leq \boldsymbol{b} - \boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \boldsymbol{p}_{i,\{N_p\}} \Delta \bar{\boldsymbol{x}}_k. \tag{3.29}$$

Finally, it is possible to write the system's constraints in the matricial form $\boldsymbol{\mathcal{M}} \underset{\longrightarrow}{\delta \boldsymbol{u}} \leq \boldsymbol{\mathcal{N}}$, with matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{N}}$ defined as

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix} \boldsymbol{C}_2 \\ -\boldsymbol{C}_2 \\ \boldsymbol{\mathcal{H}}_i \\ -\boldsymbol{\mathcal{H}}_i \\ \boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \boldsymbol{h}_{i,\{N_p\}} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{N}} = \begin{bmatrix} \boldsymbol{C}_1 \left(\boldsymbol{u}^{max} - \Delta \boldsymbol{u}_{k-1} \right) - \underset{\longrightarrow}{\boldsymbol{u}_k^{tr}} \\ -\boldsymbol{C}_1 \left(\boldsymbol{u}^{min} - \Delta \boldsymbol{u}_{k-1} \right) + \underset{\longrightarrow}{\boldsymbol{u}_k^{tr}} \\ \Delta \bar{\boldsymbol{x}}^{max} - \boldsymbol{\mathcal{P}}_i \Delta \bar{\boldsymbol{x}}_k \\ -\Delta \bar{\boldsymbol{x}}^{min} + \boldsymbol{\mathcal{P}}_i \Delta \bar{\boldsymbol{x}}_k \\ \boldsymbol{b} - \boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}} \boldsymbol{p}_{i,\{N_p\}} \Delta \bar{\boldsymbol{x}}_k \end{bmatrix}. \tag{3.30}$$

Therefore, the constrained LTI-MPC strategy can be generally defined through the optimization problem

$$\begin{aligned} \underset{\underset{\longrightarrow}{\delta \boldsymbol{u}}}{min} \quad & \tfrac{1}{2} \underset{\longrightarrow}{\delta \boldsymbol{u}}' \boldsymbol{\Lambda} \underset{\longrightarrow}{\delta \boldsymbol{u}} + \boldsymbol{f}' \underset{\longrightarrow}{\delta \boldsymbol{u}} + f_0, \\ \text{subject to} \quad & \boldsymbol{\mathcal{M}} \underset{\longrightarrow}{\delta \boldsymbol{u}} \leq \boldsymbol{\mathcal{N}}. \end{aligned} \tag{3.31}$$

3.2 Linear time-variant model predictive controller

This section deals with the design of the LTV-MPC presented in Santos et al. (2017b).

3.2.1 Problem statement

Consider the following finite-dimensional nonlinear system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{3.32}$$

where $\boldsymbol{x_k} \in \mathbb{X} \subset \mathbb{R}^n$ is the state vector, and $\boldsymbol{u_k} \in \mathbb{U} \subset \mathbb{R}^m$ is the input vector, with $\mathbb{X}$ and $\mathbb{U}$ being, respectively, the set of admissible states and inputs. Further, $\boldsymbol{f} : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{X}$ is the state-transition map defined by the nonlinear system model.

For the LTV-MPC, the control problem is defined as designing a controller able to perform trajectory tracking of the suspended load while stabilizing the UAV. Further, the controller must address the problems of performing yaw angle regulation and take-off and landing maneuvers. In this model predictive controller framework a time-variant discrete model with state-space representation is considered for state predictions and, aiming an improved performance, an incremental formulation is used. Besides of performing stably trajectory tracking, the controller is required to satisfy the system's constraints even in the presence of constant external disturbances and parametric uncertainties.

Consider the mapping of equation (3.32) from the continuous-time to the discrete-time domain after being linearized through first-order expansion in Taylor series around a

time-varying trajectory. Then, the error model can be expressed as

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k, \tag{3.33}$$

where $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{tr}$ and $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$ with the superscript $(\cdot)^{tr}$ denoting the desired trajectory, $\boldsymbol{\zeta}_k$ is the vector of time-varying parameters, and $\boldsymbol{A}(\boldsymbol{\zeta}_k)$ and $\boldsymbol{B}(\boldsymbol{\zeta}_k)$ are, respectively, the state and input Jacobian linear model matrices. Additionally, $\Delta \boldsymbol{x}_k \in \mathbb{E} \subset \mathbb{R}^n$, and the pair $(\boldsymbol{A}(\boldsymbol{\zeta}_k), \boldsymbol{B}(\boldsymbol{\zeta}_k))$ are assumed to be controllable for all $\boldsymbol{\zeta}_k \in \Delta \subset \mathbb{R}^v$, with $\mathbb{E}$ and Δ being compact sets.

Similarly to the previous section, the state-space model (3.33) gives the one-step ahead prediction and can be used recursively to obtain the prediction model considering a prediction horizon N_p and control horizon N_c. Thus, the two-step ahead prediction is given by

$$\begin{aligned} \Delta \boldsymbol{x}_{k+2} &= \boldsymbol{A}(\boldsymbol{\zeta}_{k+1})\Delta \boldsymbol{x}_{k+1} + \boldsymbol{B}(\boldsymbol{\zeta}_{k+1})\Delta \boldsymbol{u}_{k+1} \\ &= \boldsymbol{A}(\boldsymbol{\zeta}_{k+1})\left[\boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k\right] + \boldsymbol{B}(\boldsymbol{\zeta}_{k+1})\Delta \boldsymbol{u}_{k+1} \\ &= \boldsymbol{A}(\boldsymbol{\zeta}_{k+1})\boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{A}(\boldsymbol{\zeta}_{k+1})\boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k + \boldsymbol{B}(\boldsymbol{\zeta}_{k+1})\Delta \boldsymbol{u}_{k+1}. \end{aligned} \tag{3.34}$$

Continuing the recursive procedure, the n-step ahead prediction yields

$$\begin{aligned} \Delta \boldsymbol{x}_{k+n} = &\left(\prod_{\ell=1}^{n} \boldsymbol{A}(\boldsymbol{\zeta}_{k+n-\ell})\right)\Delta \boldsymbol{x}_k + \left(\prod_{\ell=1}^{n-1} \boldsymbol{A}(\boldsymbol{\zeta}_{k+n-\ell})\right)\boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k + \cdots + \\ &\boldsymbol{B}(\boldsymbol{\zeta}_{k+n-1})\Delta \boldsymbol{u}_{k+n-1}. \end{aligned} \tag{3.35}$$

Aiming to improved the control system's performance with additional integrators, the control increment $\delta \boldsymbol{u}_k \in \mathbb{V} \subset \mathbb{R}^m$ is selected to be the control input, with $\mathbb{V}$ being a compact set limiting the control increment. As made in (3.8), the discrete linearized system can be rewritten in the incremental form as

$$\underbrace{\begin{bmatrix} \Delta \boldsymbol{x}_{k+1} \\ \Delta \boldsymbol{u}_k \end{bmatrix}}_{\Delta \bar{\boldsymbol{x}}_{k+1}} = \underbrace{\begin{bmatrix} \boldsymbol{A}(\boldsymbol{\zeta}_k) & \boldsymbol{B}(\boldsymbol{\zeta}_k) \\ \boldsymbol{0} & \mathbb{I} \end{bmatrix}}_{\bar{\boldsymbol{A}}(\boldsymbol{\zeta}_k)} \underbrace{\begin{bmatrix} \Delta \boldsymbol{x}_k \\ \Delta \boldsymbol{u}_{k-1} \end{bmatrix}}_{\Delta \bar{\boldsymbol{x}}_k} + \underbrace{\begin{bmatrix} \boldsymbol{B}(\boldsymbol{\zeta}_k) \\ \mathbb{I} \end{bmatrix}}_{\bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k)} \delta \boldsymbol{u}_k. \tag{3.36}$$

Considering the case where $N_c < N_p$ and assuming $\delta \boldsymbol{u}_{k+i} = \boldsymbol{0}, \ \forall\, i \geq N_c$, the N_p-step ahead prediction yields

$$\begin{aligned} \Delta \bar{\boldsymbol{x}}_{k+N_p} = &\left(\prod_{\ell=1}^{N_p} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell})\right)\Delta \bar{\boldsymbol{x}}_k + \left(\prod_{\ell=1}^{N_p-1} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell})\right)\bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\delta \boldsymbol{u}_k + \cdots + \\ &\left(\prod_{\ell=1}^{N_p-N_c} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell})\right)\bar{\boldsymbol{B}}(\boldsymbol{\zeta}_{k+N_c-1})\delta \boldsymbol{u}_{k+N_c-1}. \end{aligned} \tag{3.37}$$

Thereby, it is possible to write the prediction model as

$$\underrightarrow{\Delta \bar{\boldsymbol{x}}} = \boldsymbol{\mathcal{P}}\Delta \bar{\boldsymbol{x}}_k + \boldsymbol{\mathcal{H}}\underrightarrow{\delta \boldsymbol{u}}, \tag{3.38}$$

where the matrices $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{H}}$ are given by

$$\boldsymbol{\mathcal{P}} = \left[\left(\prod_{\ell=1}^{1} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+1-\ell}) \right)^T \quad \left(\prod_{\ell=1}^{2} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+2-\ell}) \right)^T \quad \cdots \quad \left(\prod_{\ell=1}^{N_p} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right)^T \right]^T, \tag{3.39}$$

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k) & \mathbf{0} & \cdots & \mathbf{0} \\ \left(\prod_{\ell=1}^{2-1} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+2-\ell}) \right) \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k) & \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_{k+1}) & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\prod_{\ell=1}^{N_p-1} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k) & \left(\prod_{\ell=1}^{N_p-2} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_{k+1}) & \cdots & \left(\prod_{\ell=1}^{N_p-N_c} \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_{k+N_p-\ell}) \right) \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_{k+N_c-1}) \end{bmatrix}. \tag{3.40}$$

Note that the difference between the incremental prediction model when considering the LTV system (3.38) and the LTI system (3.9) lies on the necessity of on-line computation, since the latter varies due to the time-varying parameters $\boldsymbol{\zeta}_k$.

3.2.2 LTV-MPC optimization problem

Consider the standard quadratic cost function

$$\mathcal{J} = \sum_{i=0}^{N_p} \|\Delta \bar{\boldsymbol{x}}_{k+i}\|_{\boldsymbol{\mathcal{Q}}}^2 + \sum_{j=0}^{N_c-1} \|\delta \boldsymbol{u}_{k+j}\|_{\boldsymbol{\mathcal{R}}}^2, \tag{3.41}$$

where $\boldsymbol{\mathcal{Q}} > 0$ and $\boldsymbol{\mathcal{R}} > 0$ are, respectively, weighting matrices of states error and control effort. Note that, unlike equation (3.10) that considers a terminal cost function, here it is not used due to the time-variant nature of the system, which would require to solve an LMI problem similar to (3.17) at each sampling period k, making the computational cost of the controller prohibitive.

The cost function (3.41) can be written in the matricial form by means of the prediction model (3.38) as follows

$$\mathcal{J} = \left(\boldsymbol{\mathcal{H}} \underset{\rightarrow}{\delta \boldsymbol{u}} + \boldsymbol{\mathcal{P}} \Delta \bar{\boldsymbol{x}}_k \right)^T \boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}} \left(\boldsymbol{\mathcal{H}} \underset{\rightarrow}{\delta \boldsymbol{u}} + \boldsymbol{\mathcal{P}} \Delta \bar{\boldsymbol{x}}_k \right) + \underset{\rightarrow}{\delta \boldsymbol{u}}^T \boldsymbol{\Omega}_{\boldsymbol{\mathcal{R}}} \underset{\rightarrow}{\delta \boldsymbol{u}}, \tag{3.42}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}} = \mathbf{blkdiag}(\boldsymbol{\mathcal{Q}}, \cdots, \boldsymbol{\mathcal{Q}})$ and $\boldsymbol{\Omega}_{\boldsymbol{\mathcal{R}}} = \mathbf{blkdiag}(\boldsymbol{\mathcal{R}}, \cdots, \boldsymbol{\mathcal{R}})$ are block diagonal matrices.

Finally, the equation (3.42) can be rewritten in the canonical quadratic form as

$$\mathcal{J} = \frac{1}{2} \underset{\rightarrow}{\delta \boldsymbol{u}}^T \boldsymbol{\Lambda} \underset{\rightarrow}{\delta \boldsymbol{u}} + \boldsymbol{f}^T \underset{\rightarrow}{\delta \boldsymbol{u}} + f_0, \tag{3.43}$$

where

$$\begin{aligned}\boldsymbol{\Lambda} &= 2\left(\boldsymbol{\mathcal{H}}^T\boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}}\boldsymbol{\mathcal{H}} + \boldsymbol{\Omega}_{\boldsymbol{\mathcal{R}}}\right),\\ \boldsymbol{f}^T &= 2\left(\boldsymbol{\mathcal{P}}\Delta\bar{\boldsymbol{x}}\right)^T\boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}}\boldsymbol{\mathcal{H}},\\ f_0 &= \left(\boldsymbol{\mathcal{P}}\Delta\bar{\boldsymbol{x}}_k\right)^T\boldsymbol{\Omega}_{\boldsymbol{\mathcal{Q}}}\left(\boldsymbol{\mathcal{P}}\Delta\bar{\boldsymbol{x}}_k\right).\end{aligned}$$

Adding constraints on the objective variable $\delta\underrightarrow{\boldsymbol{u}}$, the most general optimization problem must be solved

$$\begin{aligned}\underset{\delta\underrightarrow{\boldsymbol{u}}}{min} \quad & \tfrac{1}{2}\delta\underrightarrow{\boldsymbol{u}}^T\boldsymbol{\Lambda}\delta\underrightarrow{\boldsymbol{u}} + \boldsymbol{f}^T\delta\underrightarrow{\boldsymbol{u}} + f_0,\\ \text{subject to} \quad & \boldsymbol{\mathcal{M}}\,\delta\underrightarrow{\boldsymbol{u}} \leq \boldsymbol{\mathcal{N}}.\end{aligned} \tag{3.44}$$

Likewise for the LTI-MPC, the constraints considered above can be used to bound the control signal amplitude avoiding saturation in the actuators and to limit the maximum state error, when mapped to the amplitude of the control increment $\delta\boldsymbol{u}$. Matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{N}}$ are then given by

$$\boldsymbol{\mathcal{M}} = \begin{bmatrix}\boldsymbol{C}_2\\ -\boldsymbol{C}_2\\ \boldsymbol{\mathcal{H}}\\ -\boldsymbol{\mathcal{H}}\end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{N}} = \begin{bmatrix}\boldsymbol{C}_1\left(\boldsymbol{u}^{max} - \Delta\boldsymbol{u}_{k-1}\right) - \underrightarrow{\boldsymbol{u}_k^{tr}}\\ -\boldsymbol{C}_1(\boldsymbol{u}^{min} - \Delta\boldsymbol{u}_{k-1}) + \underrightarrow{\boldsymbol{u}_k^{tr}}\\ \Delta\bar{\boldsymbol{x}}^{max} - \boldsymbol{\mathcal{P}}\Delta\bar{\boldsymbol{x}}_k\\ -\Delta\bar{\boldsymbol{x}}^{min} + \boldsymbol{\mathcal{P}}\Delta\bar{\boldsymbol{x}}_k\end{bmatrix}, \tag{3.45}$$

where matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ are defined as

$$\boldsymbol{C}_1 = \begin{bmatrix}\mathbb{I}\\ \mathbb{I}\\ \vdots\\ \mathbb{I}\end{bmatrix} \quad \text{and} \quad \boldsymbol{C}_2 = \begin{bmatrix}\mathbb{I} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbb{I} & \mathbb{I} & \cdots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbb{I} & \mathbb{I} & \cdots & \mathbb{I}\end{bmatrix}.$$

Summarizing, since the relation (3.19) still holds for the LTV-MPC, in order to compute the control action applied to the system, the optimization problem (3.44) with (3.45) must be solved.

3.3 Suspended load trajectory tracking control problem

In this section the trajectory tracking problem of a suspended load carried by a Tilt-rotor UAV is solved from the load's perspective through the previous designed LTI-MPC and LTV-MPC. The model (A.49), developed in the appendix A, is used to derive the linearized models (3.2) and (3.33).

In order to compute both, time-invariant and time-variant linearized models, to design the predictive controllers, the equilibrium points of the system must be found. Therefore,

consider the vehicle in hovering without any external disturbances ($\boldsymbol{d} = \mathbf{0}$). The equilibrium point can be obtained after solving $\dot{\boldsymbol{x}}_{\mathcal{L}} = \boldsymbol{\varphi}_{\mathcal{L}}(\boldsymbol{x}_{\mathcal{L}}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{L}}^T & \ddot{\boldsymbol{q}}_{\mathcal{L}}^T \end{bmatrix}^T = \mathbf{0}$, which leads to the system of algebraic equations

$$\boldsymbol{\vartheta}_{\mathcal{L}}\left(\boldsymbol{q}_{\mathcal{L}}^{eq}\right) - \boldsymbol{G}_{\mathcal{L}}\left(\boldsymbol{q}_{\mathcal{L}}^{eq}\right) = \mathbf{0}, \tag{3.46}$$

where $\boldsymbol{q}_{\mathcal{L}}^{eq} = [x_{\mathcal{L}}^{eq}\ y_{\mathcal{L}}^{eq}\ z_{\mathcal{L}}^{eq}\ \phi_{\mathcal{L}}^{eq}\ \theta_{\mathcal{L}}^{eq}\ \psi_{\mathcal{L}}^{eq}\ \gamma_1^{eq}\ \gamma_2^{eq}\ \alpha_R^{eq}\ \alpha_L^{eq}]^T$.

Nevertheless, (3.46) has more variables than equations meaning that an infinity set of real numbers can solve the algebraic problem. To overcome this issue, let the states $x_{\mathcal{L}}$, $y_{\mathcal{L}}$, and $z_{\mathcal{L}}$ assume any value and let $\psi_{\mathcal{L}} = 0$.[3] Thus, considering the physical parameters defined on table A.1, the equilibrium values for the states and inputs are given by

$$\begin{aligned} &\phi^{eq} = 0,\ \theta^{eq} = 0,\ \gamma_1^{eq} = 0.0001563,\ \gamma_2^{eq} = 0.0287134,\ \alpha_R^{eq} = 0.0288375,\ \alpha_L^{eq} = 0.0283718, \\ &f_R^{eq} = 9.7903455,\ f_L^{eq} = 9.8252665,\ \tau_{\alpha_R}^{eq} = 0,\ \tau_{\alpha_L}^{eq} = 0. \end{aligned} \tag{3.47}$$

For discretization of the linearized Jacobians, it is assumed a sampling time of $T_s = 12$ ms. Furthermore, the saturation level of the Tilt-rotor UAV actuators and the maximum state error allowed, used in (3.30) for the LTI-MPC and in (3.45) for the LTV-MPC, are

$$\begin{array}{llll} \Delta x = [-1, 1], & \Delta y = [-1, 1], & \Delta z = [-1, 1], & \\ \Delta\phi = [-0.8, 0.8], & \Delta\theta = [-0.8, 0.8], & \Delta\psi = [-0.8, 0.8], & \\ \Delta\gamma_1 = [-0.8, 0.8], & \Delta\gamma_2 = [-0.8, 0.8], & \Delta\alpha_R = [-0.8, 0.8], & \Delta\alpha_L = [-0.8, 0.8], \\ f_R = [0, 15], & f_L = [0, 15], & \tau_{\alpha_R} = [-2, 2], & \tau_{\alpha_L} = [-2, 2], \end{array} \tag{3.48}$$

where the error limitations were chosen regarding measurement errors and disturbances affecting the system, and the actuators' bounds depends on physical constraints.

The Bryson's rule (Johnson & Grimble, 1987) was used as starting point to synthesize the MPC's weighting matrices used in (3.10) and (3.41). The basic idea of Bryson's method is to scale all variables by dividing each weight matrix diagonal entries by the square of the maximum variation of the variable associated with that entry. Further, the diagonal numerators can be modified regarding the design goals. Thus, following this procedure, the weight matrices are given by

$$\begin{aligned} \boldsymbol{\mathcal{Q}} = \operatorname{diag}\Bigg(&\frac{40}{2^2}, \frac{40}{2^2}, \frac{20}{2^2}, \frac{5}{(\pi/2)^2}, \frac{5}{(\pi/2)^2}, \frac{10}{(\pi)^2}, \frac{10}{(\pi/2)^2}, \frac{10}{(\pi/2)^2}, \frac{0.1}{(\pi/2)^2}, \frac{0.1}{(\pi/2)^2}, \frac{10}{2^2}, \frac{10}{2^2}, \frac{5}{2^2}, \\ &\frac{1}{(\pi/3)^2}, \frac{1}{(\pi/3)^2}, \frac{1}{(\pi/4)^2}, \frac{5}{(3\pi)^2}, \frac{5}{(3\pi)^2}, \frac{0.1}{(3\pi)^2}, \frac{0.1}{(3\pi)^2}, 40, 40, 40, 20, \\ &\frac{40}{(15 - f_R^{eq})^2}, \frac{40}{(15 - f_L^{eq})^2}, \frac{20}{(2 - \tau_{\alpha_R})^2}, \frac{20}{(2 - \tau_{\alpha_L}^{eq})^2} \Bigg), \end{aligned} \tag{3.49}$$

[3] From now on, the subscript $\mathcal{L}$ used to differ the system from the load's perspective from the UAV's perspective will be omitted since this chapter deals only with the first one.

$$\boldsymbol{\mathcal{R}} = \text{diag}\left(\frac{200}{(15 - f_R^{eq})^2}, \frac{200}{(15 - f_L^{eq})^2}, \frac{1000}{(2 - \tau_{\alpha_R}^{eq})^2}, \frac{1000}{(2 - \tau_{\alpha_L}^{eq})^2}\right). \tag{3.50}$$

Furthermore, the prediction and control horizons, chosen considering the trade-off between good performance and small computational cost, are $N_p = 100$ and $N_c = 10$. All the above defined parameters are common to both controllers in order to compare their performances when addressing the suspended load trajectory tracking problem.

3.3.1 Desired trajectory

To explore the controllers' capabilities, two trajectories to be tracked by the suspended load, composed of several interconnected paths described by polynomial and/or sinusoidal functions, are proposed. For the first proposed trajectory, the vehicle starts in hovering and tracks a square-like trajectory without any yaw movement during the execution (see Figure 3.1). The main purpose of this trajectory is to compare fairly the LTI-MPC and the LTV-MPC by means of the performance indexes Mean Squared Error (MSE) and Integrated Absolute Derivative of Control Signal (IADU). For the second trajectory, the Tilt-rotor UAV starts with vertical take-off, followed by a straight line tracking with changes in direction together with yaw movements, and ends with vertical landing (see Figure 3.2). The main goal of this trajectory is to show the ability of the LTV-MPC to deal with yaw movements and take-off and landing maneuvers, where the length of the rope and total mass of the vehicle vary, without relying only on the robustness given by the feedback nature of the MPC.

Moreover, to evaluate the disturbance compensation of the proposed strategy, external forces are applied to the suspended load when performing both trajectories. Figure 3.3 shows the disturbance profile for the desired trajectory, which may represent sustained wind gusts affecting the load. The magnitude of the disturbances may seem low at a first glance, however the mass of the load is very small (see Table A.1). Additionally, to better evaluate the behavior of the proposed control strategies in the presence of uncertainties, measurement's noise is considered and assumed to have Gaussian probability distribution with zero mean and measurement error defined as three times the standard deviation. For simulation purposes, position errors are ± 0.15 [m], angular position errors are ± 0.02 [rad], velocity errors are ± 0.01 [m/s], and angular velocity errors are ± 0.002 [rad/s].

3.3.2 Linear time-invariant MPC

Aiming to obtain the discrete linear time-invariant state-space model in order to construct the prediction model (3.9), the equations of motion (A.49) must be linearized around an equilibrium point. Let $\boldsymbol{x}^{eq}$ and $\boldsymbol{u}^{eq}$ denote, respectively, the equilibrium state and inputs able to maintain the system in hovering. Then, linearizing the state-space equations (A.49)

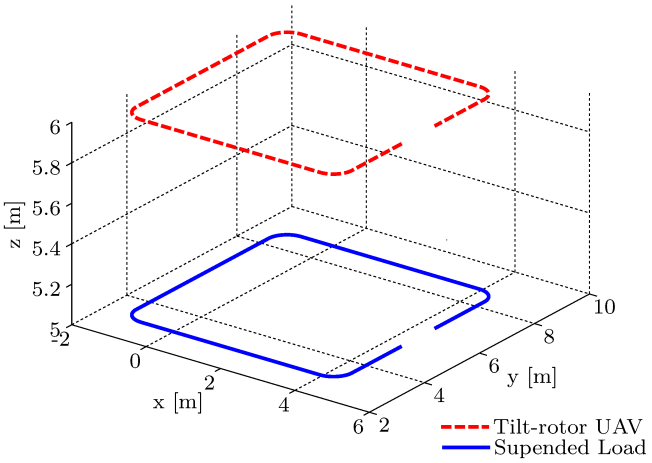


Figure 3.1: Three-dimensional representation of the first proposed trajectory.

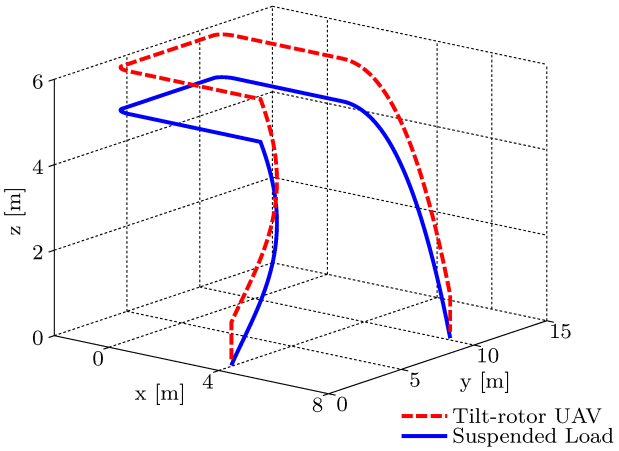


Figure 3.2: Three-dimensional representation of the second proposed trajectory.

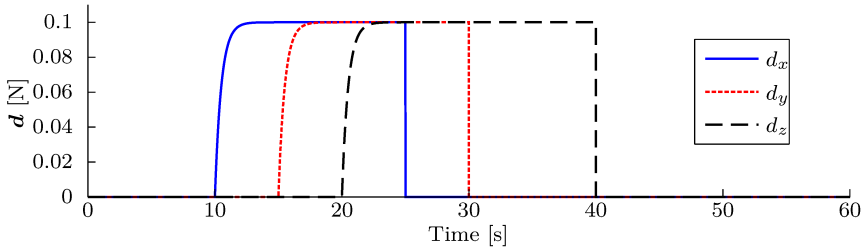


Figure 3.3: Profile of the disturbance forces applied to the load and expressed in the inertial frame $\mathcal{I}$.

around these points, through first-order expansion in Taylor series, yields to

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{B}\Delta\boldsymbol{u}, \tag{3.51}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{eq}$, $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}^{eq}$, and

$$\boldsymbol{A} = \frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{x}}\bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{eq}\\ \boldsymbol{u}=\boldsymbol{u}^{eq}}} \in \mathbb{R}^{20\times 20}, \qquad \boldsymbol{B} = \frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{u}}\bigg|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{eq}\\ \boldsymbol{u}=\boldsymbol{u}^{eq}}} \in \mathbb{R}^{20\times 4}, \tag{3.52}$$

being $\boldsymbol{x}^{eq} = [(\boldsymbol{q}^{eq})^T \ (\dot{\boldsymbol{q}}^{eq})^T]^T$ and $\boldsymbol{u}^{eq} = [f_R^{eq} \ f_L^{eq} \ \tau_{\alpha_R}^{eq} \ \tau_{\alpha_L}^{eq}]^T$, with the equilibrium values obtained in (3.47) and $\boldsymbol{q}^{eq} = [0 \ 0 \ 0 \ \phi^{eq} \ \theta^{eq} \ 0 \ \gamma_1^{eq} \ \gamma_2^{eq} \ {\alpha_{\mathrm{R}}}^{eq} \ {\alpha_{\mathrm{L}}}^{eq}]^T$.

To improve the trajectory tracking of the regulated variables and provide constant disturbance and parametric uncertainties rejection, the error state vector $\Delta\boldsymbol{x}$ is augmented with integral actions (Raffo et al., 2010), yielding to

$$\Delta\hat{\boldsymbol{x}} = \begin{bmatrix} \Delta\boldsymbol{x} \\ \int (\boldsymbol{\xi} - \boldsymbol{\xi}^{tr}) \\ \int (\psi - \psi^{tr}) \end{bmatrix} \in \mathbb{R}^{24}, \tag{3.53}$$

whose dynamics are given by

$$\Delta\dot{\hat{\boldsymbol{x}}}(t) = \underbrace{\left[\begin{array}{cccccc|c|c} \multicolumn{6}{c|}{\multirow{2}{*}{$\boldsymbol{A}$}} & & \boldsymbol{0}_{20\times4} \\ & & & & & & \multirow{4}{*}{$\boldsymbol{0}_{4\times14}$} & \multirow{4}{*}{$\boldsymbol{0}_{4\times4}$} \\ 1 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 1 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & \end{array}\right]}_{\hat{\boldsymbol{A}}} \Delta\hat{\boldsymbol{x}}(t) + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{0}_{4\times4} \end{bmatrix}}_{\hat{\boldsymbol{B}}} \Delta\boldsymbol{u}(t). \tag{3.54}$$

To obtain a discrete prediction model using the incremental form (3.8) and, thereafter, improve performance provided by the four additional input integrators added to the closed-loop system (Rossiter, 2013), it is necessary to map the model (3.54) from the continuous-time to the discrete-time domain, which yields to

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}_d \Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}_d \Delta\boldsymbol{u}_k, \tag{3.55}$$

being the matrices $\hat{\boldsymbol{A}}_d$ and $\hat{\boldsymbol{B}}_d$ obtained after discretizing the model using a zero-order hold with sampling time T_s. Thereafter, the incremental model can be obtained extending the system (3.55) as in (3.8), yielding to

$$\Delta\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{A}} \Delta\bar{\boldsymbol{x}}_k + \bar{\boldsymbol{B}} \delta\boldsymbol{u}_k. \tag{3.56}$$

The trajectory to be tracked by the suspended load and its control signal are defined as

$$\boldsymbol{x}_k^{tr} = [(\boldsymbol{q}_k^{tr})^T \ (\dot{\boldsymbol{q}}_k^{tr})^T]^T, \tag{3.57}$$

$$\boldsymbol{u}_k^{\mathrm{tr}} = \boldsymbol{\vartheta}^{+} \left[\boldsymbol{M}\ddot{\boldsymbol{q}}_k^{\mathrm{tr}} + \boldsymbol{C}\dot{\boldsymbol{q}}_k^{\mathrm{tr}} + \boldsymbol{G}\right], \tag{3.58}$$

where $(\cdot)^+$ denotes the left pseudoinverse, and $\boldsymbol{q}_k^{\mathrm{tr}}$, $\dot{\boldsymbol{q}}_k^{\mathrm{tr}}$ and $\ddot{\boldsymbol{q}}_k^{\mathrm{tr}}$ are provided reference signals with $\boldsymbol{q}_k^{tr} = [x_k^{tr} \ y_k^{tr} \ z_k^{tr} \ \phi^{eq} \ \theta^{eq} \ \psi^{tr} \ \gamma_1^{eq} \ \gamma_2^{eq} \ {\alpha_{\mathrm{R}}}^{eq} \ {\alpha_{\mathrm{L}}}^{eq}]^T$. Notice that $\boldsymbol{u}_k^{tr}$, since it is computed using a left pseudoinverse, will be an exact solution to the dynamic equations (A.49) $\forall k \in \mathbb{N}$ only if the trajectory is feasible.

Finally, having the model (3.56), the LTI-MPC algorithm for trajectory tracking of the suspended load carried by a Tilt-rotor UAV can be described in general terms by the Algorithm 3.1.

Algorithm 3.1 Linear time-invariant MPC algorithm

1: Obtain $\boldsymbol{\mathcal{P}}_i$ and $\boldsymbol{\mathcal{H}}_i$ through (3.9) using (3.56).
2: Compute $\boldsymbol{P}$ through (3.17) using the model (3.56) and matrices $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{R}}$ defined, respectively, in (3.49) and (3.50).
3: Calculate $\boldsymbol{\Omega}$ by iterating the one-step operator defined at (3.27) and find its H-representation $\boldsymbol{\mathcal{H}}_{\boldsymbol{\Omega}}\Delta\bar{\boldsymbol{x}}_{k+N_p} \leq \boldsymbol{b}$.
4: Set $\Delta\boldsymbol{u}_0 = 0$.
5: **procedure** LTI-MPC($\boldsymbol{x}_k$, $\Delta\boldsymbol{u}_{k-1}$, $\overrightarrow{\boldsymbol{q}^{tr}}$, $\overrightarrow{\dot{\boldsymbol{q}}^{tr}}$, $\overrightarrow{\ddot{\boldsymbol{q}}^{tr}}$)
6: Obtain the vector $\Delta\bar{\boldsymbol{x}}_k$.
7: Compute $\overrightarrow{\boldsymbol{u}^{tr}}$ using (3.58) from k up to $k + N_c$.
8: Obtain $\boldsymbol{\Lambda}$, $\overrightarrow{\boldsymbol{f}^T}$, and f_0 from (3.12).
9: Construct the matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{N}}$ as in (3.30).
10: Solve the optimization problem (3.31) to obtain $\overrightarrow{\delta\boldsymbol{u}}$.
11: Compute $\boldsymbol{u}_k = \boldsymbol{u}_k^{tr} + \Delta\boldsymbol{u}_{k-1} + \delta\boldsymbol{u}_k$.
12: Set $\Delta\boldsymbol{u}_{k-1} = \Delta\boldsymbol{u}_k$.
13: **return** $\boldsymbol{u}_k$ and $\Delta\boldsymbol{u}_{k-1}$.
14: **end procedure**

3.3.3 Linear time-variant MPC

Similarly, aiming to obtain the discrete linear time-varying state-space model in order to construct the prediction model (3.38), the equations of motion (A.49) must be linearized around a time-varying trajectory. Additionally, due to limited computational resources, this process needs to be done with most of the physical parameters numerically evaluated. However, it is possible to let some physical parameters as variables in a way that they will appear in the linearized Jacobians after finished the linearization process.

Let $\boldsymbol{x}^{tr}$ and $\boldsymbol{u}^{tr}$ denote, respectively, the state vector trajectory and the control inputs able to drive the system along this trajectory.Then, by linearizing the state-space equations (A.49) around these trajectories using the first-order expansion in Taylor series, the LTV linearized model can be expressed as

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}(t)\Delta\boldsymbol{x} + \boldsymbol{B}(t)\Delta\boldsymbol{u}, \tag{3.59}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{tr}$, $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}^{tr}$, and

$$\boldsymbol{A}(t) = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{tr}\\ \boldsymbol{u}=\boldsymbol{u}^{tr}}} \in \mathbb{R}^{20\times 20}, \quad \boldsymbol{B}(t) = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{tr}\\ \boldsymbol{u}=\boldsymbol{u}^{tr}}} \in \mathbb{R}^{20\times 4}. \tag{3.60}$$

In this work, the trajectory values for $\boldsymbol{x}$ and $\boldsymbol{u}$ are given by

$$\boldsymbol{x}^{tr} = [(\boldsymbol{q}^{tr})^T \ (\dot{\boldsymbol{q}}^{tr})^T]^T, \tag{3.61}$$

$$\boldsymbol{u}^{\text{tr}} = \boldsymbol{\vartheta}^{+} \left[\boldsymbol{M}\ddot{\boldsymbol{q}}^{\text{tr}} + \boldsymbol{C}\dot{\boldsymbol{q}}^{\text{tr}} + \boldsymbol{G}\right], \tag{3.62}$$

where $\boldsymbol{q}^{\text{tr}}$, $\dot{\boldsymbol{q}}^{\text{tr}}$ and $\ddot{\boldsymbol{q}}^{\text{tr}}$ are provided reference signals with $\mathrm{l}\boldsymbol{q}^{tr} = [x^{tr}(t)\ y^{tr}(t)\ z^{tr}(t)\ \phi^{eq}\ \theta^{eq}\ \psi^{tr}(t)\ \gamma_1^{eq}\ \gamma_2^{eq}\ {\alpha_{\mathrm{R}}}^{eq}\ {\alpha_{\mathrm{L}}}^{eq}]^T$.

Therefore, by linearizing the system using (3.60) with the trajectories defined in (3.61) and (3.62) added to the rope's length $l(t)$ as a time-varying parameter, the linearized jacobians are $\boldsymbol{A}(\boldsymbol{\zeta}(t))$ and $\boldsymbol{B}(\boldsymbol{\zeta}(t))$, where $\boldsymbol{\zeta}(t) = [x^{tr}(t)\ y^{tr}(t)\ z^{tr}(t)\ \psi^{tr}(t)\ \dot{x}^{tr}(t)\ \dot{y}^{tr}(t)\ \dot{z}^{tr}(t)\ \dot{\psi}^{tr}(t)\ \ddot{x}^{tr}(t)\ \ddot{y}^{tr}(t)\ \ddot{z}^{tr}(t)\ \ddot{\psi}^{tr}(t)\ l(t)]^T$ is the vector of time-varying parameters.

Similarly to the previous section, the state vector $\Delta\boldsymbol{x}$ can be augmented with integral actions seeking improved tracking performance. Therefore, the augmented state vector can be expressed as

$$\Delta\hat{\boldsymbol{x}} = \begin{bmatrix} \Delta\boldsymbol{x} \\ \int (\boldsymbol{\xi} - \boldsymbol{\xi}^{tr}) \\ \int (\psi - \psi^{tr}) \end{bmatrix} \in \mathbb{R}^{24}, \tag{3.63}$$

whose dynamics are given by

$$\Delta\dot{\hat{\boldsymbol{x}}}(t) = \underbrace{\left[\begin{array}{cccccc|c|c} \multicolumn{6}{c|}{\boldsymbol{A}(\boldsymbol{\zeta}(t))} & & \boldsymbol{0}_{20\times 4} \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & \multirow{4}{*}{$\boldsymbol{0}_{4\times 14}$} & \multirow{4}{*}{$\boldsymbol{0}_{4\times 4}$} \\ 0 & 1 & 0 & 0 & 0 & 0 & & \\ 0 & 0 & 1 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & \end{array}\right]}_{\hat{\boldsymbol{A}}(\boldsymbol{\zeta}(t))} \Delta\hat{\boldsymbol{x}}(t) + \underbrace{\begin{bmatrix} \boldsymbol{B}(\boldsymbol{\zeta}(t)) \\ \boldsymbol{0}_{4\times 4} \end{bmatrix}}_{\hat{\boldsymbol{B}}(\boldsymbol{\zeta}(t))} \Delta\boldsymbol{u}(t). \tag{3.64}$$

In order to represent the system in its incremental form, the model (3.64) can be expressed as

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}_d(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}_d(\boldsymbol{\zeta}_k)\Delta\boldsymbol{u}_k, \tag{3.65}$$

being the matrices $\hat{\boldsymbol{A}}_d(\boldsymbol{\zeta}_k)$ and $\hat{\boldsymbol{B}}_d(\boldsymbol{\zeta}_k)$ obtained after discretizing the model using a zero-order hold with sampling time T_s. Thereafter, the incremental model can be obtained extending the system (3.65) as in (3.8), yielding to

$$\Delta\bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\bar{\boldsymbol{x}}_k + \bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\delta\boldsymbol{u}_k. \tag{3.66}$$

Finally, having the model (3.66), the LTV-MPC for trajectory tracking of the suspended load carried by a Tilt-rotor UAV able to cope with yaw angle regulation and take-off and landing maneuvers can be described in general terms by the Algorithm 3.2.

Algorithm 3.2 Linear time-variant MPC algorithm

Set $\Delta \boldsymbol{u}_0 = 0$.
procedure LTV-MPC($\boldsymbol{x}_k$, $\Delta \boldsymbol{u}_{k-1}$, $\overrightarrow{\boldsymbol{q}^{tr}}$, $\overrightarrow{\dot{\boldsymbol{q}}^{tr}}$, $\overrightarrow{\ddot{\boldsymbol{q}}^{tr}}$)
    Obtain $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{H}}$ through (3.39) and (3.40) using the model (3.66).
    Obtain the vector $\Delta \bar{\boldsymbol{x}}_k$.
    Compute $\overrightarrow{\boldsymbol{u}^{tr}}$ using (3.62) from k up to $k + N_c$.
    Obtain $\boldsymbol{\Lambda}$, $\overrightarrow{\boldsymbol{f}^T}$, and f_0 using (3.43).
    Construct the matrices $\boldsymbol{\mathcal{M}}$ and $\boldsymbol{\mathcal{N}}$ as in (3.45).
    Solve the optimization problem (3.44) to obtain $\overrightarrow{\delta \boldsymbol{u}}$.
    Compute $\boldsymbol{u}_k = \boldsymbol{u}_k^{tr} + \Delta \boldsymbol{u}_{k-1} + \delta \boldsymbol{u}_k$.
    Set $\Delta \boldsymbol{u}_{k-1} = \Delta \boldsymbol{u}_k$.
    return $\boldsymbol{u}_k$ and $\Delta \boldsymbol{u}_{k-1}$.
end procedure

3.3.4 Simulation results

This subsection presents the numerical simulation results obtained with the proposed controllers when performing the trajectories previously described. Three simulations scenarios are carried out: (i) the first one uses the trajectory presented in Figure 3.1 in order to compare the performance of the LTI-MPC and the LTV-MPC[4]; (ii) the second scenario considers the trajectory of Figure 3.2 aiming to show the advantages of working with the LTV-MPC, mainly due to its capacity to cope with yaw movements and to deal with take-off and landing maneuvers; and (iii) the third one shows, while performing the trajectory 3.1, how the computational cost can be reduced when using the LTI-MPC with the inclusion of the terminal region and terminal cost into the optimization problem.

The simulations have been carried out using the MATLAB/Simulink® environment. A detailed analysis of the control systems' performance is provided when solving the trajectory tracking problem of a suspended load carried by a Tilt-rotor UAV, as well as, a comparative analysis between both controllers.

Scenario 1

The trajectories performed by the Tilt-rotor UAV and the suspended load using the LTI-MPC and LTV-MPC are shown in Figures 3.4 and 3.5, while the tracking error is illustrated in Figure 3.6. The trajectory tracking was performed successfully for both controllers with similar performance along the proposed trajectory. A slight difference that stands out is related with the yaw movement regulation, for which the LTV-MPC achieve better results, and the altitude regulation, where the LTI-MPC performed better. The first difference can be explained by the inclusion of the yaw angle as time-varying parameter of the controller, while the second one is due to the larger domain of attraction

[4]For the sake of comparison, in this scenario the measurement's noise is removed aiming at a better visualization of the differences between the proposed controllers.

of the LTV-MPC. The enlargement of its domain of attraction allows the control system to perform trajectory tracking in a bigger operation range when compared to the LTI-MPC. On the other hand, in a small range, the LTV-MPC worsen its performance front to the LTI-MPC since the latter was design specifically for small ranges. On other words, the LTV-MPC can work with sufficiently good performance in an enlarged operational range while the LTI-MPC has better performance for specific ones, but cannot ensure performance or even stability for large ranges.

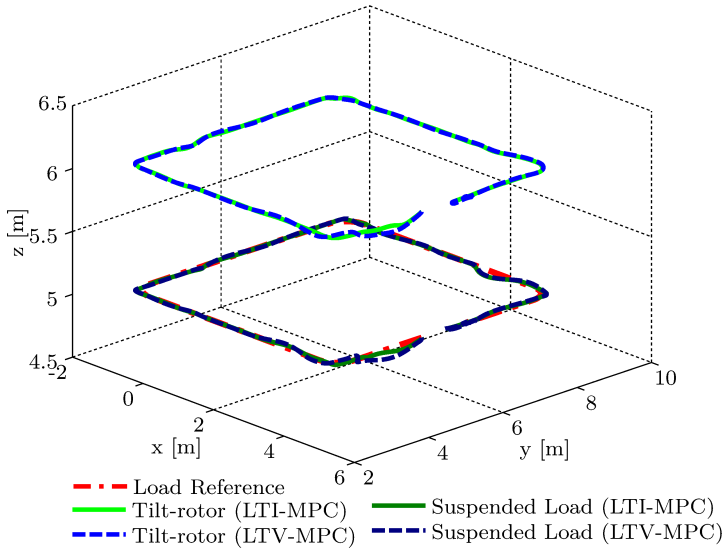


Figure 3.4: Trajectory tracking using LTI-MPC and LTV-MPC when performing the trajectory presented in Figure 3.1.

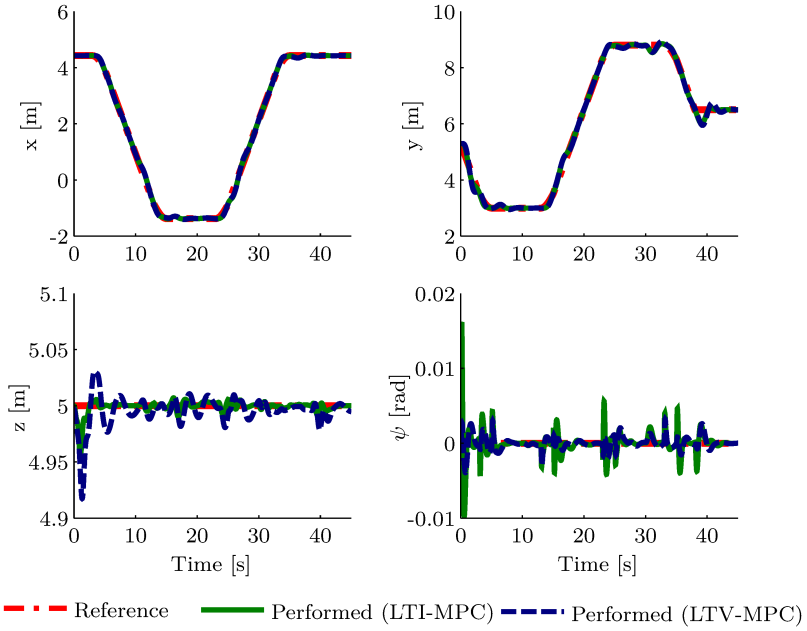


Figure 3.5: Time evolution of the regulated variables (x, y, z, ψ) and their desired trajectories.

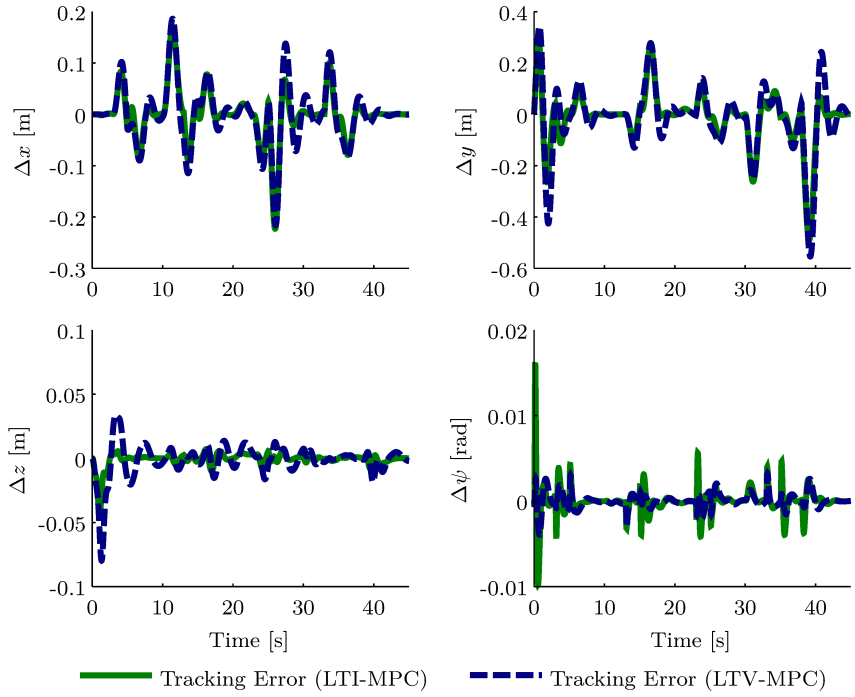


Figure 3.6: Time evolution of the tracking errors of the regulated variables (x, y, z, ψ).

Figure 3.7 shows the time evolution of the remaining degrees of freedom for the LTI-MPC and LTV-MPC, which are kept stable throughout the trajectory. Since the aircraft's behavior is described implicitly by those variables, one can conclude that the UAV was stabilized along the trajectory.

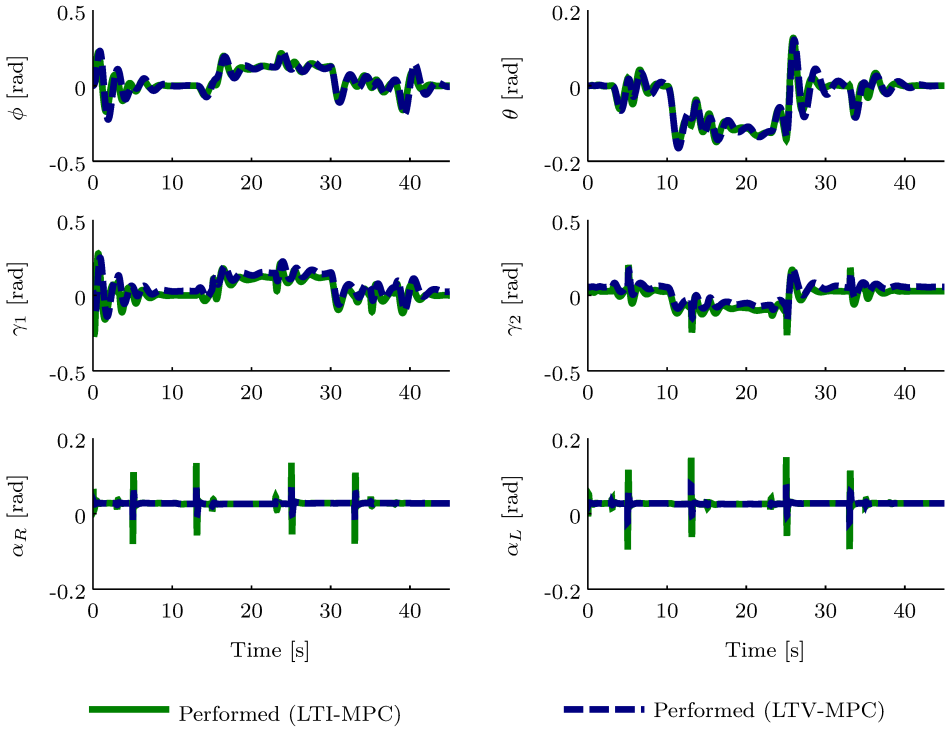


Figure 3.7: Time evolution of the remaining degrees of freedom.

Figure 3.8 shows the actuator signals generated by the MPCs. For both cases, the control signals did not violate the actuators' constraints despite the LTI-MPC control signals presented some peaks when executing direction changes, i.e., the corners of the square-like trajectory. In fact, it is interesting to notice that the for a Tilt-rotor UAV designed considering the parameters of Table A.1 and the actuators limits (3.47), the vehicle will mainly flies using its mid range span of actuation. Thus, despite the MPC constraints are active constraints, for the considered UAV and simulation scenario the controller does not need to use the actuators close to their limit ranges.

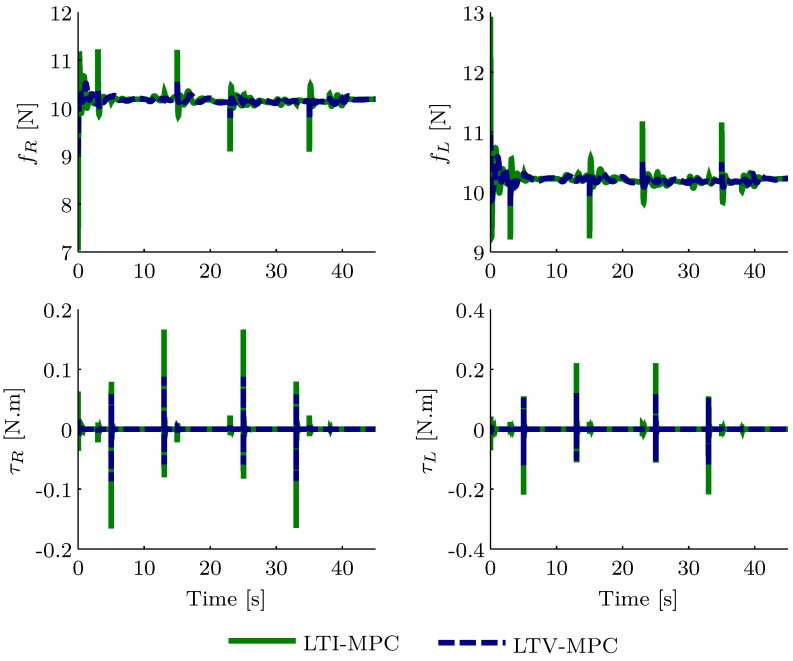


Figure 3.8: Applied thrusts and torques to the Tilt-rotor UAV.

In order to compare the controllers' performance, the MSE and IADU performance indexes for discrete systems are used

$$\text{MSE}_i = \frac{1}{n}\sum_{k=0}^{n}\left(\boldsymbol{x}\left[i\right]_k - \boldsymbol{x}\left[i\right]_k^{tr}\right)^2, \qquad \text{IADU}_i = \sum_{k=0}^{n}\left[\left|\boldsymbol{u}[i]_{k+2} - \boldsymbol{u}[i]_{k+1}\right| - \left|\boldsymbol{u}[i]_{k+1} - \boldsymbol{u}[i]_k\right|\right], \quad (3.67)$$

where $\boldsymbol{x}[i]$ and $\boldsymbol{u}[i]$ are, respectively, the *i-th* position of the state and control vector, and the rectangle rule is used as integral approximation.

Table 3.1 shows the performance indexes for both controllers, which corroborate the analysis previously made. Overall, the LTI-MPC has better performance when executing trajectory tracking control in the vicinity of the linearization points. The only exception occurs to the yaw movement for which the LTV-MPC has better performance despite being kept equal to zero along the trajectory. When it comes to control effort, the LTV-MPC presents signals smoother with smaller peaks.

Table 3.1: Performance indexes, obtained through the numerical results.

	LTI-MPC	LTV-MPC
States	MSE	
x	0.00242810	0.00335758
y	0.01144221	0.01768196
z	0.00003113	0.00017094
ψ	0.00000264	0.00000076
Inputs	IADU	
f_R	24.2534	9.16095
f_L	23.7565	9.4805
τ_{α_R}	2.6999	1.3370
τ_{α_L}	3.5426	2.0421

Scenario 2

The obtained results when the suspended load is carried by the Tilt-rotor UAV performing the trajectory presented in Figure 3.2 with the LTV-MPC are shown in Figures 3.9 and 3.10. Figure 3.11 shows the tracking error of the regulated states x, y, z, and ψ. The trajectory tracking was performed successfully from the take-off to landing, and throughout the different paths that compose the trajectory, including yaw angle regulation. For this scenario, it was not possible to compare the results of the LTV-MPC with the LTI-MPC since the time-invariant controller became unstable during the execution of the proposed trajectory. The main reason of the unstable behavior can be explained by the inability of the LTI-MPC to cope with rope's length variation at take-off and landing and, mainly, due to its inability to perform yaw movements regulation.

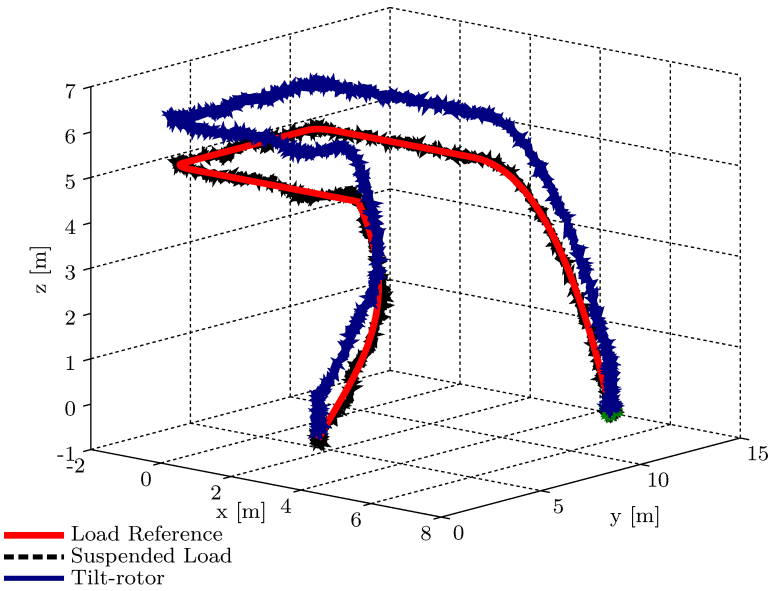


Figure 3.9: Trajectory tracking using LTV-MPC when performing the trajectory presented in Figure 3.2.

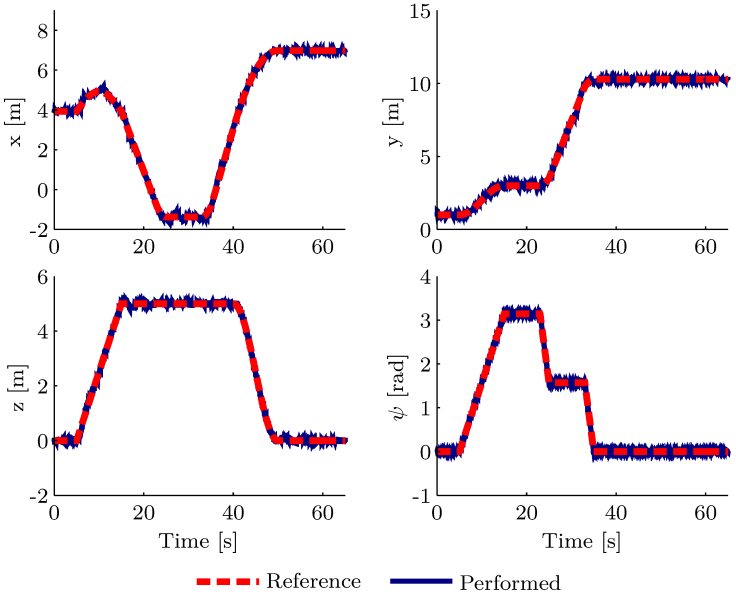


Figure 3.10: Time evolution of the regulated variables (x, y, z, ψ).

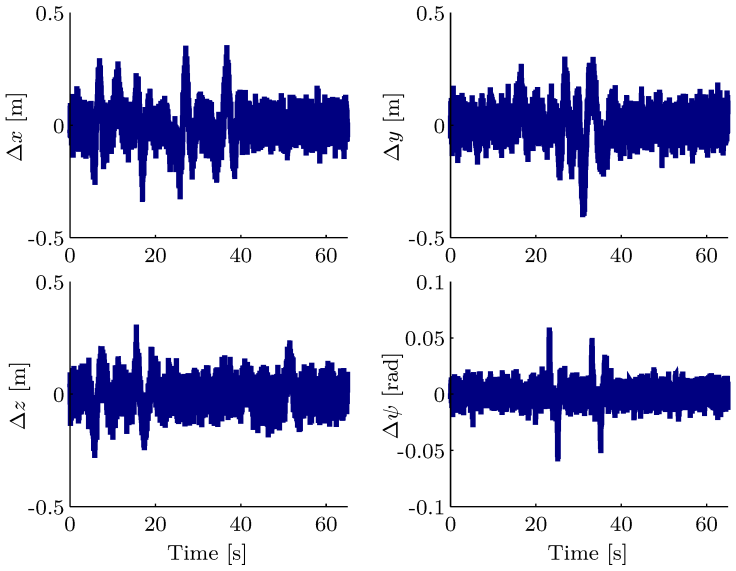


Figure 3.11: Time evolution of the tracking errors.

During the vertical take-off, the aircraft starts to fly while the load remains in the ground. Only when the distance between the aircraft and the floor is greater than the rope's length, the Tilt-rotor UAV starts to carry the load. Vice-versa, in the landing maneuver the aircraft flies free of load once it has touched the ground. Figure 3.12 highlights this behavior in the first 5 and last 10 seconds of simulation. At the beginning of the simulation, the rope's length increases until it reaches its maximum value, then, the aircraft starts to carry the suspended load and the total mass increases. Likewise, during the landing, last 10 seconds, the load touches the ground and the rope's length decreases until the aircraft landing finishes. Also, Figure 3.12 shows the total mass reduction when the load touches the ground. The controller was capable of dealing with this problem due

to the incorporation of l_k in the time-varying parameters vector $\boldsymbol{\zeta}_k$ and to the model-based nature of predictive controllers. The load's mass was not incorporated in $\boldsymbol{\zeta}_k$ for two main reasons: (i) in order to consider the mass variation as a time-varying parameter in the same way that l_k was considered, the load's mass needs to be estimated or informed to the controller before the flight starts; (ii) it is reasonable to assume that the relation between the UAV's mass and the load's mass is small enough to be rejected by the controller as parametric uncertainty when the actual load's mass is different from the one considered in Table A.1.

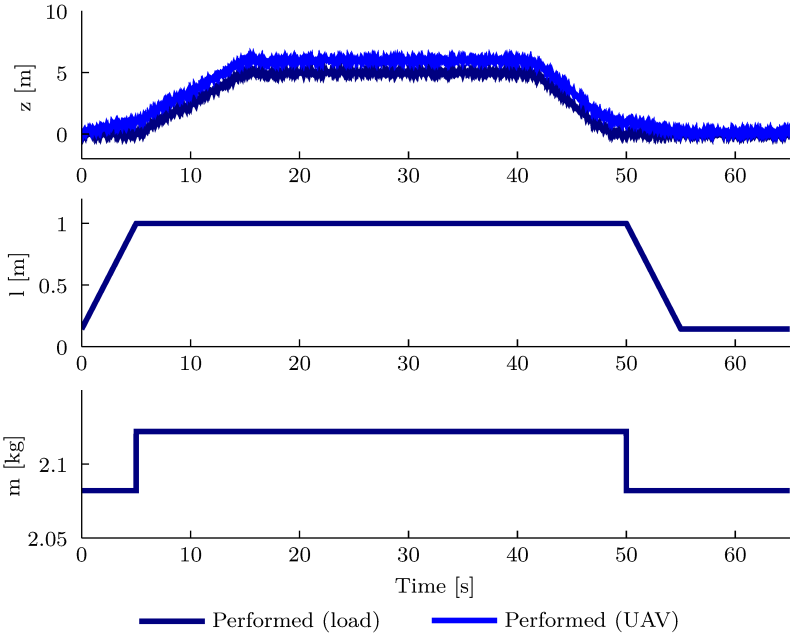


Figure 3.12: Analysis of the time evolution of the load and UAV altitude in take-off and landing maneuvers together with the rope's length and the total mass variation.

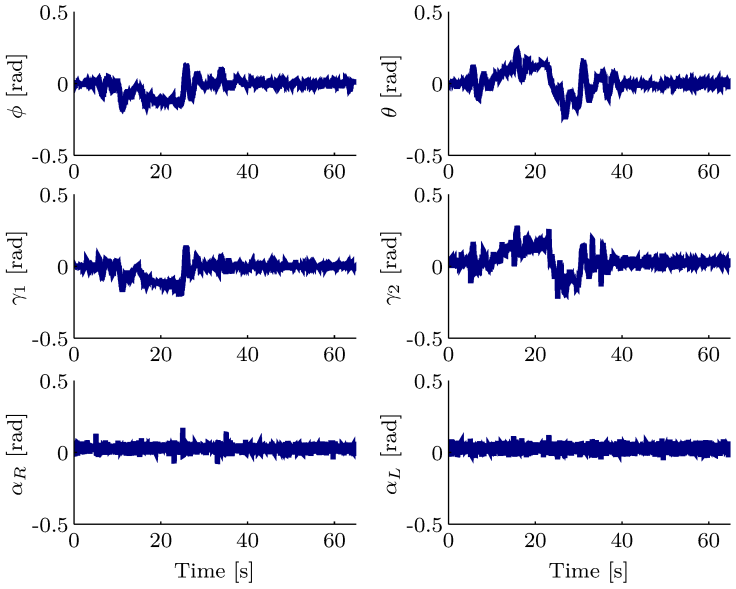


Figure 3.13: Time evolution of the remaining degrees of freedom.

Figure 3.13 shows that the system's remaining degrees of freedom are kept stable during the trajectory execution and, therefore, the goal of stabilizing the UAV during the suspended load trajectory tracking was successfully achieved. Further, Figure 3.14 shows the control input applied to the system, which, despite being noisy, did not exceed the actuators' limits.

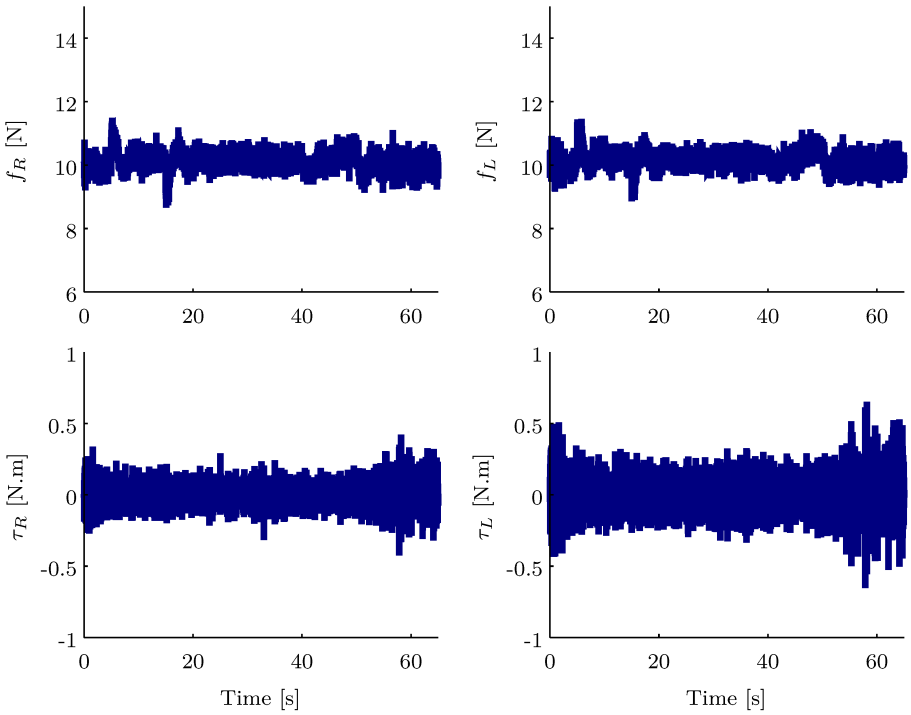


Figure 3.14: Control inputs applied to the Tilt-rotor UAV.

Scenario 3

This scenario shows that despite not being able to perform yaw movement regulation and cope with situations where the rope length either varies or is different from the value used when the controller was being tuned, the LTI-MPC can have its computational cost reduced. This feature is possible due to the inclusion of the terminal region and terminal cost in the formulation of the optimization problem, which allow to significantly reduce the prediction horizon. The same achievement cannot be easily obtained for the LTV-MPC considered in this work because of the high computation cost to compute the stabilizing sets for the LTV model (3.65).

Therefore, to show this feature the prediction and control horizon was reduced to $N_p = 2$ and $N_c = 1$, and the LTI-MPC is required to perform again the trajectory presented in Figure 3.1. Its performance was compared to the one obtained with scenario 1 when the prediction and control horizons were bigger. After presenting the performance indexes, a comparison between the computational cost is performed using the mean time of execution of each controller's simulation step.

Figures 3.15 and 3.16 show the trajectory tracking performed by the LTI-MPC with reduced horizon. Figure 3.17 shows the time evolution of the regulated variables' tracking error. The controller was able to perform the load trajectory tracking with success despite the reduction of the horizons. Further, Figures 3.18 and 3.19 illustrate, respectively, that the system's remaining degrees of freedom are kept stable and the control inputs generated by the predictive controller did not overpass the actuators' saturation limits.

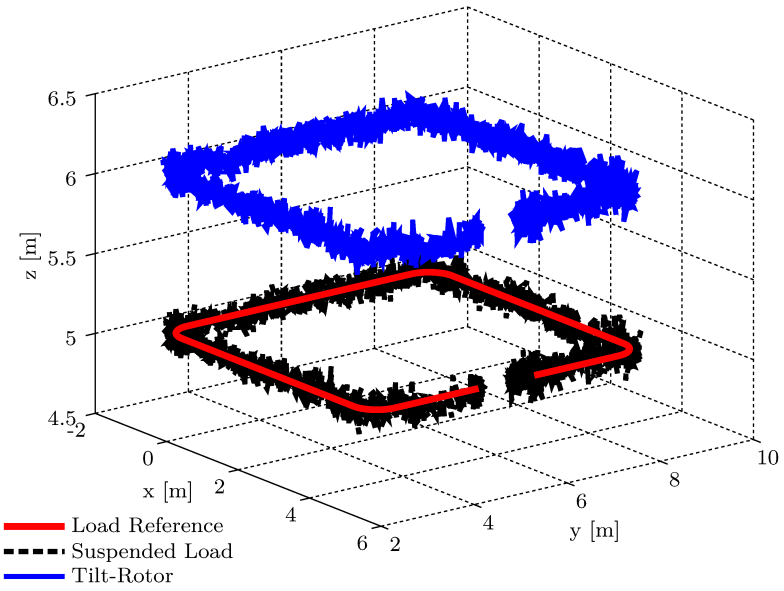


Figure 3.15: Trajectory tracking using LTI-MPC when performing the trajectory presented at Figure 3.1 with reduced prediction and control horizon.

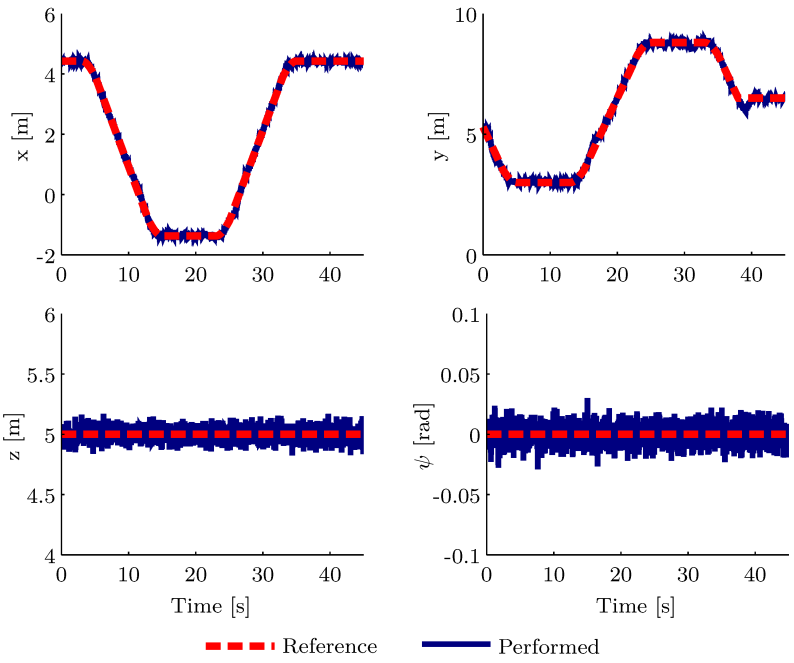


Figure 3.16: Time evolution of x, y, z, ψ.

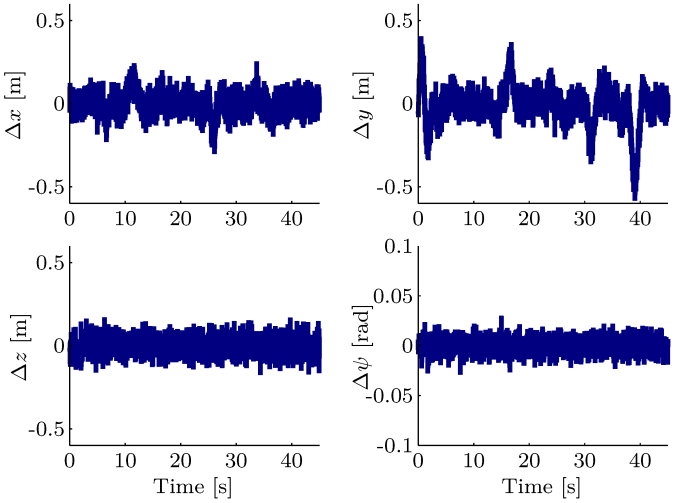


Figure 3.17: Time evolution of the tracking errors of the regulated variables.

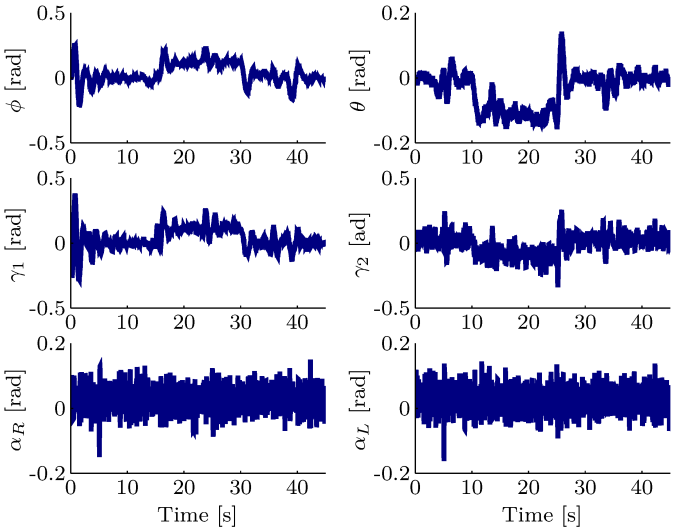


Figure 3.18: Time evolution of the remaining degrees of freedom.

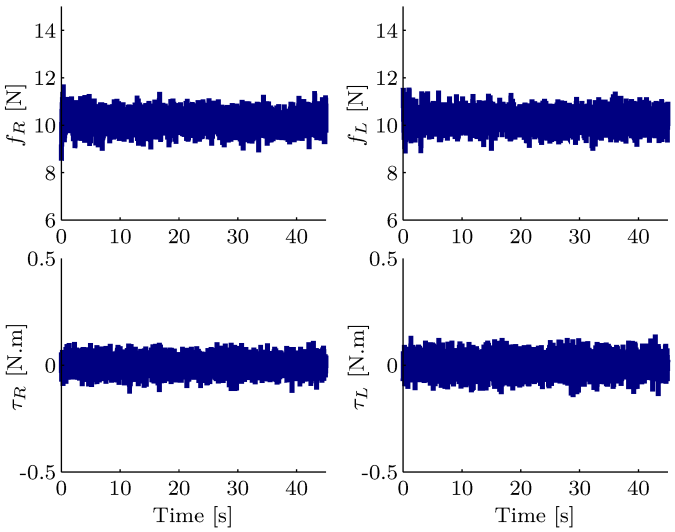


Figure 3.19: Applied thrusts and torques to the Tilt-rotor UAV.

Table 3.2 shows the MSE and IADU performance indexes for the LTI-MPC with different predictive and control horizons. Observe that the MSE index indicates that the tracking performance for x, y, z, and ψ were better when considering the controller with smaller horizons. Looking at the IADU index, it is possible to conclude that the controller with reduced horizon has smaller control effort. Therefore, it is possible to state that the reduction of N_p and N_c, allowed by the inclusion of the stability regions, optimizes the controller from the computational cost point of view without losing trajectory tracking performance.

Table 3.2: Performance indexes of the LTI-MPCs with $N_p = 100$ and $N_c = 10$ and with $N_p = 2$ and $N_c = 1$.

	LTI-MPC ($N_p = 100$, $N_c = 10$)	LTI-MPC ($N_p = 2$, $N_c = 1$)
States	MSE	
x	0.00495174	0.00428795
y	0.01455406	0.01312305
z	0.00270609	0.00257808
ψ	0.00000552	0.00000542
Inputs	IADU	
f_R	$2.4199 \cdot 10^3$	$1.5576 \cdot 10^3$
f_L	$2.4229 \cdot 10^3$	$1.5865 \cdot 10^3$
τ_{α_R}	$0.5615 \cdot 10^3$	$0.1768 \cdot 10^3$
τ_{α_L}	$0.8017 \cdot 10^3$	$0.2070 \cdot 10^3$

Moreover, Table 3.3 shows the average and worst wall-clock time for each controller algorithm simulated in this chapter. In order to compute these values, the controllers were simulated considering the same initial conditions in a loop of 3000 steps, from which their average and the worst time of execution were obtained. The LTV-MPC has the higher computational cost because of the necessity to rebuilt the prediction model using the LTV model at each sampling period, once the model Jacobians matrices are constantly changing with ζ_k. This also makes it computationally prohibitive to include the terminal cost and terminal region in the optimization problem formulation, since they would also have to be evaluated online. Therefore, besides having a costly algorithm, the prediction and control horizons, which are tuning parameters, need to be large in order to avoid oscillatory behaviors (Camacho & Bordons, 2004) that could destabilize the whole system.

Table 3.3: Wall-clock time for each simulated controller.

	LTI-MPC ($N_p = 2$, $N_c = 1$)	LTI-MPC ($N_p = 100$, $N_c = 10$)	LTV-MPC
Average (s)	0.0032	0.0550	0.2022
Worst (s)	0.0116	0.1298	0.4362

Since the computational cost of the predictive controllers are directly related with the size of their horizons, the reduction of the LTI-MPC horizon from $N_p = 100$ and $N_c = 10$

to $N_p = 2$ and $N_c = 1$ significantly decreases its computational cost, explaining why it has the smaller wall-clock time. This is an important feature, mainly because predictive controllers, knowing for being costly, are often disregarded when having a control problem that needs to run in embedded systems and has high dimension, i.e., it has a high number of degrees of freedom to be controlled; for instance, the application considered in this work.

Remark. *The MPC usually have the underlying condition of knowing a priori the trajectory to be performed. However, in addition to assuming the desired trajectory completely known, this chapter also assumes that the trajectory does not change during the flight execution. For the LTV-MPC those hypotheses allow to optimize the control algorithm by computing off-line the system model matrices* $\bar{\boldsymbol{A}}(\boldsymbol{\zeta}_k)$ *and* $\bar{\boldsymbol{B}}(\boldsymbol{\zeta}_k)$ *from* $k = 1$ *up to* $k = N_p$*, and to store that in the digital system memory, as a First In, First Out (FIFO) queue. Hence, during the flight performance, after sliding the horizon, only the model matrices for* $k = N_p$ *need to be computed to update the FIFO structure. In fact, without this procedure the average time of execution for the LTV-MPC was* 0.3954 *seconds with the worst time being* 0.7797 *seconds.*

3.4 Final remarks

In this chapter two linear model predictive controllers were presented. Both strategies were developed using a state-space model in the incremental form. For the first strategy, the linearized model is time-invariant, which leads to a LTI-MPC and, for the second strategy, a time-variant model leading to a LTV-MPC. The LTI-MPC optimization problem was developed considering constraints on the amplitude of the state deviation and control inputs. Further, a terminal cost and terminal constraint set were included in order to ensure close-loop stability for the linearized system in the vicinity of its linearization point, allowing to work with reduced prediction and control horizons without losing performance. The LTV-MPC had also considered in its optimization problem constraints on the amplitude of the state deviation and control inputs, but, due to the prohibitive computational cost, the stabilizing regions were not included. However, because of the the time-varying nature of its model and the model-based nature of predictive controllers, the LTV-MPC formulation could cope with problems that require controllers with larger domain of attraction.

Both controllers were used to solve the load transportation control problem using a Tilt-rotor UAV. Specifically, to perform trajectory tracking of the suspended load in helicopter-flight mode while keeping the vehicle stabilized. Additionally, the controllers were required to perform trajectory tracking of some desired degrees of freedom and ensure closed-loop stability for the remaining ones, rejecting constant external disturbances and

parametric uncertainties, and satisfying constraints on the state error amplitude while keeping the control signal inside the actuators' limits. Numerical simulations were made to corroborate the controllers' ability to satisfy the design requirements and highlight the difference between both strategies. When applied to the load transportation problem, the LTV-MPC was able to cope with yaw movements regulation which are a common drawback of linearized controllers when used for such application due to the necessity of defining the yaw angle equal to zero for the system linearization process. Also, the time-variant approach was able to solve the interesting practical problem of the rope's length variation without scheduling between different controllers or relying only on the controllers' robustness. This problem occurs, for instance, during take-off and landing maneuvers and could also occur due to parametric uncertainties, i.e., the use of rope with different dimension. Without the rope length as a controller's time-varying parameter, the choice of a rope with different length would necessarily imply the necessity to re-tune the controller before perform the flight. On the other hand, the LTV-MPC showed to be computationally costly, which could make its implementation on the UAV's embedded system impracticable. However, the LTI-MPC, due to the stabilizing regions, showed to be able of performing trajectory tracking using small prediction and control horizons, which could be implemented on an embedded system without any further research on how to computationally optimize predictive controller algorithms.

In the next chapter three new predictive controllers will be designed. The first one is an Economic Model Predictive Controller that includes in its formulation the idea of potential fields, enabling obstacle avoidance together with the definitions of the so-called no-fly zones. After, using a cascade structure, a Tube-Based Predictive Controller will be designed to control the x, y, γ_1, and γ_2 dynamics with increased performance when dealing with model uncertainties due to the use of some well-known set theory results, aiming to add robustness to the controller. Finally, both controllers will be put together to formulate a Tube-Based Predictive Controller with Economic Criteria robust to uncertainties and able to perform obstacle avoidance.

4

Robust Tube-Based Model Predictive Control with Economic Criteria

This chapter deals with the design of model predictive controllers taking into account advanced features in order to modify the well-known linear formulation presented in Chapter 3, commonly referred as standard MPC. Essentially, we are looking to provide robustness to the controller against uncertainties using set theory tools and perform obstacle avoidance by adding economic criteria within the optimization problem in order to obtain a feasible collision-safe trajectory. Similar to Chapter 3, the controller is designed to solve the load transportation problem using a Tilt-rotor UAV ensuring obstacle avoidance. Besides, it is also required the closed-loop stability assurance, constant external disturbances rejection, parametric uncertainties attenuation, and constraints satisfaction on state deviations and control inputs. Nevertheless, unlike the previous chapter, the load transportation problem is solved from the UAV's perspective, i.e., the controller must perform trajectory tracking of the UAV in helicopter-flight mode while keeping the load stabilized. This change of perspective is due to the high computational cost of the control strategies presented in this chapter and the necessity to represent the problem in a hierarchal fashion. Initially at this chapter an economic model predictive controller (EMPC) and a robust tube-based model predictive controller (RMPC) will be generally designed. Afterward, both controllers will be joined to reach the intended robust tube-based economic controller (REMPC). Finally, the three proposed controllers will be particularized for the trajectory tracking of the Tilt-rotor UAV with load stabilization problem, with the EMPC performing a

whole-body control and the RMPC and REMPC controlling the UAV's planar position and reducing the load's swing through the cascade structure designed at Appendix B. Numerical simulations are presented to corroborate the performance of the proposed controllers.

4.1 Economic model predictive control

The ability of MPC strategies to deal with multivariate interactions and constraints made them interesting to control constrained multiple-input multiple-output (MIMO) nonlinear systems. They are usually used to steer the system to an operational point or throughout a desired trajectory stably, without steady-state error, and with small time response by means of a quadratic cost function and a linear process representation of the system. An underlying design assumption is to consider that the reference is feasible to be executed and fulfills economic objectives such as: efficiency of operation, capacity, profitability, sustainability, etc (Ellis et al., 2017).

These goals are commonly addressed by upper level systems which are responsible to dictate feasible economic-oriented references to the MPC. A common formulation in industrial control process is a two-layer strategy, where a real time optimizer (RTO) computes setpoints according to economic objectives and, in an inner layer, an MPC strategy provides stability and constraints satisfaction (D'Jorge et al., 2017). A parallel to robotics can be made, for instance, if the navigation problem is looked as an economic goal in a way that a feasible and collision-free trajectory needs to be generated by a path planner in an upper layer while dynamic controllers address the trajectory tracking problem. Another example of economic criteria when working with robotics is the classic optimal control problem of fuel minimization. Two-layered approaches have been widely used due to the good performance and low-cost computational burden when compared to non-decoupled schemes. However, the separation into two different problems often means that the control law is designed disregarding transients, which could be a problem if the algorithms run at near frequency rates. The EMPC approach proposes an one-layer scheme by moving the economically motivated stage cost into the MPC formulation to deal with the stabilizing control problem, solved by a quadratic stage cost, while guaranteeing the economic goals required by an additional stage cost.

Regarding the control application proposed in this chapter, the main purpose of working with EMPC is to gather the path planning and trajectory tracking problems into an unified non-decoupled problem. Therefore, the economic stage cost will be constructed by means of potential functions, which are largely used in path planning problems (Choset et al., 2015). The objective is to design a strategy able to steer the robotic system through a predefined trajectory when it is not obstructed by any obstacle and, if necessary, around the obstacle to avoid collision until returning to the initial trajectory becomes a safe option

again.

4.1.1 Problem statement

Consider a finite-dimensional nonlinear system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t)\right), \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$ denotes the state vector and $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^m$ denotes the input vector, with $\mathbb{X}$ and $\mathbb{U}$ being, respectively, the set of admissible states and inputs. Further, $\boldsymbol{f} : \mathbb{X} \times \mathbb{U} \mapsto \mathbb{X}$ is the state-transition map defined by the nonlinear model that allows to predict the system's future states.

The control problem in this section for the EMPC is defined as the problem of designing a model predictive controller able to perform trajectory tracking while avoiding obstacles. The controller's algorithm considers a linear discrete-time model with state-space representation in order compute the state predictions. Besides, it must ensure closed-loop stability throughout the trajectory even in the presence of external constant disturbances and unmodelled dynamics.

Therefore, consider the mapping of equation (4.1) from the continuous-time to the discrete-time domain after being linearized through first-order expansion in Taylor series, which leads to the system's error model

$$\Delta\boldsymbol{x}_{k+1} = \boldsymbol{A}\Delta\boldsymbol{x}_k + \boldsymbol{B}\Delta\boldsymbol{u}_k, \tag{4.2}$$

where $\Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{tr}$ and $\Delta\boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$ with the superscript $(\cdot)^{tr}$ denoting desired trajectory variables, and $\boldsymbol{A}$ and $\boldsymbol{B}$ being, respectively, the state and input Jacobians linear model matrices. Additionally, the pair $(\boldsymbol{A}, \boldsymbol{B})$ is assumed to be controllable, and $\Delta\boldsymbol{x}_k \in \mathbb{E} \subset \mathbb{R}^n$, with $\mathbb{E}$ being a bounded set limiting the state error.

The EMPC cost function can be generally defined as

$$\mathcal{J} = \sum_i \ell\left(\Delta\boldsymbol{x}_k, \Delta\boldsymbol{u}_k\right) + \sum_k \ell_e\left(\boldsymbol{\varepsilon}_k\right), \tag{4.3}$$

where $\ell(\cdot)$ is the standard MPC quadratic stabilizing stage cost, and $\ell_e(\cdot)$ is the economic stage cost with $\boldsymbol{\varepsilon}_k$ being the parameters which the economic criteria are function of.

Since in this work the economic stage cost is used to define an obstacle-free trajectory, the parameters representing the economic criteria are defined as $\boldsymbol{\varepsilon}_k = \begin{bmatrix} \boldsymbol{x}_k^T & (\boldsymbol{x}_k^{tr})^T & (\boldsymbol{\xi}^{obs})^T \end{bmatrix}^T$, where $\boldsymbol{\xi}^{obs} \in \mathbb{O} \subset \mathbb{R}^3$ is a vector representing an obstacle, with $\mathbb{O}$ being a set representing all points in the robot workspace that are obstructed by some obstacle.[1] Therefore, the

[1] In this work the problem of obstacle detection is not addressed, therefore all obstacles are assumed to be perfectly known.

EMPC cost function for obstacle avoidance can be written as

$$\mathcal{J} = \sum_k \ell\left(\Delta \boldsymbol{x}_k, \Delta \boldsymbol{u}_k\right) + \sum_k \ell_e\left(\boldsymbol{x}_k, \boldsymbol{x}_k^{tr}, \boldsymbol{\xi}^{obs}\right), \tag{4.4}$$

with the economic stage cost being a function designed to allow obstacle avoidance.

4.1.2 Potential functions applied to robot navigation

It is easy to see how the potential functions presented in Chapter 2 can be used to solve the robot safe navigation problem by establishing a parallel with electric potential energy. The conservative Coulomb forces between two positively charged particles create a repulsive force, while between particles with different charge create an attraction force (Halliday et al., 2013). Therefore, it is possible to say that a repulsive potential function (see equation (2.1)) represents the interaction between positively charged particles with its energy increasing as the distance between them decreases and going to zero as their distance become sufficiently large. On the other hand, an attractive potential function (see equation (2.2)) describes the interaction between positively and negatively charged particles with the energy increasing as the particles become distant and decreasing as they get closer. If a robot and the obstacles are looked as positively charged particles and a desired position in the workspace as a negatively charged particle, the generated gradient, which is a combination of attractive and repulsive potential functions (see equation (2.3)), could drive the robot from its initial condition to its goal while avoiding obstacles.

Let $\boldsymbol{\xi}^{tr} = [x^{tr}\ y^{tr}\ z^{tr}]^T$ denotes the desired trajectory by which the robot should be moving through its workspace, and $\boldsymbol{\xi}^{goal} = [x^{goal}\ y^{goal}\ z^{goal}]^T$ be the collision-free trajectory that the robot must perform in order to avoid obstacles. Therefore, the attraction potential function defined in the equation (2.1) can be rewritten as

$$U_{att} = \frac{1}{2}\boldsymbol{\kappa} d\left(\boldsymbol{\xi}^{tr}, \boldsymbol{\xi}^{goal}\right)^2, \tag{4.5}$$

with $\boldsymbol{\kappa}$ being a weighting matrix.

Furthermore, letting $\boldsymbol{\xi} = [x\ y\ z]^T$ denotes the robot position and assuming non punctual obstacles, equation (2.2) can be rewritten as

$$U_{rep} = \frac{1}{2}\boldsymbol{\lambda}\left(\frac{1}{\min\left\{d\left(\boldsymbol{\xi}, \boldsymbol{\xi}^{obs}\right)\right\}}\right)^2, \tag{4.6}$$

where $\min\left\{d\left(\boldsymbol{\xi}, \boldsymbol{\xi}^{obs}\right)\right\}$ is the smaller Euclidean distance between the robot and the obstacle given by the 2-norm distance with $\boldsymbol{\xi}^{obs}$ denoting the obstacle position, and $\boldsymbol{\lambda}$ is a weighting matrix.

Since the robot's workspace can be obstructed by multiple obstacles and considering a safe distance d^* from the obstacle from which the repulsive field will be disregarded, the

equation (4.6) can be rewritten as (Choset et al., 2015)

$$U_{rep} = \sum U_{rep_l} = \begin{cases} \frac{1}{2}\boldsymbol{\lambda}\left(\frac{1}{\min\left\{d_l\left(\boldsymbol{\xi},\boldsymbol{\xi}_l^{obs}\right)\right\}} - \frac{1}{d_l^*}\right)^2, & \text{if } \min\left\{d_l\left(\boldsymbol{\xi},\boldsymbol{\xi}_l^{obs}\right)\right\} \leq d_l^*, \\ 0, & \text{if } \min\left\{\xi_l\left(\boldsymbol{\xi},\boldsymbol{\xi}_l^{obs}\right)\right\} > d_l^*, \end{cases} \tag{4.7}$$

where U_{rep_l} is the potential function related to the l-th obstacle, $d_l\left(\boldsymbol{\xi},\boldsymbol{\xi}_l^{obs}\right)$ is the Euclidean distance between the robot and the l-th obstacle, and d_l^* gives the safe distance from the l-th obstacle.

Therefore, the potential function able to address the safe navigation problem and that will define the EMPC economic criteria is obtained by gathering the equations (4.5) and (4.7), yielding to

$$U = U_{att} + \sum U_{rep_l}. \tag{4.8}$$

4.1.3 Economic MPC optimization problem

As previously stated, the proposed EMPC cost function (equation (4.4)) is composed by a standard quadratic stage cost and an economic stage cost. Considering the attraction and repulsive potential fields proposed in equations (4.5) and (4.7), the economic stage cost can be expressed as

$$\ell_e\left(\boldsymbol{x}_k, \boldsymbol{x}_k^{tr}, \boldsymbol{\xi}^{obs}\right) = \sum_{i=0}^{N_p} \left\|\boldsymbol{\xi}_{k+i}^{goal} - \boldsymbol{\xi}_{k+i}^{tr}\right\|_{\boldsymbol{\kappa}}^2 + \sum_{l}\sum_{i=0}^{N_p} \chi_l \left\| \frac{1}{\min\left\{\left\|\boldsymbol{\xi}_{k+i} - \boldsymbol{\xi}_l^{obs}\right\|\right\}} - \frac{1}{d_l^*}\right\|_{\boldsymbol{\lambda}}^2 \tag{4.9}$$

where N_p is the prediction horizon and χ_l is a function defined as

$$\chi_l = \begin{cases} 1, & \text{if } \min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} \leq d_l^*, \\ 0, & \text{if } \min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} > d_l^*, \end{cases} \tag{4.10}$$

with $c \in \mathbb{N}$ being a scaling factor.

Since $\boldsymbol{\xi}^{goal}$ will be a decision variable of the EMPC optimization problem, which will give a collision-free trajectory, the quadratic stage cost must be defined in a way that it penalizes the error between $\boldsymbol{\xi}^{goal}$ and $\boldsymbol{\xi}$ instead of the error-vector $\Delta\boldsymbol{x}$ of the model (4.2). Therefore, let $\boldsymbol{x}^{goal}$ be the desired state vector $\boldsymbol{x}^{tr}$ with the position variables x^{tr}, y^{tr}, and z^{tr} replaced by the obstacle-free trajectory x^{goal}, y^{goal}, and z^{goal}. Thus, defining $\Delta\tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{goal}$ the quadratic stage cost can be written as

$$\ell\left(\Delta\tilde{\boldsymbol{x}}_k, \Delta\boldsymbol{u}_k\right) = \sum_{i=0}^{N_p-1} \left\|\Delta\tilde{\boldsymbol{x}}_{k+i}\right\|_{\boldsymbol{\mathcal{Q}}}^2 + \sum_{j=0}^{N_c-1} \left\|\Delta\boldsymbol{u}_{k+j}\right\|_{\boldsymbol{\mathcal{R}}}^2 + \left\|\Delta\tilde{\boldsymbol{x}}_{k+N_p}\right\|_{\boldsymbol{P}}^2, \tag{4.11}$$

where $\boldsymbol{\mathcal{Q}} > 0$ and $\boldsymbol{\mathcal{R}} > 0$ are weighting matrices, and $\boldsymbol{P} > 0$ is the matrix used to formulate a quadratic terminal cost to ensure closed-loop stability that can be obtained using the equation (3.17) as described in Chapter 3.

Therefore, considering equations (4.9) and (4.11), the complete EMPC cost function can be expressed as

$$\mathcal{J} = \sum_{i=0}^{N_p-1} \|\Delta\tilde{\boldsymbol{x}}_{k+i}\|_{\boldsymbol{\mathcal{Q}}}^2 + \sum_{j=0}^{N_c-1} \|\Delta\boldsymbol{u}_{k+j}\|_{\boldsymbol{\mathcal{R}}}^2 + \|\Delta\tilde{\boldsymbol{x}}_{k+N_p}\|_{\boldsymbol{P}}^2 + \sum_{i=0}^{N_p} \|\boldsymbol{\xi}_{k+i}^{goal} - \boldsymbol{\xi}_{k+i}^{tr}\|_{\boldsymbol{\kappa}}^2 + \sum_{l} \sum_{i=0}^{N_p} \chi_l \left\| \frac{1}{\min\{\|\boldsymbol{\xi}_{k+i} - \boldsymbol{\xi}_l^{obs}\|\}} - \frac{1}{d_l^*} \right\|_{\boldsymbol{\lambda}}^2, \quad \forall l. \tag{4.12}$$

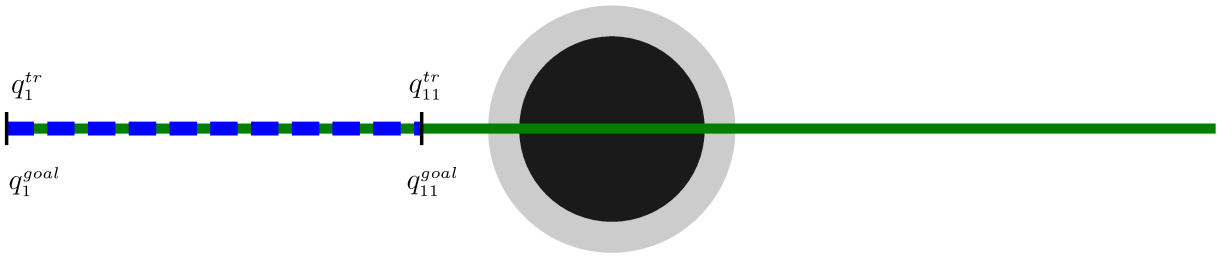


Figure 4.1: Robot system performing a trajectory (solid green line). The blue dashed line shows the collision-free trajectory for which no imminent collision is detected, the solid gray circle delimits the non-safe zone, and the solid black circle represents the obstacle.

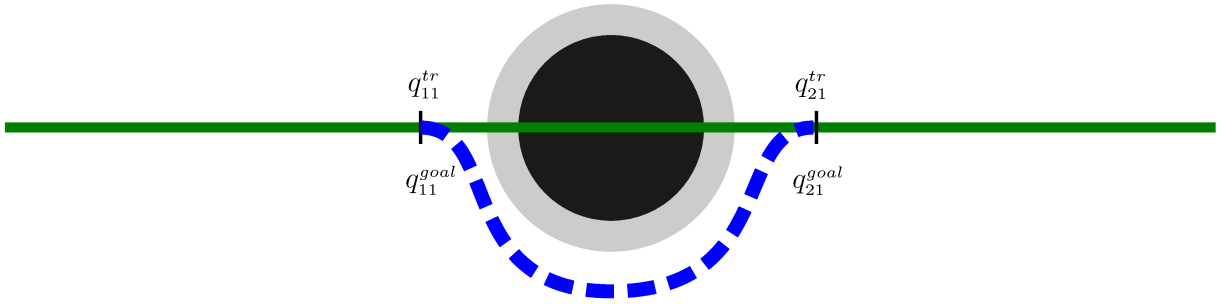


Figure 4.2: Robot system performing a trajectory (solid green line). The blue dashed line shows the collision-free trajectory while the controller is avoiding the obstacle (solid black circle) and the solid gray circle delimits the security zone.

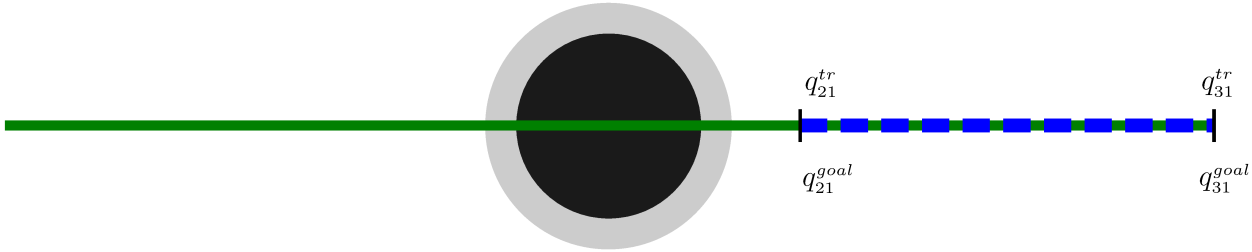


Figure 4.3: Robot system performing a trajectory (solid green line) after overcomes the obstacle (solid black line). The blue dashed line shows the collision-free trajectory and the solid gray circle delimits the security zone.

The main idea of the proposed economic cost function is illustrated in Figures 4.1 to 4.3. Consider a trajectory to be performed within 30 time-steps with a prediction horizon

of $N_p = 10$. Initially, from the time instant $k = 1$ up to $k = 11$ (see Figure 4.1), the proposed trajectory (solid green line) and the collision-free trajectory (dashed blue line) are equal since the trajectory within the predicted horizon lies outside the safe-distance zone (solid gray circle). For this case, only the attractive field and the quadratic stage cost act in order to keep $\boldsymbol{\xi}_k^{tr} = \boldsymbol{\xi}_k^{goal}$ for $k = 1, \cdots, 11$. After, from the time instant $k = 11$ up to $k = 21$ (see Figure 4.2), the proposed trajectory moves inside the non-safe zone colliding with the obstacle (black circle). Thus, both the attractive and repulsive fields act in order to find a collision-free trajectory that goes around the obstacle (solid black circle) and, after overcoming it, make the alternative trajectory equals to the proposed trajectory again. Finally, from the time instant $k = 21$ up to $k = 31$ (see Figure 4.3), the repulsive field is again disregarded since there is no imminent collision. The attractive field together with the quadratic stage cost act again in order to make $\boldsymbol{\xi}_k^{tr} = \boldsymbol{\xi}_k^{goal}$ for $k = 21, \cdots, 31$.

Further, in order to finish the construction of the optimization problem for the EMPC, the following constraints are considered:

1. Initial condition constraint:

   When initializing the control algorithm the system's initial condition is considered without collision and state error, that is

   $$\Delta \boldsymbol{x}_0 = \Delta \tilde{\boldsymbol{x}}_0 = \mathbf{0}. \tag{4.13}$$

   Afterward, at each iteration of the algorithm the following initial condition constraint is considered

   $$\Delta \tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{goal}. \tag{4.14}$$

   Note that the constraint (4.14) depends on $\boldsymbol{x}_k^{goal}$ which is a decision variable of the EMPC optimization problem, i.e., it is unknown. Therefore, its value must be stored from the previous iterations of the control algorithm making the initialization constraint (4.13) essential to the algorithm feasibility.

2. Model constraint:

   In Chapter 3, prediction models were explicitly derived in equations (3.6), (3.7), and (3.9). Here, the model-based nature of the EMPC represented by the state prediction procedure is implicitly considered as a constraint. Both strategies are equivalent, differing only in the notation and in the algorithm implementation.

   $$\Delta \tilde{\boldsymbol{x}}_{k+i+1} = \boldsymbol{A} \Delta \tilde{\boldsymbol{x}}_{k+i} + \boldsymbol{B} \Delta \boldsymbol{u}_{k+i}, \qquad \forall i = 1, \cdots, N_p. \tag{4.15}$$

   Note that here the model constraint also ensures feasibility to the collision-free trajectory since the evolution of $\Delta \tilde{\boldsymbol{x}}_{k+i}$, which is function of $\boldsymbol{x}_{k+i}^{goal}$, must be subjected

to the model, being $\Delta \boldsymbol{u}_{k+i}$ a bounded signal.

3. Maximum state error constraint:

   As previously stated, the trajectory tracking error is bounded by a set $\mathbb{E}$. Thus, to ensure this condition, the following constraint must be regarded

$$\Delta \tilde{\boldsymbol{x}}_{k+i} \in \mathbb{E}, \qquad \forall i = 1, \cdots, N_p. \tag{4.16}$$

4. Input signal constraints:

   In order to avoid saturation on the actuators the following constraint is considered

$$\boldsymbol{u}_{k+i} \in \mathbb{U}, \qquad \forall i = 0, \cdots, N_c - 1, \tag{4.17}$$

   which can be easily mapped to decision variable $\Delta \boldsymbol{u}_{k+i}$

5. Terminal set constraint:

   Aiming to ensure closed-loop stability, a terminal set constraint is defined to force the last element of the predicted state vector to belong to a positively invariant set (see Definition 2.13). Therefore,

$$\Delta \tilde{\boldsymbol{x}}_{k+N_p} \in \boldsymbol{\Omega}, \tag{4.18}$$

   where $\boldsymbol{\Omega}$ denotes a maximal control invariant set (see Definition 2.16), which can be obtained using the same iterative procedure, based on the one-step operator (3.27) described in Chapter 3.

6. No-fly zone constraints:

   Although the potential functions have been designed to perform safe navigation, they do not strictly ensure that some collision will not occur. For that reason, hard constraints, the so-called no-fly zones, are imposed to ensure that the robot position and the alternative generated trajectory do not invade the obstacle space. Hence,

$$\boldsymbol{\xi}_{k+i} \in \mathbb{R}^3 - \mathbb{O}, \qquad \forall i = 1, \cdots, N_p, \tag{4.19}$$

$$\boldsymbol{\xi}^{goal}_{k+i} \in \mathbb{R}^3 - \mathbb{O}, \qquad \forall i = 1, \cdots, N_p. \tag{4.20}$$

Finally, the optimization problem for the EMPC strategy, giving the optimal control sequence to be applied to the system and the collision-free trajectory to be performed by the robot, is stated as

$$\begin{aligned} \min_{\vec{\boldsymbol{u}}, \vec{\boldsymbol{x}}, \vec{\boldsymbol{x}}^{goal}} \quad & \mathcal{J}, \\ \text{subject to} \quad & (4.13 - 4.20), \end{aligned} \tag{4.21}$$

with $\underset{\rightarrow}{\boldsymbol{u}} = \left[\boldsymbol{u}_k^T \ \cdots \ \boldsymbol{u}_{k+N_c-1}^T\right]^T$ and $\underset{\rightarrow}{\boldsymbol{x}} = \left[(\boldsymbol{x}_{k+1})^T \ \cdots \ (\boldsymbol{x}_{k+N_p})^T\right]^T$.

4.2 Robust tube-based model predictive control

Closed-loop model based control methods have an inherited robustness due to the process of feed the information of the controlled states back into the control algorithm, which allows the controller to deal with unmodelled dynamics, measurement errors, and unknown and/or unmodelled disturbances. However, since these uncertainties are not taken into account in the control design process, the domain of attraction for which the controllers will be able to keep the system controlled using only their feedback nature are small, mainly for those algorithms based on model linearization. Therefore, it becomes necessary to work with controllers that, by design concept, could cope with uncertainties (Langson et al., 2004b). As stated in Bemporad & Morari (1999), a control system is robust only when it maintains its stability and performance specifications in the presence of bounded disturbances, which is a desirable feature.

When it comes to formulate robust MPCs, a common approach is to define regions in the state-space that bounds the difference between the nominal and the uncertain systems. MPCs working with admissible regions that contains the possible trajectories of an uncertain system are commonly referred as tube-based and these regions, the so-called reachable sets, can be obtained using set-theory tools. In this work, the control of time-varying systems will be considered. Therefore, despite bound additive uncertainties, the reachable sets must also bound the system's realizations regarding the time-varying model parameters. Further details about the controller presented in this section can be obtained in the work of Sánchez (2011).

The robust tube-based model predictive control strategy considers that the states' trajectory of the nominal and the uncertain systems are different, and the mismatch between them needs to be controlled and bounded. Therefore, the control law applied to the system considers the sum of two different control policies: (i) a pre-stabilizing policy able to control the nominal system with state and input constraints' regions shaped by the reachable sets in other to take into account the uncertainties, and (ii) a control policy to compensate the mismatch between the nominal and uncertain systems.

4.2.1 Problem statement

Consider a finite-dimensional uncertain nonlinear system of the form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\right), \tag{4.22}$$

where $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^m$ is the input vector, and $\boldsymbol{w} \in \mathbb{W} \subset \mathbb{R}^n$ is the uncertainty vector with $\mathbb{X}$, $\mathbb{U}$, and $\mathbb{W}$ being, respectively, the set of admissible states, admissible inputs, and the set of bound uncertainties. Further, $\boldsymbol{f} : \mathbb{X} \times \mathbb{U} \times \mathbb{W} \mapsto \mathbb{X}$ is the state-transition map defined by the nonlinear system model.

The control problem for the RMPC can be defined as the problem of designing a model predictive control system able to perform trajectory tracking and robust to bounded uncertainties and unmodelled dynamics. The controller's algorithm considers a linear discrete-time model with state-space representation in order to compute the state predictions, and it must ensure closed-loop stability throughout the trajectory.

Now, consider the mapping of equation (4.22) from the continuous-time to the discrete-time domain after being linearized through first-order expansion in Taylor series around a time-varying trajectory, which gives the uncertain error model

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{4.23}$$

where $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{tr}$ and $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$, $\boldsymbol{w}_k$ is the vector of additive uncertainties, and $\boldsymbol{\zeta}_k \in \Delta \subset \mathbb{R}^v$ denotes the vector of time-varying model parameters with Δ being a convex polytope with 2^v vertices. Further, $\boldsymbol{A}(\boldsymbol{\zeta}_k)$ and $\boldsymbol{B}(\boldsymbol{\zeta}_k)$ are, respectively, the state and input Jacobian linear model matrices, which are assumed to have polytopic form in order to represent the system's realization considering the parameters $\boldsymbol{\zeta}$ at the time instant k, and to compose a controllable linear system.

The nominal linear time-varying error model can be obtained by considering the system (4.23) without additive uncertainty, i.e, $\mathbb{W} = \{0\}$, yielding to

$$\Delta \boldsymbol{x}_{k+1}^{nom} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k^{nom} + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.24}$$

where $\Delta \boldsymbol{x}_k^{nom} = \boldsymbol{x}_k^{nom} - \boldsymbol{x}_k^{tr}$ with the superscript $(\cdot)^{nom}$ denoting the state vector predicted without uncertainties, and $\boldsymbol{g}_k$ is the policy able to control the nominal system.

The mismatch error between the uncertain system (4.23) and the nominal system (4.24) can be defined as

$$\begin{aligned} \boldsymbol{e}_{k+1} &= \Delta \boldsymbol{x}_{k+1} - \Delta \boldsymbol{x}_{k+1}^{nom} \\ &= \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k + \boldsymbol{w}_k - \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k^{nom} - \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k \\ &= \boldsymbol{A}(\boldsymbol{\zeta}_k)\boldsymbol{e}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\left[\Delta \boldsymbol{u}_k - \boldsymbol{g}_k\right] + \boldsymbol{w}_k. \end{aligned} \tag{4.25}$$

Since the control objective is to compensate the mismatch error $\boldsymbol{e}_k$ while controlling the nominal system through some desired trajectory, the control input that controls the uncertain system can be defined as (Sánchez, 2011)

$$\Delta \boldsymbol{u}_k = \boldsymbol{K}\left(\boldsymbol{\zeta}_k\right)\boldsymbol{e}_k + \boldsymbol{g}_k, \tag{4.26}$$

where $\boldsymbol{K}(\boldsymbol{\zeta}_k) \in \mathbb{R}^{n\times m}$ is an adaptive feedback control gain designed to control the mismatch error.

Using the definition (4.26), the mismatch error model can be rewritten as

$$\boldsymbol{e}_{k+1} = [\boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}(\boldsymbol{\zeta}_k)]\,\boldsymbol{e}_k + \boldsymbol{w}_k. \tag{4.27}$$

Further, to design the RMPC the follows elements must be obtained: (i) the adaptive feedback gain that controls the mismatch error, (ii) the reachable sets to envelope the system's evolution in the space ensuring robustness, (iii) the state error and control input constraints, (iv) the terminal constraint and terminal regions to ensure closed-loop stability, and (v) an MPC strategy to control the nominal system. In the following subsections these elements are defined.

4.2.2 Mismatch error adaptive controller

In order to obtain the adaptive feedback gain able to control the mismatch error (4.26), consider the following Lyapunov function

$$V(\boldsymbol{e}_k) = \boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k, \tag{4.28}$$

where $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ is a Lyapunov matrix. Further, to obtain an asymptotically stable control system, the conditions $\boldsymbol{P} > 0$ and $V(\boldsymbol{e}_{k+1}) - V(\boldsymbol{e}_k) < 0$ must be satisfied.

Aiming to add some performance requirements to the problem, $V(\boldsymbol{e}_k)$ is considered as an upper bound for the linear quadratic regulator (LQR) cost function. Thereby, the performance of the mismatch error adaptive controller should be as close to the LQR controller performance as possible. Therefore, the following optimal control problem is considered

$$V(\boldsymbol{e}_0) \geq \min_{v[0,\infty)} \sum_{k=0}^{\infty} \boldsymbol{e}_k^T \boldsymbol{\mathcal{Q}} \boldsymbol{e}_k + \boldsymbol{v}_k^T \boldsymbol{\mathcal{R}} \boldsymbol{v}_k. \tag{4.29}$$

where

$$\boldsymbol{v}_k = \boldsymbol{K}(\boldsymbol{\zeta}_k)\,\boldsymbol{e}_k, \tag{4.30}$$

and the matrices $\boldsymbol{\mathcal{Q}} > 0$ and $\boldsymbol{\mathcal{R}} > 0$ are, respectively, weighting matrices for the state error and control effort.

Considering the conditions for asymptotic stability and the performance condition (4.29), it is possible to define the following inequalities

$$\boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k > 0, \tag{4.31}$$

$$\boldsymbol{e}_k^T \left(\boldsymbol{A}_f^T \boldsymbol{P} \boldsymbol{A}_f\right) \boldsymbol{e}_k - \boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k \leq -\boldsymbol{e}_k^T \left(\boldsymbol{\mathcal{Q}} + \boldsymbol{K}(\boldsymbol{\zeta}_k)^T \boldsymbol{\mathcal{R}} \boldsymbol{K}(\boldsymbol{\zeta}_k)\right) \boldsymbol{e}_k, \tag{4.32}$$

where $\boldsymbol{A}_f = \boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}(\boldsymbol{\zeta}_k)$.

Removing the the state mismatch error variables, the inequalities can be rewritten as

$$\boldsymbol{P} > 0, \tag{4.33}$$

$$\boldsymbol{A}_f^T \boldsymbol{P} \boldsymbol{A}_f + \boldsymbol{K}(\boldsymbol{\zeta})^T \boldsymbol{\mathcal{R}} \boldsymbol{K}(\boldsymbol{\zeta}) + \boldsymbol{\mathcal{Q}} - \boldsymbol{P} \leq 0. \tag{4.34}$$

The Jacobian linear model matrices $\boldsymbol{A}(\boldsymbol{\zeta})$ and $\boldsymbol{B}(\boldsymbol{\zeta})$ are assumed to have polytopic representation, i.e., it is possible to represent each of them as a convex sum regarding the uncertain parameter $\boldsymbol{\zeta}$. This assumption allows one to represent the inequalities (4.33) and (4.34) as linear matrix inequalities conditions (see Definition 2.25). Although (4.33) is already a valid LMI condition, the inequality (4.34) is not due to its nonlinear terms. Thereafter, equation (4.34) needs to be manipulated.

Rearranging (4.34) and applying the Schur complement twice (see Definition 2.27), it holds that

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{A}_f^T & \boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{K}(\boldsymbol{\zeta})^T \boldsymbol{\mathcal{R}}^{\frac{1}{2}} \\ \boldsymbol{A}_f & \boldsymbol{P}^{-1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \mathbf{0} & \mathbb{I} & \mathbf{0} \\ \boldsymbol{\mathcal{R}}^{\frac{1}{2}} \boldsymbol{K}(\boldsymbol{\zeta}) & \mathbf{0} & \mathbf{0} & \mathbb{I} \end{bmatrix} \geq 0. \tag{4.35}$$

In order to remove the nonlinear terms, it is necessary to pre and post multiply (3.15) by the block diagonal matrix $\mathbf{blkdiag}(\boldsymbol{P}^{-1}, \mathbb{I}, \mathbb{I}, \mathbb{I})$, and to define $\boldsymbol{S} = \boldsymbol{P}^{-1}$, $\boldsymbol{Y}(\boldsymbol{\zeta}) = \boldsymbol{K}(\boldsymbol{\zeta})\boldsymbol{P}^{-1}$. Thus, the LMI condition used to calculate a state feedback controller that ensure stability and performance when controlling the system's mismatch error can be written as

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{H}^T & \boldsymbol{S}\boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{Y}(\boldsymbol{\zeta})^T \boldsymbol{\mathcal{R}}^{\frac{1}{2}} \\ \boldsymbol{H} & \boldsymbol{S} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{\mathcal{Q}}^{\frac{1}{2}} \boldsymbol{S} & \mathbf{0} & \mathbb{I} & \mathbf{0} \\ \boldsymbol{\mathcal{R}}^{\frac{1}{2}} \boldsymbol{Y}(\boldsymbol{\zeta}) & \mathbf{0} & \mathbf{0} & \mathbb{I} \end{bmatrix} \geq 0, \tag{4.36}$$

where $\boldsymbol{H} = \boldsymbol{A}(\boldsymbol{\zeta})\boldsymbol{S} + \boldsymbol{B}(\boldsymbol{\zeta})\boldsymbol{Y}(\boldsymbol{\zeta})$. Note that the condition (4.36) must hold for each of the 2^v vertices of the polytope Δ.

Finally, the following optimization problem must be considered

$$\begin{aligned} \min_{\boldsymbol{S}>0, \boldsymbol{Y}(\boldsymbol{\zeta}) \forall \boldsymbol{\zeta}} \quad & \mathrm{Tr}(\boldsymbol{P}), \\ \text{subject to} \quad & (4.36) \ \ \forall \boldsymbol{\zeta} \in \Delta, \end{aligned} \tag{4.37}$$

where $\mathrm{Tr}(\cdot)$ is the trace operator.

The solution of the optimization problem (4.37) gives the Lyapunov positive definite matrix $\boldsymbol{P}$ and also the feedback gains $\boldsymbol{K}(\boldsymbol{\zeta})$ able to control the system for each vertex of the polytope Δ. In Santos & Raffo (2016a), inspired by the work of Gonzalez et al. (2010),

a similar problem was solved using these gains to create an adaptive control scheme by means of an LP problem in order to find at each time-step an optimal feedback gain for the system's current realization. Although this approach could be used to obtain the adaptive feedback gain for the mismatch error control problem considered in this work, it will require online computation increasing the complexity of the RMPC algorithm. Therefore, the approach proposed in Sánchez (2011) to obtain the feedback gain $\boldsymbol{K}(\boldsymbol{\zeta})$ in a more computational efficient way is considered.

Hence, considering the matrix $\boldsymbol{P}$ known after the solution of the problem (4.37), the inequality (4.32) could be reevaluated and solved explicitly for the control input $\boldsymbol{v}_k$ through

$$\min_{\boldsymbol{v}_k \forall k} \; \boldsymbol{e}_k^T \boldsymbol{\mathcal{Q}} \boldsymbol{e}_k + \boldsymbol{v}_k^T \boldsymbol{\mathcal{R}} \boldsymbol{v}_k + \left(\boldsymbol{A}(\boldsymbol{\zeta}_k)\boldsymbol{e}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{v}_k\right)^T \boldsymbol{P} \left(\boldsymbol{A}(\boldsymbol{\zeta}_k)\boldsymbol{e}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{v}_k\right) - \boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k. \tag{4.38}$$

The cost function of the problem (4.38) can be rewritten with respect to $\boldsymbol{v}_k$ as

$$\begin{aligned} f^* = \; & \boldsymbol{e}_k^T \boldsymbol{\mathcal{Q}} \boldsymbol{e}_k + \boldsymbol{v}_k^T \boldsymbol{\mathcal{R}} \boldsymbol{v}_k + \boldsymbol{e}_k^T \boldsymbol{A}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{A}(\boldsymbol{\zeta}_k) \boldsymbol{e}_k + \boldsymbol{v}_k^T \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{B}(\boldsymbol{\zeta}_k) \boldsymbol{v}_k + \\ & 2\boldsymbol{v}_k^T \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{A}(\boldsymbol{\zeta}_k) \boldsymbol{e}_k - \boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k. \end{aligned} \tag{4.39}$$

Assuming the system's current realization known, i.e., the uncertain model parameter vector is either measured or estimated at each sampling period k, it is possible to analytically solve (4.38) through differentiation of (4.39) with respect to the control input. This leads to

$$\frac{\partial f^*}{\partial \boldsymbol{v}_k} = 2\boldsymbol{\mathcal{R}} \boldsymbol{v}_k + 2\boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{B}(\boldsymbol{\zeta}_k) \boldsymbol{v}_k + 2\boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{A}(\boldsymbol{\zeta}_k) \boldsymbol{e}_k = 0, \tag{4.40}$$

which by simple manipulation gives

$$\boldsymbol{v}_k = -\left(\boldsymbol{\mathcal{R}} + \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{B}(\boldsymbol{\zeta}_k)\right)^{-1} \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{A}(\boldsymbol{\zeta}_k) \boldsymbol{e}_k. \tag{4.41}$$

Finally, using the relation (4.30), the feedback gain for the mismatch error adaptive controller can be written as

$$\boldsymbol{K}(\boldsymbol{\zeta}_k) = -\left(\boldsymbol{\mathcal{R}} + \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{B}(\boldsymbol{\zeta}_k)\right)^{-1} \boldsymbol{B}(\boldsymbol{\zeta}_k)^T \boldsymbol{P} \boldsymbol{A}(\boldsymbol{\zeta}_k). \tag{4.42}$$

Note, that instead of solving an online LP problem as made in Santos & Raffo (2016a), equation (4.42) needs only to perform numerical algebraic matrix operations.

4.2.3 Reachable sets

The reachable sets, which define the region around the nominal trajectory that envelops the system states for any bounded uncertainties and disturbances, can be defined as (Sánchez, 2011)

$$\mathscr{R}_{k+i+1} = \left(\boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}(\boldsymbol{\zeta}_k)\right) \mathscr{R}_{k+i} \oplus \mathbb{W}, \quad \forall i = 0, \cdots, N_p - 1, \tag{4.43}$$

with N_p being the prediction horizon. Further, in order to recursively use equation (4.43) to compute the reachable sets along the prediction horizon, consider the initialization condition $\mathscr{R}_k = \{0\}$.

Note that the equation (4.43) needs to be evaluated online because of its dependency on the time-variant Jacobian linear model matrices. If the system was time-invariant, i.e. $\boldsymbol{A}(\boldsymbol{\zeta}_{k+1}) = \boldsymbol{A}(\boldsymbol{\zeta}_k)$ and $\boldsymbol{B}(\boldsymbol{\zeta}_{k+1}) = \boldsymbol{B}(\boldsymbol{\zeta}_k)$ $\forall k$, equation (4.43) can still be used to compute the reachable sets. However, these operations could be performed offline since the sets becomes constant robust invariant sets.

4.2.4 State and input constraints

In order to take advantage of one of the main features of predictive controllers, constraints on the amplitude of the control signal and on the state error are considered. Therefore, let the sets $\mathbb{E} \in \mathbb{R}^n$ and $\mathbb{V} \in \mathbb{R}^m$ denote, respectively, the bounding trajectory tracking error set and the admissible control input set for the uncertain model (4.23). Thus, the constraints can be written as

$$\Delta \boldsymbol{x}_k \in \mathbb{E}, \tag{4.44}$$

$$\Delta \boldsymbol{u}_k \in \mathbb{V}. \tag{4.45}$$

Since the MPC strategy is used to control the nominal system in the RMPC formulation, the constraints (4.44) and (4.45) must be redefined relatively to the nominal state error $\Delta \boldsymbol{x}_k^{nom}$ and control input $\boldsymbol{g}_k$ considered in the model (4.24). Therefore, using the reachable sets (4.43) to reshape the bounding sets $\mathbb{E}$ and $\mathbb{V}$ in order to have tighter constraints, it yields to

$$\bar{\mathbb{E}}_i = \mathbb{E} \ominus \mathscr{R}_i, \qquad \forall i = 1, \cdots, N_p, \tag{4.46}$$

$$\bar{\mathbb{V}}_i = \mathbb{V} \ominus \boldsymbol{K}(\boldsymbol{\zeta}_k)\mathscr{R}_i, \qquad \forall i = 0, \cdots, N_c - 1. \tag{4.47}$$

Note that the Pontryagin difference needs to be performed online increasing the complexity of the MPC algorithm.

Finally, the constraints for the nominal MPC strategy are described as

$$\Delta \boldsymbol{x}_{k+i}^{nom} \in \bar{\mathbb{E}}_i, \qquad \forall i = 1, \cdots, N_p, \tag{4.48}$$

$$\boldsymbol{g}_{k+i} \in \bar{\mathbb{V}}_i, \qquad \forall i = 0, \cdots, N_c - 1. \tag{4.49}$$

4.2.5 Terminal cost and terminal constraint

Aiming to ensure closed-loop stability to the MPC, a terminal cost and a terminal constraint set to be added into the nominal MPC strategy are considered (Mayne et al., 2000).

A terminal cost able to ensure closed-loop stability can be defined using a Lyapunov function $V(\Delta \boldsymbol{x}_k) = \Delta \boldsymbol{x}_k^T \boldsymbol{P} \Delta \boldsymbol{x}_k$. Thus, the matrix $\boldsymbol{P}$ obtained from the optimization problem (4.37) can be used to formulate a quadratic terminal cost since it is a Lyapunov function able to ensure closed-loop stability for the uncertain model (4.23).[2]

As for Chapter 3, a terminal constraint ensuring the last element of the predicted state sequence to belong to an invariant set is obtained by means of the one-step operator. However, instead of considering the operator presented at equation (3.27) used to obtain a maximal control invariant set, its definition is modified in order to include the uncertainties and define a maximal robust control invariant set (see Definitions 2.23 and 2.24). Thus, consider the robust one-step operator

$$\tilde{Q}(\boldsymbol{\Omega}) = \{\Delta \boldsymbol{x} \in \mathbb{E} : \exists\, \boldsymbol{K}(\boldsymbol{\zeta})\, \Delta \boldsymbol{x} \in \mathbb{V},\ (\boldsymbol{A}(\boldsymbol{\zeta}) + \boldsymbol{B}(\boldsymbol{\zeta})\boldsymbol{K}(\boldsymbol{\zeta}))\, \Delta \boldsymbol{x} + \boldsymbol{w} \in \boldsymbol{\Omega},\ \forall \boldsymbol{w} \in \mathbb{W}\}. \tag{4.50}$$

Note that the one-step operator as defined in (4.50) must be evaluated for each vertex of the polytope $\boldsymbol{\Delta}$.

Using the one-step operator (4.50), the maximal robust control invariant set can be obtained by means of the following iterative procedure:

1. Initialization: $\boldsymbol{\Omega}_0 = \mathbb{E} \cap \{\omega \in \mathbb{R}^n : \boldsymbol{K}(\boldsymbol{\zeta})\, \omega \in \mathbb{V},\ \forall \boldsymbol{\zeta}\}$.
2. Iteration: $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k \cap \tilde{Q}(\boldsymbol{\Omega}_k)$.
3. Terminal condition: stop when $\boldsymbol{\Omega}_{k+1} = \boldsymbol{\Omega}_k$ or $\boldsymbol{\Omega}_{k+1} = \emptyset$. Set $\boldsymbol{\Omega} = \boldsymbol{\Omega}_\infty = \boldsymbol{\Omega}_{k+1}$.

The terminal constraint set obtained through the iterative procedure considers the uncertain system. Hence, in order to use $\boldsymbol{\Omega}$ to constraint the nominal system, the Pontryagin difference with the reachable set at the N_p-step is considered

$$\Delta \boldsymbol{x}_{k+N_p}^{nom} \in \boldsymbol{\Omega} \ominus \mathscr{R}_{k+N_p}. \tag{4.51}$$

4.2.6 MPC strategy

Since the mismatch error adaptive controller and the reachable sets are dealing with the mismatch between the nominal and the uncertain system, an MPC strategy is designed to deal only with the nominal control problem.

Therefore, consider the cost function

$$\mathcal{J} = \sum_{i=0}^{N_p-1} \left\|\Delta \boldsymbol{x}_{k+i}^{nom}\right\|_{\boldsymbol{Q}}^2 + \sum_{j=0}^{N_c-1} \left\|\boldsymbol{g}_{k+j}\right\|_{\boldsymbol{\mathcal{R}}}^2 + \left\|\Delta \boldsymbol{x}_{k+N_p}^{nom}\right\|_{\boldsymbol{P}}^2, \tag{4.52}$$

[2] Although the LMI used to formulate the optimization problem (4.37) was obtained for the mismatch error model (4.27), it is easy to see that the matrix $\boldsymbol{P}$ forms a Lyapunov function for the uncertain model (4.23), since their closed-loop model are the same.

with N_c being the control horizon.

Moreover, in order to consider the complete optimization problem design, an initial state constraint and a model constraint must be defined. Therefore,

$$\Delta \boldsymbol{x}_0^{nom} = \mathbf{0}. \tag{4.53}$$

$$\Delta \boldsymbol{x}_{k+i+1}^{nom} = \boldsymbol{A}(\boldsymbol{\zeta})\Delta \boldsymbol{x}_{k+i}^{nom} + \boldsymbol{B}(\boldsymbol{\zeta})\boldsymbol{g}_{k+i}, \qquad \forall i = 1, \cdots, N_p. \tag{4.54}$$

Thus, the MPC problem for the nominal system can be written as

$$\begin{aligned} \min_{\substack{\underset{\rightarrow}{\boldsymbol{g}}, \underset{\rightarrow}{\boldsymbol{x}}}} \quad & \mathcal{J}, \\ \text{subject to} \quad & (4.48), (4.49), (4.51), (4.53), (4.54), \end{aligned} \tag{4.55}$$

with $\underset{\rightarrow}{\boldsymbol{g}} = \begin{bmatrix} \boldsymbol{g}_k^T & \cdots & \boldsymbol{g}_{k+N_c-1}^T \end{bmatrix}^T$ and $\underset{\rightarrow}{\boldsymbol{x}} = \begin{bmatrix} (\boldsymbol{x}_{k+1})^T & \cdots & (\boldsymbol{x}_{k+N_p})^T \end{bmatrix}^T$.

4.3 Robust tube-based economic model predictive control

This section proposes an REMPC strategy formulated combining the controllers EMPC and RMPC previously presented in this chapter. The main goal is to design a controller able to perform safe navigation and robust to uncertainties, such as: unmodelled dynamics, measurements errors, and unknown and/or unmodelled disturbances. This can be accomplished by simply using the EMPC strategy as pre-stabilizing policy rather than the MPC strategy used in the presented RMPC algorithm.

Therefore, the REMPC strategy can be formulated as a combination of two control policies: (i) an EMPC policy able to control the nominal system while performing obstacle avoidance, and (ii) a control policy to compensate the mismatch between the nominal and the uncertain system.

4.3.1 Problem statement

As for the RMPC problem statement, a finite-dimensional uncertain nonlinear system is considered

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{w}(t)\right), \tag{4.56}$$

where $\boldsymbol{x} \in \mathbb{X} \subset \mathbb{R}^n$ denotes the state vector, $\boldsymbol{u} \in \mathbb{U} \subset \mathbb{R}^m$ denotes the input vector, and $\boldsymbol{w} \in \mathbb{W} \subset \mathbb{R}^n$ denotes the uncertainty-vector with $\mathbb{X}$, $\mathbb{U}$, and $\mathbb{W}$ being bounded sets.

The control problem for the REMPC can be defined as the problem of designing a model predictive control system able to perform trajectory tracking while avoiding obstacles and robust to bounded uncertainties and unmodelled dynamics. The controller's algorithm

considers a linear discrete-time model with state-space representation in order compute the state predictions and must ensure closed-loop stability throughout the trajectory.

The nonlinear model (4.56) is linearized around a time-varying trajectory and discretized to generate the time-varying uncertain model

$$\Delta \boldsymbol{x}_{k+1} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{u}_k + \boldsymbol{w}_k, \tag{4.57}$$

where $\Delta \boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{tr}$, $\Delta \boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_k^{tr}$, $\boldsymbol{w}_k$ is the vector of additive uncertainties, and $\boldsymbol{\zeta}_k \in \Delta$ denotes the vector of time-varying model parameters with Δ being a convex polytope. The matrices $\boldsymbol{A}(\boldsymbol{\zeta}_k)$ and $\boldsymbol{B}(\boldsymbol{\zeta}_k)$ are assumed to have polytopic form and to compose a controllable system.

Further, the nominal model is defined considering (4.57) with $\mathbb{W} = \{0\}$, yielding to

$$\Delta \boldsymbol{x}_{k+1}^{nom} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta \boldsymbol{x}_k^{nom} + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.58}$$

with $\Delta \boldsymbol{x}_k^{nom} = \boldsymbol{x}_k^{nom} - \boldsymbol{x}_k^{tr}$, and $\boldsymbol{g}_k$ being the nominal control input. Therefore, the complete control policy for the uncertain system can be defined as

$$\Delta \boldsymbol{u}_k = \boldsymbol{K}\left(\boldsymbol{\zeta}_k\right)\boldsymbol{e}_k + \boldsymbol{g}_k, \tag{4.59}$$

with $\boldsymbol{e}_k = \Delta \boldsymbol{x}_k - \Delta \boldsymbol{x}_k^{nom}$ being the mismatch error.

As for equation (4.25), considering the models (4.57) and (4.24) together with the definition (4.59), the mismatch error model can be written as

$$\boldsymbol{e}_{k+1} = \left[\boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}\left(\boldsymbol{\zeta}_k\right)\right]\boldsymbol{e}_k + \boldsymbol{w}_k. \tag{4.60}$$

4.3.2 Mismatch error adaptive controller

The procedure to obtain the mismatch error adaptive controller can be seen at subsection 4.2.2, but it was briefly presented here to made the section self-contained.

Let $V\left(\boldsymbol{e}_k\right) = \boldsymbol{e}_k^T \boldsymbol{P} \boldsymbol{e}_k$ be a Lyapunov function for the mismatch error system. Moreover, as considered for the RMPC, let $V\left(\boldsymbol{e}_k\right)$ be an upper bound for LQR cost function. Thus, regarding the conditions for asymptotic stability,

$$V\left(\boldsymbol{e}_k\right) \geq 0 \quad \text{and} \quad V\left(\boldsymbol{e}_{k+1}\right) - V\left(\boldsymbol{e}_k\right) < 0, \tag{4.61}$$

the follow inequalities can be written

$$\boldsymbol{P} > 0, \tag{4.62}$$

$$\boldsymbol{A}_f^T \boldsymbol{P} \boldsymbol{A}_f + \boldsymbol{K}\left(\boldsymbol{\zeta}\right)^T \boldsymbol{\mathcal{R}} \boldsymbol{K}\left(\boldsymbol{\zeta}\right) + \boldsymbol{\mathcal{Q}} - \boldsymbol{P} \leq 0, \tag{4.63}$$

with $\boldsymbol{A}_f = \boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}(\boldsymbol{\zeta}_k)$.

Hence, an LMI condition can be defined from (4.63), yielding to

$$\begin{bmatrix} \boldsymbol{S} & \boldsymbol{H}^T & \boldsymbol{S}\boldsymbol{\mathcal{Q}}^{\frac{1}{2}} & \boldsymbol{Y}(\boldsymbol{\zeta})^T\boldsymbol{\mathcal{R}}^{\frac{1}{2}} \\ \boldsymbol{H} & \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\mathcal{Q}}^{\frac{1}{2}}\boldsymbol{S} & \boldsymbol{0} & \mathbb{I} & \boldsymbol{0} \\ \boldsymbol{\mathcal{R}}^{\frac{1}{2}}\boldsymbol{Y}(\boldsymbol{\zeta}) & \boldsymbol{0} & \boldsymbol{0} & \mathbb{I} \end{bmatrix} \geq 0, \tag{4.64}$$

where $\boldsymbol{S} = \boldsymbol{P}^{-1}$, $\boldsymbol{Y}(\boldsymbol{\zeta}) = \boldsymbol{K}(\boldsymbol{\zeta})\boldsymbol{P}^{-1}$, $\boldsymbol{H} = \boldsymbol{A}(\boldsymbol{\zeta})\boldsymbol{S} + \boldsymbol{B}(\boldsymbol{\zeta})\boldsymbol{Y}(\boldsymbol{\zeta})$, and $\boldsymbol{\mathcal{Q}} > 0$ and $\boldsymbol{\mathcal{R}} > 0$ are, respectively, weighting matrices for the state error and the control effort.

In order to obtain a matrix $\boldsymbol{P}$ that shapes a Lyapunov function to fulfill the conditions (4.61), the follow optimization problem must be considered

$$\begin{aligned} \min_{\boldsymbol{S}>0, \boldsymbol{Y}(\boldsymbol{\zeta})\forall\boldsymbol{\zeta}} \quad & \mathrm{Tr}(\boldsymbol{P}), \\ \text{subject to} \quad & (4.64) \ \ \forall\boldsymbol{\zeta} \in \Delta, \end{aligned} \tag{4.65}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace operator.

Finally, having $\boldsymbol{P}$ and assuming the system's current realization known, the feedback gain for the mismatch error adaptive controller is given by

$$\boldsymbol{K}(\boldsymbol{\zeta}_k) = -\left(\boldsymbol{R} + \boldsymbol{B}(\boldsymbol{\zeta}_k)^T\boldsymbol{P}\boldsymbol{B}(\boldsymbol{\zeta}_k)\right)^{-1}\boldsymbol{B}(\boldsymbol{\zeta}_k)^T\boldsymbol{P}\boldsymbol{A}(\boldsymbol{\zeta}_k). \tag{4.66}$$

4.3.3 Constraints

As for the EMPC, the trajectory tracking must be performed considering the collision-free trajectory $\boldsymbol{\xi}^{goal} \in \mathbb{R}^3$ instead of the initial proposed one $\boldsymbol{\xi}^{tr} \in \mathbb{R}^3$. Thus, the uncertain linear model (4.57) can be rewritten as

$$\Delta\tilde{\boldsymbol{x}}_{k+1} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta\tilde{\boldsymbol{x}}_k + \boldsymbol{B}(\boldsymbol{\zeta}_k)\Delta\boldsymbol{u}_k + \boldsymbol{w}_k, \tag{4.67}$$

where $\Delta\tilde{\boldsymbol{x}}_k = \boldsymbol{x}_k - \boldsymbol{x}_k^{goal}$ with $\boldsymbol{x}_k^{goal}$ being the desired state vector $\boldsymbol{x}^{tr}$ with the initial position trajectory $\boldsymbol{\xi}^{tr}$ replaced by the obstacle-free position trajectory $\boldsymbol{\xi}^{goal}$.

Moreover, considering $\mathbb{W} = \{0\}$ for the model (4.67), the nominal model is given by

$$\Delta\tilde{\boldsymbol{x}}_{k+1}^{nom} = \boldsymbol{A}(\boldsymbol{\zeta}_k)\Delta\tilde{\boldsymbol{x}}_k^{nom} + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.68}$$

where $\Delta\tilde{\boldsymbol{x}}_k^{nom} = \boldsymbol{x}_k^{nom} - \boldsymbol{x}_k^{goal}$. Thus, the control input for the uncertain system can be rewritten as

$$\Delta\boldsymbol{u}_k = \boldsymbol{K}(\boldsymbol{\zeta}_k)\,\tilde{\boldsymbol{e}}_k + \boldsymbol{g}_k, \tag{4.69}$$

with $\tilde{\boldsymbol{e}}_k = \Delta\tilde{\boldsymbol{x}}_k - \Delta\tilde{\boldsymbol{x}}_k^{nom}$.

Further, to shape the EMPC pre-stabilizing strategy constraints, consider the reachable

sets obtained through the iterative procedure

$$\mathscr{R}_{k+i+1} = (\boldsymbol{A}(\boldsymbol{\zeta}_k) + \boldsymbol{B}(\boldsymbol{\zeta}_k)\boldsymbol{K}(\boldsymbol{\zeta}_k))\,\mathscr{R}_{k+i} \oplus \mathbb{W}, \quad \forall i = 0, \cdots, N_p - 1, \tag{4.70}$$

with N_p being the prediction horizon and $\mathscr{R}_k = \{0\}$.

The same constraints presented at section 4.1.3 for the EMPC are considered to formulate the REMPC pre-stabilizing control policy. However, in order to take the uncertainties into account, the reachable sets are used to shape the constraints into tighter ones.

1. Initial condition constraint:

   The system is assumed to start without any state error and collision. Thus,

   $$\Delta\tilde{\boldsymbol{x}}_0^{nom} = \mathbf{0}. \tag{4.71}$$

   Further, the last computed collision-free trajectory is considered as the initial condition

   $$\Delta\tilde{\boldsymbol{x}}_k^{nom} = \boldsymbol{x}_k^{nom} - \boldsymbol{x}_k^{goal}. \tag{4.72}$$

2. Model constraint:

   Aiming to consider the prediction process into the optimization problem and ensure feasibility to the collision-free trajectory, the model constraint is defined as

   $$\Delta\tilde{\boldsymbol{x}}_{k+i+1}^{nom} = \boldsymbol{A}(\boldsymbol{\zeta})\Delta\tilde{\boldsymbol{x}}_{k+i}^{nom} + \boldsymbol{B}(\boldsymbol{\zeta})\boldsymbol{g}_{k+i}, \quad \forall i = 1, \cdots, N_p. \tag{4.73}$$

3. Maximum state error constraint:

   Let $\mathbb{E} \in \mathbb{R}^n$ be the trajectory tracking error bounding set for the uncertain system (4.67). Thus,

   $$\Delta\tilde{\boldsymbol{x}}_k \in \mathbb{E}. \tag{4.74}$$

   Therefore, using the reachable sets (4.70) to reshape $\mathbb{E}$ as $\bar{\mathbb{E}}_i = \mathbb{E} \ominus \mathscr{R}_i$ $\forall i = 1, \cdots, N_p$, the maximum state error constraint can be defined as

   $$\Delta\tilde{\boldsymbol{x}}_k^{nom} \in \bar{\mathbb{E}}_i, \quad \forall i = 1, \cdots, N_p. \tag{4.75}$$

4. Input signal constraints:

   Similarly to the state error constraint, let $\mathbb{V} \in \mathbb{R}^m$ define a set of admissible control

inputs for the uncertain system. Thus,

$$\Delta \boldsymbol{u}_k \in \mathbb{V}. \tag{4.76}$$

Using the reachable sets to redefine $\mathbb{V}$ as $\bar{\mathbb{V}}_i = \mathbb{V} \ominus \boldsymbol{K}(\boldsymbol{\zeta}_k)\mathscr{R}_i \ \forall i = 0, \cdots, N_c - 1$, the constraint can be expressed as

$$\boldsymbol{g}_{k+i} \in \bar{\mathbb{V}}_i, \qquad \forall i = 0, \cdots, N_c - 1. \tag{4.77}$$

5. Terminal set constraint:

   Aiming closed-loop stability, a terminal constraint for the uncertain system is considered

   $$\Delta \tilde{\boldsymbol{x}}_{k+N_p} \in \boldsymbol{\Omega}, \tag{4.78}$$

   where $\boldsymbol{\Omega}$ is a maximal robust control invariant set that can be obtained through the procedure presented at the subsection 4.2.5.

   Further, the last element of the reachable set sequence $\mathscr{R}_{k+N_p}$ is used to modify the terminal constraint (4.78) in order to be used in the EMPC pre-stabilizing control strategy. Therefore,

   $$\Delta \tilde{\boldsymbol{x}}^{nom}_{k+N_p} \in \boldsymbol{\Omega} \ominus \mathscr{R}_{k+N_p}. \tag{4.79}$$

6. No-fly zone constraints:

   Finally, to strictly ensure that no collision will occur, the no-fly zone constraints are considered

   $$\boldsymbol{\xi}_{k+i} \in \mathbb{R}^3 - \mathbb{O}, \qquad \forall i = 1, \cdots, N_p, \tag{4.80}$$
   $$\boldsymbol{\xi}^{goal}_{k+i} \in \mathbb{R}^3 - \mathbb{O}, \qquad \forall i = 1, \cdots, N_p, \tag{4.81}$$

   with $\boldsymbol{\xi}_{k+i}$ denoting the position and $\mathbb{O}$ being a set representing all points in the workspace that are obstructed by obstacles.

4.3.4 EMPC strategy

Having the constraints defined in the previous subsection, an EMPC strategy to deal with the nominal control problem can be designed. Hence, consider the economic cost function

$$\begin{aligned} \mathcal{J} = &\sum_{i=0}^{N_p-1} \left\|\Delta \tilde{\boldsymbol{x}}^{nom}_{k+i}\right\|^2_{\boldsymbol{Q}} + \sum_{j=0}^{N_c-1} \left\|\boldsymbol{g}_{k+j}\right\|^2_{\boldsymbol{\mathcal{R}}} + \left\|\Delta \tilde{\boldsymbol{x}}^{nom}_{k+N_p}\right\|^2_{\boldsymbol{P}} + \\ &\sum_{i=0}^{N_p} \left\|\boldsymbol{\xi}^{goal}_{k+i} - \boldsymbol{\xi}^{tr}_{k+i}\right\|^2_{\boldsymbol{\kappa}} + \sum_{l} \sum_{i=0}^{N_p} \chi_l \left\| \frac{1}{\min\left\{\left\|\boldsymbol{\xi}_{k+i} - \boldsymbol{\xi}^{obs}_l\right\|\right\}} - \frac{1}{d^*_l} \right\|^2_{\boldsymbol{\lambda}}, \quad \forall l, \end{aligned} \tag{4.82}$$

where N_c is the control horizon, $\boldsymbol{P}$ is the Lyapunov matrix obtained in the equation (4.65), $\boldsymbol{\kappa}$ and $\boldsymbol{\lambda}$ are weighting matrices, $\boldsymbol{\xi}_l^{obs}$ denotes the position of the l-th obstacle, and

$$\chi_l = \begin{cases} 1, & \text{if } \min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} \leq d_l^*, \\ 0, & \text{if } \min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} > d_l^*, \end{cases} \tag{4.83}$$

with $c \in \mathbb{N}$ being a scaling factor and d_l^* being a safe distance from the l-th obstacle and $\min\left\{d_l\left(\boldsymbol{\xi}_{k+N_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\}$ denoting the smaller Euclidean distance between $\boldsymbol{\xi}_{k+N_p}^{tr}$ and $\boldsymbol{\xi}_l^{obs}$.[3]

Hence, the pre-stabilizing control policy for the REMPC strategy can be defined through the optimization problem as

$$\begin{aligned} &\min_{\underset{\rightarrow}{\boldsymbol{g}}, \underset{\rightarrow}{\boldsymbol{x}}, \underset{\rightarrow}{\boldsymbol{x}}^{goal}} \quad \mathcal{J}, \\ &\text{subject to} \quad (4.71 - 4.73), (4.75), (4.77), (4.79 - 4.81), \end{aligned} \tag{4.84}$$

with $\underset{\rightarrow}{\boldsymbol{g}} = \begin{bmatrix} \boldsymbol{g}_k^T & \cdots & \boldsymbol{g}_{k+N_c-1}^T \end{bmatrix}^T$ and $\underset{\rightarrow}{\boldsymbol{x}} = \begin{bmatrix} (\boldsymbol{x}_{k+1})^T & \cdots & (\boldsymbol{x}_{k+N_p})^T \end{bmatrix}^T$.

4.4 Tilt-rotor UAV trajectory tracking control

In this section the trajectory tracking problem of a Tilt-rotor UAV carrying a suspended load is solved from the UAV's perspective using the three controllers previously designed in this chapter. The EMPC proposed at section 4.1 solves the trajectory tracking problem while performing obstacle avoidance through a whole-body control approach, i.e, controlling all the UAV's degrees of freedom. The RMPC presented at section 4.2 is used as an outer-loop controller for the cascade structure shown at Appendix B to perform trajectory tracking of the UAV's planar motion while stabilizing the load, i.e., to control x, y, γ_1, and γ_2 dynamics. Moreover, similar to the RMPC, the REMPC presented at section 4.3 is used as an outer-loop controller to regulate the UAV's planar motion and stabilizes the suspended load while performing obstacle avoidance in the $x - y$ plane.

The necessity to work with a cascade structure when using the tube-based controllers is due to dimensionality problems, which are common in MPC strategies but it is worse in the tube-based approaches. Mainly, this happens because of the computational cost associated with the Minkowski sum and the Pontryagin difference, which increases with the control system's state-space dimension. While in the whole-body approach the predictive controller must control at least twenty degrees of freedom used to describe the Tilt-rotor with suspended load dynamic model, using the cascade structure the outer controller only deals with eight of them. Afterwards, all the remaining degrees of freedom that are not controlled by the predictive controllers are addressed by a nonlinear IOFL inner-loop controller. For more details about the dynamic model from the UAV's perspective and

[3] Further information about how the terms of the nonlinear economic cost are derived can be found at section 4.1.

the cascade structure used in this section, the reader should refer, respectively, to the Appendices A and B.

Moreover, in this section, simulations will be carried out to corroborate the controller's trajectory tracking good performance and, for those with economic criteria, the obstacle avoidance ability. The simulations were made using the MATLAB/Simulink® environment with the help of the YALMIP solver and the MPT toolbox.

4.4.1 Discrete whole-body linearized model

The model (A.30), described in Appendix A, is considered to obtain the discrete whole-body linearized model to be used in the implementation of the EMPC. Thus, its equilibrium point must be found in order to expand (A.30) through first-order Taylor series.

Therefore, considering the vehicle in hovering without any external disturbances ($\boldsymbol{d} = \mathbf{0}$), the equilibrium point can be obtained after solving $\dot{\boldsymbol{x}}_{\mathcal{B}} = \boldsymbol{\varphi}_{\mathcal{B}}(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{B}}^T & \ddot{\boldsymbol{q}}_{\mathcal{B}}^T \end{bmatrix}^T = \mathbf{0}$, which leads to the system of algebraic equations

$$\boldsymbol{\vartheta}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}^{eq}\right) - \boldsymbol{G}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}^{eq}\right) = \mathbf{0}. \tag{4.85}$$

where $\boldsymbol{q}_{\mathcal{B}}^{eq} = [x_{\mathcal{B}}^{eq}\ y_{\mathcal{B}}^{eq}\ z_{\mathcal{B}}^{eq}\ \phi_{\mathcal{B}}^{eq}\ \theta_{\mathcal{B}}^{eq}\ \psi_{\mathcal{B}}^{eq}\ \alpha_R^{eq}\ \alpha_L^{eq}\ \gamma_1^{eq}\ \gamma_2^{eq}]^T$.

The algebraic problem (4.85) can be solved by an infinity set of real numbers since it has more variables than equations. Thus, let the states $x_{\mathcal{B}}$, $y_{\mathcal{B}}$, and $z_{\mathcal{B}}$ assume any values and $\psi_{\mathcal{B}} = 0$.[4] Therefore, considering the physical parameters defined on table A.1, the equilibrium values for the UAV's states and inputs are given by

$$\begin{aligned} &\phi^{eq} = -0.000154,\ \theta^{eq} = -0.0411,\ \alpha_R^{eq} = 0.0411,\ \alpha_L^{eq} = 0.0409,\ \gamma_1^{eq} = 0.000153,\ \gamma_2^{eq} = 0.0409, \\ &f_R^{eq} = 10.1838,\ f_L^{eq} = 10.2197,\ \tau_{\alpha_R}^{eq} = 0,\ \tau_{\alpha_L}^{eq} = 0. \end{aligned} \tag{4.86}$$

Let $\boldsymbol{x}^{eq}$ and $\boldsymbol{u}^{eq}$ denote, respectively, the equilibrium state and input vectors able to maintain the system in hovering. Then, the equations of motion (A.30) can be linearized around these points, yielding to

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}\Delta\boldsymbol{x} + \boldsymbol{B}\Delta\boldsymbol{u}, \tag{4.87}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{eq}$, $\Delta\boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}^{eq}$, and

$$\boldsymbol{A} = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{eq} \\ \boldsymbol{u}=\boldsymbol{u}^{eq}}} \in \mathbb{R}^{20\times 20}, \qquad \boldsymbol{B} = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{d})}{\partial\boldsymbol{u}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{eq} \\ \boldsymbol{u}=\boldsymbol{u}^{eq}}} \in \mathbb{R}^{20\times 4}, \tag{4.88}$$

being $\boldsymbol{x}^{eq} = [(\boldsymbol{q}^{eq})^T\ (\dot{\boldsymbol{q}}^{eq})^T]^T$ and $\boldsymbol{u}^{eq} = [f_R^{eq}\ f_L^{eq}\ \tau_{\alpha_R}^{eq}\ \tau_{\alpha_L}^{eq}]^T$, with $\boldsymbol{q}^{eq} = [0\ 0\ 0\ \phi^{eq}\ \theta^{eq}\ 0\ \alpha_{\mathrm{R}}{}^{eq}\ \alpha_{\mathrm{L}}{}^{eq}\ \gamma_1^{eq}\ \gamma_2^{eq}]^T$.

[4] From now on, the subscript $\mathcal{B}$ used to differ the system from the UAV's perspective from the load's perspective will be dropped, since this chapter deals only with the first one.

Similar to Chapter 3, to improve the trajectory tracking performance, the error state vector $\Delta\boldsymbol{x}$ is augmented with integral actions, yielding to

$$\Delta\hat{\boldsymbol{x}} = \begin{bmatrix} \Delta\boldsymbol{x} \\ \int (\boldsymbol{\xi} - \boldsymbol{\xi}^{tr}) \\ \int (\psi - \psi^{tr}) \end{bmatrix} \in \mathbb{R}^{24}, \tag{4.89}$$

whose dynamics are given by

$$\Delta\dot{\hat{\boldsymbol{x}}} = \underbrace{\left[\begin{array}{cccccc|c|c} \multicolumn{6}{c|}{\boldsymbol{A}} & & \mathbf{0}_{20\times4} \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & & \\ 0 & 1 & 0 & 0 & 0 & 0 & \mathbf{0}_{4\times14} & \mathbf{0}_{4\times4} \\ 0 & 0 & 1 & 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & 0 & 1 & & \end{array}\right]}_{\hat{\boldsymbol{A}}} \Delta\hat{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \boldsymbol{B} \\ \mathbf{0}_{4\times4} \end{bmatrix}}_{\hat{\boldsymbol{B}}} \Delta\boldsymbol{u}. \tag{4.90}$$

Finally, the discrete whole-body linearized model can be obtained after map the model (4.90) from the continuous-time to the discrete-time domain, which yields to

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}_d \Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}_d \Delta\boldsymbol{u}_k, \tag{4.91}$$

being the matrices $\hat{\boldsymbol{A}}_d$ and $\hat{\boldsymbol{B}}_d$ obtained after discretizing the model using a zero-order hold with sampling time $T_s = 12$ ms.

The trajectory to be tracked by the suspended load and its control signal are defined as

$$\boldsymbol{x}_k^{tr} = [(\boldsymbol{q}_k^{tr})^T \ (\dot{\boldsymbol{q}}_k^{tr})^T]^T, \tag{4.92}$$

$$\boldsymbol{u}_k^{\mathrm{tr}} = \boldsymbol{\vartheta}^+ \left[\boldsymbol{M}\ddot{\boldsymbol{q}}_k^{\mathrm{tr}} + \boldsymbol{C}\dot{\boldsymbol{q}}_k^{\mathrm{tr}} + \boldsymbol{G}\right], \tag{4.93}$$

where $(\cdot)^+$ denotes the left pseudoinverse, and $\boldsymbol{q}_k^{\mathrm{tr}}$, $\dot{\boldsymbol{q}}_k^{\mathrm{tr}}$ and $\ddot{\boldsymbol{q}}_k^{\mathrm{tr}}$ are provided reference signals with $\boldsymbol{q}_k^{tr} = [x_k^{tr} \ y_k^{tr} \ z_k^{tr} \ \phi^{eq} \ \theta^{eq} \ \psi^{tr} \ \gamma_1^{eq} \ \gamma_2^{eq} \ \alpha_{\mathrm{R}}{}^{eq} \ \alpha_{\mathrm{L}}{}^{eq}]^T$. Notice that $\boldsymbol{u}_k^{tr}$, since it is computed using a left pseudoinverse, will be an exact solution to the dynamic equations (A.30) $\forall k \in \mathbb{N}$ only if the trajectory is feasible.

4.4.2 Discrete outer-loop linearized model

The second-order dynamic model (B.31), obtained in Appendix B, is used to derive the outer-loop linearized model necessary to implement the RMPC and REMPC strategies. Therefore, in order to expand the outer system in Taylor series, the model (B.31) must be rewritten in a state-space representation and, later, its equilibrium point must be found.

Thus, consider the state-space representation

$$\dot{\boldsymbol{x}}_{\boldsymbol{q}_2} = \boldsymbol{\varphi}_{\boldsymbol{q}_2}\left(\boldsymbol{x}_{\boldsymbol{q}_2}, \boldsymbol{\Upsilon}_{\boldsymbol{q}_2}, \boldsymbol{\delta}_{\boldsymbol{q}_2}\right) = \begin{bmatrix} \dot{\boldsymbol{q}}_2 \\ \ddot{\boldsymbol{q}}_2 \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}}_2 \\ \boldsymbol{M}_{\boldsymbol{q}_2}^{-1}\left[\boldsymbol{\Upsilon}_{\boldsymbol{q}_2} + \boldsymbol{\delta}_{\boldsymbol{q}_2} - \bar{\boldsymbol{n}}_{\boldsymbol{q}_2}\right] \end{bmatrix}, \tag{4.94}$$

where $\boldsymbol{\Upsilon}_{\boldsymbol{q}_2}$ is an intermediary control input vector, and $\boldsymbol{\delta}_{\boldsymbol{q}_2}$ is a vector containing uncertainties and external disturbances (see Appendix B, section B.3).[5]

Moreover, considering the vehicle in hover-flight and not affected by any external disturbances, i.e., $\dot{\boldsymbol{x}} = \boldsymbol{0}$ and $\boldsymbol{\delta}_{\boldsymbol{q}_2} = \boldsymbol{0}$, the equilibrium point for the system (4.94) can be obtained by solving the algebraic problem

$$\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\Upsilon}, \boldsymbol{\delta}) = \begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \boldsymbol{0}, \tag{4.95}$$

which solution has an infinity set of real numbers since it has more variables than equations. Therefore, letting x and y assume any value and considering the physical parameters defined on table A.1, the solution of the algebraic system (4.95) gives uniquely the follow equilibrium values

$$\gamma_1^{eq} = 0.000154,\ \gamma_2^{eq} = 0.0411,\ \Upsilon_1^{eq} = 0,\ \Upsilon_2^{eq} = 0. \tag{4.96}$$

Let $\boldsymbol{x}^{tr}$ and $\boldsymbol{\Upsilon}^{tr}$ denote, respectively, the state vector trajectory and the intermediary control inputs able to drive the system along this trajectory. Then, the state-space equations (4.94) can be linearized around these trajectories, resulting in the model

$$\Delta\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{\zeta}(t))\Delta\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\zeta}(t))\Delta\boldsymbol{\Upsilon}, \tag{4.97}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}^{tr}$, $\Delta\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon} - \boldsymbol{\Upsilon}^{tr}$, $\boldsymbol{\zeta}(t)$ denotes a vector of time-varying model parameters. The linearized Jacobians are given by

$$\boldsymbol{A}(\boldsymbol{\zeta}(t)) = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\Upsilon}, \boldsymbol{\delta})}{\partial\boldsymbol{x}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{tr} \\ \boldsymbol{\Upsilon}=\boldsymbol{\Upsilon}^{tr}}} \in \mathbb{R}^{8\times 8}, \quad \boldsymbol{B}(\boldsymbol{\zeta}(t)) = \left.\frac{\partial\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{\Upsilon}, \boldsymbol{\delta})}{\partial\boldsymbol{\Upsilon}}\right|_{\substack{\boldsymbol{x}=\boldsymbol{x}^{tr} \\ \boldsymbol{\Upsilon}=\boldsymbol{\Upsilon}^{tr}}} \in \mathbb{R}^{8\times 2}. \tag{4.98}$$

In this work, the trajectory values for $\boldsymbol{x}$ and $\boldsymbol{\Upsilon}$ are given by

$$\boldsymbol{x}^{tr} = [(\boldsymbol{q}^{tr})^T\ (\dot{\boldsymbol{q}}^{tr})^T]^T, \tag{4.99}$$

$$\boldsymbol{\Upsilon}^{tr} = \boldsymbol{M}\ddot{\boldsymbol{q}}^{tr} + \bar{\boldsymbol{n}}, \tag{4.100}$$

where $\boldsymbol{q}^{\mathrm{tr}}$, $\dot{\boldsymbol{q}}^{\mathrm{tr}}$, and $\ddot{\boldsymbol{q}}^{\mathrm{tr}}$ are provided reference signals with $\boldsymbol{q}^{tr} = [x^{tr}(t)\ y^{tr}(t)\ \gamma_1^{eq}\ \gamma_2^{eq}]^T$.

[5] From now on, the subscript $\boldsymbol{q}_2$ used to differ the outer-loop from the inner-loop system will be dropped to simplify the notation. The reader can easily understand by context when a variable, for instance $\boldsymbol{x}$ and $\boldsymbol{q}$, is being used as a whole-body system's variable or an outer-loop system's variable since the first one refers only to the EMPC strategy and the latter to the RMPC and REMPC strategies.

Considering the state vector trajectory defined in (4.99), the time-varying parameters become $\boldsymbol{\zeta}(t) = [\ddot{x}^{tr}(t)\ \ddot{y}^{tr}(t)]^T$, which is limited by the polytope $\Delta \in \mathbb{R}^2$ with 2^2 vertices. Therefore, the linearized Jacobian matrices can be rewritten in a convex polytopic representation as

$$\boldsymbol{A}(\boldsymbol{\zeta}(t)) = \sum_{i=1}^{4} \iota_i \boldsymbol{A}_i, \tag{4.101}$$

$$\boldsymbol{B}(\boldsymbol{\zeta}(t)) = \sum_{i=1}^{4} \iota_i \boldsymbol{B}_i, \tag{4.102}$$

with $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ being, respectively, the matrices $\boldsymbol{A}(\boldsymbol{\zeta}(t))$ and $\boldsymbol{B}(\boldsymbol{\zeta}(t))$ evaluated at the i-th vertex of Δ. Moreover, the constraints $0 \leq \iota_1 \leq 1$ and $\sum_{i=1}^{n} \iota_i = 1$ must hold for $i = 1, \cdots, 4$.

Aiming to improve the trajectory tracking performance of the states x and y, the state vector $\Delta\boldsymbol{x}$ is augmented with integral actions as

$$\Delta\hat{\boldsymbol{x}} = \begin{bmatrix} \Delta\boldsymbol{x} \\ \int (x - x^{tr}) \\ \int (y - y^{tr}) \end{bmatrix} \in \mathbb{R}^{10}, \tag{4.103}$$

whose dynamics are given by

$$\Delta\dot{\hat{\boldsymbol{x}}} = \underbrace{\left[\begin{array}{cc|c|c} \multicolumn{2}{c|}{\boldsymbol{A}(\boldsymbol{\zeta}(t))} & & \boldsymbol{0}_{8\times 2} \\ \hline 1 & 0 & \multirow{2}{*}{$\boldsymbol{0}_{2\times 6}$} & \multirow{2}{*}{$\boldsymbol{0}_{2\times 2}$} \\ 0 & 1 & & \end{array}\right]}_{\hat{\boldsymbol{A}}(\boldsymbol{\zeta}(t))} \Delta\hat{\boldsymbol{x}} + \underbrace{\begin{bmatrix} \boldsymbol{B}(\boldsymbol{\zeta}(t)) \\ \boldsymbol{0}_{2\times 2} \end{bmatrix}}_{\hat{\boldsymbol{B}}(\boldsymbol{\zeta}(t))} \Delta\boldsymbol{\Upsilon}. \tag{4.104}$$

Finally, the discrete outer-loop linearized model can be obtained after map the model (4.104) from the continuous-time to the discrete-time domain, which yields to

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\Delta\boldsymbol{\Upsilon}_k, \tag{4.105}$$

being the matrices $\hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)$ and $\hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)$ obtained after discretizing the model using a zero-order hold with sampling time $T_s = 120$ ms.

4.4.3 Desired trajectory

Aiming to explore the capabilities of the three controllers proposed in this chapter, two trajectories to be tracked by the Tilt-rotor UAV carrying a suspended load are proposed. The first one intends to show the obstacle avoidance feature achieved with the economic criteria included in the optimization problem by means of the potential functions. In this trajectory the vehicle needs to displace itself along the $X^{\mathcal{I}}$-axis while avoiding three spherical obstacles that are obstructing its path (see Figure (4.4)). In this trajectory, the obstacle avoidance capability of the EMPC and the REMPC strategies are shown. The

second trajectory, on the other hand, is composed of several interconnected paths described by polynomial and sinusoidal functions (see Figure 4.5). In this case, the vehicle starts in hovering and tracks a square-like trajectory while performing yaw movements during the execution. The main goal of this trajectory is to show the RMPC trajectory tracking performance and robustness against modeling uncertainties and disturbances. Therefore, when performing the second trajectory, external forces are applied to the Tilt-rotor UAV. Figure 4.6 shows the disturbance profile for the desired trajectory, which may represent sustained wind gusts affecting the load. Additionally, to better evaluate the behavior of the proposed control strategies in the presence of uncertainties, measurement's noise is considered for both trajectories and assumed to have Gaussian probability distribution with zero mean and measurement error defined as three times the standard deviation. For simulation purposes, position errors are ±0.15 [m], angular position errors are ±0.02 [rad], velocity errors are ±0.01 [m/s], and angular velocity errors are ±0.002 [rad/s].

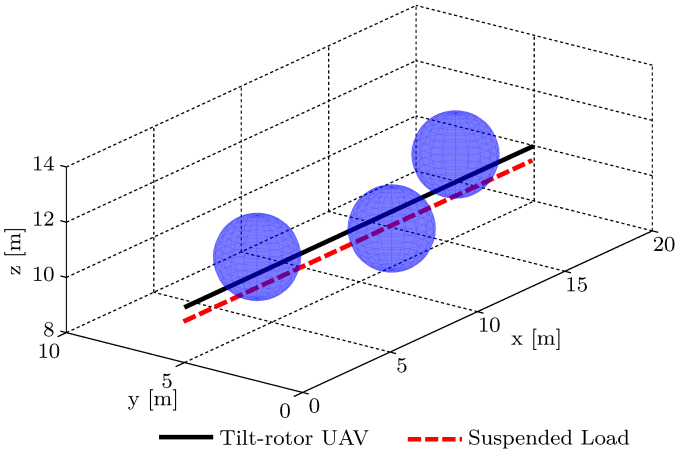


Figure 4.4: Three-dimensional representation of the first proposed trajectory, with the blue spheres being the obstacles.

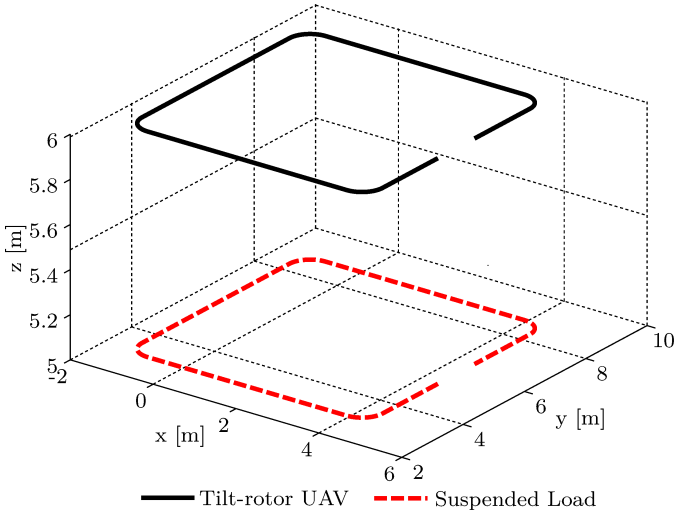


Figure 4.5: Three-dimensional representation of the second proposed trajectory.

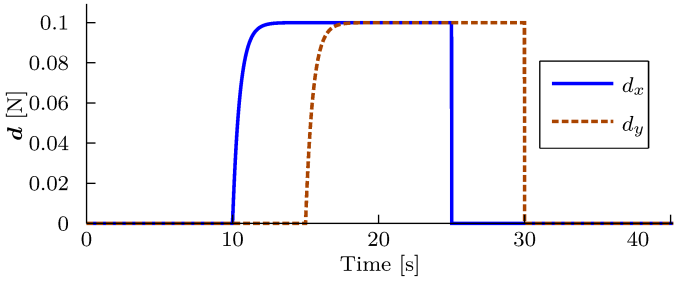


Figure 4.6: Profile of the disturbance forces applied to the Tilt-rotor UAV expressed in the inertial frame $\mathcal{I}$.

4.4.4 Economic MPC

Aiming to implement the EMPC strategy to solve the load transportation problem, the discrete-time augmented error model (4.91) is considered. However, as stated in the section 4.1, the state error vector must be modified in order to track the collision-free trajectory instead of the initially prosed one. Therefore, the state error vector can be rewritten as

$$\Delta\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^{goal} = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^{goal} \\ \int (\boldsymbol{\xi} - \boldsymbol{\xi}^{goal}) \\ \int (\psi - \psi^{tr}) \end{bmatrix}, \tag{4.106}$$

where $\boldsymbol{x}^{goal} = [(\boldsymbol{q}^{goal})^T \ \dot{\boldsymbol{q}}^T]^T$, $\boldsymbol{\xi}^{goal} = [x^{goal} \ y^{goal} \ z^{goal}]^T$, and $\boldsymbol{q}^{goal} = [(\boldsymbol{\xi}^{goal})^T \ \phi^{eq} \ \theta^{eq} \ \psi^{tr} \ \alpha_R^{eq} \ \alpha_L^{eq} \ \gamma_1^{eq} \ \gamma_2^{eq}]^T$, with the equilibrium values being given by (4.86).

Thus, the model (4.91) can be modified into

$$\Delta\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}_d \Delta\tilde{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}_d \Delta\boldsymbol{u}_k, \tag{4.107}$$

which is the error model used in the EMPC strategy implementation.

Finally, having the model (4.107) and assuming the obstacles position $\boldsymbol{\xi}_l^{obs} \in \mathbb{O}$ perfectly known for all l, the EMPC strategy for trajectory tracking of the Tilt-rotor UAV carrying a suspended load while performing obstacle avoidance can be described in general terms by the Algorithm 4.1.

Tunning parameters

Similar to Chapter 3, the Bryson's rule was used as starting point to synthesize the MPC's weighting matrices $\mathcal{Q}$ and $\mathcal{R}$, which are given by

$$\boldsymbol{\mathcal{Q}} = \text{diag}\left(\frac{10}{2^2}, \frac{10}{2^2}, \frac{10}{2^2}, \frac{1}{(\pi/2)^2}, \frac{1}{(\pi/2)^2}, \frac{15}{(\pi)^2}, \frac{1}{(\pi/2)^2}, \frac{1}{(\pi/2)^2}, \frac{30}{(\pi/2)^2}, \frac{30}{(\pi/2)^2}, \frac{5}{2^2}, \frac{5}{2^2}, \frac{5}{2^2}, \right.$$
$$\left. \frac{1}{(\pi/3)^2}, \frac{1}{(\pi/3)^2}, \frac{1}{(\pi/4)^2}, \frac{1}{(3\pi)^2}, \frac{1}{(3\pi)^2}, \frac{15}{(3\pi)^2}, \frac{15}{(3\pi)^2}, 10, 10, 10, 5\right), \tag{4.108}$$

Algorithm 4.1 Economic MPC algorithm

1: Compute $\boldsymbol{P}$ through (3.17) using the model (4.107) and matrices $\mathcal{Q}$ and $\mathcal{R}$.
2: Calculate $\boldsymbol{\Omega}$ by iterating the one-step operator defined at (3.27).
3: Set $\boldsymbol{x}_1^{goal} = \boldsymbol{x}_1^{tr}$.
4: **procedure** EMPC($\boldsymbol{x}_k$, $\underrightarrow{\boldsymbol{q}}^{tr}$, $\underrightarrow{\dot{\boldsymbol{q}}}^{tr}$, $\underrightarrow{\ddot{\boldsymbol{q}}}^{tr}$, $\boldsymbol{\xi}_l^{obs}$, $\boldsymbol{x}_k^{goal}$)
5: Obtain the vector $\Delta\tilde{\boldsymbol{x}}_k$.
6: Compute $\underrightarrow{\boldsymbol{u}}^{tr}$ using (4.93) from k up to $k + N_c$.
7: Check if $\min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} \leq d_l^* \ \forall l$.
8: Write the cost function (4.12).
9: Write the constraints (4.13 – 4.20).
10: Solve the optimization problem (4.21) to obtain $\underrightarrow{\boldsymbol{u}}$ and $\underrightarrow{\boldsymbol{x}}^{goal}$.
11: Set $\boldsymbol{x}_k^{goal} = \boldsymbol{x}_{k+1}^{goal}$.
12: **return** $\boldsymbol{u}_k$ and $\boldsymbol{x}_k^{goal}$.
13: **end procedure**

$$\boldsymbol{\mathcal{R}} = \text{diag}\left(\frac{1}{(15 - f_R^{eq})^2}, \frac{1}{(15 - f_L^{eq})^2}, \frac{0.1}{(2 - \tau_{\alpha_R}^{eq})^2}, \frac{0.1}{(2 - \tau_{\alpha_L}^{eq})^2}\right). \tag{4.109}$$

Furthermore, the saturation level of the Tilt-rotor UAV actuators and the maximum state error between the vehicle and the colision-free trajectory, are

$$\begin{aligned}
&x - x^{goal} = [-1, 1], && y - y^{goal} = [-1, 1], && z - z^{goal} = [-1, 1],\\
&\phi - \phi^{eq} = [-0.5, 0.5], && \theta - \theta^{eq} = [-0.5, 0.5], && \psi - \psi^{tr} = [-0.5, 0.5],\\
&\alpha_R - \alpha_R^{eq} = [-0.5, 0.5], && \alpha_L - \alpha_L^{eq} = [-0.5, 0.5], && \gamma_1 - \gamma_1^{eq} = [-0.5, 0.5],\\
&\gamma_2 - \gamma_2^{eq} = [-0.5, 0.5], && f_R = [0, 15], && f_L = [0, 15],\\
&\tau_{\alpha_R} = [-2, 2], && \tau_{\alpha_L} = [-2, 2],
\end{aligned} \tag{4.110}$$

where the error limitations were chosen regarding measurement errors and the actuators' bounds depends on physical constraints.

The potential functions weighting matrices, obtained by try-and-error, are $\boldsymbol{\kappa} = 0.1 \cdot \mathbb{I}_{3\times3}$ and $\boldsymbol{\lambda} = 15 \cdot \mathbb{I}_{3\times3}$, and the security distance from the obstacle defining when the repulsive potential field starts to act is $d_l^* = 0.5$. Moreover, the prediction and control horizons are $N_p = 10$ and $N_c = 5$, and the scaling factor is $c = 10$.

Simulation results

The Figure 4.7 shows the Tilt-rotor UAV with suspended load performing a straight trajectory passing through three spherical obstacles. Is is possible to see that when the desired trajectory approaches the obstacles, the vehicle finds an alternative trajectory to follow in order to avoid collision. Further, after passing through the obstacle, the vehicle returns to the original trajectory. The first behavior can be explained by the action of the repulsive potential field and the second by the attractive potential field.

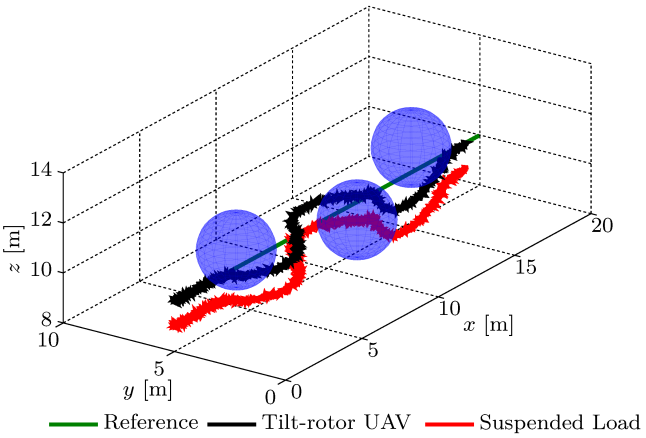


Figure 4.7: Trajectory tracking using the EMPC when performing the trajectory presented in Figure 4.4.

Figures 4.8 and 4.9 show, respectively, the projection of the trajectory presented in the Figure 4.7 decomposed in the $y-x$ and $z-x$ planes. Note that the Tilt-rotor UAV deviates from the first two obstacles by finding an alternative trajectory that maintains its altitude. On the other hand, the last obstacle is overcame by changing its altitude. This behavior can be explained since, using the EMPC strategy proposed in this section, the control system has three degrees of freedom to find a collision-free trajectory, i.e., it performs obstacle avoidance in the tree-dimensional space. Moreover, since the obstacle avoidance feature is obtained through the optimization of an economic stage cost, the controller is always looking for the alternative trajectory that minimizes the artificial potential field energy.

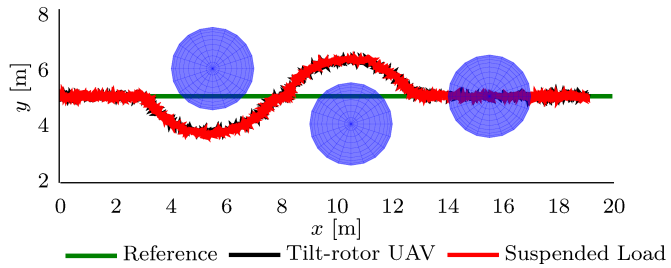


Figure 4.8: Projection in the $y-x$ plane of the trajectory shown in Figure 4.7.

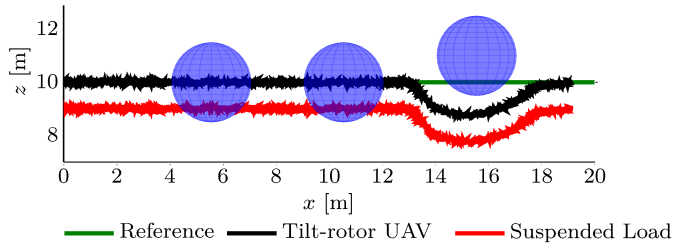


Figure 4.9: Projection in the $z-x$ plane of the trajectory shown in Figure 4.7.

Figure 4.10 presents the time evolution of the regulated variables x, y, z, and ψ. It is possible to notice the moment when the UAV stop following the desired trajectory in order to avoid collision. Observe that no yaw movement is performed since, like the LTI-MPC presented in the Chapter 3, the EMPC, as formulated in this chapter, cannot cope with yaw angle regulation. Also, notice in Figure 4.10 the presence of oscillatory behavior in the x evolution, this can be explained by the fact that, despite the controller being tracking the $\boldsymbol{\xi}^{goal}$, it still have the same initial velocity references. Furthermore, Figure 4.11 shows the trajectory tracking error between the vehicle and the collision-free trajectory, $\boldsymbol{\xi}^{goal}$.

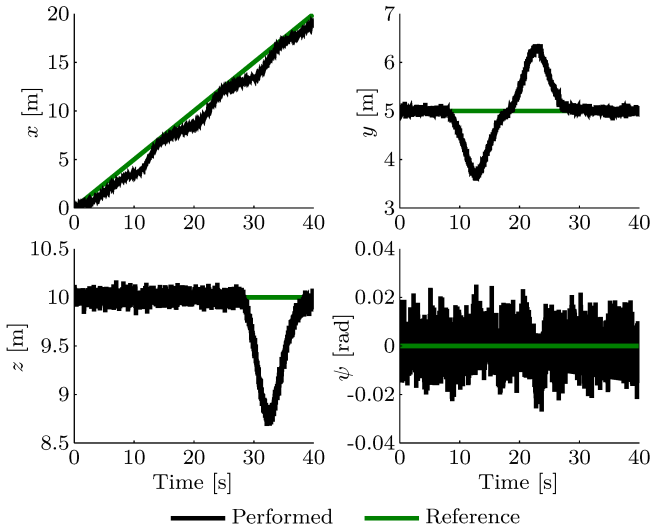


Figure 4.10: Time evolution of the regulated variables (x, y, z, ψ) and their desired trajectories.

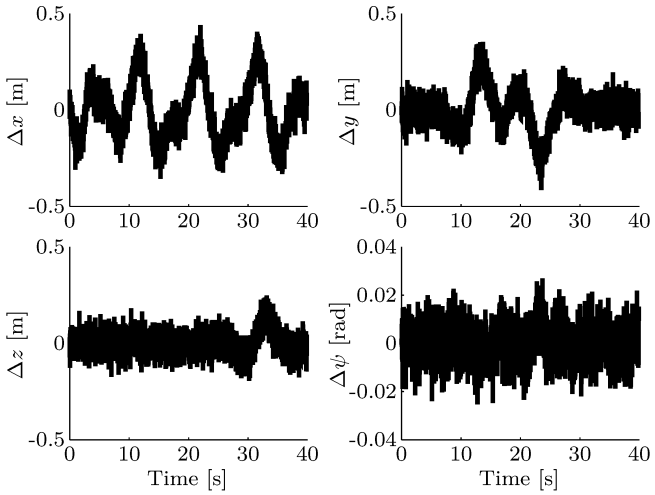


Figure 4.11: Time evolution of the tracking errors of the regulated variables (x, y, z, ψ).

Finally, Figure 4.12 shows the time evolution of the remaining degrees of freedom, which are kept stable throughout the simulation. Therefore, it is possible to state that the EMPC strategy was able to ensure closed-loop stability for the considered simulation scenario. Further, Figure 4.13 shows the actuator signals applied to the Tilt-rotor UAV.

Note that the controller was able to perform the trajectory tracking without reaching the actuators saturation level during the trajectory execution.

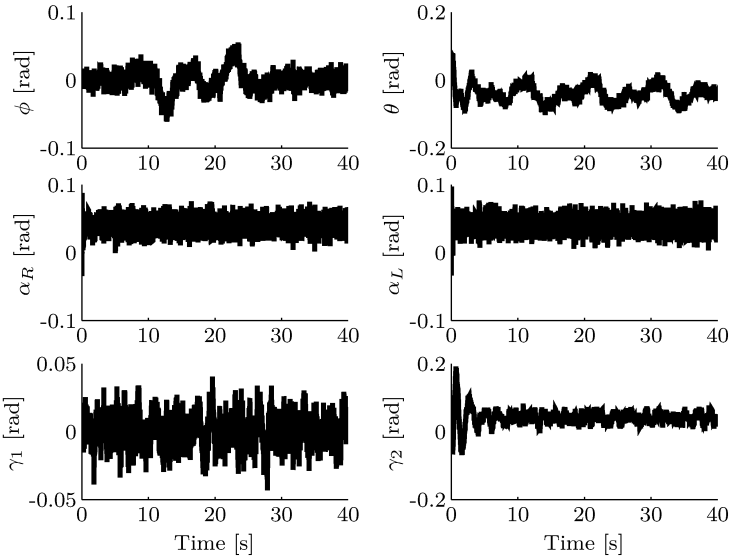


Figure 4.12: Time evolution of the remaining degrees of freedom.

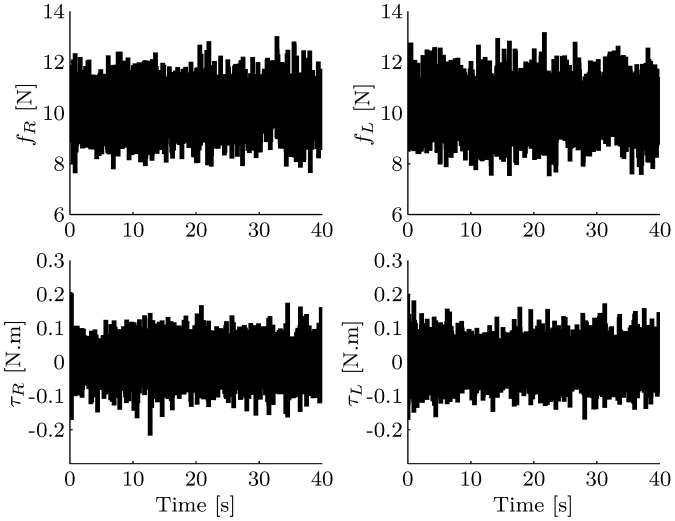


Figure 4.13: Applied thrusts and torques to the Tilt-rotor UAV.

4.4.5 Robust tube-based MPC

Aiming to implement the RMPC strategy to solve the load transportation problem through the cascade structure proposed at Appendix B, the augmented linear discrete-time outer-loop model (4.105) is used as the nominal model, i.e.,

$$\Delta\hat{\boldsymbol{x}}_{k+1}^{nom} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k^{nom} + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.111}$$

with $\boldsymbol{g}_k$ being the nominal model control input.

The uncertain system is described considering the additive uncertainty $\boldsymbol{w}$, which includes parametric uncertainties, unmodelled dynamics, and the external disturbance vector $\boldsymbol{\delta}$. Thus, the uncertain error model can be written by

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\Delta\boldsymbol{\Upsilon}_k + \boldsymbol{w}. \tag{4.112}$$

Moreover, the mismatch error is defined as

$$\boldsymbol{e}_k = \Delta\hat{\boldsymbol{x}}_k - \Delta\hat{\boldsymbol{x}}_k^{nom}. \tag{4.113}$$

Therefore, it is possible to describe in general terms the RMPC strategy used to control the Tilt-rotor UAV in the outer-loop by the Algorithm 4.2.

Algorithm 4.2 Robust Tube-Based MPC algorithm

1: Compute $\boldsymbol{P}$ through (4.37) using the matrices $\hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)$, $\hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)$, $\mathcal{Q}$, and $\mathcal{R}$.
2: Calculate $\boldsymbol{\Omega}$ by iterating the one-step operator defined at (4.50).
3: Set $\Delta\hat{\boldsymbol{x}}_0^{nom} = \Delta\hat{\boldsymbol{x}}_0 = \boldsymbol{0}$.
4: **procedure** RMPC($\boldsymbol{x}_k$, $\overrightarrow{\boldsymbol{q}^{tr}}$, $\overrightarrow{\dot{\boldsymbol{q}}^{tr}}$, $\overrightarrow{\ddot{\boldsymbol{q}}^{tr}}$)
5: Compute the feedback gain $\boldsymbol{K}(\boldsymbol{\zeta}_k)$ using (4.42).
6: Obtain the augmented vectors $\Delta\hat{\boldsymbol{x}}_k$ and $\Delta\hat{\boldsymbol{x}}_k^{nom}$.
7: Calculate the mismatch error $\boldsymbol{e}_k$.
8: Compute the reachable sets, $\mathscr{R}$, from k up to $k+N_p$ through (4.43).
9: Write the nominal model constraints using the equations (4.48), (4.49), and (4.51).
10: Solve the optimization problem (4.55) to obtain $\overrightarrow{\boldsymbol{g}}$.
11: Calculate $\Delta\boldsymbol{\Upsilon}_k = \boldsymbol{K}(\boldsymbol{\zeta}_k)\boldsymbol{e}_k + \boldsymbol{g}_k$.
12: Compute $\boldsymbol{\Upsilon}_k^{tr}$ using (4.100).
13: **return** $\boldsymbol{\Upsilon}_k$.
14: **end procedure**

After finished the execution of the RMPC algorithm, the intermediary control signal, $\boldsymbol{\Upsilon}_k$, is obtained. However, since the outer-loop must send roll and pitch reference angles to the inner-loop, equation (B.30) is necessary to obtain ϕ_k and θ_k from $\boldsymbol{\Upsilon}_k$.

Tunning parameters

In order to define the tuning parameters, it is important to find the maximum and minimum values allowed to the intermediary control input. Thus, considering the equation (B.29) evaluated at the values γ_1^{eq}, γ_2^{eq}, $\phi = [-0.5, 0.5]$, $\theta = [-0.5, 0.5]$, $f_R = [0, 15]$, and $f_L = [0, 15]$, the allowed range for $\boldsymbol{\Upsilon}$ is given by

$$\Upsilon_1 = [-13, 13], \quad \Upsilon_2 = [-15, 15]. \tag{4.114}$$

Moreover, considering the limits for the state error $\Delta x = [-1, 1]$, $\Delta y = [-1, 1]$, $\Delta\gamma_1 =$

$[-0.5, 0.5]$, and $\Delta\gamma_2 = [-0.5, 0.5]$, the bounding sets are given by

$$\begin{aligned}\mathbb{E} &= \{\Delta x, \Delta y \in \pm 1 \text{ m}, \Delta\gamma_1, \Delta\gamma_2 \in \pm 0.5 \text{ rad}\}, \\ \mathbb{V} &= \{\Upsilon_1 \in \pm 13, \Upsilon_2 \in \pm 15\}, \\ \mathbb{W} &= \{w_x, w_y \in \pm 0.1 \text{ m}, w_{\gamma_1}, w_{\gamma_1} \in \pm 0.01 \text{ rad}, \}, \end{aligned}$$

with $\mathbb{W}$ being chosen by try-and-error procedure.

Now, using the Bryson's rule, the weighting matrices $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{R}}$ can be described as

$$\boldsymbol{\mathcal{Q}} = \text{diag}\left(\frac{1}{2^2}, \frac{1}{2^2}, \frac{5}{(\pi/2)^2}, \frac{5}{(\pi/2)^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{(3\pi)^2}, \frac{1}{(3\pi)^2}, 10, 10\right), \tag{4.115}$$

$$\boldsymbol{\mathcal{R}} = \text{diag}\left(\frac{1}{(13 - \Upsilon_1^{eq})^2}, \frac{1}{(15 - \Upsilon_2^{eq})^2}\right). \tag{4.116}$$

Moreover, the prediction and control horizons, chosen considering a trade-off between good performance and computational cost, are $N_p = 5$ and $N_c = 2$. Further, the maximum absolute accelerations are 1 m/s^2, which gives the box-like polytope $\Delta = \{\pm 1, \pm 1\} \in \mathbb{R}^2$.

Simulation results

Figure 4.14 shows the results of the Tilt-rotor UAV carrying a suspended load while performing a square-like trajectory when affected by the disturbances shown in Figure 4.6. The RMPC is controlling the system planar motion and providing the load stabilization, while the nonlinear inner-loop IOFL controller is dealing with the altitude and the yaw regulation, and stabilizing the remaining degrees of freedom of the system.

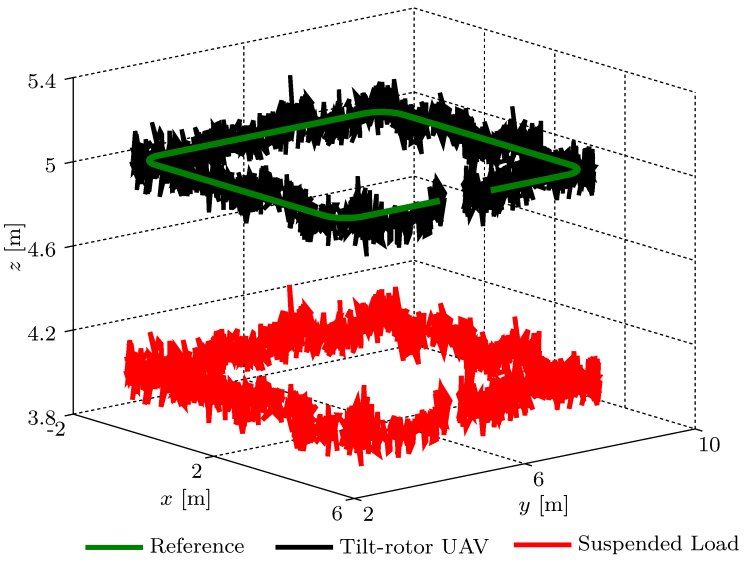


Figure 4.14: Trajectory tracking using the RMPC when performing the trajectory presented in Figure 4.5.

Figure 4.15 presents the time evolution of the regulated variables x, y, z, and ψ. Observe that the RMPC successfully performs trajectory tracking for the states x and y, as well as,

the nonlinear controller for the states z and ψ. Although the formulation of the RMPC presented in this chapter is not able to perform yaw's regulation, the cascade structure proposed to reduce the dimensionality problems makes not only the implementation of the tube-based controller possible but also allowed yaw movements. This allows the aircraft to perform trajectory tracking always head-on to the trajectory, which can be seen in Figure 4.15. Moreover, Figure 4.16 shows the tracking error of the regulated variables. Although some oscillations can be seen in Figure 4.15 and, consequently, in Figure 4.16, it should be notice that they are not sustained and their amplitude are small.

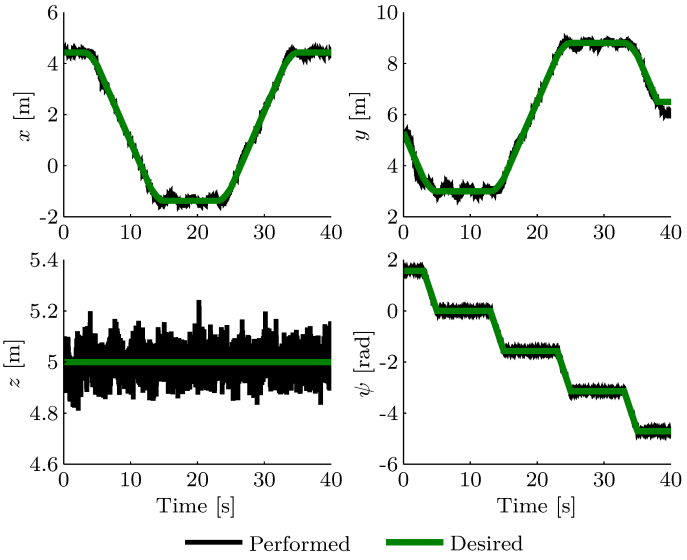


Figure 4.15: Time evolution of the regulated variables (x, y, z, ψ) when performing a square-like trajectory.

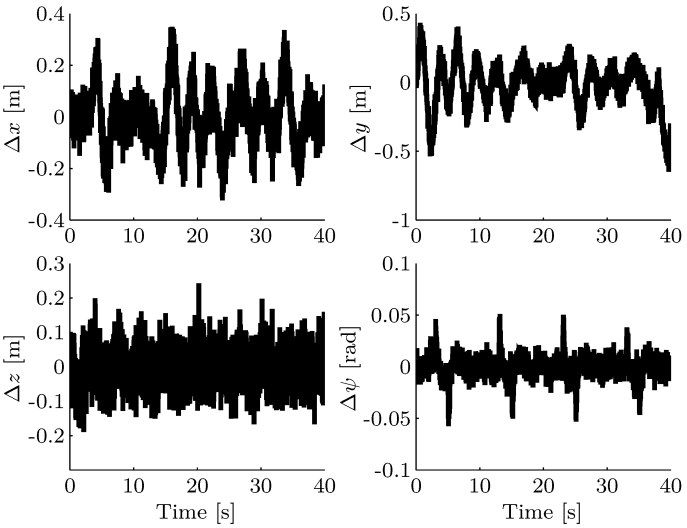


Figure 4.16: Time evolution of the tracking error when performing a square-like trajectory.

Furthermore, Figure 4.17 shows that the remaining system's degrees of freedom are kept stable during the trajectory execution, meaning that the cascade structure composed by the RMPC and the nonlinear IOFL controller was able to ensure closed-loop stability

for the Tilt-rotor UAV during the considered simulation scenario. Figure 4.18 shows that the input signals computed by the inner-loop controller did not saturate the vehicle's actuators. Moreover, it is possible to notice that some peaks appear in the torque input signals, which is explained by the required direction change in the corners of the square-like trajectory.

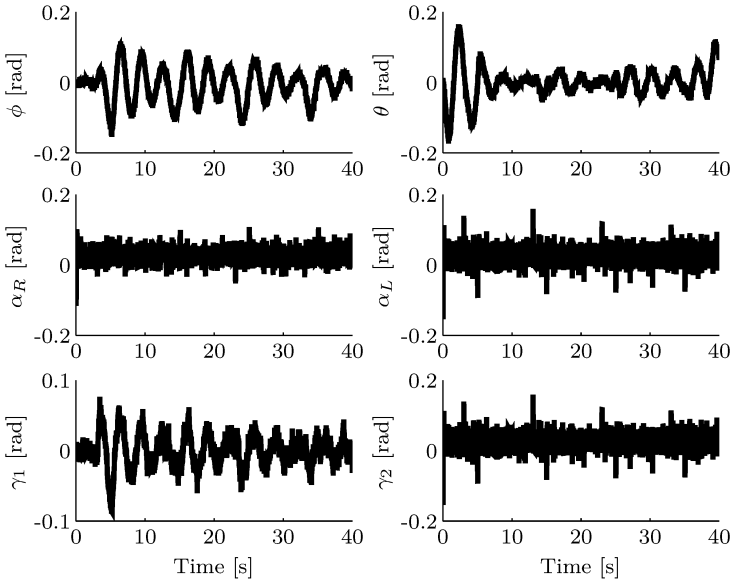


Figure 4.17: Time evolution of the remaining degrees of freedom when considering the square-like trajectory.

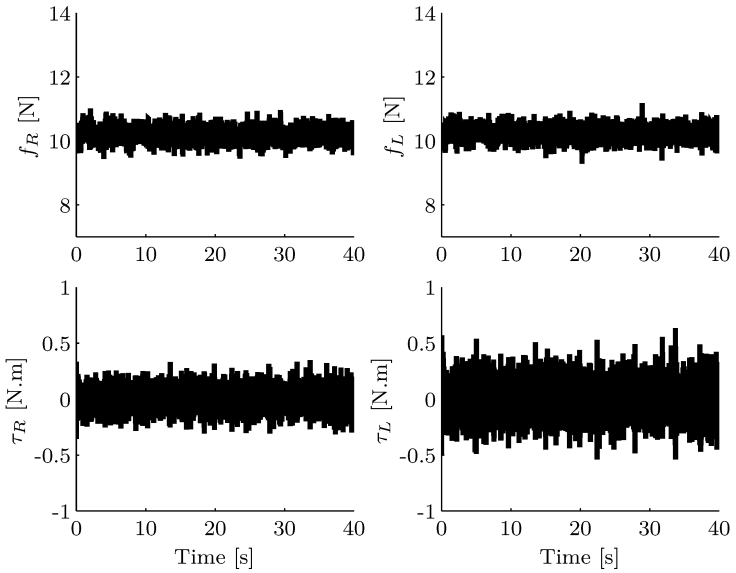


Figure 4.18: Inputs applied to the Tilt-rotor UAV.

4.4.6 Robust tube-based economic MPC

Similar to the RMPC strategy, to solve the load transportation problem using the REMPC as an outer-loop controller in the cascade structure proposed in Appendix B, the augmented

linear discrete-time model (4.105) is considered as the nominal model, i.e.,

$$\Delta\hat{\boldsymbol{x}}_{k+1}^{nom} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k^{nom} + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.117}$$

with $\boldsymbol{g}_k$ being the nominal model control input. Therefore, the uncertain linear system is described considering an additive uncertainty as described in the previous subsection

$$\Delta\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\hat{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\Delta\boldsymbol{\Upsilon}_k + \boldsymbol{w}. \tag{4.118}$$

To perform trajectory tracking considering the collision-free trajectory, the state error vector must be rewritten as

$$\Delta\tilde{\boldsymbol{x}} = \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^{goal} = \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}^{goal} \\ \int (x - x^{goal}) \\ \int (y - y^{goal}) \end{bmatrix}, \tag{4.119}$$

where $\boldsymbol{x}^{goal} = [(\boldsymbol{q}^{goal})^T \ \dot{\boldsymbol{q}}^T]^T$ and $\boldsymbol{q}^{goal} = [x^{goal} \ y^{goal} \ \gamma_1^{eq} \ \gamma_2^{eq}]^T$, with the equilibrium values being given by (4.96).

Thus, the models (4.117) and (4.118) are modified, respectively, leading to

$$\Delta\tilde{\boldsymbol{x}}_{k+1}^{nom} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\tilde{\boldsymbol{x}}_k^{nom} + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\boldsymbol{g}_k, \tag{4.120}$$

$$\Delta\tilde{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)\Delta\tilde{\boldsymbol{x}}_k + \hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)\Delta\boldsymbol{\Upsilon}_k + \boldsymbol{w}, \tag{4.121}$$

which are the error models considered in the REMPC strategy implementation.

Finally, redefining the mismatch error as $\tilde{\boldsymbol{e}}_k = \Delta\tilde{\boldsymbol{x}}_k - \Delta\tilde{\boldsymbol{x}}_k^{nom}$, the control input for the uncertain system is given by

$$\Delta\boldsymbol{\Upsilon}_k = \boldsymbol{K}(\boldsymbol{\zeta}_k)\,\tilde{\boldsymbol{e}}_k + \boldsymbol{g}_k. \tag{4.122}$$

Assuming the obstacles position $\boldsymbol{\xi}_l^{obs} \in \mathbb{O}$ known for all l, the Algorithm 4.3 describes in general terms the REMPC strategy used in the outer-loop to control the Tilt-rotor UAV's outer-loop. After executed the algorithm, the intermediary control signal is obtained and equation (B.30) needs to be considered to obtain the inner-loop references ϕ_k and θ_k from $\boldsymbol{\Upsilon}_k$.

Tunning parameters

Similar to the RMPC strategy, the following bounding sets are considered

$$\begin{aligned}
\mathbb{E} &= \{(x - x^{goal}), (y - y^{goal}) \in \pm 1 \text{ m}, \Delta\gamma_1, \Delta\gamma_2 \in \pm 0.5 \text{ rad}\}, \\
\mathbb{V} &= \{\Upsilon_1 \in \pm 13, \Upsilon_2 \in \pm 15\}, \\
\mathbb{W} &= \{w_x, w_y \in \pm 0.1 \text{ m}, w_{\gamma_1}, w_{\gamma_1} \in \pm 0.01 \text{ rad}, \},
\end{aligned}$$

Algorithm 4.3 Robust Tube-Based Economic MPC algorithm

Compute $\boldsymbol{P}$ through (4.65) using the matrices $\hat{\boldsymbol{A}}(\boldsymbol{\zeta}_k)$, $\hat{\boldsymbol{B}}(\boldsymbol{\zeta}_k)$, $\mathcal{Q}$, and $\mathcal{R}$.
Calculate $\boldsymbol{\Omega}$ by iterating the one-step operator defined at (4.50).
Set $\boldsymbol{x}_1^{goal} = \boldsymbol{x}_1^{tr}$.
Set $\Delta\tilde{\boldsymbol{x}}_0^{nom} = \Delta\tilde{\boldsymbol{x}}_0 = \boldsymbol{0}$.
procedure REMPC($\boldsymbol{x}_k$, $\overrightarrow{\boldsymbol{q}}^{tr}$, $\overrightarrow{\dot{\boldsymbol{q}}}^{tr}$, $\overrightarrow{\ddot{\boldsymbol{q}}}^{tr}$, $\boldsymbol{\xi}_l^{obs}$, $\boldsymbol{x}_k^{goal}$)
    Compute the feedback gain $\boldsymbol{K}(\boldsymbol{\zeta}_k)$ using (4.66).
    Write the augmented vectors $\Delta\tilde{\boldsymbol{x}}_k$ and $\Delta\tilde{\boldsymbol{x}}_k^{nom}$.
    Write the mismatch error $\tilde{\boldsymbol{e}}_k$.
    Compute the reachable sets $\mathscr{R}$ from k up to $k+N_p$ through (4.70).
    Check if $\min\left\{d_l\left(\boldsymbol{\xi}_{k+cN_p}^{tr}, \boldsymbol{\xi}_l^{obs}\right)\right\} \leq d_l^* \ \forall l$.
    Write the cost function (4.82).
    Obtain the constraints (4.71), (4.72), (4.73), (4.75), (4.77), (4.79), (4.80), (4.81).
    Solve the optimization problem (4.84) to obtain $\overrightarrow{\boldsymbol{g}}$ and $\overrightarrow{\boldsymbol{x}}^{goal}$
    Set $\boldsymbol{x}_k^{goal} = \boldsymbol{x}_{k+1}^{goal}$.
    Calculate $\Delta\boldsymbol{\Upsilon}_k = \boldsymbol{K}(\boldsymbol{\zeta}_k)\,\tilde{\boldsymbol{e}}_k + \boldsymbol{g}_k$.
    Compute $\boldsymbol{\Upsilon}_k^{tr}$ using (4.100).
    return $\boldsymbol{\Upsilon}_k$ and $\boldsymbol{x}_k^{goal}$.
end procedure

being the weighting matrices given by

$$\boldsymbol{\mathcal{Q}} = \operatorname{diag}\left(\frac{1}{2^2}, \frac{1}{2^2}, \frac{5}{(\pi/2)^2}, \frac{5}{(\pi/2)^2}, \frac{1}{2^2}, \frac{1}{2^2}, \frac{1}{(3\pi)^2}, \frac{1}{(3\pi)^2}, 10, 10\right), \tag{4.123}$$

$$\boldsymbol{\mathcal{R}} = \operatorname{diag}\left(\frac{1}{(13-\Upsilon_1^{eq})^2}, \frac{1}{(15-\Upsilon_2^{eq})^2}\right). \tag{4.124}$$

The prediction and control horizons are $N_p = 10$ and $N_c = 5$. Further, the maximum absolute accelerations are 1 m/s², given the polytope $\Delta = \{\pm 1, \pm 1\} \in \mathbb{R}^2$. The potential functions weighting matrices are $\boldsymbol{\kappa} = 0.1 \cdot \mathbb{I}_{2\times 2}$ and $\boldsymbol{\lambda} = 15 \cdot \mathbb{I}_{2\times 2}$. The security distance is $d_l^* = 0.5$, and the scaling factor is $c = 10$.

Simulation results

Figure 4.19 shows the Tilt-rotor UAV performing trajectory tracking while carrying a suspended load and when the desired path is obstructed by obstacles to be overcame by the vehicle. Similar to the EMPC, the cascade structure composed by the REMPC and the nonlinear IOFL controller was able to perform obstacle avoidance successfully (see Figure 4.19). The economic stage cost of the REMPC is the reason why the vehicle looks to an alternative collision-free trajectory when approaching obstacles and returns to the desired one after passing through it.

Figures 4.20 and 4.21 show, respectively, the trajectory projection presented in Figure 4.19, which is decomposed in the $y-x$ and $z-x$ planes. Observe that, unlike the simulation using the EMPC strategy (see Figure 4.19), the UAV deviates from all the obstacles using

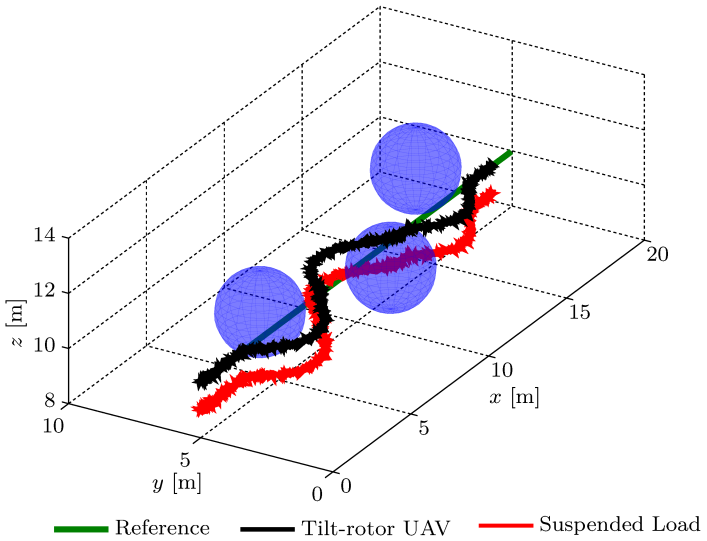


Figure 4.19: Trajectory tracking using the REMPC in a cascade structure when performing the trajectory presented in Figure 4.4.

only its planar motion, i.e., without changing its altitude. This behavior is expected due to the necessity of use the REMPC in a cascade control scheme to avoid dimensionality problems and to work with smaller sampling frequencies. Moreover, the REMPC only controls the states x, y, γ_1, and γ_2. Therefore, it does not have any action on z. On the other hand, the use of the cascade structure allows to have a control system able to perform planar motion with obstacle avoidance, besides to consider yaw's movements regulation, which could not be achieved if the REMPC strategy, as presented in section 4.3, was used considering the whole-body control approach.

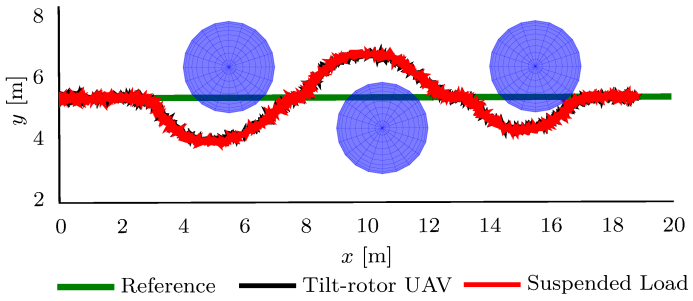


Figure 4.20: Projection in the $y - x$ plane of the trajectory shown in Figure 4.19.

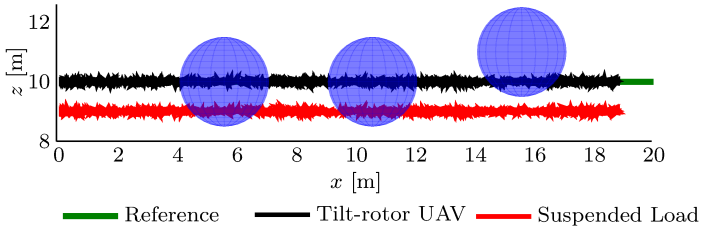


Figure 4.21: Projection in the $z - x$ plane of the trajectory shown in Figure 4.19.

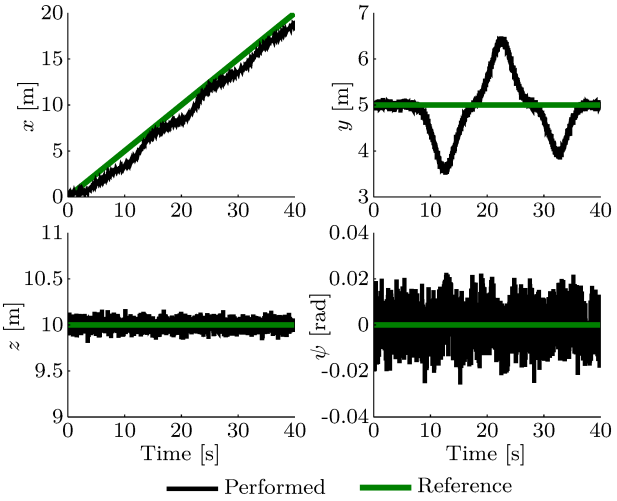


Figure 4.22: Time evolution of the regulated variables (x, y, z, ψ).

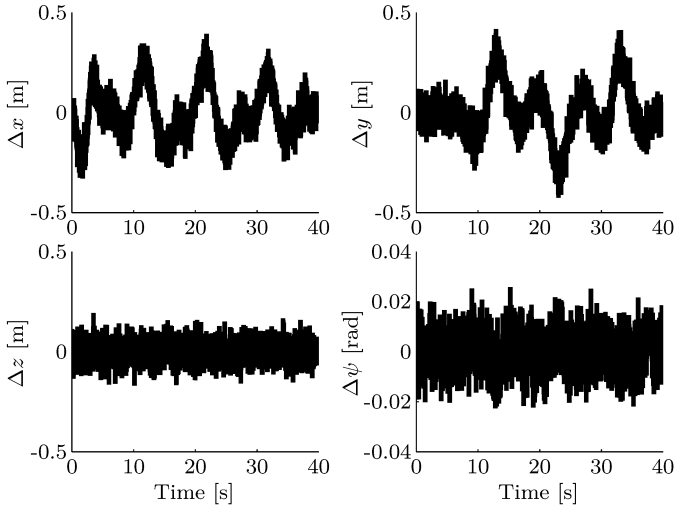


Figure 4.23: Time evolution of the tracking errors of the regulated variables (x, y, z, ψ).

Further, Figures 4.22 and 4.23 show, respectively, the time evolution of the regulated variables x, y, z, and ψ, and its tracking errors. Observe that the difference between the desired trajectory and the performed one in the y time evolution is due to the obstacle avoidance feature. Moreover, note that the tracking error Δx and Δy are obtained with respect to the collision-free trajectory, while the Δz and $\Delta\psi$ consider the error with respect to the initial proposed one since these states are addressed by the inner-loop controller.

Finally, Figures 4.24 and 4.25 show, respectively, the time evolution of the remaining degrees of freedom and the control inputs computed by the inner-loop controller. All the remaining degrees of freedom are kept stable by the cascade structure using the REMPC, which shows the closed-loop stability feature for the considered simulation. Further, thrusts and torques provided by the nonlinear controller, respectively, to the rotors and servomotors do not exceed their saturation levels.

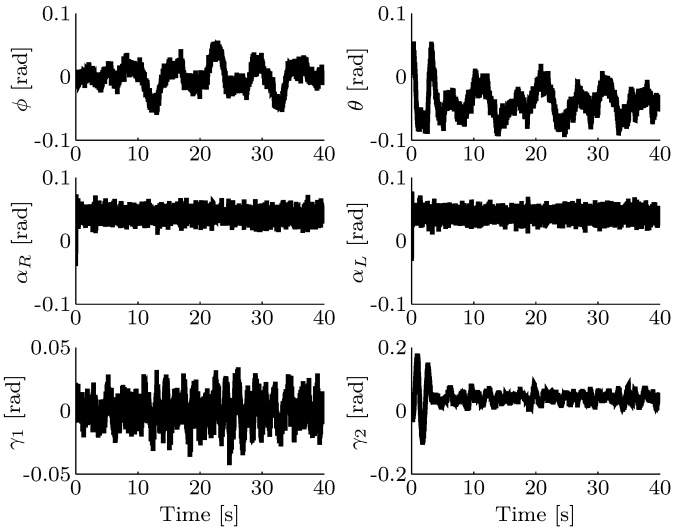


Figure 4.24: Time evolution of the remaining degrees of freedom when using the REMPC in a cascade structure.

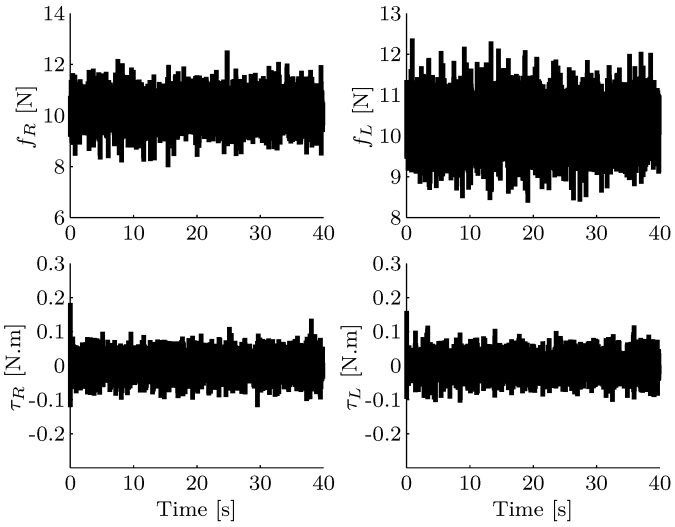


Figure 4.25: Applied thrusts and torques to the Tilt-rotor UAV when using the REMPC in a cascade structure.

4.5 Final remarks

In this chapter three types of predictive controllers with advanced features were proposed aiming to solve the load transportation problem using a Tilt-rotor UAV from the UAV's perspective. The main goal was to design a control system able to perform trajectory tracking and obstacle avoidance while providing closed-loop stability assurance, constant external disturbances rejection, parametric uncertainties attenuation, and constraints satisfaction on state deviations and control inputs. First, an economic model predictive controller was derived with its economically motivated stage cost defined by potential functions in order to perform obstacle avoidance through an one-layer scheme. The LTI-MPC proposed in Chapter 3 was took as basis for its design, and the EMPC were use to solve the load transportation problem while performing obstacle avoidance through

a whole-body approach. Secondly, a robust tube-based model predictive controller was presented aiming to address robustness issues without only rely on the inherited robustness presented in feedback controllers. Therefore, the RMPC, using set theory tools, considers uncertainties by design concept, which increases its domain of attraction when compared to the standard MPC formulations. The RMPC was used to solve the load transportation problem through a cascade structure in order to avoid dimensionality problems that could made the implementation of this controller impractical. Finally, the main goal of this chapter was achieved when the EMPC and RMPC was gathered into a single strategy to create a robust tube-based economic model predictive controller able to formally address the robustness problem and to perform obstacle avoidance. This strategy was applied to solve the load transportation problem using a Tilt-rotor UAV through a cascade structure, avoiding the dimensionality problems presented in RMPC.

Numerical results have shown that the EMPC successfully performed trajectory tracking avoiding obstacles obstructing the aircraft's workspace. Therefore, in the considered simulation scenario, this controller was able to achieve the specified requirements, i.e., perform trajectory tracking while ensuring closed-loop stability and obstacle avoidance due to the combination of the standard quadratic with stability regions stage cost and the economic stage cost. Moreover, as for the LTI-MPC controller, the quadratic stage cost and the inclusion of integral actions also make the EMPC strategy capable of dealing with constant external disturbances and parametric uncertainties rejection. The RMPC strategy was also successfully used to solve the load transportation problem using a Tilt-rotor UAV. The presented numerical results corroborate the good trajectory tracking performance of these controllers even in presence of disturbances and the parametric uncertainties represented by the model dependency on the reference accelerations. Moreover, due to the use of a cascade structure with a nonlinear controller in the inner-loop, it was possible to perform yaw's movements regulation in a way that the vehicle could always be head-on to the trajectory. Yet, the numerical results had also shown that the combination of obstacle avoidance and robustness to uncertainties were possible using the REMPC strategy. However, due to the necessity of working in a cascade structure, only planar obstacle avoidance was possible.

Despite being able to bring advanced features into the load transportation using a Tilt-rotor UAV problem, the controllers proposed in this chapter have the drawback of being computationally costly. The cost of the EMPC algorithm is mainly due to the nonlinear cost function, the optimization problem, and the number of constraints necessary in its formulation. The RMPC algorithm cost, on the other side, is due to the computational burden associated with the Minkowski sum and the Pontryagin difference, which are usualy obtained through the interception of hyperplanes. Finally, since the REMPC gathers the benefit and drawback of both controllers, its algorithm is the one with higher computation. Since this work does not address the implementation issues related to these controllers

algorithms, the results presented here can only be obtained through numerical simulations and the control strategies cannot yet be tested in the real system. In the next chapter some general conclusions about this thesis are given and some future works, including the implementation issues and the obstacle detection problem, are proposed.

5 Conclusion

5.1 Overview

This thesis dealt with the load transportation control problem using a Tilt-rotor UAV from two different approaches. In order to solve this problem five different formulations of model predictive controllers were proposed. Initially, the suspended load trajectory tracking with the UAV stabilization was addressed using both LTI-MPC and LTV-MPC control strategies. After, the UAV trajectory tracking problem with load stabilization was considered and solved through a whole-body approach using the EMPC and through a cascade structure using the RMPC and REMPC control strategies. It was required for all designed controllers to be able to perform trajectory tracking while ensuring closed-loop stability throughout the trajectory, to reject constant external disturbances and parametric uncertainties, and to satisfy constraints on state deviations and control inputs. Numerical results have shown that these goals were achieved by all proposed model predictive controllers.

The first proposed control strategy was a linear time-invariant model predictive controller for trajectory tracking of the suspended load considering a standard quadratic stage cost together with stability regions. For prediction purposes, the nonlinear state-space equations of motion were linearized around an equilibrium pointand adapted to consider the incremental form of MPC aiming to achieve improved performance by means of the additional integrators. The main goal of using the standard MPC formulation was to provide a computationally efficient control algorithm suitable to be implemented in the ProVANT 2.0 UAV's embedded system observing the required 12 ms sampling time. This

is an improvement of the previous results obtained in Santos & Raffo (2016b) and Andrade et al. (2016). In fact, the inclusion of a terminal cost and terminal constraint set, despite formally ensure closed-loop stability in the vicinity of the linearization point, have also allowed the reduction of the prediction and control horizons without losing performance. These features allowed to reduce the control algorithm time of execution from 55 ms to 3.2 ms, in the average case, and from 129.8 ms to 11.6 ms, in the worst case. Therefore, the proposed strategy, even in the worst case, can be successfully used in the real system to solve the considered load transportation control problem.

The second designed predictive control strategy also solved the trajectory tracking of the suspended load using a Tilt-rotor UAV. However, it differs from the previous control strategy by considering a linear time-variant model as the foundation of its prediction process. For the LTV-MPC, the nonlinear state-space equations of motion were linearized around a generic trajectory considering the rope's length as a time-varying parameter and also extended to consider the incremental form of MPC. In this formulation, the stability regions were not considered due to the inability to express the LTV model in a polytopic form and the prohibitive computation cost to compute this regions in an on-line approach. However, because of the time-varying model and model-based nature of predictive controllers, the LTV-MPC formulation was able to solve problems requiring a larger domain of attraction, such as: yaw movement regulation, and rope's length variation during take-off and landing maneuvers. The linearized controllers that have been proposed to solve this problem, with exception to those considering scheduling between control strategies, are unable to cope with yaw regulation. Moreover, the rope's length variation is only regarded as an uncertainty to be rejected by the controllers due to its robustness against model uncertainties. In this context, the proposed controller has the feature of addressing this practical problem in a simple way without relying only on the controllers' robustness. However, this advantage comes with the drawback of increasing the control algorithm computational cost due to the high number of required on-line computation. Therefore, without further research on the topic of how to reduce the computational cost of this controller, it cannot be implemented in the aircraft's embedded system.

For the third proposed control strategy, the load transportation control problem was solved considering the Tilt-rotor UAV's trajectory tracking and obstacle avoidance. For the development of the EMPC, the nonlinear state-space equations of motion were linearized around an equilibrium point and all obstacles were considered to be perfectly known. The optimization problem was designed through a cost function formulated by a standard MPC quadratic stage cost together with an economic oriented one. The quadratic cost, as for the LTI-MPC strategy, was used to ensure stability and performance, while the economic cost was considered to allow obstacle avoidance. Attraction and repulsive potential functions were use to formulate the economic criteria, and constraints were added into the optimization problem in order to define no-fly zones areas and giving a

stronger assurance against collisions. The presented numerical results, obtained through simulations, showed that the EMPC strategy was able to perform obstacle avoidance while performing trajectory tracking stably. Nevertheless, despite being formulated with terminal cost and terminal constraint and, hence, being able to work with small prediction and control horizons, due to the EMPC nonlinear optimization problem, this control strategy has shown to be computationally costly, which prevents it to be implemented in the real aircraft without further investigation.

The fourth proposed control strategy was a robust tube-based model predictive controller used to solve the load transportation problem from the UAV's perspective through a cascade structure. The choice of using a hierarchical approach, in which the RMPC control strategy only deals with the planar motion trajectory tracking and the load stabilization, was made seeking to reduce the control problem's dimension and to allow the controller to run in a slower frequency rate. In order to design the proposed controller, the outer-loop model was linearized around a generic trajectory, which resulted in a linear time-variant model with politopic representation. The proposed control strategy is said to be robust since it considers in its formulation the effect of both additive and modeling uncertainties. This approach used set theory tools to define tube regions in order to constrain the uncertain system's evolution. Its optimal control policy was used to steer the nominal system throughout the desired trajectory, while an adaptive law reduced the mismatch error between the uncertain and the nominal systems. Numerical simulation results were shown to corroborate the controller's good trajectory tracking performance in the presence of external disturbances and modeling uncertainties, which was mainly associated with the decoupling and linearization process. Despite the use of the RMPC control strategy to only address the outer-loop control problem, this strategy still had high computational cost due to the on-line calculation of Minkowski sums and Pontryagin differences. Therefore, further researches on the topic of computational efficiency of this control algorithm still need to be made in order to make its implementation practicable.

Finally, the fifth control strategy was proposed aiming to gather the RMPC's robustness with the EMPC's obstacle avoidance. This controller also solved the load transportation problem from the UAV's perspective. Similar to the RMPC, the robust tube-based economic model predictive controller was used in a cascade structure seeking to reduce the control problem complexity. For the REMPC, the outer-loop model was also linearized around a generic trajectory, which resulted in a LTV model. Moreover, all obstacles were also considered to be perfectly known. Since this controller only dealt with the planar motion trajectory tracking and the load stabilization, the obstacle avoidance feature was only achieved in the $x - y$ plane. Numerical simulation results were proposed to show the controller's ability of joining the two previously proposed control strategies' features. Although being able to gather their features, the REMPC also gathered their drawbacks. Therefore, this control strategy has high computational cost and will be only possible

to be applied in the real system when the previous controller's computational efficiency problems are solved.

5.2 Contributions

The main contribution of this thesis are:

- The design of an LTI-MPC with suitable computational cost for being implemented in the Tilt-rotor UAV's embedded system;
- The design of an LTV-MPC able to address the yaw's movements regulation problem and the rope's length variation during take-off and landing maneuvers;
- The design of an EMPC considering potential functions as economic criteria to solve the obstacle avoidance problem though an one-layer scheme;
- The design of a RMPC to control a Tilt-rotor UAV used for load transportation tasks considering, by design concept, both additive and model uncertainties;
- The design of a REMPC gathering the robustness and obstacle avoidance features to solve the load transportation control problem using a Tilt-rotor UAV.

5.3 Future works

This section suggests possible future works that could improve the results of this thesis.

- **Improve the LTV-MPC computational cost.** The computational cost of the LTV-MPC is mainly due to the inability of finding a politopic representation for the linearized model through first-order Taylor series expansion and the large prediction and control horizons considered. The first problem can be worked around by obtaining the linear parameter varying model using a subspace identification process, while the second one can be addressed investigating scheduling MPC techniques that, formally, ensure stability allowing the reduction of the controller's horizons.
- **Inclusion of input rate constraints.** For all MPC strategies formulated in this work it will be necessary to include input rate constraints in order to successfully validate them in an experimental setup. Although the proposed strategy did not took into account the actuators dynamics, the inclusion of input rate constraints will consider, in a simpler way, the actuators time response avoiding the controllers to calculate unrealistic control actions.

- **Obstacle avoidance considering the aircraft's orientation and velocities.** The EMPC proposed in this work considers only the aircrafts position in order to find a collision-free trajectory. The inclusion of the orientation could improve the performance, for instance, by keeping the vehicle head-on to the trajectory. Moreover, due to the Tilt-rotor UAV's dynamics be described by second-order differential equations, the controller needs to control the generalized coordinates and their time derivatives. Therefore, depending on the desired trajectory velocities amplitude, when a collision-free trajectory is found but the desired velocities are kept the same, the control system may not be able to perform the collision-free trajectory properly and could even become unstable.

- **Inclusion of load's collision avoidance.** Since load's collisions are undesirable due to many aspects, such as: capacity of destabilize the whole-system, damage of the cargo and the environment, among others; the EMPC formulation can be modified in order to also includes load's collision avoidance.

- **Study of obstacle detection strategies.** For future implementation of the EMPC control strategy in the Tilt-rotor UAV, the obstacle detection problem must be addressed.

- **Study of computational efficiency of the nonlinear optimization algorithms.** In order to be able to implement the EMPC in the UAV's embedded system, observing the desired 12 ms time period for the control algorithm execution, the study of computational efficient algorithms to solve the controller's optimization problem is required.

- **Compute the adaptive controller through $\mathcal{H}_\infty$ approach.** By changing the LMI formulated in order to compute the mismatch error adaptive controller's gain for the RMPC and REMPC strategies, it is possible to include some additional features into the problem. For instance, since the optimization problem is formulated only to find a Lyapunov positive definite matrix $\mathbf{P}$, it could be rewrite as an $\mathcal{H}_\infty$ problem aiming to have additional robustness features ensured by the Small Gain Theorem (Boyd et al., 1991).

- **Improve the RMPC's computational cost.** The investigation of efficient ways to perform Minkowski sums and Pontriagin differences are necessary to reduce the algorithm's time of execution. For instance, the use of more conservative sets to compute the reachable sets and to define the restrictions, e.g., zonotopes, can be considered.

- **Ensure no saturation for the control inputs in the RMPC strategy.** Although the MPC strategy used to control the nominal system are able to ensure that the

control inputs will not reach some desired saturation levels, due to the combination of the MPC law with mismatch error adaptive control law, the applied control input has none assurance of no saturation.

- **Validation of the proposed techniques in the Gazebo platform.** A simulation platform based on Gazebo was developed in the ProVANT project to validate control and estimation strategies (Lara et al., 2017). Therefore, the validation of the proposed control strategies in this platform would allow to test their ability to solve the load transportation problem in a more realistic scenario.

- **Validation of the proposed techniques in experimental setup.** An important step in this research would be the application of the proposed control strategies in the ProVANT 2.0 UAV to corroborate their performance in the real system.

- **EMPC stability analysis.** Analyze the stability of the EMPC algorithm presented in Chapter 4.

Bibliography

Alexis, K., Nikolakopoulos, G., & Tzes, A. (2014). On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances. *Asian Journal of Control*, 16(1), 209–224.

Alexis, K., Papachristos, C., Siegwart, R., & Tzes, A. (2015). Sampling-based receding horizon collision-free control for a class of micro aerial vehicles. In *Proc. of the 23rd Mediterranean Conference on Control and Automation* (pp. 675–680).

Almeida, M. M. (2014). Control strategies of a tilt-rotor uav for load transportation. Master's thesis, Federal University of Minas Gerais, Graduate Program in Electrical Engineering, Belo Horizonte, Brazil.

Almeida, M. M., Donadel, R., Raffo, G. V., & Becker, L. B. (2014). Full Control of a Tiltrotor UAV for Load Transportation. In *Proc. of the XX Congresso Brasileiro de Automática* (pp. 2097–2104).

Almeida, M. M. & Raffo, G. V. (2015). Nonlinear control of a tiltrotor UAV for load transportation. In *Proc. of the 11th IFAC Symposium on Robot Control* (pp. 234–239).

Amiri, N., Ramirez-Serrano, A., & Davies, R. (2011). Modelling of opposed lateral and longitudinal tilting dual-fan unmanned aerial vehicle. In *Proc. of the 18th IFAC World Congress* (pp. 2054–2059).

Amiri, N., Ramirez-Serrano, A., & Davies, R. J. (2013). Integral backstepping control of an unconventional dual-fan unmanned aerial vehicle. *Journal of Intelligent & Robotic Systems*, 69(1), 147–159.

Amrit, R., Rawligns, J. B., & Biegler, L. T. (2013). Optimizing process economics online using model predictive control. *Computers and Chemical Engineering*, 58(1), 334–343.

Andrade, R. (2016). Predictive control strategies for unmanned aerial vehicles in cargo transportation tasks. Master's thesis, Federal University of Santa Catarina.

Andrade, R., Raffo, G. V., & Rico, J. E. N. (2016). Model predictive control of a tilt-rotor UAV for load transportation. In *Proc. of the European Control Conference* (pp. 2165–2170).

Beard, B. R. W., McLain, T. W., Nelson, D. B., Kingston, D., & Johanson, D. (2006). Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7), 1306–1324.

Bemporad, A. & Morari, M. (1999). *Robust model predictive control: A survey*, (pp. 207 – 226). Springer London.

Bernard, M. & Kondak, K. (2009). Generic slung load transportation system using small size helicopters. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 3258–3264).

Bernard, M., Kondak, K., Maza, I., & Ollero, A. (2011). Autonomous transportation and deployment with aerial robots for search and rescue missions. *Journal of Field Robotics*, 28(6), 914–931.

Bisgaard, M. (2008). *Modeling, Estimation and Control of Helicopter Slung Load System*. PhD thesis, Aalborg University.

Bisgaard, M., la Cour-Harbo, A., & Bendtsen, J. D. (2009). Swing damping for helicopter slung load systems using delayed feedback. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*.

Blanchini, F. & Miani, S. (2007). *Set-Theoretic Methods in Control*. Birkhauser Basel, 1st edition.

Boyd, S., Ghaoui, L. E., Feron, E., & Balakrishnan, V. (1991). *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics.

Camacho, E. F. & Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag London.

Cardoso, D. N., Raffo, G. V., & Esteban, S. (2016). A robust adaptive mixing control for improved forward flight of a tilt-rotor UAV. In *Proc. of 19th International Conference on Intelligent Transportation Systems* (pp. 1432–1437).

Castillo, C. L., Moreno, W., & Valavanis, K. P. (2007). Unmanned helicopter waypoint trajectory tracking using model predictive control. In *Proc. of the 2007 Mediterranean Conference on Control and Automation* (pp. 1–8).

Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., & Thrun, S. (2015). *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press.

Chung, R. H. H. (2017). Planejamento de trajetória com robustez e desvio de obstáculos para um helicóptero com três graus de liberdade. Master's thesis, Instituto Tecnológico de Aeronáutica, Pós-Graduação em Engenharia Eletrônica e Computação, São José dos Campos, Brasil.

Dai, S., Lee, T., & Bernstein, D. S. (2014). Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass. In *Proc. of the 53rd IEEE Conference on Decision and Control* (pp. 6155–6160).

D'Jorge, A., Ferramosca, A., & González, A. (2017). A robust gradient-based mpc for integrating real time optimizer (rto) with control. volume 54 (pp. 65–80).: Elsevier.

Donadel, R., de Almeida Neto, M. M., Raffo, G. V., & Becker, L. B. (2014a). Path tracking control of a small scale tiltrotor unmanned aerial vehicle. In *Proc. of the XX Congresso Brasileiro de Automática* (pp. 1450–1457).

Donadel, R., Raffo, G. V., & Becker, L. B. (2014b). Modeling and Control of a Tiltrotor UAV for Path Tracking. In *Proc. of the 19th IFAC World Congress* (pp. 3839–3844).

Ellis, M., Durand, H., & Christofides, P. D. (2014). A tutorial review of economic model predictive control methods. *Journal of process control*, 24(1), 1156–1178.

Ellis, M., Liu, J., & Christofides, P. D. (2017). *Economic Model Predictive Control: Theory, Formulations and Chemical Process Applications*. Springer.

Faust, A., Palunko, I., Cruz, P., Fierro, R., & Tapia, L. (2013). Learning swing-free trajectories for UAVs with a suspended load. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 4887–4894).

Gonzalez, R., Fiacchini, M., Alamo, T., Guzman, J., & Rodriguez, F. (2010). Adaptive control for a mobile robot under slip conditions using an lmi-based approach. *European Journal of Control*, 16(2), 144 – 155.

Goodarzi, F. G. (2016). Autonomous aerial payload delivery with quadrotor using varying length cable. In *Proc. of the 2016 International Conference on Advanced Mechatronic Systems* (pp. 394–399).

Halliday, D., Resnick, R., & Walker, J. (2013). *Fundamentals of Physics*. Wiley, extender, 10th edition.

Hinojosa, A. I., Ferramosca, A., Gonzalez, A. H., & Odloak, D. (2017). One-layer gradient-based mpc + rto of a propylene/propane splitter. *Computers and Chemical Engineering*, 106(1), 160–170.

Johnson, M. A. & Grimble, M. J. (1987). Recent trends in linear optimal quadratic multivariable control system design. *IEE Proceedings on Control Theory and Applications*, 134(1), 53–71.

Kane, T. R. & Levinson, D. A. (1985). *Dynamics: Theory and Applications*. Mcgraw-Hill College.

Kang, Y. & Hedrick, J. K. (2006). Design of nonlinear model predictive controller for a small fixed-wing unmanned aerial vehicle. In *Proc. of the AIAA Guidance, Navigation, and Control Conference* (pp. 1–12).

Kayacan, E., Kayacan, E., Ramon, H., & Saeys, W. (2015). Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trialer system. *IEEE/ASME Transaction on Mechatronics*, 20(1), 447–456.

Ke, F., Li, Z., & Yang, C. (2018). Robust tube-based predictive control for visual servoing of constrained differential-drive modile robots. *IEEE Transactions on Industrial Electronics*, 65(4), 3437–3446.

Kendoul, F., Fantoni, I., & Lozano, R. (2006). Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Transactions on Robotics*, 22(6), 1297–1302.

Kerrigan, E. C. (2000). *Robust Constraint Satisfaction: Invariant Sets an Predictive Control*. PhD thesis, University of Cambridge, Department of Engineering.

Kerrigan, E. C. & Maciejowski, J. M. (2000). Invarian sets for constrained nonlinear discrete-time systems with application to feasibility in model predictive control. In *Proc. of the 39th IEEE Conference on Decision and Control* (pp. 4951–4956).

Khalil, H. K. (2001). *Nonlinear Systems*. Pearson, 3rd edition.

Kirk, D. E. (2004). *Optimal Control Theory: an introduction*. Dover Publications, Inc.

Kunz, K., Huck, S. M., & Summers, T. H. (2013). Fast model predictive control of miniature helicopters. In *Proc. of the 2013 European Control Conference* (pp. 1377–1382).

la Cour-Harbo, A. & Bisgaard, M. (2009). State-control trajectory generation for helicopter slung load system using optimal control. *Proc. of the AIAA Guidance, Navigation, and Control Conference*.

Langson, W., Chryssochoos, I., Rakovic, S. V., & Mayne, D. Q. (2004a). Robust model predictive control using tubes. *Automatica*, 40(1), 125–133.

Langson, W., Chryssochoos, I., Rakovik, S. V., & Mayne, D. Q. (2004b). Robust model predictive control using tubes. *Automatica*, 40(1), 125–133.

Lara, A. V., Rego, B. S., Raffo, G. V., & Arias-Garcia, J. (2017). Desenvolvimento de um ambiente de simulação de vants tilt-rotor para testes de estratégias de controle. In *Proc. of the XII Simpósio Brasileiro de Automação Inteligente* (pp. 2135–2141).

Lee, J. H. & Yu, J. (1997). Worst-case formulation of model predictive control for systems with bounded parameters. *Automatica*, 33(5), 763–781.

Lee, T. (2018). Geometric control of quadrotor uavs transporting a cable-suspended rigid body. *Transactions on Control Systems Technology*, 26(1), 255–264.

Lee, T., Sreenath, K., & Kumar, V. (2013). Geometric control of cooperating multiple quadrotor UAVs with a suspended payload. In *Proc. of the 52nd IEEE Conference on Decision and Control* (pp. 5510–5515).

Mac, T. T., Copot, C., Hernandez, A., & Keyser, R. (2016). Improved potential field method for unknown obstacle avoidance using uav in indoor enviroment. In *Proc. of the IEEE 14th International Symposium on Applied Machine Intelligence and Informatics* (pp. 345–350).

Mayne, D. Q., Rakovic, S. V., Findeisen, R., & Allgower, F. (2009). Robust output feedback model predictive control of constrained linear systems: Time varying case. *Automatica*, 45(9), 2082–2087.

Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Merino, L., Caballero, F., Dios, J. R., & Ollero, A. (2005). Cooperative fire detection using unmanned aerial vehicles. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 1884–1889).

Nascimento, T. P., Conceicao, A. G. S., & Moreira, A. P. (2014). Multi-robot systems formation control with obstacle avoidance. In *Proc. of the 19th IFAC World Congress* (pp. 5703–5708).

Nguyen, H.-N. (2014). *Constrained Control of Uncertain, Time-Varying, Discrete-time Systems: An interpolation-Based Approach*. Springer, 1st edition.

Palunko, I., Cruz, P., & Fierro, R. (2012). Agile load transportation: safe and efficient load manipulation with aerial robots. *IEEE Robotics & Automation Magazine*, 19(3), 69–79.

Palunko, I., Faust, A., Cruz, P., Tapia, L., & Fierro, R. (2013). A reinforcement learning approach towards autonomous suspended load manipulation using aerial robots. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 4881–4886).

Papachristos, C., Alexis, K., & Tzes, A. (2013). Model predictive hovering-translation control of an unmanned tri-tiltrotor. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 5425–5432).

Park, S., Bae, J., Kim, Y., & Kim, S. (2013). Fault tolerant flight control system for the tilt-rotor uav. *Journal of the Franklin Institute*, 350(9), 2535–2559.

Pereira, P. O. & Dimarogonas, D. V. (2016). Lyapunov-based generic controller design for thrust-propeller underactuated systems. In *Proc. of the European Control Conference* (pp. 594–599).

Pereira, P. O., Herzog, M., & Dimarogonas, D. V. (2016). Slung load transportation with a single aerial vehicle and disturbance rejection. In *Proc. of the 24th Mediterranean Conference on Control and Automation* (pp. 671–676).

Perez, A., Platt, R., Konidaris, G., Kaelbling, L., & Lozano-Perez, T. (2012). Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 2537–2542).

Petkar, S., Umbarkar, S., Mejari, M., Singh, N. M., & Kazi, F. (2016). Robust tube based mpc for pvtol trajectory tracking using systems flatness property. In *Proc. of the 2016 International Conference on Unmanned Aircraft Systems* (pp. 1095–1101).

Raffo, G. V. & Almeida, M. M. (2016). Nonlinear robust control of a quadrotor UAV for load transportation with swing improvement. In *Proc. of the American Control Conference* (pp. 3156–3162).

Raffo, G. V. & Almeida, M. M. (2017). A load transportation nonlinear control strategy using a tilt-rotor uav. *Journal of Advanced Transportation*, 2017, 1–32.

Raffo, G. V., Ortega, M. G., & Rubio, F. R. (2010). An integral predictive/nonlinear $\mathcal{H}_\infty$ control structure for a quadrotor helicopter. *Automatica*, 46(1), 29–39.

Raffo, G. V., Ortega, M. G., & Rubio, F. R. (2011). Nonlinear $\mathcal{H}_\infty$ controller for the quad-rotor helicopter with input coupling. In *Proc. of the 18th IFAC World Congress*, volume 18 (pp. 13834–13839).

Rawlings, J. B., Angeli, D., & Bates, C. N. (2012). Fundamentals of economic model predictive control. In *Proc. of the 51st IEEE Conference on Decision and Control* (pp. 3851–3861).

Rego, B. S. (2016). Path tracking control of a suspended load using a tilt-rotor uav. Master's thesis, Federal University of Minas Gerais, Graduate Program in Electrical Engineering, Belo Horizonte, Brazil.

Rego, B. S. & Raffo, G. V. (2016a). Path tracking control based on guaranteed state estimation for a tilt-rotor UAV. In *Proc. of the XXI Congresso Brasileiro de Automática* (pp. 1241–1246).

Rego, B. S. & Raffo, G. V. (2016b). Suspended load path tracking by a tilt-rotor UAV. In *Proc. of the 1st IFAC Conference on Cyber-Physical & Human-Systems* (pp. 229–234).

Rego, B. S. & Raffo, G. V. (2016c). Suspended load path tracking control based on zonotopic state estimation using a tilt-rotor UAV. In *Proc. of the IEEE 19th International Conference on Intelligent Transportation Systems* (pp. 1445–1451).

Rossiter, J. A. (2013). *Model-based predictive control: a practical approach*. CRC press.

Ryan, A. & Hedrick, J. K. (2005). A mode-switching path planner for uav-assisted search and rescue. In *Proc. of the 44th IEEE Conference on CDC-ECC'05* (pp. 1471–1476).

Sanchez, A., Escareno, J., Garcia, O., & Lozano, R. (2008). Autonomous hovering of a noncyclic tiltrotor uav: Modeling, control and implementation. volume 41 (pp. 803 – 808).

Sánchez, R. G. (2011). *Contributions to Modelling and Control of Mobile Robots in Off-Road Conditions*. PhD thesis, Universidad de Sevilla.

Santos, M. A., Cardoso, D. N., Rego, S. B., Raffo, G. V., & Esteban, S. (2017a). A discrete robust adaptive control of a tilt-rotor uav for an enlarged flight envelope. In *Proc. of the 56th IEEE Conference on Decision and Control* (pp. 5208–5214).

Santos, M. A. & Raffo, G. V. (2016a). Adaptive control of a tilt-rotor UAV in load transportation tasks - a lmi based approach. In *Proc. of the XXI Congresso Brasileiro de Automática* (pp. 2461–2466).

Santos, M. A. & Raffo, G. V. (2016b). Path tracking model predictive control of a tilt-rotor UAV carrying a suspended load. In *Proc. of the IEEE 19th International Conference on Intelligent Transportation Systems* (pp. 1458–1463).

Santos, M. A., Rego, B. S., Raffo, G. V., & Ferramosca, A. (2017b). Suspended load path tracking control strategy using a tilt-rotor UAV. *Journal of Advanced Transportation*, 2017, 1–22.

Shabana, A. A. (2005). *Dynamics of Multibody Systems*. Cambridge University Press.

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: modelling, planning and control*. Springer Science & Business Media.

Skokaert, P. & Mayne, D. (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(1), 1136–1142.

Slotine, J.-J. S. & Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.

Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley & Sons, Inc.

Sreenath, K., Lee, T., & Kumar, V. (2013a). Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load. In *Proc. of the 52nd IEEE Conference on Decision and Control* (pp. 2269–2274).

Sreenath, K., Michael, N., & Kumar, V. (2013b). Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 4873–4880).

Tang, S. & Kumar, V. (2015). Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload. In *Proc. of the IEEE International Conference on Robotics and Automation* (pp. 2216–2222).

Tokekar, P., Vander Hook, J., Mulla, D., & Isler, V. (2013). Sensor planning for a symbiotic uav and ugv system for precision agriculture. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5321–5326).

Wang, F., Liu, P., Zhao, S., Chen, B. M., Phang, S. K., Lai, S., Lee, T. H., & Cai, C. (2014). Guidance, navigation and control of an unmanned helicopter for automatic cargo transportation. In *Proc. of the 33rd Chinese Control Conference* (pp. 1013–1020).

Yanguo, S. & Huanjin, W. (2009). Design of flight control system for a small unmanned tilt rotor aircraft. *Chinese Journal of Aeronautics*, 22(3), 250–256.

A

Tilt-Rotor UAV with Suspended Load Modeling

This appendix briefly describes the equations of motion of a Tilt-rotor UAV with suspended load from two different perspectives: (i) the UAV's perspective with the vehicle as a free body coupled to the load; (ii) the load's perspective with the load as a free body coupled to the UAV. Both models are obtained through the Euler-Lagrange formulation, in which the coupling between the aircraft and the load is considered naturally. Further details about the modeling process can be found in Almeida (2014) and Rego (2016), respectively, from the UAV's and the load's perspective.

The Tilt-rotor UAV with suspended load is shown in Figure A.1. The system can be seen as a multi-body mechanical system composed of four bodies: the Tilt-rotor UAV itself composed by three bodies, a main body and two thrusters' groups, and the suspended load being the fourth body. The system is actuated through the aircraft's thrusters group composed by a servomotor to tilt the propellers and a rotor to generate the lift force.

For modeling purposes, all bodies are assumed to be rigid; the load is assumed to be attached to the main body by a massless inelastic rope through two revolute joints; the rope is connected to the aircraft's geometric center; the main body's center of mass does not coincide with the aircraft's geometric center; and the thrusters groups' centers of mass are located at their respective tilting axes.

Figure A.1: The Tilt-rotor UAV with suspended load.

A.1 Model dynamics using Euler-Lagrange

The dynamics of a general mechanical system can be well described by its Lagrangian, which is defined as the difference between the kinetic, $\mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and the potential, $\mathcal{U}(\boldsymbol{q})$, energies,

$$\mathscr{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathcal{K}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \mathcal{U}(\boldsymbol{q}), \tag{A.1}$$

being $\boldsymbol{q}$ the generalized coordinates of the system.

The Euler-Lagrange equations of motion that describe the general mechanical system can be obtained through the Lagrangian by (Spong et al., 2006)

$$\frac{d}{dt}\left(\frac{\partial \mathscr{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \dot{\boldsymbol{q}}}\right) - \left(\frac{\partial \mathscr{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})}{\partial \boldsymbol{q}}\right) = \boldsymbol{\vartheta}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{A.2}$$

where $\boldsymbol{\vartheta}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the generalized force and torque vector composed by input forces, drag forces, and external disturbances forces.

The equation (A.2) can be rewritten in its canonical matricial form as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\vartheta}(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{A.3}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the Coriolis and centripetal forces matrix, and $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational force vector.

Therefore, in order to derive the equations of motion using the Euler-Lagrange formulation, the kinetic and potential energies of each body composing the mechanical system must be obtained. For each rigid body, these energies can be computed through the volume integrals (Siciliano et al., 2009)

$$\mathcal{K}_i = \frac{1}{2}\int_{V_i} \rho_i (\dot{\boldsymbol{p}}_i^{\mathcal{I}})^T (\dot{\boldsymbol{p}}_i^{\mathcal{I}}) \mathrm{d}V_i, \tag{A.4}$$

$$\mathcal{U}_i = -\int_{V_i} \rho_i \boldsymbol{g}^T \boldsymbol{p}_i^{\mathcal{I}} \mathrm{d}V_i, \tag{A.5}$$

where ρ_i is the density, V_i is the volume, $m_i = \int_{V_i} \rho_i V_i$ is the mass, $\boldsymbol{g} \triangleq [0\ 0\ -g_z]^T$ denotes the gravitational acceleration vector expressed in the inertial frame $\mathcal{I}$, and $\boldsymbol{p}_i^{\mathcal{I}}$ is a point rigidly attached to the i-th body expressed in $\mathcal{I}$.

The total kinetic and potential energy of the system can be obtained by the sum of each body kinetic and potential energies, i.e, $\mathcal{K} = \sum \mathcal{K}_i$ and $\mathcal{U} = \sum \mathcal{U}_i$ (Shabana, 2005). Thereafter, they can be used to compute the inertia matrix and the gravitational force vector, respectively, since

$$\mathcal{K} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{A.6}$$

and

$$\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial \mathcal{U}}{\partial \boldsymbol{q}}. \tag{A.7}$$

The Coriolis and centripetal forces matrix can be obtained from the inertia matrix through the Christoffel symbols of the first kind as (Spong et al., 2006)

$$c_{kj} = \sum_{i=1}^{10} \frac{1}{2}\left[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\right]\dot{q}_i, \tag{A.8}$$

where c_{kj} is the (i,j)-th entry of the matrix $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and m_{ij} is the (i,j)-th entry of the matrix $\boldsymbol{M}(\boldsymbol{q})$.

Finally, the last term that must be obtained to have the system's dynamics represented by equation (A.3) consists of the contributions of all non-conservative forces and torques applied to the system. Hence, let $\boldsymbol{f}$ and $\boldsymbol{\tau}$ denote, respectively, the general non-conservative forces applied in a point $\boldsymbol{p}$ and the general non-conservative torques applied to a body with a reference frame $\mathcal{F}$ rigidly attached to it. The contributions of $\boldsymbol{f}$ and $\boldsymbol{\tau}$ to the generalized force and torque vector $\boldsymbol{\vartheta}(\boldsymbol{q})$ can be obtained through the following mappings (Kane & Levinson, 1985)

$$\boldsymbol{\vartheta}_{\boldsymbol{f}} = (\mathcal{J}_{\boldsymbol{p}})^T \boldsymbol{f}^{\mathcal{I}} \in \mathbb{R}^n, \tag{A.9}$$

$$\boldsymbol{\vartheta}_{\boldsymbol{\tau}} = (\mathcal{W}_{\mathcal{F}})^T \boldsymbol{\tau}^{\mathcal{I}} \in \mathbb{R}^n, \tag{A.10}$$

with $\mathcal{J}_{\boldsymbol{p}} = \partial \dot{\boldsymbol{p}}^{\mathcal{I}} / \partial \dot{\boldsymbol{q}}$ and $\mathcal{W}_{\mathcal{F}} = \partial \boldsymbol{\omega}_{\mathcal{I}\mathcal{F}}^{\mathcal{I}} / \partial \dot{\boldsymbol{q}}$, where $\boldsymbol{\omega}_{\mathcal{I}\mathcal{F}}^{\mathcal{I}}$ is the angular velocity between the frames $\mathcal{I}$ and $\mathcal{F}$ expressed in $\mathcal{I}$.

Thus, to summarize the modeling process, first the system's frames definitions should be made in order to describe $\boldsymbol{p}_i^{\mathcal{I}}$ and $\dot{\boldsymbol{p}}_i^{\mathcal{I}}$ through basic kinematics transformations; second, the quadratic term $(\dot{\boldsymbol{p}}_i^{\mathcal{I}})^T (\dot{\boldsymbol{p}}_i^{\mathcal{I}})$ needs to be obtained to compute the kinetic energy and, using equation (A.6), identify the inertia matrix; third, the equation (A.8) together with the derived inertia matrix should be used to obtain the Coriolis and centripetal forces matrix; fourth, knowing $\boldsymbol{p}_i^{\mathcal{I}}$, the total potential energy needs to be calculated and differentiated using (A.7) to obtain the gravitational force vector; and fifth, the Jacobians $\mathcal{J}_{\boldsymbol{p}}$ and $\mathcal{W}_{\boldsymbol{p}}$ need to be calculated through equations (A.9) and (A.10) in order to derive the generalized

force and torque vector.

A.2 Dynamic model from the UAV's perspective

Aiming to obtain the kinematics from the UAV's perspective six frames are defined to describe the Tilt-rotor UAV motion with suspended load: the inertial frame $\mathcal{I}$, and the moving frames $\mathcal{B}$ and $\mathcal{C}_i$, which are, respectively, frames rigidly attached to the main body center of rotation and to the center of mass of the i-th body (see Figure A.2).

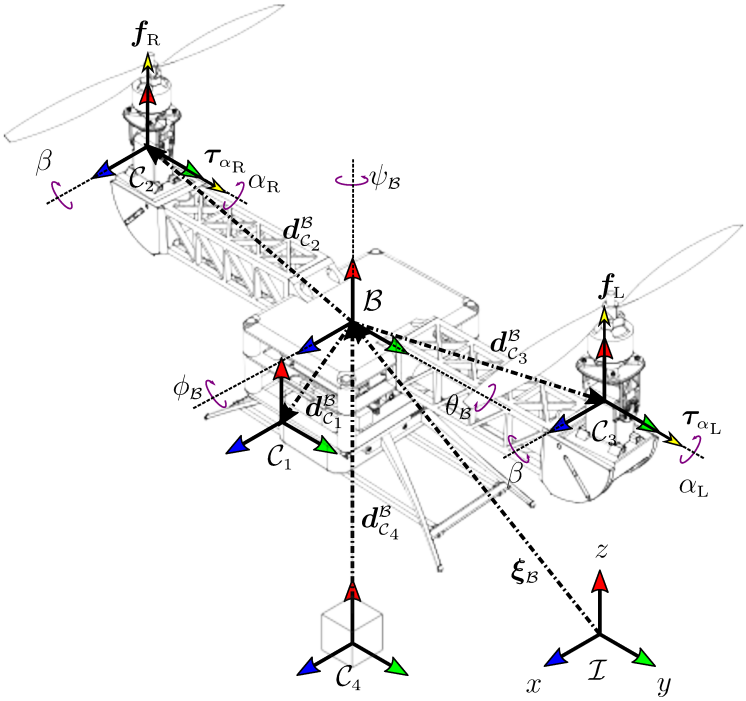


Figure A.2: Tilt-rotor UAV frames and variables definition considering the system from the UAV's perspective.

The position of the body frame's origin represented in the inertial frame is given by $\boldsymbol{\xi}_\mathcal{B} = [x_\mathcal{B}\ y_\mathcal{B}\ z_\mathcal{B}]^T$, and the attitude by $\boldsymbol{\eta}_\mathcal{B} = [\phi_\mathcal{B}\ \theta_\mathcal{B}\ \psi_\mathcal{B}]^T$, described by Euler angles using the ZYX convention about local axes. The orientation of $\mathcal{C}_2$ with respect to $\mathcal{B}$ is obtained by a rotation α_R around the $\mathrm{Y}^\mathcal{B}$-axis and a constant negative tilt β around the $\mathrm{X}^\mathcal{B}$-axis. Likewise, the rotation of the frame $\mathcal{C}_3$ with respect to $\mathcal{B}$ can be obtained considering rotations α_L around the $\mathrm{Y}^\mathcal{B}$-axis and a constant tilt β around the $\mathrm{X}^\mathcal{B}$-axis.[1] Moreover, using a parametrization that considers the load as a pendulum with length l and two degrees of freedom represented by γ_1 and γ_2 (rotations around $\mathrm{X}^\mathcal{B}$-axis and $\mathrm{Y}^\mathcal{B}$-axis, respectively) (Almeida & Raffo, 2015); the position of the suspended load with respect to the main body becomes a simple forward kinematic problem.

[1] β is a fixed tilt angle introduced in the mechanical design seeking to increase the controllability along the $\mathrm{Y}^\mathcal{B}$-axis (Raffo et al., 2011).

Therefore, the generalized coordinates that describe the motion of the Tilt-rotor UAV are chosen as

$$\boldsymbol{q}_{\mathcal{B}} = [\boldsymbol{\xi}_{\mathcal{B}}^T \ \boldsymbol{\eta}_{\mathcal{B}}^T \ \boldsymbol{\alpha}^T \ \boldsymbol{\gamma}^T]^T, \tag{A.11}$$

with $\boldsymbol{\alpha} = [\alpha_R \ \alpha_L]^T$ and $\boldsymbol{\gamma} = [\gamma_1 \ \gamma_2]^T$.

Furthermore, the system's inputs are given by $\boldsymbol{u} = [f_R \ f_L \ \tau_R \ \tau_L]^T$, where f_R and f_L are the thrust forces, respectively, of the right and left propellers; likewise, τ_R and τ_L are the torques applied by the servomotors. Additionally, the system underactuated behavior can be noticed from the generalized coordinates definition and the amount of inputs available.

A.2.1 Kinematics

Considering the Euler angles using the ZYX convention about local axes, the rotation matrix between the frames $\mathcal{B}$ and $\mathcal{I}$ is given by (Spong et al., 2006)

$$\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} = \begin{bmatrix} c_{\psi_{\mathcal{B}}} c_{\theta_{\mathcal{B}}} & c_{\psi_{\mathcal{B}}} s_{\theta_{\mathcal{B}}} s_{\phi_{\mathcal{B}}} - s_{\psi_{\mathcal{B}}} c_{\phi_{\mathcal{B}}} & c_{\psi_{\mathcal{B}}} s_{\theta_{\mathcal{B}}} c_{\phi_{\mathcal{B}}} + s_{\psi_{\mathcal{B}}} s_{\phi_{\mathcal{B}}} \\ s_{\psi_{\mathcal{B}}} c_{\theta_{\mathcal{B}}} & s_{\psi_{\mathcal{B}}} s_{\theta_{\mathcal{B}}} s_{\phi_{\mathcal{B}}} + c_{\psi_{\mathcal{B}}} c_{\phi_{\mathcal{B}}} & s_{\psi_{\mathcal{B}}} s_{\theta_{\mathcal{B}}} c_{\phi_{\mathcal{B}}} - c_{\psi_{\mathcal{B}}} s_{\phi_{\mathcal{B}}} \\ -s_{\theta_{\mathcal{B}}} & c_{\theta_{\mathcal{B}}} s_{\phi_{\mathcal{B}}} & c_{\theta_{\mathcal{B}}} c_{\phi_{\mathcal{B}}} \end{bmatrix}, \tag{A.12}$$

with $s_{(\cdot)} = \sin(\cdot)$ and $c_{(\cdot)} = \cos(\cdot)$.

A point rigidly attached to the main body can be represented in the inertial frame by

$$\boldsymbol{p}_{\mathcal{B}}^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \boldsymbol{p}^{\mathcal{B}} + \boldsymbol{\xi}_{\mathcal{B}}, \tag{A.13}$$

likewise, a point rigidly attached to the frame $\mathcal{C}_i$ can be expressed in the body frame as

$$\boldsymbol{p}_i^{\mathcal{B}} = \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{p}^{\mathcal{C}_i} + \boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}, \quad \forall i, \tag{A.14}$$

where $\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}$ denotes the translation between the frames $\mathcal{B}$ and $\mathcal{C}_i$ with $i \in \{1, 2, 3, 4\}$.

Therefore, combining the equations (A.13) and (A.14), a point rigidly attached to $\mathcal{C}_i$ can be expressed in the inertial frame by

$$\boldsymbol{p}_i^{\mathcal{I}} = \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \left(\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{p}^{\mathcal{C}_i} + \boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}} \right) + \boldsymbol{\xi}_{\mathcal{B}}, \quad \forall i. \tag{A.15}$$

Furthermore, taking the first time derivative of equation (A.15), it is possible to express the velocity of the *i-th* body in the inertial frame as

$$\dot{\boldsymbol{p}}_i^{\mathcal{I}} = \dot{\boldsymbol{R}}_{\mathcal{B}}^{\mathcal{I}} \left(\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{p}^{\mathcal{C}_i} + \boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}} \right) + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \left(\dot{\boldsymbol{R}}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{p}^{\mathcal{C}_i} + \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \dot{\boldsymbol{p}}^{\mathcal{C}_i} + \dot{\boldsymbol{d}}_{\mathcal{C}_i}^{\mathcal{B}} \right) + \dot{\boldsymbol{\xi}}_{\mathcal{B}}. \tag{A.16}$$

In order to be able of expanding the derivative terms of the equation (A.16), since $\boldsymbol{R}_i^j = \boldsymbol{R}_i^j \boldsymbol{S}(\boldsymbol{\omega}_{ij}^i)$ with $\boldsymbol{S}(\cdot)$ being a skew symmetric matrix (Spong et al., 2006), the angular velocities $\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}}$, $\boldsymbol{\omega}_{\mathcal{C}_2\mathcal{B}}^{\mathcal{C}_2}$, $\boldsymbol{\omega}_{\mathcal{C}_3\mathcal{B}}^{\mathcal{C}_3}$, and $\boldsymbol{\omega}_{\mathcal{C}_4\mathcal{B}}^{\mathcal{C}_4}$, and the relative velocity $\dot{\boldsymbol{d}}_{\mathcal{C}_4}^{\mathcal{B}}$ between the frames $\mathcal{C}_4$

and $\mathcal{B}$ must to be obtained. As shown in Almeida (2014), these terms can be described by

$$\boldsymbol{\omega}_{\mathcal{B}\mathcal{I}}^{\mathcal{B}} = \boldsymbol{W}_{\eta_{\mathcal{B}}}\dot{\boldsymbol{\eta}}_{\mathcal{B}}, \tag{A.17}$$

$$\boldsymbol{\omega}_{\mathcal{C}_2\mathcal{B}}^{\mathcal{C}_2} = \dot{\alpha}_R \boldsymbol{a}_y, \tag{A.18}$$

$$\boldsymbol{\omega}_{\mathcal{C}_3\mathcal{B}}^{\mathcal{C}_3} = \dot{\alpha}_L \boldsymbol{a}_y, \tag{A.19}$$

$$\boldsymbol{\omega}_{\mathcal{C}_4\mathcal{B}}^{\mathcal{C}_4} = \boldsymbol{P}\boldsymbol{\gamma}, \tag{A.20}$$

$$\dot{\boldsymbol{d}}_{\mathcal{C}_4}^{\mathcal{B}} = \boldsymbol{L}\boldsymbol{\gamma}, \tag{A.21}$$

where

$$\boldsymbol{W}_{\eta_{\mathcal{B}}} = \begin{bmatrix} 1 & 0 & -s_{\theta_{\mathcal{B}}} \\ 0 & c_{\phi_{\mathcal{B}}} & s_{\phi_{\mathcal{B}}} c_{\theta_{\mathcal{B}}} \\ 0 & -s_{\phi_{\mathcal{B}}} & c_{\phi_{\mathcal{B}}} c_{\theta_{\mathcal{B}}} \end{bmatrix}, \boldsymbol{a}_y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \boldsymbol{P} = \begin{bmatrix} 1 & 0 \\ 0 & c_{\gamma_1} \\ 0 & -s_{\gamma_1} \end{bmatrix}, \boldsymbol{L} = \begin{bmatrix} l s_{\gamma_1} s_{\gamma_2} & -l c_{\gamma_1} c_{\gamma_2} \\ l c_{\gamma_1} & 0 \\ l s_{\gamma_1} c_{\gamma_2} & l c_{\gamma_1} s_{\gamma_2} \end{bmatrix}.$$

A.2.2 Dynamic model

The nonlinear dynamic model of the Tilt-rotor UAV with suspended load, from the UAV's perspective, can be described by the Euler-Lagrange equation, as in (A.3), by

$$\boldsymbol{M}_{\mathcal{B}}(\boldsymbol{q}_{\mathcal{B}})\ddot{\boldsymbol{q}}_{\mathcal{B}} + \boldsymbol{C}_{\mathcal{B}}(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}})\dot{\boldsymbol{q}}_{\mathcal{B}} + \boldsymbol{G}_{\mathcal{B}}(\boldsymbol{q}_{\mathcal{B}}) = \boldsymbol{\vartheta}_{\mathcal{B}}(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}}). \tag{A.22}$$

After obtained $\dot{\boldsymbol{p}}_i^{\mathcal{I}}$ in equation (A.16), the quadratic term $(\dot{\boldsymbol{p}}_i^{\mathcal{I}})^T(\dot{\boldsymbol{p}}_i^{\mathcal{I}})$ can be computed and the kinetic energy can be represented as in equation (A.6) in order to obtain the inertia matrix. Therefore, as shown in detail in Almeida (2014), the inertia matrix can be written as

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

where the $*$ terms indicate symmetry with respect to the main diagonal and

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

with $m = \sum m_i$, $\boldsymbol{J} = \sum \boldsymbol{J}_i$ and $\boldsymbol{H} = \boldsymbol{S}\left(\sum m_i \boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}\right)$, being $\boldsymbol{J}_i$ the inertia tensor of the i-th body given by

$$\boldsymbol{J}_i = \boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} \boldsymbol{I}_i \left(\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}}\right)^T + m_i \boldsymbol{S}\left(\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}\right)^T \boldsymbol{S}\left(\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}\right), \tag{A.24}$$

where $\boldsymbol{I}_i$ is the moment of inertia of the i-th body.

Further, the Coriolis and centripetal forces matrix $\boldsymbol{C}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}}\right)$ can be obtained directly from the inertia matrix using equation (A.8).

As shown in Almeida (2014), the total system's potential energy is given by

$$\mathcal{U}_{\mathcal{B}} = \left(\boldsymbol{g}^{\mathcal{I}}\right)^T \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \left(\sum_{i=1}^{4} m_i \boldsymbol{d}_{C_i}^{\mathcal{B}}\right) + \left(\boldsymbol{g}^{\mathcal{I}}\right)^T m \boldsymbol{\xi}_{\mathcal{B}}. \tag{A.25}$$

Thus, the gravitational force vector $\boldsymbol{G}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right)$ can be obtained using equation (A.7).

The generalized force and torque vector $\boldsymbol{\vartheta}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}}\right)$ can be described as a combination of forces and torques generated by the actuators, $\boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{u}}$, the friction between the rope and the Tilt-rotor, $\boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{fr}}$, and the external disturbances applied to the vehicle, $\boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{d}}$. Hence, the generalized force and torque vector can be expressed as

$$\boldsymbol{\vartheta}_{\mathcal{B}} = \boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{u}} + \boldsymbol{\vartheta}_{\mathcal{B}\mathrm{fr}} + \boldsymbol{\vartheta}_{\mathcal{B}\mathrm{d}}, \tag{A.26}$$

where

$$\boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{u}} = \boldsymbol{B}\boldsymbol{u} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{r}_R & \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}\boldsymbol{r}_L & 0 & 0 \\ \boldsymbol{W}_{\eta_{\mathcal{B}}}^T\boldsymbol{\tau}_R & \boldsymbol{W}_{\eta_{\mathcal{B}}}^T\boldsymbol{\tau}_L & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_R \\ f_L \\ \tau_{\alpha_R} \\ \tau_{\alpha_L} \end{bmatrix}, \tag{A.27}$$

$$\boldsymbol{\vartheta}_{\mathcal{B}\mathrm{fr}} = -\boldsymbol{\mu}\dot{\boldsymbol{q}}_{\mathcal{B}} = -\mathbf{blkdiag}\left(0,0,0,0,0,0,0,0,\mu_\gamma,\mu_\gamma\right)\dot{\boldsymbol{q}}_{\mathcal{B}}, \tag{A.28}$$

$$\boldsymbol{\vartheta}_{\mathcal{B}\boldsymbol{d}} = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times2} \end{bmatrix}^T \boldsymbol{d}_{\mathcal{B}}, \tag{A.29}$$

with $\boldsymbol{d}_{\mathcal{B}} = \begin{bmatrix} d_{x\mathcal{B}} & d_{y\mathcal{B}} & d_{z\mathcal{B}} \end{bmatrix}^T$ denoting the disturbance vector, and

$$\boldsymbol{r}_R = \begin{bmatrix} s_{\alpha_R} \\ c_{\alpha_R}s_\beta \\ c_{\alpha_R}c_\beta \end{bmatrix}, \boldsymbol{r}_L = \begin{bmatrix} s_{\alpha_L} \\ -c_{\alpha_L}s_\beta \\ c_{\alpha_L}c_\beta \end{bmatrix}, \boldsymbol{\tau}_R = \begin{bmatrix} -c_{\alpha_R}c_\beta d^y - \frac{k_\tau}{b}s_{\alpha_R} \\ s_{\alpha_R}d^z + \frac{k_\tau}{b}c_{\alpha_R}s_\beta \\ s_{\alpha_R}d^y + \frac{k_\tau}{b}c_{\alpha_R}c_\beta \end{bmatrix}, \boldsymbol{\tau}_L = \begin{bmatrix} c_{\alpha_L}c_\beta d^y + \frac{k_\tau}{b}s_{\alpha_L} \\ s_{\alpha_L}d^z - \frac{k_\tau}{b}c_{\alpha_L}s_\beta \\ -s_{\alpha_L}d^y - \frac{k_\tau}{b}c_{\alpha_L}c_\beta \end{bmatrix},$$

being $d^z = d_{C_2z}^{\mathcal{B}} = d_{C_3z}^{\mathcal{B}}$ and $d^y = |d_{C_2y}^{\mathcal{B}}|$. The derivation of each term of the equation (A.26) can be found with detail in Almeida (2014).

A.2.3 State-space representation

Considering the obtained inertia matrix, $\boldsymbol{M}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right)$, the Coriolis and centripetal forces matrix, $\boldsymbol{C}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}}\right)$, the gravitational force vector, $\boldsymbol{G}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right)$, and the generalized torque and force vector, $\boldsymbol{\vartheta}_{\mathcal{B}}$, it is possible to write the Euler-Lagrange equations of motion in a state-space

representation, yielding to

$$\dot{\boldsymbol{x}}_{\mathcal{B}} = \boldsymbol{\varphi}_{\mathcal{B}}\left(\boldsymbol{x}_{\mathcal{B}}, \boldsymbol{u}, \boldsymbol{d}\right) = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{B}} \\ \ddot{\boldsymbol{q}}_{\mathcal{B}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{B}} \\ \boldsymbol{M}_{\mathcal{B}}^{-1}\left[\boldsymbol{\vartheta}_{\mathcal{B}} - \boldsymbol{C}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}, \dot{\boldsymbol{q}}_{\mathcal{B}}\right)\dot{\boldsymbol{q}}_{\mathcal{B}} - \boldsymbol{G}_{\mathcal{B}}\left(\boldsymbol{q}_{\mathcal{B}}\right)\right] \end{bmatrix}. \tag{A.30}$$

The model derived in this section is used to design the controllers of Chapter 4 and to obtain the nonlinear cascade structure presented in Appendix B.

A.3 Dynamic model from the load's perspective

In order to formulate the dynamic model from the load's perspective, six reference frames are defined, as shown in Figure A.3: the inertial frame $\mathcal{I}$, the moving frames $\mathcal{B}$ and $\mathcal{C}_i$, which are, respectively, frames rigidly attached to the main body center of rotation and to the center of mass of the i-th body, and the suspended load group center of mass frame, $\mathcal{L}$. The load's position with respect to the inertial frame $\mathcal{I}$ is denoted by $\boldsymbol{\xi}_{\mathcal{L}} = [x_{\mathcal{L}}\ y_{\mathcal{L}}\ z_{\mathcal{L}}]^T$, and its attitude by $\boldsymbol{\eta}_{\mathcal{L}} = [\phi_{\mathcal{L}}\ \theta_{\mathcal{L}}\ \psi_{\mathcal{L}}]^T$, described by Euler angles using the ZYX convention about local axes. The displacement vector from $\mathcal{L}$ to $\mathcal{B}$ corresponds to the rope, and is expressed in $\mathcal{L}$ by $\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}} = [0\ 0\ l]^T$, with l being the rope's length. The displacement vectors from $\mathcal{B}$ to $\mathcal{C}_i$ expressed in $\mathcal{B}$ are model parameters of the Tilt-rotor UAV and are denoted by $\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}$ with $i \in \{1, 2, 3\}$. Further, the orientation of the aircraft's geometric center frame with respect to $\mathcal{L}$ is parametrized by two angles, $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2]^T$; and the orientations of the thrusters' groups with respect to $\mathcal{B}$ is parametrized by $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{\mathrm{R}} & \alpha_{\mathrm{L}} \end{bmatrix}^T$.

Aiming to obtain the equations of motion describing explicitly the time evolution of the load's position and orientation, the generalized coordinates are chosen as

$$\boldsymbol{q}_{\mathcal{L}} = \begin{bmatrix} \boldsymbol{\xi}_{\mathcal{L}}^T & \boldsymbol{\eta}_{\mathcal{L}}^T & \boldsymbol{\gamma}^T & \boldsymbol{\alpha}^T \end{bmatrix}^T. \tag{A.31}$$

As considered in the previous section, the system's inputs are the thrust forces and torques given by $\boldsymbol{u} = [f_R\ f_L\ \tau_R\ \tau_L]^T$, which show again the system's underactuated behavior.

A.3.1 Kinematics

Considering the Euler angles vector $\boldsymbol{\eta}_{\mathcal{L}}$ to describe the orientation of the load with respect to $\mathcal{I}$, the associated rotation matrix can be defined as

$$\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}} = \begin{bmatrix} \mathrm{C}_{\psi_{\mathcal{L}}}\mathrm{C}_{\theta_{\mathcal{L}}} & \mathrm{C}_{\psi_{\mathcal{L}}}\mathrm{S}_{\theta_{\mathcal{L}}}\mathrm{S}_{\phi_{\mathcal{L}}} - \mathrm{S}_{\psi_{\mathcal{L}}}\mathrm{C}_{\phi_{\mathcal{L}}} & \mathrm{C}_{\psi_{\mathcal{L}}}\mathrm{S}_{\theta_{\mathcal{L}}}\mathrm{C}_{\phi_{\mathcal{L}}} + \mathrm{S}_{\psi_{\mathcal{L}}}\mathrm{S}_{\phi_{\mathcal{L}}} \\ \mathrm{S}_{\psi_{\mathcal{L}}}\mathrm{C}_{\theta_{\mathcal{L}}} & \mathrm{S}_{\psi_{\mathcal{L}}}\mathrm{S}_{\theta_{\mathcal{L}}}\mathrm{S}_{\phi_{\mathcal{L}}} + \mathrm{C}_{\psi_{\mathcal{L}}}\mathrm{C}_{\phi_{\mathcal{L}}} & \mathrm{S}_{\psi_{\mathcal{L}}}\mathrm{S}_{\theta_{\mathcal{L}}}\mathrm{C}_{\phi_{\mathcal{L}}} - \mathrm{C}_{\psi_{\mathcal{L}}}\mathrm{S}_{\phi_{\mathcal{L}}} \\ -\mathrm{S}_{\theta_{\mathcal{L}}} & \mathrm{C}_{\theta_{\mathcal{L}}}\mathrm{S}_{\phi_{\mathcal{L}}} & \mathrm{C}_{\theta_{\mathcal{L}}}\mathrm{C}_{\phi_{\mathcal{L}}} \end{bmatrix}.$$

On the other hand, the rotation matrices associated, respectively, to the orientation of the aircraft's geometric center frame with respect to $\mathcal{L}$ and the orientations of the thrusters'

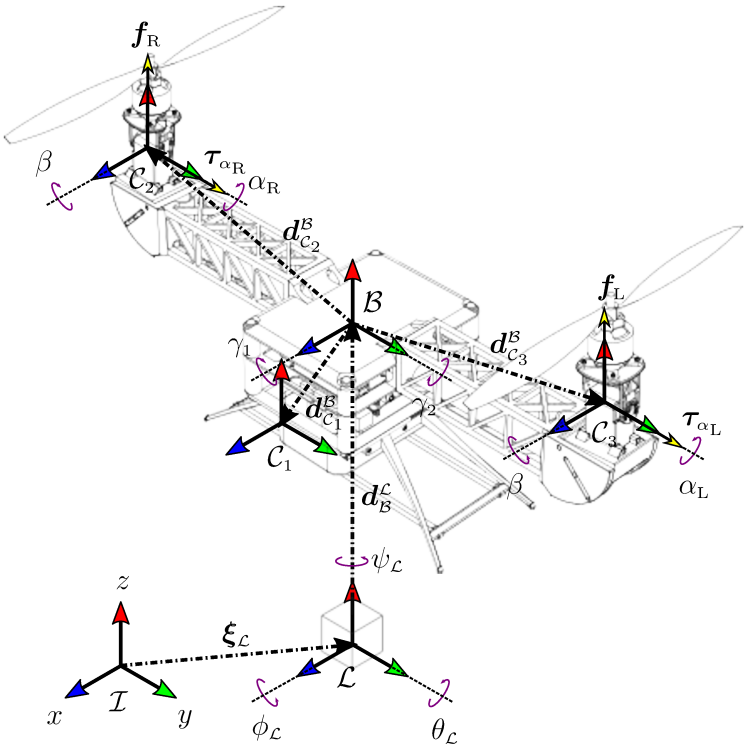


Figure A.3: Tilt-rotor UAV frames and variables definition considering the system from the load's perspective (adapted from Rego (2016)).

groups with respect to $\mathcal{B}$ are given by

$$\boldsymbol{R}^{\mathcal{L}}_{\mathcal{B}} = \begin{bmatrix} \mathrm{c}_{\gamma_2} & 0 & -\mathrm{s}_{\gamma_2} \\ \mathrm{s}_{\gamma_1}\mathrm{s}_{\gamma_2} & \mathrm{c}_{\gamma_1} & \mathrm{s}_{\gamma_1}\mathrm{c}_{\gamma_2} \\ \mathrm{c}_{\gamma_1}\mathrm{s}_{\gamma_2} & -\mathrm{s}_{\gamma_1} & \mathrm{c}_{\gamma_1}\mathrm{c}_{\gamma_2} \end{bmatrix}, \; \boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_2} = \begin{bmatrix} \mathrm{c}_{\alpha_{\mathrm{R}}} & 0 & \mathrm{s}_{\alpha_{\mathrm{R}}} \\ -\mathrm{s}_{\beta}\mathrm{s}_{\alpha_{\mathrm{R}}} & \mathrm{c}_{\beta} & \mathrm{s}_{\beta}\mathrm{c}_{\alpha_{\mathrm{R}}} \\ -\mathrm{c}_{\beta}\mathrm{s}_{\alpha_{\mathrm{R}}} & -\mathrm{s}_{\beta} & \mathrm{c}_{\beta}\mathrm{c}_{\alpha_{\mathrm{R}}} \end{bmatrix}, \; \boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_3} = \begin{bmatrix} \mathrm{c}_{\alpha_{\mathrm{L}}} & 0 & \mathrm{s}_{\alpha_{\mathrm{L}}} \\ \mathrm{s}_{\beta}\mathrm{s}_{\alpha_{\mathrm{L}}} & \mathrm{c}_{\beta} & -\mathrm{s}_{\beta}\mathrm{c}_{\alpha_{\mathrm{L}}} \\ -\mathrm{c}_{\beta}\mathrm{s}_{\alpha_{\mathrm{L}}} & \mathrm{s}_{\beta} & \mathrm{c}_{\beta}\mathrm{c}_{\alpha_{\mathrm{L}}} \end{bmatrix}.$$

The angular velocities of the system are given by $\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}\mathcal{I}} = \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}\dot{\boldsymbol{\eta}}_{\mathcal{L}}$, $\boldsymbol{\omega}^{\mathcal{B}}_{\mathcal{B}\mathcal{L}} = \boldsymbol{Q}\dot{\boldsymbol{\gamma}}$, $\boldsymbol{\omega}^{\mathcal{C}_1}_{\mathcal{C}_1\mathcal{B}} = \mathbf{0}_{3\times 1}$, $\boldsymbol{\omega}^{\mathcal{C}_2}_{\mathcal{C}_2\mathcal{B}} = \boldsymbol{a}_y\dot{\alpha}_{\mathrm{R}}$, and $\boldsymbol{\omega}^{\mathcal{C}_3}_{\mathcal{C}_3\mathcal{B}} = \boldsymbol{a}_y\dot{\alpha}_{\mathrm{L}}$, where, for instance, $\boldsymbol{\omega}^{\mathcal{L}}_{\mathcal{L}\mathcal{I}}$ denotes the angular velocity of $\mathcal{L}$ with respect to $\mathcal{I}$ expressed in $\mathcal{L}$, with

$$\boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}} = \begin{bmatrix} 1 & 0 & -\mathrm{s}_{\theta_{\mathcal{L}}} \\ 0 & \mathrm{c}_{\phi_{\mathcal{L}}} & \mathrm{s}_{\phi_{\mathcal{L}}}\mathrm{c}_{\theta_{\mathcal{L}}} \\ 0 & -\mathrm{s}_{\phi_{\mathcal{L}}} & \mathrm{c}_{\phi_{\mathcal{L}}}\mathrm{c}_{\theta_{\mathcal{L}}} \end{bmatrix}, \qquad \boldsymbol{Q} = \begin{bmatrix} -\mathrm{c}_{\gamma_2} & 0 \\ 0 & -1 \\ \mathrm{s}_{\gamma_2} & 0 \end{bmatrix}, \tag{A.32}$$

and $\boldsymbol{a}_y = [0 \; 1 \; 0]^T$.

From the rigid transformations of the system, the forward kinematics of points that belong to each rigid body are given by

$$\boldsymbol{p}^{\mathcal{I}}_{\mathcal{L}} = \boldsymbol{\xi}_{\mathcal{L}} + \boldsymbol{R}^{\mathcal{I}}_{\mathcal{L}}\boldsymbol{p}^{\mathcal{L}}, \tag{A.33}$$

$$\boldsymbol{p}^{\mathcal{I}}_{i} = \boldsymbol{\xi}_{\mathcal{L}} + \boldsymbol{R}^{\mathcal{I}}_{\mathcal{L}}\boldsymbol{d}^{\mathcal{L}}_{\mathcal{B}} + \boldsymbol{R}^{\mathcal{I}}_{\mathcal{L}}\boldsymbol{R}^{\mathcal{L}}_{\mathcal{B}}\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_i} + \boldsymbol{R}^{\mathcal{I}}_{\mathcal{L}}\boldsymbol{R}^{\mathcal{L}}_{\mathcal{B}}\boldsymbol{R}^{\mathcal{B}}_{\mathcal{C}_i}\boldsymbol{p}^{\mathcal{C}_i}, \tag{A.34}$$

where $\boldsymbol{p}^{\mathcal{I}}_{\mathcal{L}}$ is the position of a point rigid attached to the suspended load body expressed in

the inertial frame, and $\boldsymbol{p}_i^{\mathcal{I}}$ is a point rigid attached to the frame $\mathcal{C}_i$ expressed in the inertial frame. As shown in Rego (2016), the time derivatives of the equations (A.33) and (A.34) can be written as

$$\dot{\boldsymbol{p}}_{\mathcal{L}}^{\mathcal{I}} = \dot{\boldsymbol{\xi}}_{\mathcal{L}} + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}} S(\boldsymbol{p}^{\mathcal{L}})^T \boldsymbol{\omega}_{\mathcal{L}\mathcal{I}}^{\mathcal{L}}, \tag{A.35}$$

$$\begin{aligned}\dot{\boldsymbol{p}}_i^{\mathcal{I}} = \dot{\boldsymbol{\xi}}_{\mathcal{L}} &+ \left[\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}} S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})^T + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}})^T (\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}})^T + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} S(\boldsymbol{p}^{\mathcal{C}_i})^T (\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}})^T\right] \boldsymbol{\omega}_{\mathcal{L}\mathcal{I}}^{\mathcal{L}} \\ &+ \left[\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}})^T + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} S(\boldsymbol{p}^{\mathcal{C}_i})^T (\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}})^T\right] \boldsymbol{\omega}_{\mathcal{B}\mathcal{L}}^{\mathcal{B}} + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_i}^{\mathcal{B}} S(\boldsymbol{p}^{\mathcal{C}_i})^T \boldsymbol{\omega}_{\mathcal{C}_i\mathcal{B}}^{\mathcal{C}_i}.\end{aligned} \tag{A.36}$$

A.3.2 Dynamic model

The nonlinear dynamic model of the Tilt-rotor UAV with suspended load from the load's perspective can be described by the Euler-Lagrange equations of motion as

$$\boldsymbol{M}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}})\,\ddot{\boldsymbol{q}}_{\mathcal{L}} + \boldsymbol{C}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}, \dot{\boldsymbol{q}}_{\mathcal{L}})\,\dot{\boldsymbol{q}}_{\mathcal{L}} + \boldsymbol{G}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}) = \boldsymbol{\vartheta}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}, \dot{\boldsymbol{q}}_{\mathcal{L}}), \tag{A.37}$$

After derive $\boldsymbol{p}_{\mathcal{L}}^{\mathcal{I}}$ and $\boldsymbol{p}_i^{\mathcal{I}}$ with equations (A.33) and (A.34), the quadratic terms $(\dot{\boldsymbol{p}}_{\mathcal{L}}^{\mathcal{I}})^T(\dot{\boldsymbol{p}}_{\mathcal{L}}^{\mathcal{I}})$ and $(\dot{\boldsymbol{p}}_i^{\mathcal{I}})^T(\dot{\boldsymbol{p}}_i^{\mathcal{I}})$ can be evaluated and the kinetic energy can be represented as in equation (A.6) in order to obtain the inertia matrix. Therefore, the inertia matrix can be written as (Rego, 2016)

$$\boldsymbol{M}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}) = \begin{bmatrix} (m_{\mathcal{L}}+m)\mathbb{I}_{3\times3} & \boldsymbol{m}_{12} & -\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_m)\boldsymbol{Q} & \boldsymbol{0}_{3\times1} & \boldsymbol{0}_{3\times1} \\ * & \boldsymbol{m}_{22} & \boldsymbol{m}_{23} & \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^T \boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{I}_2\boldsymbol{a}_y & \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^T \boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{I}_3\boldsymbol{a}_y \\ * & * & \boldsymbol{Q}^T\boldsymbol{J}\boldsymbol{Q} & \boldsymbol{Q}^T\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{I}_2\boldsymbol{a}_y & \boldsymbol{Q}^T\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{I}_3\boldsymbol{a}_y \\ * & * & * & \boldsymbol{a}_y^T\boldsymbol{I}_2\boldsymbol{a}_y & 0 \\ * & * & * & * & \boldsymbol{a}_y^T\boldsymbol{I}_3\boldsymbol{a}_y \end{bmatrix}, \tag{A.38}$$

with $*$ denoting terms that are deduced by symmetry, and

$$\begin{aligned}\boldsymbol{m}_{12} &= -m\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}} S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}} - \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_m)(\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}})^T\boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}, \\ \boldsymbol{m}_{22} &= \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^T\left[\boldsymbol{I}_{\mathcal{L}} + \boldsymbol{D} - S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_m)(\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}})^T - \boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_m)(\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}})^T S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\right]\boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}, \\ \boldsymbol{m}_{23} &= \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^T\left[-S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_m) + \boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{J}\right]\boldsymbol{Q},\end{aligned}$$

where $m = \sum_{i=1}^{3} m_i$, $\boldsymbol{J} = \sum_{i=1}^{3}\boldsymbol{J}_i$, $\boldsymbol{D} = \sum_{i=1}^{3}\boldsymbol{D}_i$, and $\boldsymbol{d}_m = \sum_{i=1}^{3} m_i\boldsymbol{d}_{\mathcal{C}_i}^{\mathcal{B}}$.

The Coriolis and centripetal forces matrix can be easily obtained from the inertia matrix using equation (A.8). Besides, as shown in Rego (2016), the total potential energy is given by

$$\mathcal{U}_{\mathcal{L}} = -\boldsymbol{g}^T\left[(m_{\mathcal{L}}+m)\boldsymbol{\xi}_{\mathcal{L}} + m\boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}} + \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{d}_m\right]. \tag{A.39}$$

Therefore, the gravitational force vector can be obtained using equation (A.7).

The last term that needs to be obtained to completely describe the system from the load's perspective through the equation (A.37) is the generalized force and torque vector,

$\boldsymbol{\vartheta}_{\mathcal{L}}$, that can be described as (Rego, 2016)

$$\boldsymbol{\vartheta}_{\mathcal{L}} = \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{R}}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{L}}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{R}}}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{L}}}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,R}}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,L}}} + \boldsymbol{\vartheta}_{\mathcal{L}\mathrm{fr}} + \boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{d}}, \tag{A.40}$$

where $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{R}}}$ and $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{L}}}$ represents, respectively, the contribution of the lift force generated by the right and the left propellers; $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{R}}}}$ and $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{L}}}}$ are, respectively, the contribution of the torques generated by the right and left servomotors; $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,R}}}$ and $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,L}}}$ are, respectively, the effect of the drag torques generated by the right and left propellers; $\boldsymbol{\vartheta}_{\mathcal{L}\mathrm{fr}}$ denotes the contribution of the viscous friction at the point of connection between the rope and the Tilt-rotor UAV; and $\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{d}}$ is the external disturbances applied to the suspended load. All terms of equation (A.40) were carefully derived in Rego (2016) and can be written as

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{R}}} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T} S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z + \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_{\mathcal{C}_2}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{Q}^T S(\boldsymbol{d}_{\mathcal{C}_2}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z \\ 0 \\ 0 \end{bmatrix} f_{\mathrm{R}}, \tag{A.41}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{f}_{\mathrm{L}}} = \begin{bmatrix} \boldsymbol{R}_{\mathcal{L}}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T} S(\boldsymbol{d}_{\mathcal{B}}^{\mathcal{L}})\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z + \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}} S(\boldsymbol{d}_{\mathcal{C}_3}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{Q}^T S(\boldsymbol{d}_{\mathcal{C}_3}^{\mathcal{B}})\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z \\ 0 \\ 0 \end{bmatrix} f_{\mathrm{L}}, \tag{A.42}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{R}}}} = \begin{bmatrix} (\mathbf{0}_{3\times1})^T & (\mathbf{0}_{3\times1})^T & (\mathbf{0}_{2\times1})^T & 1 & 0 \end{bmatrix}^T \tau_{\alpha_{\mathrm{R}}}, \tag{A.43}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\alpha_{\mathrm{L}}}} = \begin{bmatrix} (\mathbf{0}_{3\times1})^T & (\mathbf{0}_{3\times1})^T & (\mathbf{0}_{2\times1})^T & 0 & 1 \end{bmatrix}^T \tau_{\alpha_{\mathrm{L}}}, \tag{A.44}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,R}}} = \lambda_{\mathrm{R}}\frac{k_\tau}{b}\begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{Q}^T\boldsymbol{R}_{\mathcal{C}_2}^{\mathcal{B}}\boldsymbol{a}_z \\ 0 \\ 0 \end{bmatrix} f_{\mathrm{R}}, \tag{A.45}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{\tau}_{\mathrm{drag,L}}} = \lambda_{\mathrm{L}}\frac{k_\tau}{b}\begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{W}_{\boldsymbol{\eta}_{\mathcal{L}}}^{T}\boldsymbol{R}_{\mathcal{B}}^{\mathcal{L}}\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z \\ \boldsymbol{Q}^T\boldsymbol{R}_{\mathcal{C}_3}^{\mathcal{B}}\boldsymbol{a}_z \\ 0 \\ 0 \end{bmatrix} f_{\mathrm{L}}, \tag{A.46}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\mathrm{fr}} = -\mathbf{blkdiag}(0,0,0,0,0,0,\mu_\gamma,\mu_\gamma,0,0)\dot{\boldsymbol{q}}_{\mathcal{L}}, \tag{A.47}$$

$$\boldsymbol{\vartheta}_{\mathcal{L}\boldsymbol{d}} = \begin{bmatrix} \mathbb{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times2} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix}^T \boldsymbol{d}_{\mathcal{L}}, \tag{A.48}$$

with $\boldsymbol{d}_{\mathcal{L}} = \begin{bmatrix} d_{x\mathcal{L}} & d_{y\mathcal{L}} & d_{z\mathcal{L}} \end{bmatrix}^T$ denoting the disturbance vector and $\boldsymbol{a}_z = [0\ 0\ 1]^T$.

A.3.3 State-space representation

After obtained the inertia matrix, $\boldsymbol{M}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}})$, the Coriolis and centripetal forces matrix, $\boldsymbol{C}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}, \dot{\boldsymbol{q}}_{\mathcal{L}})$, the gravitational force vector, $\boldsymbol{G}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}})$, and the generalized torque and force vector, $\boldsymbol{\vartheta}_{\mathcal{L}}$, it is possible to rewrite the model in a state-space representation, yielding to

$$\dot{\boldsymbol{x}}_{\mathcal{L}} = \boldsymbol{\varphi}_{\mathcal{L}}(\boldsymbol{x}_{\mathcal{L}}, \boldsymbol{u}, \boldsymbol{d}) = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{L}} \\ \ddot{\boldsymbol{q}}_{\mathcal{L}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}}_{\mathcal{L}} \\ \boldsymbol{M}_{\mathcal{L}}^{-1}\left[\boldsymbol{\vartheta}_{\mathcal{L}} - \boldsymbol{C}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}}, \dot{\boldsymbol{q}}_{\mathcal{L}})\,\dot{\boldsymbol{q}}_{\mathcal{L}} - \boldsymbol{G}_{\mathcal{L}}(\boldsymbol{q}_{\mathcal{L}})\right] \end{bmatrix}. \tag{A.49}$$

The model derived in this section is used to design the controllers of Chapter 3.

A.4 Model parameters

Table A.1: Model parameters of the Tilt-rotor UAV with suspended load.

Parameter	Value
$m_{\mathcal{L}} = m_4$	0.09000 [Kg]
m_1	1.70249 [Kg]
m_2, m_3	0.13973 [Kg]
l	1 [m]
$\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_4} = \boldsymbol{d}^{\mathcal{L}}_{\mathcal{B}}$	$[0\ 0\ l]^T$ [m]
$\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_1}$	$[-0.00433\ \ 0.00060\ \ -0.04559]^T$ [m]
$\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_2}$	$[0.00002\ \ -0.27761\ \ 0.05493]^T$ [m]
$\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_3}$	$[0.00077\ \ 0.27761\ \ 0.05493]^T$ [m]
$\boldsymbol{I}_4 = \boldsymbol{I}_{\mathcal{L}}$	$2.645 \cdot 10^{-6} \cdot \mathbb{I}_{3\times 3}$ [Kg·m^2]
$\boldsymbol{I}_1$	$\begin{bmatrix} 3697.66749 & 0.36342 & -9.51029 \\ * & 840.10403 & 0.61804 \\ * & * & 3865.05354 \end{bmatrix} \cdot 10^{-6}$ [Kg·m^2]
$\boldsymbol{I}_2$	$\begin{bmatrix} 441.68245 & 0 & 0 \\ * & 441.67985 & -1.07006 \\ * & * & 0.64418 \end{bmatrix} \cdot 10^{-6}$ [Kg·m^2]
$\boldsymbol{I}_3$	$\begin{bmatrix} 441.68245 & 0 & 0 \\ * & 441.67985 & 1.07006 \\ * & * & 0.64418 \end{bmatrix} \cdot 10^{-6}$ [Kg·m^2]
$\boldsymbol{g}$	$[0\ 0\ -9.81]^T$ [m/s^2]
k_τ	$1.7 \cdot 10^{-7}$ [N·m·s^2]
b	$9.5 \cdot 10^{-6}$ [N·s^2]
β	5 [°]
μ_γ	0.005 [N·m/(rad/s)]

The model physical parameters of the Tilt-rotor UAV with suspended load considered in this work are shown in Table A.1. Mass, inertia and displacement parameters of the aircraft were obtained from CAD model, designed in Solidworks® software. The parameters

related to the suspended load, as well as k_τ and b, are the same considered in Almeida & Raffo (2015). Moreover, the gravitational acceleration is assumed constant.

A.5 Considerations about the embedded system

The description of the UAV's embedded system and its peripherals are beyond the scope of this work. Those readers interesting to know more details about the sensors and actuators, or how the controllers' sampling time are chosen regarding their limitations, should refer to the work of Andrade (2016) where a detailed description of the Tilt-rotor UAV is provided. For the sake of simplicity in this work the control problem is solved through state-feedback structure with all states assumed to be known either by measurement or estimation.

B

Nonlinear Cascade Control Strategy

This appendix presents a nonlinear two-level cascade control framework able to solve the path tracking problem of a Tilt-rotor UAV while transporting a suspended load stably. The inner-loop is designed in order to control the attitude and altitude while stabilizing the thrusters' group tilting angles actuating on the UAV's propellers and servos. Besides, the outer-loop controls the vehicle's planar position while stabilizing the load angles calculating the desired roll and pitch angles to be set as external reference for the inner-loop.[1]

In the upcoming sections some assumptions about the system are made seeking to decouple its dynamics for control purposes. Also, regarding the inner-loop, a nonlinear controller is designed using the input-output feedback linearization (IOFL) method with dynamic extension together with a mixed discrete $\mathcal{H}_2/\mathcal{H}_\infty$ with pole placement constraints controller to deal with the linearized system resulting from the diffeomorphism obtained after the IOFL techniques was applied. The outer-loop control problem is not addressed in this appendix since the RMPC and REMPC controllers presented in Chapter 4 were designed for such end.

The nonlinear cascade control strategy developed here was mainly based on the work of Raffo & Almeida (2017), been only modified to include the mixed discrete $\mathcal{H}_2/\mathcal{H}_\infty$ with pole placement constraints controller formulated in Rego (2016) and used to control a Tilt-rotor with tail surfaces in Santos et al. (2017a). Additionally, the readers should refer to Slotine & Li (1991) and Khalil (2001) for further information about the IOFL method

[1] Further information about the Tilt-rotor UAV considered in this appendix can be seen at section A.2 from Appendix A, where the modeling process to describe the load transportation problem from the UAV's perspective is presented.

and dynamic extension technique.

B.1 Decoupled system

Consider the model defined in the equation (A.22) representing the dynamics of the Tilt-rotor UAV with suspended load from the UAV's perspective.[2] In order to decouple the model (A.22) into two independent dynamic models, some assumptions regarding the system's equation of motion needs to be made (Raffo & Almeida, 2017). Specially, some degrees of freedom that have their dynamics coupled are assumed uncoupled making the follow hypotheses about some of the inertia matrix's entries (see equation (A.23)):

1. Despite the physical parameters presented at Table A.1, for control purposes, the center of mass of the main body frame $\mathcal{C}_1$ is assumed to be exactly placed at its own geometric center frame $\mathcal{B}$, i.e., $d^{\mathcal{B}}_{\mathcal{C}_1} = \mathbf{0}$; also, the rotors are assumed to have the same mass and to be aligned with the frame $\mathcal{B}$ along the $X^{\mathcal{B}}$-axis and $Z^{\mathcal{B}}$-axis being equally distant to the main body frame origin regarding the $Y^{\mathcal{B}}$-axis, i.e., $m_2 = m_3$ and $\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_2} = \begin{bmatrix} 0 & d^{\mathcal{B}}_{\mathcal{C}_{2y}} & 0 \end{bmatrix}^T = -\boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_3}$; leading to $\boldsymbol{H} = \boldsymbol{S}\left(m_4 \boldsymbol{d}^{\mathcal{B}}_{\mathcal{C}_4}\right)$. Additionally, assuming $m_4 << m$, the inertia matrix's entry $\boldsymbol{m}_{12} = \boldsymbol{R}^{\mathcal{I}}_{\mathcal{B}} \boldsymbol{H} \boldsymbol{W}_{\eta} \approx \mathbf{0}$. Therefore, the coupling between the attitude and position dynamics is assumed to be negligible.

2. The coupling between the altitude and load angles dynamics is assumed to be negligible, i.e., $\boldsymbol{m}_{15} \approx \begin{bmatrix} \vec{\boldsymbol{i}} & \vec{\boldsymbol{j}} & \mathbf{0} \end{bmatrix} \boldsymbol{m}_{15}$.

3. The coupling between the attitude and tilting angles dynamics is assumed to be negligible, i.e., $\boldsymbol{m}_{23} \approx \mathbf{0}$ and $\boldsymbol{m}_{24} \approx \mathbf{0}$.

4. The coupling between the attitude dynamics and the load motion is also assumed to be negligible, i.e., $\boldsymbol{m}_{25} \approx \mathbf{0}$.

Thereafter, equation (A.22) can be split into two new uncoupled models having the generalized coordinates $\boldsymbol{q}_1 = \begin{bmatrix} z & \boldsymbol{\eta}^T & \boldsymbol{\alpha}^T \end{bmatrix}^T$ and $\boldsymbol{q}_2 = \begin{bmatrix} x & y & \boldsymbol{\gamma}^T \end{bmatrix}^T$, yielding to

$$\boldsymbol{M}_{\boldsymbol{q}_1}(\boldsymbol{q}_1)\,\ddot{\boldsymbol{q}}_1 + \boldsymbol{C}_{\boldsymbol{q}_1}(\boldsymbol{q}_1, \dot{\boldsymbol{q}}_1)\,\dot{\boldsymbol{q}}_1 + \boldsymbol{G}_{\boldsymbol{q}_1}(\boldsymbol{q}_1) = \boldsymbol{\vartheta}_{\boldsymbol{q}_1}(\boldsymbol{q}_1, \dot{\boldsymbol{q}}_1), \tag{B.1}$$

$$\boldsymbol{M}_{\boldsymbol{q}_2}(\boldsymbol{q}_2)\,\ddot{\boldsymbol{q}}_2 + \boldsymbol{C}_{\boldsymbol{q}_2}(\boldsymbol{q}_2, \dot{\boldsymbol{q}}_2)\,\dot{\boldsymbol{q}}_2 + \boldsymbol{G}_{\boldsymbol{q}_2}(\boldsymbol{q}_2) = \boldsymbol{\vartheta}_{\boldsymbol{q}_2}(\boldsymbol{q}_2, \dot{\boldsymbol{q}}_2), \tag{B.2}$$

where $\boldsymbol{M}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i} \boldsymbol{M} \boldsymbol{I}^T_{\boldsymbol{q}_i}$, $\boldsymbol{C}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i} \boldsymbol{C} \boldsymbol{I}^T_{\boldsymbol{q}_i}$, $\boldsymbol{G}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i} \boldsymbol{G}$, and $\boldsymbol{\vartheta}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i} \boldsymbol{\vartheta}$ with

$$\boldsymbol{I}_{\boldsymbol{q}_1} = \begin{bmatrix} \mathbf{0}_{\mathbf{6\times 2}} & \mathbb{I}_{6\times 6} & \mathbf{0}_{\mathbf{6\times 2}} \end{bmatrix}, \qquad \boldsymbol{I}_{\boldsymbol{q}_2} = \begin{bmatrix} \mathbb{I}_{2\times 2} & \mathbf{0}_{\mathbf{2\times 6}} & \mathbf{0}_{\mathbf{2\times 2}} \\ \mathbf{0}_{\mathbf{2\times 2}} & \mathbf{0}_{\mathbf{2\times 6}} & \mathbb{I}_{2\times 2} \end{bmatrix}.$$

[2] The subscript $\mathcal{B}$ used to mark the system's equation of motion as derived from the UAV's perspective at section A.2 will be dropped since this appendix considers only this perspective.

Considering equation (A.26) and equations (A.27) to (A.29), $\boldsymbol{\vartheta}_{\boldsymbol{q}_i}$ can be expressed as

$$\boldsymbol{\vartheta}_{\boldsymbol{q}_i} = \boldsymbol{B}_{\boldsymbol{q}_i}\boldsymbol{u} - \boldsymbol{\mu}_{\boldsymbol{q}_i}\dot{\boldsymbol{q}}_i + \boldsymbol{\delta}_{\boldsymbol{q}_i}, \tag{B.3}$$

where $\boldsymbol{B}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{B}$, $\boldsymbol{\mu}_{\boldsymbol{q}_i} = \boldsymbol{I}_{\boldsymbol{q}_i}\boldsymbol{\mu}\boldsymbol{I}_{\boldsymbol{q}_i}^T$, and $\boldsymbol{\delta}_{\boldsymbol{q}_i}$ is a vector containing the unmodeled dynamics from the decoupled procedure and external disturbances $\boldsymbol{\vartheta}_{\mathrm{d}}$.

Using the expression (B.3), the uncoupled models (B.1) and (B.2) can be rewritten in the state-space representation as

$$\begin{aligned} \dot{\boldsymbol{x}}_{\boldsymbol{q}_i} &= \begin{bmatrix} \dot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{q}}_i \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{q}}_i \\ \boldsymbol{M}_{\boldsymbol{q}_i}^{-1}\left[\boldsymbol{B}_{\boldsymbol{q}_i}\boldsymbol{u} + \boldsymbol{\delta}_{\boldsymbol{q}_i} - \left[\boldsymbol{C}_{\boldsymbol{q}_i}\left(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i\right) + \boldsymbol{\mu}_{\boldsymbol{q}_i}\right]\dot{\boldsymbol{q}}_i - \boldsymbol{G}_{\boldsymbol{q}_i}\left(\boldsymbol{q}_i\right)\right] \end{bmatrix} \\ &= \boldsymbol{f}_{\boldsymbol{q}_i}(\boldsymbol{x}_{\boldsymbol{q}_i}) + \boldsymbol{g}_{\boldsymbol{u}_{\boldsymbol{q}_i}}\boldsymbol{u} + \boldsymbol{g}_{\boldsymbol{d}_{\boldsymbol{q}_i}}\boldsymbol{\delta}_{\boldsymbol{q}_i}. \end{aligned} \tag{B.4}$$

B.2 Inner-loop control

This section deals with the design of a nonlinear controller able to control the dynamics of the generalized coordinate $\boldsymbol{q}_1$ described by the model (B.1). First, the IOFL method with dynamic extension technique is used to obtain a linear system through a diffeomorphism. Hereafter, a robust discrete $\mathcal{H}_2/\mathcal{H}_\infty$ controller is designed to control the resulting linear system while dealing with the unmodeled dynamics present in the decoupled system. Instead of the continuous $\mathcal{H}_2/\mathcal{H}_\infty$ controller used in Raffo & Almeida (2017), this work uses its discrete version to avoid uncertainties coming from the discretization process since the controllers presented here will be implemented in digital computers.

B.2.1 IOFL with dynamic extension

Let $\boldsymbol{h}_{\boldsymbol{q}_1} = \begin{bmatrix} z & \phi & \theta & \psi \end{bmatrix}^T$ be the outputs and $\boldsymbol{u} = \begin{bmatrix} f_R & f_L & \tau_{\alpha_R} & \tau_{\alpha_L} \end{bmatrix}^T$ be the inputs of the system (B.1). Therefore, considering equation (B.4), a nonlinear system affine in the inputs can be written as[3]

$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{g}_{\boldsymbol{d}}(\boldsymbol{x})\boldsymbol{\delta}, \\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}). \end{aligned} \tag{B.5}$$

The concepts of Lie derivative and relative degree of a system are essential to apply the IOFL techniques (Slotine & Li, 1991). Recalling, the Lie Derivative of a scalar field $\boldsymbol{h}(\boldsymbol{x})$ in the direction of a vector field $\boldsymbol{f}(\boldsymbol{x})$ results in a third scalar field given by

$$\mathcal{L}_{\boldsymbol{f}}\boldsymbol{h}(\boldsymbol{x}) = \nabla\boldsymbol{h}(\boldsymbol{x}) \cdot \boldsymbol{f}(\boldsymbol{x}), \tag{B.6}$$

[3]Since this section address only the inner-loop control problem, to simplify the notation, in the remaining of the section the subscript $\boldsymbol{q}_1$ will be dropped and $\boldsymbol{q}$ will be used to denotes $\boldsymbol{q}_1$.

where $\nabla \boldsymbol{h}(\boldsymbol{x}) = \left[\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial x_1} \quad \cdots \quad \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial x_n}\right]$ is the gradient of $\boldsymbol{h}(\boldsymbol{x})$. Further, the relative degree r_i to an output y_i is defined as the number of times the output needs to be differentiated until some input appears in the resulting expression. Thus, the system relative degree is given by the sum of all outputs' relative degree $r = \sum r_i$, which can be obtained through the expression

$$r_i = \inf\left\{k : \exists j, \mathcal{L}_{\boldsymbol{g}_j}\mathcal{L}_{\boldsymbol{f}}^{k-1}\boldsymbol{h}_i(\boldsymbol{x}) \neq 0\right\}, \tag{B.7}$$

where $\boldsymbol{g}_j$ is the j-th column of $\boldsymbol{g}_{\boldsymbol{u}}(\boldsymbol{x})$.

The output differentiating procedure used to compute the relative degree can be generally expressed as (Slotine & Li, 1991)

$$y_i^{(r_i)} = \mathcal{L}_{\boldsymbol{f}}^{r_i}\boldsymbol{h}_i(\boldsymbol{x}) + \sum_{j=1}^{m}\mathcal{L}_{\boldsymbol{g}_j}\mathcal{L}_{\boldsymbol{f}}^{r_i-1}\boldsymbol{h}_i(\boldsymbol{x})\boldsymbol{u}_j, \tag{B.8}$$

with $\mathcal{L}_{\boldsymbol{g}_j}\mathcal{L}_{\boldsymbol{f}}^{r_i-1}\boldsymbol{h}_i(\boldsymbol{x}) \neq 0$ for at least one j when performing the r_i-th differentiation and $\boldsymbol{u}_j$ being the j-th system's input.

In order to have a system with a set of input/outputs fully feedback linearizable, its relative degree should be equal to the number of state variables to avoid the presence of internal dynamics in the transformed system. However, as shown in Raffo & Almeida (2017), the relative degree of the system (B.1) is $r = 8$ while $\boldsymbol{x} \in \mathbb{R}^{12}$. To overcome this issue, the authors propose to use the dynamic extension technique (Slotine & Li, 1991) augmenting the state vector to $\bar{\boldsymbol{x}} = \boldsymbol{h}_a(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix}\boldsymbol{q}^T & \dot{\boldsymbol{q}}^T & f_R & \dot{f}_R & f_L & \dot{f}_L\end{bmatrix}^T$ and redefining the input vector as $\bar{\boldsymbol{u}} = \begin{bmatrix}\ddot{f}_R & \ddot{f}_L & \tau_{\alpha_R} & \tau_{\alpha_L}\end{bmatrix}^T$, which gives the new state-space equation

$$\begin{aligned}\dot{\bar{\boldsymbol{x}}} &= \bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}) + \textstyle\sum_{i=1}^{4}\bar{\boldsymbol{g}}_i(\bar{\boldsymbol{x}})\bar{\boldsymbol{u}}_i + \bar{\boldsymbol{g}}_{\boldsymbol{d}}(\bar{\boldsymbol{x}})\boldsymbol{\delta},\\ \boldsymbol{y} &= \boldsymbol{h}(\boldsymbol{x}),\end{aligned} \tag{B.9}$$

where $\bar{\boldsymbol{f}}(\bar{\boldsymbol{x}}) = \begin{bmatrix}\dot{\boldsymbol{q}}^T & \ddot{\boldsymbol{q}}^T & \dot{f}_R & 0 & \dot{f}_L & 0\end{bmatrix}^T$, $\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}\left(-\left[\boldsymbol{C} + \boldsymbol{\mu}\right]\dot{\boldsymbol{q}} - \boldsymbol{G} + \boldsymbol{\delta}\right) + \boldsymbol{B}_1 f_R + \boldsymbol{B}_2 f_L$, $\bar{\boldsymbol{u}}_i$ is the i-th input, $\boldsymbol{B}_i$ is the i-th column of $\boldsymbol{B}_{\boldsymbol{q}}$, and $\bar{\boldsymbol{g}}_i(\bar{\boldsymbol{x}})$ is given by

$$\bar{\boldsymbol{g}}_1 = \begin{bmatrix}\mathbf{0_{13\times 1}}\\ 1\\ \mathbf{0_{2\times 1}}\end{bmatrix}, \bar{\boldsymbol{g}}_2 = \begin{bmatrix}\mathbf{0_{15\times 1}}\\ 1\end{bmatrix}, \bar{\boldsymbol{g}}_3 = \begin{bmatrix}\mathbf{0_{6\times 1}}\\ \boldsymbol{B}_3\\ \mathbf{0_{4\times 1}}\end{bmatrix}, \bar{\boldsymbol{g}}_4 = \begin{bmatrix}\mathbf{0_{6\times 1}}\\ \boldsymbol{B}_4\\ \mathbf{0_{4\times 1}}\end{bmatrix}. \tag{B.10}$$

As shown in Raffo & Almeida (2017), after applying the dynamic extension technique the system's relative degree becomes $r = 16$, hence it is now fully feedback linearizable.

In order to obtain the transformed linearized states $\boldsymbol{\Theta}(\boldsymbol{x})$, in which $\boldsymbol{\Theta}(\cdot)$ defines a local diffeomorphism, the follow input transformation must be considered

$$\bar{\boldsymbol{u}} = \boldsymbol{\Delta}^{-1}(\boldsymbol{v} - \boldsymbol{b}) \tag{B.11}$$

with $\boldsymbol{v} = \begin{bmatrix} v_z & v_\phi & v_\theta & v_\psi \end{bmatrix}^T$ being the additional control inputs. Using the equation (B.8) with the augmented state vector $\bar{\boldsymbol{x}}$ and the input $\bar{\boldsymbol{u}}$, it is possible to define

$$\begin{aligned} \boldsymbol{\Delta}(\bar{\boldsymbol{x}}) \in \mathbb{R}^{4\times 4} \quad &: \quad \boldsymbol{\Delta}_{ij}(\bar{\boldsymbol{x}}) = \mathcal{L}_{\bar{\boldsymbol{g}}_j} \mathcal{L}^3_{\bar{\boldsymbol{f}}} \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}\right) \\ \boldsymbol{b}(\bar{\boldsymbol{x}}) \in \mathbb{R}^{4\times 1} \quad &: \quad \boldsymbol{b}_i(\bar{\boldsymbol{x}}) = \mathcal{L}^4_{\bar{\boldsymbol{f}}} \boldsymbol{h}_i\left(\bar{\boldsymbol{x}}\right), \end{aligned} \tag{B.12}$$

where $\boldsymbol{\Delta}_{ij}(\bar{\boldsymbol{x}})$ is the (i,j)-th entry of the matrix $\boldsymbol{\Delta}(\bar{\boldsymbol{x}})$ and $\boldsymbol{b}_i(\bar{\boldsymbol{x}})$ is the i-th row of $\boldsymbol{b}(\bar{\boldsymbol{x}})$. Therefore, the system's outputs become a simple linear relation yielding to

$$\boldsymbol{y}^{(r)} = \ddddot{\boldsymbol{h}}\left(\bar{\boldsymbol{x}}\right) = \boldsymbol{v} + \boldsymbol{\pi}(\boldsymbol{\delta}), \tag{B.13}$$

with $\boldsymbol{\pi}(\boldsymbol{\delta})$ being a term containing unmodeled dynamics and unknown external disturbances.

Thereafter, a Proportional-Integral-Derivative (PID)-like controller with feed-forward term is proposed in order to regulate the outputs z, ϕ, θ, and ψ (Raffo & Almeida, 2017). Then, the additional linear control laws can be designed as

$$v_i = \dddot{i}^{tr} + K_{ddd_i}\dddot{e}_i + K_{dd_i}\ddot{e}_i + K_{d_i}\dot{e}_i + K_{p_i}e_i + K_{i_i}\int e_i dt, \tag{B.14}$$

where $e_i = i - i^{tr}$ with $i \in \{z, \phi, \theta, \psi\}$ and $(\cdot)^{tr}$ denotes the desired trajectory.

Defining the vector $\tilde{\boldsymbol{e}} = \begin{bmatrix} \int \boldsymbol{e} dt & \boldsymbol{e} & \dot{\boldsymbol{e}} & \ddot{\boldsymbol{e}} & \dddot{\boldsymbol{e}} \end{bmatrix}^T$ with $\boldsymbol{e} = \begin{bmatrix} e_z & e_\phi & e_\theta & e_\psi \end{bmatrix}^T$, and $\boldsymbol{K} = \textbf{blkdiag}\left(\boldsymbol{K}_i, \boldsymbol{K}_p, \boldsymbol{K}_d, \boldsymbol{K}_{dd}, \boldsymbol{K}_{ddd}\right)$, the linearized dynamics can be written as follows

$$\begin{cases} \dot{\tilde{\boldsymbol{e}}} = \boldsymbol{A}\tilde{\boldsymbol{e}} + \boldsymbol{B}_{\boldsymbol{u}}\tilde{\boldsymbol{u}} + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{\pi}, \\ \tilde{\boldsymbol{u}} = \boldsymbol{K}\tilde{\boldsymbol{e}}, \end{cases} \tag{B.15}$$

where $\boldsymbol{A}$, $\boldsymbol{B}_{\boldsymbol{u}}$, and $\boldsymbol{B}_{\boldsymbol{\pi}}$ are matrices with appropriated dimension.

B.2.2 Discrete mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control

In order to design a discrete mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control to compute the feedback gain $\boldsymbol{K}$, the model (B.15) can be discretized through a zero-order hold with sample time T_s, yielding to

$$\begin{aligned} \tilde{\boldsymbol{e}}_{k+1} &= \boldsymbol{A}\tilde{\boldsymbol{e}}_k + \boldsymbol{B}_{\boldsymbol{u}}\tilde{\boldsymbol{u}}_k + \boldsymbol{B}_{\boldsymbol{\pi}}\boldsymbol{\pi}_k, \\ \tilde{\boldsymbol{u}}_k &= \boldsymbol{K}\tilde{\boldsymbol{e}}_k. \end{aligned} \tag{B.16}$$

Defining the cost variables

$$\tilde{\boldsymbol{z}}_k = \boldsymbol{H}\tilde{\boldsymbol{e}}_k + \boldsymbol{D}_{\boldsymbol{u}}\tilde{\boldsymbol{u}}_k, \tag{B.17}$$

$$\boldsymbol{z}_k = \boldsymbol{H}\tilde{\boldsymbol{e}}_k + \boldsymbol{D}_{\boldsymbol{u}}\tilde{\boldsymbol{u}}_k + \boldsymbol{D}_{\boldsymbol{\pi}}\boldsymbol{\pi}_k, \tag{B.18}$$

where $\boldsymbol{H}$, $\boldsymbol{D}_{\boldsymbol{u}}$, and $\boldsymbol{D}_{\boldsymbol{\pi}}$ are weighting matrices.

Let $\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)$ denotes the discrete-time transfer function from $\boldsymbol{\pi}$ to $\tilde{\boldsymbol{z}}$, with $\varsigma \in \mathbb{C}$. Then,

the $\mathcal{H}_2$ norm of the system (B.16) considering the cost variable (B.17) is defined as $\|\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)\|_2 = \sqrt{\sum_{k=0}^{\infty} \text{trace}\{\boldsymbol{\psi}_k^T \boldsymbol{\psi}_k\}} < \sqrt{\sigma}$, with $\boldsymbol{\psi}_k = \mathcal{Z}^{-1}\{\boldsymbol{\Psi}_{\boldsymbol{\pi}\tilde{\boldsymbol{z}}}(\varsigma)\}$ and $\sigma \in \mathbb{R}$. Further, the $\mathcal{H}_\infty$ norm of the system (B.16) considering the cost variable (B.18) is defined by $\|\boldsymbol{\Psi}_{\boldsymbol{\pi}\boldsymbol{z}}(\varsigma)\|_\infty = \sup_{\boldsymbol{\pi}\neq 0} \|\boldsymbol{\pi}_k\|_2 / \|\boldsymbol{z}_k\|_2 < \sqrt{\varrho}$, with $\varrho \in \mathbb{R}$.

Given the state-feedback control law $\tilde{\boldsymbol{u}}_k = \boldsymbol{K}\tilde{\boldsymbol{e}}_k$, the gain matrix $\boldsymbol{K}$ that minimizes an upper-bound for the $\mathcal{H}_2$ norm while guaranteeing a prescribed upper-bound $\sqrt{\tilde{\varrho}} > \sqrt{\varrho}$ for the $\mathcal{H}_\infty$ norm is obtained by solving the optimization problem

$$\min_{\boldsymbol{P},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{N}} \text{tr}\{\boldsymbol{N}\} \quad \text{subject to (B.19), (B.20), (B.21)}$$

$$\begin{bmatrix} \boldsymbol{N} & \boldsymbol{H}\boldsymbol{X} + \boldsymbol{D}_{\boldsymbol{u}}\boldsymbol{Y} \\ * & \boldsymbol{X} + \boldsymbol{X}^T - \boldsymbol{P} \end{bmatrix} > 0, \tag{B.19}$$

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{Y} & \boldsymbol{B}_{\boldsymbol{\pi}} \\ * & \boldsymbol{X} + \boldsymbol{X}^T - \boldsymbol{P} & \boldsymbol{0} \\ * & * & \mathbb{I} \end{bmatrix} > 0, \tag{B.20}$$

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{A}\boldsymbol{X} + \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{Y} & \boldsymbol{B}_{\boldsymbol{\pi}} & \boldsymbol{0} \\ * & \boldsymbol{X} + \boldsymbol{X}^T - \boldsymbol{P} & \boldsymbol{0} & \boldsymbol{X}^T\boldsymbol{H}^T + \boldsymbol{Y}^T\boldsymbol{D}_{\boldsymbol{u}}^T \\ * & * & \mathbb{I} & \boldsymbol{D}_{\boldsymbol{\pi}}^T \\ * & * & * & \tilde{\varrho}\mathbb{I} \end{bmatrix} > 0, \tag{B.21}$$

where $\boldsymbol{K} = -\boldsymbol{Y}\boldsymbol{X}^{-1}$ and $*$ denote the elements that are deduced by symmetry.

Furthermore, the system's time response can be improved by means of pole placement constraints using LMIs regions to restrict the complex plane. Thus, let $\mathbb{D}_1 = \text{Re}(\text{eig}\{\tilde{\boldsymbol{A}}\}) > \varepsilon$ and $\mathbb{D}_2 = |\text{eig}\{\tilde{\boldsymbol{A}}\}| < \varpi$ denote subsets of the complex plane, where $\tilde{\boldsymbol{A}} = \boldsymbol{A} - \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{K}$, and $\{\varepsilon, \varpi\} \in \mathbb{R}^+$. Therefore, $\text{eig}\{\tilde{\boldsymbol{A}}\} \subset \mathbb{D}_1 \cap \mathbb{D}_2$ if $\exists\, \boldsymbol{T} > 0$ and $\boldsymbol{Y}$ such that

$$\boldsymbol{T}\boldsymbol{A}^T + \boldsymbol{A}\boldsymbol{T} + \boldsymbol{Y}^T\boldsymbol{B}_{\boldsymbol{u}}^T + \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{Y} - 2\varepsilon\boldsymbol{T} > 0, \tag{B.22}$$

$$\begin{bmatrix} -\varpi\boldsymbol{T} & \boldsymbol{A}\boldsymbol{T} + \boldsymbol{B}_{\boldsymbol{u}}\boldsymbol{Y} \\ * & -\varpi\boldsymbol{T} \end{bmatrix} < 0, \tag{B.23}$$

where $\boldsymbol{K} = -\boldsymbol{Y}\boldsymbol{T}^{-1}$. The constraints (B.22) and (B.23) can be merged into the $\mathcal{H}_2/\mathcal{H}_\infty$ control problem by making $\boldsymbol{X} = \boldsymbol{X}^T = \boldsymbol{T} > 0$.

Those whom are interested to see the complete derivation of the LMIs (B.19), (B.20), (B.21), (B.22), and (B.23) that compose the $\mathcal{H}_2/\mathcal{H}_\infty$ control problem with pole placement constraints should refer to Rego (2016).

B.3 Outer-loop

This section deal with the manipulation of the model (B.2) allowing the design of a system able to control the generalized coordinates $\boldsymbol{q}_2$ actuating on the angles ϕ and θ. As proposed

in Raffo & Almeida (2017), an intermediary control input variable is defined for such end.

From (A.27) and regarding the definition $\boldsymbol{B}_{q_2} = \boldsymbol{B}\boldsymbol{I}_{q_2}$, the effect of the generalized force vector generated by the actuators on the subsystem (B.2) can be written as

$$\boldsymbol{B}_{q_2}\boldsymbol{u} = \begin{bmatrix} \boldsymbol{T}_x^{\mathcal{I}} & \boldsymbol{T}_y^{\mathcal{I}} & \mathbf{0} & \mathbf{0} \end{bmatrix}^T, \tag{B.24}$$

with $\boldsymbol{T}_x^{\mathcal{I}}$ and $\boldsymbol{T}_y^{\mathcal{I}}$ being, respectively, translational forces along $\mathrm{X}^{\mathcal{B}}$ and $\mathrm{Y}^{\mathcal{B}}$ axes expressed in the inertial frame.

These translational forces can be represented as

$$\begin{bmatrix} \boldsymbol{T}_x^{\mathcal{I}} \\ \boldsymbol{T}_y^{\mathcal{I}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}} \left(\boldsymbol{r}_R f_R + \boldsymbol{r}_L f_L \right) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix} \left(\boldsymbol{r}_R f_R + \boldsymbol{r}_L f_L \right), \tag{B.25}$$

where r_{ij} is the (i,j)-th entry of the matrix $\boldsymbol{R}_{\mathcal{B}}^{\mathcal{I}}$. The generalized force vector expressed in the body frame $(\boldsymbol{r}_R f_R + \boldsymbol{r}_L f_L)$ can be decomposed into the projections $f_x^{\mathcal{B}}$, $f_y^{\mathcal{B}}$, and $f_z^{\mathcal{B}}$. The drawback idea behind the actuation using the roll (ϕ) and pitch (θ) angles is to change the projection of $f_z^{\mathcal{B}}$ along the $\mathrm{X}^{\mathcal{I}}$ and $\mathrm{Y}^{\mathcal{I}}$ axes of the inertia frame in order to control the planar motion movements and stabilize the load (Raffo & Almeida, 2017). The remaining projections of the generalized force vector are assumed to be known disturbances compensated by the controller.

Considering the relation (B.3) for $\boldsymbol{q}_2$, the model (B.2) can be rewritten as

$$\boldsymbol{M}_{q_2}\ddot{\boldsymbol{q}}_2 + \left[\boldsymbol{C}_{q_2} + \boldsymbol{\mu}_{q_2}\right]\dot{\boldsymbol{q}}_2 + \boldsymbol{G}_{q_2} - \boldsymbol{F}'_{q_2} = \boldsymbol{F}''_{q_2} + \boldsymbol{\delta}_{q_2}, \tag{B.26}$$

where $\boldsymbol{F}'_{q_2} = \begin{bmatrix} r_{11}f_x^{\mathcal{B}} + r_{12}f_y^{\mathcal{B}} & r_{21}f_x^{\mathcal{B}} + r_{22}f_y^{\mathcal{B}} & 0 & 0 \end{bmatrix}^T$ and $\boldsymbol{F}''_{q_2} = \begin{bmatrix} r_{13}f_z^{\mathcal{B}} & r_{23}f_z^{\mathcal{B}} & 0 & 0 \end{bmatrix}^T$. Moreover, defining the auxiliary vector $\boldsymbol{n}_{q_2} = \begin{bmatrix} n_x & n_y & n_{\gamma_1} & n_{\gamma_2} \end{bmatrix}^T$ as

$$\boldsymbol{n}_{q_2} = \left[\boldsymbol{C}_{q_2} + \boldsymbol{\mu}_{q_2}\right]\dot{\boldsymbol{q}}_2 + \boldsymbol{G}_{q_2} - \boldsymbol{F}'_{q_2}, \tag{B.27}$$

the outer loop model can be rewritten as

$$\boldsymbol{M}_{q_2}\ddot{\boldsymbol{q}}_2 + \begin{bmatrix} n_x \\ n_y \\ n_{\gamma_1} \\ n_{\gamma_2} \end{bmatrix} = \begin{bmatrix} \left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right) f_z^{\mathcal{B}} \\ \left(s_\psi s_\theta c_\phi - c_\psi s_\phi\right) f_z^{\mathcal{B}} \\ 0 \\ 0 \end{bmatrix} + \boldsymbol{\delta}_{q_2}. \tag{B.28}$$

Since the nonlinear model (B.28) cannot be easily expressed as affine in control inputs ϕ and θ, intermediary input variables Υ_1 and Υ_2 are defined as

$$\begin{cases} \Upsilon_1 = \left(s_\psi s_\phi + c_\psi s_\theta c_\phi\right) f_z^{\mathcal{B}} - n_x, \\ \Upsilon_2 = \left(s_\psi s_\theta c_\phi - c_\psi s_\phi\right) f_z^{\mathcal{B}} - n_y, \end{cases} \tag{B.29}$$

forming a set of nonlinear equations which solution is given by (Raffo & Almeida, 2017)

$$\begin{cases} \phi &= \arcsin\left[\sigma\left(\frac{s_\psi(\Upsilon_1+n_x)-c_\psi\left(\Upsilon_2+n_y\right)}{f_z^{\mathcal{B}}}\right)\right], \\ \theta &= \arcsin\left[\sigma\left(\frac{c_\psi(\Upsilon_1+n_x)+s_\psi\left(\Upsilon_2+n_y\right)}{f_z^{\mathcal{B}}c_\phi}\right)\right], \end{cases} \tag{B.30}$$

where $\sigma(a)$ is a saturation function given by $\sigma(a) = \min(1, \max(-1, a))$.

Aiming to evaluate the equation (B.30), the projections $f_x^{\mathcal{B}}$, $f_y^{\mathcal{B}}$, and $f_z^{\mathcal{B}}$ are assumed to be known and can be obtained from the inner-loop controller, which also allows to simply obtain the terms n_x and n_y. Moreover, the yaw (ψ) angle is assumed to be measurable.

Replacing the transformation (B.29) into (B.28), the outer loop nonlinear model become

$$\boldsymbol{M}_{q_2}\ddot{\boldsymbol{q}}_2 + \bar{\boldsymbol{n}}_{q_2} = \boldsymbol{\Upsilon}_{q_2} + \boldsymbol{\delta}_{q_2}, \tag{B.31}$$

where $\bar{\boldsymbol{n}}_{q_2} = \begin{bmatrix} 0 & 0 & n_{\gamma_1} & n_{\gamma_2} \end{bmatrix}^T$ and $\boldsymbol{\Upsilon}_{q_2} = \begin{bmatrix} \Upsilon_1 & \Upsilon_2 & 0 & 0 \end{bmatrix}^T$. Therefore, the MPC strategies to control the outer-loop dynamics must be designed for the model (B.31) to obtain the intermediary control inputs $\boldsymbol{\Upsilon}_{q_2}$ and hence be transformed into roll and pitch references for the inner-loop controller through the relations presented at (B.30).

B.4 Cascade control structure

The cascade control structure proposed in this appendix is presented in Figure B.1. The inner-loop controller deals with the generalized coordinates $\boldsymbol{q}_1$ by actuating on the thrusters' group torques and lift forces, moreover the outer-loop controller regulates $\boldsymbol{q}_2$ by actuating on the desired roll and pitch angles, which are reference signals to the inner-loop. The inner controller designed in the previous section works with sampling period T_s while the outer controller algorithms designed in Chapter 4 must be executed with sampling period equal or bigger than $10T_s$. These sampling periods are chosen in order to avoid the external controller to actuates while the inner one still in transient period.

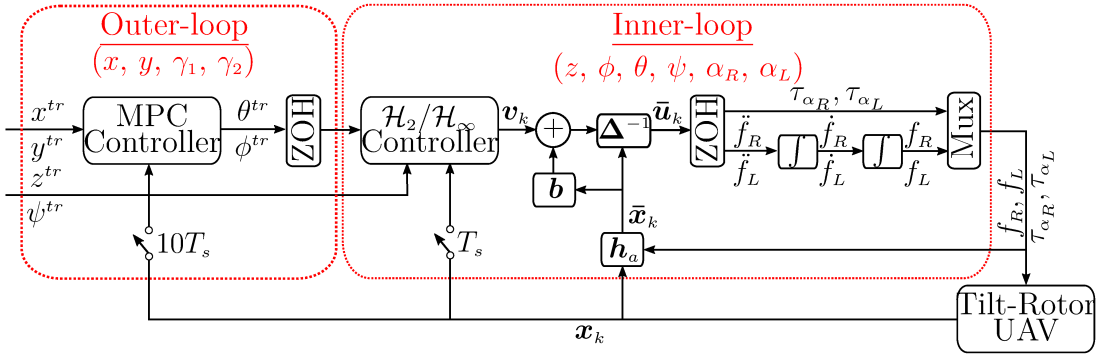


Figure B.1: The proposed control cascade structure.