Advanced modeling and control of complex systems

Original Citation:
Advanced modeling and control of complex systems / Perin, Marco. - ELETTRONICO. - (2026).

Published version
DOI:

Publisher: Politecnico di Bari

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Italian National Ph.D. Program in Autonomous Systems

ACADEMIC DISCIPLINE: SYSTEMS AND CONTROL ENGINEERING (IINF-04/A)

Final Dissertation

Advanced modeling and control of complex systems

by
Marco Perin

Administrative Headquarters:

Politecnico di Bari – Department of Electrical and Information Engineering

Hosting University:

Università degli studi di Padova – Department of Information Engineering

Referees:	Supervisors:
Prof. Marco Frego	Prof. Angelo Cenedese Angelo Cenedese
Prof. Antonino Sferlazza	Dott. Giulia Michieletto Giulia Michieletto

Coordinator of Ph.D Program
Prof. Mariagrazia Dotoli

Course no. 38, 01/11/2022-31/10/2025

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il sottoscritto Perin Marco nato a Bassano del Grappa il 13/11/1998 residente a Colceresa in via Braglio 19 e-mail marco.perin.mr@gmail.com iscritto al 3° anno di Corso di Dottorato di Ricerca in Autonomous systems ciclo 38 ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:
Advanced modeling and control of complex systems

DICHIARA

1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
2) di essere iscritto al Corso di Dottorato di ricerca in Autonomous systems ciclo 38, corso attivato ai sensi del "*Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari*", emanato con D.R. n.286 del 01.07.2013;
3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archivierà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito http://www.creativecommons.it/Licenze), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali od economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Luogo e data Colceresa,17/01/2026 Firma ____________________

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Luogo e data Colceresa, 17/01/2026 Firma ____________________

Marco Perin

Advanced modeling and control of complex systems

Thesis submitted for the degree of Philosophiae Doctor

Italian National Ph.D. Program in Autonomous Systems
University of Padova

Tutors
Prof. Engr. *Angelo Cenedese*
Dott. *Giulia Michieletto*

POLITECNICO DI BARI

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

2025

The doctoral scholarship was funded by the European Union - Next Generation EU, Mission 4 Component 1 Investment 4.1 - Call for tender No. 2152 of Italian Ministry of University and Research; Concession Decree No. 2152 adopted by the Italian Ministry of University and Research, CUP D93C22000850005.

Dissertation submitted for the degree of *Philosophiae Doctor*
Italian National Ph.D. Program in Autonomous Systems

Cycle:
38th

Administrative Headquarters:
Politecnico di Bari

Hosting University:
University of Padova

Title:
Advanced modeling and control of complex systems

Ph.D Candidate:
Marco Perin, University of Padova (Padova, Italy)

Tutors:
Prof. Engr. Angelo Cenedese, University of Padova (Padova, Italy)
Dott. Michieletto Giulia, University of Padova (Padova, Italy)

Coordinator:
Prof. Engr. Mariagrazia Dotoli, Politecnico di Bari (Bari, Italy)

External Reviewers:
Prof. Marco Frego, Free University of Bolzano (Bolzano, Italy)
Prof. Antonino Sferlazza, University of Palermo (Palermo, Italy)

Last version:
Jan 14, 2026

Abstract

The latest engineering developments oriented towards the control of robotic systems encompass a diverse range of strategies and approaches in order to achieve this result. This is particularly valid for systems that show high complexity: from a single, but very complex agent (for example a humanoid robot, a multicopter drone, an industrial process, ...) to a huge number of systems, interconnected in possibly complex ways (an ecosystem, a social network, multiple robots that need coordination in order to successfully accomplish a task). All of these scenarios, together with all the combinations of systems in between, fall into the complex system category.

In this manuscript, we then address the modeling problem by extending what in literature is called the *planar hexarotor*, that is a multirotor drone with the propellers all spinning on the same plane, by tilting these propellers by fixed angles, and studying what is possible to achieve with this change. This is done in two steps, i.e. with two different tilting angles: the first already adds noticeable properties, while the second turns out to be way more interesting from the view of practical utility, turning out to be extremely use-case specific, and making considerations on its employment more delicate.

After the entire modeling of these hexarotors, the control part enters the scene: with a good model of such platforms, is it possible to exploit it in order to create controllers that are tailored for it? An entire chapter of this thesis is dedicated to this problem, showing two novel control schemes.

Later on, the focus moves away from the control of only hexarotors, and shifts towards controllers that can be more general, and can possibly be applied to a wide variety of dynamical systems. This is where Contraction Theory is used, having already proved to be robust in literature. The state of the art about it, though, presents only very few examples of practical applications, due to various reasons: first, the calculations to find controllers that make the system contractive can be very tedious, if not impossible. Secondly, even if calculations are computationally affordable, sometimes the assumptions can turn out to be too restrictive. This is why in this manuscript we build upon methods that do not require integration at run-time, and we extend them with a double contribution first, an adaptive-control layer is added to said controller, and later on the original state-feedback controller is enhanced to an output-feedback controller. The adaptive control layer is added because the controller we build upon, like many others, heavily depend on the model: if an uncertainty that is not negligible arises, various control methods fail to even achieve boundedness of the controlled system. This is greatly reduced by the fact that a contractive system is very likely to provide error bounds, and not diverge, if the uncertainty is bounded. In order to strengthen the convergence even these scenarios though, an adaptive control layer is proved to work remarkably well, combining robustness of the contractive controller with the adaptation, jointly achieving convergence under structured uncertainty. The problem of output feedback instead finds immediate usefulness in the simple fact that most of the time, not all the state is observable, and usually resorting to observers leads to instabilities that mine the overall effectiveness of the controller. We expand the existing controller, showing how it is possible to achieve output-feedback as well, at the cost of higher computational complexity and stricter assumptions.

We're still pioneers, we barely begun.
Our greatest accomplishments cannot be behind us,
cause our destiny lies above us.
Interstellar

Contents

List of Figures

List of Tables

Acronyms

CCM Control Contraction Metric 47

CoM Centre of Mass 4, 5, 11, 22, 67

CT Contraction Theory 2, 43–46

DF Differential Flatness vii, 2, 9, 33, 35, 36, 67

DNN Deep Neural Network 60–62, 67

FC Flatness-based Controller iv, 33, 35, 36, 38–42

Gazebo An open-source 3D robotics simulator commonly used with ROS (Robot Operating System) for testing and validating robotic systems in virtual environments 25

GTMR Generically Tilted MultiRotor 11

HC Hierarchical Controller iv, 33, 36–42

LMI Linear Matrix Inequality 61, 66, 74

MLP Multi Layer Perceptron 62, 66

PINN Physics Informed Neural Network 67, 76

ROS Robot Operating System vi

SGTMR Star shaped Generically Tilted MultiRotor 23

STH Star-shaped Tilted Hexarotor iii, 2, 4, 5, 7, 12, 14, 17, 20, 22, 25, 29, 30, 33–39, 42, 67

STM Star-shaped Tilted Multirotor 4, 5

UAV Unmanned Aerial Vehicle iii, 2, 4, 7, 11, 12, 18, 22, 23, 29, 67

ZM Zero Moment 7, 8

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of *Philosophiae Doctor* in the PhD National Programme in Autonomous System (DAuSy).

During the period for which I was participating in this program, I had the chance to tackle problems that are at the

The first paper published is [1], where the control problem is addressed by considering both classical control strategies (Differential Flatness (DF)) and new specific controllers that work using hierarchical control loops to be able to enforce particular properties in the controlled system.

The same system used passively in [1] was extensively studied in [2], where its maneuverability was studied thoroughly, using rigorous metrics to evaluate its performances and giving guidelines for the choice of the specific platform configuration based on the final task.

After these works, various advancements have been achieved in both the modeling and control aspects of the research. These developments are thoroughly documented and discussed in detail throughout the chapters of this manuscript.

The full list of papers written by the author is reported hereafter.

List of Papers Written by the Author

International Conference Proceedings

[1] Perin, M., Bertoni, M., Michieletto, G., Oboe, R., and Cenedese, A., “Trajectory tracking for tilted hexarotors with concurrent attitude regulation,” in *2024 American Control Conference (ACC)*, ISSN: 2378-5861, Jul. 2024, pp. 1550–1555. DOI: 10.23919/ACC60939.2024.10644887. [Online]. Available: https://ieeexplore.ieee.org/document/10644887.

[2] Perin, M., Bertoni, M., Viezzer, N., Michieletto, G., and Cenedese, A., “Star-shaped tilted hexarotor maneuverability: Analysis of the role of the tilt cant angles,” in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 1791–1797. DOI: 10.1109/CASE59546.2024.10711709.

Introduction

Chapter 1

Introduction

1.1 Thesis structure

This thesis is organized into two parts, each comprising two principal sections that address distinct yet interconnected aspects of my PhD research.

The first part focuses on the modeling problem, systematically investigating various UAV configurations. It begins by establishing a rigorous mathematical framework for describing the actuation capabilities of multirotor systems. The analysis then characterizes the maneuverability properties of two multirotor architectures, examining how design choices influence achievable motions and force/torque generation capabilities. This investigation provides a comprehensive understanding of their dynamic behavior, forming the theoretical foundation for the control strategies discussed in the subsequent part.

The second part addresses the control problem, focusing on trajectory tracking. Initially, the Star-shaped Tilted Hexarotor (STH) model from Part I serves as the basis for designing two specialized nonlinear control strategies for this system. In the final chapter, we explore more general control frameworks using Contraction Theory (CT). Contributions extend existing state-feedback methodologies originally developed for synchronization. An adaptive control layer enhances convergence robustness against known model uncertainties, while an output-feedback reformulation eliminates the need for full state measurements, improving practical applicability.

1.1.1 Contribution

Part I The majority of the presented results of Part I stem from my personal research efforts. While this work builds upon the foundational contributions found in [3], [4] for the multirotor modeling and the force-moment decoupling, all results obtained in this manuscript represent novel contributions to the existing literature.

This novelty spans across both chapters of the modeling part: the analysis of the α-tilted multirotor configuration in chapter 2, and the investigation of the α-β-tilted architecture in chapter 3. Each of these studies provides original insights into the modeling and characterization of distinct multirotor designs. with experimental validation.

Part II A collection of useful extensions to existing control methodologies in the literature is presented in Part II. In chapter 4, two distinct control approaches are investigated and compared. The first approach leverages Differential Flatness (DF), a well-established technique in control theory. The second approach extends the hierarchical control method from [5] to address the trajectory tracking problem, whereas the original work was limited to tracking fixed poses. This extension represents a meaningful generalization of the existing methodology.

In the final chapter, chapter 5, the contributions focus on advancing the contraction-based controller proposed in [6]. Specifically, two key extensions are developed: first, an adaptive control layer is integrated to improve robustness and convergence speed in the presence of structured model uncertainties; second, the controller is reformulated to operate as an output-feedback system rather than relying solely on state feedback. Both extensions are supported by theoretical proofs and validated through numerical simulations on nonlinear chaotic systems, demonstrating that synchronization and trajectory tracking objectives are successfully achieved.

1.2 Preliminaries

1.2.1 Notation

General notation

$\in$	belongs to
$\forall$	for all
$\rightarrow$	maps to
$\mathrm{conv}(\cdot)$	convex hull operator
$\dot{\mathbf{x}} = \frac{d}{dt}\mathbf{x}$	time derivative of $\mathbf{x}$

Groups and Sets

$\mathbf{I}_n$	n-dimensional identity matrix
$\mathbf{0}_n$	n-dimensional zero matrix
$A \times B$	cartesian product between sets A and B
$\mathbb{R}$	set of real numbers
$\mathbb{R}_{\geq 0}$	set of real non-negative numbers
$\mathbb{R}^n$	set of real n-dimensional vectors
$\mathbb{R}^{n\times m}$	set of real n by m matrices
$\mathbb{S}^n$	n-dimensional sphere
$SO(3)$	Special Orthogonal group: $SO(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\right\}$
$\mathbf{R}_{\{x,y,z\}}(\cdot) \in SO(3)$	canonical rotation matrix around the x, y and z axes

Operations on vectors and matrices

$\mathrm{Im}(M)$	image of M
$\ker(M)$	kernel of M
$\det(M)$	determinant of M
$M^\top$	transpose of M
M^{-1}	inverse of the matrix M
$M^\dagger$	Moore-Penrose pseudoinverse of the matrix M
$M = M^\top \succ 0$	M is positive definite
$M = M^\top \succeq 0$	M is positive semi-definite
$\mathbf{v}_1 \cdot \mathbf{v}_2$	dot product $\mathbf{v}_1 \cdot \mathbf{v}_2 = \mathbf{v}_1^\top \mathbf{v}_2$
$\mathbf{v}_1 \times \mathbf{v}_2$	cross product $\mathbf{v}_1 \times \mathbf{v}_2$
$\mathbf{q}_1 \circ \mathbf{q}_2$	quaternion composition operator of $\mathbf{q}_1$ and $\mathbf{q}_2$
$\lvert x \rvert$	absolute value of x
$[\mathbf{v}]_\times$	skew matrix of the vector $\mathbf{v}$ s.t. $[\mathbf{v}]_\times \mathbf{v}_2 = \mathbf{v}_1 \times \mathbf{v}_2$
$\lVert\mathbf{v}\rVert$	euclidean norm of $\mathbf{v}$, i.e. $\sqrt{\mathbf{v}^\top \mathbf{v}}$
$\lVert\mathbf{v}\rVert_M$	euclidean weighted norm of $\mathbf{v}$ by M, i.e. $\sqrt{\mathbf{v}^\top M \mathbf{v}}$
$\mathcal{L}_\varphi P(\cdot)$	Lie derivative of the tensor P w.r.t the vector field φ

1.2.2 Multirotor model

We refer to the Star-shaped Tilted Multirotor (STM) as a multi-rotor platform actuated by n propellers, each of which is placed around the UAV Centre of Mass (CoM) and is possibly tilted around the axis corresponding to the vehicle arm. In particular, in this work, the attention is restricted to the STHs wherein the tilt angles cannot vary during flight.

We introduce the global inertial reference frame $\mathscr{F}_W = \{O_W, (\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W)\}$ (*world frame*) whose axes directions are identified by the unit vectors $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$ of the canonical basis of $\mathbb{R}^3$ for sake of simplicity, and the local reference frame $\mathscr{F}_B = \{O_B, (\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B)\}$ (*body frame*), in-built with the vehicle and centered in its CoM.

The position and orientation of an STH in the 3D space are thus described by the vector $\mathbf{p} \in \mathbb{R}^3$ denoting the position of O_B in $\mathscr{F}_W$, and by the rotation matrix $\mathbf{R} \in SO(3)$ representing the orientation of $\mathscr{F}_B$ with respect to $\mathscr{F}_W$. Then, each propeller is distributed evenly around the CoM, aligned on the horizontal plane, resulting in the positions $\mathbf{p}_i \in \mathbb{R}^3$ defined in (1.3) for each propeller $i \in \{1 \dots n\}$, in $\mathscr{F}_B$. To fully describe the platform motion, we need to specify both the position and orientation dynamics. The position kinematics is straightforward, relating the time derivative of the position vector $\mathbf{p}$ to the linear velocity $\mathbf{v}$. For the orientation dynamics, we employ the rotation matrix $\mathbf{R} \in SO(3)$, whose time derivative can be expressed through the skew-symmetric matrix associated with the angular velocity vector $\boldsymbol{\omega}$.

The kinematic equations are thus given by:

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{1.1a}$$

$$\dot{\mathbf{R}} = \mathbf{R}\,[\boldsymbol{\omega}]_\times\,, \tag{1.1b}$$

where (1.1a) describes the translational kinematics, while (1.1b) captures the rotational kinematics. In (1.1b), the operator $[\cdot]_\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ maps a vector $\boldsymbol{\omega} \in \mathbb{R}^3$ to its associated skew-symmetric matrix, defined as:

$$[\boldsymbol{\omega}]_\times = \begin{bmatrix} 0 & -\boldsymbol{\omega}_z & \boldsymbol{\omega}_y \\ \boldsymbol{\omega}_z & 0 & -\boldsymbol{\omega}_x \\ -\boldsymbol{\omega}_y & \boldsymbol{\omega}_x & 0 \end{bmatrix}, \tag{1.2}$$

where $\boldsymbol{\omega}_x$, $\boldsymbol{\omega}_y$, and $\boldsymbol{\omega}_z$ denote the components of $\boldsymbol{\omega}$ along the body frame axes. This representation is particularly useful as it allows the cross product $\boldsymbol{\omega} \times \mathbf{v}$ to be written in matrix form as $[\boldsymbol{\omega}]_\times \mathbf{v}$ for any vector $\mathbf{v} \in \mathbb{R}^3$.

Then, in (1.3), $\ell \in \mathbb{R}_{>0}$ is the distance between O_B and the propeller CoM assumed on the $(\mathbf{x}_B, \mathbf{y}_B)$ plane and $\mathbf{R}_z(\cdot) \in SO(3)$ is the canonical rotation matrix around the z axis.

$$\mathbf{p}_i = \ell\,\mathbf{R}_z\left((i-1)\frac{2\pi}{n}\right)\mathbf{e}_1, \tag{1.3}$$

The symmetric disposition around the vertical axis of the UAV, forming an n-gon, is what gives the STM the “Star-shaped” part of its name.

Then, by rotating around its spinning axis $\mathbf{z}_{P_i}$, each i-th propeller generates in its CoM a *thrust force* $\mathbf{f}_i \in \mathbb{R}^3$ and a *drag moment* $\boldsymbol{\tau}_i^d \in \mathbb{R}^3$ having constant direction in $\mathscr{F}_B$ depending on the tilt angle. The spinning axis $\mathbf{z}_{P_i}$ will be vector in which we will explore various configurations, thus it will be discussed later on in the manuscript. Both the thrust force and the drag moment, together with the emerging *thrust moment* $\boldsymbol{\tau}_i^t = \mathbf{p}_i \times \mathbf{f}_i \in \mathbb{R}^3$, depend on the propeller spinning rate $\omega_i \in \mathbb{R}_{\geq 0}$ according to the popular models

$$\mathbf{f}_i = c_{f_i}\omega_i^2\mathbf{z}_{P_i}, \tag{1.4}$$

$$\boldsymbol{\tau}_i^d = \kappa_i c_{\tau_i}\omega_i^2\mathbf{z}_{P_i}, \tag{1.5}$$

$$\boldsymbol{\tau}_i^t = c_{f_i}\omega_i^2(\mathbf{p}_i \times \mathbf{z}_{P_i}) \tag{1.6}$$

where $c_{f_i}, c_{\tau_i} \in \mathbb{R}_{\geq 0}$ are constant parameters depending on the rotor geometric features and $\kappa_i \in \{-1, 1\}$ allows for distinguishing whether the i-th propeller spins counterclockwise (CCW, $\kappa_i = 1$) or clockwise (CW, $\kappa_i = -1$).

Considering all propeller actions regulated through the assignable *control input* $u_i = \omega_i^2 \in \mathbb{R}_{\geq 0}$, the *total control force* $\mathbf{f}_c \in \mathbb{R}^3$ and the *total control moment* $\boldsymbol{\tau}_c \in \mathbb{R}^3$ applied in the platform CoM and expressed in $\mathscr{F}_B$ result to be

$$\begin{aligned} \mathbf{f}_c &= \textstyle\sum_{i=1}^{n} \mathbf{f}_i && (1.7\text{a}) \\ &= \textstyle\sum_{i=1}^{n} c_{f_i} \mathbf{z}_{P_i} u_i, \\ \boldsymbol{\tau}_c &= \textstyle\sum_{i=1}^{n} (\boldsymbol{\tau}_i^t + \boldsymbol{\tau}_i^d) && (1.7\text{b}) \\ &= \textstyle\sum_{i=1}^{n} \left(c_{f_i} \mathbf{p}_i \times \mathbf{z}_{P_i} + \kappa c_{\tau_i} \mathbf{z}_{P_i} \right) u_i. \end{aligned}$$

During all the main dissertation body, the number of propellers n will be 6, therefore instead of STMs we will consider STHs, and $i \in \{1 \ldots 6\}$. Introducing the *control input vector* $\mathbf{u} = \left[u_1 \ \cdots \ u_6\right]^\top \in \mathbb{R}^6$, expressions (1.7a) and (1.7b) can be shortened to become

$$\mathbf{f}_c = \mathbf{F}\mathbf{u} \quad \text{and} \quad \boldsymbol{\tau}_c = \mathbf{M}\mathbf{u}, \tag{1.8}$$

where the *control force input matrix* $\mathbf{F} \in \mathbb{R}^{3 \times 6}$ and the *control moment input matrix* $\mathbf{M} \in \mathbb{R}^{3 \times 6}$ depend on the tilt angle and on the geometric and aerodynamic parameters of the propellers. In the rest of the paper, we account for platforms actuated by a set of rotors with equal actuation and aerodynamic characteristics, as well as a balanced choice of CW/CCW spinning directions. Specifically, for $i \in \{1 \ldots 6\}$, we assume that $u_i \in \bar{\mathcal{U}} = [0, \bar{u}]$ with $\bar{u} \in \mathbb{R}_{\geq 0}$, $c_{f_i} = c_f$, $c_{\tau_i} = c_\tau$ with $c_f > c_\tau$, and $\kappa_i = (-1)^i$.

Adopting the Euler-Newton approach, the dynamics of the platform can be described through the equations in (1.9):

$$\begin{aligned} m\ddot{\mathbf{p}} &= -mg\mathbf{e}_3 + \mathbf{R}\mathbf{f}_c(\alpha) = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{F}_\alpha \mathbf{u}, && (1.9\text{a}) \\ \mathbf{J}\dot{\boldsymbol{\omega}} &= -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c(\alpha) = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}_\alpha \mathbf{u}, && (1.9\text{b}) \end{aligned}$$

where $g \in \mathbb{R}_{>0}$ and $m \in \mathbb{R}_{>0}$ are the gravity constant and the total platform mass, while $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ is the positive definite inertia matrix of the vehicle computed in $\mathscr{F}_B$.

In the following sections, we present the complete dynamic model of the platform using two alternative orientation representations that will be employed throughout this manuscript: rotation matrices and quaternions. These representations are mathematically equivalent but offer different advantages depending on the application context. The rotation matrix formulation, often parametrized through Euler angles, provides an intuitive geometric interpretation and is widely used in classical control design. On the other hand, the quaternion representation avoids singularities inherent to Euler angles and offers computational efficiency, making it particularly suitable for high-performance control implementations and attitude estimation algorithms. Both formulations are presented here to provide a comprehensive framework for the control design and analysis that will be developed in subsequent chapters of this work.

1.2.2.1 Euler-angles representation

With this convention, the state vector is represented by

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \boldsymbol{\delta} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^{12}, \tag{1.10}$$

where $\mathbf{p} \in \mathbb{R}^3$ denotes the position, $\mathbf{v} \in \mathbb{R}^3$ represents the linear velocity expressed in the world frame $\mathscr{F}_W$,, $\boldsymbol{\delta} = \left[\phi \ \theta \ \psi\right]^\top \in \mathbb{R}^3$ contains the Euler angles (roll, pitch, yaw) describing the attitude, and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity expressed in the body frame $\mathscr{F}_B$. The relationship between the angular velocity $\boldsymbol{\omega}$ and the time derivative of the Euler angles $\dot{\boldsymbol{\delta}}$ is given by (1.11), where $s_\phi = \sin(\phi)$, $c_\phi = \cos(\phi)$, $s_\theta = \sin(\theta)$, and $c_\theta = \cos(\theta)$.

$$\boldsymbol{\omega} = \mathbf{W}(\boldsymbol{\delta})\dot{\boldsymbol{\delta}}, \quad \mathbf{W}(\boldsymbol{\delta}) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & c_\theta s_\phi \\ 0 & -s_\phi & c_\theta c_\phi \end{bmatrix} \tag{1.11}$$

The complete state-space representation of the multirotor dynamics is obtained by combining the kinematic equations previously introduced with the dynamic equations of motion. Assembling the full model requires integrating the position kinematics $\dot{\mathbf{p}} = \mathbf{v}$, the orientation kinematics $\dot{\boldsymbol{\delta}} = \mathbf{W}(\boldsymbol{\delta})^{-1}\boldsymbol{\omega}$, and the dynamic equations governing the evolution of velocities. The resulting state-space representation, expressed in terms of Euler angles, is presented in compact form in (1.12).

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.12a}$$

$$\dot{\boldsymbol{\delta}} = \mathbf{W}(\boldsymbol{\delta})^{-1}\boldsymbol{\omega} \tag{1.12b}$$

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}(\boldsymbol{\delta})\mathbf{F}\mathbf{u} \tag{1.12c}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u} \tag{1.12d}$$

1.2.2.2 Quaternion-based representation

Alternatively, the orientation dynamics can be formulated using quaternions, avoiding the singularities inherent to Euler angle representations. In this formulation, the orientation is described by the unit quaternion $\mathbf{q} = \begin{bmatrix}\eta & \boldsymbol{\epsilon}^\top\end{bmatrix}^\top \in \mathbb{S}^3$, where $\eta \in \mathbb{R}$ denotes the scalar part and $\boldsymbol{\epsilon} \in \mathbb{R}^3$ represents the vector part. The quaternion $\mathbf{q}$ encodes the relative orientation between the body frame $\mathscr{F}_B$ and the world frame $\mathscr{F}_W$, subject to the normalization constraint $\eta^2 + \|\boldsymbol{\epsilon}\|^2 = 1$.

The state vector in this quaternion-based representation is composed of

$$\mathbf{x} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{S}^3 \times \mathbb{R}^3, \tag{1.13}$$

where $\mathbf{p} \in \mathbb{R}^3$ and $\mathbf{v} \in \mathbb{R}^3$ retain their previous meanings as position and linear velocity in $\mathscr{F}_W$, while $\mathbf{q} \in \mathbb{S}^3$ replaces the Euler angles $\boldsymbol{\delta}$ for orientation description, and $\boldsymbol{\omega} \in \mathbb{R}^3$ remains the angular velocity in $\mathscr{F}_B$.

The orientation kinematics in quaternion form extends the previously introduced kinematic framework. While the position kinematics $\dot{\mathbf{p}} = \mathbf{v}$ remains unchanged, the rotational kinematics is now expressed through the quaternion derivative, which relates the time evolution of $\mathbf{q}$ to the angular velocity $\boldsymbol{\omega}$ via the quaternion composition operation $\circ$. This composition operation, defined for two quaternions $\mathbf{q}_1 = \begin{bmatrix}\eta_1 & \boldsymbol{\epsilon}_1^\top\end{bmatrix}^\top$ and $\mathbf{q}_2 = \begin{bmatrix}\eta_2 & \boldsymbol{\epsilon}_2^\top\end{bmatrix}^\top$, yields

$$\mathbf{q}_1 \circ \mathbf{q}_2 = \begin{bmatrix} \eta_1\eta_2 - \boldsymbol{\epsilon}_1^\top\boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{\epsilon}_1 \times \boldsymbol{\epsilon}_2 \end{bmatrix}. \tag{1.14}$$

The quaternion kinematics equation can be derived by noting that the quaternion derivative is related to the angular velocity through

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \tag{1.15}$$

which, when expanded using the quaternion composition operation, yields

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta\mathbf{I}_3 + [\boldsymbol{\epsilon}]_\times \end{bmatrix}\boldsymbol{\omega}, \tag{1.16}$$

where $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$ is the identity matrix and $[\boldsymbol{\epsilon}]_\times$ denotes the skew-symmetric matrix associated with $\boldsymbol{\epsilon}$ as defined in (1.2).

The complete quaternion-based state-space model of the multirotor is then obtained by combining the kinematic and dynamic equations:

$$\dot{\mathbf{p}} = \mathbf{v} \tag{1.17a}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \mathbf{I}_3 - [\boldsymbol{\epsilon}]_\times \end{bmatrix}\boldsymbol{\omega} \tag{1.17b}$$

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}(\mathbf{q})\mathbf{F}\mathbf{u} \tag{1.17c}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}\mathbf{u} \tag{1.17d}$$

where $\mathbf{J} \in \mathbb{R}^{3\times 3}$ represents the vehicle inertia matrix in $\mathscr{F}_B$, $[\boldsymbol{\epsilon}]_\times$ stands for the skew-symmetric matrix associated to the vector $\boldsymbol{\epsilon}$, and $\mathbf{e}_3 \in \mathbb{R}^3$ refers to the third column of the identity matrix $\mathbf{I}_3 \in \mathbb{R}^{3\times 3}$, identifying the direction of $\mathbf{z}_W$.

It is the possible to apply the force analysis from [4], in order to decouple forces and torques of these systems.

1.2.2.3 Control Force Decomposition

Introducing the full-rank matrices $\mathbf{A} \in \mathbb{R}^{6\times 3}$ and $\mathbf{B} \in \mathbb{R}^{6\times 3}$ so that $\mathrm{Im}(\mathbf{A}) = \mathrm{Im}(\mathbf{M}^\top)$ and $\mathrm{Im}(\mathbf{B}) = \ker(\mathbf{M})$, it is possible to express any control input vector $\mathbf{u} \in \mathcal{U} = \bar{\mathcal{U}}^6$ as the sum of two components, namely

$$\mathbf{u} = \mathbf{u}_A + \mathbf{u}_B = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}\begin{bmatrix} \tilde{\mathbf{u}}_A \\ \tilde{\mathbf{u}}_B \end{bmatrix}, \quad \tilde{\mathbf{u}}_A, \tilde{\mathbf{u}}_B \in \mathbb{R}^3, \tag{1.18}$$

where $\mathbf{u}_A \in \mathcal{U}_A = \mathcal{U} \cap \mathrm{Im}(\mathbf{A})$ and $\mathbf{u}_B \in \mathcal{U}_B = \mathcal{U} \cap \mathrm{Im}(\mathbf{B})$.

The decomposition (1.18) in turns implies the decomposition of any control force vector. Specifically, denoting with $\mathcal{F} = \left\{\mathbf{f}_c \in \mathbb{R}^3 \mid \mathbf{f}_c = \mathbf{F}\mathbf{u}, \mathbf{u} \in \mathcal{U}\right\} \subseteq \mathrm{Im}(\mathbf{F})$ the *control force space*, any $\mathbf{f}_c \in \mathcal{F}$ can be expressed as the sum of

$$\begin{aligned} \mathbf{f}_c^A &= \mathbf{F}\mathbf{u}_A \in \mathcal{F}_A \subseteq \mathrm{Im}(\mathbf{F}\mathbf{A}), \\ \mathbf{f}_c^B &= \mathbf{F}\mathbf{u}_B \in \mathcal{F}_B \subseteq \mathrm{Im}(\mathbf{F}\mathbf{B}) \end{aligned} \tag{1.19}$$

where

$$\begin{aligned} \mathcal{F}_A &= \left\{\mathbf{f}_c \in \mathbb{R}^3 \mid \mathbf{f}_c = \mathbf{F}\mathbf{u}_A, \mathbf{u}_A \in \mathcal{U}_A\right\}, \\ \mathcal{F}_B &= \left\{\mathbf{f}_c \in \mathbb{R}^3 \mid \mathbf{f}_c = \mathbf{F}\mathbf{u}_B, \mathbf{u}_B \in \mathcal{U}_B\right\}. \end{aligned} \tag{1.20}$$

From (1.18) it follows also that $\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u}_A$, hence $\mathbf{f}_c^A$ represents the ‘spurious’ force arising from the requirement of achieving a certain control moment, while $\mathbf{f}_c^B$ corresponds to the force that can be independently assigned.

According to [4], a UAV is fully decoupled when $\mathbf{f}_c$ can be assigned in $\mathcal{F}$ independently on $\boldsymbol{\tau}_c$, i.e., when the *Zero Moment (ZM) control force space* $\mathcal{F}_B$ coincides with $\mathcal{F}$. For the STHs, $\mathcal{F}_A = \{\mathbf{0}_3\}$ and $\mathcal{F}_B = \mathcal{F}$ for any model that is going to be considered in this work: any STH is fully decoupled regardless of the tilt angle selection. However, the choice of the parameters that we are going to change, will affects $\mathcal{F}_B = \mathcal{F}$, thus influences the platform maneuverability.

Notation regarding the multirotor model

$\alpha \in \Gamma_\alpha = [0, 90°)$	tilt cant angle
$\beta \in \Gamma_\beta = [0, 90°)$	tilt dihedral angle
$\mathbf{F}_\star \in \mathbb{R}^{3\times 6}$	control force input matrix
$\mathbf{M}_\star \in \mathbb{R}^{3\times 6}$	control moment input matrix
$\mathbf{A}_\star \in \mathbb{R}^{6\times 3}$	matrix s.t. $\mathrm{Im}(\mathbf{A}_\star) = \mathrm{Im}(\mathbf{M}_\star^\top)$
$\mathbf{B}_\star \in \mathbb{R}^{6\times 3}$	matrix s.t. $\mathrm{Im}(\mathbf{B}_\star) = \ker(\mathbf{M}_\star)$
$\mathbf{H}_\star \in \mathbb{R}^{3\times 3}$	matrix s.t. $\mathbf{H}_\star = \mathbf{F}_\star \mathbf{B}_\star$
$\mathcal{U} = \bar{\mathcal{U}}^6, \bar{\mathcal{U}} = [0, \bar{u}]$	space of control input vector, $\bar{u} \in \mathbb{R}_{\geq 0}$
$\mathcal{F}(\star) \in \mathbb{R}^3$	control force (ctrl frc) space
$\mathcal{F}_B(\star) \in \mathbb{R}^3$	Zero Moment (ZM) ctrl frc space
$\mathcal{F}_B^h(\star) \in \mathbb{R}^3$	ZM control force space with gravity compensation
$V_{\mathcal{F}_B} \in \mathbb{R}_{\geq 0}$	volume of $\mathrm{conv}(\mathcal{F}_B)$
$A_{\mathcal{F}_B^h} \in \mathbb{R}_{\geq 0}$	area of $\mathrm{conv}(\mathcal{F}_B^h)$
$V_{\mathcal{F}_B^h} \in \mathbb{R}_{\geq 0}$	extra-hovering frc ctrl volume
$r_o \in \mathbb{R}_{\geq 0}$	outer circle radius of $\mathrm{conv}(\mathcal{F}_B^h)$
$r_i \in \mathbb{R}_{\geq 0}$	inner circle radius of $\mathrm{conv}(\mathcal{F}_B^h)$

In the previous table the symbol $\star$ stands for the parameter for which the term depends. In this manuscript it will either be α or (α, β)

1.2.3 Trajectory tracking control

The second part of this manuscript treats the trajectory tracking problem. This is, to control a dynamical system in a way that it makes it follow a reference trajectory.

Specifically, considering a general nonlinear time-varying system to be written as

Problem 1.2.1 (*Trajectory tracking*)

We define a reference state trajectory $\mathbf{z}(t) \in \mathbb{R}^n$, *for which the corresponding output trajectory is* $\mathbf{y}_z(t) \in \mathbb{R}^m$, *and the controlled system trajectory* $\mathbf{x}(t) \in \mathbb{R}^n$ *with output* $\mathbf{y}_x(t) \in \mathbb{R}^m$, *as follows:*

$$\dot{\mathbf{z}} = f(\mathbf{z}, t) + g(\mathbf{z}, t)\psi(t) = \varphi(\mathbf{z}, t) \tag{1.21a}$$
$$\mathbf{y}_z = h(\mathbf{z})$$
$$\dot{\mathbf{x}} = f(\mathbf{x}, t) + g(\mathbf{x}, t)u(\cdot) \tag{1.21b}$$
$$\mathbf{y}_x = h(\mathbf{x})$$

In (1.21), $f(\cdot, t) : \mathbb{R}^n \times \mathbb{R}_{>0} \to \mathbb{R}^n$ *is the system dynamics function,* $g(\cdot, t) : \mathbb{R}^n \times \mathbb{R}_{>0} \to \mathbb{R}^{n\times p}$ *is the input matrix function and* $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ *is the output function. Then,* $u : U_I \to \mathbb{R}^m$ *represents a general controller that takes as function a general set* U_I *(usually coincident the inputs or the outputs of the same system), and results in* p *control inputs. finally,* $\psi : \mathbb{R}_{>0} \to \mathbb{R}^m$ *is the reference trajectory input, that is the input trajectory generating the desired state (or output) trajectory. The functions* f, g, h *are the same for both systems.*

Moreover, the function $\varphi : \mathbb{R}^n \times \mathbb{R}_{>0} \to \mathbb{R}^n$ *is the function that represents the time-varying dynamics of the reference system, in order to consider it as an autonomous system.*

It is then possible to define the error between the reference trajectory and the controlled system one as

$$\tilde{\mathbf{x}}(t) = \mathbf{z}(t) - \mathbf{x}(t) \tag{1.22a}$$

or

$$\widetilde{\mathbf{y}_x}(t) = \mathbf{y}_z(t) - \mathbf{y}_x(t) \tag{1.22b}$$

depending if the signal to track is the state signal or the output one.

And the problem of trajectory tracking is to minimize the error of as much as possible

Remark 1.2.1 (*Reference notation*)

In the remainder of this manuscript, particularly in Part II, an alternative notational convention is adopted for reference trajectories. Specifically, the subscript $(\cdot)_r$ *is used to denote* reference *quantities, as an alternative to the* $(\cdot)_z$ *notation introduced above. Therefore, depending on the context, the reference state trajectory may be denoted as* $\mathbf{x}_r$ *instead of* $\mathbf{z}$*. Both notations refer to the same mathematical object and are used interchangeably throughout the text.*

1.2.4 Differential Flatness

DF is a structural property of dynamical systems that simplifies trajectory generation and control design [7]. A system that possess this property admits a special set of outputs, called *flat outputs*, through which all system states and inputs can be algebraically expressed using only the flat outputs and a finite number of their time derivatives.

This property is particularly valuable in the context of trajectory tracking, as it enables the direct computation of feedforward control inputs from a desired output trajectory without solving differential equations. Furthermore, differentially flat systems allow for straightforward trajectory planning in the flat output space, with guaranteed feasibility in the state and input spaces.

Many underactuated mechanical systems, including multirotors in certain configurations, exhibit differential flatness, making this framework highly relevant for the control problems addressed in this thesis.

Moreover, when control problems include constrained optimization, formulating them in the DF framework can simplify the problem structure. A flatness-based representation reduces the number of differential constraints that numerical solvers must handle. As a result, this reformulation can dramatically accelerate computation, in some cases by orders of magnitude [8]. This computational benefit makes differential flatness particularly attractive for real-time trajectory generation and model predictive control.

Definition 1.2.1 (*Differential Flatness*)

A system is said to be differentially flat if it is possible to express its states and inputs as functions of a set of outputs and a finite number of its derivatives (flat outputs). Formally, introducing the state, input, and output vectors, namely $\mathbf{x} \in \mathbb{R}^n$*,* $\boldsymbol{\mu} \in \mathbb{R}^m$ *and* $\mathbf{y} \in \mathbb{R}^m$ *with* $n, m \in \mathbb{N}$*, for a differentially flat system we can identify the functions* $g_x(\cdot)$ *and* $g_\mu(\cdot)$ *such that outputs of the kind*

$$\mathbf{y} = \mathbf{y}(\mathbf{x}, \boldsymbol{\mu}, \dot{\boldsymbol{\mu}}, \ddot{\boldsymbol{\mu}}, \dots) \tag{1.23}$$

can be used to write (1.24).

$$\mathbf{x} = g_x(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots) \text{ and } \boldsymbol{\mu} = g_\mu(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}, \dots). \tag{1.24}$$

For linear systems,differential flatness is equivalent to controllability. While all flat systems are controllable, not all controllable systems are flat.

Note that it is possible that the *flat input* $\boldsymbol{\mu}$ coincides with the original system inputs (often referenced as $\mathbf{u}$ in this manuscript).

Part I

Modeling: Actuation and Maneuverability

Chapter 2

Alpha tilted hexarotor

Contents

2.1 Introduction

The maneuverability analysis presented in this thesis begins with the examination of the first configuration that we investigate for the Generically Tilted MultiRotor (GTMR) described in subsection 1.2.2. The comprehensive forces analysis methodology conducted in [4] will be systematically exploited and adapted to facilitate the study of this specific configuration, which is illustrated and detailed in Figure 2.1.

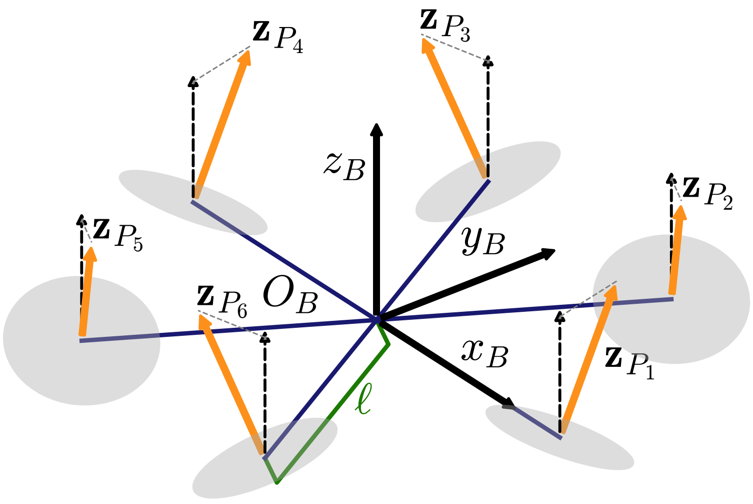


Figure 2.1: STH model - orange arrows represent the tilted rotors spinning axes. The cant angle α defines the tilt of each propellers' spinning axes relative to each propeller vertical axis $\mathbf{z}_B$.

2.2 STH Actuation Model: Cant angle addition

Starting from the base model in subsection 1.2.2, we now explicitly state the spinning axis $\mathbf{z}_{P_i}$ for each propeller. Written in (2.1), it is described by a concatenation of the two canonical rotations around the axes x and z. The rotation around the z axis results in a symmetric allocation of the rotors with respect to the UAV CoM, while the rotation around the x axis introduces novel actuation properties to the hexarotor under investigation.

$$\mathbf{z}_{P_i}(\alpha) = \mathbf{R}_z\left((i-1)60^\circ\right)\mathbf{R}_x((-1)^i\alpha)\mathbf{e}_3. \tag{2.1}$$

This selection uniquely characterizes an α-STH, where each configuration is defined by a specific *tilt cant angle* $\alpha \in \Gamma_\alpha := [0, 90^\circ)$[1]. Notably, star-shaped collinear hexarotors correspond to the special case $\alpha = 0$ and are therefore treated as a particular realization of STHs. Consequently, all equations in the STH model presented in subsection 1.2.2 inherit the dependence on the angle α through (2.1). By following the derivation chain from (2.1), we obtain the control force matrix $\mathbf{F}_\alpha$ and control moment matrix $\mathbf{M}_\alpha$ as expressed in (2.2).

$$\mathbf{F}_\alpha = c_f \begin{bmatrix} 0 & \frac{\sqrt{3}}{2}s_\alpha & -\frac{\sqrt{3}}{2}s_\alpha & 0 & \frac{\sqrt{3}}{2}s_\alpha & -\frac{\sqrt{3}}{2}s_\alpha \\ s_\alpha & -\frac{1}{2}s_\alpha & -\frac{1}{2}s_\alpha & s_\alpha & -\frac{1}{2}s_\alpha & -\frac{1}{2}s_\alpha \\ c_\alpha & c_\alpha & c_\alpha & c_\alpha & c_\alpha & c_\alpha \end{bmatrix} \tag{2.2a}$$

$$\begin{aligned} \mathbf{M}_\alpha = c_\tau &\begin{bmatrix} 0 & \frac{\sqrt{3}}{2}r\,c_\alpha & \frac{\sqrt{3}}{2}r\,c_\alpha & 0 & -\frac{\sqrt{3}}{2}r\,c_\alpha & -\frac{\sqrt{3}}{2}r\,c_\alpha \\ -r\,c_\alpha & -\frac{1}{2}r\,c_\alpha & \frac{1}{2}r\,c_\alpha & r\,c_\alpha & \frac{1}{2}r\,c_\alpha & -\frac{1}{2}r\,c_\alpha \\ r\,s_\alpha & -r\,s_\alpha & r\,s_\alpha & -r\,s_\alpha & r\,s_\alpha & -r\,s_\alpha \end{bmatrix} \\ + c_\tau &\begin{bmatrix} 0 & -\frac{\sqrt{3}}{2}s_\alpha & -\frac{\sqrt{3}}{2}s_\alpha & 0 & -\frac{\sqrt{3}}{2}s_\alpha & -\frac{\sqrt{3}}{2}s_\alpha \\ s_\alpha & \frac{1}{2}s_\alpha & -\frac{1}{2}s_\alpha & -s_\alpha & -\frac{1}{2}s_\alpha & \frac{1}{2}s_\alpha \\ c_\alpha & -c_\alpha & c_\alpha & -c_\alpha & c_\alpha & -c_\alpha \end{bmatrix} \end{aligned} \tag{2.2b}$$

where $r = (c_f/c_\tau)\ell \in \mathbb{R}_{\geq 0}$, and c and s stand for cosine and sine function, respectively.

This results in the *control force* and *control torque* displayed in (2.3), once again highlighting the dependency on the angle α.

$$\mathbf{f}_c(\alpha) = \mathbf{F}_\alpha\mathbf{u} \tag{2.3a}$$

$$\boldsymbol{\tau}_c(\alpha) = \mathbf{M}_\alpha\mathbf{u} \tag{2.3b}$$

Remark 2.2.1 (*STH case study*)

> *In the remaining part of this chapter, the numerical results refer to the case study of the STH platform described in [2], characterized by the parameters reported in Table 2.1. This is simply due to the physical availability of the UAV with these parameters in the university laboratory, easing the practical assessment of the results in the future.*

For any choice of tilt angle $\alpha \in \Gamma_\alpha$, it follows that $1 \leq \text{rk}\mathbf{f}_\alpha \leq \text{rk}\mathbf{M}_\alpha \leq 3$ as explained in [4]. In detail, the control moment input matrix is always full rank when $\alpha \in \Gamma_\alpha$, while the control force input matrix is rank deficient when $\alpha = 0$: this happens for common collinear hexarotors. Note also that $\text{rk}\mathbf{f}_\alpha = 2$ when $\alpha = \pm 90^\circ$, corresponding to the case wherein the thrust force generated by any propeller lies on the $(\mathbf{x}_B, \mathbf{y}_B)$ plane.

Finally, a STH is fully actuated if $\text{rk}\mathbf{C}_\alpha = 6$ being $\mathbf{C}_\alpha = \left[\mathbf{f}_\alpha^\top\ \mathbf{M}_\alpha^\top\right]^\top \in \mathbb{R}^{6\times 6}$. This condition is satisfied if $\alpha \in \Gamma_\alpha \backslash \{0\}$.

2.3 Maneuverability Analysis

In this section, we devise a geometric strategy for evaluating STH maneuverability by assessing its capacity to generate arbitrary control forces independently of control

m[kg]	ℓ [m]	c_f[N/Hz^2]	c_τ[Nm/Hz^2]	$\bar{u}$ [Hz^2]
3.500	0.385	$1.500e-03$	$4.590e-05$	108^2

Table 2.1: Parameters of the STH platform case study

---

[1] Negative values of α yield symmetric results; thus, the analysis remains valid by considering only non-negative angles.

moments while ensuring gravity compensation. Our objectives are twofold: to provide an analysis approach for evaluating an STH's suitability for specific tasks by leveraging its actuation constraints, and to establish design guidelines for new STHs that meet specific maneuverability requirements.

To ease the notation, hereafter we drop the dependence on the tilt angle α when not necessary. Moreover, we recall that all newly introduced symbols are summarized within subsection 1.2.2.

2.4 Zero-moment Control Force Volume

To investigate the zero-moment control force space, we first make the observation that any $\mathbf{f}_c^B \in \mathcal{F}_B$ can be expressed as the product $\mathbf{f}_c^B = \mathbf{f}_\alpha \mathbf{B}_\alpha \tilde{\mathbf{u}}_B$' with $\tilde{\mathbf{u}}_B = \begin{bmatrix}\tilde{u}_{B,1} & \tilde{u}_{B,2} & \tilde{u}_{B,3}\end{bmatrix}^\top \in \bar{\mathcal{U}}^3$. In addition, exploiting (2.2), we verify that $\mathbf{B}_\alpha = \begin{bmatrix}\mathbf{I}_3 & \mathbf{I}_3\end{bmatrix}^\top$ for any $\alpha \in \Gamma_\alpha$. This implies that for any $\mathbf{u}_B \in \mathcal{U}_B$ it is $u_{B,k} = u_{B,k+3} = \tilde{u}_{B,k}$ with $\tilde{u}_{B,k} \in \bar{\mathcal{U}}$ being $k \in \{1,2,3\}$. More interestingly, by introducing the matrix $\mathbf{H}_\alpha = \mathbf{f}_\alpha \mathbf{B}_\alpha \in \mathbb{R}^{3\times 3}$, we have that any $\mathbf{f}_c^B \in \mathcal{F}_B$ is such that

$$\mathbf{f}_c^B = \mathbf{H}_\alpha \tilde{\mathbf{u}}_B = c_f \begin{bmatrix} 0 & \sqrt{3}s_\alpha & -\sqrt{3}s_\alpha \\ 2s_\alpha & -s_\alpha & -s_\alpha \\ 2c_\alpha & 2c_\alpha & 2c_\alpha \end{bmatrix} \begin{bmatrix} \tilde{u}_{B,1} \\ \tilde{u}_{B,2} \\ \tilde{u}_{B,3} \end{bmatrix}. \tag{2.4}$$

The zero-moment control force space is characterized by linear combinations of the columns of matrix $\mathbf{H}_\alpha$, scaled by the maximum control input available to each propeller. This corresponds to the maximum admissible value for each $\tilde{u}_{B,k}$, where $k \in \{1,2,3\}$. More precisely, the set $\mathcal{F}_B$ can be expressed as the collection of all forces obtained by taking convex combinations of the column vectors of $\mathbf{H}_\alpha$, each scaled by the upper bound $\bar{u}$ on the control input. Formally, as stated in (2.4), we have

$$\mathcal{F}_B = \left\{ \textstyle\sum_{i=1}^3 \epsilon_i \, \bar{u} \mathbf{H}_\alpha \mathbf{e}_i, \ \epsilon_i \in [0,1] \right\}. \tag{2.5}$$

From a geometrical perspective, the expression in (2.5) characterizing the set $\mathcal{F}_B$ corresponds to a bounded, convex, and finite polytope embedded within the three-dimensional force space. Given the kinematic structure previously defined and the actuator constraints, this polytope emerges as a natural representation of the zero-moment control force capabilities. The volume $V_{\mathcal{F}_B} \in \mathbb{R}_{\geq 0}$ of this polytope serves as a suitable and meaningful scalar index for quantifying the maneuverability properties of STHs. In particular, a larger polytope volume signifies an enhanced capability to generate diverse force combinations across different directions, while simultaneously maintaining a zero net moment condition on the platform. This volume metric can be computed analytically through the determinant of the matrix $\bar{u}\mathbf{H}_\alpha$, as presented in (2.6):

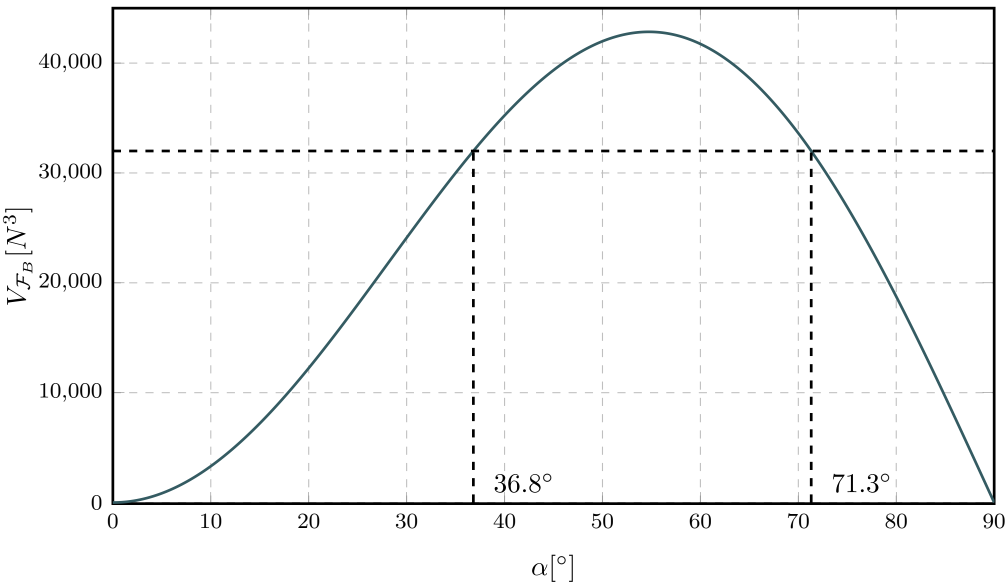
$$V_{\mathcal{F}_B} = \left|\det\left(\bar{u}\mathbf{H}_\alpha\right)\right|. \tag{2.6}$$

The determinant of the matrix $\bar{u}\mathbf{H}_\alpha$ equals the magnitude of the scalar triple product of its column vectors. This corresponds to the volume of the parallelepiped they span in the three-dimensional force space. The absolute value in (2.6) ensures this volume is treated as positive. In our analysis, we are not concerned with the sign, which would merely represent the spatial orientation of the vectors forming $\bar{u}\mathbf{H}_\alpha$.

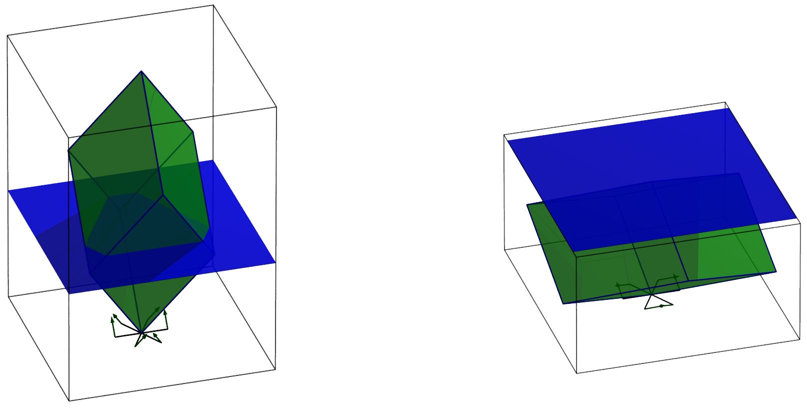
Therefore, the index $V_{\mathcal{F}_B}$ represents the volume of the convex hull spanned by the column vectors of $\bar{u}\mathbf{H}_\alpha$. Given the kinematic structure previously defined, the index (2.6) can be computed in closed form. Its value depends on both the actuator parameters and the selected tilt angle as follows:

$$V_{\mathcal{F}_B} = 12\sqrt{3}(c_f \bar{u})^3 c_\alpha s_\alpha^2. \tag{2.7}$$

Figure 2.2a shows the variation of the index (2.7) as a function of $\alpha \in \Gamma_\alpha$. The volume of the polytope representing $\mathcal{F}_B$ diminishes as the tilt angle approaches the boundaries



(a) Variation of the zero-moment control force volume $V_{\mathcal{F}_B}$ as a function of the tilt angle $\alpha \in \Gamma_\alpha$



(b) $\alpha = 36.8°$ $V_{\mathcal{F}_B} = 32000[N^3]$

(c) $\alpha = 71.3°$ $V_{\mathcal{F}_B} = 32000[N^3]$

Figure 2.2: Investigation of $V_{\mathcal{F}_B}$: (a) value of $V_{\mathcal{F}_B}$ in function of $\alpha \in \Gamma_\alpha$, (b-c) 3D polytopes representing the zero-moment force spaces characterized by the same volume (corresponding to the tilt angles marked in orange in (a)) - the blue plane represents the hovering plane introduced in section 2.5.

of Γ_α (i.e., 0° or 90°), and reaches higher values for intermediate angles. The curve exhibits asymmetry, with a maximum occurring near $\alpha \sim 55^\circ$. Notably, distinct values of α may produce identical values of $V_{\mathcal{F}_B}$. Consequently, this single index is insufficient for comprehensive maneuverability characterization. As an illustration, the 3D polytopes shown in Figure 2.2b-Figure 2.2c depict the zero-moment force spaces for different α configurations: despite having approximately equal volumes, their geometric dimensions (height, width, depth) vary considerably. This underscores the necessity of additional metrics to fully capture the implications of α selection on the STH agility.

Aiming at investigating STH maneuverability for interactive tasks, we analyze static hovering conditions where UAVs must counteract gravity and reject moment disturbances. This necessity leads to a reduction in $\mathcal{F}_B$, and the magnitude of the resulting extra-hovering zero-moment force space is a valuable indicator of platform maneuverability.

2.5 Gravitation Compensation Constraint

First, we focus on the convex hull of the intersection between $\mathcal{F}_B$ and the *hovering plane*: it is the plane perpendicular to $\mathbf{e}_3$ describing the control force space

$$\mathcal{F}_B = \left\{ \mathbf{f}_c = \begin{bmatrix} f_c^x & f_c^y & f_c^z \end{bmatrix}^\top \in \mathbb{R}^3 \mid f_c^z = mg. \right\} \tag{2.8}$$

More precisely, recalling the definition of $\mathbf{f}_c^B$ in (2.4), we investigate $\operatorname{conv}(\mathcal{F}_B^h)$, that is the convex hull of the set $\mathcal{F}_B^h$ defined in (2.9).

$$\mathcal{F}_B^h = \left\{ \mathbf{f}_c = c_f \begin{bmatrix} \sqrt{3} s_\alpha (\tilde{u}_{B,2} - \tilde{u}_{B,3}) \\ s_\alpha (2\tilde{u}_{B,1} - \tilde{u}_{B,2} - \tilde{u}_{B,3}) \\ 2c_\alpha (\tilde{u}_{B,1} + \tilde{u}_{B,2} + \tilde{u}_{B,3}) \end{bmatrix}, \right. \\ \left. \sum_{k=1}^{3} \tilde{u}_{B,k} = \frac{mg}{2c_f c_\alpha},\ \tilde{u}_{B,k} \in \bar{\mathcal{U}}, k \in \{1,2,3\} \right\}. \tag{2.9}$$

Given these premises, to figure out the vertexes of $\operatorname{conv}(\mathcal{F}_B^h)$, we investigate the actuators' extreme working conditions and we distinguish the following cases:

A. $\bar{u} \geq \frac{mg}{2c_f c_\alpha}$ then it is required the action of at least a pair of propellers to counteract the gravity;

B. $\frac{mg}{4c_f c_\alpha} \leq \bar{u} < \frac{mg}{2c_f c_\alpha}$ then it is required the action of at least two pairs of propellers to counteract the gravity;

C. $\frac{mg}{6c_f c_\alpha} \leq \bar{u} < \frac{mg}{4c_f c_\alpha}$ then it is required the action of all the propellers to counteract the gravity;

D. $\bar{u} < \frac{mg}{6c_f c_\alpha}$ then the STH is not capable of counteract the gravity force and therefore is not able to take off.

Note that, based on the actuators parameters c_f and $\bar{u}$, the selection of the tilt angle is constrained to different sub-intervals of Γ_α corresponding to cases A-D.

Focusing on the cases A-C, we verify that the actuators extreme working conditions correspond to the selection of the input parameters reported in Table 2.2. In its last column, we provide the corresponding expressions for the f_c^x and f_c^y control force components, indicating also whether these values correspond to a minimum/maximum in light of (2.9). Thus, it is possible to evaluate the area $A_{\mathcal{F}_B^h} \in \mathbb{R}_{\geq 0}$ of the resulting convex hull $\operatorname{conv}(\mathcal{F}_B^h)$ for the cases A-C. In detail, exploiting the shoelace/Gauss's area formula, the value of $A_{\mathcal{F}_B^h}$ can be computed as

$$A_{\mathcal{F}_B^h} = \frac{1}{2} \left| \sum_{j=1}^{n_\bullet} (f_{c,j}^x f_{c,j+1(\mathrm{mod} n_\bullet)}^y - f_{c,j+1(\mathrm{mod} n_\bullet)}^x f_{c,j}^y) \right| \tag{2.10}$$

A	$\tilde{u}_{B,1} = 0$	$\tilde{u}_{B,3} = 0$	$\tilde{u}_{B,2} = \frac{mg}{2c_f c_\alpha}$	$f_{c,1}^x = \frac{\sqrt{3}}{2} mg\, t\alpha$	$(f_{c,max}^x)$	$f_{c,1}^y = -\frac{mg}{2}\, t\alpha$	$(f_{c,min}^y)$
	$\tilde{u}_{B,1} = 0$	$\tilde{u}_{B,2} = 0$	$\tilde{u}_{B,3} = \frac{mg}{2c_f c_\alpha}$	$f_{c,2}^x = -\frac{\sqrt{3}}{2} mg\, t\alpha$	$(f_{c,min}^x)$	$f_{c,2}^y = -\frac{mg}{2}\, t\alpha$	$(f_{c,min}^y)$
	$\tilde{u}_{B,2} = 0$	$\tilde{u}_{B,3} = 0$	$\tilde{u}_{B,1} = \frac{mg}{2c_f c_\alpha}$	$f_{c,3}^x = 0$		$f_{c,3}^y = mg\, t\alpha$	$(f_{c,max}^y)$
B	$\tilde{u}_{B,2} = \bar{u}$	$\tilde{u}_{B,3} = 0$	$\tilde{u}_{B,1} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,1}^x = \sqrt{3} c_f \bar{u}\, s_\alpha$	$(f_{c,max}^x)$	$f_{c,1}^y = mg\, t\alpha - 3c_f \bar{u}\, s_\alpha$	
	$\tilde{u}_{B,2} = \bar{u}$	$\tilde{u}_{B,1} = 0$	$\tilde{u}_{B,3} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,2}^x = \sqrt{3} \left(2c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha\right)$		$f_{c,2}^y = -\frac{mg}{2}\, t\alpha$	$(f_{c,min}^y)$
	$\tilde{u}_{B,3} = \bar{u}$	$\tilde{u}_{B,2} = 0$	$\tilde{u}_{B,1} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,3}^x = -\sqrt{3} c_f \bar{u}\, s_\alpha$	$(f_{c,min}^x)$	$f_{c,3}^y = mg\, t\alpha - 3c_f \bar{u}\, s_\alpha$	
	$\tilde{u}_{B,3} = \bar{u}$	$\tilde{u}_{B,1} = 0$	$\tilde{u}_{B,2} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,4}^x = -\sqrt{3} \left(2c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha\right)$		$f_{c,4}^y = -\frac{mg}{2}\, t\alpha$	$(f_{c,min}^y)$
	$\tilde{u}_{B,1} = \bar{u}$	$\tilde{u}_{B,3} = 0$	$\tilde{u}_{B,2} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,5}^x = \sqrt{3} \left(\frac{mg}{2} t\alpha - c_f \bar{u}\, s_\alpha\right)$		$f_{c,5}^y = 3c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha$	$(f_{c,max}^y)$
	$\tilde{u}_{B,1} = \bar{u}$	$\tilde{u}_{B,2} = 0$	$\tilde{u}_{B,3} = \frac{mg}{2c_f c_\alpha} - \bar{u}$	$f_{c,6}^x = -\sqrt{3} \left(\frac{mg}{2} t\alpha - c_f \bar{u}\, s_\alpha\right)$		$f_{c,6}^y = 3c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha$	$(f_{c,max}^y)$
C	$\tilde{u}_{B,1} = \bar{u}$	$\tilde{u}_{B,2} = \bar{u}$	$\tilde{u}_{B,3} = \frac{mg}{2c_f c_\alpha} - 2\bar{u}$	$f_{c,1}^x = \sqrt{3} \left(3c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha\right)$	$(f_{c,max}^x)$	$f_{c,1}^y = 3c_f \bar{u}\, s_\alpha - \frac{mg}{2}\, t\alpha$	$(f_{c,max}^y)$
	$\tilde{u}_{B,2} = \bar{u}$	$\tilde{u}_{B,3} = \bar{u}$	$\tilde{u}_{B,1} = \frac{mg}{2c_f c_\alpha} - 2\bar{u}$	$f_{c,2}^x = 0$		$f_{c,2}^y = mg\, t\alpha - 6c_f \bar{u}\, s_\alpha$	$(f_{c,min}^y)$
	$\tilde{u}_{B,1} = \bar{u}$	$\tilde{u}_{B,3} = \bar{u}$	$\tilde{u}_{B,2} = \frac{mg}{2c_f c_\alpha} - 2\bar{u}$	$f_{c,3}^x = -\sqrt{3} \left(3c_f \bar{u}\, s_\alpha - \frac{mg}{2} t\alpha\right)$	$(f_{c,min}^x)$	$f_{c,3}^y = 3c_f \bar{u}\, s_\alpha - \frac{mg}{2}\, t\alpha$	$(f_{c,max}^y)$

Table 2.2: Vertices of the convex hull of $\mathcal{F}_B^h$ in the cases A-C and the corresponding values of the coefficients $\tilde{u}_{B,k}$, $k = 1, 2, 3$.

where $n_\bullet \in \mathbb{N}$, with $\bullet = \{A, B, C\}$, denotes the number of vertexes in the considered cases ($n_A = n_C = 3$ and $n_B = 6$).

It follows that the area $A_{\mathcal{F}_B^h}$ can be expressed as (2.11):

$$A_{\mathcal{F}_B^h} = \begin{cases} \frac{3\sqrt{3}}{4}\left(mg\,t\alpha\right)^2 & \text{A} \\ \frac{3\sqrt{3}}{4}\left(mg\,t\alpha\right)^2 - 9\sqrt{3}\left(\frac{mg}{2}t\alpha - c_f\bar{u}\,s_\alpha\right)^2 & \text{B} \\ 3\sqrt{3}\left(3c_f\bar{u}\,s_\alpha - \frac{mg}{2}t\alpha\right)^2 & \text{C} \end{cases} \tag{2.11}$$

The variation of the area $A_{\mathcal{F}_B^h}$ as a function of the tilt angle $\alpha \in \Gamma_\alpha$ is illustrated in Figure 2.3. This figure depicts how the area of the convex hull of the zero-moment control force space evolves under hovering conditions. When examining this plot, a notable characteristic emerges: the resulting curve exhibits a pronounced asymmetric behavior. This asymmetry is considerably more evident compared to the volume variation previously presented in Figure 2.2a. The heightened asymmetric nature of the curve provides valuable additional insight. Specifically, it reveals a more nuanced understanding of the relationship between the tilt angle selection and the system's force generation capabilities. This is particularly relevant when maintaining zero moment conditions during hovering flight operations.

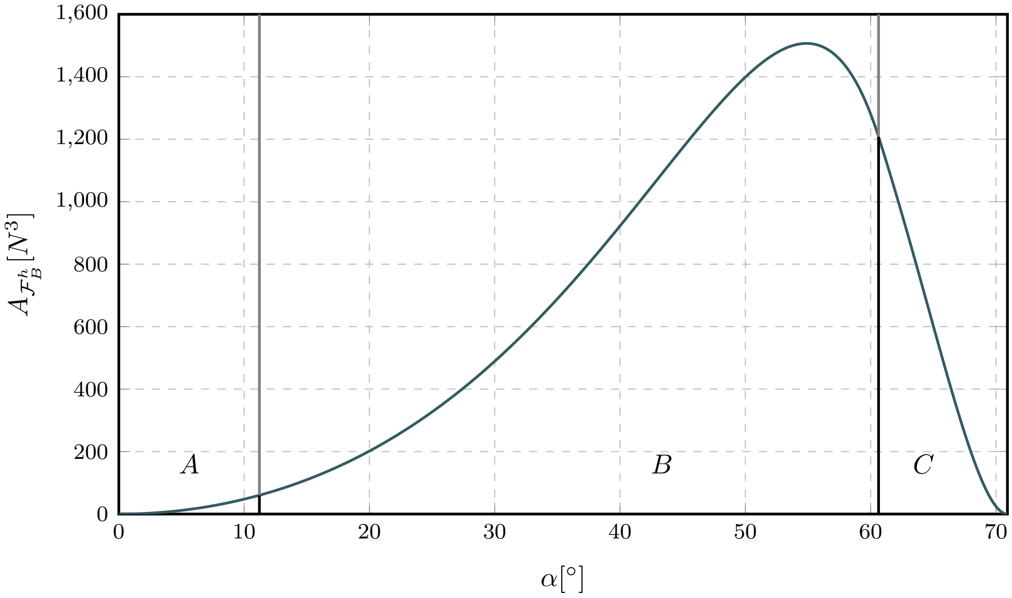
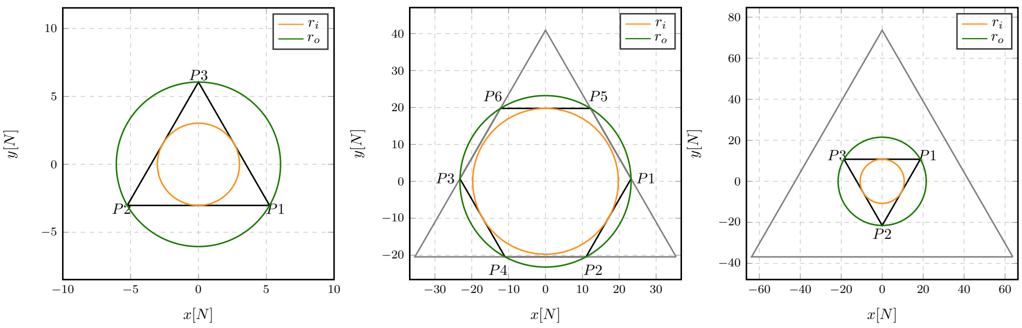


Figure 2.3: Value of $A_{\mathcal{F}_B^h}$ in function of $\alpha \in \Gamma_\alpha$, highlighting cases A-C.

Furthermore, several important observations emerge from this analysis that warrant careful consideration. Although the maximum value of the curve appears at a tilt angle similar to that observed for the zero-moment control force volume $V_{\mathcal{F}_B}$, the maximum angle that can be set while still generating enough force, results to be smaller by approximately 20°.

This practical limitation provides compelling evidence that the zero-moment control force volume alone, while mathematically interesting, offers an overly narrow perspective for real-world applications. More precisely, many configurations that appeared viable when evaluated solely through $V_{\mathcal{F}_B}$ now reveal insufficient force generation capability to sustain hover conditions. Given the kinematic structure and actuation constraints already established, it becomes evident that a more comprehensive set of metrics is essential for thorough maneuverability characterization.

To comprehensively characterize the maneuverability of STHs, we also examine the radii of the outer (circumscribed) and inner (inscribed) circles of the convex hull $\mathrm{conv}(\mathcal{F}_B^h)$.



(a) $\alpha = 10^\circ$ (case A) **(b)** $\alpha = 50^\circ$ (case B) **(c)** $\alpha = 65^\circ$ (case C)

Figure 2.4: For α selection in the cases A-C, representation of the inner c_i and outer c_o circles and of the convex hull of $\mathcal{F}_B^h$ delimited by P_i, $i \in \{1, \dots, 6\}$ corresponding to the extreme force components in Table 2.2, denoting the forces that guarantee the UAV can hover.

The radius $r_o \in \mathbb{R}_{\geq 0}$ of the outer circle quantifies the dynamic capability of the platform, capturing the extremes of the achievable f_x and f_y force components. Conversely, the radius $r_i \in \mathbb{R}_{\geq 0}$ of the inner circle serves as an index of the STH robustness, reflecting the maximum magnitude of the force that is possible to generate along the whole horizontal plane of $\mathscr{F}_B$.

In all three cases mentioned, these circles are centered at the origin of the (f_c^x, f_c^y)-plane. Their radii can be determined using the coordinates of the extreme points listed in Table 2.2.

Specifically, they result in

A. $r_i = |f_{c,1}^y|$ and $r_o = |f_{c,3}^y|$,

B. $r_i = \min\left\{|f_{c,2}^y|, |f_{c,5}^y|\right\}$ and $r_o = |[f_{c,1}^x \;\; f_{c,1}^y]|$,

C. $r_i = |f_{c,1}^y|$ and $r_o = |f_{c,2}^y|$.

In Figure 2.4, for each of the three cases, the convex hull of $\mathcal{F}_B^h$ and the corresponding inner and outer circles are depicted. In cases A and C, $\mathrm{conv}(\mathcal{F}_B^h)$ forms an equilateral triangle. Conversely, in case B, $\mathrm{conv}(\mathcal{F}_B^h)$ is a hexagon with vertices lying on the perimeter of the triangle depicted in case A. As the parameter α increases, these vertices gradually move away from the vertices of the original triangle, eventually aligning with those of the triangle depicted in case C.

The radii of the inner and outer circles variation as α varies is then displayed in Figure 2.5. The outer circle's radius attains its maximum value when cases B and C coincide; at this point, $\mathrm{conv}(\mathcal{F}_B^h)$ is a triangle with the vertices on the perimeter of the triangle introduced in the case A. Concerning the inner circle's radius, it reaches its maximum in case B, where $\mathrm{conv}(\mathcal{F}_B^h)$ is a regular hexagon.

2.6 Extra-hovering Control Force Volume

Finally, we investigate the volume $V_{\mathcal{F}_B^h} \in \mathbb{R}_{\geq 0}$ of the space of the forces exceeding the gravity compensation required by the hovering conditions. Formally, we focus on the space $\{\mathbf{f}_c = [f_c^x \; f_c^y \; f_c^z]^\top \in \mathcal{F}_B \mid f_c^z \geq mg\}$. To facilitate the analysis of this extra-hovering force space, we must examine the same three distinct cases A-C previously introduced in section 2.5, which were established based on the actuator saturation constraints and the hovering requirements. Recall that the kinematic structure and actuation constraints have already been explicitly defined in the previous sections. For representative values of α corresponding to each of these cases, illustrative examples of the extra-hovering control force space are presented in Figure 2.6. In these visualizations, the polytope representing the zero-moment control force space is depicted in green, while its intersection with the

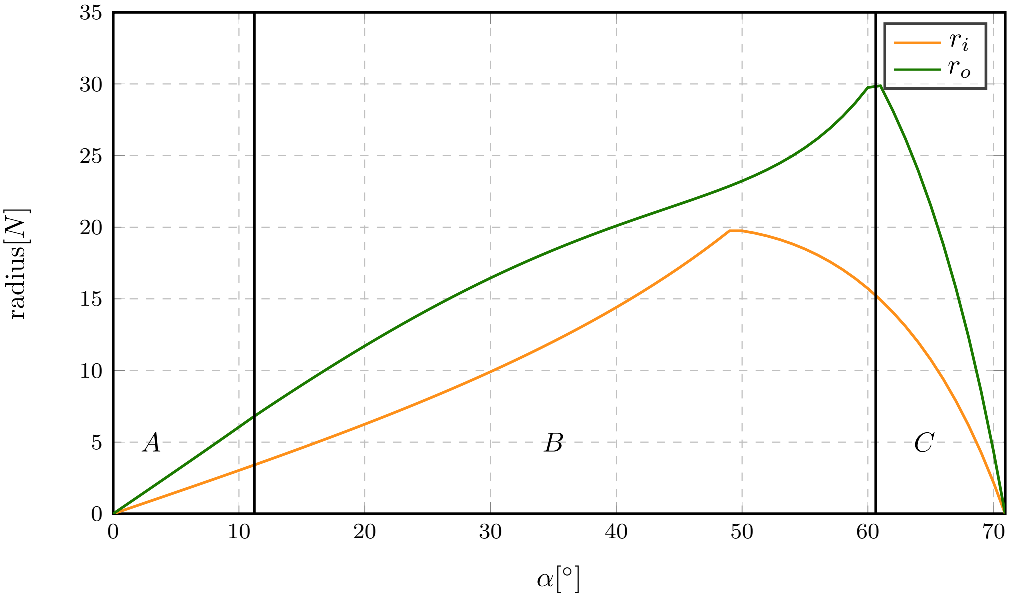
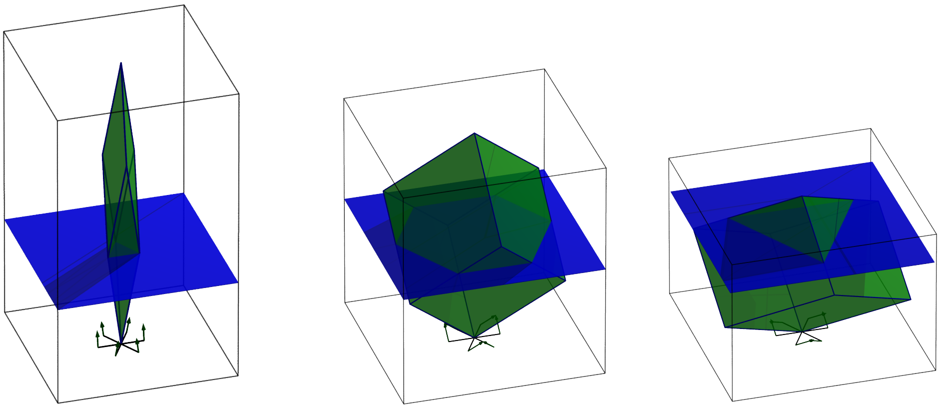


Figure 2.5: Value of r_i and r_o in function of $\alpha \in \Gamma_\alpha$, highlighting cases A-C. This plot highlights how the two radii have peaks that correspond to different angles, and how both suddenly drops after a certain angle

hovering plane is illustrated in blue. Across all three cases A-C, the volume $V_{\mathcal{F}_B^h}$ can be systematically computed by leveraging geometric considerations and exploiting the polyhedral structure of the feasible force space.



(a) $\alpha = 10^\circ$ (case A) **(b)** $\alpha = 50^\circ$ (case B) **(c)** $\alpha = 65^\circ$ (case C)

Figure 2.6: Extra-hovering control force space for various α selection in the cases A,B,C. On the bottom, a representation of the frame of the UAV, with green arrows representing the forces for each propeller. The blue plane represents the hovering plane $\mathcal{F}_B^h$

Case A The volume $V_{\mathcal{F}_B^h}$ can be computed by subtracting from the polytope representing the zero-moment control force space $V_{\mathcal{F}_B}$ the portion of the same polytope under the hovering plane. We observe that this results in a pyramid that has a height equal to mg and $A^h_{\mathcal{F}_B}$ as the base. Therefore, in case A, it is $V_{\mathcal{F}_B^h} = V_{\mathcal{F}_B} - \frac{1}{3}A^h_{\mathcal{F}_B}mg$.

Case B The geometric structure of the extra-hovering control force polytope in case B exhibits greater complexity than in case A. As illustrated in Figure 2.7, the polytope can be decomposed into a large pyramid $\overline{\mathcal{P}}$, from which three smaller, geometrically identical pyramids are removed. We denote one of these smaller pyramids as $\underline{\mathcal{P}}$, noting that by symmetry the analysis applies uniformly to all three.

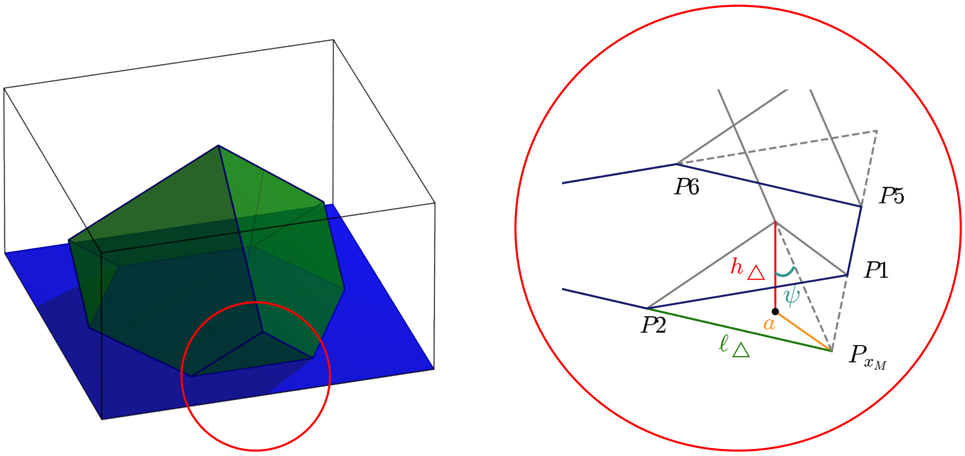


Figure 2.7: Case B: polytope representing the extra-hovering control force space for $\alpha = 25°$.This figure can be used to understaind better the terms $\ell_\triangle$, $h_\triangle$ and ψ used in section 2.6

To determine the volume of this polytope, we begin by computing the height of the main pyramid $\overline{\mathcal{P}}$. This height is given by $h_+ = h_{\max} - mg$, where $h_{\max} = (\mathbf{e}_3^\top \mathbf{f}_\alpha)(\bar{u}\mathbb{1}_6) = 6c_f c_\alpha \bar{u}$ represents the maximum realizable force along the z-axis when all propellers operate at maximum speed, with $\mathbb{1}_n$ denoting the n-dimensional vector of ones.

Following the notation established in Figure 2.7, we examine the equilateral triangle formed by vertices P_1, P_2, and P_{x_M}. This triangle, whose equilateral nature stems from the platform's geometric symmetries previously defined through the kinematic structure, serves as the base of the smaller pyramid $\underline{\mathcal{P}}$. Here, P_{x_M} denotes the extremum along the x-axis of the dashed triangle.

The height of $\underline{\mathcal{P}}$ can be expressed as $h_\triangle = h_+ - 2c_\alpha c_f \bar{u}$. The subtracted term corresponds to the projection along the z-axis of one of the polytope's edges. This edge has length $2c_f\bar{u}$, matching the norm of the vectors that generate the space $\mathcal{F}_B$.

To compute the area of the base of $\underline{\mathcal{P}}$, we introduce the angle $\psi \in [0, 180°]$ between the aforementioned edge and the height $h_\triangle$. From this geometric configuration, we can determine the length of segment a as $\bar{a} = h_\triangle \tan\psi$. The side length of the equilateral triangle then follows as $\ell_\triangle = 2\sin(60°)a$.

Consequently, the volume of the smaller pyramid is $V_\triangle = \frac{1}{3}A_\triangle h_\triangle = \frac{1}{3}\left(\frac{\sqrt{3}}{4}\ell_\triangle^2\right)h_\triangle$.

For the larger pyramid $\overline{\mathcal{P}}$, the base area comprises the sum of $A_{\mathcal{F}_B^h}$ and the combined areas of the three smaller pyramid bases. This yields a volume of $V_{\overline{\mathcal{P}}} = \frac{1}{3}(A_{\mathcal{F}_B^h} + 3A_\triangle)h_+$.

Finally, the volume of the extra-hovering control force space in case B is obtained by subtracting the volumes of the three smaller pyramids: $V_{\mathcal{F}_B^h} = \frac{1}{3}(A_{\mathcal{F}_B^h} + 3A_\triangle)h_+ - 3V_\triangle$.

Case C Given the kinematic structure and constraints established for case B, case C admits a considerably simpler geometric interpretation. The extra-hovering control force polytope in this case corresponds exactly to the large pyramid $\overline{\mathcal{P}}$ identified in the previous analysis, with none of the smaller pyramids being subtracted.

This simplification arises because the actuation constraints in case C ensure that all propellers must contribute to gravity compensation, thereby eliminating the geometric configurations that gave rise to the smaller pyramids in case B.

Consequently, the volume computation reduces to $V_{\mathcal{F}_B^h} = \frac{1}{3} A_{\mathcal{F}_B^h} h_+$, where $A_{\mathcal{F}_B^h}$ denotes the base area of the convex hull on the hovering plane, and h_+ represents the height of the pyramid as previously defined.

Having analyzed all three cases A-B-C, we can now provide a unified expression for the extra-hovering control force volume $V_{\mathcal{F}_B^h}$. The complete formulation, accounting for the distinct geometric structures arising from different actuator saturation regimes, is presented in (2.12):

$$V_{\mathcal{F}_B^h} = \begin{cases} V_{\mathcal{F}_B} - \dfrac{1}{3} A^h_{\mathcal{F}_B}\, mg & \text{A.} \\ \dfrac{1}{3}\left(A_{\mathcal{F}_B^h} + \dfrac{3\sqrt{3}}{4}\ell_\triangle^2\right) h_+ - \left(\dfrac{\sqrt{3}}{4}\ell_\triangle^2\right) h_\triangle & \text{B.} \\ \dfrac{1}{3} A_{\mathcal{F}_B^h}\, h_+ & \text{C.} \end{cases} \tag{2.12}$$

This formulation reveals how the tilt angle α, through its influence on the actuator saturation boundaries, fundamentally shapes the extra-hovering force generation capacity of the STH platform.

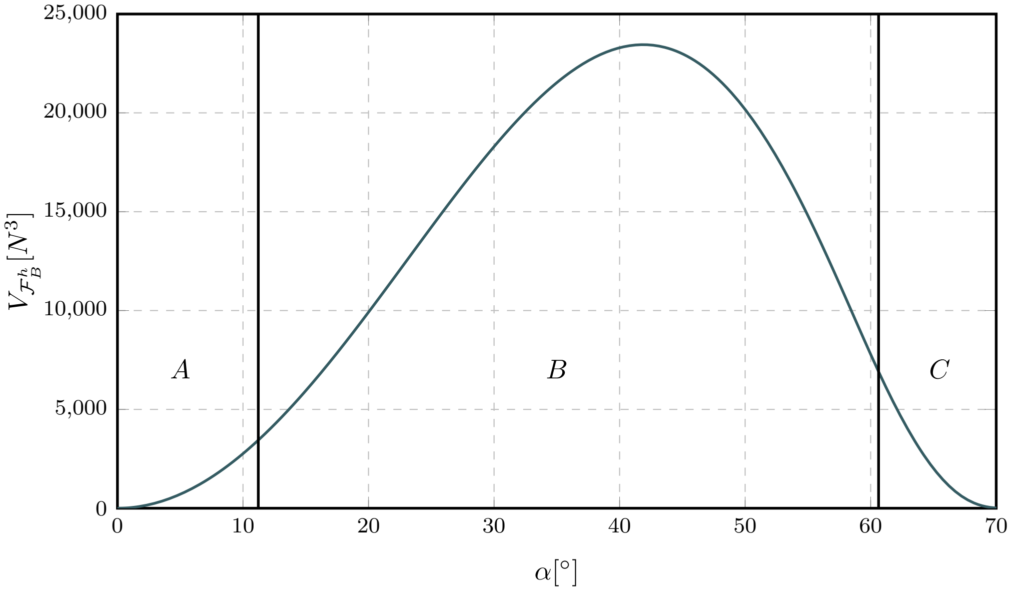


Figure 2.8: Extra-hovering control force volume $V_{\mathcal{F}_B^h}$ as a function of the tilt angle $\alpha \in \Gamma_\alpha$, illustrating the optimal configuration for force generation beyond gravity compensation

The relationship between the extra-hovering control force volume $V_{\mathcal{F}_B^h}$ and the tilt angle α is illustrated in Figure 2.8 A detailed examination of this relationship reveals several important characteristics that merit careful consideration. Notably, the curve exhibits a well-defined maximum, occurring at approximately $\alpha \approx 42^\circ$, which falls within the parameter range of case B. This maximum represents the optimal configuration for extra-hovering force generation, given the kinematic structure and actuator constraints that have been explicitly established in the preceding sections.

An interesting observation emerges when comparing this result with the optimal angles identified for the other metrics that have been previously examined. Specifically, $V_{\mathcal{F}_B^h}$ achieves its peak value at a notably smaller tilt angle than those that maximize $V_{\mathcal{F}_B}$, $A_{\mathcal{F}_B^h}$, and the outer radius r_o. This discrepancy highlights the distinct physical interpretations of these various maneuverability indices and underscores the complexity inherent in selecting an optimal design configuration for STH platforms.

2.7 Discussion

To enhance comprehension of the introduced metrics, we discuss their significance in relation to the angles that individually maximize them, as outlined in Table 2.3. This concluding section serves as an example of design analysis.

Metrics	$\alpha\,[^\circ]$				
	42	49.5	54.5	55	60.5
$V_{\mathcal{F}_B}\,[N^3]$	37039	41802	**42843**	42843	41523
$A_{\mathcal{F}^h_B}\,[N^2]$	1022	1379	1505	**1506**	1222
$r_o\,[N]$	20.71	23.05	25.26	25.55	**30.34**
$r_i\,[N]$	15.46	**19.81**	18.66	18.48	15.34
$V_{\mathcal{F}^h_B}\,[N^3]$	**23450**	20577	15441	14802	7089

Table 2.3: Proposed metrics for different values of α — case B.

We observe that the index $V_{\mathcal{F}_B}$ reaches its maximum at $\alpha = 54.5^\circ$. Interestingly, this angle also nearly maximizes $A_{\mathcal{F}^h_B}$, initially suggesting the optimality of this configuration for both metrics. However, it is important to emphasize that the proximity of these two maxima is coincidental and specific to the particular parameters chosen in this case study. For different platform configurations, these optimal angles may diverge significantly.

A more detailed examination reveals a notable disparity in the behavior of the largest attainable radii. This inconsistency becomes particularly evident when comparing with the previously discussed metrics, and even more so when examining the radii themselves. From a robustness-oriented perspective, selecting a smaller value of α could enable higher force application along the hovering plane. This enhanced capability, however, necessarily comes at the expense of a reduced maximum achievable force for specific combinations of f_x and f_y components.

This more conservative design approach finds further support when examining the extra-hovering control force volume $V_{\mathcal{F}^h_B}$. This metric reaches its peak at $\alpha = 42^\circ$, indicating significantly enhanced hovering capabilities at this configuration. Such capabilities prove valuable not only for improving overall robustness, but also for specific tasks that necessitate substantial force application in directions other than the purely vertical one. Consequently, when pursuing a conservative design strategy that balances multiple performance objectives, the choice $\alpha = 50^\circ$ could represent a reasonable compromise among the various competing requirements.

It is crucial to note, given the kinematic structure already established, that increasing the cant angle α inevitably leads to reduced propulsive efficiency. This fundamental trade-off must be carefully considered in the final design selection process.

Chapter 3

Alpha-Beta Tilted hexarotor

Contents

3.1 Introduction

This chapter studies the further extension of the multirotor model in (1.9) to study the impact of an additional tilt angle - the *dihedral* angle β - to the platform maneuverability.

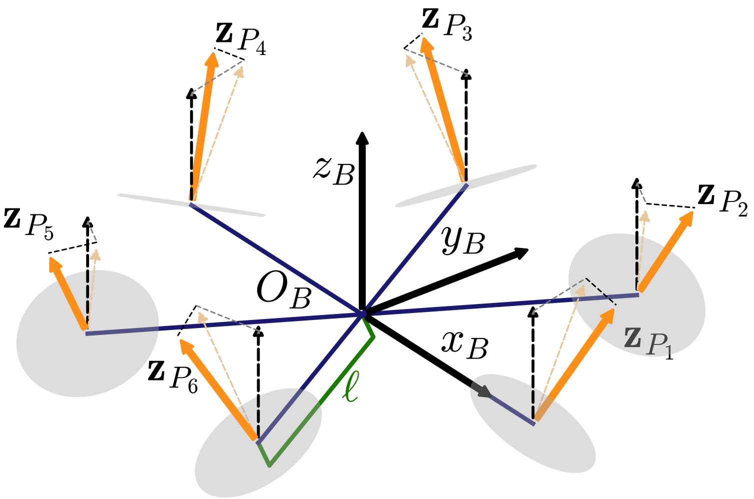


Figure 3.1: Depiction of an $\alpha - \beta$ Star-shaped Tilted Hexarotor (STH)

The modified UAV configuration is illustrated in Figure 3.1, which demonstrates the geometric effect of a positive β angle on the propeller orientation. Specifically, this parameter induces an additional outward tilt of the propellers relative to the vehicle's CoM, thereby altering the thrust vectoring characteristics of the system.

The rationale behind the incorporation of this additional design variable into the vehicle architecture is motivated by theoretical and experimental literature. It has been rigorously demonstrated that for Star shaped Generically Tilted MultiRotor (SGTMR) configurations, the introduction of such a geometric parameter is not only advantageous but rather is a necessary condition for achieving fault tolerance in the event of a single actuator failure [3]. This is of critical importance for ensuring the operational robustness and safety of multirotor systems deployed in demanding or safety-critical applications.

The objective of this chapter is to conduct a comprehensive analysis of the influence of the β parameter on various performance metrics and operational characteristics of the proposed UAV configuration. Through systematic investigation, this analysis aims to establish a rigorous theoretical foundation and to derive practical design guidelines that can inform the selection of appropriate parameter values for diverse operational scenarios and application-specific requirements.

3.2 STH Actuation Model: Dihedral angle addition

The dihedral angle β comes out in the model equations in the early definition of the propellers rotational axis, in the same equation where the cant angle is found α. Specifically, the spinning axis $\mathbf{z}_{P_i}$ is extended: (3.1) now includes the rotation matrix regarding β, in (3.1).

$$\mathbf{z}_{P_i}(\alpha, \beta) = \mathbf{R}_z\left((i-1)\tfrac{\pi}{3}\right)\mathbf{R}_y(\beta)\mathbf{R}_x((-1)^i\alpha)\mathbf{e}_3. \tag{3.1}$$

Therefore, all the terms that depend on the α angle in chapter 2, now also depends on β. In particular:

$$\mathbf{f}_i(\alpha, \beta) = c_{f_i}\omega_i^2\mathbf{z}_{P_i}(\alpha, \beta), \tag{3.2}$$

$$\boldsymbol{\tau}_i^d(\alpha, \beta) = \kappa_i c_{\tau_i}\omega_i^2\mathbf{z}_{P_i}(\alpha, \beta), \tag{3.3}$$

$$\boldsymbol{\tau}_i^t(\alpha, \beta) = \mathbf{p}_i \times \mathbf{f}_i(\alpha, \beta) \tag{3.4}$$

The extension of the geometric configuration to include the dihedral angle β necessitates a corresponding modification of the actuation matrices that characterize the force and moment generation capabilities of the system. Consequently, the matrices $\mathbf{F}_\alpha$ and $\mathbf{M}_\alpha$ introduced in (2.2), which originally depended solely on the cant angle α, must now be generalized to incorporate the influence of both angular parameters. This generalization yields the extended matrices $\mathbf{F}_{\alpha,\beta}$ and $\mathbf{M}_{\alpha,\beta}$ as defined in (3.5). These augmented matrices fully capture the combined geometric effects of both the cant angle and the dihedral angle on the thrust force and moment generation characteristics of the propulsion system.

$$[\mathbf{F}_{\alpha,\beta}]_i = \mathbf{f}_i(\alpha, \beta) \tag{3.5a}$$

$$[\mathbf{F}_{\alpha,\beta}]_i = \boldsymbol{\tau}_i^d(\alpha, \beta) + \boldsymbol{\tau}_i^t(\alpha, \beta) \tag{3.5b}$$

This change then reflects in the control force and control torque (1.7) as well, allowing us to rewrite (2.3) with the new dependency on β as in (3.6).

$$\mathbf{f}_c(\alpha, \beta) = \mathbf{F}_{\alpha,\beta}\mathbf{u} \tag{3.6a}$$

$$\boldsymbol{\tau}_c(\alpha, \beta) = \mathbf{M}_{\alpha,\beta}\mathbf{u} \tag{3.6b}$$

Finally, the full dynamical model (1.9) is rewritten with the new terms as in (3.7b).

$$m\ddot{\mathbf{p}} = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{f}_c(\alpha, \beta) = -mg\mathbf{e}_3 + \mathbf{R}\mathbf{F}_{\alpha,\beta}\mathbf{u}, \tag{3.7a}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \boldsymbol{\tau}_c(\alpha, \beta) = -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} + \mathbf{M}_{\alpha,\beta}\mathbf{u}, \tag{3.7b}$$

where $\mathbf{u} = [u_1, \ldots, u_6]^\top$ represents the vector of squared propellers speed, and $\mathbf{R} \in SO(3)$ denotes the rotation matrix from the body-fixed frame to the inertial frame.

It is important to emphasize that the dependency of the control force and torque on both geometric parameters α and β is explicitly retained in the notation throughout the analysis in this chapter. This explicit representation serves to highlight the fundamental influence of these design parameters on the system's actuation characteristics and control authority. The matrices $\mathbf{F}_{\alpha,\beta}$ and $\mathbf{M}_{\alpha,\beta}$ are fully determined by the geometric configuration of the propellers, encapsulating both the cant angle α and the dihedral angle β effects.

3.3 Unconstrained Force Allocation Analysis

The initial analytical investigation undertaken in this study concerns the characterization of the complete set of forces that can be feasibly generated by the actuation system. This analysis is conducted by exploring the entire admissible input space $\mathcal{U}$, while deliberately neglecting any constraints that may be imposed on the resultant torques produced as a consequence of the force allocation. By adopting this unconstrained formulation, it becomes possible to establish the theoretical bounds on the force generation capabilities of the system, independent of the rotational moment limitations.

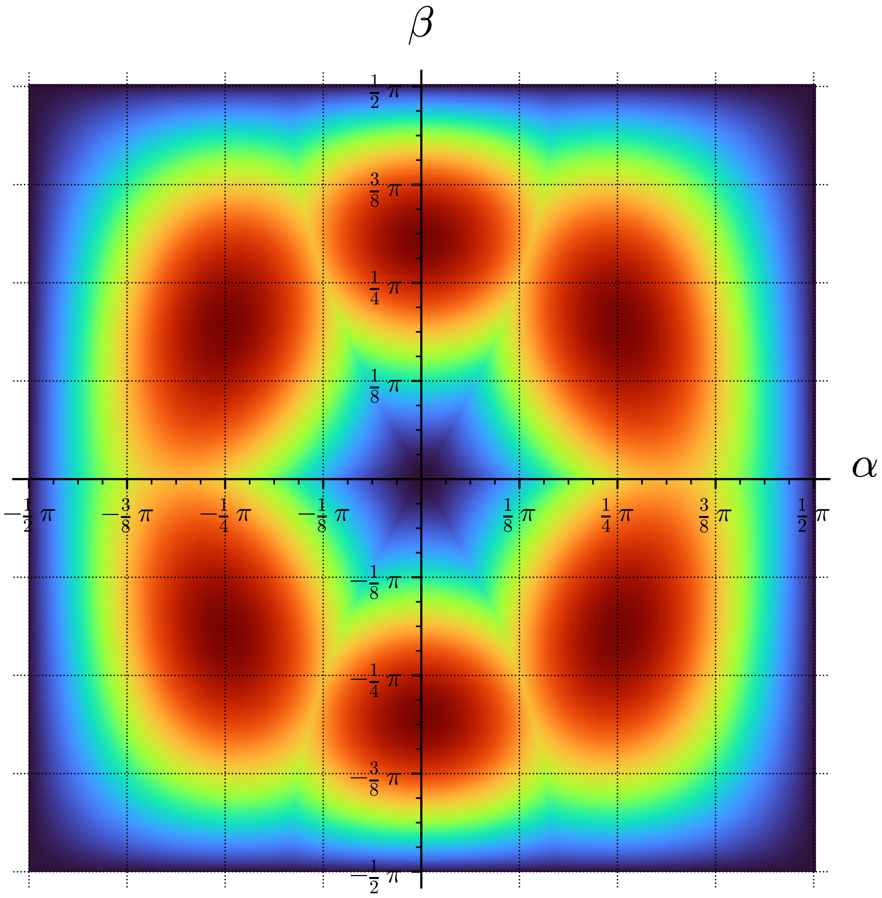


Figure 3.2: Volume of the unconstrained force polytope $V_{\mathcal{F}}$ as a function of cant angle $\alpha \in [0, \pi]$ and dihedral angle $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The surface exhibits six distinct peaks corresponding to optimal actuation configurations.

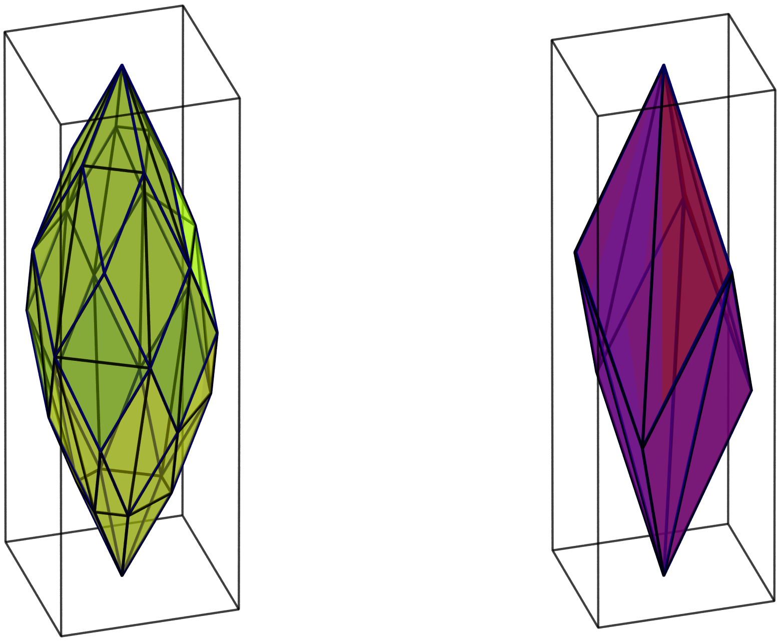
The theoretical bounds on force generation are illustrated in Figure 3.2, which presents the volume $V_{\mathcal{F}}$ of the unconstrained force polytope as a function of the geometric parameters α and β. The surface topology reveals six distinct peaks, indicating configurations that maximize the attainable force space. This visualization provides fundamental insights into the intrinsic actuation authority of the vehicle configuration and serves as a baseline for subsequent constrained analyses.

Comparing the resulting force volume plot of Figure 3.2 with the representation in Figure 2.2a shows the enhanced actuation capabilities of the proposed system. Although visual inspection of Figure 3.2 may not immediately reveal this relationship, examination of the α axis at $\beta = 0$ demonstrates data agreement between the two approaches. This correspondence provides partial validation of the proposed methodology.

Additionally, the dimension given by varying β shows that a significant volume increase can be achieved. In particular, considering the full range of possible $(\alpha, \beta) \in \Gamma_\alpha \times \Gamma_\beta$, the peaks in volume are 6, and their volume is quite higher than the maximum value restricted to $\beta = 0$. This turns out not to be a coincidence: indeed, plotting the same graph for a multirotor with similar mathematical description, but 8 propellers, yields 8 peaks. This is displayed in Figure A.1b, where this observation is immediate to perform.

3.3.1 Validation

In order to validate the theoretical setup, a realistic simulation with Gazebo has been used. This was done by using the physical model for the STH, and fixing it with a joint to a static link, measuring the resulting forces and comparing it to the ones that are expected from the presented results. To do this, the same input that are needed to recreate the full polytope was provided, and the comparison is presented in Figure 3.3a.



(a) Polytope of all the forces attainable by the platform. The discrepancy between symbolic calculation and realistic simulations is so low that the two polytopes are overlapping.
(volume difference <1‰)

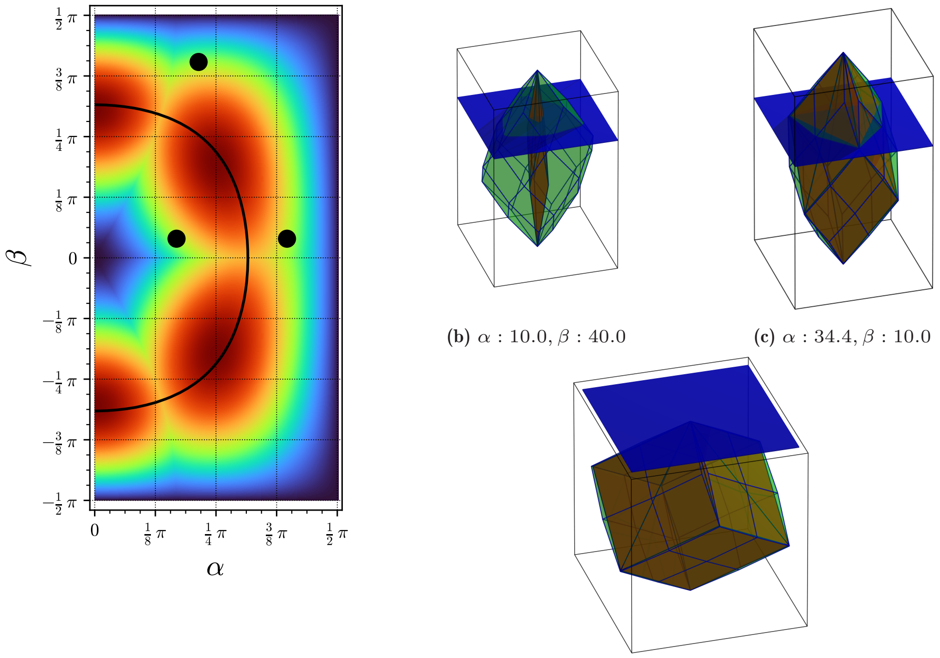
(b) Polytope of attainable forces with no torque generation. The discrepancy is only noticeable from the rendering artifacts that shade the same surface both red and dark blue.
(volume difference ∼1%)

Figure 3.3: Force polytope comparison between theoretical calculations and Gazebo ($\alpha : 20, \beta : 10$)

The polytopes derived from theoretical calculations and Gazebo simulations exhibit such close alignment that their overlap renders visual distinction in the plots nearly impossible. Specifically, when comparing the boundary surfaces of the two polytopes, the vertices and facets are found to coincide to within the visualization resolution. Quantitative analysis confirms this observation, revealing that the volume difference between the theoretical and measured polytopes is less than 1‰, as indicated in Figure 3.3a.

This negligible discrepancy serves to validate the accuracy of the proposed actuation model and the correctness of the theoretical framework developed. The minor deviation observed can be attributed primarily to numerical precision limitations inherent in the simulation environment, as well as to discretization effects in the polytope computation algorithm.

3.4 Zero-moment Control Force Volume



(b) α : 10.0, β : 40.0

(c) α : 34.4, β : 10.0

(a) Figure 3.2 restricted to the I quadrant, highlighting points that correspond to the same value of $V_{\mathcal{F}}$.

(d) α : 57.3, β : 5.73

Figure 3.4: Investigation of $V_{\mathcal{F}}$: (a) value of $V_{\mathcal{F}}$ in function of $(\alpha, \beta) \in \Gamma_\alpha \times \Gamma_\beta$, (b-c) 3D polytopes representing the zero-moment force spaces characterized by the same volume (related to the tilt angles marked in orange in Fig. 3.4a) - the blue plane represents the hovering plane introduced in section 3.5.

We now move on to consider only the volume of forces that do not generate any moment as a side effect: to do this, the procedure explained in subsubsection 1.2.2.3 is followed. In [2], as explained in chapter 2, a particular focus was posed to the matrix resulting from $\mathbf{F}_\alpha \mathbf{B}_\alpha$. This was because it could be expressed in a convenient form, due to simplifications, thus leading to useful reasonings guided by intuition. However, the introduction of the *dihedral angle* β significantly increases the complexity of the resulting matrix $\mathbf{F}_{\alpha,\beta}\mathbf{B}_{\alpha,\beta}$, rendering its closed-form expression impractical. Instead, it is possible to write $\mathbf{B}_{\alpha,\beta}$ with the complex terms substituted by variables. This results in the matrix structure presented in (3.8), which reveals that while a clear structural pattern is discernible in the matrix organization, the practical utility of an explicit analytical representation is severely limited. The terms v_0, v_1, v_2 appearing in the matrix are composed of highly complex expressions involving trigonometric functions of both α and β, whose fully expanded forms require prohibitively large amounts of space to represent. Even after applying extensive symbolic simplifications using computational algebra systems, these expressions remain unwieldy and offer little insight into the underlying physical relationships. Consequently, for the purposes of this analysis, we retain the compact notation with symbolic placeholders.

$$\mathbf{B}_{\alpha,\beta} = \begin{bmatrix} 1 & 0 & v0 \\ 0 & 1 & v0 \\ 0 & v_1 & 1 \\ 1 & v_2 & 0 \\ v_2 & 1 & 0 \\ v_1 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Subsequently, by visualizing the volume data in a manner analogous to Figure 3.2, with the analysis restricted to the domain $\alpha > 0$ (exploiting the inherent symmetry of the configuration to avoid redundant representation), several key observations emerge regarding the influence of the dihedral angle β on the zero-moment force generation capabilities.

Contrary to initial expectations coming from the previous analysis, the introduction of a non-zero dihedral angle β does not enhance the zero-moment force volume. In fact, examination of the data reveals that any deviation from $\beta = 0$ consistently results in a reduction of the achievable force volume. This degradation effect is clearly observable across the entire parameter space investigated, as demonstrated in Figure 3.5. The physical interpretation of this phenomenon lies in the geometric coupling between the cant and dihedral angles: while the cant angle α provides beneficial thrust vectoring for moment-free force generation, the additional outward tilt introduced by β disrupts this optimal configuration, thereby constraining the set of forces that can be generated without producing parasitic moments.

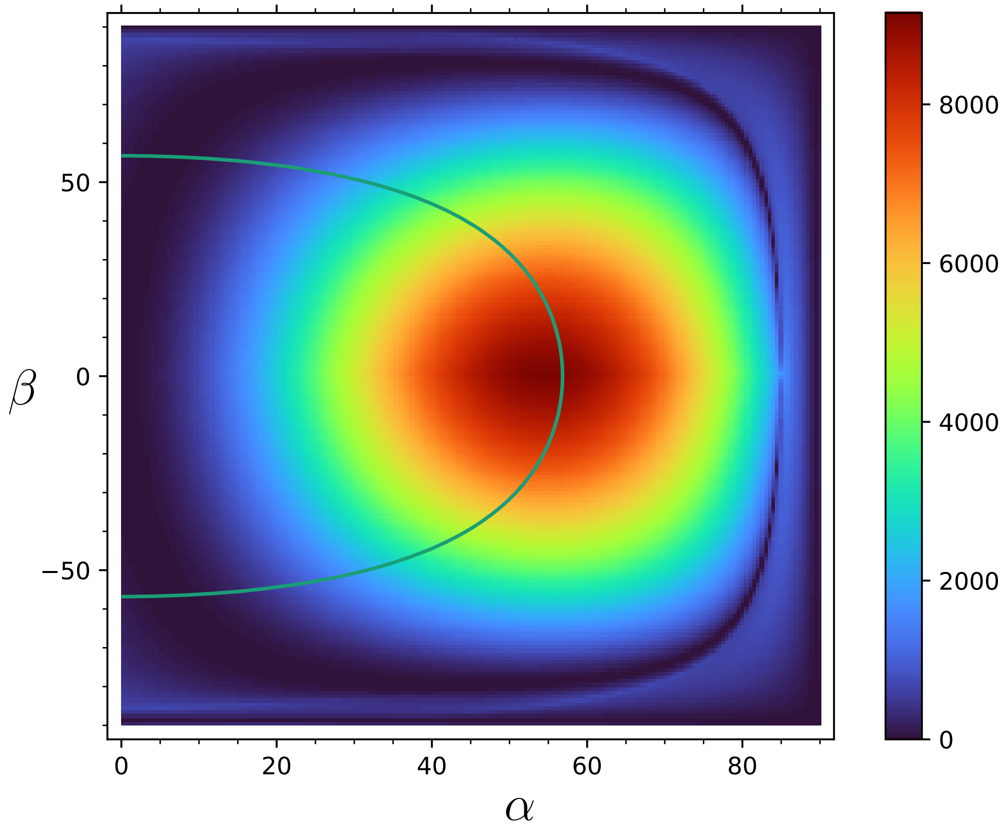


Figure 3.5: $V_{\mathcal{F}_B}$ as a function of α and β ($\alpha \in [0, \pi], \beta \in [-\pi/2, \pi/2]$). The overlaid boundary indicates the region where the zero-moment force polytope is entirely below $\mathbf{f}_c$, meaning that the area outside of it represents configurations where the maximum vertical forces are not sufficient to sustain static hover.

This suggests that the optimal value for the angles seems to be with α between 40° and 70°, with β close to 0. Looking at the analysis done in [2], it is straightforward to see how the gravity is not always counteracted with such high angles.

To aid in understanding this problem, the hovering plane is also displayed together with the polytopes in Figure 3.4b-Figure 3.4d to provide a visual representation of how the control forces interact with the hovering condition. This visualization helps in understanding the spatial relationship between the polytopes and the hovering plane, offering insights into the system's ability to maintain static hovering under different configurations.

3.4.1 Validation

The resulting data was once again validated against the measures obtained from Gazebo, resulting in the visualization presented in Figure 3.3b, which demonstrates the constrained force polytope similarity between theoretical and simulated results.

Despite the absolute error magnitude is significantly larger than in the previous analysis in subsection 3.3.1, the quantitative assessment reveals that the relative volume difference remains approximately 1%, as indicated in the caption of Figure 3.3b. This level of similarity is deemed acceptable for the purposes of model validation, particularly considering the inherent complexity of the constrained force allocation problem and the numerical sensitivities associated with polytope computations involving high-order trigonometric expressions. The validation therefore confirms that the theoretical framework adequately captures the essential characteristics of the zero-moment force generation capabilities, with deviations remaining within reasonable bounds for engineering analysis and design purposes.

3.5 Gravitation Compensation Constraint

In order to address the hovering condition problem, the analysis previously carried out for the α angle in section 2.6 is expanded, repeating it after the addition of the β angle to the model. The resulting plot is shown in in Figure 3.6, and now it will be discussed.

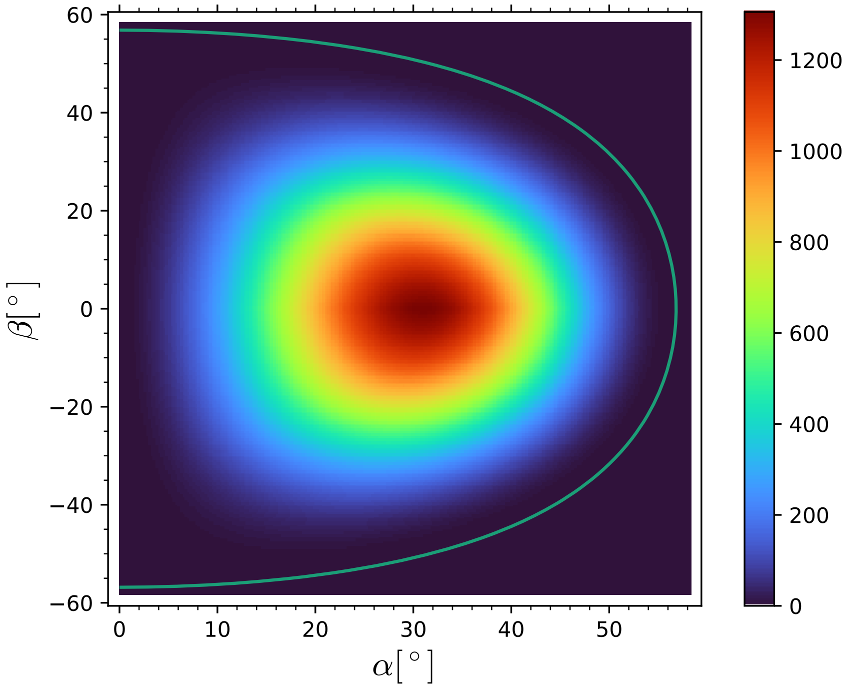


Figure 3.6: Volume $V^h_{\mathcal{F}_B}$ of the zero-moment force polytope constrained above the hovering plane as a function of cant angle $\alpha \in [0, \pi/3]$ and dihedral angle $\beta \in [-\pi/3, \pi/3]$

To obtain this plot, the reasoning is similar to section 2.6: we start by considering the space inside $\mathcal{F}_B$ and above the *hovering plane* $\mathbf{f}_c$. This space is such that all the forces inside of it are able to sustain the UAV's weight, preventing its fall. The boundary where the resulting polyhedron lays all under the hovering plane is displayed with the blue line. All the points (α, β) inside of it represents a configuration in which there is no force that can attain static hovering.

Analysis of the polytope volume restricted to the region above the hovering plane serves to quantify the available control authority of the system. This analysis is conducted under the constraint that gravitational equilibrium must be maintained. The resulting volume metric provides a measure of the force generation capacity that remains accessible while the vehicle maintains static hovering conditions.

Indeed, Figure 3.6 plots the volume of the polytope that is constrained above the hovering plane. Once again, examining Figure 3.4 can aid in understanding this concept. The results align with physical intuition. The volume exhibits positive values only within the hovering boundary region. Furthermore, it decreases progressively as configurations approach this boundary. Notably, the volume reaches relatively low values even for configurations that are not in close proximity to the boundary limit. This characteristic is particularly pronounced for low values of the *cant angle* α.

The visualization reveals an important finding regarding the influence of the *dihedral angle* β on the achievable volume. Specifically, the introduction of a non-zero β angle has a markedly detrimental effect on the resulting polytope volume. Even relatively modest values of β, approximately $20°$, result in a reduction of the volume by half at the α value corresponding to the maximum. This substantial degradation underscores the sensitivity of the zero-moment force generation capability to variations in the dihedral angle parameter.

Having established the fundamental characteristics of the zero-moment force polytope and its interaction with the hovering constraint, we now proceed to investigate an emerging capability enabled by the STH configuration. This new capability concerns the ability to maintain static hovering equilibrium while the vehicle assumes a non-zero attitude with respect to the gravitational field. Such a phenomenon is not achievable with conventional planar multirotor configurations. The underlying physical principle is straightforward. The proposed configuration can now generate control forces that extend beyond the vertical axis z of the UAV body frame. In particular, lateral force components become available as part of the control authority.

This capability represents a significant extension of the operational envelope compared to traditional multirotor platforms. It warrants further systematic analysis to quantify the range of feasible tilted hovering states as functions of the geometric parameters α and β. The following section addresses this investigation.

3.6 Tilted static hovering analysis

A shift in perspective reveals that, since the multirotor system can generate force vectors with both vertical and horizontal components, static hovering can occur while the vehicle is tilted. This section presents an analysis of the maximum tilt angle that can be achieved for various system configurations.

To do this, the hovering constraint is not $\mathbf{f}_c^z \geq mg$ anymore, but we require the total force $\mathbf{f}_c$ have its magnitude $\|\mathbf{f}_c\| > mg$. This changes the geometrical intuition of the *hovering plane* from a plane to a spherical cap, as represented in Figure 3.7.

The volume of the polytope above this new constraint is then displayed in Figure 3.8. In this figure, the role of the β angle is more interesting. By increasing it, in fact, the maximum tilt angle increases when α is less than $35°$. This fact clearly shows, once again, how the choice of the β angle is very dependent on the use case. In the context of this application, in fact, β values up to $20°$ can be beneficial to improve the result if α has values smaller than the optimal one.

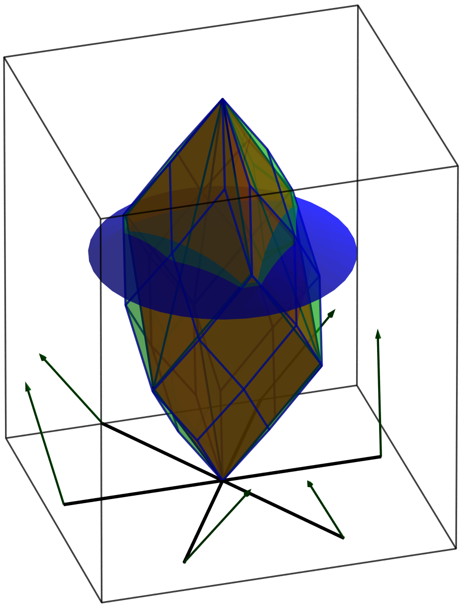


Figure 3.7: 3D polytope representing the zero-moment force space with spherical hovering constraint (dome). Configuration: $\alpha = 34.4°$, $\beta = 10°$. The blue dome represents the spherical constraint $\|\mathbf{f}_c\| \geq mg$ for tilted hovering capability.

3.7 Discussion

Having established the fundamental characteristics of the force polytopes and their dependence on both geometric parameters α and β through systematic analysis, we now synthesize these findings to derive comprehensive design guidelines for the STH configuration. We then assess design recommendations based on practical trade-offs in order to aid the hexarotor design.

Maximum unconstrained forces As visible in Figure 3.2, the newly introduced β angle can lead to the possibility of significantly increasing the volume of body-frame force capabilities. This in general suggests that the new parameter can lead to improved controlling possibilities. Care needs to be taken, since increasing this angle also reduces general efficiency, since higher input values are required to achieve the same vertical force.

Maximum constrained forces When constraining to zero-moment forces, Figure 3.5 shows in fact a decreasing trend as β increases. This implies that configurations with nonzero dihedral angles should be avoided in applications where zero-torque force generation is critical, such as physical interaction tasks or lateral trajectory tracking.

Maximum hovering volume and angle From Figure 3.6, the maximum orientation range that preserves hovering becomes narrower with larger β. The decline confirms that dihedral tilt limits the angular workspace where stable hovering is possible. This effect imposes strict limits on the allowable β when extensive force-related disturbance rejection or agile maneuvers are needed.

3.7.1 Final design guidelines

The combined results suggest the optimal motor alignment lies in the region $\alpha \in [25°, 40°]$, $\beta = 0°$ for most aerial manipulation and interaction applications. The introduction of $\beta > 0$ should only be considered if fault tolerance or rotor visibility

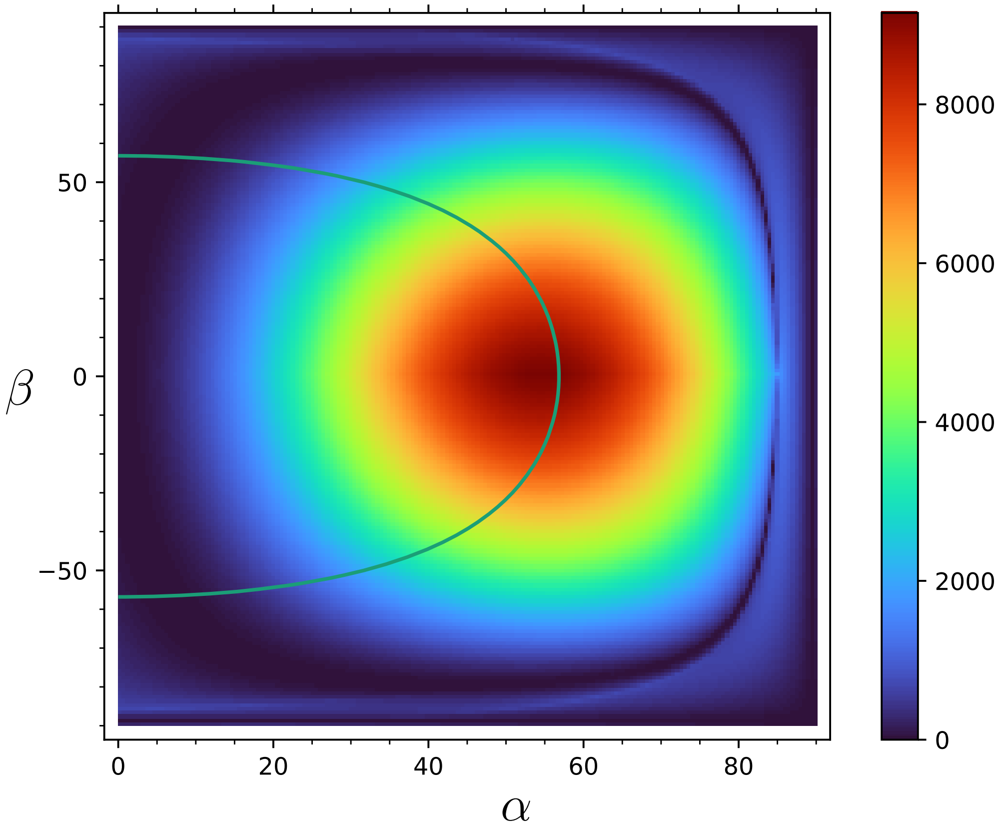


Figure 3.8: Maximum hovering tilt angle as a function of cant angle $\alpha \in [0, \pi]$ and dihedral angle $\beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The surface shows the maximum vehicle tilt that maintains static hovering equilibrium under the spherical constraint $\|\mathbf{f}_c\| \geq mg$.

are priorities. For instance, a small β may ease wiring or provide better ducted fan integration.

3.7.2 Performance trade-off

The added β angle slightly increases design and building complexity, risking the increase of aerodynamic coupling between propellers. However, this comes at a notable cost that is not being fault-tolerant to even one propeller, and missing the chance to greatly increase the general force polytope when the generated torques are not of interest. From a control perspective, canted-only configurations remain preferable due to higher available force in lateral directions and wider orientation workspace.

3.7.3 Conclusion

In summary, while dual-angle tilting offers some mechanical robustness enchantment, our results show no compelling benefit from aerodynamic or control performance perspectives. Therefore, unless specific mechanical or safety constraints exist, favoring $\beta = 0$ (or very small) remains the most effective design decision for general-purpose fully-actuated platforms.

Part II

Control: Trajectory tracking control

Chapter 4

Classical Control Methods

Contents

4.1 Introduction

The design of effective controllers for trajectory tracking and stabilization represents a fundamental challenge in multi-variable control systems, requiring careful consideration of system dynamics, actuation constraints, and performance objectives.

Classical linear control approaches, rooted in well-established theoretical frameworks, offer robust stability guarantees and straightforward implementation through techniques such as PID control, LQR design, and loop shaping methods. These methods typically rely on linearization around operating points and provide predictable behavior within their region of validity. In contrast, nonlinear control strategies directly account for the full system dynamics without linearization assumptions, potentially achieving superior performance through techniques that better exploit the geometric structure of the problem and handle actuation limitations more effectively [9].

This chapter investigates two distinct control strategies to address the trajectory tracking and hovering stabilization problem for fully-actuated aerial systems: one based on classical control principles and another employing nonlinear control techniques. The control objectives require achieving precise position tracking while maintaining prescribed orientations, with particular emphasis on the hovering condition where the system must stabilize at a fixed position with null linear and angular velocities [3]. The comparison between these approaches reveals fundamental trade-offs between control authority, computational complexity, robustness to model uncertainties, and actuation efficiency. These considerations become particularly relevant when independent force and moment generation introduces spurious thrust components that affect overall system performance and energy consumption. The fully-actuated STH serves as the experimental platform for implementing and validating these control methodologies, providing the decoupled actuation capabilities that can be exploited for this purpose.

The two distinct control architectures we propose are the following: a DF-based controller (FC), which serves as a reference benchmark, and a hierarchical controller (HC). While DF approaches have been extensively studied in the literature, existing work has primarily addressed standard collinear UAVs [10], [11]. For such platforms, the position and yaw angle are typically selected as flat outputs, with an internal attitude controller designed to regulate roll and pitch angles. By contrast, we exploit the full actuation capabilities of STHs to formulate the DF problem using complete pose as flat output, yielding a control scheme that encompasses both position and orientation. The second controller extends the hierarchical nonlinear hovering regulator from [5], [12], adapting it beyond static stabilization to address simultaneous trajectory tracking and attitude regulation. The architecture incorporates trajectory-derived feedforward terms, enabling zero steady-state tracking error for position trajectories and constant attitude references.

To evaluate and compare the capabilities of both control strategies, we conduct comprehensive numerical validation through MATLAB-Simulink simulations. The simulation framework accounts for realistic operating constraints and environmental conditions, systematically evaluating controller performance across multiple scenarios. This numerical investigation encompasses both ideal flight conditions, which serve to establish baseline performance characteristics, and realistic scenarios that incorporate representative wind disturbance models with varying intensity and temporal characteristics. The comparative analysis provides quantitative insights into the distinct robustness properties exhibited by the two control architectures, particularly regarding their capacity to maintain acceptable tracking performance and stability margins when subjected to significant external perturbations. Furthermore, the simulation results illuminate the fundamental trade-offs between tracking accuracy, actuator utilization, and disturbance rejection capabilities inherent to each control approach, thereby informing controller selection for specific operational requirements and environmental contexts.

4.2 Trajectory Tracking with Attitude Regulation

The STH model of Part I is further studied, shifting perspective from the modeling problem to the control one. Since STHs are fully actuated platforms, their control can be straightforward. However, we investigate novel control strategies by adapting established nonlinear control methods, previously employed for other systems, alongside innovative hierarchical approaches specifically tailored for these platforms.

Starting by considering the Trajectory tracking problem defined in Problem 1.2.1, we will now tailor it to this platform, since the reference that we want to track is now represented by position and attitude over time.

To define it, we recall the basic allocation matrices, that maps the STH input to the control inputs, from (1.8), reported stacked for convenience in (4.1).

$$\begin{bmatrix} \mathbf{f}_c \\ \boldsymbol{\tau}_c \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \mathbf{u} \tag{4.1}$$

with $\mathbf{f}_c, \boldsymbol{\tau}_c \in \mathbb{R}^3$ are the control force and the control moment in the body frame $\mathscr{F}_B$.

Remark 4.2.1

> *The control input matrices depend on the tilt angles. The condition* $(\alpha, \beta) \neq (0, 0)$ *guarantees the full rank property for* $\mathbf{F}$, $\mathbf{M}$, *and for the matrix* $\left[\mathbf{F}^\top \; \mathbf{M}^\top\right]^\top \in \mathbb{R}^{6\times 6}$, *ensuring the full actuation of the UAV.*

Then the full dynamical model introduced in subsection 1.2.2 is used, representing the full dynamical model for the STH that we are interested in.

Remark 4.2.2 (*Attitude representation*)

> *Throughout this chapter, the STH orientation is primarily represented using the quaternion formulation introduced in subsubsection 1.2.2.2. For any quaternion* $\mathbf{q} \in \mathbb{S}^3$, *we denote its scalar part by* $\eta \in \mathbb{R}$ *and its vector part by* $\boldsymbol{\epsilon} \in \mathbb{R}^3$.
>
> *We also employ rotation matrices and Euler angles where appropriate. The notation* $\mathbf{R}(\mathbf{q}) \in SO(3)$ *denotes the rotation matrix corresponding to quaternion* $\mathbf{q}$. *When using Euler angles* $\boldsymbol{\delta} = \left[\phi \; \theta \; \psi\right]^\top$, *the rotation matrix* $\mathbf{R}(\mathbf{q})$ *is expressed according to the ZYX (yaw-pitch-roll) convention.*

And finally we are able to formalize the new trajectory tracking problem, in the scenario that is specific for STHs.

Problem 4.2.1 (*Trajectory Tracking with concurrent Attitude Regulation*)

> *Design a control solution that concurrently guarantees zero tracking error for sufficiently*

smooth references of position $\mathbf{p}_r \in \mathbb{R}^3$ *and attitude* $\mathbf{q}_r \in \mathbb{S}^3$ *(or equivalently* $\boldsymbol{\delta}_r \in (\mathbb{S}^1)^3$ *so that* $\mathbf{R}(\mathbf{q}_r) = \mathbf{R}(\boldsymbol{\delta}_r)$*).*

4.3 Flatness-based Controller

The first controller build upon the fact that, as we will show hereafter, any STH turns out to be a differentially flat system given a suitable choice of the state, input, and output vectors (subsection 4.3.1). Therefore, we describe the Flatness-based Controller (FC) structure which exploits such property and is characterized by an ad-hoc feedback action (subsection 4.3.2).

4.3.1 STH Differential Flatness (DF)

Recalling Definition 1.2.1 of a *differentially flat* system in subsection 1.2.4, we now proceed to assess the DF of STHs. We select the state vector to be $\mathbf{x} = \left[\mathbf{p}^\top \; \mathbf{v}^\top \; \boldsymbol{\delta}^\top \; \boldsymbol{\omega}^\top\right]^\top \in \mathbb{R}^{12}$, so that (1.12) can be rewritten as the linear state-space system

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}\boldsymbol{\mu} - \mathbf{g} \tag{4.2}$$

In (4.2), by denoting with $\mathbf{0}$ and $\mathbf{0}_3$ the zero (column) vector in $\mathbb{R}^3$ and the zero matrix in $\mathbb{R}^{3\times3}$, the matrices $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{12\times12}$ and $\mathbf{B} \in \mathbb{R}^{12\times6}$ and the vector $\mathbf{g} \in \mathbb{R}^{12}$ are defined as

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{W}(\boldsymbol{\delta})^{-1} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \frac{1}{m}\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{J}^{-1} \end{bmatrix} \quad \mathbf{g} = g\begin{bmatrix} \mathbf{0} \\ \mathbf{e}_3 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{4.3}$$

with $\mathbf{W}(\boldsymbol{\delta}) \in \mathbb{R}^{3\times3}$ depending on the UAV attitude represented thought the Euler angles conventions and defined in (1.11). The input vector $\boldsymbol{\mu}$ in (4.2) (*flat input*) is related to the input vector $\mathbf{u}$ in (1.8) (*dynamics input*). In particular, it is

$$\boldsymbol{\mu} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{0}_3 \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{R}(\boldsymbol{\delta}) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \mathbf{u}. \tag{4.4}$$

Finally, we define the (flat) output vector by accounting for the position and the orientation of the STH, namely, we select $\mathbf{y} = [\mathbf{y}_{\boldsymbol{p}}, \; \mathbf{y}_{\boldsymbol{\delta}}] = \left[\mathbf{p}^\top \; \boldsymbol{\delta}^\top\right]^\top \in \mathbb{R}^3 \times (\mathbb{S}^1)^3$. With this choice, we have that

$$\mathbf{y} = \mathbf{C}\mathbf{x} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \end{bmatrix} \mathbf{x}. \tag{4.5}$$

Exploiting (4.2), (4.4), and (4.5), one can verify that the STH is a differentially flat system since it holds that

$$\mathbf{x} = g_x(\mathbf{y}, \dot{\mathbf{y}}) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{W}(\boldsymbol{\delta})^{-1} \end{bmatrix} \dot{\mathbf{y}}, \tag{4.6}$$

and $\boldsymbol{\mu} = g_\mu(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$ because of (4.4) and given that the input vector $\mathbf{u}$ can in turn be expressed as $\mathbf{u} = g_u(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}})$, i.e.

$$\mathbf{u} = g_u(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix}^{-1} \begin{bmatrix} m\mathbf{R}(\boldsymbol{\delta})^\top(\ddot{\mathbf{p}} + g\mathbf{e}_3) \\ \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix}. \tag{4.7}$$

Thus, it results $\boldsymbol{\mu} = g_\mu(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}) = f(g_x(\mathbf{y}, \dot{\mathbf{y}}), g_u(\mathbf{y}, \dot{\mathbf{y}}, \ddot{\mathbf{y}}))$, satisfying the requirements needed to show the system DF, that is being able to calculate the flat inputs and the system's state by mean of the flat outputs and (a finite number of) their derivatives.

4.3.2 Controller Architecture

The FC block scheme is depicted in Figure 4.1, highlighting the DF transformation that generates the reference *flat input* and state reference trajectory $\mathbf{x}_r$. The controller then results in a simple static state feedback controller with a cascaded transformation, acting on the STH dynamics.

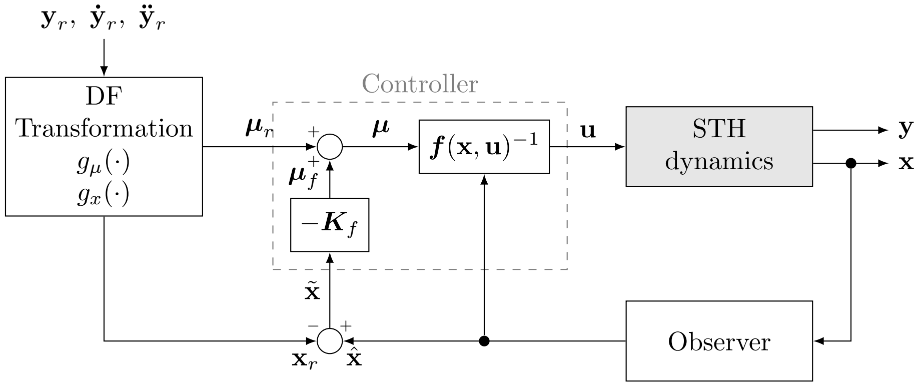


Figure 4.1: Flatness-based Controller (FC) architecture

The controller receives as input the reference signals $\mathbf{y}_r = \left[\mathbf{p}_r^\top\ \boldsymbol{\delta}_r^\top\right]^\top \in \mathbb{R}^6$, along with their first and second time derivatives. Note that the reference position and attitude must be at least twice continuously differentiable (C^2 functions) to ensure smooth control trajectories.

The *DF Transformation* block processes these reference signals to compute two key quantities: the reference state trajectory $\mathbf{x}_r \in \mathbb{R}^{12}$ and the reference *flat input* $\boldsymbol{\mu}_r \in \mathbb{R}^6$. This computation leverages the DF property through the function $g_\mu(\cdot)$ previously introduced in (4.7).

Subsequently, the reference state trajectory $\mathbf{x}_r$ is utilized to generate the feedback control component $\boldsymbol{\mu}_f \in \mathbb{R}^6$, which accounts for deviations from the desired trajectory.

Formally, it is $\boldsymbol{\mu}_f = -\mathbf{K}_f(\mathbf{x} - \mathbf{x}_r)$, where the gain matrix $\mathbf{K}_f \in \mathbb{R}^{6\times12}$ is computed by adopting the LQR approach on system (4.2). The feedback contribution $\boldsymbol{\mu}_f$ is then combined with the feedforward term $\boldsymbol{\mu}_r$ to obtain the total flat input $\boldsymbol{\mu} = \boldsymbol{\mu}_r + \boldsymbol{\mu}_f$. Subsequently, this flat input vector is converted to the dynamics input vector $\mathbf{u}$ by inverting the relation in (4.4). This inversion procedure requires full state feedback, which in practical implementations necessitates the presence of a state observer.

4.4 Hierarchical Controller

The proposed HC constitutes an extension of the nonlinear control approach described in [5], [12]. Having established the platform kinematics and dynamics in subsection 1.2.2, we now leverage the specific structural properties of the STH to develop this hierarchical control strategy. The controller design relies on a fundamental algebraic condition that must be satisfied by the control input allocation matrices: namely, the rank condition $\operatorname{rank}(\mathbf{M}\bar{\mathbf{F}}) = 3$ must hold, where $\bar{\mathbf{F}} \in \mathbb{R}^{6\times3}$ is defined such that $\operatorname{Im}(\bar{\mathbf{F}}) = \ker(\mathbf{F})$. This structural requirement guarantees the existence of a matrix $\mathbf{H} \in \mathbb{R}^{6\times6}$ with the property that $\mathbf{M}\mathbf{H}\mathbf{M}^\top$ is invertible and satisfies the null space condition $\mathbf{F}\mathbf{M}_\mathbf{H}^\dagger = \mathbf{0}_3$. Here, the term $\mathbf{M}_\mathbf{H}^\dagger = \mathbf{H}\mathbf{M}^\top(\mathbf{M}\mathbf{H}\mathbf{M}^\top)^{-1} \in \mathbb{R}^{6\times3}$ denotes the generalized right pseudo-inverse of $\mathbf{M}$ weighted by the matrix $\mathbf{H}$. This mathematical construction enables the decoupling of force and moment generation, which forms the theoretical foundation for the hierarchical control architecture.

The HC input is then designed as

$$\mathbf{u} = \mathbf{M}_\mathbf{H}^\dagger \boldsymbol{\tau}_r + \bar{\mathbf{u}} f_c \tag{4.8}$$

where $\boldsymbol{\tau}_r \in \mathbb{R}^3$ is the reference moment and $f_c = \|\mathbf{F}_c\| \in \mathbb{R}_{\geq 0}$. The vector $\bar{\mathbf{u}} \in \mathbb{R}^6$ is selected in $\ker(\mathbf{M})$ so that the product $\mathbf{F}\bar{\mathbf{u}}$ identifies a direction in the force space $\mathrm{Im}(\mathbf{F}) \cap \mathbb{S}^2$, referred to as *zero-moment preferential direction* $\mathbf{d}_*$, along which the control force can be independently assigned with respect to the control moment. Note that, based on (1.8), the input (4.8) implies $\mathbf{F}_c = \mathbf{F}\mathbf{u} = \mathbf{d}_* f_c$ and $\boldsymbol{\tau}_c = \mathbf{M}\mathbf{u} = \boldsymbol{\tau}_r$.

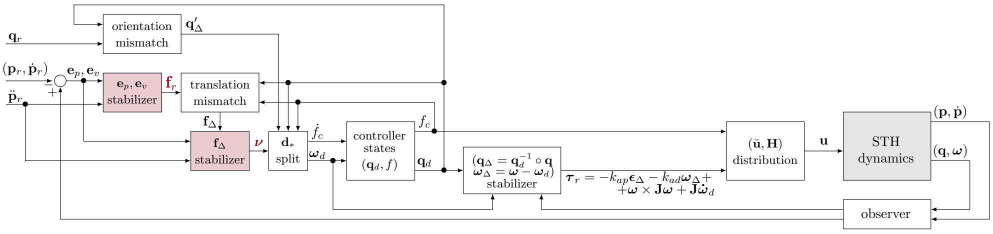


Figure 4.2: Hierarchical Controller (HC) architecture

To address Problem 4.2.1, we select $\bar{\mathbf{u}}$ so that the resulting zero-moment preferential direction corresponds to

$$\mathbf{d}_* = \mathbf{d}/\|\mathbf{d}\|, \quad \mathbf{d} = \mathbf{R}\left(\mathbf{q}_r\right)^\top \left(mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r\right), \tag{4.9}$$

entailing that the resulting control force $\mathbf{F}_c = \mathbf{d}_* f_c$ is oriented in $\mathscr{F}_B$, as to counterbalance the gravity force while acting along the direction of the reference position trajectory. The HC structure is reported in Figure 4.2. The controller architecture introduces two internal state variables: the control force intensity $\|\mathbf{F}_c\| = f_c \in \mathbb{R}_{\geq 0}$ and the *desired orientation* $\mathbf{q}_d \in \mathbb{S}^3$. The desired orientation represents the rotation that drives the *force mismatch* vector $\mathbf{F}_\Delta \in \mathbb{R}^3$ to zero, where this mismatch is defined as the difference between the *desired control force* $\mathbf{R}\left(\mathbf{q}_d\right)\mathbf{d}_* f_c$ and the *reference force* $\mathbf{F}_r \in \mathbb{R}^3$, both expressed in the world frame $\mathscr{F}_W$.

The design of the reference force $\mathbf{F}_r$ represents a key modification with respect to the original hovering regulator presented in [5]. While the original formulation addressed static stabilization, the present extension accommodates dynamic trajectory tracking requirements. To achieve precise tracking of the reference position profile while simultaneously compensating for gravitational effects, the reference force is formulated as

$$\mathbf{F}_r = mg\mathbf{e}_3 + m\ddot{\mathbf{p}}_r - k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v, \tag{4.10}$$

where $\mathbf{e}_p = \mathbf{p} - \mathbf{p}_r \in \mathbb{R}^3$ and $\mathbf{e}_{\dot{p}} = \dot{\mathbf{p}} - \dot{\mathbf{p}}_r \in \mathbb{R}^3$ are the position and velocity error vectors, and $k_{pp}, k_{pd} \geq 0$ are the corresponding gains. As proven in [5], zeroing the force mismatch vector $\mathbf{F}_\Delta$ implies the stabilization of the STH translational dynamics. This is possible by imposing

$$\boldsymbol{\omega}_d = \boldsymbol{\omega}_d^0 + \boldsymbol{\omega}'_d = \frac{1}{f}\left[\mathbf{d}_*\right]_\times \mathbf{R}\left(\mathbf{q}_d\right)^\top \boldsymbol{\nu} - k_q \mathbf{d}_* \mathbf{d}_*^\top \boldsymbol{\epsilon}'_\Delta \tag{4.11a}$$

$$\dot{f}_c = \left(\mathbf{R}\left(\mathbf{q}_d\right)\mathbf{d}_*\right)^\top \boldsymbol{\nu} \tag{4.11b}$$

where $\boldsymbol{\omega}_d \in \mathbb{R}^3$ fulfills a twofold action, regulated by the gain $k_q \geq 0$: by means of $\boldsymbol{\omega}_d^0 \in \mathbb{R}^3$, it accommodates the UAV orientation along the desired position trajectory; while, through $\boldsymbol{\omega}'_d \in \mathbb{R}^3$, it regulates the dynamics of the controller state $\mathbf{q}_d$ towards $\mathbf{q}_r$ by acting on an orientation mismatch term $\mathbf{q}'_\Delta \in \mathbb{S}^3$, $\mathbf{q}'_\Delta = \left[\eta'_\Delta \; \boldsymbol{\epsilon}'_\Delta\right]^\top = \mathbf{q}_r^{-1} \circ \mathbf{q}_d$ between the reference and the desired orientations. The vector $\boldsymbol{\nu} \in \mathbb{R}^3$ in (4.11) constitutes an additional virtual input of the controller that, differently from [5] and consequently to (4.10), we set as

$$\boldsymbol{\nu} = \frac{k_{pd}k_{pp}}{m}\mathbf{e}_p + \left(\frac{k_{pd}^2}{m} - k_{pp}\right)\mathbf{e}_v - \left(\frac{k_{pd}}{m} + k_\Delta\right)\mathbf{F}_\Delta - m\,\ddot{\mathbf{p}}_r, \tag{4.12}$$

with $k_\Delta \geq$ being an additional gain.

Remark 4.4.1

Assuming that $\mathbf{q}$ *has converged to* $\mathbf{q}_r$*, the choices* (4.11) *and* (4.12) *ensure that* $\dot{\mathbf{F}}_\Delta = -k_\Delta \mathbf{F}_\Delta$*. Then, also the force mismatch converges to zero, and the position tracking is fulfilled. Indeed, recalling that* $\mathbf{F}_\Delta = \mathbf{R}(\mathbf{q}_d)\,\mathbf{d}_* f_c - \mathbf{F}_r$ *with* $\mathbf{F}_r$ *as in* (4.10)*, it is possible to verify that*

$$\begin{aligned} \dot{\mathbf{F}}_\Delta &= \mathbf{R}(\mathbf{q}_d)\,\mathbf{d}_* \dot{f}_c + \dot{\mathbf{R}}(\mathbf{q}_d)\,\mathbf{d}_* f - \dot{\mathbf{F}}_r && (4.13a) \\ &= \dot{\mathbf{F}}_{\Delta,1} + \dot{\mathbf{F}}_{\Delta,2} + \dot{\mathbf{F}}_{\Delta,3} && (4.13b) \end{aligned}$$

with

$$\begin{aligned} \dot{\mathbf{F}}_{\Delta,1} &= \mathbf{R}(\mathbf{q}_d)\,\mathbf{d}_* \mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} && (4.14a) \\ \dot{\mathbf{F}}_{\Delta,2} &= \boldsymbol{\nu} - \mathbf{R}(\mathbf{q}_d)\,\mathbf{d}_* \mathbf{d}_*^\top \mathbf{R}(\mathbf{q}_d)^\top \boldsymbol{\nu} && (4.14b) \\ \dot{\mathbf{F}}_{\Delta,3} &= k_{pp}\mathbf{e}_v + \frac{k_{pd}}{m}\left(-k_{pp}\mathbf{e}_p - k_{pd}\mathbf{e}_v + \mathbf{F}_\Delta\right) + m\,\dddot{\mathbf{p}}_r && (4.14c) \end{aligned}$$

The HC architecture in Figure 4.2 is completed by an appropriate selection of the reference control moment $\boldsymbol{\tau}_r \in \mathbb{R}^3$ that ensures the zeroing of $\mathbf{q}_\Delta = \left[\eta_\Delta \ \boldsymbol{\epsilon}_\Delta\right]^\top = \mathbf{q}_d^{-1} \circ \mathbf{q} \in \mathbb{S}^3$. Since the control of the rotational dynamics and the realization of $\boldsymbol{\tau}_r$ are as in [5], we do not report here the detailed derivation. Nonetheless, we point out the following fact.

Remark 4.4.2

The designed controller guarantees the regulation of the UAV orientation towards the desired one (internal controller state), rather than the reference one (external controller input), thus highlighting the lower priority assigned to attitude regulation in the hierarchical architecture.

4.5 Validation

To assess the performance of both the FC and HC solutions, we account for a STH required to track a circular reference path while concurrently adjusting its orientation in 3D space. Specifically, $\mathbf{p}_r$ is designed as a circular trajectory with radius 2 m at a constant altitude of 1 m from the ground and a set of smoothsteps is imposed on the components of $\boldsymbol{\delta}_r$. In detail, motivated by the UAV planar structure on the xy plane, the reference roll and pitch angles ϕ_r and θ_r are defined as a smoothstep sequence of respectively $\left[-7° \ 0° \ 7°\right]$ and $\left[0° \ 3.5° \ 7°\right]$, while the reference yaw angle ψ_r is designed to be more aggressive, being $\left[90° \ 135° \ \ldots \ 270°\right]$.

The initial conditions for such a task are those of ground parking (i.e., with zero pose and velocities), therefore the first phase in the proposed scenario involves a take-off action.

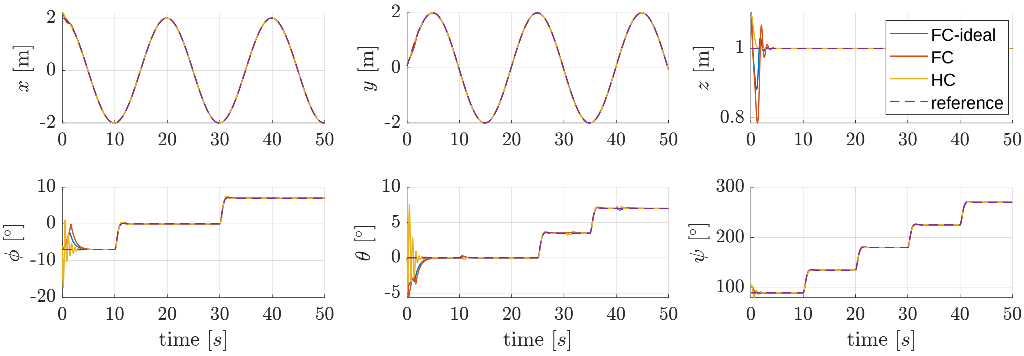


Figure 4.3: Scenario A: **unwindy** flight conditions. Position (top row) and attitude (bottom row) behaviors for the three control architectures: FC-ideal, FC, HC.

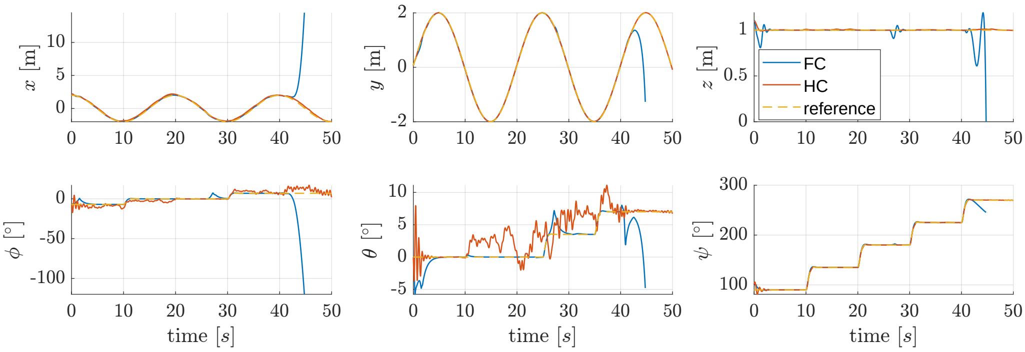


Figure 4.4: Scenario B: **windy** flight conditions. Position (top row) and attitude (bottom row) behaviors for the two control architectures: FC, HC.

The simulation is performed in the MATLAB-Simulink environment by modeling several real-world non-idealities. We consider a STH with diameter of ~0.8 m (propellers included), mass of ~3.5 kg and tilt angles set to $\alpha = 25$ deg and $\beta = 10$ deg (guaranteeing the existence of the matrix $\mathbf{H}$ involved in the definition of the HC input (4.8)). For both controllers, the feedback signals of position and orientation and their derivatives are affected by a time delay $t_f = 12\,\mathrm{m}s$ and additive Gaussian noise with zero mean and variance as illustrated in Table 4.1, mimicking the presence of the observer. Moreover, the UAV state is made available to the controller at lower sampling frequency of 100 Hz, according to the features of a typical IMU sensor; the propellers spinning rates, taking action in the definition of the entries of the vector $\mathbf{u}$, are bounded in $[0,\ 83.5^2]\mathrm{Hz}^2$.

	x component	y component	z component
$\mathbf{p}$ [m^2]	$4.099 * 10^{-7}$	$2.838 * 10^{-7}$	$2.105 * 10^{-8}$
$\boldsymbol{\delta}$ [deg^2]	0.0012	0.0011	0.0011
$\mathbf{v}$ [(m/s)^2]	$2.050 * 10^{-6}$	$1.419 * 10^{-6}$	$1.050 * 10^{-7}$
$\boldsymbol{\omega}$ [(deg/s)^2]	0.0024	0.0022	0.0022

Table 4.1: simulation parameters - noise variance

To comparatively evaluate the control architectures discussed in Sections 4.3-4.4, two different scenarios are taken into account:

A. *unwindy flight conditions* - the considered STH platform is required to fulfill the described task under the given assumptions about the signal delay and observation noise models (but without wind disturbances);

B. *windy flight conditions* - we perform the tests by adding further disturbances induced by the wind action.

As regards the wind action, we consider a shear component and a Dryden turbulence component, both of them directed as the x axis of $\mathscr{F}_W$ and such that the wind speed is equal to 10 m/s at 6 m of altitude. In addition, we also model the occurrence of a wind gust at $T_w = 25$ s in order for its velocity (along the three world frame directions) to be equal to $[2\;2\;1]^\top$ m/s after 2 s. The effect of the whole wind action results in an adverse force $\mathbf{F}_w \in \mathbb{R}^3$ proportional to UAV invested area $A \in \mathbb{R}$. Formally, it is $\mathbf{F}_w = \rho A \mathbf{d}_w$, where $\rho \in \mathbb{R}$ is the (time-varying) air pressure coefficient defined according to the COESA atmosphere model and dependent on the vehicle altitude, and $\mathbf{d}_w \in \mathbb{R}^3$ is the difference between the wind velocity resulting from all its components and the UAV velocity. Then, A is estimated as $A = (1 - d_v)A_\ell + d_v A_u$, where $A_\ell = 0.111\,\mathrm{m}^2$ and $A_u = 0.885\,\mathrm{m}^2$ respectively approximate the lateral and upper area of the considered STH platform and $d_v \in \mathbb{R}$ is computed as $d_v = (\mathbf{d}_w / \|\mathbf{d}_w\|) \cdot \mathbf{z}_B$.

The performance of the FC and HC in scenarios A and B are also compared to an ideal situation, where the FC solution is adopted, and no delay or observation noise is affecting the dynamics. In this case (named *FC-ideal*), the feedback control component has to compensate only for the initial conditions mismatch: this represents almost the best possible solution to the position tracking and attitude regulation problem given the system dynamics and constraints.

For validation, we perform 500 Monte-Carlo (MC) simulations in both scenarios. In the remainder of this section, we discuss the achieved results by showing in figures some representative MC evolutions and by evaluating the following performance indexes in a summary table:

- the *position tracking error norm* $e_p = \|\mathbf{e}_p\| \in \mathbb{R}$;
- the *attitude tracking error* $e_a = 2\arccos(\mathbf{q}^T\mathbf{q}_r) \in \mathbb{R}$, computed as the Riemannian geodesic distance on $\mathbb{S}^3$;
- the *control input norm* $u_n = \|\mathbf{u}\| \in \mathbb{R}$, which provides an intuition on the controller energy consumption;
- the *control input excess* $u_e \in \mathbb{R}$ defined in case of input saturation as the difference between the maximum value among the unbounded entries of $\mathbf{u}$ and the upper limit of $83.5^2\mathrm{Hz}^2$. This gives an insight into the control feasibility.

Figure 4.3 reports the pose trend in a single representative test for scenario A. Under unwindy flight conditions, the FC and the HC perform similarly in position tracking and attitude regulation: both the position and orientation references are followed with very small (if not zero) steady-state errors.

The situation is completely different in presence of wind disturbances: Figure 4.4 reports the position and orientation trends for scenario B. From these results, it can be appreciated how, with respect to the unwindy case, the performance of the HC is almost invariant as regards the accuracy of the position tracking and remains reasonably close to the reference, converging to steady-state zero error, for the attitude regulation. Conversely, the FC solution is not robust in the presence of the wind gust action: both the position and attitude regulation error diverge after just over 42 s.

Table 4.2 reports the mean values of the performance indexes computed on all the MC trails. Examining the second and third columns of Table 4.2, which correspond to controllers FC and HC respectively in scenario A, we observe comparable average performance metrics for pose control tasks. Both architectures achieve similar levels of position tracking accuracy and attitude regulation precision under unperturbed flight conditions.

Regarding energy consumption characteristics, the two control approaches demonstrate equivalent behavior, as evidenced by nearly identical values of the control input norm index u_n. However, a notable distinction emerges when examining the control input excess metric: the HC architecture exhibits a significantly lower mean value of u_e compared to the FC implementation. This reduced saturation level implies that the hierarchical controller maintains greater residual control authority, which can be exploited for rejecting external disturbances and compensating for modeling uncertainties.

Turning our attention to scenario B, wherein wind disturbances are present, we report performance metrics exclusively for the HC architecture, as it represents the only stable solution under these challenging conditions. The results confirm that the hierarchical controller maintains acceptable position tracking accuracy despite the adverse environmental conditions, though this comes at the expense of degraded attitude regulation performance when subjected to wind action. This trade-off between translational and rotational tracking quality aligns with the hierarchical priority structure explicitly designed into the controller, as previously discussed in Remark 4.4.2.

Finally, the fundamental lack of robustness exhibited by the FC architecture in disturbed conditions can be anticipated by examining the control input excess metric u_e. For the flatness-based controller, this index reaches values approximately one order of magnitude higher than those observed for the HC implementation. In contrast, the hierarchical controller maintains u_e values in scenario B that remain nearly identical to those measured in the unperturbed scenario A, indicating consistent saturation behavior regardless of environmental disturbances. This stability in actuator utilization patterns further corroborates the superior robustness properties of the hierarchical control approach.

	FC-ideal	FC-A	HC-A	FC-B	HC-B
e_p [m]	0.010	0.042	0.041	–	0.064
e_a [deg]	0.466	0.556	0.439	–	1.352
u_n [Hz]	4011	4007	4012	–	4012
u_e [Hz]	0.722	2.052	0.118	–	0.120

Table 4.2: controllers performance indexes

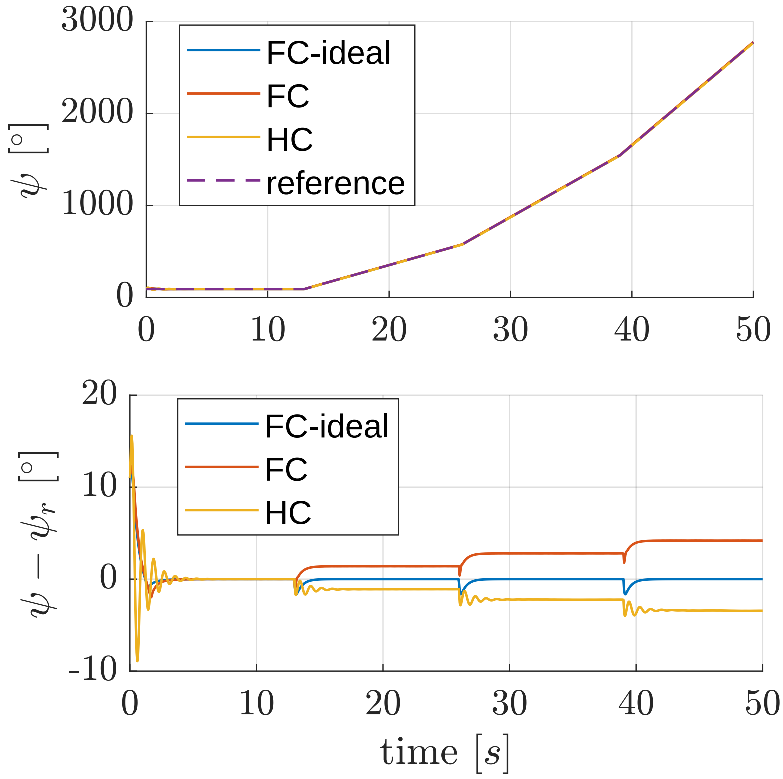


Figure 4.5: Scenario C: **unwindy** flight conditions with angle ramp reference. ψ (top) and $\psi - \psi_r$ (bottom) behaviors for the three control architectures: FC-ideal, FC, HC.

We conclude this analysis by presenting a further unwindy scenario where the attitude is dynamically changing with a ramp reference (scenario C). In detail, we impose constant reference roll and pitch angles while the yaw reference angle is designed to change with ramps of different slopes. Interestingly, this situation translates into a steady-state error in the attitude regulation task, which turns out to be proportional to the ramp slope. This fact is shown in Figure 4.5. We conjecture that this behavior is related to the imposed constant yaw rate, which is not taken into account since the feedback control scheme does not consider an angular rate reference, differently from what is done with the translational dynamics with the velocity reference. Current activity is ongoing to get further insights into this behavior and to devise a suitable control action to compensate for the attitude errors.

4.6 Discussion

In this chapter, we have presented two distinct control architectures for the STH platform, both designed to address the simultaneous trajectory tracking and attitude regulation problem described in Problem 4.2.1. The first solution, referred to as the FC, leverages the differential flatness property inherent to the STH dynamics. This controller synthesizes a feedforward control action that enables concurrent tracking of both position and orientation reference trajectories, augmented with an LQR-based feedback component to enhance robustness against model uncertainties and initial condition errors. The second approach, designated as the HC, implements a nonlinear hierarchical regulation strategy characterized by a cascaded control structure. Within this architecture, the orientation tracking objective is deliberately assigned lower priority compared to the position control objective, reflecting operational scenarios where maintaining spatial trajectory accuracy takes precedence over precise attitude regulation.

The comparative performance evaluation of both control strategies has been conducted through extensive numerical validation in the MATLAB-Simulink environment. The simulation campaign encompasses diverse operational scenarios, including both ideal unperturbed flight conditions and realistic wind-disturbed environments with varying disturbance characteristics. Results from the unperturbed scenarios reveal that the FC architecture achieves exceptional tracking accuracy for both position and attitude references, establishing it as the preferred solution for controlled indoor environments where external disturbances remain negligible. The flatness-based approach fully exploits the platform's kinematic and dynamic properties, as previously characterized in subsection 1.2.2, to generate optimal reference trajectories in the flat output space.

In contrast, the simulation results under wind-disturbed conditions demonstrate markedly different performance characteristics. The proposed HC solution, which extends the static hovering regulator framework originally introduced in [5], successfully maintains acceptable tracking and regulation performance despite significant external perturbations. This robustness stems from the hierarchical structure's ability to preserve residual control authority, as evidenced by the consistently lower control input excess metrics reported in Table 4.2. Unlike the FC approach, which exhibits instability when subjected to substantial wind disturbances, the hierarchical controller demonstrates superior disturbance rejection capabilities, rendering it particularly well-suited for demanding outdoor operational scenarios.

Several promising directions for future investigation emerge from this work. A primary objective involves enhancing the HC architecture to enable complete 6D trajectory tracking capabilities by extending beyond the current attitude regulation framework to accommodate fully dynamic orientation reference profiles. This enhancement would require reformulating the control law to properly account for angular acceleration references, analogous to the treatment of linear acceleration in the current position tracking implementation. Additionally, adaptive strategies should be developed to provide systematic mechanisms for adjusting the trade-off between positioning priority and orientation regulation throughout arbitrary trajectory execution. Such mechanisms could incorporate task-dependent weighting functions or environmental condition monitoring to dynamically reconfigure the hierarchical priority structure, thereby optimizing overall system performance across diverse operational requirements and varying disturbance conditions.

Chapter 5

Contraction Theory-Based Methods

Contents

5.1 Introduction

This chapter introduces contraction theory and its applications to control systems. We begin by establishing the fundamental concepts of contraction analysis, followed by an overview of adaptive control techniques. The subsequent sections present our main contributions: adaptive state-feedback controllers and output-feedback controllers that combine contraction guarantees with parameter adaptation capabilities.

5.1.1 Contraction Theory

Contraction Theory (CT) offers a fundamentally different perspective on stability analysis compared to classical Lyapunov methods. Rather than examining how individual trajectories behave relative to equilibrium points, the theory focuses on the evolution of distances between neighboring trajectories [13], [14]. This shift from absolute to relative stability analysis has proven particularly valuable for understanding complex nonlinear behaviors and designing robust controllers that operate across diverse operating conditions.

The theoretical foundations rest on the concept of contracting metrics— specialized distance measures on the state space that quantify how trajectories separate or converge. When a dynamical system is contracting with respect to such a metric, any pair of trajectories exponentially approach one another, regardless of their initial separation [15], [16]. This property ensures predictable long-term behavior that is independent of initial conditions, providing stronger guarantees than asymptotic stability alone.

The practical implications of this framework are significant: a contracting system possesses at most one equilibrium point (or periodic orbit, in time-varying systems), and all trajectories converge to it exponentially. Moreover, the framework naturally extends to tracking problems, where the goal is to follow time-varying reference trajectories rather than reaching static equilibrium points.

Control Contraction Metrics (CCMs) provide systematic methodology for synthesizing feedback controllers that guarantee exponential convergence to reference trajectories [17], [18]. Unlike traditional Lyapunov-based approaches requiring verification for each specific trajectory, CCM-based controllers offer trajectory-independent guarantees, enabling a single controller to handle arbitrary feasible reference paths. Recent implementations in mobile robotics showcase practical advantages, achieving superior tracking performance without requiring a priori knowledge of target trajectories [19]. Safety-critical applications benefit from explicit, quantifiable bounds on tracking errors, enabling formal verification of safety constraints and guaranteed performance margins [20], [21]. Neural Contraction Metrics (NCMs) leverage deep neural networks to approximate optimal contraction metrics for high-dimensional systems where analytical solutions are intractable [22], [23]. These learned metrics maintain theoretical guarantees while enabling real-time implementation on computationally limited platforms.

Synchronization problems between dynamical systems have been addressed using contraction-based approaches, with applications ranging from networked control to multi-agent robotics [6], [24]. The combination of CT with adaptive control naturally addresses parametric uncertainties while maintaining stability guarantees. Contraction-based L1-adaptive controllers merge fast adaptation with robust stability properties for systems with matched uncertainties [25]. Integration with Gaussian Process regression enables probabilistic uncertainty quantification alongside contraction guarantees [26]. The theoretical developments in contraction analysis extend naturally to systems evolving on non-Euclidean manifolds, where the state space possesses intrinsic geometric structure beyond the flat Euclidean space considered thus far. This extension proves particularly valuable when analyzing rigid body dynamics, where rotations naturally live on the special orthogonal group $SO(3)$, or systems constrained to other curved manifolds [27]. The differential geometric framework of contraction theory adapts seamlessly to these settings, providing stability guarantees that respect the underlying manifold geometry.

Recent theoretical advances have also revealed deep connections between contraction theory and the dynamics of neural networks during learning. By analyzing how network parameters evolve during training through the lens of contraction analysis, researchers can design learning algorithms with provable stability properties [28]. This perspective offers a rigorous mathematical foundation for understanding and controlling the learning process, complementing existing empirical approaches to neural network training.

5.1.1.1 A gentle introduction

While CT's mathematical rigor provides powerful guarantees, its technical presentation can present a barrier to entry for newcomers to the field. This section aims to provide an accessible introduction to the core concepts before proceeding to the formal development.

To understand CT, we first revisit the fundamental stability concepts in control theory. Consider a dynamical system starting from an initial condition $\mathbf{x}_0$ that satisfies

$$\|\mathbf{x}_0\| \leq \delta, \quad \delta \in \mathbb{R}_{>0} \tag{5.1}$$

We can now introduce three fundamental stability concepts in control theory, ordered by increasing strength.

A system is **Lyapunov stable** at an equilibrium point if, for any given $\varepsilon > 0$, there exists a $\delta > 0$ such that trajectories starting within δ of the equilibrium remain within ε of it for all future time. In other words, if the system starts close enough to the equilibrium, it stays bounded within a neighborhood of that point indefinitely.

A system is **Asymptotically stable** if it is Lyapunov stable and, additionally, trajectories starting near the equilibrium actually converge to it as time approaches infinity. This is a stronger condition than Lyapunov stability, as it requires not just boundedness but actual convergence to the equilibrium point.

The strongest stability concept is **Exponential stability**, where the convergence to the equilibrium occurs at an exponential rate. Specifically, the norm of the state decreases exponentially with time, bounded by $\|\mathbf{x}(t)\| \leq \alpha e^{-\lambda t} \|\mathbf{x}_0\|$ for some positive constants α and λ. This provides both convergence guarantees and explicit bounds on the convergence rate, making it particularly valuable for control applications where performance specifications must be met.

These concepts are recapped in the following table:

Stability	Discrete Time	Continuous Time
Lyapunov	$\lVert\mathbf{x}_n\rVert \leq \varepsilon$	$\lVert\mathbf{x}(t)\rVert \leq \varepsilon$
Asymptotic	$\lVert\mathbf{x}_n\rVert \overset{n\to\infty}{\to} 0$	$\lVert\mathbf{x}(t)\rVert \overset{t\to\infty}{\to} 0$
Exponential	$\lVert\mathbf{x}_n\rVert \leq \alpha c^n \lVert\mathbf{x}_0\rVert$	$\lVert\mathbf{x}(t)\rVert \leq \alpha e^{-\lambda t} \lVert\mathbf{x}_0\rVert$

where ε,α,c and λ are positive constants.

Then, a system is said to be strongly contracting if, considering $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ to be two different trajectories of the same dynamical system the following condition holds:

$$\text{Discrete Time} \qquad \left\lVert\mathbf{x}_n^{(1)} - \mathbf{x}_n^{(2)}\right\rVert \leq \rho^n \left\lVert\mathbf{x}_0^{(1)} - \mathbf{x}_0^{(2)}\right\rVert \tag{5.2a}$$

$$\text{Continuous Time} \qquad \left\lVert\mathbf{x}^{(1)}(t) - \mathbf{x}^{(2)}(t)\right\rVert \leq e^{-c(t-t_0)} \left\lVert\mathbf{x}_0^{(1)} - \mathbf{x}_0^{(2)}\right\rVert \tag{5.2b}$$

The first thing to notice is that (5.2a) do not state the convergence of a single trajectory to an equilibrium, but instead the condition regards two different state trajectories converging one to another. Strong contractivity implies exponential stability, and also the existence of a single equilibrium point (provided that the function f of the dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$ is continuous and is time invariant.)

The property of CT to be based on any two different trajectories instead of giving conditions on one trajectory to an equilibrium point, is what makes it so powerful. This is because it makes it straightforward to derive bounds on two trajectories under the effect of external inputs, bounded disturbances or even unmodeled dynamics.

Having established the fundamental intuition, we now proceed to develop the formal mathematical framework of contraction theory.

5.1.1.2 Euclidean Contraction Theory

Building on the intuition from strong contractivity in (5.2a), we now formalize the mathematical framework for Euclidean contraction theory. The key insight is to introduce a metric matrix M that defines how we measure distances between trajectories.

For the Euclidean case, we consider the simplest choice: $M = I$, the identity matrix. This means distances are measured using the standard Euclidean norm:

$$\lVert\delta\mathbf{x}\rVert_M = \sqrt{\delta\mathbf{x}^\top M \delta\mathbf{x}} \overset{M=I}{=} \sqrt{\delta\mathbf{x}^\top \delta\mathbf{x}} = \lVert\delta\mathbf{x}\rVert \tag{5.3}$$

Consider the autonomous system dynamics $\dot{\mathbf{x}} = f(\mathbf{x}, t)$ and examine how the distance between two infinitesimally close trajectories evolves. The virtual displacement $\delta\mathbf{x}$ between neighboring trajectories satisfies:

$$\delta\dot{\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}}\delta\mathbf{x} \tag{5.4}$$

The time derivative of the squared distance metric is:

$$\frac{d}{dt}\left(\delta\mathbf{x}^\top M \delta\mathbf{x}\right) = \delta\mathbf{x}^\top \left(\frac{\partial f^\top}{\partial \mathbf{x}} M + M\frac{\partial f}{\partial \mathbf{x}}\right)\delta\mathbf{x} \tag{5.5}$$

For uniform contraction with $M = I$, the system is contracting if there exists $\lambda > 0$ such that:

$$\frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f^\top}{\partial \mathbf{x}} \preceq -2\lambda I \tag{5.6}$$

for all $\mathbf{x}$ in the state space. This condition ensures that the distance between any two trajectories decreases exponentially at rate λ, directly recovering the exponential bound from (5.2a) [15].

While Euclidean contraction provides the most straightforward analysis, it is often conservative as it does not exploit the geometric structure of the underlying dynamical system [16]. Nevertheless, this framework has proven valuable in networked systems and consensus problems, where the uniform metric naturally aligns with the problem structure [13].

5.1.1.3 State-Dependent Contraction Metrics

Moving beyond the Euclidean case, we now introduce a state-dependent metric matrix $M(\mathbf{x}) = M(\mathbf{x})^\top \succ 0$ that adapts to the local geometry of the state space. This generalization significantly expands the applicability of contraction theory by allowing different distance measures at different points in the state space [13], [14].

The distance between infinitesimally close trajectories is now measured as:

$$\|\delta\mathbf{x}\|_M = \sqrt{\delta\mathbf{x}^\top M(\mathbf{x})\delta\mathbf{x}} \tag{5.7}$$

Following the same approach as in the Euclidean case, we examine the time derivative of the squared distance metric:

$$\frac{d}{dt}\left(\delta\mathbf{x}^\top M(\mathbf{x})\delta\mathbf{x}\right) = \delta\mathbf{x}^\top \left(\dot{M}(\mathbf{x}) + \frac{\partial f}{\partial \mathbf{x}}^\top M(\mathbf{x}) + M(\mathbf{x})\frac{\partial f}{\partial \mathbf{x}}\right)\delta\mathbf{x} \tag{5.8}$$

Note that unlike the Euclidean case, the metric matrix itself now varies with the state, contributing the $\dot{M}(\mathbf{x})$ term.

The system is contracting with respect to $M(\mathbf{x})$ if there exists $\lambda > 0$ such that:

$$\dot{M}(\mathbf{x}) + M(\mathbf{x})\frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{x}}^\top M(\mathbf{x}) \preceq -2\lambda M(\mathbf{x}) \tag{5.9}$$

for all $\mathbf{x}$ in the state space. This condition ensures exponential convergence between trajectories at rate λ, analogous to (5.2a).

The state-dependent metric can be interpreted as defining a Riemannian metric on the state space [14]. This geometric perspective reveals that systems may not exhibit contraction in standard Euclidean coordinates, yet become contracting when viewed through an appropriate coordinate transformation induced by $M(\mathbf{x})$.

Finding suitable state-dependent metrics remains a challenging problem. Traditional approaches employ sum-of-squares programming for polynomial systems [14], while recent developments leverage neural networks to approximate metrics for complex high-dimensional systems, as discussed later in this work. Moreover, additional work present in Appendix B explains and improve numerical methods to estimate the metric for discrete-time autonomous systems.

5.1.2 Adaptive Control

While CT provides powerful trajectory convergence guarantees, real-world systems often exhibit parametric uncertainties that cannot be known a priori. Adaptive control techniques enable online parameter estimation to handle uncertain system parameters while maintaining stability guarantees [29], [30]. Classical approaches include Model Reference Adaptive Control (MRAC) [31] and L1 adaptive control, which balances fast adaptation with robust filtering [32]. Recent work combines adaptive control with CT to preserve exponential trajectory convergence under parametric uncertainties [25], [33].

A key challenge is balancing adaptation speed with robustness to noise and unmodeled dynamics. L1 adaptive control addresses this through separated fast adaptation and robust filtering, providing guaranteed transient performance [32]. Recent developments integrate learning-based methods, using neural networks and Gaussian Processes for uncertainty quantification [34], [35]. The combination of adaptive control with contraction theory enables handling parametric uncertainties while maintaining exponential trajectory convergence [25], [33], as demonstrated in this work.

5.2 Problem Statement

We define a reference state trajectory $\mathbf{z}(t)$, for which the corresponding output trajectory is $\mathbf{y}_z(t)$, and the controlled system trajectory $\mathbf{x}(t)$ with output $\mathbf{y}_x(t)$, as follows:

$$\begin{aligned} \dot{\mathbf{z}} &= f(\mathbf{z},t) + g(\mathbf{z},t)\psi(t) = \varphi(\mathbf{z},t) && (5.10a) \\ \mathbf{y}_z &= h(\mathbf{z}) \\ \dot{\mathbf{x}} &= f(\mathbf{x},t) + g(\mathbf{x},t)\left[\psi(t) + u(\mathbf{x},\mathbf{z},t)\right] && (5.10b) \\ \mathbf{y}_x &= h(\mathbf{x}) \end{aligned}$$

with $u : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ represents the controller, and $\psi : \mathbb{R} \to \mathbb{R}^m$ is the reference trajectory input. The functions f, g, h are the same for both systems. It is then possible to define the error between the reference trajectory and the controlled one as

$$\tilde{\mathbf{x}}(t) = \mathbf{z}(t) - \mathbf{x}(t) \tag{5.11}$$

The goal is to get the error between trajectories goes to zero exponentially, i.e.

$$\|\tilde{\mathbf{x}}(t)\| \le e^{-\lambda t}\,\|\tilde{\mathbf{x}}(t - t_0)\| \tag{5.12}$$

5.2.1 State of the art

The controllers developed in this work build upon the theoretical foundations established in prior research on contraction-based synchronization and control [6, 24]. The key innovation lies in integrating adaptive control strategies, drawing inspiration from the methodology presented in [33], to extend these contraction-based approaches beyond the traditional Control Contraction Metric (CCM) framework. This integration enables the proposed controllers to handle parametric uncertainties while maintaining the exponential convergence guarantees characteristic of contraction theory. Notably, the adaptive framework demonstrates that contraction-based stability analysis remains applicable even when the controller does not explicitly rely on CCMs, thereby broadening the scope of tractable problems within the contraction theory paradigm.

In order to present the achieved results, it is necessary to recall few assumptions that establish the conditions under which the proposed controllers in [6, 24] can be successfully applied. These assumptions will provide the mathematical framework needed to extend them, therefore formulating and proving the main theorems.

First, they fix the controller structure to be the one in (5.13), where u_f stands for the *feedback* part of the input.

$$u(\mathbf{x},\mathbf{z},t) = -\kappa\left[u_f(\mathbf{x},t) - u_f(\mathbf{z},t)\right] \tag{5.13}$$

And then the following assumptions are made.

Assumption 5.2.1

There exists a C^1 Matrix function $P = P^\top : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n\times n} \succ 0$, and parameters $\bar{p}, \underline{p}, \rho, \lambda > 0$ such that the following two conditions hold $\forall \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R}$:

$$\mathscr{L}_\varphi P(\mathbf{x},t) - \rho P(\mathbf{x},t) g(\mathbf{x},t) g^\top(\mathbf{x},t) P(\mathbf{x},t) \preceq -2\lambda P(\mathbf{x},t) \tag{5.14a}$$

$$\underline{p} I \preceq P(\mathbf{x},t) \preceq \bar{p} I \tag{5.14b}$$

In addition to Assumption 5.2.1, a second key assumption is required to establish the structural properties of the control-affine system. Specifically, it is necessary to ensure that the input matrix $g(\mathbf{x},t)$ preserves the geometric structure induced by the metric $P(\mathbf{x},t)$. This requirement can be formalized through the concept of a *Killing vector field*, which guarantees that the effect of control inputs does not distort the metric tensor along the system trajectories. Mathematically, this translates into the following condition: does not influence the norm, and this translates to the following assumption:

Assumption 5.2.2 (*Killing vector field*)

> *g is a* Killing vector field *[36] for P, i.e.:*
>
> $$\mathscr{L}_g P(\mathbf{x}, t) = 0 \quad \forall \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R} \tag{5.15}$$

The final assumption required to complete the theoretical framework is presented here for completeness. This assumption establishes an integrability condition that links the controller structure to the metric $P(\mathbf{x}, t)$. While this assumption plays a crucial role in the state-feedback case, it will be subsequently modified and relaxed when we address the output-feedback scenario in section 5.4, where direct state measurements are not available.

Assumption 5.2.3 (*integrability condition*)

> *The learned controller $u_f(\mathbf{x}, t)$ satisfies the following equation*
>
> $$\frac{\partial u_f}{\partial \mathbf{x}} = g^\top(\mathbf{x}, t) P(\mathbf{x}, t) \quad \forall \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R} \tag{5.16}$$

Summing up their result, under these assumptions it is possible to find a controller that can solve the synchronization problem described by section 5.2.

5.3 Adaptive control

Building on these foundations, this work specifically addresses parametric uncertainties through an adaptive control layer. The proposed approach extends the state-feedback controller from [6] by incorporating online parameter estimation while preserving the exponential convergence guarantees established by contraction theory.

5.3.1 Problem Statement

section 5.2 is then extended to incorporate parametric uncertainties in the system dynamics. Specifically, we consider the case where the system is subject to unknown constant parameters that affect the dynamics in a structured and known manner.

We begin by reformulating the trajectory tracking problem by introducing an additive uncertainty term to both the reference and controlled trajectories. Let $\boldsymbol{\vartheta} \in \mathbb{R}^d$ denote a vector of p unknown constant parameters, and let $\Delta : \mathbb{R}^n \to \mathbb{R}^{n \times d}$ be a known regressor matrix that characterizes how these parameters enter the system dynamics. The extended system formulation, with the so called *matched uncertainty* in literature, is then given by:

$$\begin{aligned}
\dot{\mathbf{z}} &= f(\mathbf{z}, t) + g(\mathbf{z}, t)\psi(t) && - \Delta^\top(\mathbf{x})\boldsymbol{\vartheta} && (5.17a)\\
\mathbf{y}_z &= h(\mathbf{z}) \\
\dot{\mathbf{x}} &= f(\mathbf{x}, t) + g(\mathbf{x}, t)\left[\psi(t) + u(\mathbf{x}, \mathbf{z}, t)\right] && - \Delta^\top(\mathbf{x})\boldsymbol{\vartheta} && (5.17b)\\
\mathbf{y}_x &= h(\mathbf{x})
\end{aligned}$$

The new term $-\Delta^\top(\mathbf{x})\boldsymbol{\vartheta}$ that appears represents the uncertainty term that enters in the system dynamics. This uncertainty is supposed to have a known structure, as formalized in the following assumption:

Assumption 5.3.1

> *There exists a term $\phi(\mathbf{x})$ such that the adaptation term in (5.17) satisfies the following*

equation:

$$\Delta^\top(\mathbf{x})\boldsymbol{\vartheta} = g(\cdot)\phi^\top(\mathbf{x})\boldsymbol{\vartheta} \tag{5.18}$$

To enable the controller to operate effectively under these uncertain conditions, it becomes necessary to appropriately modify Assumption 5.2.1. This modification is required to properly account for the additional uncertainty term that appears in (5.17), ensuring that the contraction-based analysis remains valid in the presence of parametric uncertainties.

Assumption 5.3.2

There exists a C^1 *Matrix function* $P = P^\top : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^{n\times n} \succ 0$, *and parameters* $\bar{p}, \underline{p}, \rho, \lambda > 0$ *such that the following two conditions hold* $\forall \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R}$*:*

$$\begin{aligned}\mathscr{L}_\varphi P(\mathbf{x},t) &- \rho P(\mathbf{x},t)g(\mathbf{x},t)g^\top(\mathbf{x},t)P(\mathbf{x},t)\\ &- P(\mathbf{x},t)g(\mathbf{x},t)\frac{\partial\phi^\top}{\partial\mathbf{x}}(\mathbf{x})\boldsymbol{\vartheta} - \boldsymbol{\vartheta}^\top\frac{\partial\phi}{\partial\mathbf{x}}(\mathbf{x})g^\top(\mathbf{x},t)P(\mathbf{x},t)\\ &\preceq -2\lambda P(\mathbf{x},t)\end{aligned} \tag{5.19a}$$

where

$$\underline{p}I \preceq P(\mathbf{x},t) \preceq \bar{p}I \tag{5.19b}$$

Now, with all the necessary assumptions and preliminaries established, we are in a position to present the main theoretical result of this chapter, which characterizes the adaptive synchronization controller and establishes its convergence properties under parametric uncertainties.

Theorem 5.3.1

Given the system as in (5.17), *and provided that Assumption 5.2.2, Assumption 5.3.2 and Assumption 5.3.1 hold, the controller satisfying Assumption 5.2.3 and Assumption 5.3.1, and defined by*

$$u(\mathbf{x}) = -\kappa\left[u_f(\mathbf{x},t) - u_f(\mathbf{z},t)\right] + \phi(\mathbf{x})^\top\hat{\boldsymbol{\vartheta}} \tag{5.20a}$$

where

$$\dot{\hat{\boldsymbol{\vartheta}}} = -\Gamma\left[\phi(\mathbf{x})g(\cdot)P(\cdot)(\mathbf{x}(t) - \mathbf{z}(t)) + \sigma\boldsymbol{\vartheta}\right] \tag{5.20b}$$

solves the synchronization problem.

In order to proceed with the proof it is useful to point out this remark:

Remark 5.3.1

The following equivalence holds

$$\frac{\partial u}{\partial\mathbf{x}}(\mathbf{x},\mathbf{z},t) \tag{5.21a}$$

$$= -\kappa\frac{\partial u_f}{\partial\mathbf{x}}(\mathbf{x}) + \frac{\partial\phi}{\partial\mathbf{x}}^\top(\mathbf{x})\boldsymbol{\vartheta} \tag{5.21b}$$

$$\overset{(5.16)}{=} -\kappa g^\top(\mathbf{x},t)P(\mathbf{x},t) \tag{5.21c}$$

With the necessary kinematics now fully characterized, we are in a position to present the central theoretical result of this chapter. This theorem establishes the adaptive synchronization controller in its complete form, providing a rigorous characterization of both the control law and the parameter adaptation mechanism. Furthermore, it proves that these components, when working together, ensure exponential convergence properties even in the presence of parametric uncertainties.

Proof 5.3.1

The proof follows a similar course of the main proof in [6]: Define $Z(t)$ and $X(t)$ as solution to the system dynamics equations in (5.10), with initial conditions $\mathbf{z}_0$ and $\mathbf{x}_0$, defined for $t > t_0$, and apply the error equation (5.11), resulting in $\tilde{X}(t) = X(t) - Z(t)$. It is therefore possible to define an "interpolant" on the error as

$$\Gamma(s,t) \quad s.t. \quad \Gamma(0,t) = 0,\ \Gamma(1,t) = \tilde{X}(t) \tag{5.22}$$

which, denoting $\mathcal{X}(s,t) = \Gamma(s,t) + Z(t)$, has dynamic:

$$\begin{aligned}
\frac{\partial \Gamma}{\partial t}(s,t) =& f(\mathcal{X}(\cdot),t) - f(Z(t),t) \\
&+ g(\mathcal{X}(\cdot),t)\psi(t) - g(Z(t),t)\psi(t) && (5.23a) \\
&+ g(\mathcal{X}(\cdot),t)u(\mathbf{y}_x,\mathbf{y}_z,t) \\
=& \varphi(\mathcal{X}(\cdot),t) - \varphi(Z(t),t) \\
&+ g(\mathcal{X}(\cdot),t)u(\mathbf{y}_x,\mathbf{y}_z,t) && (5.23b) \\
=& \varphi(\mathcal{X}(\cdot),t) - \varphi(Z(t),t) \\
&- \kappa g(\mathcal{X}(\cdot),t)\left[u_f(\mathcal{X}(\cdot),t) - u_f(Z(t),t)\right] && (5.23c)
\end{aligned}$$

The Lyapunov function $V(t)$ is constructed as follows

$$V(t) = \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s,t)P(\mathcal{X}(\cdot),t)\frac{\partial \Gamma}{\partial s}(s,t)ds \tag{5.24}$$

In order to take the time derivative of (5.24), it is first useful to show (dropping the arguments (s,t) to favor readability)

$$\begin{aligned}
&\frac{d}{dt}[P(\mathcal{X},t)_{ij}] && (5.25) \\
&= \frac{\partial P_{ij}}{\partial \mathbf{x}}\frac{\partial}{\partial t}(\Gamma + Z) + \frac{\partial P_{ij}}{\partial t}(\Gamma + Z, t) && (5.26) \\
&= \frac{\partial P_{ij}}{\partial \mathbf{x}}[(5.23b) + \varphi(Z,t)] + \frac{\partial P_{ij}}{\partial t}(\Gamma + Z, t) && (5.27)
\end{aligned}$$

Therefore, by using (5.27) and the Killing vector field condition from Assumption 5.2.2, we can write the time derivative of (5.24) in the following form

$$\dot{V}(t) = \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s,t)T_1(s,t)\frac{\partial \Gamma}{\partial s}(s,t)ds \tag{5.28a}$$

with

$$\begin{aligned}
&T_1(s,t) \\
&= \mathscr{L}_\varphi P(\mathcal{X},t) + P(\mathcal{X},t)g(\mathcal{X},t)\frac{\partial u}{\partial \mathbf{x}}(\mathbf{y}_x,\mathbf{y}_z,t) \\
&\qquad\qquad + \frac{\partial u^\top}{\partial \mathbf{x}}(\mathbf{y}_x,\mathbf{y}_z,t)g^\top(\mathcal{X},t)P(\mathcal{X},t) && (5.28b) \\
&= \mathscr{L}_\varphi P(\mathcal{X},t) + P(\cdot)g(\cdot)\frac{\partial u}{\partial \mathbf{x}}(\mathcal{X},Z,t) \\
&\qquad\qquad + \frac{\partial u^\top}{\partial \mathbf{x}}(\mathcal{X},Z,t)g^\top(\cdot)P(\cdot) && (5.28c) \\
&\overset{(5.21c)}{=} \mathscr{L}_\varphi P(\cdot) - 2\kappa P(\cdot)g(\cdot)g^\top(\cdot)P(\cdot) && (5.28d) \\
&\qquad - P(\mathbf{x},t)g(\mathbf{x},t)\frac{\partial \phi}{\partial \mathbf{x}}^\top(\mathbf{x})\boldsymbol{\vartheta} - \boldsymbol{\vartheta}^\top\frac{\partial \phi}{\partial \mathbf{x}}(\mathbf{x})g^\top(\mathbf{x},t)P(\mathbf{x},t) && (5.28e)
\end{aligned}$$

$$\tag{5.28f}$$

$$\overset{(5.19a)}{\preceq} -\lambda P(\cdot) \quad \text{if} \quad \kappa \geq \frac{\rho}{2} \tag{5.28g}$$

The rest of the proof follows the lines of the main one in [6]-III-A: (5.28g) in fact implies that $\dot{V}(t) \leq -\lambda V(t)$*, that in turns implies*

$$V(t) \leq e^{-\lambda(t-t_0)} V(t_0) \quad \forall t > t_0 \tag{5.29}$$

It is then possible to bound $V(t)$ *from its definition in (5.24), leading to*

$$V(t) \geq \underline{p} \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s,t) \frac{\partial \Gamma}{\partial s}(s,t) ds \geq \underline{p} \left\| \tilde{X}(t) \right\|^2 \tag{5.30}$$

Combining this with the fact that $V(t_0) \leq \bar{p} \left\| \tilde{\mathbf{x}}(t_0) \right\|^2$*, it is possible to write*

$$\underline{p} \left\| \tilde{X}(t) \right\|^2 \overset{(5.30)}{\leq} V(t) \overset{(5.29)}{\leq} e^{-\lambda(t-t_0)} \bar{p} \left\| \tilde{\mathbf{x}}(t_0) \right\|^2 \tag{5.31}$$

rewritten,

$$\left\| \tilde{X}(t) \right\|^2 \leq e^{-\lambda(t-t_0)} \frac{\bar{p}}{\underline{p}} \left\| \tilde{\mathbf{x}}(t_0) \right\|^2 \tag{5.32}$$

This, together with the error definition in (5.11) and the fact that $\tilde{X}(t) = \tilde{\mathbf{x}}(t)$ *leads to*

$$\left\| \tilde{\mathbf{x}}(t) \right\|^2 \leq e^{-\lambda(t-t_0)} \frac{\bar{p}}{\underline{p}} \left\| \tilde{\mathbf{x}}(t_0) \right\|^2 \tag{5.33}$$

concluding the proof. ☐

To validate the theoretical results presented in this section, we are going to demonstrates the performance of the adaptive controller through numerical simulations. The Lorenz attractor system serves as the test case, chosen for its chaotic dynamics and nonlinear structure, which provides a challenging benchmark for evaluating the controller's ability to achieve exponential convergence under parametric uncertainty. Two scenarios are considered: first, a single unknown parameter case, and second, a more challenging two-parameter uncertainty configuration. Both cases demonstrate the effectiveness of the adaptive control framework in achieving trajectory tracking with simultaneous parameter estimation.

5.3.2 Numerical results

5.3.2.1 Lorentz Attractor model

The model used to showcase the controller capabilities is the Lorenz' attractor, with the input matrix g designed in order to exclude the possibility of feedback linearizing solutions.

$$\dot{\mathbf{x}} = \begin{bmatrix} \sigma(\mathbf{x}_2 - \mathbf{x}_1) \\ \mathbf{x}_1(\rho - \mathbf{x}_3) - \mathbf{x}_2 \\ \mathbf{x}_1\mathbf{x}_2 - \beta\mathbf{x}_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 + \sin(\mathbf{x}_1) \\ 0 \end{bmatrix} u \tag{5.34a}$$

$$\tag{5.34b}$$

with $\sigma = 10, \rho = 28$ and $\beta = 8/3$. This system is of interest since it is nonlinear and, with the chosen parameters, is chaotic.

One parameter

In the single-parameter case, the regressor function is defined as

$$\phi^\top(\mathbf{x}) = \begin{bmatrix}-\mathbf{x}_2\end{bmatrix} \quad \Rightarrow \quad \Delta^\top(\mathbf{x}) = \begin{bmatrix} -\mathbf{x}_2 \\ -2\mathbf{x}_2 + \mathbf{x}_2 \sin(\mathbf{x}_1) \\ 0 \end{bmatrix} \tag{5.35}$$

The adaptation gain matrix was set to $\Gamma = \mathbf{I}$, while the regularization parameter for parameter drift prevention was chosen as $\sigma = -0.01$. These parameter values were selected to ensure stable adaptation dynamics while maintaining sufficient convergence rate for the parameter estimation.

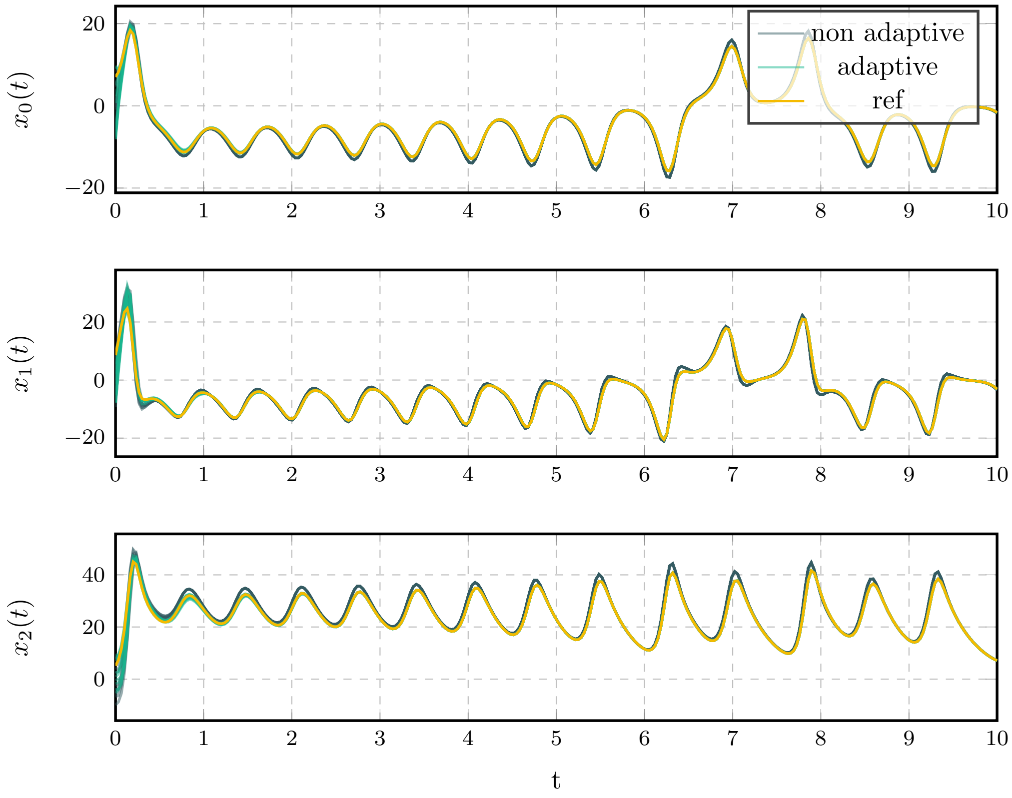


Figure 5.1: State evolution for one parameter adaptive control showing convergence of the controlled trajectory to the reference trajectory.

The simulation results demonstrate the effectiveness of the adaptive controller with one unknown parameter. The state evolution in Figure 5.1 shows successful tracking of the reference trajectory, while Figure 5.2 confirms exponential convergence of the tracking error, down to a minimum residual error.

The parameter evolution depicted in Figure 5.3 provides valuable insight into the adaptation mechanism. The plot demonstrates that the estimated parameter successfully converges to its true value over the course of the simulation, validating the theoretical prediction of parameter identification alongside trajectory tracking. This convergence behavior confirms that the adaptive law effectively estimates the unknown parameter while simultaneously ensuring tracking performance, thus achieving both control and identification objectives.

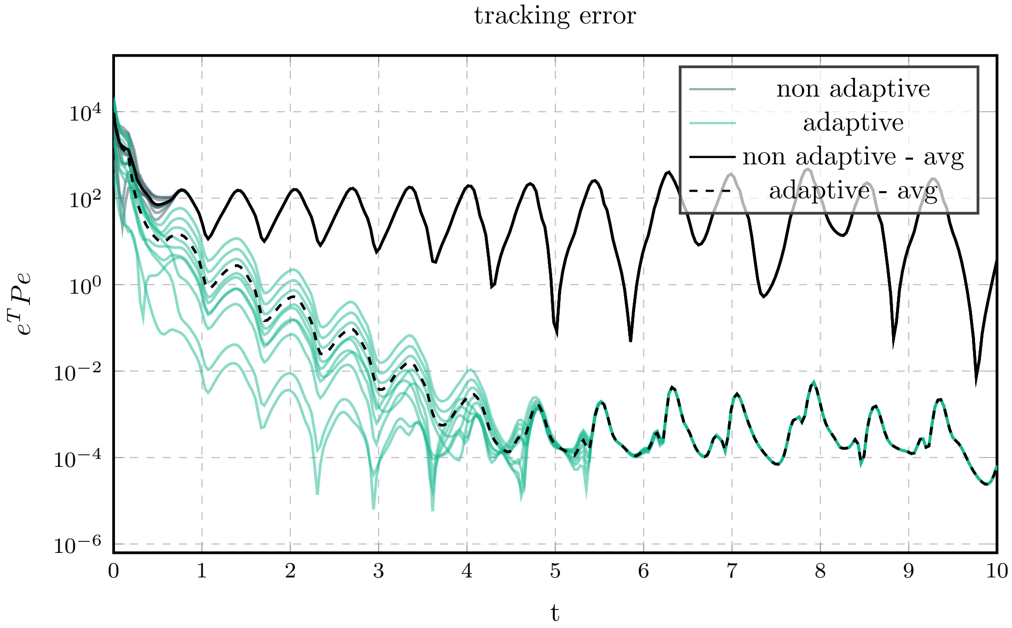


Figure 5.2: Error evolution for one parameter adaptive control demonstrating exponential convergence to zero.

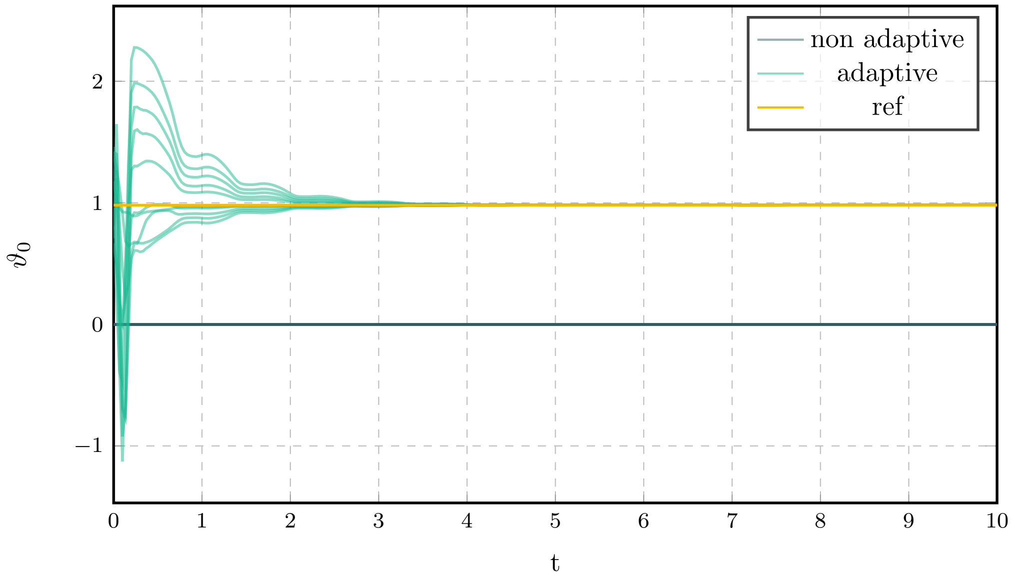


Figure 5.3: Parameters evolution for two parameter occurring during the trajectory tracking phase.

Two parameters

In the two-parameter case, the regressor function is extended to capture additional uncertainty in the system dynamics. The regressor is defined as

$$\phi^\top(\mathbf{x}) = \begin{bmatrix} -\mathbf{x}_2 \\ \mathbf{x}_2 - \mathbf{x}_1 \end{bmatrix} \quad \Rightarrow \quad \Delta^\top(\mathbf{x}) = \begin{bmatrix} -\mathbf{x}_2 & \mathbf{x}_2 - \mathbf{x}_1 \\ -2\mathbf{x}_2 - \mathbf{x}_2 \sin(\mathbf{x}_1) & (\mathbf{x}_2 - 2\mathbf{x}_1)(2 - \sin(\mathbf{x}_1)) \\ 0 & 0 \end{bmatrix} \tag{5.36}$$

The adaptation gain matrix was chosen as $\Gamma = \mathbf{I}$, consistent with the single-parameter configuration. Similarly, the regularization parameter was set to $\sigma = -0.01$ to prevent parameter drift. This choice of parameters maintains uniform adaptation dynamics across both test cases, facilitating meaningful comparison of controller performance when estimating different numbers of uncertain parameters.

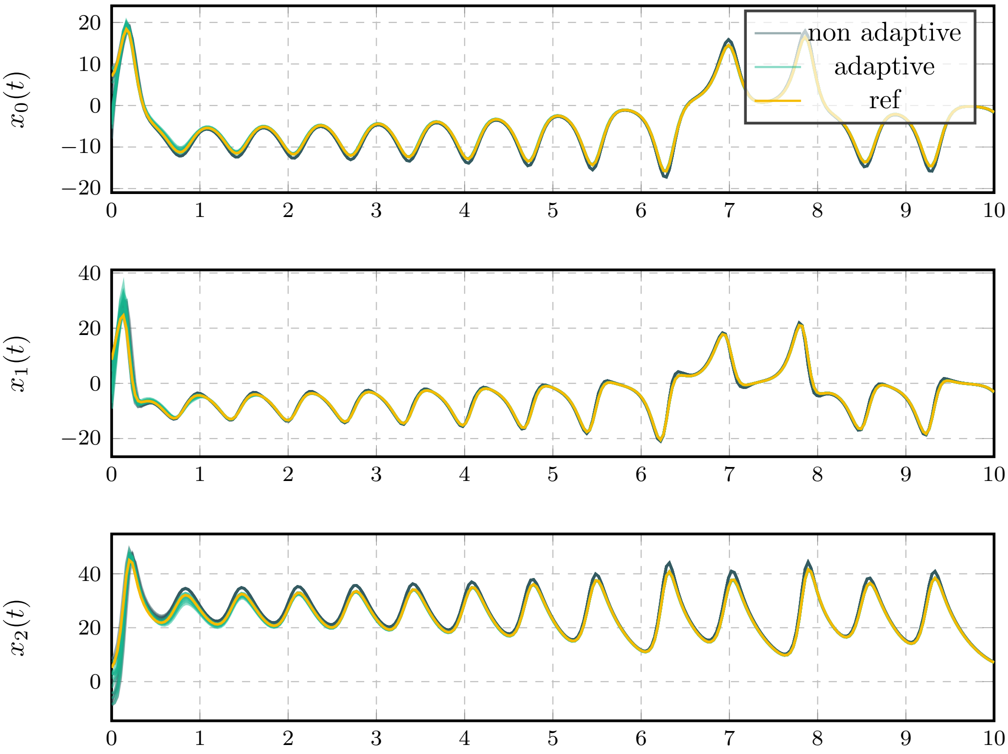


Figure 5.4: State evolution for two parameter adaptive control showing convergence of the controlled trajectory to the reference trajectory.

The simulation results for the two-parameter case mirror the successful performance observed in the single-parameter configuration. As illustrated in Figure 5.4, the state trajectory demonstrates excellent tracking behavior, with the controlled system closely following the reference trajectory despite the presence of two unknown parameters.

The error dynamics shown in Figure 5.5 confirm that the adaptive control scheme maintains exponential convergence properties, achieving tracking errors that are orders of magnitude smaller than those obtained with the non-adaptive baseline controller.

This performance improvement validates the effectiveness of the adaptive extension in handling multiple simultaneous parametric uncertainties while preserving the convergence guarantees established by contraction theory. Once again, the parameters estimate show convergence to the true ones, as it can be seen in Figure 5.6.

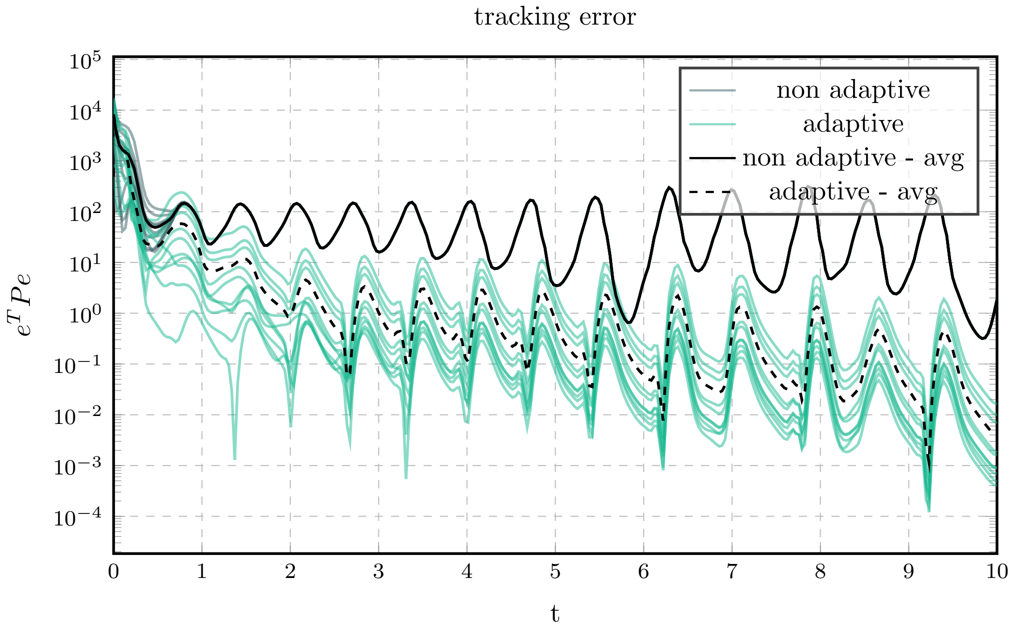


Figure 5.5: Error evolution for two parameter adaptive control demonstrating exponential convergence to zero.

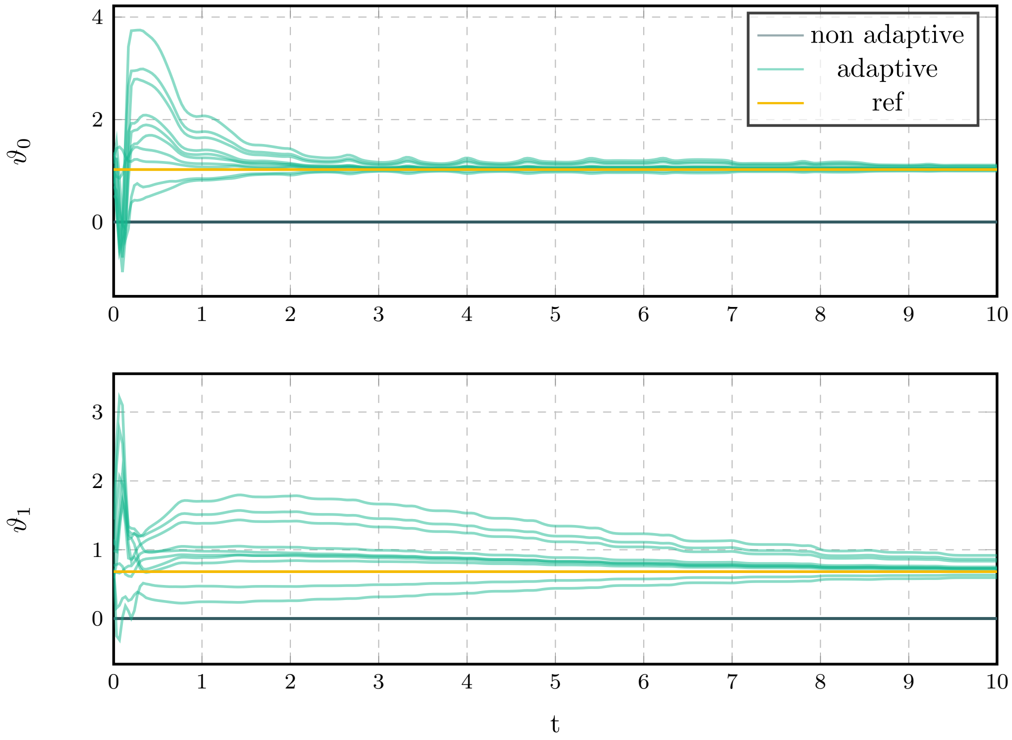


Figure 5.6: Parameters evolution for two parameters occurring during the trajectory tracking phase.

5.3.3 Discussion

This work demonstrates how the state-feedback controller presented in [6] can be extended to incorporate adaptive control techniques, enabling the system to handle parametric uncertainties while preserving exponential convergence guarantees. The numerical results presented in the previous sections show that the adaptive extension achieves significant improvements in tracking performance, with error reductions of several orders of magnitude compared to non-adaptive approaches.

The key contribution lies in the synthesis of contraction theory with adaptive parameter estimation, which allows the controller to maintain the stability and convergence properties of the original design while simultaneously adapting to unknown system parameters. The theoretical framework ensures that the tracking error converges exponentially, as demonstrated both analytically and through simulation on the chaotic Lorenz attractor system.

5.3.3.1 Future work

Several promising directions emerge for extending and validating the proposed approach.

First, comprehensive validation on a broader class of nonlinear systems is essential to establish the practical applicability of the adaptive control framework. This should include testing on systems with different structural properties, varying degrees of nonlinearity, and realistic physical constraints. Following successful simulation studies, experimental validation on physical platforms would provide crucial insights into the controller's performance under real-world conditions, including measurement noise, actuator limitations, and model uncertainties beyond parametric variations.

Second, extending the theoretical framework to handle more general classes of uncertainties represents an important research challenge. The current formulation addresses matched uncertainties, where the uncertainty enters through the control input channel. Developing techniques to handle unmatched uncertainties, time-varying parameters, or combined parametric and non-parametric uncertainties would significantly broaden the applicability of the proposed approach.

Finally, integrating more sophisticated adaptation mechanisms could enhance the robustness and convergence properties of the controller. Advanced techniques such as σ-modification for improved robustness, projection-based adaptation for parameter bounds enforcement, or composite adaptation laws combining prediction and tracking errors could provide additional performance benefits while maintaining the theoretical guarantees established by the contraction framework.

5.4 Output feedback control

5.4.1 Problem Statement

We consider the main couple of dynamical system defined in section 5.2 with (5.10) with the controller (5.13) redefined as in (5.37). Noticeably, the controller takes $\mathbf{y}_z$ and $\mathbf{y}_x$ as input, in place of $\mathbf{z}$ and $\mathbf{x}$ in the original, thus making it an output-feedback controller. The controller then takes the following form:

$$u(\mathbf{y}_x, \mathbf{y}_z, t) = -\kappa \left[u_f(\mathbf{y}_x, t) - u_f(\mathbf{y}_z, t) \right] \tag{5.37}$$

with $u : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, $\psi : \mathbb{R} \to \mathbb{R}^m$ is the reference trajectory input gain, $\kappa \in \mathbb{R}$ is a scalar gain and $u_f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ stands for the feedback part of the controller, that needs to be found.

Remark 5.4.1 (*The controller is distributed*)

Note that the controller terms $u_f(\cdot, t)$ in (5.37) depend only on a single input variable in addition to time. This implies that the feedback component u_f to be designed will depend solely on one of the system's outputs, rather than requiring simultaneous access to multiple output measurements.

As remarked in [6], this property enables synchronization through a distributed feedback law, where each agent can compute its control action independently using only local and reference output information.

5.4.2 Controller design

Finally, since the trajectory tracking task is analyzed through the lens of incremental stability, the following assumption is required:

Assumption 5.4.1 (*Integrability condition*)

The learned controller $u_f(y, t)$ satisfies the following equation

$$\frac{\partial u_f}{\partial \mathbf{y}_x} \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t) = g^\top(\mathbf{x}, t) P(\mathbf{x}, t) \quad \forall \mathbf{x} \in \mathbb{R}^n, t \in \mathbb{R} \tag{5.38}$$

Finally, before presenting the main theorem, it is necessary to establish the relationship between the controller's partial derivatives and the metric $P(\mathbf{x}, t)$. This connection is crucial for the subsequent stability analysis, similar to Remark 5.3.1, and will be formalized through the following differential condition.

Remark 5.4.2

The following equivalence holds

$$\begin{aligned}
&\frac{\partial u}{\partial \mathbf{x}}(\mathbf{y}_x, \mathbf{y}_z, t) && (5.39a)\\
&= \frac{\partial u}{\partial \mathbf{x}}(h(\mathbf{x}), h(\mathbf{z}), t) && (5.39b)\\
&= \frac{\partial u}{\partial \mathbf{y}_x} \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}) && (5.39c)\\
&= -\kappa \frac{\partial u_f}{\partial \mathbf{y}_x} \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t) && (5.39d)\\
&\overset{(5.38)}{=} -\kappa g^\top(\mathbf{x}, t) P(\mathbf{x}, t) && (5.39e)
\end{aligned}$$

At this point all the necessary assumptions and preliminary conditions have now been established. The integrability condition from Assumption 5.4.1 provides the crucial link between the controller structure and the metric $P(\mathbf{x}, t)$, while Assumption 5.2.1 and Assumption 5.2.2 ensure the required contractivity properties of the system. With these elements in place, we now formulate the main exponential stability result for the output feedback control problem.

Theorem 5.4.1 (*Output Feedback exponential stability*)

Consider the dynamical system (5.10) with the output feedback controller (5.37). Suppose that Assumption 5.2.1, Assumption 5.2.2, and Assumption 5.4.1 are satisfied. Then, for any gain $\kappa \geq \frac{\rho}{2}$, the closed-loop system is exponentially incrementally stable, i.e., for any two trajectories $X(t)$ and $Z(t)$ with initial conditions $\mathbf{x}_0$ and $\mathbf{z}_0$ at t_0, the tracking error $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \mathbf{z}(t)$ satisfies

$$\|\tilde{\mathbf{x}}(t)\| \leq \sqrt{\frac{\bar{p}}{\underline{p}}} e^{-\frac{\lambda}{2}(t-t_0)} \|\tilde{\mathbf{x}}(t_0)\| \quad \forall t \geq t_0 \tag{5.40}$$

where λ, $\bar{p}$, and $\underline{p}$ are the same constants used in Assumption 5.2.1.

We now proceed to establish the exponential stability guarantee for the output feedback control scheme. The proof strategy leverages the contraction framework developed in the previous sections, extending the techniques from [6] and Proof 5.3.1 to accommodate the output feedback structure. The key insight is to construct an appropriate Lyapunov function based on the learned metric $P(\mathbf{x}, t)$ and demonstrate that its time derivative satisfies a strict negativity condition under the assumptions stated in the theorem.

Central to this analysis is the integrability condition from Assumption 5.4.1, which establishes the crucial relationship between the controller's partial derivatives and the metric tensor, enabling us to bound the evolution of the tracking error. Through careful application of the differential geometric tools introduced earlier, particularly the Lie derivative analysis and the killing vector field condition, we will show that the closed-loop system exhibits exponential convergence with an explicit rate determined by the parameters λ and the control gain κ.

Proof 5.4.1

The proof follows a similar course of the main proof in [6], and Proof 5.3.1: Define $Z(t)$ and $X(t)$ as solution to the system dynamics equations in (5.10), with initial conditions $\mathbf{z}_0$ and $\mathbf{x}_0$, defined for $t > t_0$, and apply the error equation (5.11), resulting in $\tilde{X}(t) = X(t) - Z(t)$. It is therefore possible to define an "interpolant" on the error as

$$\Gamma(s, t) \quad s.t. \quad \Gamma(0, t) = 0,\ \Gamma(1, t) = \tilde{X}(t) \tag{5.41}$$

which, denoting $\mathcal{X}(s, t) = \Gamma(s, t) + Z(t)$, has dynamic:

$$\begin{aligned} \frac{\partial \Gamma}{\partial t}(s, t) =& (5.23a) \\ =& \varphi(\mathcal{X}(\cdot), t) - \varphi(Z(t), t) \\ & - \kappa g(\mathcal{X}(\cdot), t)\left[u_f(h(\mathcal{X}(\cdot)), t) - u_f(h(Z(t)), t)\right] \end{aligned} \tag{5.42a}$$

The Lyapunov function $V(t)$ is constructed as in (5.24)

$$V(t) = \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s, t) P(\mathcal{X}(\cdot), t) \frac{\partial \Gamma}{\partial s}(s, t) ds \tag{5.43}$$

In order to take the time derivative of (5.43), once again it is possible to resort to (5.27)

Therefore, with (5.27) and the Killing vector field condition from Assumption 5.2.2, we can write the time derivative of (5.43) in the following form:

$$\dot{V}(t) = \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s, t) T_1(s, t) \frac{\partial \Gamma}{\partial s}(s, t) ds \tag{5.44a}$$

with

$$
\begin{aligned}
&T_1(s,t)\\
&= \mathscr{L}_\varphi P(\mathcal{X},t) + P(\mathcal{X},t)g(\mathcal{X},t)\frac{\partial u}{\partial \mathbf{x}}(\mathbf{y}_x,\mathbf{y}_z,t)\\
&\qquad\qquad\qquad + \frac{\partial u^\top}{\partial \mathbf{x}}(\mathbf{y}_x,\mathbf{y}_z,t)g^\top(\mathcal{X},t)P(\mathcal{X},t) \qquad (5.44\text{b})\\
&= \mathscr{L}_\varphi P(\mathcal{X},t) + P(\cdot)g(\cdot)\frac{\partial u}{\partial \mathbf{x}}(h(\mathcal{X}),h(Z),t)\\
&\qquad\qquad\qquad + \frac{\partial u^\top}{\partial \mathbf{x}}(h(\mathcal{X}),h(Z),t)g^\top(\cdot)P(\cdot) \qquad (5.44\text{c})\\
&\overset{(5.39)}{=} \mathscr{L}_\varphi P(\cdot) - 2\kappa P(\cdot)g(\cdot)g^\top(\cdot)P(\cdot) \qquad (5.44\text{d})\\
&\overset{(5.14\text{a})}{\preceq} -\lambda P(\cdot) \quad \text{if} \quad \kappa \geq \frac{\rho}{2} \qquad (5.44\text{e})
\end{aligned}
$$

The rest of the proof follows the lines of the main one in [6]-III-A and is also reported in Proof 5.3.1.

□

5.4.3 Relaxing the integrability condition

It is often the case, when applying this method, that a residual error remains during the learning process. The case where Assumption 5.2.2 is not exactly satisfied is already covered in [6]. However, since Assumption 5.4.1 introduces a different constraint compared to the previous in Assumption 5.2.3, it is necessary to analyze the robustness of the proposed controller with respect to violations of this integrability condition. In particular, we will demonstrate that even when Assumption 5.4.1 is not perfectly satisfied, the tracking error remains bounded, thus providing practical guarantees for the neural network-based implementation.

Theorem 5.4.2

Suppose that Assumption 5.4.1 is not satisfied, i.e. there is an error δ as follows

$$\left|\frac{\partial u_f}{\partial \mathbf{y}_x}\frac{\partial h}{\partial \mathbf{x}}(\mathbf{x},t) - g^\top(\mathbf{x},t)P(\mathbf{x},t)\right| \leq \delta \qquad (5.45)$$

consider $\bar{p}$ from Assumption 5.2.1, and suppose that

$$|g(\mathbf{x},t)| \leq \bar{g} \qquad \forall(\mathbf{x},t) \in \mathbb{R} \times \mathbb{R}_{\geq 0} \qquad (5.46)$$

for some $\bar{g} > 0$. Then, if δ is such that $\delta \leq \frac{\lambda}{\rho\bar{g}}$, it is still possible to select κ that leads to a controller able to perform the trajectory tracking task.

Proof 5.4.2

The proof follows the same calculations of the proof for Theorem 5.4.1 until (5.44a), where $\dot{V}(t)$ turns out to be as follows

$$\dot{V}(t) = \int_0^1 \frac{\partial \Gamma^\top}{\partial s}(s,t)\,[T_1(s,t) + T_2(s,t)]\,\frac{\partial \Gamma}{\partial s}(s,t)\,ds \tag{5.47}$$

where

$$\begin{aligned} T_2(s,t) &= -\kappa\left(P(\cdot)g(\cdot)\delta + \delta g(\cdot)^\top P(\cdot)\right) && (5.48)\\ &\preceq -2\kappa\bar{g}\delta P(\cdot) && (5.49)\end{aligned}$$

This let us write (by adding and subtracting $\rho P g g^T P$)

$$T_1(\cdot) + T_2(\cdot) \preceq -(\lambda - 2\kappa\bar{g}\delta)P(\cdot) + (\rho - 2\kappa)P(\cdot)g(\cdot)g(\cdot)^\top P(\cdot) \tag{5.50}$$

From the constraint on δ given by the theorem statement, we have that $\frac{\rho}{2} \leq \frac{\lambda}{2\bar{g}\delta}$. Now, selecting $\kappa \in \left(\frac{\rho}{2}, \frac{\lambda}{2\bar{g}\delta}\right)$, With the chosen κ, the term $(\lambda - 2\kappa\bar{g}\delta)$ remains positive, thus the proof can still be carried out as in the main theorem:

5.4.4 Neural Network training

The training strategy to learn the controller is to split the learning process into two steps, as in [6]. The first is to learn the matrix $P(\cdot)$, with the Deep Neural Network (DNN) $\mathcal{P}(\cdot)$, and the second is to learn the controller $u_f(\cdot)$ with the DNN $\mathcal{U}(\cdot)$. Since the desired matrix $P(\mathbf{x}, t)$ is positive definite, hence symmetric, the network is not designed to output the full matrix, but just the upper triangular entries, filling the others accordingly.

Specifically, indicating as $\mathcal{P}_i(\mathbf{x}, t)$ the i-th entry of the DNN output vector, with a slight abuse of notation we can write $\mathcal{P}$ as (5.51):

$$\mathcal{P}(\mathbf{x}, t) = \begin{bmatrix} \mathcal{P}_1(\mathbf{x},t) & \cdots & \mathcal{P}_n(\mathbf{x},t) \\ & \ddots & \\ \mathcal{P}_n(\mathbf{x},t) & \cdots & \mathcal{P}_N(\mathbf{x},t) \end{bmatrix} \tag{5.51}$$

where $N = \frac{n(n+1)}{2}$. In this context, the time dependency is supposed to arise solely from the reference trajectory. Rather than having the networks directly depend on time through $\mathcal{P}(y, t)$ and $\mathcal{U}(y, t)$, the reference trajectory input $\psi(t)$ is provided as an additional input to the DNNs. This architectural choice results in $\mathcal{P}(y, \psi(t))$ and $\mathcal{U}(y, \psi(t))$, effectively capturing the time-varying behavior through the reference signal. This approach offers greater generality, as the networks do not require an explicit time variable for the trajectory tracking task under consideration.

The first DNN was trained with the widely used Adam [37] optimizer, wherease the second was trained using AdamW (Adam with Weight Decay Regularization [38]).

5.4.4.1 Metric training

For the metric $P(\cdot)$ Assumption 5.2.1 and Assumption 5.2.2 are the conditions that need to be satisfied. From these, the cost function to train the NN $\mathcal{P}(\cdot)$ can be defined as follows.

First, define the four costs

$$\begin{aligned} W_1 &= \mathscr{L}_\varphi \mathcal{P}(\cdot) - \rho\mathcal{P}(\cdot)g(\cdot)g^\top\mathcal{P}(\cdot)\mathcal{P}(\cdot) + 2\lambda\mathcal{P}(\cdot) && (5.52a)\\ W_2 &= \mathscr{L}_g \mathcal{P}(\cdot) - \varepsilon I && (5.52b)\\ W_3 &= -\mathscr{L}_g \mathcal{P}(\cdot) - \varepsilon I && (5.52c)\\ W_4 &= -\mathcal{P}(\cdot) + \underline{p} I && (5.52d)\end{aligned}$$

where the parameters $\rho, \lambda, \varepsilon$ and $\underline{p}$ are positive, and are taken from the output of another very small neural network, trained in parallel with $\mathcal{P}(\cdot)$, to optimize them. The details of this parallel training process can be found in [6].

Then the loss for the first neural network is defined as

$$\mathcal{L}_{\mathcal{P}}(\mathbf{x}) = \sum_{i=1}^{4} w_i \ln\left(\mathrm{ReLu}(\alpha(W_i))\right) \tag{5.53}$$

where w_i are positive weights and $\alpha(A)$ denotes the *spectral abscissa* of A, namely $\alpha(A) = \max\{\Re(\lambda) \mid \lambda \in \mathrm{spec}(A)\}$. The loss in (5.53) is different from zero if at least one of the required conditions is not satisfied, so differently from other standard DNN training processes, it is expected to reach zero. Nevertheless, in practice, it can happen that a residual training error remains, but as discussed in [6], tracking error bounds can be derived with respect to this error. At the end of this first training phase, the trained DNN is used to proceed to the next training phase.

Remark 5.4.3

Note that in [24] they use the same Linear Matrix Inequality (LMI) as in (5.14a) (With $\mathscr{L}_{\boldsymbol{\varphi}} P(..)$), but in the loss function they use the term $\mathscr{L}_{\boldsymbol{f}} P(..)$ in the first operand. We utilize $\mathscr{L}_{\boldsymbol{\varphi}} P(..)$, to be coherent with the LMI. We noticed no remarkable difference between the network trained with the two conditions, as to state that the DNN $\mathcal{P}$ can to be very robust.

5.4.4.2 Controller training

Now that the metric $\mathcal{P}(\cdot)$ has been learned, the focus can shift to the neural network $\mathcal{U}(\cdot)$ that learns the controller. Since the only assumption required on the controller is the equality constraint in Assumption 5.4.1, the loss can simply be defined as

$$\mathcal{L}_{\mathcal{U}}(\mathbf{x}) = \mathrm{MSE}\left(\frac{\partial u_f}{\partial \mathbf{y}_x}\frac{\partial h}{\partial \mathbf{x}}(\cdot), g^{\top}(\cdot)\mathcal{P}(\cdot)\right) \tag{5.54}$$

For this training as well, some residual error is allowed, and the error bounds deriving from it are discussed in subsection 5.4.3.

5.4.5 Numerical results

The first model used to test the validity of the controller is the same used in subsubsection 5.3.2.1, shown in (5.34a), that is the Lorentz Attractor model.

The two neural networks are both Multi Layer Perceptron (MLP), where both the DNNs have 4 layers of 1024 neurons, and are trained for 500 epochs.

5.4.5.1 Simulations - no noise

To validate the results, 10 simulations have been carried out without any noise injected into the system or feedback loop. From Figure 5.7 is possible to see that the controller is able to track the trajectory, with exponential error decay (after an initial, expected overshoot), until reaching a residual error.

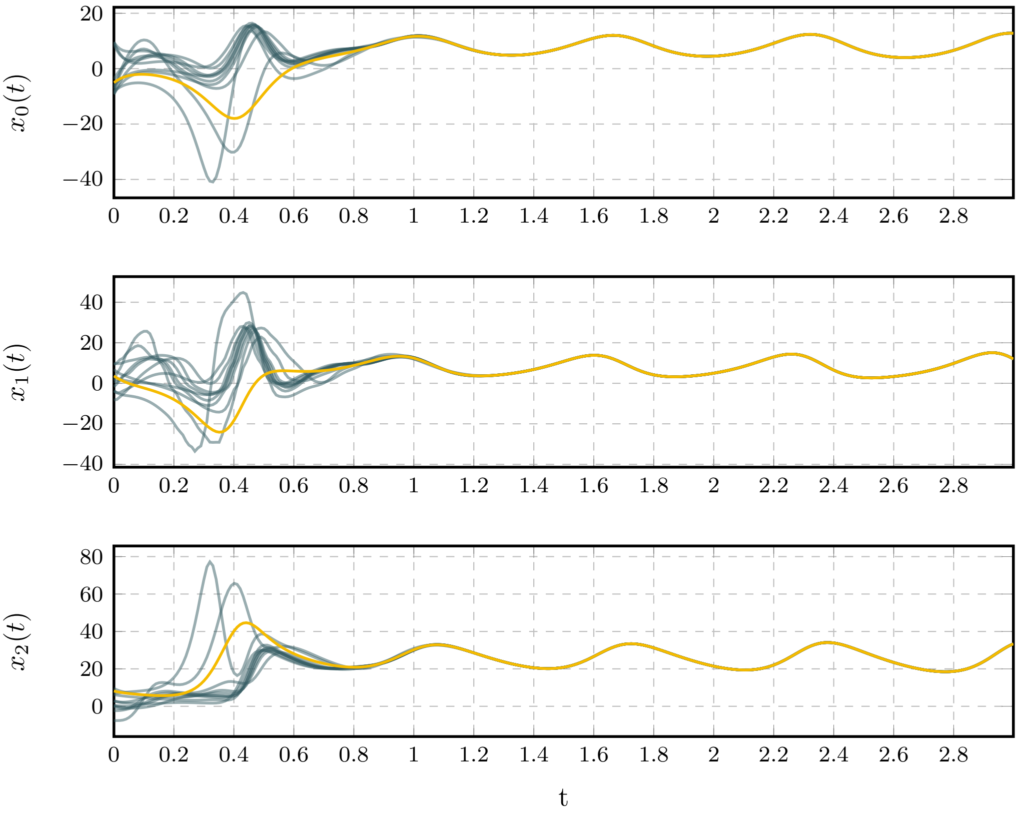


Figure 5.7: State evolution for output feedback control showing convergence of multiple trajectories from different initial conditions to the reference trajectory.

This tracking error observed in the simulations, shown in Figure 5.8, is remarkably small, with magnitudes on the order of 10^{-7} - 10^{-10}. This extremely low error level strongly suggests that the residual discrepancies are primarily attributable to numerical precision limitations and computational round-off errors, rather than fundamental limitations of the control strategy itself.

Finally, in Figure 5.8 the input signal is can also be observed. Its plot shows, as expected, that the controller ceases to provide feedback, as expected.

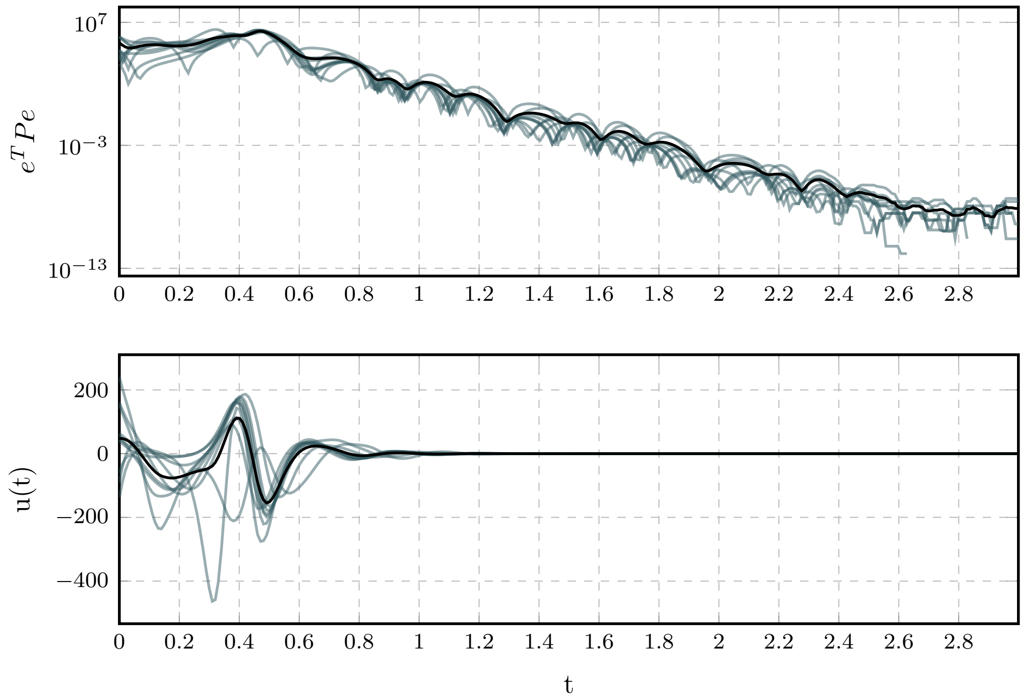


Figure 5.8: Error (top) and control input (bottom) evolution for output feedback control. The error converges and the input convergence to a neighborhood of zero for trajectories starting from different initial conditions.

5.4.5.2 Simulations - noisy scenario

To test the controller robustness, it was tested with Gaussian random noise applied both to the output that was fed back to it, and to the input that the controller exerted on the system.

The simulations show how the controller is still able to stabilize the trajectories in a neighborhood of the reference trajectory. This can be seen in Figure 5.9.

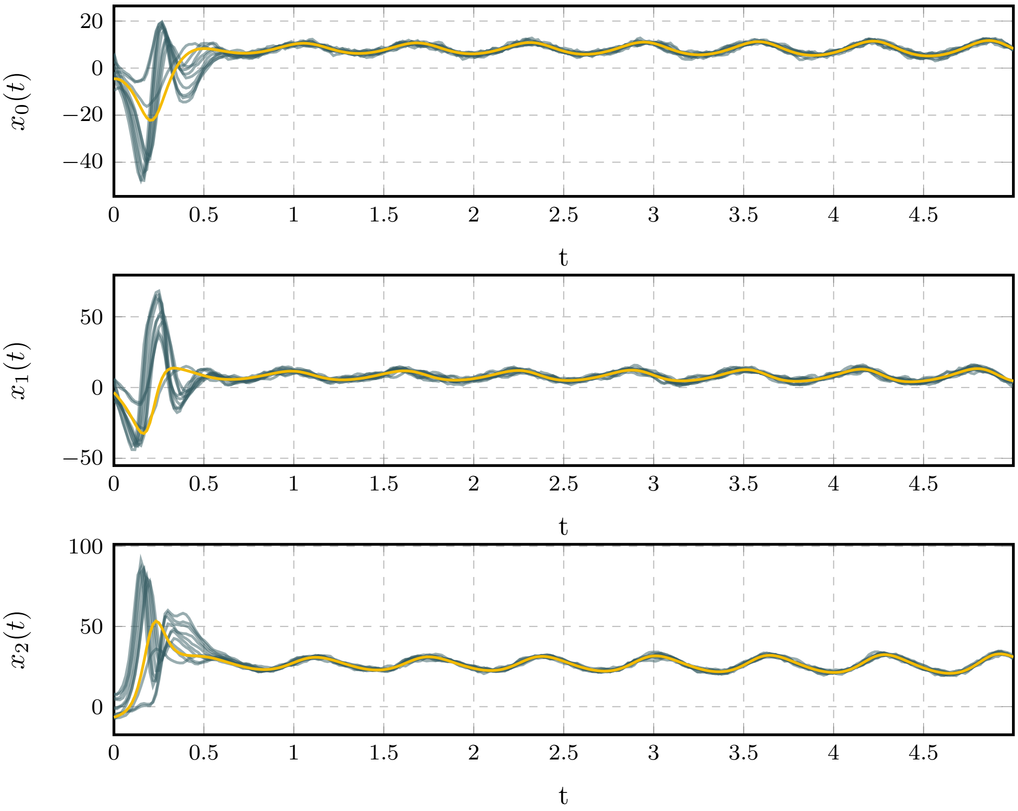


Figure 5.9: State evolution for output feedback control in a noisy scenario. The synchronization of multiple trajectories still happens, with bounded error due to noise from multiple sources.

It can be seen that the trajectories' error stays bounded also from the error plot in Figure 5.10, showing how such controller can present robustness to multiple noise sources.

Finally, the input signal shows how in this scenario, differently from the input signal of Figure 5.8, keeps acting on the controller for the whole duration of the simulation. This is to be expected, since the presence of random noise prevents the perfect synchronization to a reference trajectory that was generated without any noise.

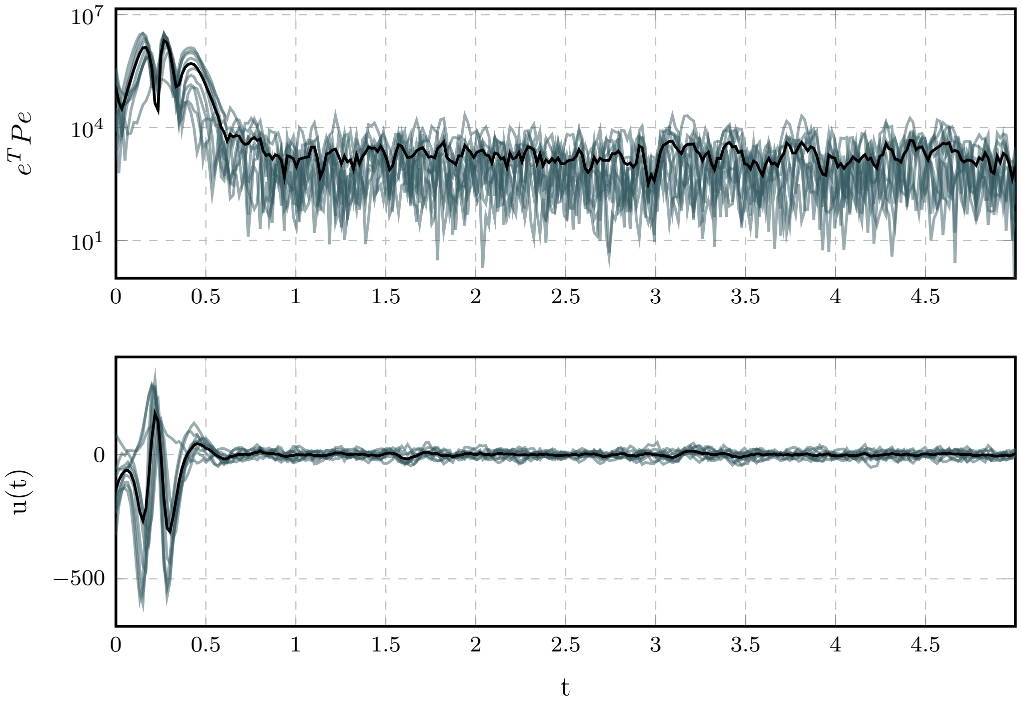


Figure 5.10: Error (top) and control input (bottom) evolution for output feedback control in a noisy scenario. The error remains bounded and the input convergence to a neighborhood of zero for trajectories starting from different initial conditions.

5.5 Discussion

This section presented an extension of the contraction-based synchronization framework from [6] to output feedback control for nonlinear trajectory tracking. Building upon the synchronization methodology developed by Giaccagli, Astolfi and co-authors, we adapted their approach to address practical scenarios where only partial state information is available through measurements, while incorporating adaptive control capabilities.

A key distinction between our approach and related work on output feedback control with contraction theory, such as [24], is that our method operates directly with pure output measurements without requiring an intermediate step of estimating the state reference trajectory. Furthermore, while [24] employs recurrent neural networks to handle the temporal dependencies in output feedback control, our approach utilizes standard multi-layer perceptrons (MLPs), simplifying both the network architecture and the training procedure.

The main contributions of this adaptation can be summarized as follows: First, we extended the systematic design procedure from [6] to output feedback controllers that guarantee exponential incremental stability through the satisfaction of three key conditions: the main LMI from Assumption 5.2.1, the killing vector field property from Assumption 5.2.2, and the integrability condition introduced in Assumption 5.4.1. The integrability condition, which relates the controller's partial derivatives to the learned metric $P(\mathbf{x}, t)$, proved to be central to establishing the connection between output measurements and closed-loop stability in this output feedback setting.

Second, we adapted the two-stage neural network training methodology from the original synchronization work to learn both the contraction metric $P(\mathbf{x}, t)$ and the output feedback controller $u_f(\mathbf{y}_x, t)$ from data. The first stage follows the approach in [6] to learn the metric by enforcing the required LMIs, while the second stage was modified to learn the controller by satisfying the integrability condition with output measurements rather than full state information. This decomposition maintains the efficient training structure of the original method while enabling the output feedback extension.

Third, we extended the robustness analysis to the output feedback case through subsection 5.4.3, showing that the approach maintains stability guarantees even when the integrability condition is satisfied only approximately, with bounded error δ. This result, paralleling similar analyses in the original work, is particularly relevant for practical implementations where neural network approximation errors may prevent perfect satisfaction of the theoretical conditions.

The numerical validation on the Lorenz attractor system demonstrated that the extended approach inherits the desirable properties of the original synchronization method. In noise-free scenarios, the controller achieved exponential convergence with tracking errors at the level of numerical precision (10^{-7} - 10^{-10}), effectively demonstrating that the theoretical stability guarantees translate into excellent practical performance. In the presence of multiple noise sources affecting both measurements and control inputs, the simulations showed that trajectories remain bounded and exhibit robust synchronization behavior, validating the practical applicability of the extended method under realistic conditions.

An important characteristic inherited from the original synchronization framework is the distributed structure of the controller, as noted in the problem formulation. The feedback terms $u_f(\mathbf{y}_x, t)$ and $u_f(\mathbf{y}_z, t)$ depend only on individual output measurements, enabling decentralized implementation where each agent can compute its control action using only local and reference information. This property, which was a key feature of the work in [6], remains valuable for multi-agent coordination and synchronization tasks.

The extensions presented here demonstrate that the contraction-based synchronization framework can be successfully adapted to output feedback control scenarios while maintaining theoretical stability guarantees and practical effectiveness. The adaptations preserve the essential structure of the original methodology while addressing the additional challenges posed by partial state measurements and adaptive control requirements.

Chapter 6

Conclusions

This dissertation has addressed the two fundamental problems of modeling and control of robotic systems. The work first focused on studying state-of-the-art aerial platforms, extending their geometric configurations, analyzing their modified properties, and proposing novel control schemes tailored to these enhanced architectures. Subsequently, the research shifted to more classical dynamical systems in order to develop new controllers applicable to general nonlinear systems.

The novel dynamical system extensively investigated in Part I is the fully-actuated multirotor aerial vehicle known as the Star-shaped Tilted Hexarotor (STH). The analysis began by studying its properties under multiple configurations of the tilt *cant angle* α in chapter 2, and then systematically extended this investigation to include the *dihedral angle* β in chapter 3. This dual-angle parameterization enabled a comprehensive characterization of the actuation polytope and force generation capabilities.

The research presented in Part II focused on developing advanced control strategies. In chapter 4, we demonstrated that two newly developed controllers for the STH platform of chapter 3 exhibit high performance in trajectory tracking and attitude regulation. Subsequently, in chapter 5, we successfully extended existing contraction-based controllers using methods that greatly enhance their applicability to real-world scenarios. Specifically, an adaptive control layer was successfully integrated to handle parametric uncertainties, and the original state-feedback controller was extended to achieve pure output-feedback control, thereby removing the requirement for full-state measurements.

6.1 Future research directions

6.1.1 STH Modeling

The model studied proved to be useful for practical considerations regarding the manufacture of adapters to put on a planar platform that was present in our lab.

The next steps for this research are to first validate all the results with an experimental setup. This would prove the correctness of the approach, and also possibly could reveal discrepancies with our results, since we do not yet account for parameter variations like the c_f and c_τ coefficients, possible different arm lenghs ℓ, and finally, the z position of the propeller is the same as the CoM of the UAV. This last statement is not true even in our experimental setup, so further research is needed to investigate all the possible effects of said variations. Finally, this study could provide useful tools to study tilted platforms with eterogeneous angles between propellers, opening up a whole new variety of specialized drones.

6.1.2 Contraction Theory-based Control

We extended existing controllers by enhancing their practical applicability. On this note, the next steps are to apply these controllers to more realistic systems, following up with realistic simulations and experimental validation next. On the computational side, improvements should be studied to make real-time the inference of the controller network. The first step to work towards this objective is to find ways of reducing its overall size, or by quantizing its states. Moreover, using the results from Appendix B, it could be possible to train the Metric DNN as a Physics Informed Neural Network (PINN), possibly improving training speed. Lastly, the DF property would prove particularly useful for the output-feedback controller, thus more research and development on this topic should be carried out.

Bibliography

[1] Perin, M., Bertoni, M., Michieletto, G., Oboe, R., and Cenedese, A., "Trajectory tracking for tilted hexarotors with concurrent attitude regulation," in *2024 American Control Conference (ACC)*, ISSN: 2378-5861, Jul. 2024, pp. 1550–1555. DOI: 10.23919/ACC60939.2024.10644887. [Online]. Available: https://ieeexplore.ieee.org/document/10644887.

[2] Perin, M., Bertoni, M., Viezzer, N., Michieletto, G., and Cenedese, A., "Star-shaped tilted hexarotor maneuverability: Analysis of the role of the tilt cant angles," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 1791–1797. DOI: 10.1109/CASE59546.2024.10711709.

[3] Michieletto, G., Ryll, M., and Franchi, A., "Fundamental actuation properties of multirotors: Force–moment decoupling and fail–safe robustness," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 702–715, 2018. DOI: 10.1109/TRO.2018.2821155.

[4] Michieletto, G., Cenedese, A., and Franchi, A., "Force-moment decoupling and rotor-failure robustness for star-shaped generically-tilted multi-rotors," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 2132–2137.

[5] Michieletto, G., Cenedese, A., Zaccarian, L., and Franchi, A., "Hierarchical nonlinear control for multi-rotor asymptotic stabilization based on zero-moment direction," *Automatica*, vol. 117, p. 108 991, 2020.

[6] Giaccagli, M., Zoboli, S., Astolfi, D., Andrieu, V., and Casadei, G., "Synchronization in networks of nonlinear systems: Contraction analysis via riemannian metrics and deep-learning for feedback estimation," May 2023. Accessed: Jul. 15, 2024. [Online]. Available: https://hal.science/hal-03801100.

[7] *Differential Flatness of Mechanical Control Systems: A Catalog of Prototype Systems*, vol. Dynamic Systems and Control: Volume 1 — Vibration Control; Dynamic Systems; Robotics; Sliding Mode Control; Robust and Nonlinear Control; Automated Modeling; Control of Manufacturing Processes; Precision Control, ASME International Mechanical Engineering Congress and Exposition, Nov. 1995, pp. 349–357. DOI: 10.1115/IMECE1995-0796. eprint: https://asmedigitalcollection.asme.org/IMECE/proceedings-pdf/IMECE95/17469/349/7548134/349_1_imece1995-0796.pdf. [Online]. Available: https://doi.org/10.1115/IMECE1995-0796.

[8] Han, Z. et al., *Differential flatness-based trajectory planning for autonomous vehicles*, Aug. 28, 2022. DOI: 10.48550/ARXIV.2208.13160. arXiv: 2208.13160 [cs]. Accessed: Jan. 18, 2023. [Online]. Available: http://arxiv.org/abs/2208.13160.

[9] Rashad, R., Goerres, J., Aarts, R., Engelen, J. B. C., and Stramigioli, S., "Fully actuated multirotor uavs: A literature review," *IEEE Robotics & Automation Magazine*, vol. 27, no. 3, pp. 97–107, 2020. DOI: 10.1109/MRA.2019.2955964.

[10] Faessler, M., Franchi, A., and Scaramuzza, D., "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.

[11] Ferrin, J., Leishman, R., Beard, R., and McLain, T., "Differential flatness based control of a rotorcraft for aggressive maneuvers," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Ieee, 2011, pp. 2688–2693.

[12] Michieletto, G., Cenedese, A., Zaccarian, L., and Franchi, A., "Nonlinear control of multi-rotor aerial vehicles based on the zero-moment direction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 13 144–13 149, 2017.

[13] Lohmiller, W. and Slotine, J.-J. E., “On contraction analysis for non-linear systems,” *Automatica*, vol. 34, no. 6, pp. 683–696, 1998. DOI: https://doi.org/10.1016/S0005-1098(98)00019-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109898000193.

[14] Manchester, I. R. and Slotine, J.-J. E., “Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design,” *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.

[15] Bullo, F., *Contraction Theory for Dynamical Systems*, 1.2. Kindle Direct Publishing, 2024. [Online]. Available: https://fbullo.github.io/ctds.

[16] Davydov, A. and Bullo, F., “Perspectives on contractivity in control, optimization, and learning,” *IEEE Control Systems Letters*, vol. 8, pp. 2087–2098, 2024. DOI: 10.1109/LCSYS.2024.3436127.

[17] Manchester, I. R., “Control contraction metrics and universal stabilizability of nonlinear systems,” *IFAC-PapersOnLine*, vol. 51, no. 23, pp. 162–167, 2018.

[18] Wang, H. et al., “A contraction theory-based adaptive robust control for the trajectory tracking of a pneumatic cylinder,” *IEEE Access*, vol. 12, pp. 118 234–118 245, 2024.

[19] Zhao, Z., Huang, J., Cao, M., and Wu, Y., “Exponentially tracking control of nonholonomic wheeled mobile robot based on contraction theory,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2022, pp. 1–6.

[20] Singh, S., Majumdar, A., Slotine, J.-J., and Pavone, M., “Robust online motion planning via contraction theory and convex optimization,” in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5883–5890.

[21] Gahlawat, A., Pan, Z., Hovakimyan, N., Theodorou, E. A., and Tsiotras, P., “Safe feedback motion planning: A contraction theory and l1-adaptive control based approach,” *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1665–1682, 2021.

[22] Chang, Y.-C., Ratliff, N., Lowell, J., and Manchester, I. R., “Neural contraction metrics for robust estimation and control: A convex optimization approach,” *IEEE Control Systems Letters*, vol. 6, 2020, arXiv preprint arXiv:2006.04361.

[23] Dawson, C., Gao, S., and Fan, C., “Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023.

[24] Zoboli, S., Janny, S., and Giaccagli, M., “Deep learning-based output tracking via regulation and contraction theory,” *IFAC-PapersOnLine*, 22nd IFAC World Congress, vol. 56, no. 2, pp. 8111–8116, Jan. 1, 2023. DOI: 10.1016/j.ifacol.2023.10.966. Accessed: May 23, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896323013496.

[25] Gahlawat, A., Yin, H., Tsiotras, P., Theodorou, E. A., and Hovakimyan, N., “Contraction l1-adaptive control using gaussian processes,” in *Learning for Dynamics and Control (L4DC)*, PMLR, 2021, pp. 617–629.

[26] Sun, W., Tsiotras, P., and Yue, Y., “Learning certified control using contraction metric,” in *Conference on Robot Learning (CoRL)*, PMLR, 2021, pp. 1519–1529.

[27] Forni, F. and Sepulchre, R., “A differential lyapunov framework for contraction analysis,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 614–628, 2014.

[28] Beik-Mohammadi, H., Hauberg, S., Arvanitidis, G., Figueroa, N., Neumann, G., and Rozo, L., “Neural contractive dynamical systems,” 2024. arXiv: 2401.09352 `[cs.RO]`. [Online]. Available: https://arxiv.org/abs/2401.09352.

[29] Åström, K. J. and Wittenmark, B., *Adaptive Control*, 2nd. Mineola, NY: Dover Publications, 2013.

[30] Ioannou, P. A. and Sun, J., *Robust Adaptive Control*. Mineola, NY: Dover Publications, 2012.

[31] Narendra, K. S. and Annaswamy, A. M., *Stable Adaptive Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[32] Hovakimyan, N. and Cao, C., *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*. Philadelphia, PA: SIAM, 2010.

[33] Lopez, B. T. and Slotine, J.-J. E., "Adaptive nonlinear control with contraction metrics," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 205–210, Jan. 2021. DOI: 10.1109/LCSYS.2020.3000190. Accessed: Jun. 18, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9109296/.

[34] Lewis, F. L., Jagannathan, S., and Yeşildirek, A., *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London: Taylor & Francis, 2013.

[35] Chowdhary, G., Yucelen, T., Mühlegg, M., and Johnson, E. N., "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *International Journal of Adaptive Control and Signal Processing*, vol. 27, no. 4, pp. 280–301, 2013.

[36] Bullo, F. and Lewis, A. D., *Geometric Control of Mechanical Systems* (Texts in Applied Mathematics). New York-Heidelberg-Berlin: Springer Verlag, 2004, vol. 49.

[37] Kingma, D. P. and Ba, J., *Adam: A method for stochastic optimization*, 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: 1412.6980 `[cs.LG]`. [Online]. Available: https://arxiv.org/abs/1412.6980.

[38] Loshchilov, I. and Hutter, F., "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:53592270.

Appendices

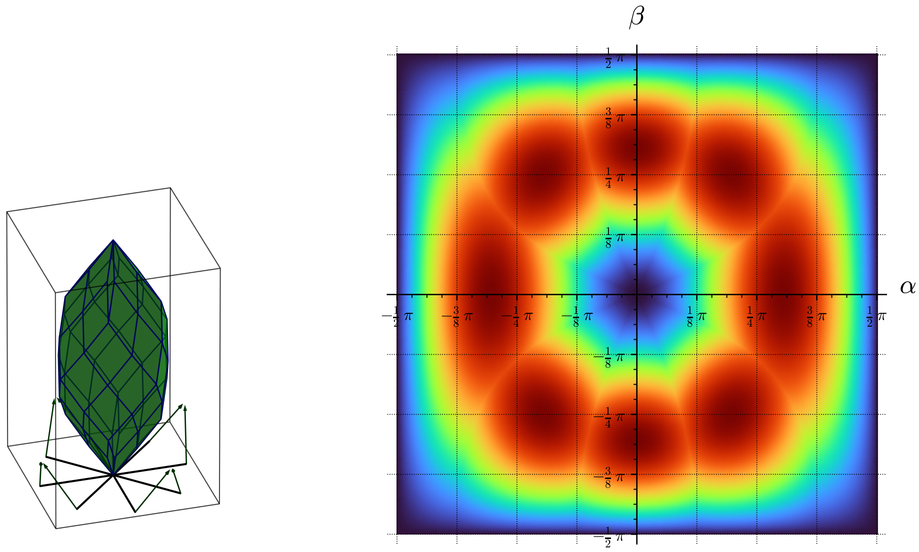
Appendix A

Octarotor alpha-beta tilted analysis

A.1 Octarotor Volume Analysis

This section presents a comprehensive volumetric analysis focused on understanding the force allocation capabilities inherent to the octarotor configuration under investigation. The central aim of this investigation is to systematically examine how the volume of the allocatable force polytope varies across different vehicle body orientations and attitudes. Through the characterization of these fundamental geometric properties, we develop crucial insights into the operational envelope of the system, along with its inherent maneuvering constraints and limitations.

Building upon the geometric framework for multi-rotor force allocation previously established in [1], this approach has proven itself as an essential tool for both effective control system design and strategic mission planning. In the present analysis, we quantify the specific influence that the tilt angles α and β exert on the system's capacity to generate forces along different spatial directions. This directional force generation capability has direct implications for the vehicle's agility characteristics and ultimately determines its fundamental performance limits during operation.



(a) Polytope of allocatable forces for the octarotor configuration

(b) Volume of allocatable force polytope as a function of body angles α and β for the octarotor configuration

Figure A.1: Octarotor force allocation analysis

The volumetric map presented in Figure A.1b reveals an intriguing pattern that warrants closer examination. Notably, the visualization exhibits the same number of distinct peaks as there are propellers in the configuration. This observation strongly suggests a correlation between the number of actuators and the geometric structure of the force allocation landscape, indicating that this phenomenon is likely not merely coincidental.

The recurring nature of this pattern, previously observed in Figure 3.2 for a different configuration, points to a potentially fundamental relationship between system architecture and force allocation geometry. Further analytical and numerical studies are planned to investigate the underlying mechanisms responsible for this behavior, with the goal of establishing a generalized understanding of how propeller count influences the topology of the allocatable force polytope volume.

Appendix B

Meshless collocation for contraction metric matrix estimation

B.1 Introduction

Meshless collocation approximates solutions to differential equations using radial basis functions (RBFs) without requiring mesh structures [2], [3]. The method employs a reproducing kernel Hilbert space framework with Wendland functions, providing compact support and optimal convergence in Sobolev spaces [2], [4].

For a system $\dot{x} = f(x)$ with equilibrium $\bar{x}$, a Lyapunov function V satisfies $\langle \nabla V(x), f(x) \rangle = -p(x)$ where $p(x) > 0$ [5]. Generalized interpolation with RBFs at collocation points approximates V, with error bounds $\|\langle \nabla s, f \rangle - \langle \nabla V, f \rangle\|_{L^\infty(\Omega)} \leq Ch^{k-1/2}$, where h is fill distance and k relates to kernel smoothness [2], [4]. Applications include contraction metrics, finite-time Lyapunov functions, and reconstruction from noisy data [5], [6], [7].

B.2 Contraction Metric Construction

Pokkakkillath et al. developed meshfree collocation for constructing contraction metrics in discrete-time autonomous systems [8]. A converse theorem establishes existence of a metric $S(x)$ solving a matrix-valued equation for systems $x_{k+1} = f(x_k)$ [8].

The meshless approximation imposes no geometric domain restrictions, enabling flexible node placement [8]. Rigorous error estimates guarantee that sufficiently accurate numerical approximations maintain contraction properties [8]. The metric $S(x)$ is symmetric positive definite, verifying $S(x) \succ 0$ and $F(S)(x) \prec 0$ where F captures trajectory contraction [8]. This enables basin of attraction determination and robustness analysis under perturbations [8], [9].

The methodology establishes a matrix-valued function $M(x)$ satisfying a specific inequality constraint. Given the discrete-time dynamics previously introduced, the construction requires:

$$Df(x)^T M(f(x)) Df(x) \leq \rho^2 M(x) \tag{B.1}$$

where $Df(x)$ denotes the Jacobian of f, and $\rho \in (0, 1)$ is the contraction rate.

To enforce this condition through collocation, the authors introduce an auxiliary operator that measures deviation from the required inequality. Specifically, they define:

$$F(M)(x) = Df(x)^T M(f(x)) Df(x) - \rho^2 M(x) - \rho^2 C \tag{B.2}$$

where C is a positive definite constant matrix that ensures strict inequality satisfaction.

The solution strategy approximates $M(x)$ using a matrix-valued basis function expansion $S(x)$. Each component of $S(x)$ is represented as a linear combination of radial basis functions centered at the collocation points.

If the collocation points are close enough to each other, $S(x)$ will be positive definite and $F(S)(x)$ will be negative definite so $S(x)$ will be a contraction metric.

In their implementation they explicitly find a system of linear equations that solves for the matrix $S(x)$. This is a remarkable point, since compared to the classic literature and the results in chapter 5, this does not require solving any LMIs.

B.3 High-Performance Implementation

The algorithm presented in [8] has been completely reimplemented with the Julia programming language, [1] employing an adaptive point placement strategy that replaces the original regular grid approach. Rather than distributing collocation points uniformly, new points are added strategically based on a computed cost function.

The key insight lies in the incremental nature of the linear system governing the coefficients of $S(x)$. When new collocation points are introduced, the existing system is expanded rather than reconstructed from scratch. This allows the numerical solver to initialize from the previously computed solution, resulting in rapid convergence to the updated metric approximation.

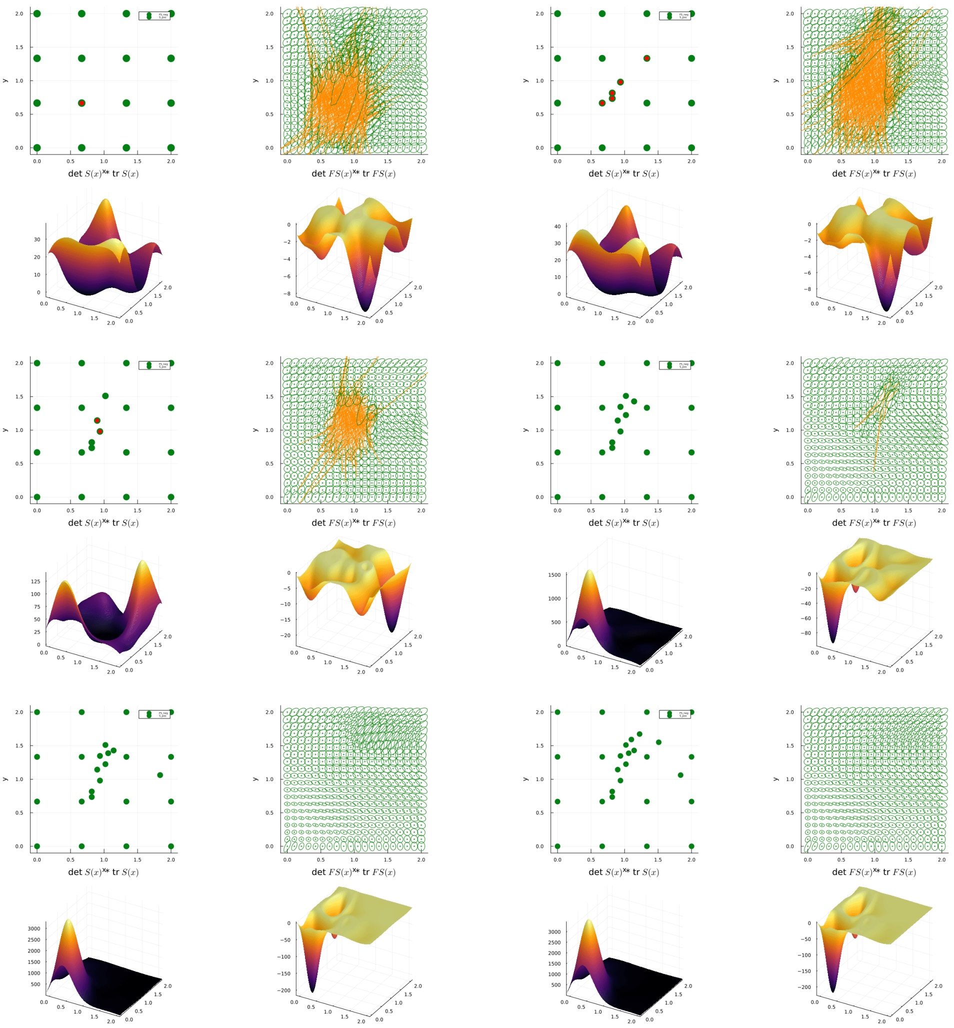


Figure B.1: Progressive estimation of the contraction metric matrix by adding collocation points based on estimation error.

The cost function itself is derived from the contraction inequality (B.1) that defines the metric. Specifically, we evaluate the approximation error of this inequality across the domain. New collocation points are then placed at locations where this error reaches its maximum value, ensuring that computational effort is concentrated in regions where the current approximation is weakest. This adaptive refinement strategy has proven

---

[1] The implementation for the library is available at https://github.com/marco-perin/MeshFreeCollocationContraction.jl

highly effective in achieving accurate metric representations with minimal computational overhead.

The adaptive refinement process is visualized in Figure B.1. Beginning with a sparse regular grid of collocation points, the algorithm incrementally adds new points at each iteration, as depicted in the successive frames. The progression clearly demonstrates how the quality of the metric approximation improves: the ellipse fields, which represent the local geometry of the contraction metric $S(x)$, become increasingly regular and smooth with each refinement step. This visual evidence confirms that the adaptive placement strategy effectively concentrates computational resources where the approximation error is highest, leading to rapid convergence toward an accurate metric representation.

This meshless collocation approach presents a compelling alternative to the deep neural network $\mathcal{P}$ employed in chapter 5. For systems with state spaces of moderate dimension, the method could potentially replace the neural network entirely, offering the advantage of directly constructing the metric without the need for training procedures. In higher-dimensional settings, where direct application becomes computationally prohibitive, the technique remains valuable: it can generate a collection of verified metric values at the collocation points, providing high-quality training data for $\mathcal{P}$ within a PINN framework. This hybrid approach would leverage the meshless method's ability to satisfy the contraction inequalities exactly at discrete points, while utilizing the neural network's capacity to generalize across the entire state space described by the dynamics previously introduced.

Bibliography

[1] Perin, M., Bertoni, M., Viezzer, N., Michieletto, G., and Cenedese, A., “Star-shaped tilted hexarotor maneuverability: Analysis of the role of the tilt cant angles,” in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, 2024, pp. 1791–1797. DOI: 10.1109/CASE59546.2024.10711709.

[2] Giesl, P. and Wendland, H., “Meshless collocation: Error estimates with application to dynamical systems,” *SIAM Journal on Numerical Analysis*, vol. 45, no. 4, pp. 1723–1741, 2007.

[3] Giesl, P., *Construction of Global Lyapunov Functions Using Radial Basis Functions* (Lecture Notes in Mathematics). Berlin: Springer, 2007, vol. 1904.

[4] Giesl, P., “Construction of a finite-time lyapunov function by meshless collocation,” *Discrete & Continuous Dynamical Systems-B*, vol. 17, no. 7, pp. 2387–2412, 2012.

[5] Giesl, P. and Wendland, H., “Construction of a contraction metric by meshless collocation,” *Discrete & Continuous Dynamical Systems-B*, vol. 24, no. 8, pp. 3843–3863, 2019.

[6] Giesl, P., Hamzi, B., Rasmussen, M., and Webster, K. N., “Approximation of lyapunov functions from noisy data,” *Journal of Computational Dynamics*, vol. 7, no. 1, pp. 57–81, 2020, arXiv preprint arXiv:1601.01568.

[7] Giesl, P. and Hafstein, S., “Construction of a contraction metric for time-periodic systems using meshless collocation,” *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 761–766, 2015, Related work on time-periodic systems.

[8] Pokkakkillath, S., Giesl, P., and Hafstein, S., “Construction of contraction metrics for discrete-time dynamical systems using meshfree collocation,” *Discrete and Continuous Dynamical Systems - Series B*, vol. 29, no. 4, pp. 1884–1913, 2024. DOI: 10.3934/dcdsb.2023166.

[9] Pokkakkillath, S., Giesl, P., and Hafstein, S., “Piecewise quadratic Lyapunov functions for stochastic differential equations by linear programming,” *Discrete and Continuous Dynamical Systems - Series B*, 2024. DOI: 10.3934/dcdsb.2024148.

Acknowledgements

Thanks to my advisor Angelo and my co-advisor Giulia, for helping me grow as a researcher. Thanks to Prof. Luca Schenato for his friendly yet professional advices.

Thank you, Prof. Francesco Bullo, for your hospitality at UCSB and for the insightful talks and suggestions. Your guidance motivated me to keep pursuing this path, showing me that it is possible to strive for great research, while keeping my feet on the ground.

Grazie a Mamma e Papà per i princìpi che mi avete trasmesso sin da piccolo ed essere sempre stati un porto sicuro dove tornare. Vi devo molto. Thank you, Sara, for always being there, even if from a different part of the globe each time. I am so proud of you, thank you for being there reliably, even without much talking. Grazie Nonna Uccia, per essere sempre stata al fianco mio e di Sara, mostrandoci cosa sono la dedizione e l'amore. Un dolce ricordo e saluto a Nonno Franco; grazie per avermi insegnato così tante cose, per avermi trasmesso la passione per il voler sistemare tutto ed aiutare sempre. Un grande ricordo anche a Nonno Sergio, per i valori che mi hai trasmesso sin da piccolo, e a Nonna Gilda, per avermi insegnato la dolcezza ancora prima chAe potessi capirla. Ringrazio anche zio Leri per la sua gentilezza velata dietro strati di pazzia, e grazie zio Fabio per avermi sempre supportato in ogni traguardo.

Thank you, Irene, for supporting me through such a long and important part of my journey. Your kind soul helped me grow into the person I am today.

Thank you, Giacomo, Francesco and Federico, for the crazy ideas, projects and adventures we shared, and for the long, long talks we have had since we met.

Thank you, Veronica, for being such a good long-time friend. Even though we do not meet that often, it always feels wonderful to spend time together. Thank you, Piero, for being such a good friend and for always being present, in time of need.

Thank you to my friends and research fellows Umberto, Luca, Chiara, Miguel, Simone, Matteo, Tommaso, Jacopo, Elena, and all the other folks who accompanied me during my years at DEI. You made them so enjoyable, even during the toughest moments.

Thank you to Emilio and Elena, you made Santa Barbara feel much more Italian. Thanks to the folks at Mech-E (and beyond) in Santa Barbara: thank you, Yohan, for being a valuable friend and for putting up with my poor climbing skills, you have been a great teacher. Thank you, Gabe, for your friendliness and kindness; I wish we had met earlier. Thanks, Abed, for the late-night conversations and the drives in your 4Runner, you made me feel at home more than once. Thank you, Sasha, for the talks we had about music and for introducing me to so many nice and niche groups. Thanks, Jared, Nibodh and Anand, for being unexpectedly great buddies, even if some jokes got lost in translation.

Thank you, Flora, for our deep and lovely talks, you have been such a sweet discovery. Thank you Davide and Carlo, for putting up with my fear of falling and for being so crazy. Thanks to all the Cobblestone people: Jury, Soel, Ester, Enri, Anna, Sofi, Chetto, Anna and Davide Palma!. You all made Padova harder to leave, in the best possible way.

A thank you goes also to all the other people that I have not mentioned here, but was present for a time in my life and helped me in any way.