

Universidade Federal de Minas Gerais  
Escola de Engenharia  
Programa de Pós-graduação em Engenharia Elétrica

Arthur Henrique Dias Nunes

# Robust guidance and control for quadcopters in safe operations

Belo Horizonte

2023

Arthur Henrique Dias Nunes

# **Robust guidance and control for quadcopters in safe operations**

Final thesis presented to the Graduate Program in Electrical Engineering of Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the Master's degree in Electrical Engineering.

Advisor: Prof. Luciano Cunha de Araújo Pimenta

Belo Horizonte  
2023

N972r

Nunes, Arthur Henrique Dias.

Robust guidance and control for quadcopters in safe operations  
[recurso eletrônico] / Arthur Henrique Dias Nunes. - 2023.

1 recurso online (109 f. : il., color.) : pdf.

Orientador: Luciano Cunha de Araújo Pimenta.

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

Bibliografia: f. 103-109.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Veículos autônomos - Teses.  
3. Controle robusto - Teses. 4. Impacto - Teses. 5 Campos vetoriais -  
Teses. I. Pimenta, Luciano Cunha de Araújo. II. Universidade Federal de  
Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
ESCOLA DE ENGENHARIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"ROBUST GUIDANCE AND CONTROL FOR QUADCOPTERS IN SAFE OPERATIONS"

ARTHUR HENRIQUE DIAS NUNES

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 18 de dezembro de 2023. Por:

Prof. Dr. Luciano Cunha de Araújo Pimenta  
DELT (UFMG) - Orientador

Prof. Dr. Guilherme Vianna Raffo  
DELT (UFMG)

Prof. Dr. Leonardo Antônio Borges Torres  
DELT (UFMG)



Documento assinado eletronicamente por **Luciano Cunha de Araujo Pimenta, Professor do Magistério Superior**, em 18/12/2023, às 16:54, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Guilherme Vianna Raffo, Professor do Magistério Superior**, em 18/12/2023, às 18:03, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Leonardo Antonio Borges Torres, Professor do Magistério Superior**, em 19/12/2023, às 08:29, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site [https://sei.ufmg.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **2905729** e o código CRC **DC6E22DB**.

***To My Inner Scientist:*** *This is dedicated to the child within me who once dreamed of becoming a scientist. May your curiosity and wonder continue to fuel my pursuit of knowledge.*

# Acknowledgements

I would like to express my sincere gratitude to the following individuals and institutions whose support, guidance, and contributions have been invaluable throughout my academic journey:

**Family and friends:** To my family and friends who have been my pillars of strength, offering patience and encouragement during all the phases of my academic pursuit, I am grateful. I would like to extend a special thank you to my father, Vanderlino Nunes da Silva Júnior, my mother, Simone Soares Dias Nunes, my grandfather, Vanderlino Nunes da Silva, and my future wife, Maria Luiza de Andrade Alves.

In loving memory of my grandmother, Cacilda Soares Dias, who may not have comprehended the complexities of my studies but recognized the importance of education in one's life. Her joy whenever I was 'going to school' will forever be cherished.

**Academic support:** I am thankful to the entire university and instructors at Universidade Federal de Minas Gerais for a great academic environment. My deepest appreciation goes to my advisor, Luciano Cunha de Araújo Pimenta, for his mentorship, encouragement, and expert guidance, which have played a crucial role in shaping the direction and quality of this research.

**Collaborators and colleagues:** My gratitude extends to my fellow students, research colleagues, and lab mates who have been both collaborators and a source of inspiration.

**Professional support:** I would like to express my appreciation to my supervisor and friend at work, Daniel Oliveira Nascimento, for his support and belief in my educational pursuits.

**Institutional support:** I acknowledge the support provided by Universidade Federal de Minas Gerais and the Graduate Program of Electrical Engineering. Their academic structure and resources have been instrumental in making this research possible.

**Dedication:** I dedicate this thesis to all who once dreamt of becoming scientists, for the eternal pursuit of knowledge and the relentless curiosity that drives us all.

Arthur.

*“Whether you think you can or think you can’t, you’re right.”*  
*(Henry Ford)*

# Resumo

O uso de quadricópteros autônomos em diversas aplicações exige operações robustas e seguras. Essas aplicações incluem vigilância, monitoramento, rastreamento de alvos, busca e salvamento, entrega de pacotes, dentre outras. Este trabalho apresenta uma nova estratégia de guiagem e controle para veículos quadricópteros, garantindo uma navegação segura e confiável ao longo de caminhos predefinidos. O problema abordado neste trabalho consiste em controlar um quadricóptero para seguir um determinado caminho evitando obstáculos previamente desconhecidos e mitigando os efeitos de perturbações desconhecidas e limitadas. É introduzido uma estrutura de guiagem e controle. Este esquema visa aumentar a rejeição de perturbações e evitar obstáculos em tarefas de seguimento de caminhos, melhorando assim a segurança e o desempenho geral do sistema. A estratégia de guiagem é uma nova abordagem de campo vetorial desenvolvida em um trabalho anterior, onde o campo guia um sistema integrador simples para seguir um caminho determinado quando for seguro e circunavegar obstáculos localmente medidos quando colisões iminentes são detectadas, fazendo uma transição suave entre ambos. Por outro lado, o controle é baseado em uma técnica conhecida de Backstepping para o sistema nominal que garante que ele siga o caminho quando não houver perturbação. Na realidade prática de sistemas perturbados, introduzimos um controle robusto adaptativo que visa minimizar o erro de rastreamento entre o sistema nominal e o sistema real. O método proposto é validado com simulações computacionais e experimentos do mundo real, mostrando a capacidade do controlador de conduzir quadricópteros autônomos em tarefas de navegação seguras. O estudo não apenas contribui para a base de conhecimento acadêmico, mas também é uma promessa substancial para aplicações práticas em indústrias que dependem de operações aéreas precisas e confiáveis. Em conclusão, apresenta-se uma abordagem para enfrentar os desafios do seguimento de caminho por quadricóptero através do desenvolvimento de um sistema robusto de guiagem e controle. Ao melhorar a segurança e a confiabilidade destes veículos aéreos não tripulados, este trabalho facilita o uso destes veículos numa vasta gama de aplicações do mundo real.

Palavras-chave: quadricóptero; veículos autônomos; controle robusto; evitamento de colisões; guiagem por campos vetoriais.

# Abstract

The usage of autonomous quadcopters in various applications requires robust and safe operations. Those applications include surveillance, monitoring, target tracking, search and rescue, and package delivery, among others. This work presents a novel guidance and control strategy for quadcopter vehicles, ensuring safe and reliable navigation along predefined paths. The problem addressed by this work consists of controlling a quadcopter to follow a given path while avoiding previously unknown obstacles and mitigating the effects of unknown but limited disturbances. We introduce a guidance and control framework. This scheme aims to enhance disturbance rejection and avoid obstacles in path-following tasks, improving overall system safety and performance. The guidance strategy is a novel vector field approach developed in our previous work, where the field guides a generic single-integrator system to follow the given path when it is safe and smoothly changes to guide the system to contour locally sensed obstacles when imminent collisions are detected. On the other hand, the control is based on a well-known Backstepping technique for the nominal system that guarantees it follows the path when no disturbance acts. In the practical reality of disturbed systems, we introduce a robust adaptive control that aims to minimize the tracking error between the nominal system and the real system. We validate the proposed method with both computational simulations and real-world experiments, showing the capability of our controller to drive an autonomous quadcopter in safe path-following navigation tasks. The study not only contributes to the academic knowledge base but also holds substantial promise for practical applications in industries that rely on precise and reliable aerial operations. In conclusion, this work presents an approach to addressing the challenges of quadcopter path-following through the development of a robust guidance and control system. By enhancing the safety and reliability of these unmanned aerial vehicles, the research facilitates their usage in a wide array of real-world applications.

*Keywords: quadcopter; autonomous vehicles; robust control; collision avoidance; vector field guidance.*

# List of Figures

|                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1 – DJI Matrice 100 quadcopter ( <a href="http://dji.com/br/matrice100">dji.com/br/matrice100</a> ) utilized by UFMG Mechatronics, Control, and Robotics (MACRO) group. . . . .                                                                                                                                                                                                                                                              | 22 |
| Figure 2 – Qualitative comparison of achieved publications with this work. From left to right we have the oldest to newest publication. From top to bottom, we have the most related to the least related publications. . .                                                                                                                                                                                                                         | 25 |
| Figure 3 – Result from [Nunes et al., 2020] which shows the robot’s trajectory for the region between the curves while avoiding obstacles. We translated the original image into English. . . . .                                                                                                                                                                                                                                                   | 31 |
| Figure 4 – Result from [Pereira et al., 2021] which shows the robot’s trajectory for the path while avoiding dynamic obstacles. . . . .                                                                                                                                                                                                                                                                                                             | 31 |
| Figure 5 – Result from [Rezende et al., 2020] which shows the convergence of the robot to a boundary set of the target curve. . . . .                                                                                                                                                                                                                                                                                                               | 32 |
| Figure 6 – Visualization of a particle or point with three-dimensional position $\mathbf{p} \in \mathbb{H}_p$ and linear velocity $\mathbf{v} \in \mathbb{H}_p$ . . . . .                                                                                                                                                                                                                                                                           | 39 |
| Figure 7 – Geometric interpretation of $\mathbf{o} \in \mathbb{H}$ , given by an angular displacement of $\phi = \frac{\pi}{4}$ , illustrated by dashed black lines, around an axis $\mathbf{n} = \hat{\mathbf{i}}\frac{1}{\sqrt{2}} + \hat{\mathbf{j}}\frac{1}{\sqrt{2}}$ illustrated by a black arrow. Geometric visualization of $\boldsymbol{\omega} \in \mathbb{H}_p$ illustrated by dotted black rings around the body (rotated) frame. . . . | 41 |
| Figure 8 – Illustration of the sets: $\mathcal{O}$ as a dark gray circle, $\mathcal{W}$ as a light gray region, and $\mathcal{S} \subset \mathcal{W}$ as a blue region. The $\lambda$ parameter that sets the boundary of $\mathcal{S}$ is also shown in red dashed lines. . . . .                                                                                                                                                                  | 44 |
| Figure 9 – Illustration of the field $\mathcal{F}(\mathbf{p}, t)$ in the two main conditions. On the left, when it is safe to follow the curve, $\mathcal{F}(\mathbf{p}, t) = \Phi(\mathbf{p}, t)$ . On the right, when imminent collision is detected, $\mathcal{F}(\mathbf{p}, t) = \Psi(\mathbf{p}, t)$ . . . . .                                                                                                                                | 46 |
| Figure 10 – Image from [Nunes et al., 2022]. It shows an illustration of an obstacle as a dark gray shape, a robot as a light gray circle with a range sensor and the quantities $\mathbf{D}_o(\mathbf{p}, t)$ and $\mathbf{D}_\lambda(\mathbf{p}, t)$ as black arrows. . . . .                                                                                                                                                                     | 49 |
| Figure 11 – Illustration from [Nunes et al., 2022] of the projection of the auxiliary field $\mathcal{A}(\mathbf{p}, t)$ used to compute a tangent vector $\mathbf{T}_\lambda(\mathbf{p}, t)$ . . . . .                                                                                                                                                                                                                                             | 50 |

|                                                                                                                                                                                                                                                                                                                                                       |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 12 – Controller architecture diagram with the combination of a nominal controller and a robust adaptive control. It shows: the nominal feedback loop with the nominal controller and the nominal system; the robust adaptive controller; and combining both control actions, the complete proposed controller that acts in the system. . . . . | 53 |
| Figure 13 – Illustration of $\mathcal{S}$ in blue, $\mathcal{C}$ in black line, and $\mathcal{O}$ as a dark gray circle. Three example trajectories that show invariance and attractiveness of $\mathcal{S}$ are plotted in red dashed lines. . . . .                                                                                                 | 57 |
| Figure 14 – Illustration of $\mathcal{S}$ in blue, $\mathcal{C}$ as a black line, and $\mathcal{O}$ as a dark gray circle. The field $\mathcal{F}$ on $\partial\mathcal{S}(t)$ is plotted as red arrows that never point outside of $\mathcal{S}$ . . . . .                                                                                           | 57 |
| Figure 15 – Crazyflie 2.1 platform picture, available at <a href="http://bitcraze.io/products/crazyflie-2-1">bitcraze.io/products/crazyflie-2-1</a> . . . . .                                                                                                                                                                                         | 71 |
| Figure 16 – Test Bench Gimbal Ring. On the left is the 3D model and on the right is the prototype. . . . .                                                                                                                                                                                                                                            | 72 |
| Figure 17 – Position Controller simulation I: Distance between the practical system and the curve in continuous lines and distance between the nominal system and the curve in dashed lines (both overlap). . . . .                                                                                                                                   | 74 |
| Figure 18 – Position Controller simulation I: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom). . . . .                                                                                                                                                                         | 75 |
| Figure 19 – Position Controller simulation I: Evolution in time of systems states. . . . .                                                                                                                                                                                                                                                            | 75 |
| Figure 20 – Position Controller simulation II: Disturbance $\mathbf{f}_d$ . . . . .                                                                                                                                                                                                                                                                   | 76 |
| Figure 21 – Position Controller simulation II: Distance between the practical system and the curve in continuous lines and distance between the nominal system and the curve in dashed lines. . . . .                                                                                                                                                 | 77 |
| Figure 22 – Position Controller simulation II: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom). . . . .                                                                                                                                                                        | 77 |
| Figure 23 – Position Controller simulation II: Error between practical and nominal positions (top) and velocities (bottom). . . . .                                                                                                                                                                                                                   | 78 |
| Figure 24 – Position Controller simulation II: Evolution in time of adaptive gains. . . . .                                                                                                                                                                                                                                                           | 78 |
| Figure 25 – Position Controller simulation II: Evolution in time of systems states. . . . .                                                                                                                                                                                                                                                           | 79 |
| Figure 26 – Position Controller simulation III: Disturbance $\mathbf{f}_d$ . . . . .                                                                                                                                                                                                                                                                  | 80 |
| Figure 27 – Position Controller simulation III: Distances between the system and the curve in blue, between the system and the obstacles in red. . . . .                                                                                                                                                                                              | 81 |
| Figure 28 – Position Controller simulation III: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom). . . . .                                                                                                                                                                       | 81 |

|                                                                                                                                                                                                                                                                     |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 29 – Position Controller simulation III: Error between practical and nominal positions (top) and velocities (bottom). . . . .                                                                                                                                | 82 |
| Figure 30 – Position Controller simulation III: Evolution in time of adaptive gains. . . . .                                                                                                                                                                        | 83 |
| Figure 31 – Position Controller simulation III: Evolution in time of systems states. . . . .                                                                                                                                                                        | 83 |
| Figure 32 – Controller architecture diagram for the simulations with nominal controller only. In this diagram, the control loop is closed with the nominal controller and the practical system. . . . .                                                             | 84 |
| Figure 33 – Position Controller simulation IV: Distances between the system and the curve in blue, between the system and the obstacles in red. . . . .                                                                                                             | 85 |
| Figure 34 – Position Controller simulation IV: Evolution in time of control inputs. . . . .                                                                                                                                                                         | 85 |
| Figure 35 – Position Controller simulation IV: Evolution in time of systems states. . . . .                                                                                                                                                                         | 86 |
| Figure 36 – Attitude Controller simulation I: disturbance $\tau_d$ . . . . .                                                                                                                                                                                        | 88 |
| Figure 37 – Attitude Controller simulation I: norm of the attitude error between the <b>practical system</b> and the reference trajectory, expressed as $\ \text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\ $ . . . . .                                                      | 89 |
| Figure 38 – Attitude Controller simulation I: control inputs for the nominal system (top), error system (middle), and practical system (bottom). . . . .                                                                                                            | 89 |
| Figure 39 – Attitude Controller simulation I: evolution in time of orientation error (top graph) and velocity error (bottom graph). In red dashed lines are the ultimate bounds $\delta_o$ and $\delta_\omega$ for orientation and velocity, respectively. . . . .  | 90 |
| Figure 40 – Attitude Controller simulation I: evolution in time of adaptive gains. . . . .                                                                                                                                                                          | 91 |
| Figure 41 – Attitude Controller simulation I: evolution in time of angular velocity (bottom) and attitude (top) in continuous lines with the reference attitude in dashed lines. Both attitudes were converted to Euler angles for a more readable result. . . . .  | 92 |
| Figure 42 – Attitude Controller simulation II: norm of the attitude error between the <b>practical system</b> and the reference trajectory, expressed as $\ \text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\ $ . . . . .                                                     | 94 |
| Figure 43 – Attitude Controller simulation II: Evolution in time of control inputs. . . . .                                                                                                                                                                         | 94 |
| Figure 44 – Attitude Controller simulation II: evolution in time of angular velocity (bottom) and attitude (top) in continuous lines with the reference attitude in dashed lines. Both attitudes were converted to Euler angles for a more readable result. . . . . | 95 |
| Figure 45 – Attitude Controller experiment I: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles. . . . .                                                                                             | 96 |
| Figure 46 – Attitude Controller experiment I: attitude error expressed as $\ \text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\ $ . . . . .                                                                                                                                    | 97 |
| Figure 47 – Attitude Controller experiment II: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles. . . . .                                                                                            | 98 |
| Figure 48 – Attitude Controller experiment II: attitude error expressed as $\ \text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\ $ . . . . .                                                                                                                                   | 98 |

|                                                                                                                                                                           |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 49 – Attitude Controller experiment III: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles. . . . . | 99 |
| Figure 50 – Attitude Controller experiment III: attitude error expressed as $\ \text{Im}\{\boldsymbol{o}_d^* \boldsymbol{o}\}\ $ . . . . .                                | 99 |

# List of abbreviations and acronyms

|      |                                           |
|------|-------------------------------------------|
| UFMG | Universidade Federal de Minas Gerais      |
| UAV  | Unmanned Aerial Vehicles                  |
| PID  | Proportional, Integrative, and Derivative |
| LQR  | Linear Quadratic Regulator                |
| LMI  | Linear Matrix Inequality                  |
| SGC  | Separated Guidance and Controller         |
| IGC  | Integrated Guidance and Controller        |
| RGB  | Red, Green, and Blue                      |

# List of symbols

|                             |                                                                                                          |
|-----------------------------|----------------------------------------------------------------------------------------------------------|
| $t$                         | A non-negative real scalar used to denote time                                                           |
| $n$                         | A natural scalar used to denote the order of a dimensional space                                         |
| $i$                         | An integer scalar used as a counter                                                                      |
| $s$                         | A real scalar used to denote the parametric value on parametric functions                                |
| $v_r$                       | A positive real scalar used to denote the desired speed of the system                                    |
| $g$                         | A positive real scalar used to denote gravity magnitude                                                  |
| $m$                         | A positive real scalar used to denote the system mass                                                    |
| $\alpha_i$                  | $i - th$ surface used for the vector field at work <a href="#">Gonçalves et al. [2010a]</a>              |
| $\mathbb{R}$                | Vector space of real numbers                                                                             |
| $\mathbb{R}^n$              | $n$ dimensional vector space of real numbers                                                             |
| $\mathbb{R}_+$              | Vector space of non-negative real numbers                                                                |
| $\mathbb{H}$                | Vector space of quaternions                                                                              |
| $\mathbb{H}_p$              | Vector space of pure quaternions                                                                         |
| $\mathbb{S}$                | Space of circle                                                                                          |
| $\hat{i}, \hat{j}, \hat{k}$ | Imaginary identities                                                                                     |
| $\rightarrow$               | Denotes a mapping from the space at left side of the symbol to the space at the right side of the symbol |
| $\forall$                   | For all                                                                                                  |
| $<$                         | Less than                                                                                                |
| $\leq$                      | Less than or equal to                                                                                    |
| $>$                         | Greater than                                                                                             |

|                              |                                                                                      |
|------------------------------|--------------------------------------------------------------------------------------|
| $\geq$                       | Greater than or equal to                                                             |
| $\text{Re}(\cdot)$           | Operator that extracts the real part of quaternion                                   |
| $\text{Im}(\cdot)$           | Operator that extracts the imaginary part of quaternion                              |
| $(\cdot)^*$                  | Operator that returns the conjugate of a quaternion                                  |
| $\ \cdot\ $                  | Operator that returns the $\ell^2$ -norm of a quaternion                             |
| $\text{vec}_3$               | Operator that returns a 3D vector given a pure quaternion                            |
| $\ln(\cdot)$                 | Logarithm map                                                                        |
| $\exp(\cdot)$                | Exponential map                                                                      |
| $\cos(\cdot)$                | Cosine                                                                               |
| $\sin(\cdot)$                | Sine                                                                                 |
| $\text{atan2}(\cdot, \cdot)$ | Four quadrants arc-tangent                                                           |
| $\mathbf{tanh}(\mathbf{q})$  | Element-wise hyperbolic tangent for pure quaternions                                 |
| $\Pi_X$                      | 3 by 3 matrix that represents the null space projector in the direction of $\vec{X}$ |
| $\mathbf{p}$                 | Pure quaternion that represents the 3D positions states of a system                  |
| $\mathbf{v}$                 | Pure quaternion that represents the 3D linear velocity states of a system            |
| $\mathbf{o}$                 | Unit quaternion that represents the 3D attitude states of a system                   |
| $\boldsymbol{\omega}$        | Pure quaternion that represents the 3D angular velocity states of a system           |
| $\bar{\mathbf{p}}$           | Pure quaternion that represents the 3D positions states of the nominal system        |
| $\bar{\mathbf{v}}$           | Pure quaternion that represents the 3D linear velocity states of the nominal system  |
| $\bar{\mathbf{o}}$           | Unit quaternion that represents the 3D attitude states of the nominal system         |
| $\bar{\boldsymbol{\omega}}$  | Pure quaternion that represents the 3D angular velocity states of the nominal system |
| $\mathbf{e}_p$               | Pure quaternion that represents the 3D error positions the error system              |

|                              |                                                                                                                                                       |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mathbf{e}_v$               | Pure quaternion that represents the 3D error velocities the error system                                                                              |
| $\mathbf{e}_o$               | Pure quaternion that represents the 3D error orientation the error system                                                                             |
| $\mathbf{e}_\omega$          | Pure quaternion that represents the 3D error angular velocities the error system                                                                      |
| $\mathcal{O}(t)$             | Time-varying pure quaternion set that represents the moving and static obstacles                                                                      |
| $\mathcal{W}(t)$             | Time-varying pure quaternion set that represents the workspace                                                                                        |
| $\mathcal{S}(t)$             | Time-varying pure quaternion set that represents the safe area inside the workspace                                                                   |
| $\partial\mathcal{S}(t)$     | Boundary of $\mathcal{S}(t)$                                                                                                                          |
| $\mathcal{C}(t)$             | Time-varying pure quaternion set that represents the target path to be followed                                                                       |
| $\mathbf{r}(s, t)$           | Parametric representation of $\mathcal{C}(t)$                                                                                                         |
| $\lambda$                    | Positive scalar that defines the safe set $\mathcal{S}(t)$                                                                                            |
| $D_{in}$                     | Positive scalar that defines when to deviate from obstacles                                                                                           |
| $D_{in0}$                    | Positive scalar that defines when to start switching the vector field to deviate from obstacles                                                       |
| $\varphi, \theta, \psi$      | Euler angles                                                                                                                                          |
| $\mathcal{F}(\mathbf{p}, t)$ | Composite vector field to achieve path following and obstacle avoidance                                                                               |
| $\Phi(\mathbf{p}, t)$        | Vector field to achieve path following                                                                                                                |
| $\Psi(\mathbf{p}, t)$        | Vector field to achieve obstacle circumnavigation (avoidance)                                                                                         |
| $\Theta(\mathbf{p}, t)$      | Scalar to weight the composition of vector fields $\Phi(\mathbf{p}, t)$ and $\Psi(\mathbf{p}, t)$ at position $\mathbf{p}$ and time $t$               |
| $s_*(\mathbf{p}, t)$         | parameter of $\mathbf{r}(s, t)$ that minimizes the distance between the system position $\mathbf{p}$ and the target path $\mathcal{C}(t)$ at time $t$ |
| $D(\mathbf{p}, t)$           | Minimum distance between the system position $\mathbf{p}$ and the target path $\mathcal{C}(t)$ at time $t$                                            |
| $T(\mathbf{p}, t)$           | Tangent component of the path at position $\mathbf{p}$ and time $t$                                                                                   |

|                                                              |                                                                                                                       |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| $G, H, \eta$                                                 | Vector field $\Phi(\mathbf{p}, t)$ gains weights to achieve desired norm and compensate for time variance             |
| $G_o, H_o, \eta_o$                                           | Vector field $\Psi(\mathbf{p}, t)$ gains weights to achieve desired norm and compensate for time variance             |
| $\mathbf{p}_{o*}$                                            | Point at obstacle set $\mathcal{O}(t)$ that is closest to the system position $\mathbf{p}$                            |
| $D_o(\mathbf{p}, t)$                                         | Minimum distance between the system position $\mathbf{p}$ and the obstacle set $\mathcal{O}(t)$                       |
| $D_\lambda(\mathbf{p}, t)$                                   | Minimum distance between the system position $\mathbf{p}$ and the obstacle set $\mathcal{O}(t)$ at distance $\lambda$ |
| $T_\lambda(\mathbf{p}, t)$                                   | Tangent component for the obstacle circumnavigation                                                                   |
| $\mathbf{f}$                                                 | Pure quaternion that represents 3D forces control input                                                               |
| $\bar{\mathbf{f}}$                                           | Pure quaternion that represents 3D forces control input for the nominal system                                        |
| $\mathbf{f}_e$                                               | Pure quaternion that represents 3D forces control input for the error system                                          |
| $\mathbf{f}_d$                                               | Pure quaternion that represents 3D disturbance forces                                                                 |
| $\boldsymbol{\tau}$                                          | Pure quaternion that represents 3D torques control input                                                              |
| $\bar{\boldsymbol{\tau}}$                                    | Pure quaternion that represents 3D torques control input for the nominal system                                       |
| $\boldsymbol{\tau}_e$                                        | Pure quaternion that represents 3D torques control input for the error system                                         |
| $\boldsymbol{\tau}_d$                                        | Pure quaternion that represents 3D disturbance torques                                                                |
| $V_p, V_{p,v}$                                               | Lyapunov functions for the nominal position controller                                                                |
| $V_{ep}, V_{epv}$                                            | Lyapunov functions for the error position controller                                                                  |
| $V_o, V_{o,\omega}$                                          | Lyapunov functions for the nominal attitude controller                                                                |
| $V_{eo}, V_{eow}$                                            | Lyapunov functions for the error attitude controller                                                                  |
| $\chi_p, \chi_o$                                             | Lumped terms that include the disturbances                                                                            |
| $\zeta_1, \zeta_2, \zeta_3, \zeta_4$                         | Constants that defines the boundary of lumped terms                                                                   |
| $\hat{\zeta}_1, \hat{\zeta}_2, \hat{\zeta}_3, \hat{\zeta}_4$ | Adaptive system parameters corresponding to $\zeta_1, \zeta_2, \zeta_3, \zeta_4$                                      |

|                                                                      |                                                                                                                                 |
|----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| $\tilde{\zeta}_1, \tilde{\zeta}_2, \tilde{\zeta}_3, \tilde{\zeta}_4$ | Estimation errors between $\zeta_1, \zeta_2, \zeta_3, \zeta_4$ and $\hat{\zeta}_1, \hat{\zeta}_2, \hat{\zeta}_3, \hat{\zeta}_4$ |
| $\rho_1, \rho_2, \rho_4, \rho_4$                                     | Parameters for the adaptive laws                                                                                                |
| $\Delta_p, \Delta_o$                                                 | Ultimate bounds for the proposed controllers                                                                                    |
| $\delta_p, \delta_v, \delta_o, \delta_\omega$                        | Ultimate bounds for the proposed controllers                                                                                    |
| $\mathbf{z}_p$                                                       | Pure quaternion that represents the position error                                                                              |
| $\mathbf{z}_v$                                                       | Pure quaternion that represents the velocity error                                                                              |
| $\mathbf{z}_o$                                                       | Pure quaternion that represents the orientation error                                                                           |
| $\mathbf{z}_\omega$                                                  | Pure quaternion that represents the angular velocity error                                                                      |
| $\kappa$                                                             | Constant value of approximately 0.2785                                                                                          |
| $K_p, K_v$                                                           | Gains for the nominal position controller                                                                                       |
| $k_{\mathcal{F}_e}, k_{ep}, k_{ev}, \epsilon_p$                      | Gains for the robust position controller                                                                                        |
| $\mathcal{F}_e$                                                      | Virtual control law for $\mathbf{e}_v$                                                                                          |
| $\mathbf{o}_d$                                                       | Attitude reference                                                                                                              |
| $\boldsymbol{\omega}_d$                                              | Angular velocity of attitude reference                                                                                          |
| $\bar{\boldsymbol{\varphi}}$                                         | Virtual control law for $\bar{\boldsymbol{\omega}}$                                                                             |
| $K_\varphi, K_o, K_\omega$                                           | Gains for the nominal attitude controller                                                                                       |
| $\mathcal{J}_1, \mathcal{J}$                                         | Optimization functionals                                                                                                        |
| $C_1, D_1, C, D$                                                     | Optimization weight matrices                                                                                                    |
| $\gamma_1, \gamma$                                                   | Guaranteed costs                                                                                                                |
| $k_{\varphi_e}, k_{eo}, k_{e\omega}, \epsilon_o$                     | Gains for the robust position controller                                                                                        |
| $\boldsymbol{\varphi}_e$                                             | Virtual control law for $\mathbf{e}_\omega$                                                                                     |

# Contents

|          |                                                          |           |
|----------|----------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                      | <b>21</b> |
| 1.1      | Contributions summary                                    | 24        |
| 1.2      | Publications                                             | 24        |
| 1.3      | Text organization                                        | 26        |
| <b>2</b> | <b>Literature review</b>                                 | <b>27</b> |
| 2.1      | Vector field guidance and obstacle avoidance             | 27        |
| 2.2      | Attitude control                                         | 29        |
| 2.3      | Quadcopter control                                       | 30        |
| <b>3</b> | <b>Theoretical background</b>                            | <b>34</b> |
| 3.1      | Mathematical notation                                    | 34        |
| 3.2      | Quaternion algebra and rigid-body dynamics               | 34        |
| 3.2.1    | Basic definitions and operations                         | 35        |
| 3.2.2    | Pure quaternions                                         | 36        |
| 3.2.3    | Logarithm and exponential maps                           | 38        |
| 3.2.4    | Description of rigid-body motions                        | 38        |
| 3.2.4.1  | Position and linear velocity                             | 38        |
| 3.2.4.2  | Orientation and angular velocity                         | 39        |
| 3.2.5    | Euler angles                                             | 41        |
| 3.2.6    | Numeric integration                                      | 42        |
| 3.3      | Constructive artificial vector field                     | 43        |
| 3.3.1    | Modeling and problem definition                          | 43        |
| 3.3.2    | Constructive method                                      | 45        |
| 3.3.2.1  | Composite Field                                          | 45        |
| 3.3.2.2  | Curve Field                                              | 47        |
| 3.3.2.3  | Obstacle Field                                           | 48        |
| 3.3.3    | Field properties                                         | 51        |
| <b>4</b> | <b>Position control</b>                                  | <b>52</b> |
| 4.1      | Modeling and problem statement                           | 52        |
| 4.2      | Integrated vector field and Backstepping nominal control | 54        |
| 4.2.1    | Step 1                                                   | 54        |
| 4.2.2    | Step 2                                                   | 55        |
| 4.2.3    | Collision avoidance                                      | 56        |
| 4.3      | Robust adaptive control                                  | 58        |
| 4.3.1    | Step 1                                                   | 58        |
| 4.3.2    | Step 2                                                   | 59        |
| 4.4      | Practical control system                                 | 61        |

|          |                                                                                       |            |
|----------|---------------------------------------------------------------------------------------|------------|
| <b>5</b> | <b>Attitude control</b>                                                               | <b>62</b>  |
| 5.1      | Modeling and problem statement                                                        | 62         |
| 5.2      | Nominal control                                                                       | 63         |
| 5.3      | Robust adaptive control                                                               | 65         |
| 5.3.1    | Step 1                                                                                | 66         |
| 5.3.2    | Step 2                                                                                | 66         |
| 5.3.3    | Practical system control                                                              | 68         |
| <b>6</b> | <b>Results and discussion</b>                                                         | <b>69</b>  |
| 6.1      | Implementation and setup                                                              | 69         |
| 6.1.1    | Notes                                                                                 | 69         |
| 6.1.2    | Simulation code                                                                       | 70         |
| 6.1.3    | Crazyflie 2.1 platform                                                                | 71         |
| 6.1.4    | Adding a custom controller to Crazyflie firmware                                      | 72         |
| 6.1.5    | Experiments architecture                                                              | 72         |
| 6.2      | Position Controller simulations                                                       | 73         |
| 6.2.1    | Position Controller simulation I: disturbance-free and obstacle-free                  | 73         |
| 6.2.2    | Position Controller simulation II: constant disturbances                              | 76         |
| 6.2.3    | Position Controller simulation III: disturbances and obstacles                        | 79         |
| 6.2.4    | Position Controller simulation IV: disturbances and obstacles with nominal controller | 84         |
| 6.3      | Attitude Controller simulations                                                       | 87         |
| 6.3.1    | Attitude Controller simulation I: disturbances                                        | 87         |
| 6.3.2    | Attitude Controller simulation II: disturbances with nominal controller               | 93         |
| 6.4      | Attitude experiments                                                                  | 96         |
| 6.4.1    | Attitude Controller experiment I: origin regulation                                   | 96         |
| 6.4.2    | Attitude Controller experiment II: trajectory-tracking A                              | 97         |
| 6.4.3    | Attitude Controller experiment III: trajectory-tracking B                             | 97         |
| 6.4.4    | Attitude Controller experiments summary and comparison                                | 100        |
| <b>7</b> | <b>Conclusion and future work</b>                                                     | <b>101</b> |
|          | <b>References</b>                                                                     | <b>103</b> |

# Chapter 1

## Introduction

The engineering field of robotics has grown over the past years. One of its goals is to extend human capacities in such a way that the robot's sensors and actuators extend the human senses to accomplish tasks that may be repetitive, exhausting, or complex. From that, robotics also aims to improve humankind's quality of life.

As examples of how robotics are extending human capacities and improving quality of life, we can mention the usage of robots to explore other planets searching for other life forms, and the usage of robots in high-precision assisted surgeries.

Among the variety of models, Unmanned Aerial Vehicles (UAVs) are of great interest. These vehicles can fit in several applications, such as surveillance [Williams and Yakimenko, 2018], monitoring [Duggal et al., 2016], target tracking [Rabah et al., 2019], search and rescue [Eid and Sham Dol, 2019], package delivery [Miranda et al., 2022], and autonomous race [Rezende et al., 2019], among many others.

This work is focused on developing a control strategy for UAVs with four rotors, namely quadcopters, also known as quadrotors or informally called *drones*. They can be used in military or civil applications. The usage of such a vehicle has grown in the past years, especially for their capability of realizing agile autonomous tasks in three-dimensional spaces. There are models ranging from remotely controlled toys to high precision used in international competitions<sup>1</sup>. Figure 1 shows a DJI Matrice 100 quadcopter<sup>2</sup> utilized by UFMG Mechatronics, Control, and Robotics (MACRO) research group.

Some previous works of our research group at UFMG already present control strategies for quadrotors as in [Rezende et al., 2020, Nunes et al., 2023]. Others use them in applications such as delivery [Rocha et al., 2020], and international competitions [Rezende et al., 2019]. The proposed controller in this work aims to evolve the previous controllers regarding safety, performance, and disturbance mitigation.

---

<sup>1</sup> Lockheed Martin Corporation, *Artificial Intelligence Robotic Racing (AIRR) Circuit race*. 2019. Available at [lockheedmartin.com/en-us/news/events/ai-innovation-challenge.html](https://lockheedmartin.com/en-us/news/events/ai-innovation-challenge.html)

<sup>2</sup> [dji.com/br/matrice100](https://dji.com/br/matrice100)



Figure 1 – DJI Matrice 100 quadcopter ([dji.com/br/matrice100](http://dji.com/br/matrice100)) utilized by UFMG Mechatronics, Control, and Robotics (MACRO) group.

Our goal is to drive a quadcopter to follow a given path. Here we consider the path as an input and we do not elaborate on how to design or find one. In general, many quadcopter applications have a clear predefined path-following task to be solved. This is the type of task we solve in this work.

However, we must ensure safety during the robot motion. Then, we introduce an obstacle avoidance feature in the path-following controller, that can be prioritized over the original path-following in such a way it aims to follow the original path only when it is safe.

In the ideal scenario of non-disturbed systems, we develop a nominal controller with guaranteed asymptotic convergence targeting the obstacle avoidance and path-following mentioned previously. However, we must also recognize the practical reality that disturbances are almost inevitable. For this case, we introduce an adaptive control approach to mitigate their effects. The controller employs a robust adaptive control strategy enabling the system to approach the nominal trajectory while bounding the maximum deviation [Sun and Huo, 2013]. This adaptive control technique has proven to be effective in enhancing the robustness and disturbance rejection capabilities of aerial vehicles.

Using both the non-linear and robust control techniques, we combine the nominal performance with the disturbance attenuation. The compromise idea is to design a controller that ensures asymptotic stability and desired tracking performance under nominal conditions, while also possessing the ability to mitigate the effects of disturbances [Mayne et al., 2005, Yang et al., 2008, Sun and Huo, 2013].

The method is constructively proposed. This means a step-by-step development where each block can be analyzed separately. This brings the benefit of being able to apply only part of the controller in other applications or systems.

We start by reviewing a vector field method for path-following with obstacle avoidance [Rezende et al., 2021, Nunes et al., 2022]. The vector field method can fit a wide variety of robot models and tasks since it is a guidance strategy that is concerned with the computation of desired velocities.

Then, our methodology starts with the *Position Controller*, where we control a point mass, *i.e.* a system without attitude that receives forces as input, to follow the path while avoiding obstacles. This controller can be applied to drive a wide variety of mobile robots since the desired forces and velocities are the core of three-dimensional spatial movements.

When taking into account the quadcopter system model, it inherits a rigid-body attitude system, just like many aircrafts and spacecrafts. The control of such rigid-body attitude systems also plays a crucial role in various engineering applications, ranging from spacecraft maneuvering to aerial robotics and autonomous vehicles [Ai et al., 2021, Mok et al., 2018, Jing et al., 2015, Heidarian and Memon, 2012, Li et al., 2018, Iwata et al., 2013].

These systems are characterized by their ability to change the orientation in a three-dimensional space. They are usually prone to the presence of disturbances, which can arise from environmental factors, external forces, or inherent uncertainties in the modeled system dynamics. Achieving accurate tracking by means of robust control of the rigid-body attitude is essential for ensuring stability, performance, and safety in many applications.

Our second controller, *Attitude Controller*, is then an attitude reference tracking control for rigid-body attitude systems. In combination with *Position Controller*, in either an integrated or separated strategy, they can be used to control a quadcopter.

In summary, we propose controllers for quadcopter systems to achieve path-following and attitude tracking, which can be used in several tasks. Our strategy combines path-following with obstacle avoidance while also possessing the ability to mitigate unknown disturbances.

To validate and demonstrate the effectiveness of our proposed control scheme, we conduct computational simulations for both controllers and real-world experiments of the Attitude Controller in a quadcopter. These showcase the performance and robustness of our approach in both path-following and attitude-tracking under uncertainties. The results obtained show the effectiveness of the combined nominal and adaptive control scheme, emphasizing its potential for practical applications.

## 1.1 Contributions summary

The main contributions of this work can be summarized as follows:

- Development of non-linear and robust controllers for quadcopter systems;
- Development of motion planning techniques based on vector fields to follow curves defined in three-dimensional space while avoiding obstacles.
- Simulations of all developed techniques. These simulations are mainly carried out in Python and MATLAB;
- Real-robot experimental results for the attitude control problem.

## 1.2 Publications

In this section, we highlight some publications achieved during the research journey that are related to the work presented in this thesis. The publications are displayed in chronological order with a brief explanation of how they relate to the presented work.

- Rezende, A. M., Gonçalves, V. M., **Nunes, A. H.**, & Pimenta, L. C. (2020, May). Robust quadcopter control with artificial vector fields. In 2020 **IEEE International Conference on Robotics and Automation (ICRA)** (pp. 6381-6387). IEEE.

Development and experiments of quadcopter control in path-following tasks.

- **Nunes, A. H.**, Araújo, Á. R., Alves, M. L., Rezende, A. M., Silva, I. B., & Pimenta, L. C. (2020, December). *Controle de robô para serviços de inspeção industrial utilizando campos vetoriais artificiais variantes no tempo*. In **Congresso Brasileiro de Automática-CBA** (Vol. 2, No. 1).

Development of a vector field in which the path resizes to avoid obstacles.

- Pereira, L. A., **Nunes, A. H.**, Rezende, A. M., Gonçalves, V. M., Raffo, G. V., & Pimenta, L. C. (2021, May). Collision-free vector field guidance and MPC for a fixed-wing UAV. In 2021 **IEEE International Conference on Robotics and Automation (ICRA)** (pp. 176-182). IEEE.

Development of an online switching vector field to follow paths and contour infinite cylinder-like obstacles.

- **Nunes, A. H.**, Rezende, A. M., Cruz, G. P., Freitas, G. M., Gonçalves, V. M., & Pimenta, L. C. (2022, December). Vector field for curve tracking with obstacle avoidance. In **2022 IEEE 61st Conference on Decision and Control (CDC)** (pp. 2031-2038). IEEE.

Development and experiments of the collision-free vector field guidance used in this work.

- **Nunes, A. H.**, Raffo, G. V., & Pimenta, L. C. (2023, May). Integrated vector field and backstepping control for quadcopters. In **2023 IEEE International Conference on Robotics and Automation (ICRA)** (pp. 1256-1262). IEEE.

Development of quadcopter Backstepping control for path-following with constant disturbance rejection and consideration of torques as control inputs.

Figure 2 shows a qualitative comparison of the mentioned publications with this work. From left to right we have the oldest to newest publication. From top to bottom, we have the most related to the least related publications.

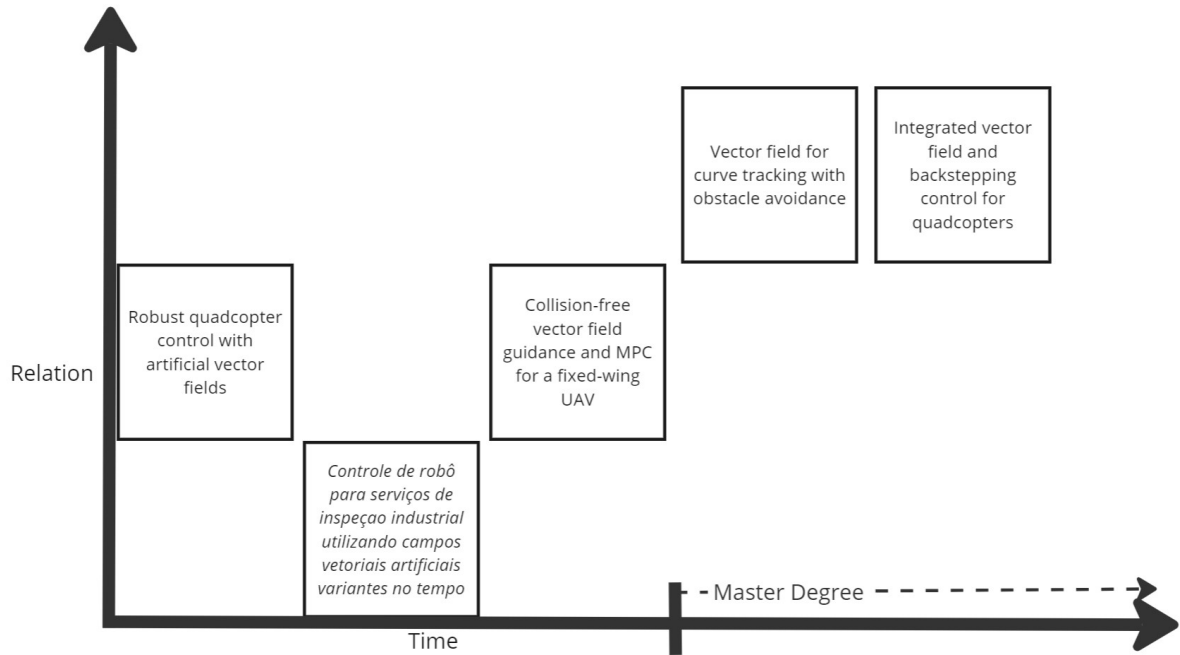


Figure 2 – Qualitative comparison of achieved publications with this work. From left to right we have the oldest to newest publication. From top to bottom, we have the most related to the least related publications.

This work is the continuity of the research where we propose unknown disturbance mitigation with robust adaptive control, and conduct experiments.

## 1.3 Text organization

This text is organized as follows. In the next Chapter, we compare our work with existing controllers by conducting a literature review. Then, Chapter 3 reviews the preliminary core concepts for the proposed methodology. Our methodology follows in the next two Chapters. Chapter 4 presents the Position Controller and Chapter 5 presents the Attitude Controller. In Chapter 6 we conduct simulations and experiments while discussing the method and results. Finally, in Chapter 7 we conclude and propose the continuity of this work.

# Chapter 2

## Literature review

In this chapter, we present a literature review to contextualize our work. Since our methodology is constructively developed, we conduct the review in distinct parts:

- First, we start by reviewing vector field guidance and obstacle avoidance methods, since the first part of our controller consists of a collision-free vector field law proposed in previous works [Rezende et al., 2021, Nunes et al., 2022];
- Then, we review attitude controllers since they can play a separate role in rigid-body attitude system controllers.
- Finally, we review control strategies focused on quadcopters since this is the final goal of the proposed method.

### 2.1 Vector field guidance and obstacle avoidance

Vector field guidance consists of a wide variety of methods that have been proposed in many works such as [Gonçalves et al., 2010a, Yao et al., 2022, 2021, 2019, Lawrence et al., 2008, Nelson et al., 2007, Spong et al., 2005, Choset et al., 2005]. In [Spong et al., 2005, Choset et al., 2005], classical methodologies of artificial vector fields for navigation of mobile robots or manipulators were presented. The main idea is to build a potential attractive field toward the target and another potential repulsive field from obstacles, calculated dynamically based on real-time measurements during navigation. In this way, the vector field is obtained from the negative of the gradient of the sum of these potential fields. By following this vector field the robot navigates from an initial configuration to a final configuration while avoiding obstacles. Such an obstacle avoidance strategy is reactive, *i.e.* it considers an obstacle-free work environment at first and takes actions to avoid obstacles as they are perceived. The problem with this approach is the presence of local minima in the potential function preventing convergence towards the target. Depending on the chosen formulation, there may also be points of singularity, where the field is undefined.

Other strategies aim to address the issue of local minima. In [Spong et al., 2005, Choset et al., 2005], other navigation algorithms free of local minima were also presented. Some of them are based on graph search strategies that analyze a discrete map of the workspace and compute another vector field capable of navigating the robot to the target. These obstacle avoidance strategies are deliberative, that is, they depend on an offline planner that computes a collision-free path before navigation. However, other issues must be considered, such as the need for prior knowledge of the map and the discretization resolution.

In general, reactive strategies are simple and suitable for unknown environments and have a low computational cost. However, they can have undesirable conditions of local and singular minima and generate non-smooth paths. Deliberative strategies can solve these problems, but generally at a higher computational cost and under the assumption of knowledge of the map and the geometry of the obstacles.

In some applications, the reference for the robot is no longer a configuration, but a time-varying path. It is still possible to adapt the classic strategies, such as introducing the usage of *waypoints*, but in general another strategy is required. In [Gonçalves et al., 2010a], it was proposed an artificial vector field strategy for robot navigation on time-varying implicit curves defined in  $n$ -dimensional spaces. Even if the reference is given as a target configuration instead of a curve, this vector field methodology can be easily adapted by making the curve tend towards a circle of zero radius or by imposing only the component of convergence on the field.

For this method, scalar functions  $\alpha_i : \mathbb{R}^n \times \mathbb{R}_+ \rightarrow \mathbb{R}$  are used to define the curve by zero level intersections,  $\alpha_i = 0, \forall i$ . In addition to the ability to perform more general tasks, convergence was proved and the authors performed experiments in two-dimensional and three-dimensional environments in [Gonçalves et al., 2010b] to validate the method. Besides, this method was also applied in other works such as [Nunes et al., 2020, Pereira et al., 2021, Rezende et al., 2020]. However, through this methodology, the curves are implicitly obtained, and it may not be trivial to find the functions  $\alpha_i$ , as they may also have points of singularity.

In [Yao et al., 2021], a vector field for static curve following in  $n$  dimensions represented by a parametric equation was proposed. The proposal extended the  $n$ -dimensional space to  $n + 1$  dimensions and applied the method from [Gonçalves et al., 2010a], resulting in a singularity-free vector field. In related works [Yao et al., 2022, 2019], obstacle avoidance was considered. However, time-varying paths and generic-shaped obstacles were not.

A new constructive vector field for time-varying path-following in  $n$ -dimensions was previously proposed in [Rezende et al., 2021]. In this recent method, the field is obtained constructively with the use of the Euclidean distance function to the curve. It uses a parametric representation of the target curve, an advantage over the intersection of

zero-level sets required in [Gonçalves et al., 2010a]. Several characteristics and advantages of the new approach were presented in [Rezende et al., 2021], including formal proofs.

Nevertheless, the presence of obstacles was not considered in that work. Then, in [Nunes et al., 2022], we extended this field to allow obstacle avoidance in the field formulation by considering obstacle circumnavigation when an imminent collision is detected. The circumnavigation is achieved by using the obstacle closest point to the system. Both [Rezende et al., 2021, Nunes et al., 2022] will be reviewed in the next Chapter.

Furthermore, artificial vector fields often consist of path-following controllers, as an alternative to those that are trajectory-tracking. Depending on the task, path-following strategies overcome trajectory-tracking strategies, as they have a smoother convergence to the path, and demand less effort from the controller, in addition to other advantages as discussed in [Rubí et al., 2020, Aguiar et al., 2005, 2004].

In this work, we use the method from [Rezende et al., 2021] with the obstacle avoidance feature introduced in [Nunes et al., 2022], resulting in a path-following controller with reactive obstacle avoidance.

## 2.2 Attitude control

In recent years, extensive research has been conducted to address the challenges associated with controlling rigid-body attitude systems. Traditional methods for regulation problems, such as PID and LQR controllers, have been widely employed to achieve nominal performance under ideal conditions [Bouabdallah et al., 2004, Bao et al., 2017]. However, these approaches are often based on linearized models, which leads to poor performance far from the operational point and does not allow aggressive maneuvers. Additionally, controllers designed only for ideal conditions often struggle to maintain satisfactory performance when subjected to disturbances.

To overcome those limitations, advanced control schemes have been proposed, aiming to combine nominal performance with disturbance attenuation capabilities using both non-linear and robust control techniques. The compromise idea is to design controllers that ensure asymptotic stability and desired tracking performance under nominal conditions, while also possessing the ability to mitigate the effects of disturbances [Sun and Huo, 2013, Mayne et al., 2005, Yang et al., 2008].

In this work, we present a non-linear robust control scheme for rigid-body attitude systems that addresses the challenges of tracking orientation references under unknown magnitude-bounded disturbances. The proposed control scheme leverages the combination of a nominal controller and a robust adaptive control approach for disturbances, enabling the system to achieve both desired performance and robustness in the face of uncertainties.

In this sense, the proposed method is similar to the work [Zhang et al., 2021], which inspired the current development. The authors proposed a guaranteed cost control with a robust adaptive auxiliary controller to solve the origin regulation problem. However, their solution cannot solve the reference tracking problem. To overcome this limitation, we use a non-linear control for trajectory tracking, combined with the robust Backstepping control. The Backstepping approach has been widely used in non-linear control, especially for aircrafts that inherit a rigid-body attitude system by design [Salierno, 2018, Salierno and Raffo, 2017, Bouabdallah and Siegwart, 2005, Huang et al., 2010, Raptis et al., 2011, Tan et al., 2016].

Our work also differs in the orientation error calculation leading to a more simple and effective robust adaptive control law.

## 2.3 Quadcopter control

For mobile robots in general, vector field guidance methods are usually used as a higher-level outer control layer, such as those previously mentioned. Then, it is necessary to combine the guidance with low-level controllers, specific for each robot model, to be able to drive the real vehicles. Usually, this results in a cascade scheme. In a previous work [Rezende et al., 2020], we proposed a cascade path-following control for quadcopters, in which the vector field from [Gonçalves et al., 2010a] was used in a guidance layer. The controller did not consider collision avoidance or uncertainty mitigation. The quadcopter model was considered to receive the total thrust and angular velocities as inputs, assuming an acro/rate control mode<sup>1</sup>.

These cascade strategies, where the guidance is placed as an outer control layer, are known as Separate Guidance and Controller (SGC). In the case of SGC methodologies, the previous works [Nunes et al., 2020, Pereira et al., 2021, Rezende et al., 2020] can be cited again. They used the artificial vector fields of [Gonçalves et al., 2010a] to guide different vehicles, as we explain next.

In [Nunes et al., 2020], the goal was to perform a kinematic control of a non-holonomic differential mobile robot in two-dimensional space. For the low-level controller, a state feedback linearization strategy was used. In addition, the curve was conditioned to a variation in time in order to allow the deviation of unknown and limited obstacles from a contraction and expansion of its size. Figure 3 is a result of the work [Nunes et al., 2020], which shows the robot's path to the region between the curves while avoiding obstacles.

---

<sup>1</sup> Acro mode, also known as rate mode, is a control mode for multirotors where the vehicle inputs are the total thrust and the angular velocities, [oscarliang.com/rate-acro-horizon-flight-mode-level](https://oscarliang.com/rate-acro-horizon-flight-mode-level).

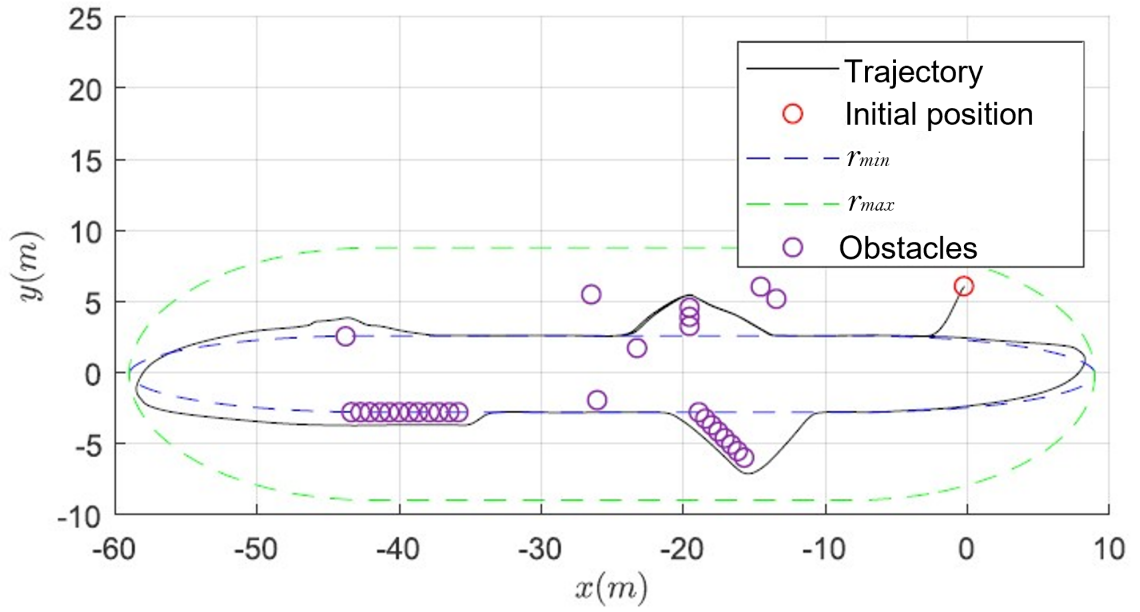


Figure 3 – Result from [Nunes et al., 2020] which shows the robot's trajectory for the region between the curves while avoiding obstacles. We translated the original image into English.

In [Pereira et al., 2021], optimal control techniques were combined with feedback linearization to control a fixed-wing UAV. Furthermore, the vector field could be smoothly switched in order to perform unknown dynamic (reactive) obstacle avoidance under the premise of having the shape of infinite vertical cylinders. Figure 4 is a result of the work [Pereira et al., 2021], which shows the trajectory of the vehicle to the curve while dodging dynamic obstacles.

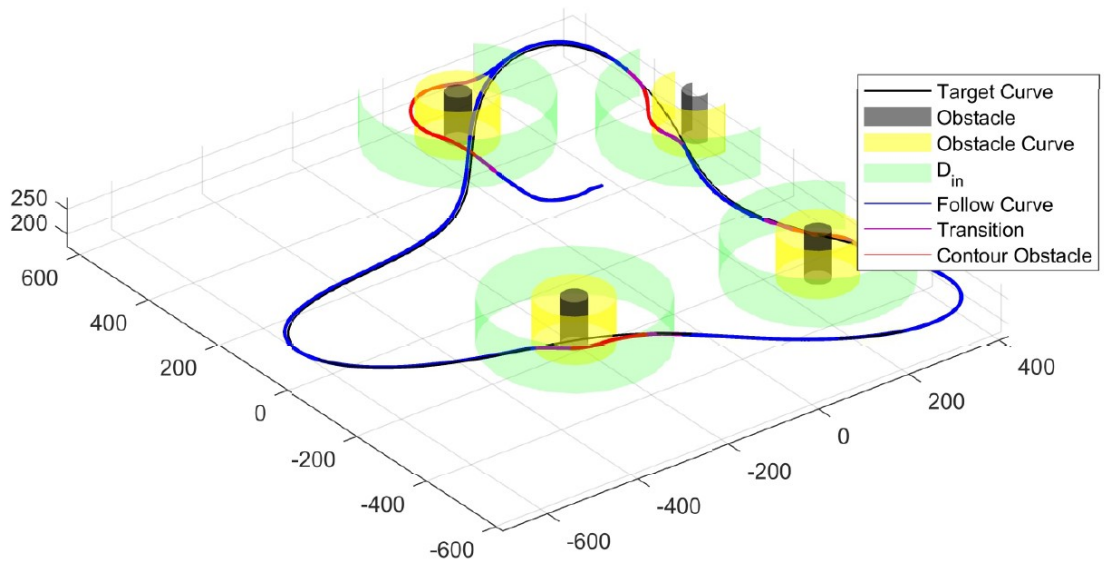


Figure 4 – Result from [Pereira et al., 2021] which shows the robot's trajectory for the path while avoiding dynamic obstacles.

The obstacle avoidance developed in this work is similar to the two previously mentioned ones since they consist of modifying the vector field at runtime through a reactive strategy. In particular, just like [Pereira et al., 2021], the detour here will take place along the contour of the nearest perceived obstacle. However, the obstacle format will not be restricted.

In the work [Rezende et al., 2020], a constructive methodology was used for the development of a quadrotor UAV controller, proving its robustness through input-output stability arguments. Figure 5 is a result of the work [Rezende et al., 2020], which shows the convergence of the robot to a limiting set of the target curve.

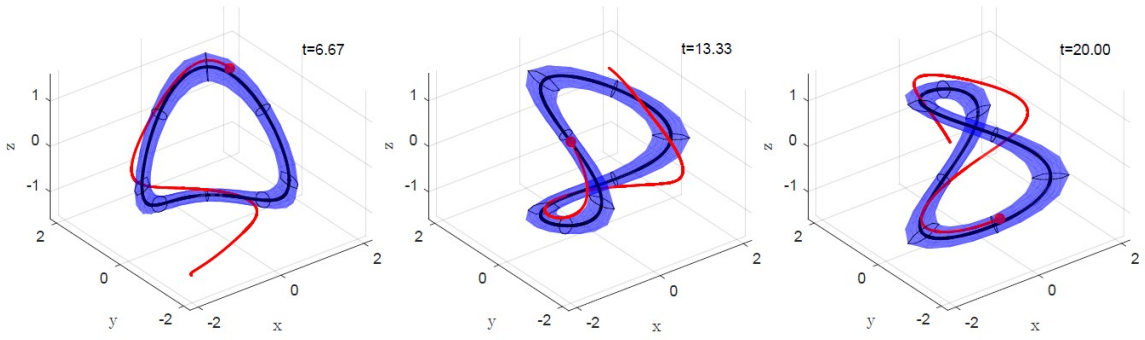


Figure 5 – Result from [Rezende et al., 2020] which shows the convergence of the robot to a boundary set of the target curve.

As opposed to the widely used SGC strategies where the guidance is placed as a separate layer from the rest of the controller, an alternative is to use an Integrated Guidance and Controller (IGC) scheme.

In [Nunes et al., 2023], we extended our previous results [Rezende et al., 2020] in an IGC strategy by: (i) integrating the new collision-free vector field [Nunes et al., 2022] in a Backstepping control; (ii) assuming a model with lower level inputs with torques rather than angular velocities; and (iii) adding constant uncertainty rejection via integral action.

Now, in this work, we further extend our previous developments by (i) adding a robust adaptive control to mitigate any unknown uncertainty rather than only rejecting constant ones; (ii) presenting analysis based on Lyapunov theory; and (iii) presenting real vehicle experiments of the attitude part.

Several other controllers for quadcopter vehicles have already been proposed in the literature [dos Santos et al., 2024, Pereira et al., 2019, Rubí et al., 2020, Iacono and Sgorbissa, 2018, Wilhelm and Clem, 2019, Mellinger and Kumar, 2011, Lee et al., 2010, Zhou and Schwager, 2014, Zhao et al., 2018]. In this work, we propose a novel quadcopter controller to allow path-following and obstacle avoidance in the presence of unknown uncertainties, such as external disturbances or inaccurate model parameters. Our method is based on artificial vector field guidance strategy [Rezende et al., 2021, Nunes et al.,

2022] and a Backstepping control [Salierno, 2018, Salierno and Raffo, 2017, Khalil, 2002, Skjetne and Fossen, 2004] with robust adaptive action [Sun and Huo, 2013].

Backstepping is a well-known non-linear control technique [Khalil, 2002] which has advantages over linear approaches, such as PID and LQR [Rubí et al., 2020, Bouabdallah et al., 2004, Özbek et al., 2016, Cowling et al., 2007], and has been successfully applied to quadcopters [Salierno, 2018, Bouabdallah and Siegwart, 2005, Huang et al., 2010, Raptis et al., 2011, Tan et al., 2016]. Generally, Backstepping controllers in quadrotor control applications have better performance in terms of distance to path and yaw error, as well as better off-path behavior and at high speeds, at the cost of greater computational and control efforts [Rubí et al., 2020].

Unlike previous works, here we use Backstepping as a path controller while also allowing obstacle avoidance. Both are accomplished by the usage of the vector field guidance in the controller. Furthermore, we incorporate robust adaptive control to mitigate the effect of disturbances.

# Chapter 3

## Theoretical background

In this Chapter, we review preliminary concepts that are required to state the methodology. We start with a brief description of the mathematical notation used in this work. Then, we review key concepts and operations regarding quaternion algebra and how they describe rigid-body motions. Finally, we review the collision-free constructive vector field guidance method proposed in [Rezende et al., 2021, Nunes et al., 2022].

### 3.1 Mathematical notation

Quaternions are represented in bold, as in  $\mathbf{q}$ . Vectors are represented with arrows, as in  $\vec{q}$ . Capital letters usually denote matrices, as in  $M$ , the exceptions are the Lyapunov functions that use  $V$ . When applied to matrices, the inequalities  $\geq 0, > 0, \leq 0, < 0$  denote if the matrices are (semi)positive definite or (semi)negative definite.

### 3.2 Quaternion algebra and rigid-body dynamics

Quaternion algebra is used in this work to represent the quantities of position, linear velocity, orientation, and angular velocity. The main advantages of this algebra are the singularity-free representation, the easier coupling between complex systems, and similarities with vector representations [Afonso Silva et al., 2022, Jia, 2019, Vilhena Adorno, 2017, Kuipers, 1999].

In the context of the controller designed in this work, the quaternions play an important role in finding the attitude control laws, since they have useful properties when it comes to orientation dynamics. For the position dynamics, we mainly use pure quaternions which can be directly replaced by 3D vectors without loss in meaning or generality. However, we chose the quaternions in both attitude and position dynamics for a more concise algebra throughout the description of the methodology. This potentially

eases a further extension to more complex systems, such as aerial manipulators [Abaunza et al., 2017].

### 3.2.1 Basic definitions and operations

We review next some basic definitions regarding the quaternions [Vilhena Adorno, 2017].

**Definition 1** (Quaternion set). *The quaternion set  $\mathbb{H}$  is formally defined by*

$$\mathbb{H} = \{\mathbf{h} = h_0 + \hat{\mathbf{i}}h_1 + \hat{\mathbf{j}}h_2 + \hat{\mathbf{k}}h_3 : h_0, h_1, h_2, h_3 \in \mathbb{R}\}, \quad (3.1)$$

where  $\hat{\mathbf{i}}, \hat{\mathbf{j}},$  and  $\hat{\mathbf{k}}$  are the imaginary identities defined as

$$\hat{\mathbf{i}}^2 = \hat{\mathbf{j}}^2 = \hat{\mathbf{k}}^2 = \hat{\mathbf{i}}\hat{\mathbf{j}}\hat{\mathbf{k}} = -1, \quad (3.2)$$

which can be related to each other according to

$$\hat{\mathbf{i}} = \hat{\mathbf{j}}\hat{\mathbf{k}} = -\hat{\mathbf{k}}\hat{\mathbf{j}}, \quad (3.3)$$

$$\hat{\mathbf{j}} = \hat{\mathbf{k}}\hat{\mathbf{i}} = -\hat{\mathbf{i}}\hat{\mathbf{k}}, \quad (3.4)$$

$$\hat{\mathbf{k}} = \hat{\mathbf{i}}\hat{\mathbf{j}} = -\hat{\mathbf{j}}\hat{\mathbf{i}}. \quad (3.5)$$

Let  $\mathbf{h} = h_0 + \hat{\mathbf{i}}h_1 + \hat{\mathbf{j}}h_2 + \hat{\mathbf{k}}h_3$  and  $\mathbf{q} = q_0 + \hat{\mathbf{i}}q_1 + \hat{\mathbf{j}}q_2 + \hat{\mathbf{k}}q_3$  be two quaternions used for the following operators definitions.

**Definition 2** (Quaternion real part, imaginary part and conjugate operators). *Let  $\text{Re}(\cdot)$  be the real part operator,  $\text{Im}(\cdot)$  be the imaginary part operator, and  $(\cdot)^*$  be the conjugate operator. They represent the following operations in a quaternion*

$$\text{Re}(\mathbf{h}) = h_0, \quad (3.6)$$

$$\text{Im}(\mathbf{h}) = \hat{\mathbf{i}}h_1 + \hat{\mathbf{j}}h_2 + \hat{\mathbf{k}}h_3, \quad (3.7)$$

$$\mathbf{h}^* = \text{Re}(\mathbf{h}) - \text{Im}(\mathbf{h}). \quad (3.8)$$

**Definition 3** (Quaternion sum). *The quaternion sum is defined as the sum of each component as in*

$$\mathbf{h} + \mathbf{q} = h_0 + q_0 + \hat{\mathbf{i}}(h_1 + q_1) + \hat{\mathbf{j}}(h_2 + q_2) + \hat{\mathbf{k}}(h_3 + q_3). \quad (3.9)$$

**Definition 4** (Quaternion product). *The product between two quaternions is defined by*

$$\begin{aligned} \mathbf{h}\mathbf{q} = & h_0q_0 - h_1q_1 - h_2q_2 - h_3q_3 \\ & + \hat{\mathbf{i}}(h_0q_1 + h_1q_0 + h_2q_3 - h_3q_2) \\ & + \hat{\mathbf{j}}(h_0q_2 - h_1q_3 + h_2q_0 + h_3q_1) \\ & + \hat{\mathbf{k}}(h_0q_3 + h_1q_2 - h_2q_1 + h_3q_0). \end{aligned} \quad (3.10)$$

*It can also be expressed in terms of scalar and cross products between the real and imaginary parts of the quaternions. To show that, consider an alternative notation for a quaternion  $\mathbf{h} = h_0 + \hat{\mathbf{i}}h_1 + \hat{\mathbf{j}}h_2 + \hat{\mathbf{k}}h_3$  expressed as*

$$\mathbf{h} = \begin{bmatrix} h_0 \\ \vec{h} \end{bmatrix}, \quad \vec{h} = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}^T. \quad (3.11)$$

*Then, equation (3.10) can be expressed as*

$$\mathbf{h}\mathbf{q} = \begin{bmatrix} h_0q_0 - \vec{h} \cdot \vec{q} \\ h_0\vec{q} + q_0\vec{h} + \vec{h} \times \vec{q} \end{bmatrix}. \quad (3.12)$$

*This alternative notation will not be used in this work. Its purpose is just to state another way to express the quaternion product.*

Note that the quaternion product has the associative property, but not the commutative property.

**Definition 5** (Quaternion norm). *Let the  $\ell^2$ -norm be the norm operator for quaternions. It is defined as*

$$\|\mathbf{h}\| \equiv \|\mathbf{h}\|_2 = \sqrt{h_0^2 + h_1^2 + h_2^2 + h_3^2} = \sqrt{\mathbf{h}\mathbf{h}^*} = \sqrt{\mathbf{h}^*\mathbf{h}}. \quad (3.13)$$

**Definition 6** (Quaternion inverse). *The inverse of a quaternion is such that  $\mathbf{h}^{-1}\mathbf{h} = \mathbf{h}\mathbf{h}^{-1} = 1$ ,  $\forall \mathbf{h} \neq 0$ . It can be found by*

$$\mathbf{h}^{-1} = \frac{\mathbf{h}^*}{\|\mathbf{h}\|^2}. \quad (3.14)$$

### 3.2.2 Pure quaternions

The subset of quaternions that has a null real part has useful properties and similarities with 3D vectors that are explored in sequence.

**Definition 7** (Pure quaternions set). *Let  $\mathbb{H}_p \subset \mathbb{H}$  be the subset of quaternions that has a null real part, named pure quaternions or vectorial quaternions. They are formally defined as*

$$\mathbb{H}_p = \{\mathbf{h} \in \mathbb{H} : \text{Re}(\mathbf{h}) = 0\}. \quad (3.15)$$

This set has correspondence with the 3D vector space  $\mathbb{R}^3$ . For that reason, a pure quaternion can be visualized in the same way as a 3D vector and it can be used in the same applications. A map between pure quaternions to 3D space is formally defined next.

**Definition 8** ( $\text{vec}_3$  map). *Let  $\mathbf{u} = \hat{\mathbf{i}}u_1 + \hat{\mathbf{j}}u_2 + \hat{\mathbf{k}}u_3 \in \mathbb{H}_p$  be a pure quaternion and  $\vec{u} = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T \in \mathbb{R}^3$  be the corresponding 3D vector. The map from the pure quaternion to the 3D vector (with its inverse) is formally defined by*

$$\text{vec}_3 : \mathbb{H}_p \rightarrow \mathbb{R}^3, \quad (3.16)$$

$$\vec{u} = \text{vec}_3(\mathbf{u}), \quad (3.17)$$

$$\mathbf{u} = \text{vec}_3^{-1}(\vec{u}). \quad (3.18)$$

**Definition 9** (Pure quaternion cross and dot products). *For all  $\mathbf{u}, \mathbf{v} \in \mathbb{H}_p$ , the cross and dot products are defined by*

$$\mathbf{u} \times \mathbf{v} = \frac{\mathbf{u}\mathbf{v} - \mathbf{v}\mathbf{u}}{2}, \quad (3.19)$$

$$\mathbf{u} \cdot \mathbf{v} = -\frac{\mathbf{u}\mathbf{v} + \mathbf{v}\mathbf{u}}{2}. \quad (3.20)$$

Recall Definitions 4 and 8 to note that performing cross and dot products operations with pure quaternions is the same as performing them in the 3D corresponding space, *i.e.*  $\mathbf{u} \times \mathbf{v} = \text{vec}_3^{-1}(\text{vec}_3(\mathbf{u}) \times \text{vec}_3(\mathbf{v}))$  and  $\mathbf{u} \cdot \mathbf{v} = \text{vec}_3(\mathbf{u}) \cdot \text{vec}_3(\mathbf{v})$ .

Using this similarity and for the sake of simplicity, we pre-multiply a pure quaternion by a  $3 \times 3$  matrix with the same interpretation for the analog case with 3D vectors. Consider the following definition.

**Definition 10** (Product between matrix and pure quaternion). *For all  $A \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{x} \in \mathbb{H}_p$ , the following equalities hold*

$$A\mathbf{x} \equiv \text{vec}_3^{-1}(A\text{vec}_3(\mathbf{x})), \quad (3.21)$$

$$\mathbf{x} \cdot A\mathbf{x} \equiv \text{vec}_3(\mathbf{x})^T A\text{vec}_3(\mathbf{x}). \quad (3.22)$$

### 3.2.3 Logarithm and exponential maps

The logarithm and exponential maps can be defined in quaternion space as follows.

**Definition 11** (Quaternion logarithm).

$$\ln(\mathbf{h}) = \ln(\|\mathbf{h}\|) + \frac{\text{Im}(\mathbf{h})}{\|\text{Im}(\mathbf{h})\|} \cos^{-1} \left( \frac{\text{Re}(\mathbf{h})}{\|\mathbf{h}\|} \right). \quad (3.23)$$

**Definition 12** (Quaternion exponential).

$$\exp(\mathbf{h}) = \exp(\text{Re}(\mathbf{h})) \left( \cos(\|\text{Im}(\mathbf{h})\|) + \frac{\text{Im}(\mathbf{h})}{\|\text{Im}(\mathbf{h})\|} \sin(\|\text{Im}(\mathbf{h})\|) \right). \quad (3.24)$$

They have the useful property of manipulating the axis and angle of a unit quaternion that represents an attitude as will be shown soon in (3.29) and (3.30).

### 3.2.4 Description of rigid-body motions

The ultimate goal of the quaternion algebra usage in this work is to describe rigid-body motions in such a way that we are allowed to propose controllers that ultimately will generate control actions to manipulate them. To achieve that, the quaternions describe the quantities of position, linear velocity, orientation, and angular velocity, as well as their derivatives.

#### 3.2.4.1 Position and linear velocity

Consider a three-dimensional point. Its position can be denoted by a pure quaternion  $\mathbf{p} \in \mathbb{H}_p$  given by

$$\mathbf{p} = \hat{\mathbf{i}}p_x + \hat{\mathbf{j}}p_y + \hat{\mathbf{k}}p_z, \quad \mathbf{p} \in \mathbb{H}_p. \quad (3.25)$$

The sub-indices  $x,y,z$  were chosen to ease the analogy with the well-known three-dimensional Cartesian space with  $xyz$  axis. In the pure quaternion case,  $\hat{\mathbf{i}}$  replaces the  $x$ -axis,  $\hat{\mathbf{j}}$  the  $y$ -axis, and  $\hat{\mathbf{k}}$  the  $z$ -axis. Hence, we can denote a 3D position simply by defining  $\mathbf{p}$  as in (3.25).

To instantaneously move a point from one position  $\mathbf{p}_1$  to another position  $\mathbf{p}_2$  we can sum up the initial position from the amount that we want to move  $\mathbf{p}_2 = \mathbf{p}_1 + \delta\mathbf{p}$ . Similarly, if we want the difference between two positions we can simply subtract the positions  $\delta\mathbf{p} = \mathbf{p}_2 - \mathbf{p}_1$ .

If this particular point with position  $\mathbf{p}$  moves in the space with respect to time, it means that it has a (linear) velocity. Let  $\mathbf{v} \in \mathbb{H}_p$  denote this velocity as

$$\mathbf{v} = \hat{\mathbf{i}}v_x + \hat{\mathbf{j}}v_y + \hat{\mathbf{k}}v_z, \quad \mathbf{v} \in \mathbb{H}_p. \quad (3.26)$$

This velocity is the rate of change of position such that  $\dot{\mathbf{p}} = \mathbf{v}$ .

Both  $\mathbf{p}$  and  $\mathbf{v}$  are given with respect to an inertial reference frame. There are no practical differences in using quaternion algebra or 3D vectors for  $\mathbf{p}$  and  $\mathbf{v}$  so far. As mentioned earlier, the main goal of using quaternions for position and linear velocity is for algebra consistency since this algebra defines the attitude dynamics in this work.

An example of values for  $\mathbf{p}$  and  $\mathbf{v}$  for a moving point is shown in Figure 6.

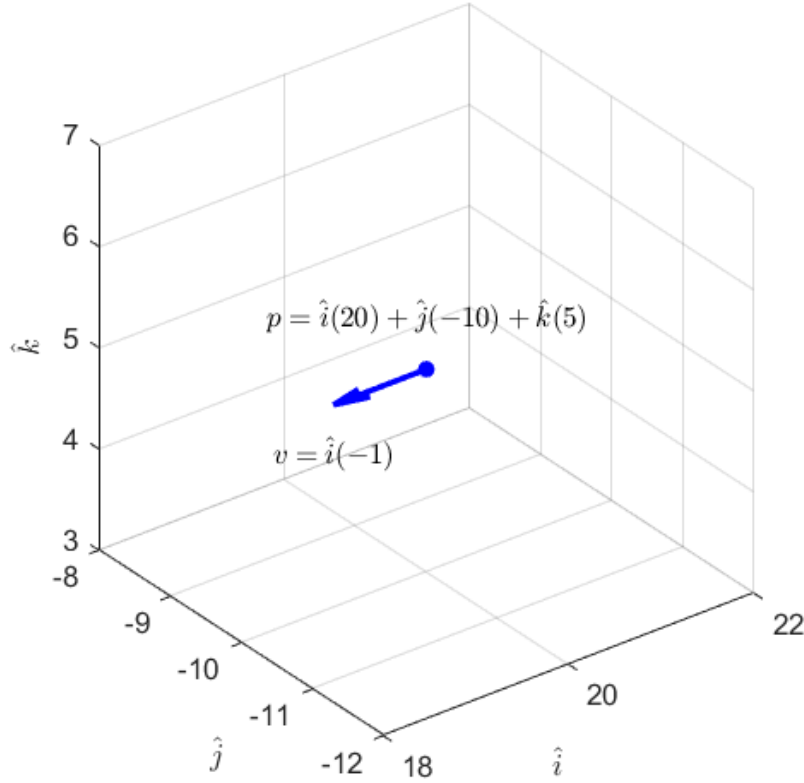


Figure 6 – Visualization of a particle or point with three-dimensional position  $\mathbf{p} \in \mathbb{H}_p$  and linear velocity  $\mathbf{v} \in \mathbb{H}_p$ .

### 3.2.4.2 Orientation and angular velocity

Consider a rigid body in a free space. We now care only about its orientation, regardless of its position. To represent its orientation we can use a unit quaternion. Let  $\mathbf{o} \in \mathbb{H}$ ,  $\|\mathbf{o}\| = 1$  be this unit quaternion that represents the orientation.

A useful way of understanding and composing  $\mathbf{o}$  is by a rotation of amount  $\phi \in \mathbb{S}$  around an axis  $\mathbf{n} \in \mathbb{H}_p$ ,  $\|\mathbf{n}\| = 1$  such that

$$\mathbf{o} = \cos\left(\frac{\phi}{2}\right) + \mathbf{n} \sin\left(\frac{\phi}{2}\right). \quad (3.27)$$

Note that  $\mathbf{n}$  denotes an axis since all pure quaternions with unit norm define a direction in space, in which the imaginary identities  $\hat{\mathbf{i}}, \hat{\mathbf{j}}$ , and  $\hat{\mathbf{k}}$  are special cases that compose the canonical axis.

To rotate a point  $\mathbf{p}$  or any quantity that is represented according to the inertial frame to the frame that has orientation  $\mathbf{o}$ , *i.e.* find  $\mathbf{p}_B$  expressed in the body frame, we perform the following computation

$$\mathbf{p}_B = \mathbf{o}^* \mathbf{p} \mathbf{o}. \quad (3.28)$$

Note that  $\mathbf{o}^* = \mathbf{o}^{-1}$  since  $\|\mathbf{o}\| = 1$ .

To combine two rotations  $\mathbf{o}_1$  and  $\mathbf{o}_2$  we simply multiply them. However, the product order affects the rotation order. To rotate by  $\mathbf{o}_1$  and then by  $\mathbf{o}_2$  all with respect to the inertial frame, we use  $\mathbf{o}_2 \mathbf{o}_1$ . Reversing the order  $\mathbf{o}_1 \mathbf{o}_2$  means rotating by  $\mathbf{o}_2$  and then by  $\mathbf{o}_1$  with respect to the inertial frame or it can also be interpreted as rotation by  $\mathbf{o}_1$  and then rotating by  $\mathbf{o}_2$  with respect to the body frame.

From (3.27), we see that a rotation of  $-\phi$  is simply  $\mathbf{o}^*$ . Combining with the rotation composition procedure, if we want to find the difference or error from one orientation to another we compute  $\delta \mathbf{o} = \mathbf{o}_2^* \mathbf{o}_1$ .

If we apply the logarithm map from Definition 11 in an orientation unit quaternion we get another representation for the orientation as a pure quaternion given by

$$\ln(\mathbf{o}) = \mathbf{n} \frac{\phi}{2}. \quad (3.29)$$

Similarly, if we apply the exponential map from Definition 12 we recover the orientation as unit quaternion:

$$\exp\left(\mathbf{n} \frac{\phi}{2}\right) = \mathbf{o}. \quad (3.30)$$

If the rigid body rotates with respect to time, it means that it has an angular velocity. Let  $\boldsymbol{\omega} \in \mathbb{H}_p$  be the angular velocity of the body-fixed reference frame w.r.t. the inertial reference frame, expressed in the body-fixed reference frame.

$$\boldsymbol{\omega} = \hat{\mathbf{i}}\omega_x + \hat{\mathbf{j}}\omega_y + \hat{\mathbf{k}}\omega_z, \quad \boldsymbol{\omega} \in \mathbb{H}_p. \quad (3.31)$$

This velocity is the rate of change of orientation such that  $\dot{\mathbf{o}} = \frac{1}{2} \mathbf{o} \boldsymbol{\omega}$ .

We adopt the usual representation of angular velocity given with respect to the body frame and the orientation describes the transformation from the inertial frame to the body frame.

While it is possible to directly visualize the quantity  $\boldsymbol{\omega}$ , we cannot do the same for the quantity  $\mathbf{o}$ . However, we still can interpret it geometrically. Figure 7 shows an

example where two frames are plotted in RGB with an angular displacement between them. This displacement or orientation is  $\phi = \frac{\pi}{4}$ , drawn by dashed black lines, around an axis  $\mathbf{n} = \hat{\mathbf{i}} \frac{1}{\sqrt{2}} + \hat{\mathbf{j}} \frac{1}{\sqrt{2}}$  drawn by a black arrow. From  $\phi$  and  $\mathbf{n}$  we obtain  $\mathbf{o}$ , (3.27). The angular velocity  $\boldsymbol{\omega}$  is drawn with dotted black rings around the body (rotated) frame.

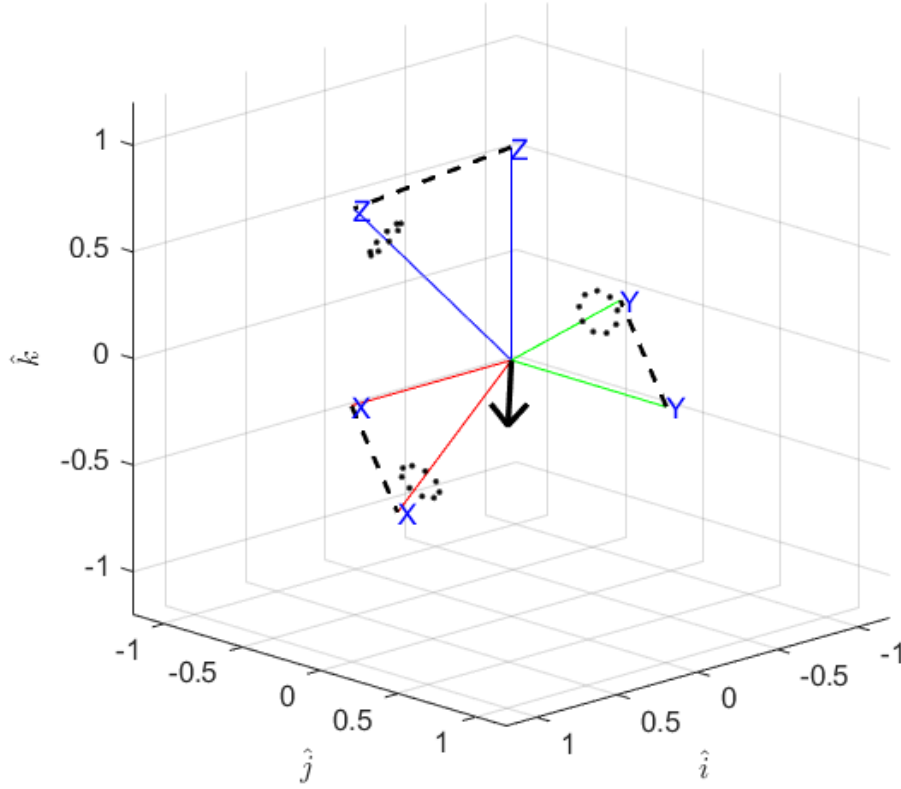


Figure 7 – Geometric interpretation of  $\mathbf{o} \in \mathbb{H}$ , given by an angular displacement of  $\phi = \frac{\pi}{4}$ , illustrated by dashed black lines, around an axis  $\mathbf{n} = \hat{\mathbf{i}} \frac{1}{\sqrt{2}} + \hat{\mathbf{j}} \frac{1}{\sqrt{2}}$  illustrated by a black arrow. Geometric visualization of  $\boldsymbol{\omega} \in \mathbb{H}_p$  illustrated by dotted black rings around the body (rotated) frame.

### 3.2.5 Euler angles

A well-known minimum representation for orientation is the set of three Euler angles. The main problem with this representation is the singularities that it introduces. Still, in some cases, it might be useful to have the Euler angles representation.

In the case of this work, we use the Euler angles only when we want to show the orientation in a more human-readable graph. Euler angles are not used for the controller design, keeping the controller entirely in quaternion algebra.

The conversion between unit quaternion  $\mathbf{o}$  to Euler angles  $\varphi$ ,  $\theta$ ,  $\psi$  is given by<sup>1</sup>

$$\begin{aligned}\mathbf{o} &= \cos\left(\frac{\phi}{2}\right) + (\hat{\mathbf{i}}n_x + \hat{\mathbf{j}}n_y + \hat{\mathbf{k}}n_z) \sin\left(\frac{\phi}{2}\right) = o_0 + \hat{\mathbf{i}}o_1 + \hat{\mathbf{j}}o_2 + \hat{\mathbf{k}}o_3, \\ \varphi &= \text{atan2}\left(2(o_0o_1 + o_2o_3), 1 - 2(o_1^2 + o_2^2)\right), \\ \theta &= -\frac{\pi}{2} + 2\text{atan2}\left(\sqrt{1 + 2(o_0o_2 - o_1o_3)}, \sqrt{1 - 2(o_0o_2 - o_1o_3)}\right), \\ \psi &= \text{atan2}\left(2(o_0o_3 + o_1o_2), 1 - 2(o_2^2 + o_3^2)\right),\end{aligned}\tag{3.32}$$

in which  $\text{atan2}(\cdot, \cdot)$  is the four-quadrant arctangent.

The Euler angles correspond to a sequence of rotations applied one after the other, such that the current rotation is performed with respect to the current frame. The convention in Aerospace Engineering to describe the orientation of a vehicle starts with the body-fixed frame axes initially aligned with the Inertial Reference Frame axes. Then we perform the following rotations:

1. Rotation by  $\psi$  around  $\hat{\mathbf{k}}_B$  or body- $z$ -axis, initially aligned with the reference  $\hat{\mathbf{k}}$  or  $z$ -axis;
2. Rotation by  $\theta$  around  $\hat{\mathbf{j}}_B$  or body- $y$ -axis, obtained after the previous rotation;
3. Rotation by  $\varphi$  around  $\hat{\mathbf{i}}_B$  or body- $x$ -axis, obtained after the previous two rotations.

The inverse conversion is given by

$$\mathbf{o} = \left(\cos\left(\frac{\psi}{2}\right) + \hat{\mathbf{k}} \sin\left(\frac{\psi}{2}\right)\right) \left(\cos\left(\frac{\theta}{2}\right) + \hat{\mathbf{j}} \sin\left(\frac{\theta}{2}\right)\right) \left(\cos\left(\frac{\varphi}{2}\right) + \hat{\mathbf{i}} \sin\left(\frac{\varphi}{2}\right)\right).\tag{3.33}$$

### 3.2.6 Numeric integration

When dealing with unit quaternions in simulations or experiments, we need to be cautious with the integration process.

At first, one can try to adopt a first-order explicit Euler integration method for  $\mathbf{o}$ , such as

$$\dot{\mathbf{o}} \approx \frac{\mathbf{o}_k - \mathbf{o}_{k-1}}{\Delta t},\tag{3.34}$$

$$\mathbf{o}_k \approx \mathbf{o}_{k-1} + \Delta t \dot{\mathbf{o}},\tag{3.35}$$

where  $\Delta t = t_k - t_{k-1}$  is the integration step.

The main problem with (3.35) for a unit quaternion is the difficulty of ensuring the unit norm at each step that uses a sum.

Other approaches can attempt to normalize  $\mathbf{o}_k$  at each step or add real parts to the pure quaternion  $\boldsymbol{\omega}$  (that will no longer be pure) to compensate for the norm change when finding  $\dot{\mathbf{o}}$ .

---

<sup>1</sup> Some implementations may use an equivalent expression for  $\theta$  given by  $\theta = \sin^{-1}(2(o_0o_2 - o_1o_3))$ .

A solution recommended by Vilhena Adorno [2017] is to use exponential integration such that the unit norm is maintained. The exponential integration method is as follows:

$$\frac{\Delta t \boldsymbol{\omega}}{2} = \frac{\Delta \phi}{2} \mathbf{n}, \quad (3.36)$$

$$\mathbf{o}_\Delta = \exp\left(\frac{\Delta t \boldsymbol{\omega}}{2}\right), \quad (3.37)$$

$$\mathbf{o}_k = \mathbf{o}_{k-1} \mathbf{o}_\Delta, \quad (3.38)$$

where  $\Delta \phi = \phi_k - \phi_{k-1}$  and  $\mathbf{o}_\Delta = \cos(\frac{\Delta \phi}{2}) + \mathbf{n} \sin(\frac{\Delta \phi}{2})$ .

### 3.3 Constructive artificial vector field

The vector field methodology used in this work was proposed in [Rezende et al., 2021] as a guidance control scheme that allows time-varying path-following in  $n$ -dimensions. It was then further extended to allow collision avoidance in [Nunes et al., 2022].

The general method is valid for all  $n$ -dimensional space. But since in this work we are only interested in rigid-body motions, more specifically quadcopter dynamics, we use the vector field in a three-dimensional space. For that reason, the vector field review follows using pure quaternion algebra applied to this 3D space.

#### 3.3.1 Modeling and problem definition

The reviewed vector field method is concerned with the first-order (translational) integrator model given by

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (3.39)$$

where  $\mathbf{p} \in \mathbb{H}_p$  (3.25) stands for the system position and  $\mathbf{v} \in \mathbb{H}_p$  (3.26) stands for the system velocity, also considered as the system input in the vector field guidance context.

Let  $\mathcal{C}(t) \subset \mathbb{H}_p$  denote the time-varying target path and consider the following assumptions.

**Assumption 1** (The curve is either open or closed). *The target time-varying curve  $\mathcal{C}(t)$  is either: (i) an open curve, i.e. an unbounded curve homeomorphic to a straight line; or (ii) a closed curve, i.e. a bounded curve homeomorphic to a circle.*

**Assumption 2** (Smoothness of curve's parametrization). *The target curve  $\mathcal{C}(t)$  has a parametric representation  $\mathbf{r}(s, t)$  with both the following properties: (i) it is twice differentiable in  $s$  for all  $t$ ; and (ii) it is differentiable in  $t$  for all  $s$ .*

The path-following problem is then formalized as

**Problem 1** (Path-following of simple integrator). *Find a control law  $\mathcal{F}(\mathbf{p}, t)$  such that if  $\mathbf{v} = \mathcal{F}$  the trajectories of the system (3.39) converges to and follows with constant speed  $v_r \in \mathbb{R}$  a given time-varying target path  $\mathcal{C}(t) \subset \mathbb{H}_p$ , under Assumptions 1 and 2.*

However, to ensure safety, the system must not collide with obstacles. Consider the following definitions about subsets of  $\mathbb{H}_p$ :

**Definition 13** (Obstacle set). *Let  $\mathcal{O}(t) \subset \mathbb{H}_p$  be the obstacle set.*

**Definition 14** (Workspace). *Let  $\mathcal{W}(t) \subset \mathbb{H}_p$  be the workspace, i.e. the set of feasible positions.*

Note that by definition  $\mathcal{W}(t) \cap \mathcal{O}(t) = \emptyset \forall t$ .

**Definition 15** (Safety set). *Let  $\mathcal{S}(t) \subset \mathcal{W}(t)$  be the set of positions in the workspace that are far from the obstacle set, as defined in*

$$\mathcal{S}(t) = \{\mathbf{p}_w(t) \in \mathcal{W}(t) \mid \|\mathbf{p}_w(t) - \mathbf{p}_o(t)\| \geq \lambda, \lambda \in \mathbb{R}_+, \forall \mathbf{p}_o(t) \in \mathcal{O}(t), \forall t\}. \quad (3.40)$$

Figure 8 shows the sets:  $\mathcal{O}$  as a dark gray circle,  $\mathcal{W}$  as a light gray region, and  $\mathcal{S} \subset \mathcal{W}$  as a blue region. The  $\lambda$  parameter that sets the boundary of  $\mathcal{S}$  is also shown in red dashed lines.

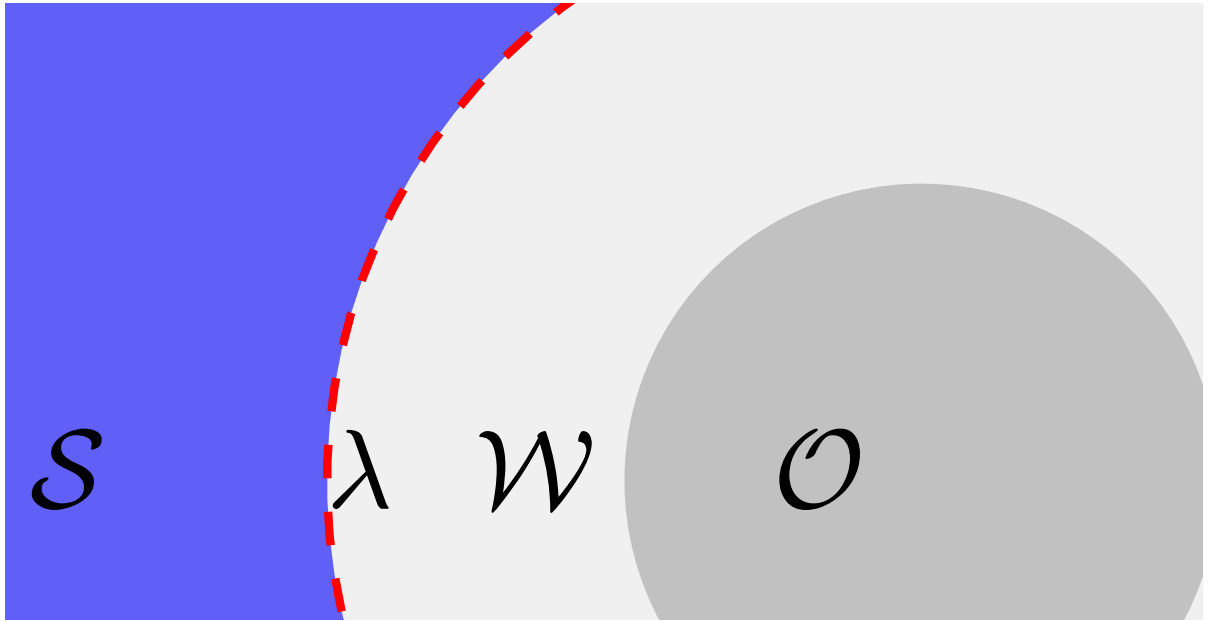


Figure 8 – Illustration of the sets:  $\mathcal{O}$  as a dark gray circle,  $\mathcal{W}$  as a light gray region, and  $\mathcal{S} \subset \mathcal{W}$  as a blue region. The  $\lambda$  parameter that sets the boundary of  $\mathcal{S}$  is also shown in red dashed lines.

Then, the collision avoidance problem is stated as

**Problem 2** (Collision avoidance). *The system must avoid obstacles by keeping at least a safe distance  $\lambda$  from the obstacle set, i.e.*

$$\mathbf{p}(t) \in \mathcal{S}(t), \quad \forall t \geq 0. \quad (3.41)$$

If obstacles are sensed between the system and the curve, the controller must prioritize the avoidance over the curve following, *i.e.* the system is allowed to move away from the curve to accomplish the obstacle deviation and then return to it once the avoidance is finished.

It means that the compromise idea is to always solve Problem 2 and, when it is safe, solve Problem 1. A formal definition of when each goal will be tackled or what means to be *safe* will be presented shortly.

### 3.3.2 Constructive method

To accomplish both path-following and obstacle avoidance following the compromise idea,  $\mathcal{F}(\mathbf{p}, t)$  will switch between: (i) a field that only follows the target curve, let us call it  $\Phi(\mathbf{p}, t)$ ; and (ii) another field that only circumnavigates the closest obstacle,  $\Psi(\mathbf{p}, t)$ , at a fixed predefined safety distance  $\lambda$ . Both fields  $\Phi(\mathbf{p}, t)$  and  $\Psi(\mathbf{p}, t)$  will be reviewed right after presenting the composite field idea.

#### 3.3.2.1 Composite Field

The idea is to make  $\mathcal{F}(\mathbf{p}, t) = \Phi(\mathbf{p}, t)$  when it is safe and make  $\mathcal{F}(\mathbf{p}, t) = \Psi(\mathbf{p}, t)$  when an imminent collision is detected. The latter scenario of imminent collision effectively means two conditions: (i) the current curve field direction points to a direction in which the obstacle distance decreases; and (ii) the distance between the system and the closest obstacle is smaller than a predefined threshold  $D_{in} > \lambda$ . Let  $\mathbf{D}_o(\mathbf{p}, t)$  be the distance vector (pure quaternion) from the closest obstacle to the system and  $\|\mathbf{D}_o(\mathbf{p}, t)\|$  be its norm.

To avoid abrupt field transitions, an intermediate state is also considered to accomplish a smooth switching. Let  $D_{in0} > D_{in}$  be another threshold value. Then, this intermediate state occurs when  $D_{in} < \|\mathbf{D}_o(\mathbf{p}, t)\| < D_{in0}$ . In this state,  $\mathcal{F}(\mathbf{p}, t)$  is a combination between both fields weighted by

$$\Theta(\mathbf{p}, t) = \frac{\|\mathbf{D}_o(\mathbf{p}, t)\| - D_{in}}{D_{in0} - D_{in}} \in [0, 1]. \quad (3.42)$$

Formal definitions about when it is *safe* to follow the target curve and when an imminent collision is detected are stated below.

**Definition 16** (Safe conditions to follow the curve). *It is considered to be safe to follow the curve when at least one of the conditions below are met:*

$$\mathbf{D}_o(\mathbf{p}, t) \cdot \Phi(\mathbf{p}, t) \geq 0, \text{ or} \quad (3.43)$$

$$\|\mathbf{D}_o(\mathbf{p}, t)\| > D_{in0}. \quad (3.44)$$

**Definition 17** (Imminent collision conditions). *It is considered to be in imminent collision when both the conditions below are met:*

$$\mathbf{D}_o(\mathbf{p}, t) \cdot \Phi(\mathbf{p}, t) < 0, \text{ and} \quad (3.45)$$

$$\|\mathbf{D}_o(\mathbf{p}, t)\| < D_{in}. \quad (3.46)$$

Then,  $\mathcal{F}(\mathbf{p}, t)$  is computed according to

$$\mathcal{F}(\mathbf{p}, t) = \begin{cases} \Phi(\mathbf{p}, t), & \text{if safe,} \\ \Psi(\mathbf{p}, t), & \text{if imminent collision,} \\ v_r \frac{\Theta(\mathbf{p}, t)\Phi(\mathbf{p}, t) + (1-\Theta(\mathbf{p}, t))\Psi(\mathbf{p}, t)}{\|\Theta(\mathbf{p}, t)\Phi(\mathbf{p}, t) + (1-\Theta(\mathbf{p}, t))\Psi(\mathbf{p}, t)\|}, & \text{otherwise.} \end{cases} \quad (3.47)$$

Figure 9 illustrates the field  $\mathcal{F}(\mathbf{p}, t)$  in the two main conditions (3.47). On the left, when it is safe to follow the curve,  $\mathcal{F}(\mathbf{p}, t) = \Phi(\mathbf{p}, t)$ . On the right, when imminent collision is detected,  $\mathcal{F}(\mathbf{p}, t) = \Psi(\mathbf{p}, t)$ .

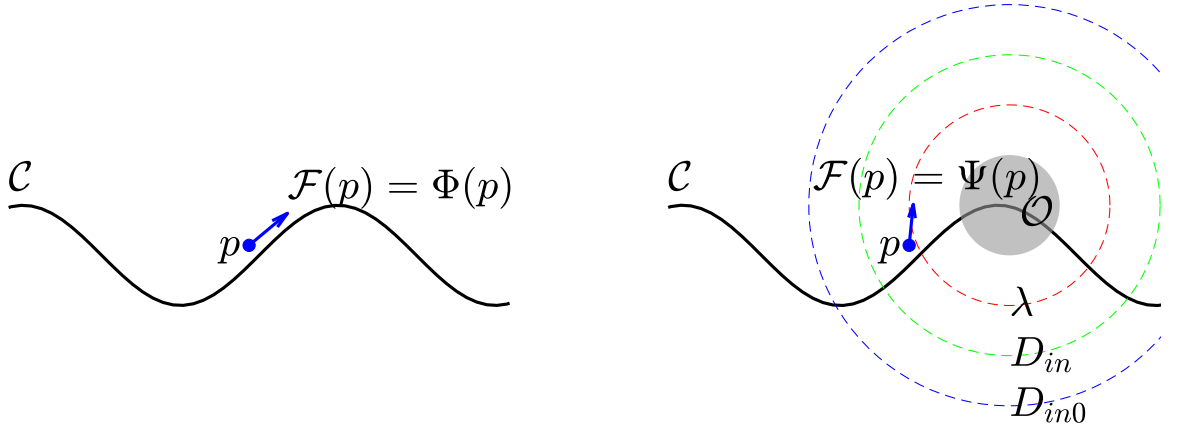


Figure 9 – Illustration of the field  $\mathcal{F}(\mathbf{p}, t)$  in the two main conditions. On the left, when it is safe to follow the curve,  $\mathcal{F}(\mathbf{p}, t) = \Phi(\mathbf{p}, t)$ . On the right, when imminent collision is detected,  $\mathcal{F}(\mathbf{p}, t) = \Psi(\mathbf{p}, t)$ .

### 3.3.2.2 Curve Field

The curve field  $\Phi(\mathbf{p}, t)$  follows the procedure from [Rezende et al., 2021]. First, consider the curve parametric representation  $\mathbf{r}(s, t)$  and then the definitions of the curve's closest point to the system

$$s_*(\mathbf{p}, t) \in \arg \min_s \|\mathbf{p} - \mathbf{r}(s, t)\|, \quad (3.48)$$

$$\mathbf{r}_*(\mathbf{p}, t) = \mathbf{r}(s_*(\mathbf{p}, t), t). \quad (3.49)$$

Note that, in general, more than one value can exist for  $s_*(\mathbf{p}, t)$  and therefore we can obtain more than one solution for the vector field. As discussed in [Rezende et al., 2021], one can pick any of the possible solutions and eventually the system will escape the regions where the field has more than one solution. Common implementation choices are: (i) the first obtained solution; (ii) the solution closest to the last one obtained.

Next, the distance between the curve and the system,  $\mathbf{D}(\mathbf{p}, t)$ , and the tangent component,  $\mathbf{T}(\mathbf{p}, t)$ , of the curve at  $s_*$  are defined as

$$\mathbf{D}(\mathbf{p}, t) = \mathbf{p} - \mathbf{r}_*(\mathbf{p}, t), \quad (3.50)$$

$$\mathbf{T}(\mathbf{p}, t) = \left. \frac{\partial \mathbf{r}(s, t)}{\partial s} \right|_{s=s_*}. \quad (3.51)$$

Those two components are weighted by scalar gains

$$G(\|\mathbf{D}(\mathbf{p}, t)\|) = \frac{2}{\pi} \tan^{-1}(k_G \|\mathbf{D}(\mathbf{p}, t)\|), \quad k_G > 0, \quad (3.52)$$

$$H(\|\mathbf{D}(\mathbf{p}, t)\|) = \pm \sqrt{1 - G(\|\mathbf{D}(\mathbf{p}, t)\|)^2}. \quad (3.53)$$

In which the choice of the sign of  $H(\|\mathbf{D}(\mathbf{p}, t)\|)$  defines the direction in which the curve will be followed.

Then, we compute a static unit field,  $\Phi_S$ , and a feed-forward component,  $\Phi_T$ , as in

$$\Phi_S(\mathbf{p}, t) = -G(\|\mathbf{D}(\mathbf{p}, t)\|) \frac{\mathbf{D}(\mathbf{p}, t)}{\|\mathbf{D}(\mathbf{p}, t)\|} + H(\|\mathbf{D}(\mathbf{p}, t)\|) \frac{\mathbf{T}(\mathbf{p}, t)}{\|\mathbf{T}(\mathbf{p}, t)\|}, \quad (3.54)$$

$$\Phi_T(\mathbf{p}, t) = -\Pi_T(\mathbf{p}, t) \frac{\partial \mathbf{D}}{\partial t}, \quad (3.55)$$

where  $\Pi_T$  is the null space projector of  $\frac{\mathbf{T}(\mathbf{p}, t)}{\|\mathbf{T}(\mathbf{p}, t)\|}$ .

The null space projector of any vector  $\vec{v} \neq 0$  is defined as a matrix that removes the component in the direction of  $\vec{v}$ ,

$$\Pi_v = I - \left( \frac{\vec{v}}{\|\vec{v}\|} \right) \cdot \left( \frac{\vec{v}}{\|\vec{v}\|} \right)^T \quad (3.56)$$

where  $I$  denotes the identity matrix of appropriate size and  $\cdot^T$  the vector transposition operator.

The field  $\Phi(\mathbf{p}, t)$  is given by

$$\Phi(\mathbf{p}, t) = \eta(\mathbf{p}, t)\Phi_S(\mathbf{p}, t) + \Phi_T(\mathbf{p}, t). \quad (3.57)$$

where  $\eta(\mathbf{p}, t) \in [0, v_r]$  is a scalar gain to ensure  $\|\Phi(\mathbf{p}, t)\| = v_r$  [Rezende et al., 2021]. This gain is given by

$$\eta(\mathbf{p}, t) = -\Phi_S(\mathbf{p}, t) \cdot \Phi_T(\mathbf{p}, t) + \sqrt{(\Phi_S(\mathbf{p}, t) \cdot \Phi_T(\mathbf{p}, t))^2 + v_r^2 - \|\Phi_T(\mathbf{p}, t)\|^2}. \quad (3.58)$$

In order to  $\exists \eta(\mathbf{p}, t) \in [0, v_r]$ , the time variation of the curve must be limited. To formalize that, we consider the following assumption:

**Assumption 3** (Limited curve time variation). *The time variation of the curve  $\mathcal{C}(t)$  is limited by*

$$\left\| \Pi_T(\mathbf{p}, t) \frac{\partial \mathbf{D}}{\partial t} \right\| \leq v_r. \quad (3.59)$$

Note that this assumption is a weaker form of limiting the curve's time variation  $\frac{\partial \mathbf{D}}{\partial t}$  since it does not consider the curve's variation inside itself, *e.g.* a spinning circle [Rezende et al., 2021].

If Assumption 3 does not hold for a given moment, there is not a real value for  $\eta$  (3.58) that can ensure both the field norm  $v_r$  and the convergence. One possible choice is to assign  $\eta = 0$  in this case, but the field norm will exceed  $v_r$  and the system may not be able to impose a velocity greater than that.

### 3.3.2.3 Obstacle Field

The field  $\Psi(\mathbf{p}, t)$  from [Nunes et al., 2022] follows a similar procedure. First, consider the obstacle set  $\mathcal{O}(t)$ . In this field, the obstacle's closest point will be used similar to the curve's closest point in the curve field. Let  $\mathbf{p}_{o*} \in \mathcal{O}(t)$  be one of the closest obstacle points, which is given by

$$\mathbf{p}_{o*}(\mathbf{p}, t) \in \arg \min_{\mathbf{p}_o \in \mathcal{O}(t)} \|\mathbf{p} - \mathbf{p}_o\|. \quad (3.60)$$

This field's goal is to make the system circumnavigate the obstacles at a fixed distance  $\lambda > 0$ . Then, the distance vectors are defined as

$$\mathbf{D}_o(\mathbf{p}, t) = \mathbf{p} - \mathbf{p}_{o*}(\mathbf{p}, t), \quad (3.61)$$

$$\mathbf{D}_\lambda(\mathbf{p}, t) = \mathbf{D}_o(\mathbf{p}, t) - \lambda \frac{\mathbf{D}_o(\mathbf{p}, t)}{\|\mathbf{D}_o(\mathbf{p}, t)\|}. \quad (3.62)$$



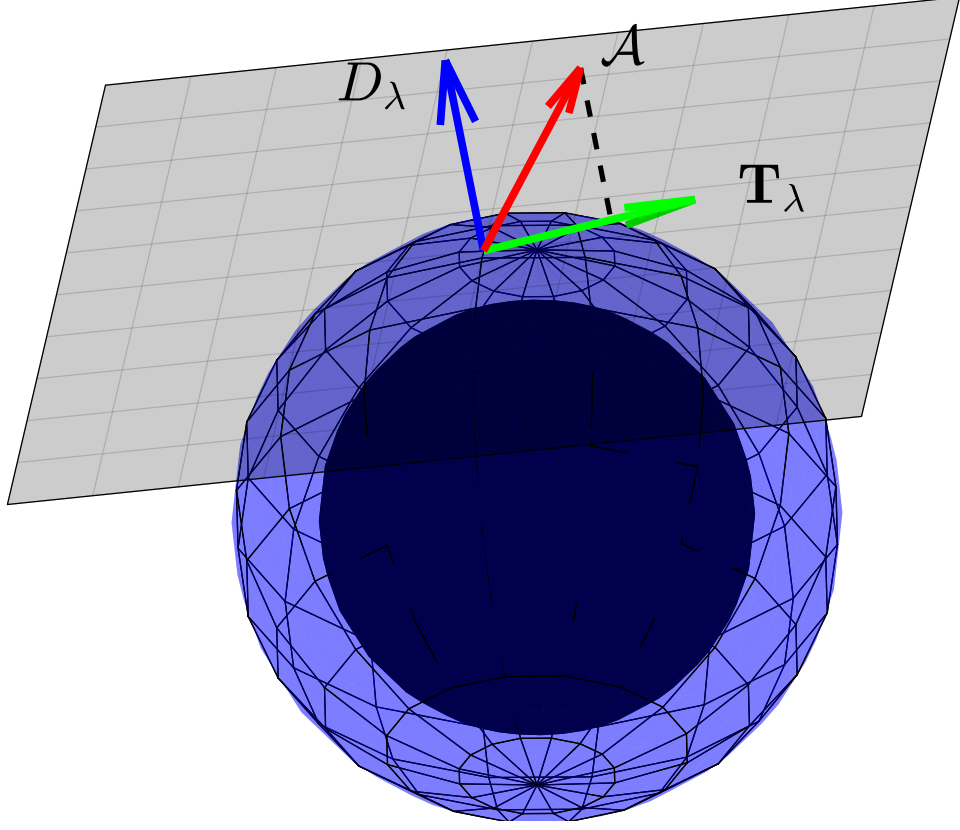


Figure 11 – Illustration from [Nunes et al., 2022] of the projection of the auxiliary field  $\mathcal{A}(\mathbf{p}, t)$  used to compute a tangent vector  $\mathbf{T}_\lambda(\mathbf{p}, t)$ .

Next, let  $\Psi_S(\mathbf{p}, t)$  be the unit static component and  $\Psi_T(\mathbf{p}, t)$  be the feed-forward component given by

$$\Psi_S(\mathbf{p}, t) = -G_o(\|\mathbf{D}_\lambda(\mathbf{p}, t)\|) \frac{\mathbf{D}_\lambda(\mathbf{p}, t)}{\|\mathbf{D}_\lambda(\mathbf{p}, t)\|} + H_o(\|\mathbf{D}_\lambda(\mathbf{p}, t)\|) \frac{\mathbf{T}_\lambda(\mathbf{p}, t)}{\|\mathbf{T}_\lambda(\mathbf{p}, t)\|}, \quad (3.64)$$

$$\Psi_T(\mathbf{p}, t) = -\Pi_{T_\lambda} \frac{\partial \mathbf{D}_\lambda}{\partial t}, \quad (3.65)$$

where

$$G_o(\|\mathbf{D}_\lambda(\mathbf{p}, t)\|) = \frac{2}{\pi} \tan^{-1}(k_{Go} \|\mathbf{D}_\lambda(\mathbf{p}, t)\|), \quad k_{Go} > 0, \quad (3.66)$$

$$H_o(\|\mathbf{D}_\lambda(\mathbf{p}, t)\|) = \sqrt{1 - G_o(\|\mathbf{D}_\lambda(\mathbf{p}, t)\|)^2}. \quad (3.67)$$

Finally, the field  $\Psi(\mathbf{p}, t)$  is

$$\Psi(\mathbf{p}, t) = \eta_o(\mathbf{p}, t) \Psi_S(\mathbf{p}, t) + \Psi_T(\mathbf{p}, t), \quad (3.68)$$

where

$$\eta_o(\mathbf{p}, t) = -\Psi_S(\mathbf{p}, t) \cdot \Psi_T(\mathbf{p}, t) + \sqrt{(\Psi_S(\mathbf{p}, t) \cdot \Psi_T(\mathbf{p}, t))^2 + v_r^2 - \|\Psi_T(\mathbf{p}, t)\|^2}. \quad (3.69)$$

### 3.3.3 Field properties

When it is safe to follow the curve, *i.e.* the curve vector field is active, the following properties hold:

- The trajectories of the system (3.39) converges to the curve  $\mathcal{C}(t)$  for any initial condition.
- There are no unstable or stable equilibrium points;
- It is robust to norm-bounded additive disturbances. It is also possible to find the finite region where the system deviates from the curve obtained from the maximum bound of the disturbance;
- The convergence to the curve is isotropic in space.

Formal proofs can be found in [Rezende et al., 2021].

When considering the scenario with obstacles in imminent collision, we cannot ensure the convergence to the curve, since the system may move away from the curve or take a detour in order to avoid collisions.

By putting the collision avoidance in the guidance method, we allow more complex non-linear controllers to run faster, such as in [Pereira et al., 2021].

# Chapter 4

## Position control

This chapter is concerned with solving the second-order position path-following problem with collision avoidance of a dynamic point mass system, under unknown bounded disturbances. The controller proposed here can be employed in the control of various mobile robots.

### 4.1 Modeling and problem statement

As explored in Chapter 3 and illustrated by Figure 6, let  $\mathbf{p}$  (3.25) denote a position and let  $\mathbf{v}$  (3.26) denote a linear velocity of a moving particle, both expressed in the inertial frame. Given the states  $\mathbf{p}$  and  $\mathbf{v}$ , and considering the control inputs as the forces  $\mathbf{f} \in \mathbb{H}_p$ , a dynamic model of a point mass is presented as

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (4.1)$$

$$\dot{\mathbf{v}} = -g\hat{\mathbf{k}} + m^{-1}\mathbf{f} + m^{-1}\mathbf{f}_d, \quad (4.2)$$

where  $g$  is the gravity acceleration,  $m \in \mathbb{R}, m > 0$  is the mass, and  $\mathbf{f}_d \in \mathbb{H}_p$  is the unknown magnitude-bounded disturbance. This system is said to be the **practical system**.

The problems addressed in this Chapter are Problem 2 of collision avoidance and Problem 1 but stated for the system (4.1)-(4.2) as formalized next.

**Problem 3** (Path-following of second-order system). *Considering the system (4.1)-(4.2), find a control law for  $\mathbf{f}$  such that the trajectories of the system converge to and follow with constant speed  $v_r \in \mathbb{R}$  a given time-varying target path  $\mathcal{C}(t) \subset \mathbb{H}_p$ , under Assumptions 1 and 2.*

Recall the Definition 16 about when it is safe to follow the path. When the conditions to be safe are met, Problem 3 will be solved by our controller. Otherwise, the system may move away from the path in order to avoid collisions and therefore prioritize Problem 2 in a compromise idea as employed by the vector field (3.47).

The solution consists of three parts.

- First, we find a nominal (*i.e.* when  $\mathbf{f}_d = 0$ ) feedback controller that ensures the path-following and the obstacle avoidance;
- Next, we find a robust adaptive control for the error between the practical system and the previously found nominal trajectory, such that it is attracted to small set around the origin, *i.e.* around zero error;
- Finally, we find the practical system control law by combining the nominal and the robust adaptive controllers.

Figure 12 shows the controller architecture diagram that highlights the combination of a nominal controller and a robust adaptive control. In this diagram, we have the nominal feedback loop with the nominal controller and the nominal system. Then, with the robust adaptive controller, we minimize the error between this nominal system and the practical system. Combining both control actions, we achieve our proposed controller.

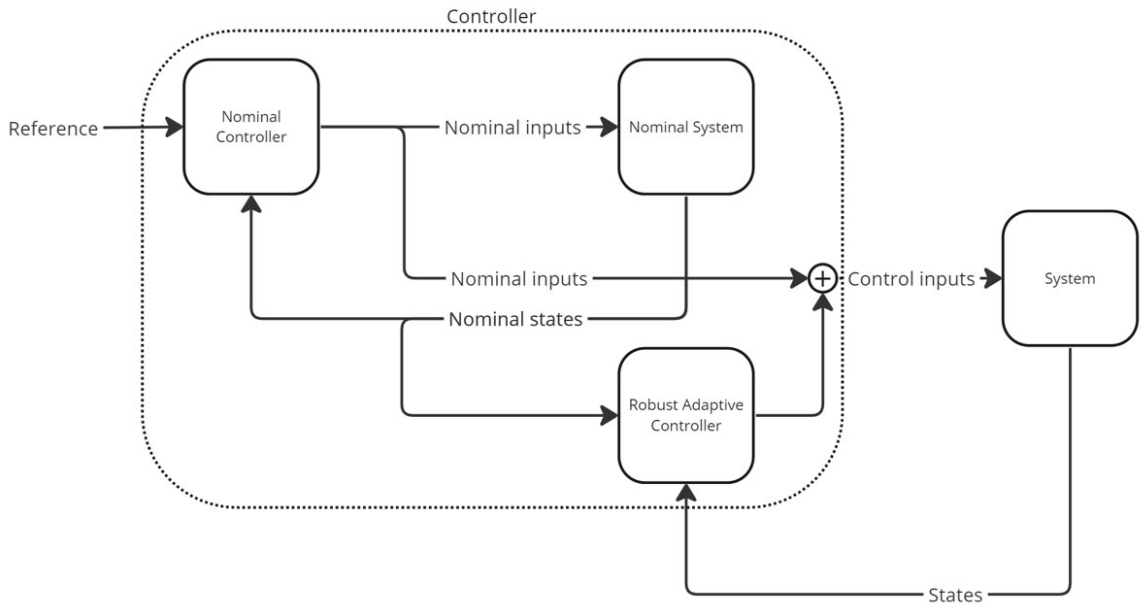


Figure 12 – Controller architecture diagram with the combination of a nominal controller and a robust adaptive control. It shows: the nominal feedback loop with the nominal controller and the nominal system; the robust adaptive controller; and combining both control actions, the complete proposed controller that acts in the system.

## 4.2 Integrated vector field and Backstepping nominal control

The first part of our method consists of a two-step Backstepping control to ensure nominal asymptotic convergence for the reference trajectory [Salierno, 2018, Salierno and Raffo, 2017].

Our Backstepping controller is integrated with the vector field guidance [Nunes et al., 2022], allowing a collision-free path-following and giving rise to an Integrated Guidance and Control (IGC) methodology.

Regarding the nominal performance, it is said to be the **nominal system** the system with null disturbances, represented with a top bar,  $\bar{\cdot}$ , as in

$$\dot{\bar{\mathbf{p}}} = \bar{\mathbf{v}}, \quad (4.3)$$

$$\dot{\bar{\mathbf{v}}} = -g\hat{\mathbf{k}} + m^{-1}\bar{\mathbf{f}}. \quad (4.4)$$

### 4.2.1 Step 1

In the first step, we consider the first-order integrator model (4.3) and design a virtual control law for the velocity. The field  $\mathcal{F}(\bar{\mathbf{p}}, t)$  (3.47) is this control law.

Let  $\mathbf{z}_p \equiv \mathbf{D}(\bar{\mathbf{p}}, t) = \bar{\mathbf{p}} - \mathbf{r}_*$  be the position error between the nominal system and the target curve.

We can ensure that  $\mathbf{z}_p = 0$  will be a uniform global asymptotic stable (UGAS) equilibrium point only when it is safe to follow the curve according to Definition 16. Otherwise, the system can be forced to increase  $\mathbf{z}_p$  to avoid collisions. In this case, for the sake of simplicity, consider for now that the conditions to follow the curve are always met. Lately, we show that when those conditions are not met, the system is controlled to avoid the obstacles regarding Problem 2.

Let  $V_p : \mathbb{H}_p \rightarrow \mathbb{R}$  be a candidate Lyapunov function, given by

$$V_p(\mathbf{z}_p) = \frac{1}{2} \mathbf{z}_p \cdot \mathbf{z}_p, \quad (4.5)$$

from which its time derivative yields

$$\begin{aligned} \dot{V}_p(\mathbf{z}_p) &= \mathbf{z}_p \cdot \dot{\mathbf{z}}_p, \\ \dot{V}_p(\mathbf{z}_p) &= \mathbf{z}_p \cdot (\mathcal{F} - \dot{\mathbf{r}}_*), \end{aligned} \quad (4.6)$$

in this case  $\dot{V}_p(\mathbf{z}_p) = \mathbf{z}_p \cdot (\Phi - \dot{\mathbf{r}}_*)$  since we are considering for now that it is safe to follow the curve.

According to Proposition 2 of [Rezende et al., 2021], we have that

$$\begin{aligned}
\dot{V}_p &= \frac{\partial V_p}{\partial \mathbf{p}} \cdot \dot{\mathbf{p}} + \frac{\partial V_p}{\partial t}, \\
\dot{V}_p &= \mathbf{D} \cdot \Phi + \frac{\partial V_p}{\partial t}, \\
\dot{V}_p &= \mathbf{D} \cdot \left( -\eta G \frac{\mathbf{D}}{\|\mathbf{D}\|} + \eta H \frac{\mathbf{T}}{\|\mathbf{T}\|} - \Pi_T \frac{\partial \mathbf{D}}{\partial t} \right) + \frac{\partial V_p}{\partial t}, \\
\dot{V}_p &= -\eta G \frac{\|\mathbf{D}\|^2}{\|\mathbf{D}\|} + \eta H \frac{\mathbf{D} \cdot \mathbf{T}}{\|\mathbf{T}\|} - \mathbf{D} \cdot \Pi_T \frac{\partial \mathbf{D}}{\partial t} + \frac{\partial V_p}{\partial t}.
\end{aligned} \tag{4.7}$$

From Lemma 2 of [Rezende et al., 2021]  $\mathbf{D} \cdot \mathbf{T} = 0$  since  $\mathbf{D} \perp \mathbf{T}$ , and from Lemma 4 of [Rezende et al., 2021]  $\frac{\partial V_p}{\partial t} = \mathbf{D} \cdot \Pi_T \frac{\partial \mathbf{D}}{\partial t}$ . This leads to

$$\dot{V}_p = -\eta G \frac{\|\mathbf{D}\|^2}{\|\mathbf{D}\|}, \tag{4.8}$$

$$\dot{V}_p(0) = 0; \quad \dot{V}_p(\mathbf{z}_p) = -\eta G \|\mathbf{z}_p\| < 0, \quad \forall \mathbf{z}_p \neq 0. \tag{4.9}$$

### 4.2.2 Step 2

In the second step, let  $\mathbf{z}_v = \bar{\mathbf{v}} - \mathcal{F}(\bar{\mathbf{p}}, t)$  be the velocity error between the nominal velocity and the vector field. Then, consider the system

$$\dot{\mathbf{z}}_p = \mathcal{F} - \dot{\mathbf{r}}_* + \mathbf{z}_v, \tag{4.10}$$

$$\dot{\mathbf{z}}_v = -g\hat{\mathbf{k}} + m^{-1}\bar{\mathbf{f}} - \mathcal{F}. \tag{4.11}$$

To find the control law for  $\bar{\mathbf{f}}$ , let  $V_{p,v} : \mathbb{H}_p^2 \rightarrow \mathbb{R}$  be a control Lyapunov function given by

$$V_{p,v}(\mathbf{z}_p, \mathbf{z}_v) = K_p V_p + \frac{1}{2} \mathbf{z}_v \cdot \mathbf{z}_v, \tag{4.12}$$

with  $K_p > 0$  and  $V_p$  from (4.5). Its time derivative yields

$$\begin{aligned}
\dot{V}_{p,v} &= K_p \dot{V}_p + \mathbf{z}_v \cdot \dot{\mathbf{z}}_v, \\
\dot{V}_{p,v} &= K_p \mathbf{z}_p \cdot \dot{\mathbf{z}}_p + \mathbf{z}_v \cdot \dot{\mathbf{z}}_v, \\
\dot{V}_{p,v} &= K_p \mathbf{z}_p \cdot (\mathcal{F} + \mathbf{z}_v - \dot{\mathbf{r}}_*) + \mathbf{z}_v \cdot (-g\hat{\mathbf{k}} + m^{-1}\bar{\mathbf{f}} - \mathcal{F}).
\end{aligned} \tag{4.13}$$

From (4.9), we know that the term  $\mathbf{z}_p \cdot (\mathcal{F} - \dot{\mathbf{r}}_*) = -\eta G \|\mathbf{z}_p\|$  is already negative definite. Then, the following feedback linearization control law is proposed as

$$\bar{\mathbf{f}} = m\mathcal{F} + mg\hat{\mathbf{k}} + m\bar{\mathbf{f}}_L, \tag{4.14}$$

with

$$\bar{\mathbf{f}}_L = -K_p \mathbf{z}_p - K_v \mathbf{z}_v, \tag{4.15}$$

where  $K_v > 0$ .

**Lemma 1.** *Applying (4.14)-(4.15), the system (4.10)-(4.11) has an asymptotic stable equilibrium point at the origin.*

*Proof.* Consider the following closed-loop system given by (4.10)-(4.11) with (3.47) and (4.14)-(4.15):

$$\dot{\mathbf{z}}_p = \mathcal{F} - \dot{\mathbf{r}}_* + \mathbf{z}_v, \quad (4.16)$$

$$\dot{\mathbf{z}}_v = -K_p \mathbf{z}_p - K_v \mathbf{z}_v. \quad (4.17)$$

Replacing (4.14)-(4.15) in (4.13) yields

$$\begin{aligned} \dot{V}_{p,v}(\mathbf{z}_p, \mathbf{z}_v) &= K_p \mathbf{z}_p \cdot (\mathcal{F} + \mathbf{z}_v - \dot{\mathbf{r}}_*) + \mathbf{z}_v \cdot \bar{\mathbf{f}}_L, \\ \dot{V}_{p,v}(0, 0) &= 0; \quad \dot{V}_{p,v}(\mathbf{z}_p, \mathbf{z}_v) = -\eta G K_p \|\mathbf{z}_p\| - K_v \|\mathbf{z}_v\|^2 < 0, \forall (\mathbf{z}_p, \mathbf{z}_v) \neq (0, 0). \end{aligned} \quad (4.18)$$

□

### 4.2.3 Collision avoidance

The controller (4.14)-(4.15) was stated only when conditions from Definition 16 are met. When they are not, we cannot ensure asymptotic convergence to the curve. However, we show that in those cases, if  $\bar{\mathbf{v}} = \mathcal{F}$ , the system avoids obstacles, therefore solving Problem 2. We then ensure the collision avoidance with the following theorem:

**Theorem 1** (Invariance and attractiveness of safety set). *The safety set  $\mathcal{S}(t)$  from Definition 15 is invariant and attractive if  $\bar{\mathbf{v}} = \mathcal{F}$ .*

Figure 13 shows in red dashed lines three trajectories as examples of invariance and attractiveness of  $\mathcal{S}$ . It also illustrates  $\mathcal{S}$  in blue,  $\mathcal{C}$  in black line, and  $\mathcal{O}$  as a dark gray circle.

*Proof.* The boundary of  $\mathcal{S}(t)$ , defined as  $\partial\mathcal{S}(t)$ , is given by all  $\bar{\mathbf{p}} \in \mathcal{W}(t)$  such that  $D_\lambda = 0 \iff D_o = \lambda$ . From (3.47), we see that the field in  $\partial\mathcal{S}(t)$  is either the curve field or the obstacle field, *i.e.*

$$\mathcal{F}(\bar{\mathbf{p}}, t) \Big|_{\bar{\mathbf{p}} \in \partial\mathcal{S}(t), D_o \cdot \Phi(\bar{\mathbf{p}}, t) \geq 0} = \Phi(\bar{\mathbf{p}}, t), \quad \mathcal{F}(\bar{\mathbf{p}}, t) \Big|_{\bar{\mathbf{p}} \in \partial\mathcal{S}(t), D_o \cdot \Phi(\bar{\mathbf{p}}, t) < 0} = \Psi(\bar{\mathbf{p}}, t). \quad (4.19)$$

When it is the curve field it points to inside  $\mathcal{S}(t)$  since  $D_o \cdot \Phi(\bar{\mathbf{p}}, t) \geq 0$ . When it is the obstacle field, since  $D_\lambda = 0$ , it is perpendicular to  $D_o$ . Therefore  $\mathcal{F}(\bar{\mathbf{p}}, t) \Big|_{\bar{\mathbf{p}} \in \partial\mathcal{S}(t)}$  never points outside of  $\mathcal{S}(t)$ . This is exemplified by Figure 14, where  $\mathcal{S}$  is in blue,  $\mathcal{C}$  is a black line, and  $\mathcal{O}$  is a dark gray circle. The field  $\mathcal{F}$  on  $\partial\mathcal{S}(t)$  is plotted as red arrows that never point outside of  $\mathcal{S}$ .

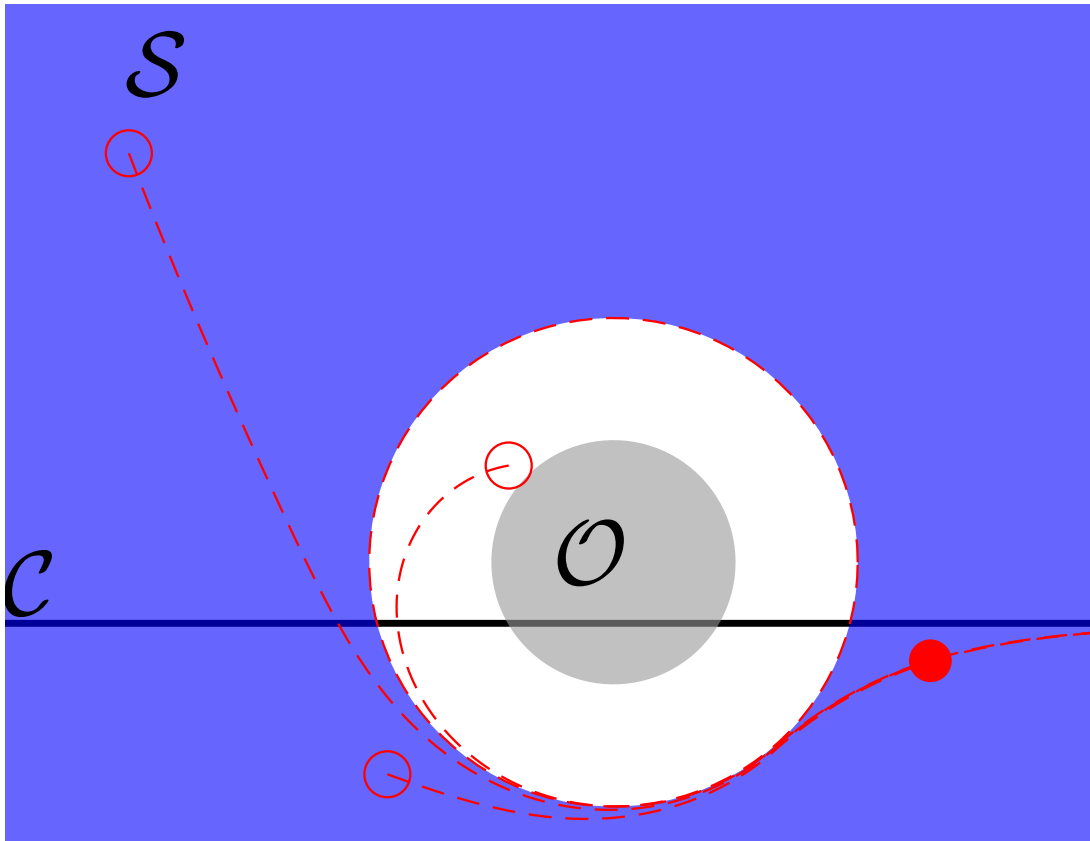


Figure 13 – Illustration of  $\mathcal{S}$  in blue,  $\mathcal{C}$  in black line, and  $\mathcal{O}$  as a dark gray circle. Three example trajectories that show invariance and attractiveness of  $\mathcal{S}$  are plotted in red dashed lines.

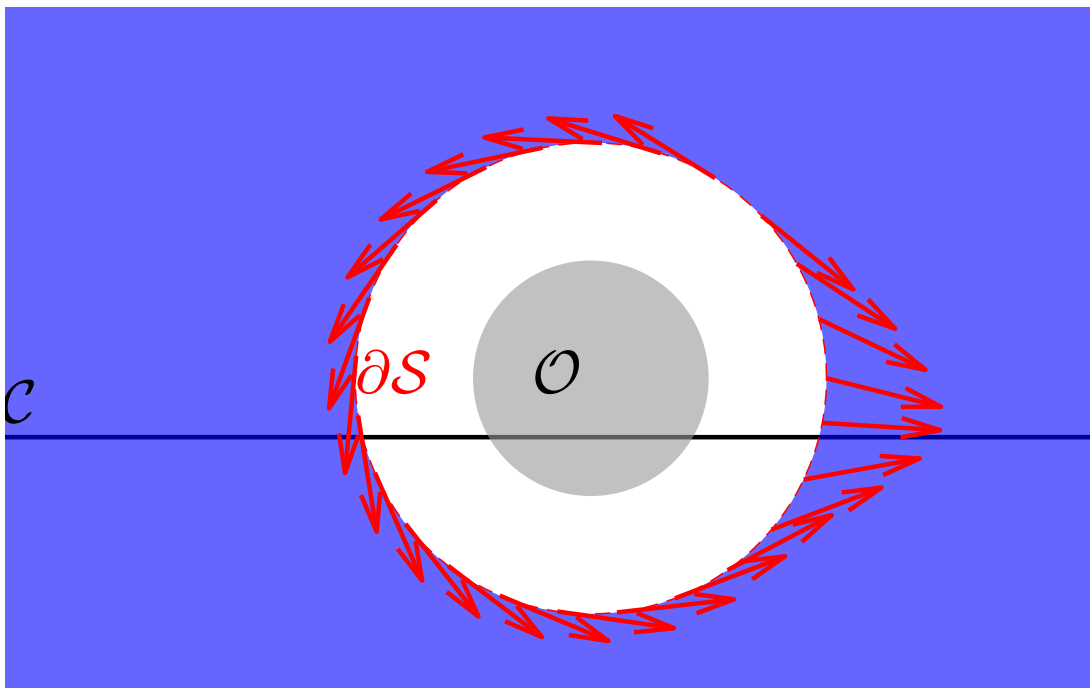


Figure 14 – Illustration of  $\mathcal{S}$  in blue,  $\mathcal{C}$  as a black line, and  $\mathcal{O}$  as a dark gray circle. The field  $\mathcal{F}$  on  $\partial\mathcal{S}(t)$  is plotted as red arrows that never point outside of  $\mathcal{S}$ .

Assuming  $\bar{\mathbf{v}} = \mathcal{F}$  and invoking Theorem 3.1 of [Blanchini, 1999], we conclude that  $\mathcal{S}(t)$  is invariant, *i.e.* if the system initial conditions are  $\bar{\mathbf{p}}(0) \in \mathcal{S}(0)$ , the trajectories of  $\bar{\mathbf{p}}(t)$  never go outside of  $\mathcal{S}(t)$ .

For the cases when  $\bar{\mathbf{p}}(0) \notin \mathcal{S}(0)$ , from (4.9) we see that  $D_\lambda$  decreases. Therefore, the nominal system trajectories converge to be inside  $\mathcal{S}(t)$  and never go out.  $\square$

### 4.3 Robust adaptive control

For the second part of our methodology, we propose a robust adaptive Backstepping control law to stabilize the error around zero between the practical system (4.1)-(4.2) and the nominal system (4.3)-(4.4).

The **error system** is a representation of the difference between the practical and nominal systems.

Let  $\mathbf{e}_p = \mathbf{p} - \bar{\mathbf{p}}$  be the error between the practical and nominal positions. Let  $\mathbf{v}_e = \mathbf{v} - \bar{\mathbf{v}}$  be an auxiliary variable that represents the error between the practical and nominal velocities such that

$$\dot{\mathbf{e}}_p = \mathbf{v}_e. \quad (4.20)$$

#### 4.3.1 Step 1

We want  $\mathbf{v}_e$  to follow a virtual control law  $\mathcal{F}_e$  such that  $\mathbf{e}_p \rightarrow 0$  if  $\mathbf{v}_e = \mathcal{F}_e$ . This virtual control law is chosen as

$$\mathcal{F}_e = -k_{\mathcal{F}_e} \mathbf{e}_p, \quad (4.21)$$

with  $k_{\mathcal{F}_e} > 0$ .

**Lemma 2.** *Applying the virtual control law (4.21) in the system (4.20), the point  $\mathbf{e}_p = 0$  becomes an asymptotic stable equilibrium point.*

*Proof.* Let  $V_{ep}$  be a candidate Lyapunov function

$$V_{ep}(\mathbf{e}_p) = \frac{1}{2} \mathbf{e}_p \cdot \mathbf{e}_p. \quad (4.22)$$

Applying the virtual control law  $\mathcal{F}_e$  (4.21), the time derivative of (4.22) yields

$$\begin{aligned} \dot{V}_{ep}(\mathbf{e}_p) &= \mathbf{e}_p \cdot \dot{\mathbf{e}}_p \\ \dot{V}_{ep}(0) &= 0; \dot{V}_{ep}(\mathbf{e}_p) = -k_{\mathcal{F}_e} \|\mathbf{e}_p\|^2 < 0, \quad \forall \mathbf{e}_p \neq 0. \end{aligned} \quad (4.23)$$

Therefore ensuring asymptotic convergence of the error  $\mathbf{e}_p$  to the origin.  $\square$

### 4.3.2 Step 2

Let  $\mathbf{e}_v = \mathbf{v}_e - \mathcal{F}_e$  be another velocity error that encompasses the virtual control law  $\mathcal{F}_e$ . The error system is represented by

$$\dot{\mathbf{e}}_p = \mathcal{F}_e + \mathbf{e}_v, \quad (4.24)$$

$$\dot{\mathbf{e}}_v = m^{-1} \mathbf{f}_e + m^{-1} \dot{\mathbf{f}}_d - \dot{\mathcal{F}}_e, \quad (4.25)$$

where  $\mathbf{f}_e \equiv \mathbf{f} - \bar{\mathbf{f}}$  is its control input.

In order to state the robust adaptive control law, consider the following assumption.

**Assumption 4** (Norm bounded disturbances). *Let*

$$\chi_p = \mathbf{f}_d - m\dot{\mathcal{F}}_e \quad (4.26)$$

be a lumped term that contains the disturbance [Zhang et al., 2021]. Using an indirect method [Zhu et al., 2011, Cai et al., 2008], the controller tackles  $\chi_p$  by focusing only on its bound. Therefore, we assume that exists unknown constants  $\zeta_1$  and  $\zeta_2$  such that  $\|\chi_p\| \leq \zeta_1 + \zeta_2 \|\mathbf{e}_v\|$ .

Those constants that bound  $\chi_p$  are estimated in the controller by adaptive gains  $\hat{\zeta}_1$  and  $\hat{\zeta}_2$  [Zhang et al., 2021].

Then, the robust adaptive control law for the error system is given by<sup>1</sup>

$$\mathbf{f}_e = -k_{ep}\mathbf{e}_p - k_{ev}\mathbf{e}_v - \hat{\zeta}_1 \mathbf{tanh}(\hat{\zeta}_1 \mathbf{e}_v / \epsilon_p) - \hat{\zeta}_2 \mathbf{e}_v. \quad (4.27)$$

where  $k_{ep}, k_{ev}, \epsilon_p > 0$  are scalar gains, and the adaptive parameters  $\hat{\zeta}_1, \hat{\zeta}_2 > 0$  have the following update laws

$$\dot{\hat{\zeta}}_1 = \rho_1 \|\mathbf{e}_v\|, \quad \dot{\hat{\zeta}}_2 = \rho_2 \|\mathbf{e}_v\|^2, \quad (4.28)$$

with  $\rho_1, \rho_2 > 0$ .

**Lemma 3.** *Considering Assumption 4, let  $V_{epv}$  be the following control Lyapunov function*

$$V_{epv} = k_{ep}V_{ep} + \frac{1}{2}m\mathbf{e}_v \cdot \mathbf{e}_v + \frac{\tilde{\zeta}_1^2}{2\rho_1} + \frac{\tilde{\zeta}_2^2}{2\rho_2}, \quad (4.29)$$

where

$$\begin{aligned} \tilde{\zeta}_1 &= \zeta_1 - \hat{\zeta}_1, \\ \tilde{\zeta}_2 &= \zeta_2 - \hat{\zeta}_2 \end{aligned} \quad (4.30)$$

<sup>1</sup> The function  $\mathbf{tanh}(\mathbf{q})$  is defined in terms of exponential maps. In the other hand, the bold  $\mathbf{tanh}(\mathbf{q})$  is an abuse of notation, where we actually mean the element-wise hyperbolic-tangent, where it is applied to each imaginary field of  $\mathbf{q}$ , i.e.  $\mathbf{tanh}(\mathbf{q}) \equiv \text{vec}_3^{-1}(\tanh(\vec{q}))$ ,  $\forall \mathbf{q} \in \mathbb{H}_p$ .

are estimation errors. By applying (4.27) in the system (4.24)-(4.25),  $\dot{V}_{epv} < 0$  if the system is outside the attractive set  $\Delta_p$  given by

$$\Delta_p = \{(\mathbf{e}_p, \mathbf{e}_v) \mid \|\mathbf{e}_p\| \leq \delta_p, \|\mathbf{e}_v\| \leq \delta_v\}, \quad (4.31)$$

$$\delta_p = \sqrt{\frac{3\kappa\epsilon_p}{k_{ep}k_{\mathcal{F}_e}}}, \quad \delta_v = \sqrt{\frac{3\kappa\epsilon_p}{k_{ev}}}, \quad (4.32)$$

with  $\kappa \approx 0.2785$ . Since we are looking for an upper bound, consider  $\kappa = 0.2786$ .

*Proof.* Consider Lemma 2 and the function (4.29). The time derivative of  $V_{epv}$  is given by

$$\begin{aligned} \dot{V}_{epv} &= k_{ep}\dot{V}_{ep} + m\mathbf{e}_v \cdot \dot{\mathbf{e}}_v + \tilde{\zeta}_1\dot{\tilde{\zeta}}_1/\rho_1 + \tilde{\zeta}_2\dot{\tilde{\zeta}}_2/\rho_2, \\ \dot{V}_{epv} &= -k_{ep}\mathbf{e}_p \cdot \dot{\mathbf{e}}_p + m\mathbf{e}_v \cdot \dot{\mathbf{e}}_v + \tilde{\zeta}_1(\dot{\tilde{\zeta}}_1 - \dot{\hat{\zeta}}_1)/\rho_1 + \tilde{\zeta}_2(\dot{\tilde{\zeta}}_2 - \dot{\hat{\zeta}}_2)/\rho_2, \\ \dot{V}_{epv} &= -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 + k_{ep}\mathbf{e}_p \cdot \mathbf{e}_v + \mathbf{e}_v \cdot (\mathbf{f}_d - m\dot{\mathcal{F}}_e + \mathbf{f}_e) - \tilde{\zeta}_1\dot{\tilde{\zeta}}_1/\rho_1 - \tilde{\zeta}_2\dot{\tilde{\zeta}}_2/\rho_2. \end{aligned} \quad (4.33)$$

Considering Assumption 4 and substituting (4.27) and (4.28) in (4.33) yields

$$\begin{aligned} \dot{V}_{epv} &= -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 + k_{ep}\mathbf{e}_p \cdot \mathbf{e}_v + \mathbf{e}_v \cdot (\chi_p + \mathbf{f}_e) - \tilde{\zeta}_1\dot{\tilde{\zeta}}_1/\rho_1 - \tilde{\zeta}_2\dot{\tilde{\zeta}}_2/\rho_2, \\ \dot{V}_{epv} &= -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 + k_{ep}\mathbf{e}_p \cdot \mathbf{e}_v + \mathbf{e}_v \cdot (\chi_p - k_{ep}\mathbf{e}_p - k_{ev}\mathbf{e}_v - \hat{\zeta}_1 \tanh(\hat{\zeta}_1\mathbf{e}_v/\epsilon_p) - \hat{\zeta}_2\mathbf{e}_v) \\ &\quad - \tilde{\zeta}_1\|\mathbf{e}_v\| - \tilde{\zeta}_2\|\mathbf{e}_v\|^2, \\ \dot{V}_{epv} &= -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 - k_{ev}\|\mathbf{e}_v\|^2 + \mathbf{e}_v \cdot (\chi_p - \hat{\zeta}_1 \tanh(\hat{\zeta}_1\mathbf{e}_v/\epsilon_p) - \hat{\zeta}_2\mathbf{e}_v) \\ &\quad - \tilde{\zeta}_1\|\mathbf{e}_v\| - \tilde{\zeta}_2\|\mathbf{e}_v\|^2, \\ \dot{V}_{epv} &= -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 - k_{ev}\|\mathbf{e}_v\|^2 + \mathbf{e}_v \cdot \chi_p - \mathbf{e}_v \cdot \hat{\zeta}_1 \tanh(\hat{\zeta}_1\mathbf{e}_v/\epsilon_p) - \hat{\zeta}_2\|\mathbf{e}_v\|^2 \\ &\quad - \tilde{\zeta}_1\|\mathbf{e}_v\| - \tilde{\zeta}_2\|\mathbf{e}_v\|^2. \end{aligned}$$

Recalling the bound of  $\chi_p$ ,  $\|\chi_p\| \leq \zeta_1 + \zeta_2\|\mathbf{e}_v\|$ , leads to

$$\begin{aligned} \dot{V}_{epv} &\leq -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 - k_{ev}\|\mathbf{e}_v\|^2 + \zeta_1\|\mathbf{e}_v\| + \zeta_2\|\mathbf{e}_v\|^2 - \hat{\zeta}_1\|\mathbf{e}_v\| \tanh(\hat{\zeta}_1\|\mathbf{e}_v\|/\epsilon_p) \\ &\quad - \hat{\zeta}_2\|\mathbf{e}_v\|^2 - (\zeta_1 - \hat{\zeta}_1)\|\mathbf{e}_v\| - (\zeta_2 - \hat{\zeta}_2)\|\mathbf{e}_v\|^2, \\ \dot{V}_{epv} &\leq -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 - k_{ev}\|\mathbf{e}_v\|^2 - \hat{\zeta}_1\|\mathbf{e}_v\| \tanh(\hat{\zeta}_1\|\mathbf{e}_v\|/\epsilon_p) + \hat{\zeta}_1\|\mathbf{e}_v\|. \end{aligned}$$

Using both the properties, [Zhang et al., 2021, Polycarpou and Ioannou, 1996],

$$\|\mathbf{q}\| \leq |q_x| + |q_y| + |q_z|, \forall \mathbf{q} \in \mathbb{H}_p, \quad (4.34)$$

$$0 \leq |a| - a \tanh(a/\epsilon_p) \leq \kappa\epsilon_p, \forall a \in \mathbb{R}, \quad (4.35)$$

yields  $0 \leq \|\mathbf{q}\| - \|\mathbf{q}\| \tanh(\|\mathbf{q}\|/\epsilon_p) \leq 3\kappa\epsilon_p, \forall \mathbf{q} \in \mathbb{H}_p$ , which leads to:

$$\dot{V}_{epv} \leq -k_{ep}k_{\mathcal{F}_e}\|\mathbf{e}_p\|^2 - k_{ev}\|\mathbf{e}_v\|^2 + 3\kappa\epsilon_p. \quad (4.36)$$

in which  $\dot{V}_{epv} < 0$  for  $\|\mathbf{e}_p\| > \sqrt{\frac{3\kappa\epsilon_p}{k_{ep}k_{\mathcal{F}_e}}}$  or  $\|\mathbf{e}_v\| > \sqrt{\frac{3\kappa\epsilon_p}{k_{ev}}}$ .  $\square$

The former inequality means that  $\delta_p = \sqrt{\frac{3\kappa\epsilon_p}{k_{ep}k_{\mathcal{F}_e}}}$  bounds an attractive set for the position error  $\mathbf{e}_p$ .

From the latter inequality and using the triangle inequality,  $\|\mathbf{e}_v\| \geq \|\mathbf{v}_e\| - \|\mathcal{F}_e\|$ , with  $\|\mathcal{F}_e\| = k_{\mathcal{F}_e}\|\mathbf{e}_p\| \leq k_{\mathcal{F}_e}\delta_p$  and finally  $\delta_{ve} = \delta_v + k_{\mathcal{F}_e}\delta_p$  bounds an attractive set for the velocity error  $\mathbf{v}_e$ .

## 4.4 Practical control system

The last part of our method is to conclude and combine the Position Controller design. Now we are able to present the complete controller scheme, which is given by

$$\mathbf{f} = \bar{\mathbf{f}} + \mathbf{f}_e, \quad (4.37)$$

with  $\bar{\mathbf{f}}$  from (4.14)-(4.15), and with  $\mathbf{f}_e$  from (4.27). The proposed method yields the following result:

**Theorem 2** (Position Controller). *Applying (4.37) in the practical system (4.1)-(4.2), the system trajectories: (i) never collide with obstacles if  $\lambda > \delta_p$ , as well as if  $\bar{\mathbf{v}} = \mathcal{F}$  and; (ii) when it is safe, asymptotically converge to a control attractive set around the target curve  $\mathcal{C}(t)$ , in which the set is (4.31).*

Note that one is able to choose the value of  $\lambda$  independently from  $\delta_p$ , such that it is always possible to make it greater than  $\delta_p$  and therefore ensure safety.

Considering  $\bar{\mathbf{v}} = \mathcal{F}$  is a theoretical necessity not addressed by this work, even though we know from Lemma 1 that  $z_v \rightarrow 0 \iff \bar{\mathbf{v}} \rightarrow \mathcal{F}$ . To address that, it is required to elaborate further on Theorem 1, which we propose as a future work.

*Proof.* Lemma 1 states that the nominal system (4.3)-(4.4) asymptotically converges to the path when it is safe and Theorem 1 states the nominal system's distance to the obstacle set is kept greater than or equal to  $\lambda$  if  $\bar{\mathbf{v}} = \mathcal{F}$ .

Furthermore, the error between the practical system and the nominal trajectory is constrained in a small set as a result of Lemma 3. This means that the system converges to an attractive set around the path when it is safe and that it keeps a distance around  $\lambda$  from the obstacles with a maximum deviation of  $\delta_p$ , *i.e.* it does not collide if  $\lambda - \delta_p > 0$ .  $\square$

# Chapter 5

## Attitude control

As opposed to the previous path-following controller, this Chapter is concerned with solving a trajectory-tracking problem. More specifically, we solve now the attitude trajectory-tracking problem of a second-order rigid-body attitude system, under unknown bounded disturbances. The Attitude Controller proposed here can be employed in various attitude systems such as aircrafts and spacecrafts.

### 5.1 Modeling and problem statement

As explored in Chapter 3 and illustrated in Figure 7, let  $\mathbf{o}$  (3.27) denote the orientation and let  $\boldsymbol{\omega}$  (3.31) denote the angular velocity of the body-fixed reference frame w.r.t. the inertial reference frame, expressed in the body-fixed reference frame. Given the states  $\mathbf{o}$  and  $\boldsymbol{\omega}$ , and considering the control inputs as the torques  $\boldsymbol{\tau} \in \mathbb{H}_p$ , a dynamic model of a second-order rigid-body attitude system can be given by

$$\dot{\mathbf{o}} = \frac{1}{2} \mathbf{o} \boldsymbol{\omega}, \quad (5.1)$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\Omega} + J^{-1} \boldsymbol{\tau} + J^{-1} \boldsymbol{\tau}_d, \quad \boldsymbol{\Omega} = -J^{-1}(\boldsymbol{\omega} \times J \boldsymbol{\omega}), \quad (5.2)$$

where  $J \in \mathbb{R}^{3 \times 3}$ ,  $J = J^\top > 0$  is the inertia tensor, and  $\boldsymbol{\tau}_d \in \mathbb{H}_p$  is the unknown magnitude-bounded disturbance. This system is called the **practical system**.

The problem solved in this Chapter is attitude trajectory-tracking, formalized below:

**Problem 4** (Attitude trajectory-tracking). *Given the rigid-body attitude system (5.1)-(5.2) with magnitude-bounded disturbances, design a controller that drives it to asymptotically converge to a small region around a given reference trajectory  $\mathbf{o}_d(t)$  with known velocity  $\boldsymbol{\omega}_d(t)$ , such that*

$$\dot{\mathbf{o}}_d = \frac{1}{2} \mathbf{o}_d \boldsymbol{\omega}_d, \quad (5.3)$$

with  $\boldsymbol{\omega}_d \in \mathbb{H}_p$  defined as the desired angular velocity of the target reference frame w.r.t. the inertial reference frame, expressed in the target reference frame. In other words, find a law for  $\boldsymbol{\tau}$  such that  $\mathbf{o}$  converges to a set around  $\mathbf{o}_d$  even when the system is disturbed.

Our approach to address Problem 4 is inspired by Zhang et al. [2021], while also incorporating several crucial additional elements that have not been previously considered. Besides using a multiplicative quaternion error as the basis for our theoretical development, we provide experimental results and consider time-varying reference attitude.

Our control design methodology consists of three parts.

- First, we find a nominal (*i.e.* when  $\boldsymbol{\tau}_d = 0$ ) feedback controller that ensures nominal asymptotic convergence to the reference  $\mathbf{o}_d(t)$ . The main advantage of our controller is in this step, where we solve the control problem for a given reference trajectory  $\mathbf{o}_d(t)$  rather than only solving the origin regulation problem, when  $\mathbf{o}_d \equiv 0$ , as in [Zhang et al., 2021].
- Next, we find a robust adaptive control for the error between the practical system and the previously found nominal trajectory, such that it is attracted to a small control attractive set around the origin.
- Finally, we find the practical system control law by combining the nominal and the robust adaptive controllers.

Similar to the controller proposed in the previous Chapter, the controller architecture is a combination of a nominal controller and a robust adaptive control. Figure 12 shows a diagram for this architecture. In this diagram, we have the nominal feedback loop with the nominal controller and the nominal system. Then, with the robust adaptive controller, we minimize the error between this nominal system and the practical system. Combining both control actions, we achieve our proposed controller.

## 5.2 Nominal control

Regarding the nominal performance, it is said to be the **nominal system** the attitude system with null disturbances, using variables represented with a top bar,  $\bar{\cdot}$ , as in

$$\dot{\bar{\mathbf{o}}} = \frac{1}{2} \bar{\mathbf{o}} \bar{\boldsymbol{\omega}}, \quad (5.4)$$

$$\dot{\bar{\boldsymbol{\omega}}} = \bar{\boldsymbol{\Omega}} + J^{-1} \bar{\boldsymbol{\tau}}, \quad \bar{\boldsymbol{\Omega}} = -J^{-1}(\bar{\boldsymbol{\omega}} \times J \bar{\boldsymbol{\omega}}). \quad (5.5)$$

The first part of our method is achieved based on the following Lemma.

**Lemma 4.** Let  $\mathbf{e} \equiv \mathbf{o}_d^* \bar{\mathbf{o}}$  be the attitude error quaternion between reference and nominal attitudes, with  $\mathbf{z}_o \equiv \text{Im}\{\mathbf{e}\} \in \mathbb{H}_p$  and the reference attitude  $\mathbf{o}_d(t)$  evolving according to (5.3). Define  $\mathbf{z}_\omega = \bar{\boldsymbol{\omega}} - \boldsymbol{\omega}_{d,\bar{\mathbf{B}}}$ , where the representation of the reference angular velocity  $\boldsymbol{\omega}_d$  in the body-fixed reference frame of the nominal system is given by  $\boldsymbol{\omega}_{d,\bar{\mathbf{B}}} = \mathbf{e}^* \boldsymbol{\omega}_d \mathbf{e}$ . The nominal system asymptotically tracks the reference attitude if the control law is defined as

$$\bar{\boldsymbol{\tau}} = \bar{\boldsymbol{\omega}} \times J \bar{\boldsymbol{\omega}} + J \dot{\boldsymbol{\omega}}_{d,\bar{\mathbf{B}}} + J \bar{\boldsymbol{\tau}}_L, \quad (5.6)$$

$$\bar{\boldsymbol{\tau}}_L = -k_o \mathbf{z}_o - K_\omega \mathbf{z}_\omega, \quad (5.7)$$

with  $k_o \in \mathbb{R}$ ,  $k_o > 0$ ,  $K_\omega \in \mathbb{R}^{3 \times 3}$ ,  $K_\omega = K_\omega^\top > 0$ .

*Proof.* Notice that the attitude error quaternion  $\mathbf{e}$  has unit norm, i.e.  $\mathbf{e} \mathbf{e}^* = 1$ , and it represents the orientation of the target reference frame w.r.t. the body-fixed reference frame of the nominal system, in the sense that  $\mathbf{a}_{\bar{\mathbf{B}}} = \mathbf{e}^* \mathbf{a}_d \mathbf{e}$  for any pure quaternions  $\mathbf{a}_{\bar{\mathbf{B}}}$  and  $\mathbf{a}_d$  representing the same vector w.r.t. the body-fixed nominal reference frame and the target reference frame, respectively. A complete alignment between these two reference frames means that  $\bar{\mathbf{o}} = \mathbf{o}_d$ , and therefore,  $\mathbf{e} = \mathbf{o}_d^* \bar{\mathbf{o}} = 1$ . The time derivative of the error is given by

$$\begin{aligned} \dot{\mathbf{e}} &= \dot{\mathbf{o}}_d^* \bar{\mathbf{o}} + \mathbf{o}_d^* \dot{\bar{\mathbf{o}}} = \left( \frac{1}{2} \mathbf{o}_d \boldsymbol{\omega}_d \right)^* \bar{\mathbf{o}} + \mathbf{o}_d^* \left( \frac{1}{2} \bar{\mathbf{o}} \bar{\boldsymbol{\omega}} \right), \\ &= \frac{1}{2} \boldsymbol{\omega}_d^* \mathbf{o}_d^* \bar{\mathbf{o}} + \frac{1}{2} \mathbf{o}_d^* \bar{\mathbf{o}} \bar{\boldsymbol{\omega}} = -\frac{1}{2} \boldsymbol{\omega}_d \mathbf{e} + \frac{1}{2} \mathbf{e} \bar{\boldsymbol{\omega}}. \end{aligned}$$

By considering  $\boldsymbol{\omega}_{d,\bar{\mathbf{B}}} = \mathbf{e}^* \boldsymbol{\omega}_d \mathbf{e}$  in the previous expression, we have that

$$\begin{aligned} \dot{\mathbf{e}} &= -\frac{1}{2} \mathbf{e} \boldsymbol{\omega}_{d,\bar{\mathbf{B}}} \mathbf{e}^* \mathbf{e} + \frac{1}{2} \mathbf{e} \bar{\boldsymbol{\omega}}, \\ \dot{\mathbf{e}} &= \frac{1}{2} \mathbf{e} (\bar{\boldsymbol{\omega}} - \boldsymbol{\omega}_{d,\bar{\mathbf{B}}}) = \frac{1}{2} \mathbf{e} \mathbf{z}_\omega, \end{aligned} \quad (5.8)$$

with  $\mathbf{z}_\omega = \bar{\boldsymbol{\omega}} - \boldsymbol{\omega}_{d,\bar{\mathbf{B}}}$ . Denoting  $\text{Re}\{\mathbf{e}\} = z_o$ , and using  $\text{Im}\{\mathbf{e}\} = \mathbf{z}_o$ , one has that  $\mathbf{e} = z_o + \mathbf{z}_o$ , and the last expression can be rewritten as

$$\dot{z}_o = -\frac{1}{2} \mathbf{z}_o^\top \mathbf{z}_\omega, \quad \dot{\mathbf{z}}_o = \frac{1}{2} (z_o \mathbf{z}_\omega + \mathbf{z}_o \times \mathbf{z}_\omega). \quad (5.9)$$

Conversely, from (5.5), we obtain

$$\dot{\mathbf{z}}_\omega = \bar{\boldsymbol{\Omega}} + J^{-1} \bar{\boldsymbol{\tau}} - \dot{\boldsymbol{\omega}}_{d,\bar{\mathbf{B}}}.$$

By choosing  $\bar{\boldsymbol{\tau}}$  from (5.6), we have that

$$\dot{\mathbf{z}}_\omega = -k_o \mathbf{z}_o - K_\omega \mathbf{z}_\omega. \quad (5.10)$$

Using the following candidate Lyapunov function:

$$V(\mathbf{e}, \mathbf{z}_\omega) = k_o \mathbf{z}_o^\top \mathbf{z}_o + k_o (1 - z_o)^2 + \frac{1}{2} \mathbf{z}_\omega^\top \mathbf{z}_\omega, \quad (5.11)$$

yields

$$\begin{aligned}
\dot{V} &= 2k_o \mathbf{z}_o^\top \left[ \frac{1}{2} (z_o \mathbf{z}_\omega + \mathbf{z}_o \times \mathbf{z}_\omega) \right] \\
&\quad - 2k_o (1 - z_o) \left( -\frac{1}{2} \mathbf{z}_o^\top \mathbf{z}_\omega \right) + \mathbf{z}_\omega^\top [-k_o \mathbf{z}_o - K_\omega \mathbf{z}_\omega], \\
\dot{V} &= k_o z_o \mathbf{z}_o^\top \mathbf{e}_\omega + k_o \mathbf{z}_o^\top \mathbf{e}_\omega - k_o z_o \mathbf{z}_o^\top \mathbf{e}_\omega - k_o \mathbf{e}_\omega^\top \mathbf{z}_o - \mathbf{e}_\omega^\top K_\omega \mathbf{e}_\omega, \\
\dot{V} &= -\mathbf{z}_\omega^\top K_\omega \mathbf{z}_\omega \leq 0.
\end{aligned} \tag{5.12}$$

Notice that  $\dot{V} = 0 \Leftrightarrow \mathbf{z}_\omega = 0$ , and the equilibrium point  $(\mathbf{z}_o = 0, \mathbf{z}_\omega = 0, z_o^2 = 1 - \|\mathbf{z}_o\|^2 = 1)$  is the only element in the largest positively invariant set where  $\dot{V} = 0$ , as it can be verified from the attitude error system dynamics given by (5.9) and (5.10). By LaSalle's invariance principle [Khalil, 2002, Corollary 4.2], the attitude error will converge asymptotically to the equilibrium point, and  $\mathbf{e} = z_o + \mathbf{z}_o \rightarrow 1 \Leftrightarrow \mathbf{o} \rightarrow \bar{\mathbf{o}}_d$ .  $\square$

### 5.3 Robust adaptive control

For the second part of our methodology, we propose a robust adaptive Backstepping control law to stabilize the error between the practical (5.1)-(5.2) and nominal (5.4)-(5.5) systems. Comparing both, the **error system** describes the mismatch between them.

Let  $\mathbf{o}_e = \bar{\mathbf{o}}^* \mathbf{o}$  be the error between the practical and nominal orientations. Accordingly, we obtain

$$\begin{aligned}
\dot{\mathbf{o}}_e &= \dot{\bar{\mathbf{o}}}^* \mathbf{o} + \bar{\mathbf{o}}^* \dot{\mathbf{o}} = \left( \frac{1}{2} \bar{\mathbf{o}} \bar{\boldsymbol{\omega}} \right)^* \mathbf{o} + \bar{\mathbf{o}}^* \left( \frac{1}{2} \mathbf{o} \boldsymbol{\omega} \right), \\
&= -\frac{1}{2} \bar{\boldsymbol{\omega}} \mathbf{o}_e + \frac{1}{2} \mathbf{o}_e \boldsymbol{\omega}.
\end{aligned}$$

By considering the nominal angular velocity w.r.t. the inertial reference expressed in the body-fixed frame  $\bar{\boldsymbol{\omega}}_B = \mathbf{o}_e^* \bar{\boldsymbol{\omega}} \mathbf{o}_e$ , yields

$$\dot{\mathbf{o}}_e = \frac{1}{2} \mathbf{o}_e (\boldsymbol{\omega} - \bar{\boldsymbol{\omega}}_B).$$

Defining  $\boldsymbol{\omega}_e = \boldsymbol{\omega} - \bar{\boldsymbol{\omega}}_B$ , results in

$$\dot{\mathbf{o}}_e = \frac{1}{2} \mathbf{o}_e \boldsymbol{\omega}_e. \tag{5.13}$$

Also, we have that

$$\begin{aligned}
\dot{\boldsymbol{\omega}}_e &= \dot{\boldsymbol{\omega}} - \dot{\bar{\boldsymbol{\omega}}}_B, \\
\dot{\boldsymbol{\omega}}_e &= \dot{\boldsymbol{\omega}} - (\dot{\mathbf{o}}_e^* \bar{\boldsymbol{\omega}} \mathbf{o}_e + \mathbf{o}_e^* \dot{\bar{\boldsymbol{\omega}}} \mathbf{o}_e + \mathbf{o}_e^* \bar{\boldsymbol{\omega}} \dot{\mathbf{o}}_e) \\
\dot{\boldsymbol{\omega}}_e &= \boldsymbol{\Omega} + J^{-1} \boldsymbol{\tau} + J^{-1} \boldsymbol{\tau}_d \\
&\quad - \mathbf{o}_e^* (\bar{\boldsymbol{\Omega}} + J^{-1} \bar{\boldsymbol{\tau}}) \mathbf{o}_e + \boldsymbol{\omega}_e \times \bar{\boldsymbol{\omega}}_B \\
\dot{\boldsymbol{\omega}}_e &= \boldsymbol{\Omega}_e + J^{-1} \boldsymbol{\tau}_d + J^{-1} \boldsymbol{\tau} - J^{-1} \bar{\boldsymbol{\tau}}_B,
\end{aligned} \tag{5.14}$$

where  $\boldsymbol{\Omega}_e = \boldsymbol{\Omega} - \mathbf{o}_e^* \bar{\boldsymbol{\Omega}} \mathbf{o}_e + \boldsymbol{\omega}_e \times \bar{\boldsymbol{\omega}}_B$  and  $\bar{\boldsymbol{\tau}}_B = J \mathbf{o}_e^* J^{-1} \bar{\boldsymbol{\tau}} \mathbf{o}_e$ .

### 5.3.1 Step 1

In the first step, we want  $\boldsymbol{\omega}_e$  to track a virtual control law  $\boldsymbol{\varphi}_e$  such that  $\boldsymbol{o}_e \rightarrow 1$  if  $\boldsymbol{\omega}_e = \boldsymbol{\varphi}_e$ . This virtual control law is chosen as

$$\boldsymbol{\varphi}_e = -k_{\varphi e} \boldsymbol{e}_o, \quad k_{\varphi e} > 0, \quad (5.15)$$

where  $\boldsymbol{e}_o \equiv \text{Im}\{\boldsymbol{o}_e\} \in \mathbb{H}_p$ . Also, let  $e_o \equiv \text{Re}\{\boldsymbol{o}_e\} \in \mathbb{R}$ .

**Lemma 5.** *Applying the virtual control law (5.15) in the system (5.13), the point  $\boldsymbol{e}_o = 1$  becomes an asymptotic stable equilibrium point.*

*Proof.* Since  $\boldsymbol{o}_e = e_o + \boldsymbol{e}_o$ , (5.13) can be rewritten as

$$\dot{e}_o = -\frac{1}{2} \boldsymbol{e}_o^\top \boldsymbol{\omega}_e, \quad \dot{\boldsymbol{e}}_o = \frac{1}{2} (e_o \boldsymbol{\omega}_e + \boldsymbol{e}_o \times \boldsymbol{\omega}_e). \quad (5.16)$$

Let  $V_{eo}$  be a candidate Lyapunov function

$$V_{eo}(\boldsymbol{o}_e) = (1 - e_o)^2 + \boldsymbol{e}_o^\top \boldsymbol{e}_o. \quad (5.17)$$

The time derivative of (5.17) is

$$\begin{aligned} \dot{V}_{eo} &= 2\boldsymbol{e}_o^\top \left[ \frac{1}{2} (e_o \boldsymbol{\omega}_e + \boldsymbol{e}_o \times \boldsymbol{\omega}_e) \right] - 2(1 - e_o) \left[ -\frac{1}{2} \boldsymbol{e}_o^\top \boldsymbol{\omega}_e \right], \\ \dot{V}_{eo} &= \boldsymbol{e}_o^\top \boldsymbol{\omega}_e. \end{aligned} \quad (5.18)$$

Applying (5.15) into (5.13), the time derivative of (5.17) yields

$$\dot{V}_{eo} = -k_{\varphi e} \|\boldsymbol{e}_o\|^2, \quad (5.19)$$

$$\dot{V}_{eo}(1) = 0, \quad (5.20)$$

$$\dot{V}_{eo}(\boldsymbol{o}_e) < 0, \quad \forall \boldsymbol{o}_e \in \mathbb{H}, \|\boldsymbol{o}_e\| = 1, \boldsymbol{o}_e \neq 1, \quad (5.21)$$

which guarantees asymptotic convergence of the error  $\boldsymbol{o}_e$  to 1, and therefore, ensures  $\boldsymbol{o} \rightarrow \bar{\boldsymbol{o}}$ .  $\square$

### 5.3.2 Step 2

For the last step, let  $\boldsymbol{e}_\omega = \boldsymbol{\omega}_e - \boldsymbol{\varphi}_e \in \mathbb{H}_p$  be the angular velocity error that encompasses this virtual control law. The error system is represented by

$$\dot{\boldsymbol{e}}_e = \frac{1}{2} \boldsymbol{o}_e \boldsymbol{\varphi}_e + \frac{1}{2} \boldsymbol{o}_e \boldsymbol{e}_\omega, \quad (5.22)$$

$$\dot{\boldsymbol{e}}_\omega = \boldsymbol{\Omega}_e + J^{-1} \boldsymbol{\tau}_e + J^{-1} \boldsymbol{\tau}_d - \dot{\boldsymbol{\varphi}}_e, \quad (5.23)$$

where  $\boldsymbol{\tau}_e \equiv \boldsymbol{\tau} - \bar{\boldsymbol{\tau}}_B$  is its control input.

Following the approach in Zhang et al. [2021], in order to design the robust adaptive control law, consider

$$\chi_o = \tau_d + J\Omega_e - J\dot{\varphi}_e \quad (5.24)$$

as a lumped term that contains the bounded disturbance  $\tau_d$ . Using an indirect method Zhu et al. [2011], Cai et al. [2008], the controller tackles  $\chi_o$  by focusing only on its bound. Accordingly, the following assumption is considered.

**Assumption 5** (Norm bounded disturbances). *There exist unknown constants  $\zeta_3$  and  $\zeta_4$  such that*

$$\|\chi_o\| \leq \zeta_3 + \zeta_4 \|\mathbf{e}_\omega\|. \quad (5.25)$$

These constants are estimated in the controller by adaptive parameters  $\hat{\zeta}_3$  and  $\hat{\zeta}_4$ .

Then, the robust adaptive control law for the error system is given by<sup>1</sup>

$$\tau_e = -k_{eo}\mathbf{e}_o - k_{e\omega}\mathbf{e}_\omega - \hat{\zeta}_3 \mathbf{tanh}(\hat{\zeta}_3 \mathbf{e}_\omega / \epsilon_o) - \hat{\zeta}_4 \mathbf{e}_\omega, \quad (5.26)$$

where  $k_{eo}, k_{e\omega}, \epsilon_o > 0$  are scalar gains, and the adaptive parameters  $\hat{\zeta}_3, \hat{\zeta}_4 > 0$  have the following update laws:

$$\dot{\hat{\zeta}}_3 = \rho_3 \|\mathbf{e}_\omega\|, \quad \dot{\hat{\zeta}}_4 = \rho_4 \|\mathbf{e}_\omega\|^2, \quad (5.27)$$

with  $\rho_3, \rho_4 > 0$ .

**Lemma 6.** *Considering Assumption 5, let  $V_{e\omega}$  be the following control Lyapunov function:*

$$V_{e\omega} = k_{eo}V_{eo} + \frac{1}{2}\mathbf{e}_\omega^\top J\mathbf{e}_\omega + \frac{\tilde{\zeta}_3^2}{2\rho_3} + \frac{\tilde{\zeta}_4^2}{2\rho_4}, \quad (5.28)$$

where

$$\begin{aligned} \tilde{\zeta}_3 &= \zeta_3 - \hat{\zeta}_3, \\ \tilde{\zeta}_4 &= \zeta_4 - \hat{\zeta}_4 \end{aligned} \quad (5.29)$$

are estimation errors. By applying (5.26) in the system (5.22)-(5.23),  $\dot{V}_{e\omega} < 0$  if the system is outside the attractive set  $\Delta_o$  given by

$$\Delta_o = \{(\mathbf{e}_o, \mathbf{e}_\omega) \mid \|\mathbf{e}_o\| \leq \delta_o, \|\mathbf{e}_\omega\| \leq \delta_\omega\}, \quad (5.30)$$

$$\delta_o = \sqrt{\frac{3\kappa\epsilon_o}{k_{eo}k_{\varphi_e}}}, \quad \delta_\omega = \sqrt{\frac{3\kappa\epsilon_o}{k_{e\omega}}}, \quad (5.31)$$

with  $\kappa \approx 0.2785$ . Since we are looking for an upper bound, consider  $\kappa = 0.2786$ .

<sup>1</sup> The function  $\mathbf{tanh}(\mathbf{q})$  is defined in terms of exponential maps. It is noteworthy that the bold  $\mathbf{tanh}(\mathbf{q})$  is an abuse of notation, in which we actually mean the element-wise hyperbolic-tangent, applied to each imaginary field of  $\mathbf{q}$ .

*Proof.* Consider Lemma 5 and the function (5.28). The time derivative of  $V_{eow}$  is given by

$$\begin{aligned}\dot{V}_{eow} = & -k_{eo}k_{\varphi e}\|\mathbf{e}_o\|^2 + k_{eo}\mathbf{e}_o^\top \mathbf{e}_\omega - \tilde{\zeta}_3\dot{\hat{\zeta}}_3/\rho_3 - \tilde{\zeta}_4\dot{\hat{\zeta}}_4/\rho_4 \\ & + \mathbf{e}_\omega^\top (J\boldsymbol{\Omega}_e + \boldsymbol{\tau}_d - J\dot{\boldsymbol{\varphi}}_e + \boldsymbol{\tau}_e).\end{aligned}\quad (5.32)$$

Then, substituting (5.26) and (5.27) in (5.32) yields

$$\begin{aligned}\dot{V}_{eow} = & -k_{eo}k_{\varphi e}\|\mathbf{e}_o\|^2 - k_{e\omega}\|\mathbf{e}_\omega\|^2 + \mathbf{e}_\omega^\top \chi_o \\ & - \mathbf{e}_\omega^\top \hat{\zeta}_3 \tanh(\hat{\zeta}_3 \mathbf{e}_\omega / \epsilon_o) - \hat{\zeta}_4 \|\mathbf{e}_\omega\|^2 \\ & - \tilde{\zeta}_3 \|\mathbf{e}_\omega\| - \tilde{\zeta}_4 \|\mathbf{e}_\omega\|^2.\end{aligned}\quad (5.33)$$

Considering Assumption 5 leads to

$$\begin{aligned}\dot{V}_{eow} \leq & -k_{eo}k_{\varphi e}\|\mathbf{e}_o\|^2 - k_{e\omega}\|\mathbf{e}_\omega\|^2 \\ & - \hat{\zeta}_3 \|\mathbf{e}_\omega\| \tanh(\hat{\zeta}_3 \|\mathbf{e}_\omega\| / \epsilon_o) + \hat{\zeta}_3 \|\mathbf{e}_\omega\|.\end{aligned}\quad (5.34)$$

Using both the following properties, Zhang et al. [2021], Polycarpou and Ioannou [1996],

(i)  $\|\mathbf{q}\| \leq \sum_{i=1}^n |q_i|, \forall \mathbf{q} \in \mathbb{R}^n$ ; and (ii)  $0 \leq |a| - a \tanh(a/\epsilon_p) \leq \kappa \epsilon_p, \forall a \in \mathbb{R}$ ; yields:

$$\dot{V}_{eow} \leq -k_{eo}k_{\varphi e}\|\mathbf{e}_o\|^2 - k_{e\omega}\|\mathbf{e}_\omega\|^2 + 3\kappa\epsilon_o. \quad (5.35)$$

In which  $\dot{V}_{eow} < 0$  for  $\|\mathbf{e}_o\| > \sqrt{\frac{3\kappa\epsilon_o}{k_{eo}k_{\varphi e}}}$  or  $\|\mathbf{e}_\omega\| > \sqrt{\frac{3\kappa\epsilon_o}{k_{e\omega}}}$ .  $\square$

The former inequality means that  $\delta_o = \sqrt{\frac{3\kappa\epsilon_o}{k_{eo}k_{\varphi e}}}$  bounds an attractive set for the attitude error  $\mathbf{e}_o$ .

From the latter inequality and using the triangle inequality,  $\|\mathbf{e}_\omega\| \geq \|\boldsymbol{\omega}_e\| - \|\boldsymbol{\varphi}_e\|$ , with  $\|\boldsymbol{\varphi}_e\| = k_{\varphi e}\|\mathbf{e}_o\| \leq k_{\varphi e}\delta_o$  and finally  $\delta_{\omega e} = \delta_\omega + k_{\varphi e}\delta_o$  bounds an attractive set for the velocity error  $\boldsymbol{\omega}_e$ .

### 5.3.3 Practical system control

The last part of our method is to combine the controller design. Now, we can present the complete controller scheme, which is given by

$$\boldsymbol{\tau} = \bar{\boldsymbol{\tau}}_B + \boldsymbol{\tau}_e, \quad (5.36)$$

The proposed method yields the following result.

**Theorem 3** (Attitude Controller). *Applying (5.36) in the practical system (5.1)-(5.2), the system trajectories asymptotically converge to a control attractive set around the time-varying reference orientation  $\mathbf{o}_d(t)$ .*

*Proof.* Lemmas 4 states that the nominal system (5.4)-(5.5) asymptotically converges to the reference orientation. Furthermore, the error between the practical system and the nominal trajectory is constrained as a result of Lemmas 6.  $\square$

# Chapter 6

## Results and discussion

In this Chapter, we show and discuss extensive simulations and real vehicle experiments to validate the proposed method. We first start by detailing the implementation and the hardware setup. Then, we present simulation results for the proposed controllers. Finally, we present real-world attitude experiments using the Crazyflie platform.

### 6.1 Implementation and setup

#### 6.1.1 Notes

The implementation is in digital and discrete domains for both the simulations and experiments. The controller algorithm is then adapted to accommodate signal sampling and quantization. For that reason, some considerations must be taken into account when implementing our method.

#### Derivatives

In the presented results, all signal derivatives are numerically approximated following the one-step-back approach as in

$$\dot{x} \approx \frac{\Delta x_k}{\Delta t} = \frac{(x_k - x_{k-1})}{\Delta t}. \quad (6.1)$$

for any signal  $x$ , where  $x_k$  is the  $k$ -th sample of  $x$  and  $\Delta t$  is the sampling interval. This may introduce a small following error in feed-forward terms.

#### Signal saturation

The proposed controller does not take into account the practical reality of control input saturation. This might come from the real system limitations or floating-point variables precision.

Some factors that can lead to saturation in the proposed controller are the adaptive gains. Since their adaptive laws are always greater or equal to zero, they never decrease. In fact, as we do not have guarantees that the errors will reach zero, those parameters may increase indefinitely, leading the discrete controller to be more and more aggressive until it becomes saturated or overflowed. In this case, we saturate the adaptive gains  $\hat{\zeta}_i$  and adopted small increasing rate  $\rho_i$ ,  $i \in [1, 2, 3, 4]$ .

### Online optimization

For the computation of the curve vector field, we must find an optimal  $s_*$  from (3.48). To compute the minimization, we used the Golden Section Search algorithm [Press, 2007]. For self-intersecting curves, we constrained the domain of the search as in [Nunes et al., 2022].

### Vector field singularities

The vector field obstacle avoidance method from [Nunes et al., 2022] is prone to discontinuities in the tangent vector (3.63) for three dimensions or more. This will happen if the vector from the robot to the obstacle points in the same direction as the curve field.

### Uniqueness of vector field solution

The vector field method is not uniquely defined in the workspace. However, from the method [Rezende et al., 2021], when the solution is not unique, one can simply pick one of the possible solutions and the system will eventually come out of the region where the solution is not unique. To overcome this in the implementation, two options are: (i) get the first solution; or (ii) get the solution closest to the last computed vector. We implemented option (i) for simplicity.

### Unit quaternion integration

For the attitude quaternion integration, we used the exponential method as in (3.38), followed by normalization to ensure  $\|\mathbf{o}_k\| = 1$ ,  $\forall k$ .

## 6.1.2 Simulation code

The simulations were implemented in Python3. Starting with the quaternion algebra used to describe the method, an auxiliary quaternion package was developed to support this algebra. Then, two separate packages were developed: position and attitude. The former included one module to represent the system, one for the controller, one for the simulator, and a last one for the obstacles, making the package suitable for the Position

Controller development. The latter included system, controller, and simulator, making them suitable for the Attitude Controller.

The code is available at [github.com/ArthurHDN/quadcopter\\_control\\_sim](https://github.com/ArthurHDN/quadcopter_control_sim).

### 6.1.3 Crazyflie 2.1 platform

Figure 15 shows a picture of the open-source flying development platform Crazyflie 2.1 used for the experiments.



Figure 15 – Crazyflie 2.1 platform picture, available at [bitcraze.io/products/crazyflie-2-1](https://bitcraze.io/products/crazyflie-2-1).

We considered the mass and the inertia tensor given in the datasheet<sup>1</sup> and experiments from [Landry et al., 2015].

$$m = 0.03 \text{ kg}, \quad (6.2)$$

$$J = \begin{bmatrix} 2.3951 & 0 & 0 \\ 0 & 2.3951 & 0 \\ 0 & 0 & 3.2347 \end{bmatrix} \cdot 10^{-5} \text{ kg m}^2. \quad (6.3)$$

<sup>1</sup> Crazyflie 2.1 datasheet available at [bitcraze.io](https://bitcraze.io).

### 6.1.4 Adding a custom controller to Crazyflie firmware

The Crazyflie platform has a well-structured framework that allows to develop applications at different abstraction levels. Since we are developing a new controller, the most suitable place to incorporate it in the Crazyflie framework is the firmware. The firmware provided by the vendor is organized in such a way that developing a new controller is achieved by adding an out-of-tree<sup>2</sup> application layer with the desired implementation.

Our implementation for the firmware is also available at [github.com/ArthurHDN/quadcopter\\_controller\\_crazyflie\\_firmware](https://github.com/ArthurHDN/quadcopter_controller_crazyflie_firmware).

### 6.1.5 Experiments architecture

For the experiments, we evaluated the Attitude Controller in the quadcopter. To achieve that, we locked the quadcopter in a Gimbal Ring Test Bench. The test bench allowed the vehicle to rotate in the three axes while keeping the same position, making it suitable for attitude-only experiments.

Figure 16 shows the Test Bench Gimbal Ring. On the left is the 3D model, and on the right is the prototype. The test bench was built with 3D-printed parts, four thin stainless steel bars, and six bearings. The 3D model is available at [github.com/ArthurHDN/crazyflie-gimbal-ring-test-bench](https://github.com/ArthurHDN/crazyflie-gimbal-ring-test-bench).

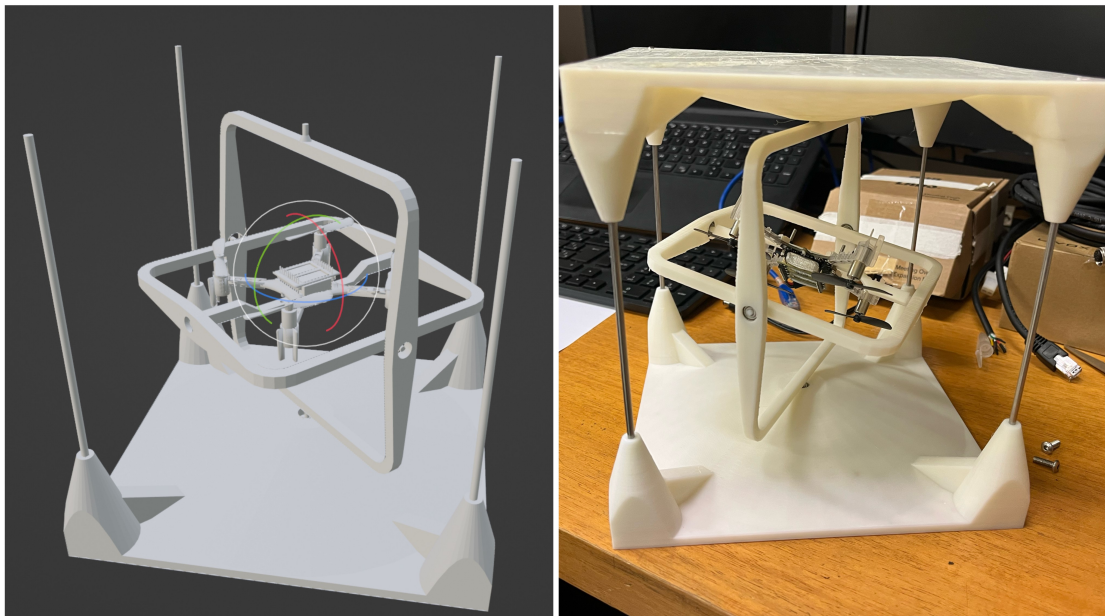


Figure 16 – Test Bench Gimbal Ring. On the left is the 3D model and on the right is the prototype.

<sup>2</sup> Out-of-tree development, most used in Linux Kernel development, refers to a software implementation that is developed and built outside the default source tree. In this case, it means that we do not have to change the firmware code itself, but instead create a separate application outside of the Crazyflie firmware source tree [fastbitlab.com/building-a-linux-kernel-module](https://fastbitlab.com/building-a-linux-kernel-module).

## 6.2 Position Controller simulations

### 6.2.1 Position Controller simulation I: disturbance-free and obstacle-free

An animation video for this simulation is available at [youtu.be/X4N3m3C1rcM](https://youtu.be/X4N3m3C1rcM).

For the first simulation of the Position Controller, we aim to show an obstacle-free and disturbance-free scenario. In this case, the system was controlled to follow a self-intersecting static curve given by

$$\mathbf{r}(s, t) = 1.5 \left( \sin(3s) \cos(5s) \hat{\mathbf{i}} + \sin(3s) \sin(5s) \hat{\mathbf{j}} + (\cos(3s) + 1.5) \hat{\mathbf{k}} \right) \text{ m}. \quad (6.4)$$

The system's initial conditions were

$$\mathbf{p}(0) = \bar{\mathbf{p}}(0) = \left( 3\hat{\mathbf{i}} - 1\hat{\mathbf{j}} \right) \text{ m}, \quad (6.5)$$

$$\mathbf{v}(0) = \bar{\mathbf{v}}(0) = 0 \text{ m s}^{-1}. \quad (6.6)$$

We used  $m$  from (6.2).

Although this is an obstacle-free scenario and therefore the collision avoidance was not activated, the obstacle avoidance parameters were

$$\lambda = 1.5 \text{ m}, \quad (6.7)$$

$$D_{in} = 2 \text{ m}, \quad (6.8)$$

$$D_{in0} = 3 \text{ m}. \quad (6.9)$$

The nominal part parameters were

$$k_G = 2.5 \text{ m}^{-1}, \quad (6.10)$$

$$v_r = 1 \text{ m s}^{-1}, \quad (6.11)$$

$$K_p = 0.1 \text{ s}^{-2}, \quad (6.12)$$

$$K_v = 10 \text{ s}^{-1}. \quad (6.13)$$

The robust parameters were

$$k_{\mathcal{F}e} = 1.1 \text{ s}^{-1}, \quad (6.14)$$

$$k_{ep} = 1 \text{ kg s}^{-2}, \quad (6.15)$$

$$k_{ev} = 1 \text{ kg s}^{-1}, \quad (6.16)$$

$$\epsilon_p = 0.4 \text{ s N}^{-1} \text{ m}^{-1}, \quad (6.17)$$

$$\rho_1 = 0.05 \text{ N m}, \quad (6.18)$$

$$\rho_2 = 0.05 \text{ kg m}^2. \quad (6.19)$$

The goal of this simulation is to show the convergence to the path in the ideal case of obstacle-free and disturbance-free scenarios.

In this simulation, the null disturbances were given by

$$\mathbf{f}_d = 0 \text{ N.} \quad (6.20)$$

The system followed the path and achieved zero error as in Figure 17, which shows the distance between the practical system and the curve in continuous lines and the distance between the nominal system and the curve in dashed lines, where both overlap.

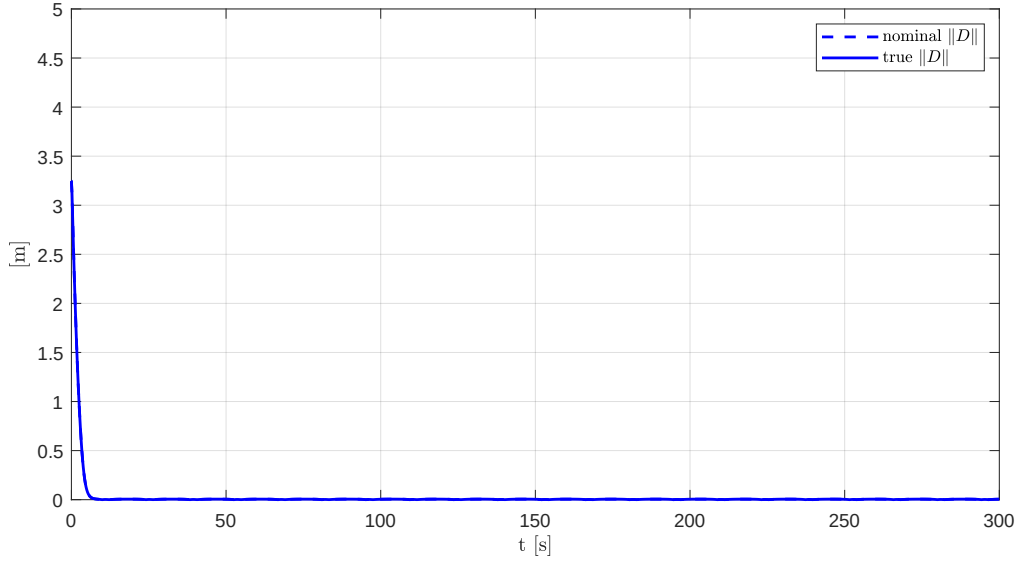


Figure 17 – Position Controller simulation I: Distance between the practical system and the curve in continuous lines and distance between the nominal system and the curve in dashed lines (both overlap).

Figure 18 shows the evolution in time for control inputs for the nominal system (top), error system (middle), and practical system (bottom). We can see that the robust control law is zero which is expected since no uncertainty existed for this simulation.

The errors between the nominal and practical systems stayed constant at zero for this simulation, *i.e.*

$$\begin{aligned} \|\mathbf{p} - \bar{\mathbf{p}}\| &= 0 \text{ m,} \\ \|\mathbf{v} - \bar{\mathbf{v}}\| &= 0 \text{ m s}^{-1}. \end{aligned}$$

Furthermore, the adaptive gains also stayed at zero, *i.e.*

$$\begin{aligned} \hat{\zeta}_1 &= 0 \text{ N,} \\ \hat{\zeta}_2 &= 0 \text{ kg s}^{-1}. \end{aligned}$$

Figure 19 shows the evolution in time of the system states.

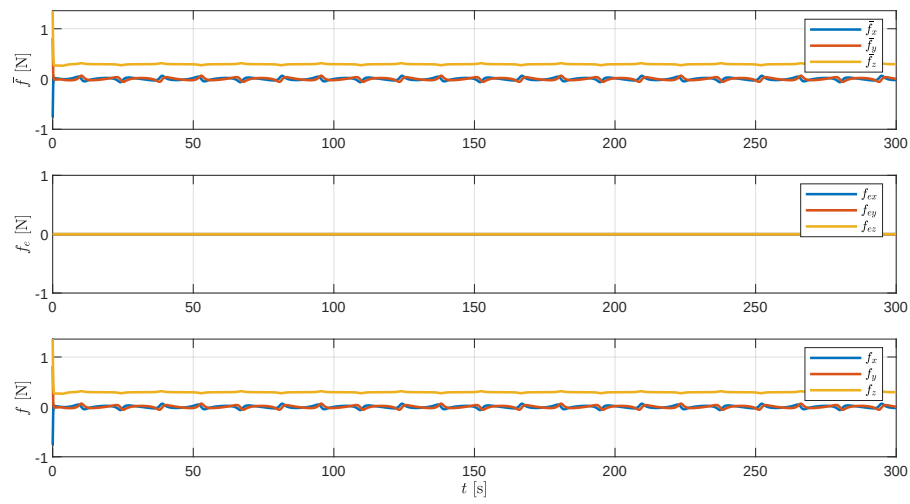


Figure 18 – Position Controller simulation I: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom).

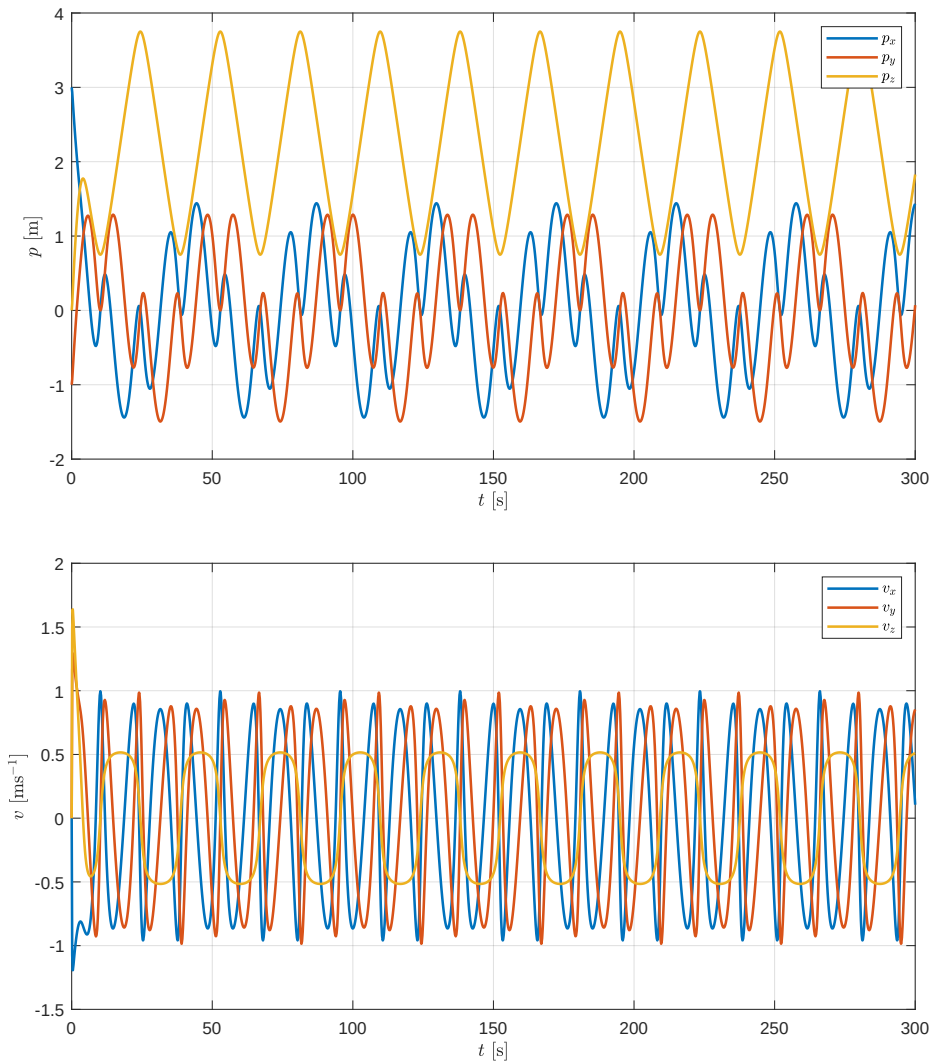


Figure 19 – Position Controller simulation I: Evolution in time of systems states.

## 6.2.2 Position Controller simulation II: constant disturbances

An animation video for this simulation is available at [youtu.be/XN-L4JjIdno](https://youtu.be/XN-L4JjIdno).

For the second simulation of the Position Controller, we repeated the scenario from the previous simulation but adding constant disturbances. To ensure the same scenario as the previous simulation:

- The target path was given by (6.4);
- The initial conditions were given by (6.5)-(6.6);
- The controller parameters were given by the equations (6.7) to (6.19).

Figure 20 shows the constant disturbance, which was given by

$$\mathbf{f}_d = \left( -0.1\mathbb{1}_{100}\hat{\mathbf{i}} + 0.02\mathbb{1}_{150}\hat{\mathbf{j}} - 0.2\mathbb{1}_{200}\hat{\mathbf{k}} \right) \text{ N}, \quad (6.21)$$

where  $\mathbb{1}_{t_0}$  is the unit step starting at time  $t = t_0$ .

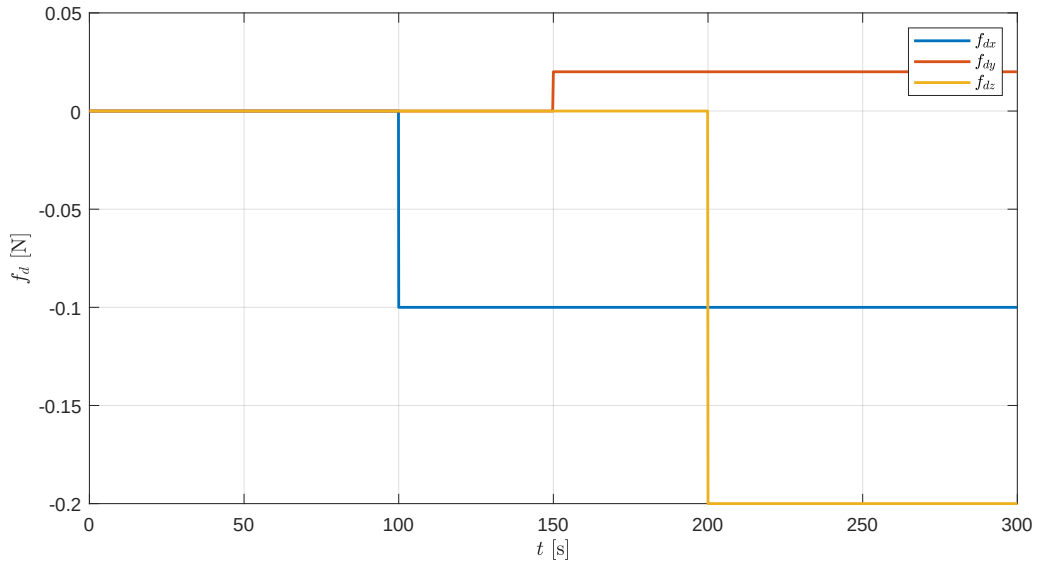


Figure 20 – Position Controller simulation II: Disturbance  $\mathbf{f}_d$ .

The system successfully followed the path with a small error due to the constant disturbances. Figure 21 shows both the nominal and practical distance to the path in continuous and dashed lines, respectively.

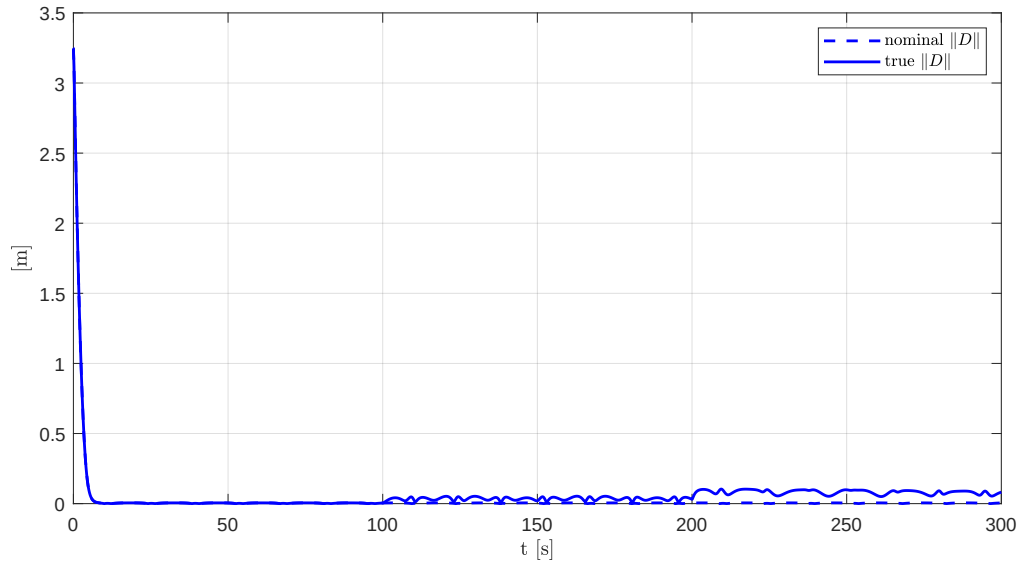


Figure 21 – Position Controller simulation II: Distance between the practical system and the curve in continuous lines and distance between the nominal system and the curve in dashed lines.

Figure 22 shows the evolution in time for control inputs for the nominal system (top), error system (middle), and practical system (bottom).

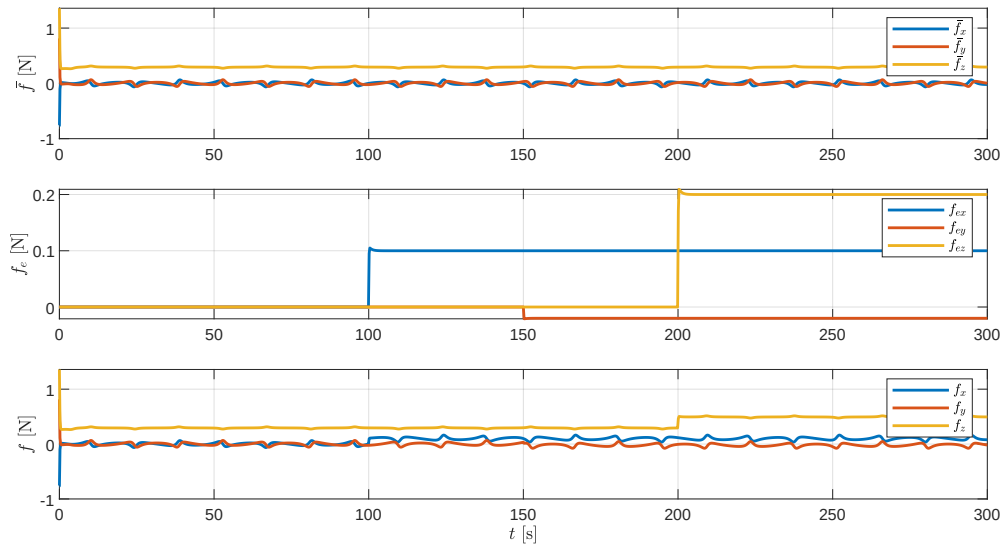


Figure 22 – Position Controller simulation II: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom).

The error between the practical and nominal systems was limited by  $\Delta_p$ , as shown in the disturbance mitigation for position and velocity in Figure 23.

Figure 24 shows the evolution in time of the adaptive gains, which increased indefinitely, since the errors never converged to zero as expected and shown in Figure 23.

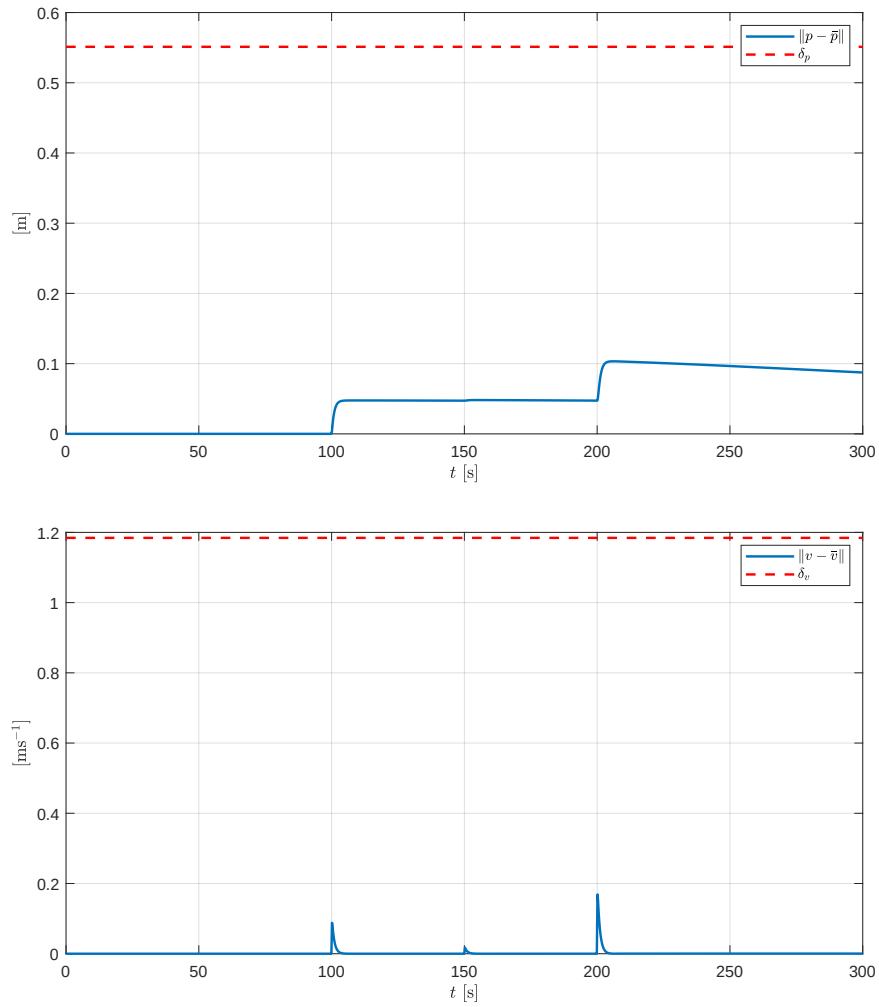


Figure 23 – Position Controller simulation II: Error between practical and nominal positions (top) and velocities (bottom).

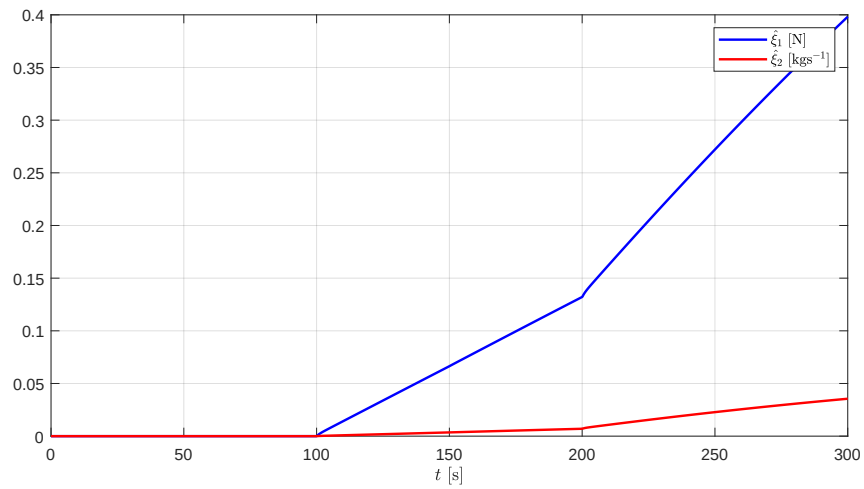


Figure 24 – Position Controller simulation II: Evolution in time of adaptive gains.

Figure 25 shows the evolution in time of the system states.

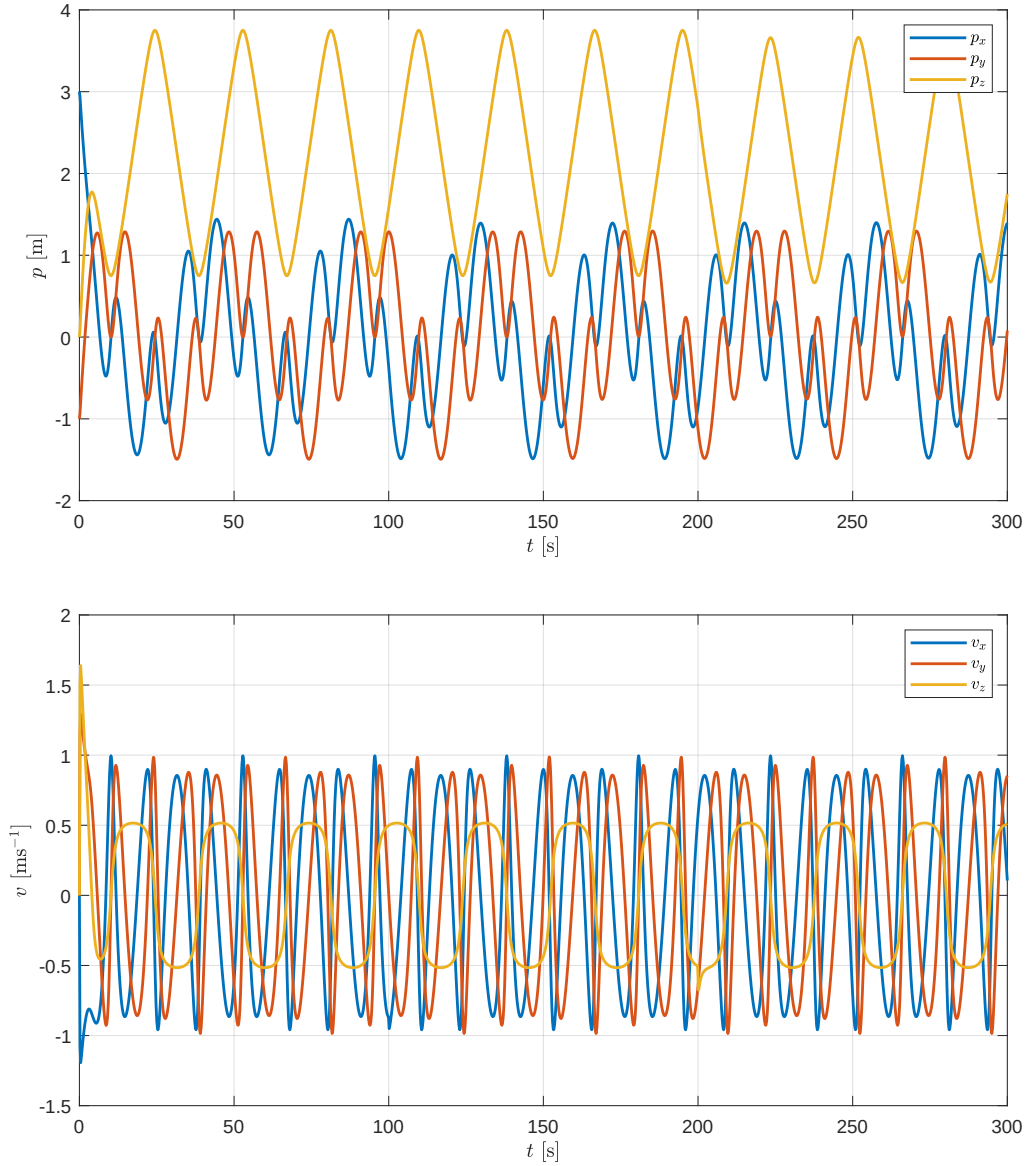


Figure 25 – Position Controller simulation II: Evolution in time of systems states.

### 6.2.3 Position Controller simulation III: disturbances and obstacles

An animation video for this simulation is available at [youtu.be/0Mtk33f2Qic](https://youtu.be/0Mtk33f2Qic).

For the third simulation of the Position Controller, we simulated a system with disturbances, model imperfections, and obstacles. For this case, the target is a moving circular path, with a parametric representation

$$\mathbf{r}(s, t) = \left( 5 \cos(s) \hat{\mathbf{i}} + 5 \sin(s) \hat{\mathbf{j}} + \left( 5 + 2 \cos \left( \frac{t}{10} \right) \right) \hat{\mathbf{k}} \right) \text{ m}, \quad (6.22)$$

in which four moving spherical obstacles were nearby.

The initial conditions were the same as the previous, (6.5)-(6.6). As for the controller, we introduced a +10% parameter estimation error given by

$$\bar{m} = 1.1m, \quad (6.23)$$

with  $m$  from (6.2).

The controller gains were the same as the previous, from (6.7) to (6.19)

The system was disturbed according to the equations below, in which Figure 26 shows an example of the generated disturbances:

$$\mathbf{f}_d = 0.02 \left( \hat{\mathbf{i}} f_{dx} + \hat{\mathbf{j}} f_{dy} + \hat{\mathbf{k}} f_{dz} \right) \text{ N}, \quad (6.24)$$

$$f_{dx} = \mathcal{N}(0, 1) \mathbf{1}_{25}, \quad (6.25)$$

$$f_{dy} = \mathcal{N}(0, 1) \mathbf{1}_{25} + 10 \sin(t/2) \mathbf{1}_{50}, \quad (6.26)$$

$$f_{dz} = \mathcal{N}(0, 1) \mathbf{1}_{25} + 10 \mathbf{1}_{75}, \quad (6.27)$$

where  $\mathcal{N}(\mu, \sigma)$  is a Gaussian sample of  $\mu$  mean and  $\sigma$  deviation.

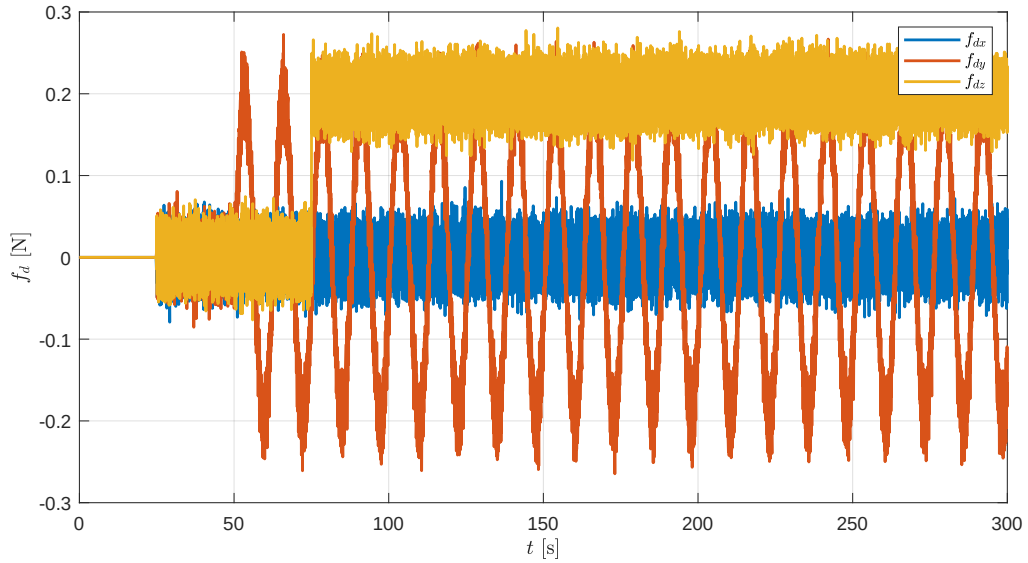


Figure 26 – Position Controller simulation III: Disturbance  $\mathbf{f}_d$ .

The system successfully followed the path when it was safe and it never collided. This can be seen in Figure 27, that shows the simulation distances. In blue is the distance from the system to the curve, and in red is the distance from the system to the closest obstacle. The parameters  $\lambda$  and  $D_{in0}$  are in dashed green and pink lines, respectively.

We can see that the system followed the path when it was safe by accomplishing a near-zero distance to the path in the majority of the simulation time. In general, when the red line (the distance to the obstacles  $\|\mathbf{D}_o\|$ ) enters the region below the pink dashed line (the switching threshold  $D_{in0}$ ) the blue line (the distance to the curve  $\|\mathbf{D}\|$ ) starts to

increase, meaning that the system had to move away for the curve in order to not collide with obstacles. Additionally, the obstacle detour is taken approximately at the parameter  $\lambda$ , as expected. This can be seen when the red line stays close to the dashed green line.

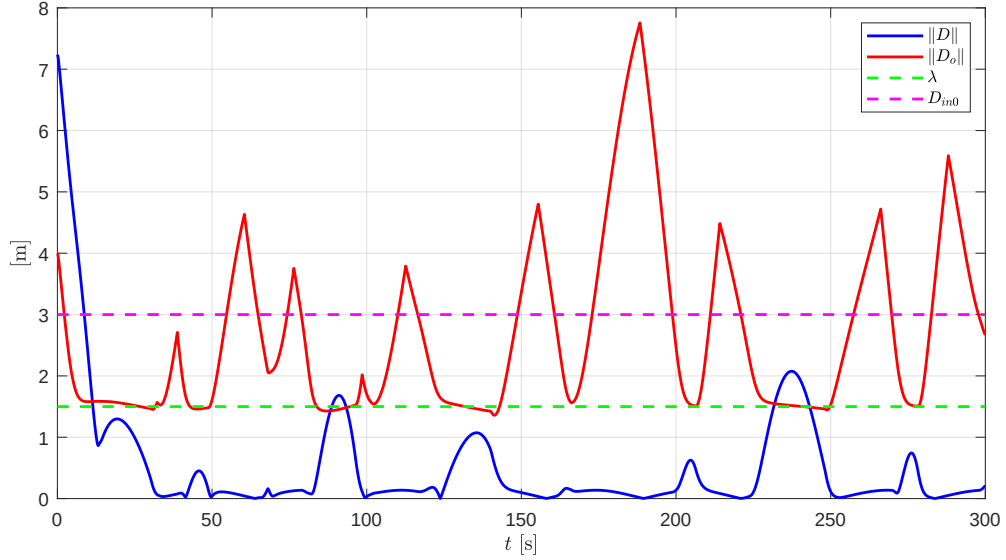


Figure 27 – Position Controller simulation III: Distances between the system and the curve in blue, between the system and the obstacles in red.

Figure 28 shows the evolution in time for control inputs for the nominal system (top), error system (middle), and practical system (bottom). We can see that the robust control law almost dominates the response, apart from some segments where the nominal control law has some chattering due to the field switching.

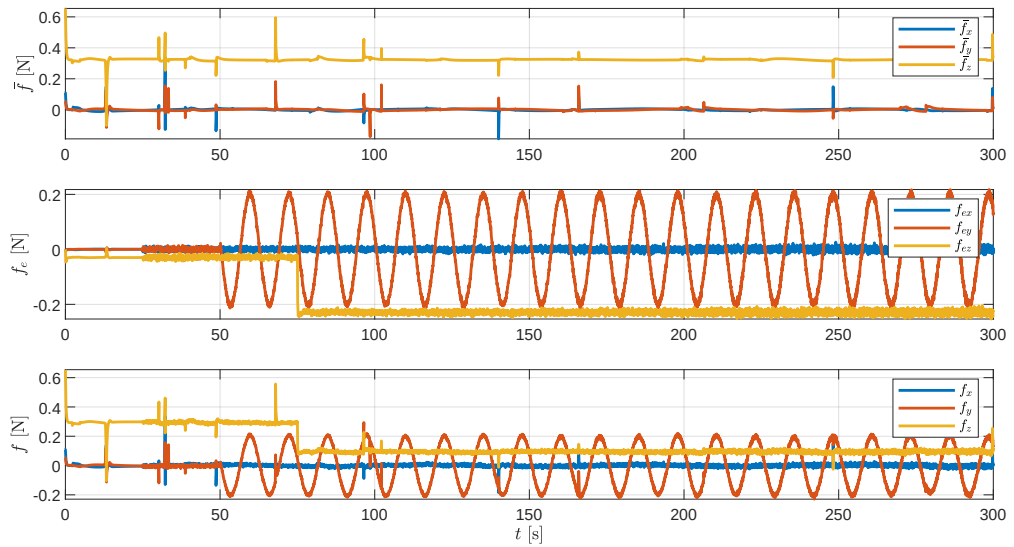


Figure 28 – Position Controller simulation III: Evolution in time of control inputs for the nominal system (top), error system (middle), and practical system (bottom).

The error between the practical and nominal systems was limited by  $\Delta_p$ , as shown in the disturbance mitigation for position and velocity in Figure 29.

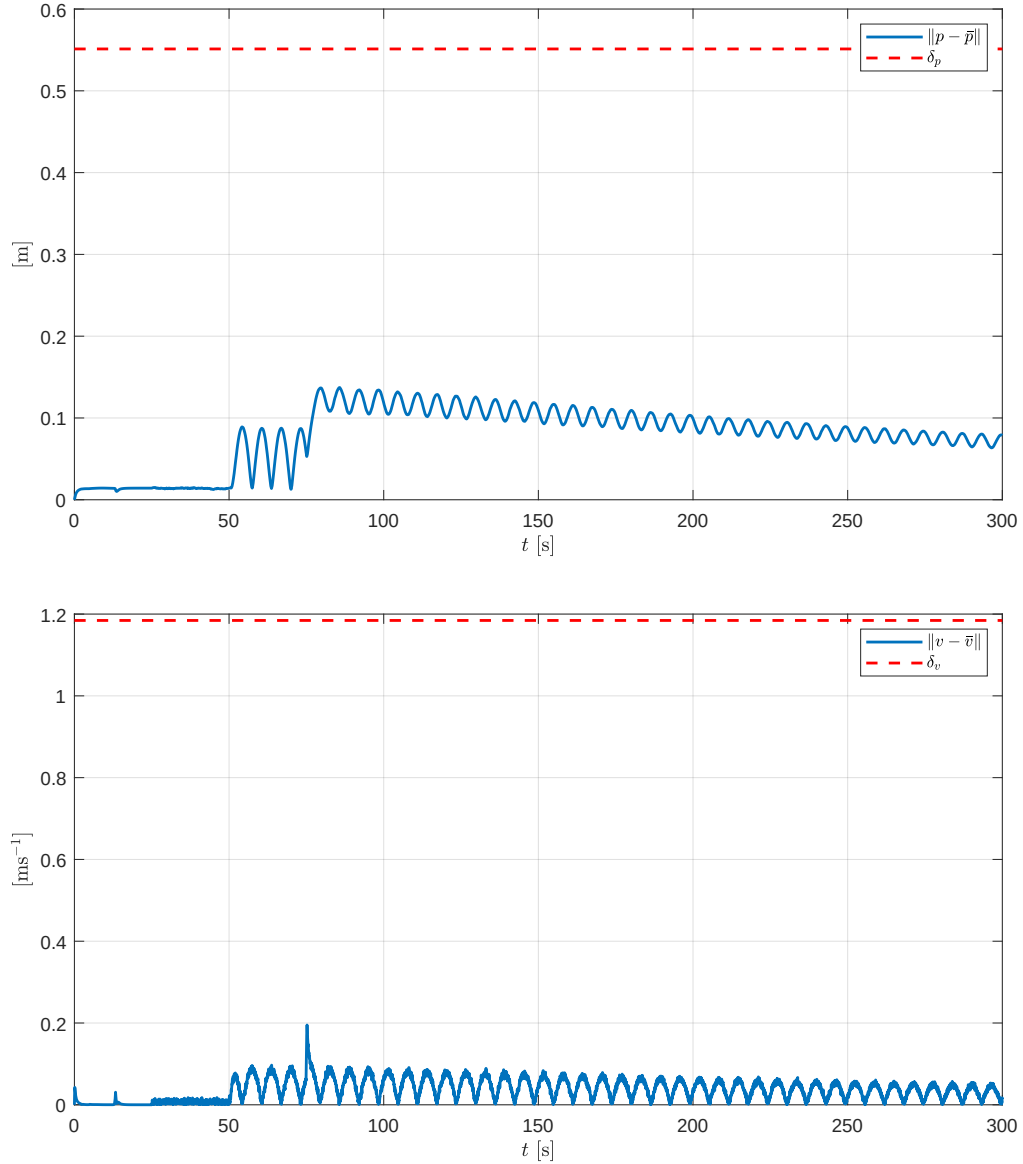


Figure 29 – Position Controller simulation III: Error between practical and nominal positions (top) and velocities (bottom).

Figure 30 shows the evolution in time of the adaptive parameters, that did not saturate in a 300s simulation, but increased indefinitely since the errors never converged to zero as expected and shown in Figure 29.

Figure 31 shows the evolution in time of the system states, which followed the moving circle as well as avoided obstacles according to the trade-off intrinsic to our control methodology.

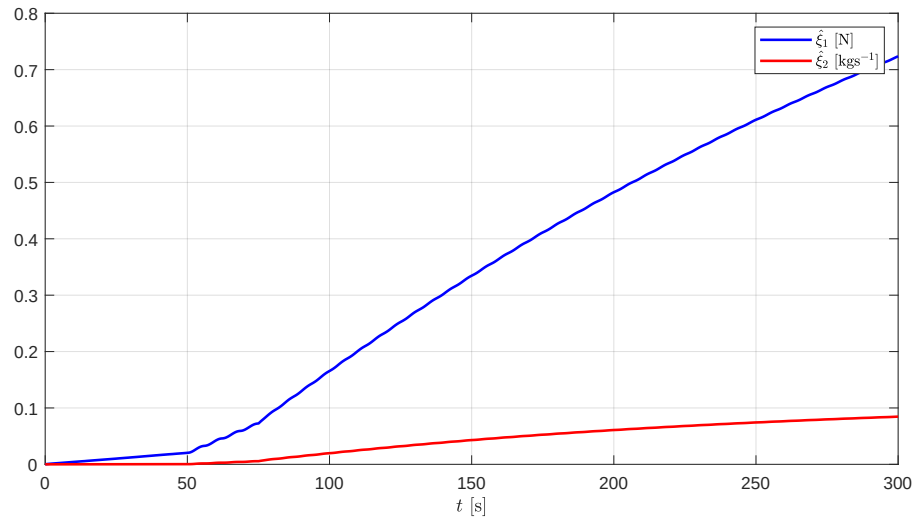


Figure 30 – Position Controller simulation III: Evolution in time of adaptive gains.

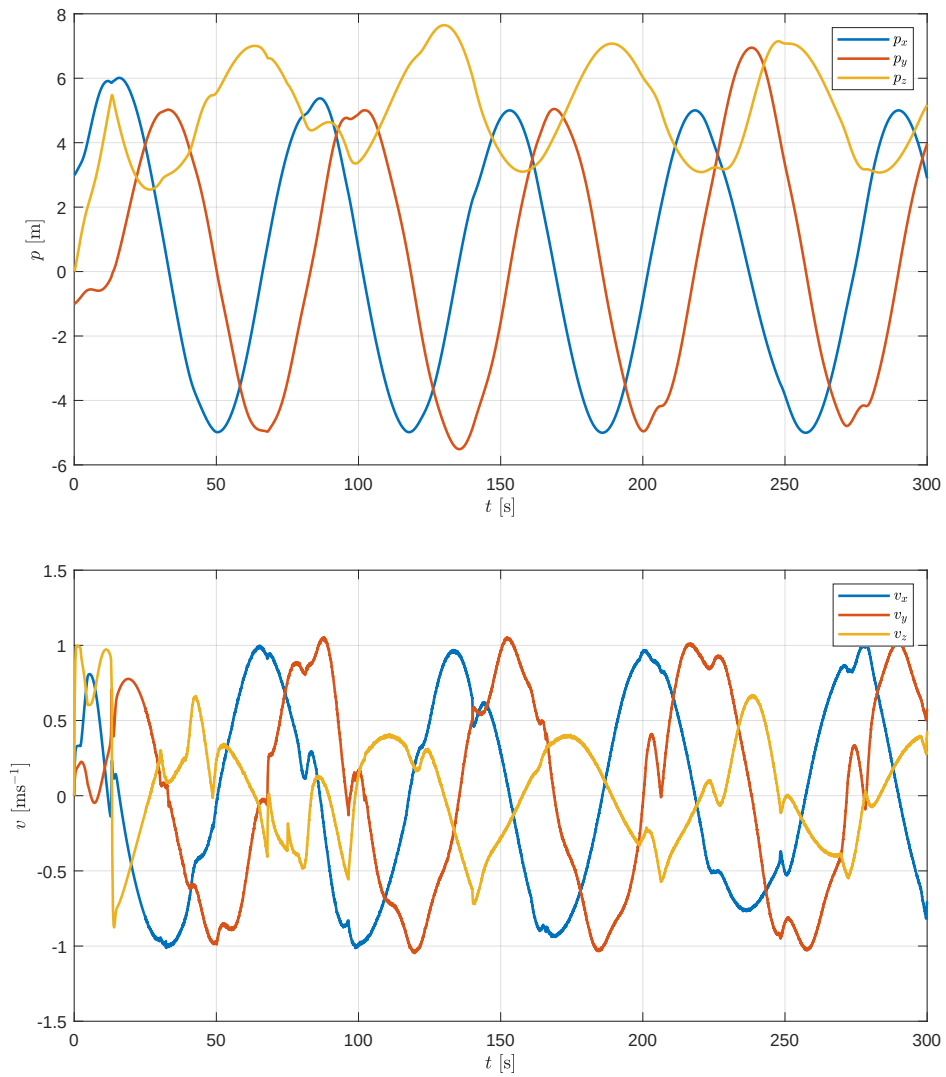


Figure 31 – Position Controller simulation III: Evolution in time of systems states.

### 6.2.4 Position Controller simulation IV: disturbances and obstacles with nominal controller

An animation video for this simulation is available at [youtu.be/sjbdzh8iuHI](https://youtu.be/sjbdzh8iuHI).

For the last simulation of the Position Controller, we removed the robust adaptive action by controlling the system only with the nominal controller and repeated the scenario from the previous simulation. This enables a comparison between both results.

To remove the robust adaptive control, we closed the control loop with the practical system and the nominal controller as shown in the diagram of Figure 32. When comparing to Figure 12 with the complete controller diagram, it is clear that for this simulation the robust adaptive controller was removed.

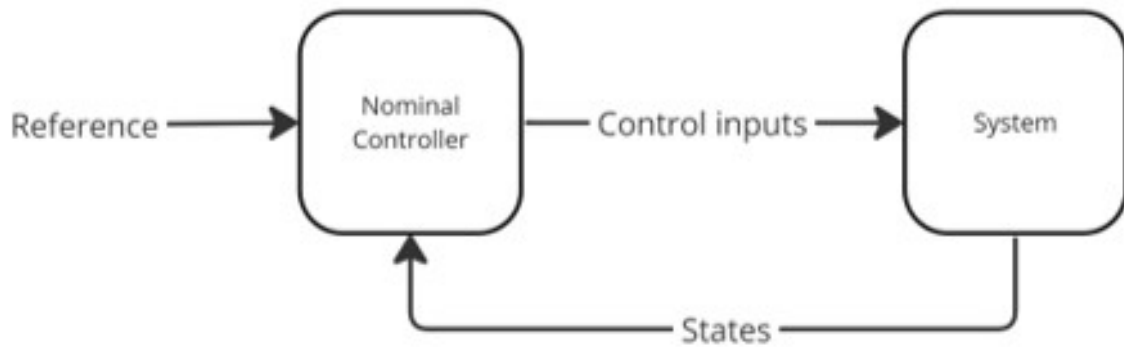


Figure 32 – Controller architecture diagram for the simulations with nominal controller only. In this diagram, the control loop is closed with the nominal controller and the practical system.

To ensure the same scenario as the previous simulation:

- The target path was given by (6.22);
- The initial conditions were given by (6.5)-(6.6);
- We introduced the parameter estimation error (6.23);
- The controller gains were given by the equations (6.7) to (6.13). Note that we do not have the robust adaptive gains for this simulation;
- The disturbance generator function was given by (6.24).

The nominal controller is also error-based, so it is expected that it controls the system to follow the path and avoid obstacles. This can be seen in Figure 33, which shows the simulation distances. In blue is the distance from the system to the curve, and in red is the distance from the system to the closest obstacle.

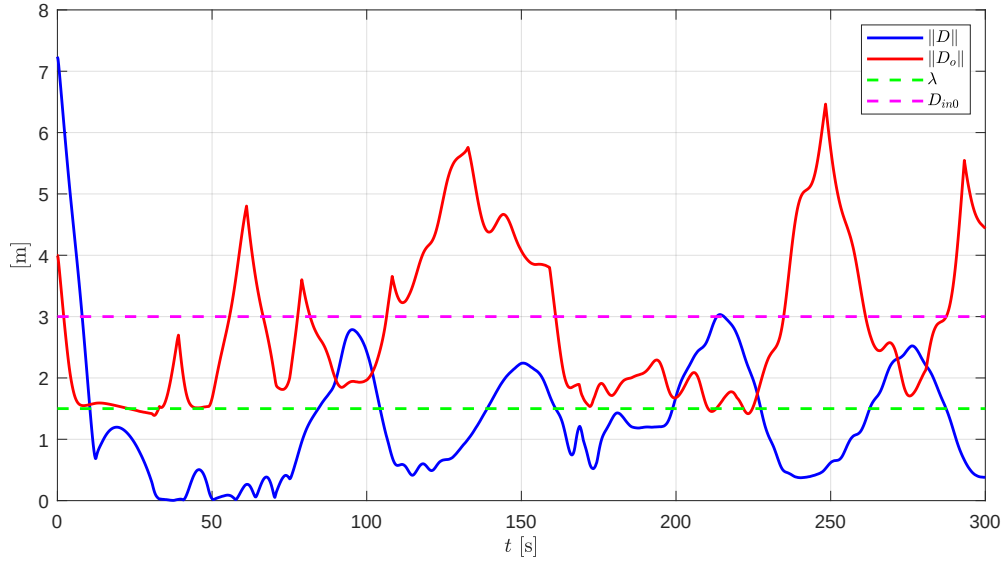


Figure 33 – Position Controller simulation IV: Distances between the system and the curve in blue, between the system and the obstacles in red.

We can see that the system tried to follow the path and did not collide with obstacles. However, when comparing to the previous result in Figure 27, the controller performed worse without the robust adaptive action, where the disturbances were not mitigated and clearly hampered the system's capability to follow the path and avoid obstacles.

Figure 34 shows the control inputs. When comparing to the previous one in Figure 28 we can see that both control inputs had a similar shape and magnitude. However, this one performed worse with greater errors to the path.

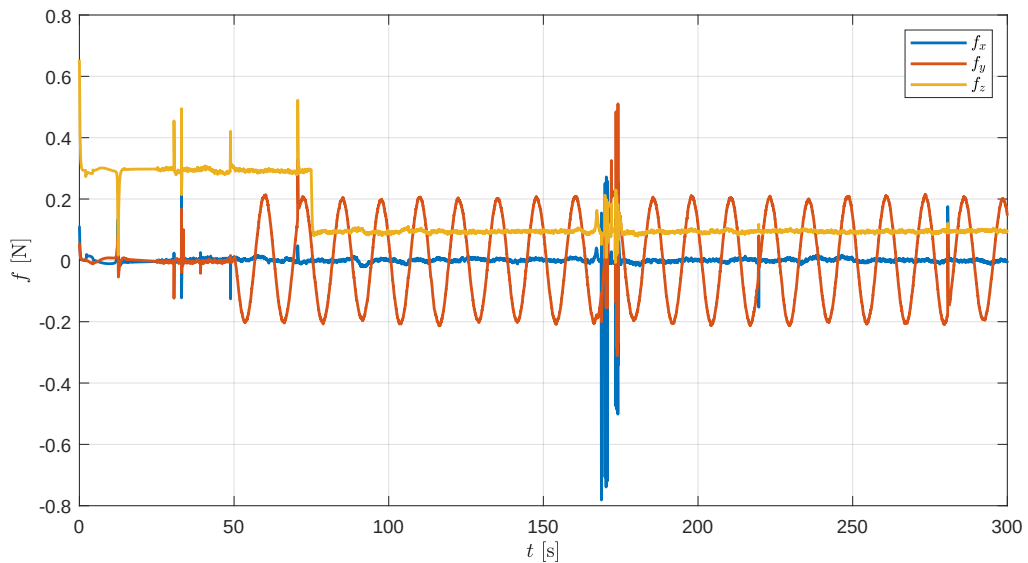


Figure 34 – Position Controller simulation IV: Evolution in time of control inputs.

The better performance when using the robust adaptive action can also be seen by the evolution in time of the system states. Compare the previous simulation system states in Figure 31 (with robust adaptive action) with this simulation system states in Figure 35 (without robust adaptive action) below. The effects of the disturbances are clearly noted affecting the velocity of the system in the simulation without robust action, while in the simulation with robust action, those effects were mitigated.

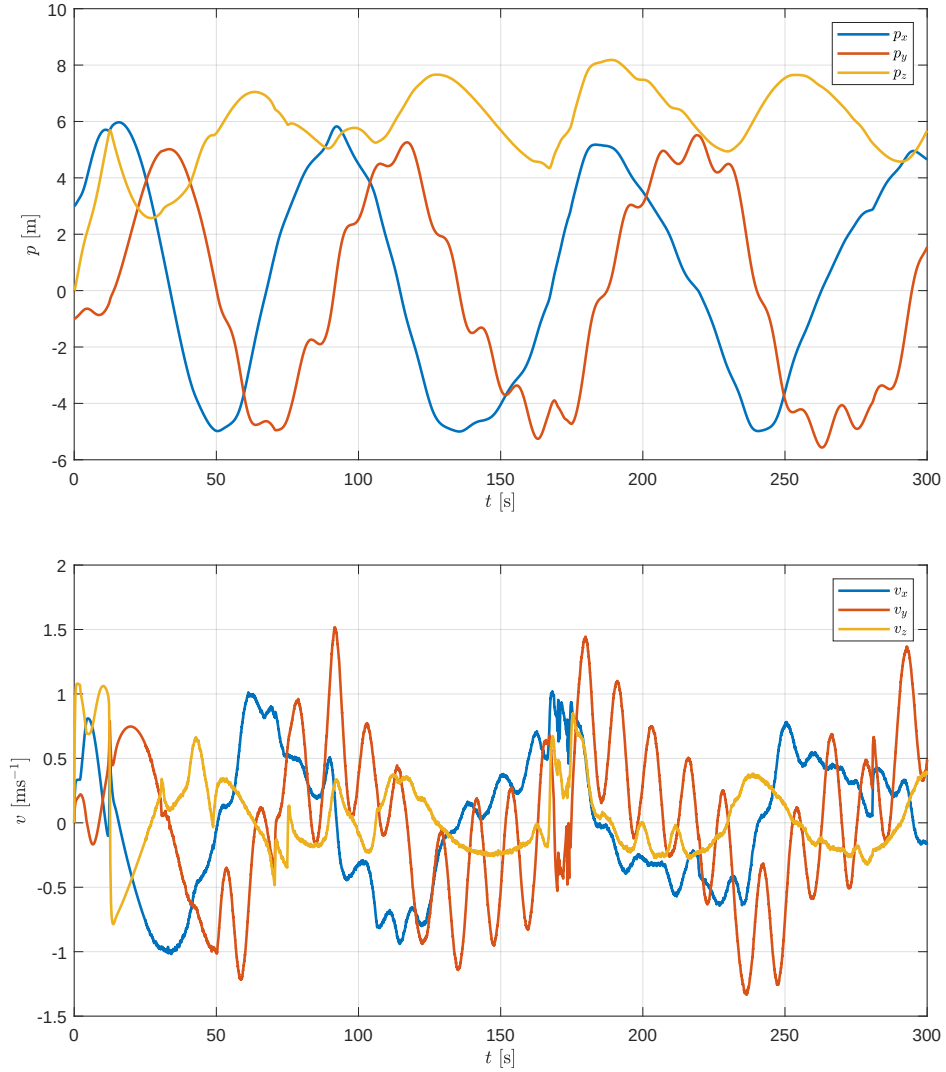


Figure 35 – Position Controller simulation IV: Evolution in time of systems states.

For this simulation, it does not make sense to show the error between practical and nominal systems, since we did not have a nominal system in the control loop. Similarly, it does not make sense to show the adaptive gains, since they were not used anywhere here.

The purpose of this simulation is to be compared with the previous one, highlighting the superiority of the combined nominal and robust adaptive control scheme.

## 6.3 Attitude Controller simulations

### 6.3.1 Attitude Controller simulation I: disturbances

An animation video for this simulation is available at [youtu.be/oJ4p-Hij0AI](https://youtu.be/oJ4p-Hij0AI).

For the first simulation of the Attitude Controller, the system was controlled to follow a reference orientation trajectory consisting in a  $\pi/3$  angle displacement in a rotating axis in  $(\hat{\mathbf{i}}, \hat{\mathbf{j}})$  plane,  $\left(\hat{\mathbf{i}} \cos\left(\frac{t}{5}\right) + \hat{\mathbf{j}} \sin\left(\frac{t}{5}\right)\right)$ , followed by a rotation of  $t/5$  around the body frame  $\hat{\mathbf{k}}$  axis, *i.e.*

$$\mathbf{o}_d(t) = \mathbf{o}_2(t)\mathbf{o}_1(t), \quad (6.28)$$

$$\mathbf{o}_1(t) = \cos\left(\frac{\pi}{6}\right) + \left(\hat{\mathbf{i}} \cos\left(\frac{t}{5}\right) + \hat{\mathbf{j}} \sin\left(\frac{t}{5}\right)\right) \sin\left(\frac{\pi}{6}\right),$$

$$\mathbf{o}_2(t) = \cos\left(\frac{t}{10}\right) + \hat{\mathbf{k}} \sin\left(\frac{t}{10}\right).$$

The system's initial conditions were

$$\mathbf{o}(0) = \bar{\mathbf{o}}(0) = 1, \quad (6.29)$$

$$\boldsymbol{\omega}(0) = \bar{\boldsymbol{\omega}}(0) = 0 \text{ rad s}^{-1}. \quad (6.30)$$

As for the controller, we introduced a +20% parameter estimation error given by

$$\bar{J} = 1.2J, \quad (6.31)$$

with  $J$  from (6.3).

The gains for the nominal controller gains were tuned as

$$k_o = 5 \text{ s}^{-2}, \quad (6.32)$$

$$K_\omega = \begin{bmatrix} 10.3813 & -0.0024 & -0.0098 \\ -0.0024 & 10.5941 & -0.0003 \\ -0.0098 & -0.0003 & 10.5928 \end{bmatrix} \text{ s}^{-1}. \quad (6.33)$$

The robust gains for the disturbance attenuation part were

$$k_{\phi e} = 2.5 \text{ s}^{-1}, \quad (6.34)$$

$$k_{eo} = 6 \cdot 10^{-3} \text{ N m rad}^{-1}, \quad (6.35)$$

$$k_{e\omega} = 6 \cdot 10^{-3} \text{ N m s rad}^{-1}, \quad (6.36)$$

$$\epsilon_o = 2 \cdot 10^{-3} \text{ s N}^{-1} \text{ m}^{-1} \text{ rad}^{-1}, \quad (6.37)$$

$$\rho_3 = 2.5 \cdot 10^{-5} \text{ N m rad}^{-1}, \quad (6.38)$$

$$\rho_4 = 2.5 \cdot 10^{-5} \text{ N m s}^2 \text{ rad}^{-3}. \quad (6.39)$$

The added disturbances are shown in Figure 36, which are given by

$$\boldsymbol{\tau}_d = (\hat{\mathbf{i}}d_x + \hat{\mathbf{j}}d_y + \hat{\mathbf{k}}d_z) \text{ N m}, \quad (6.40)$$

$$d_x = \mathcal{N}(0, 5 \cdot 10^{-5}), \quad (6.41)$$

$$d_y = \mathcal{N}(0, 5 \cdot 10^{-5})\mathbf{1}_{15} + 2.5 \cdot 10^{-3} \sin(t/2)\mathbf{1}_{40}, \quad (6.42)$$

$$d_z = \mathcal{N}(0, 5 \cdot 10^{-5})\mathbf{1}_{65} + 1 \cdot 10^{-3}\mathbf{1}_{65}, \quad (6.43)$$

where  $\mathcal{N}(\mu, \sigma)$  is a Gaussian sample of  $\mu$  mean and  $\sigma$  deviation, and  $\mathbf{1}_{t_0}$  is the unit time step starting at  $t = t_0$ .

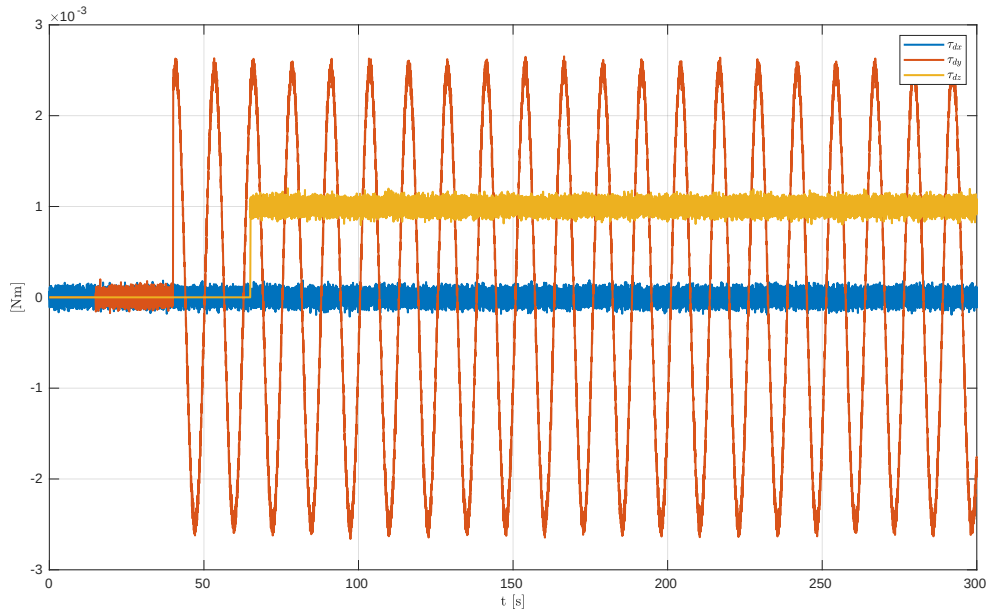


Figure 36 – Attitude Controller simulation I: disturbance  $\boldsymbol{\tau}_d$ .

Figure 37 shows the norm of the attitude error between the **practical system** and the reference trajectory (practical-reference error), expressed as  $\|\text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\|$ , which is different from the quantities  $\mathbf{z}_o$  and  $\mathbf{e}_o$  that express the errors between nominal-reference and practical-nominal attitudes errors, respectively. The system orientation converged to the desired trajectory keeping a small error below  $\delta_o$ , as expected.

Figure 38 shows the control inputs for the nominal system (top), error system (middle), and practical system (bottom). Considering the magnitude of the system parameter and the magnitude of the disturbance, it is observed that the error control law  $\boldsymbol{\tau}_e$  dominates the control input  $\boldsymbol{\tau}$ . It is also observed that the error control law  $\boldsymbol{\tau}_e$  attempts to compensate the disturbance  $\boldsymbol{\tau}_d$  (6.40) previously shown in Figure 36.

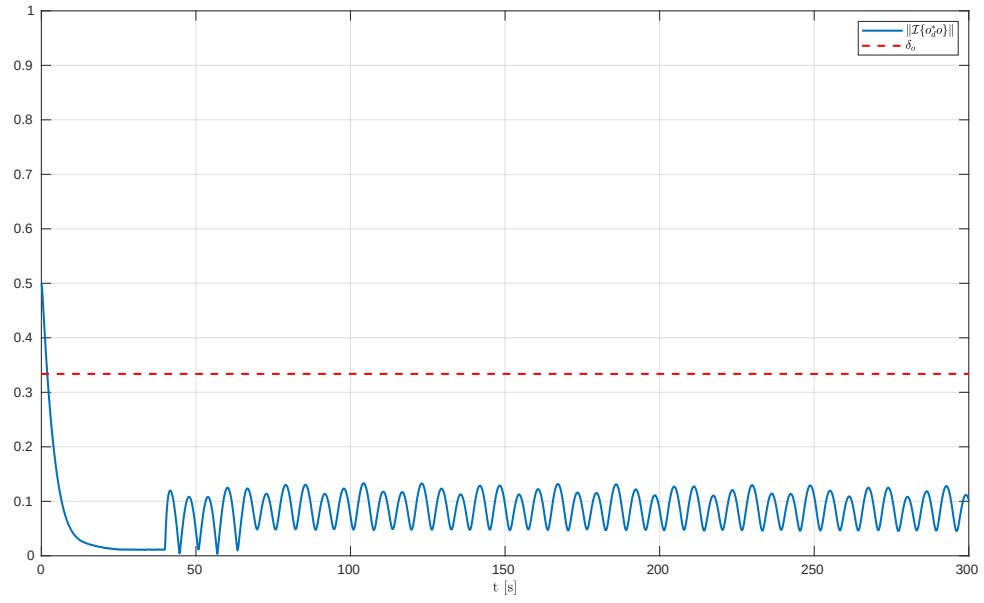


Figure 37 – Attitude Controller simulation I: norm of the attitude error between the **practical system** and the reference trajectory, expressed as  $\|\text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\|$ .

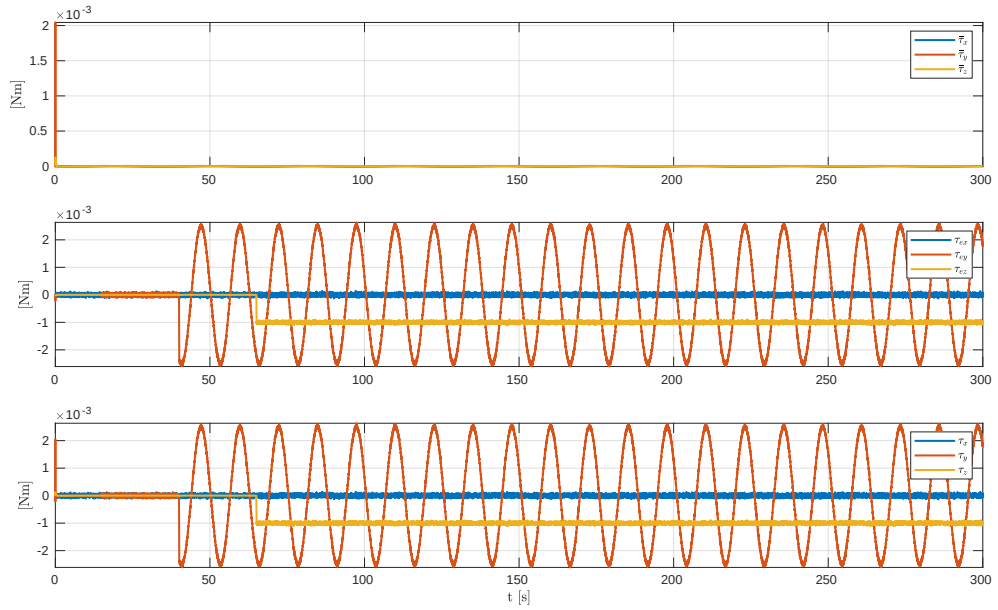


Figure 38 – Attitude Controller simulation I: control inputs for the nominal system (top), error system (middle), and practical system (bottom).

The evolution in time of the norm of the attitude and velocity errors,  $\|\mathbf{e}_o\|$  and  $\|\boldsymbol{\omega}_e\|$ , are displayed in Figure 39, as well as the ultimate bounds  $\delta_o$  and  $\delta_\omega$  respectively. It is observed that the errors stayed limited by the ultimate bounds, as expected.

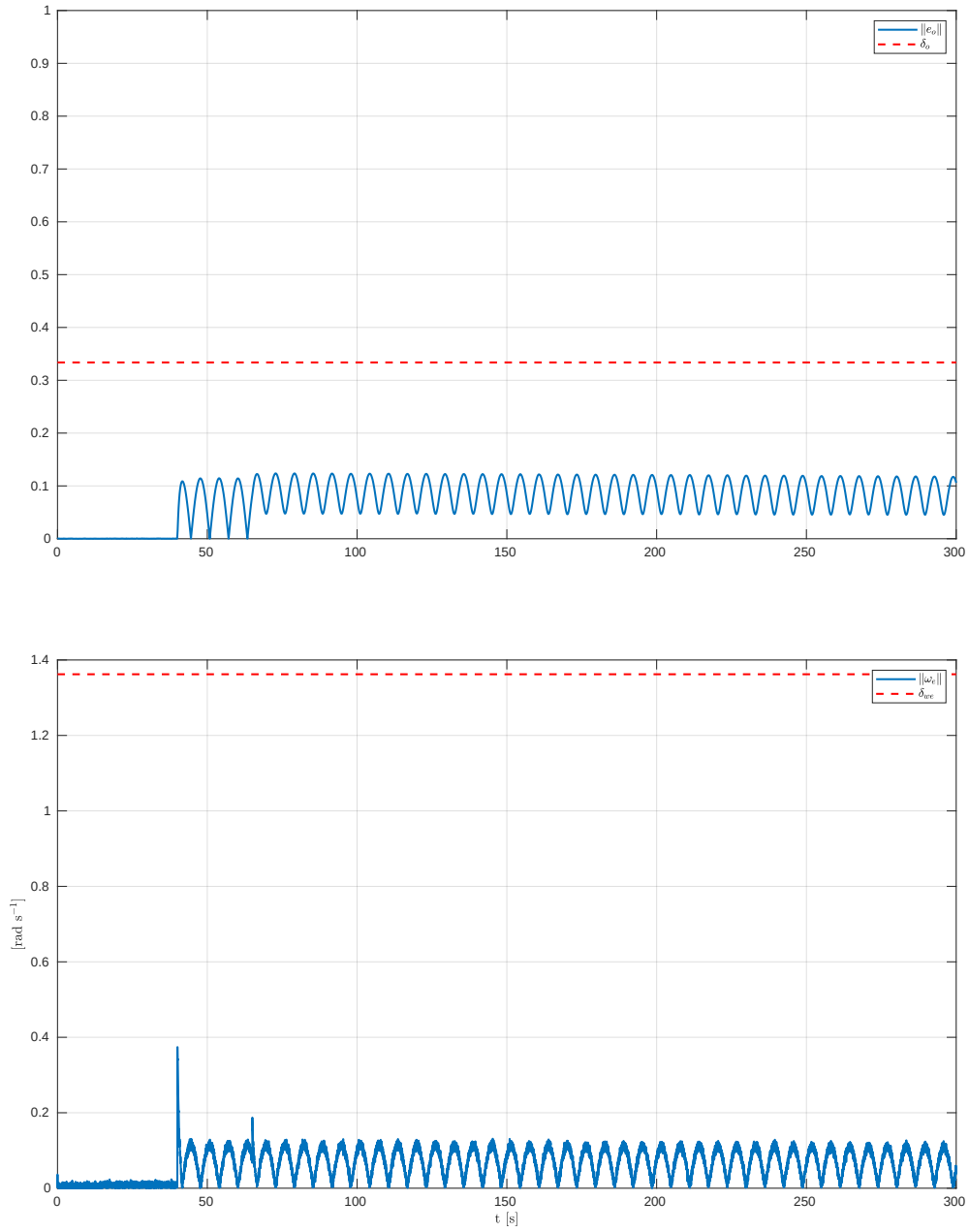


Figure 39 – Attitude Controller simulation I: evolution in time of orientation error (top graph) and velocity error (bottom graph). In red dashed lines are the ultimate bounds  $\delta_o$  and  $\delta_v$  for orientation and velocity, respectively.

Figure 40 shows the evolution in time of adaptive gains. They did not saturate for a 300s simulation, but they increased indefinitely given that the errors observed in Figure 39 never converged to zero.

Figure 41 shows the evolution in time of angular velocity (bottom) and attitude (top) in continuous lines with the reference attitude in dashed lines. Both attitudes were converted to Euler angles for a more readable result.

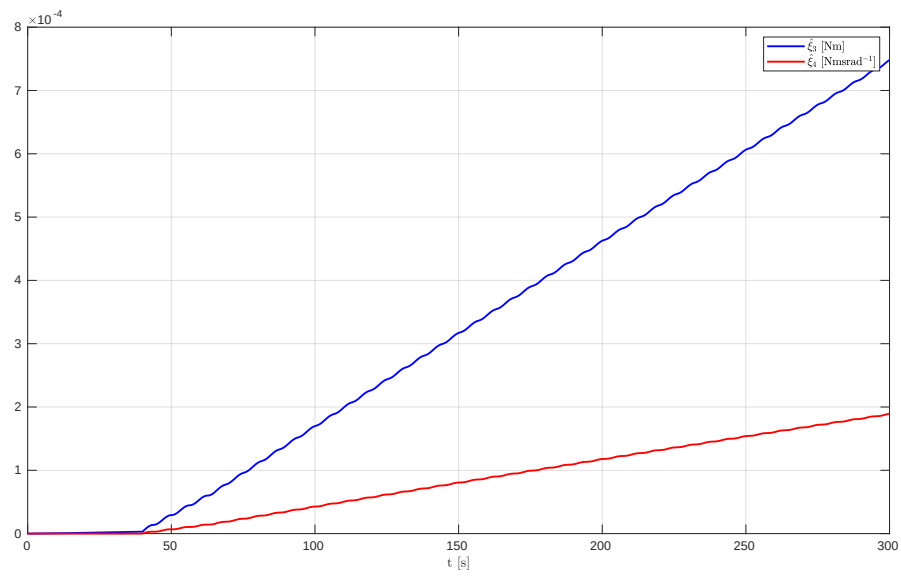


Figure 40 – Attitude Controller simulation I: evolution in time of adaptive gains.

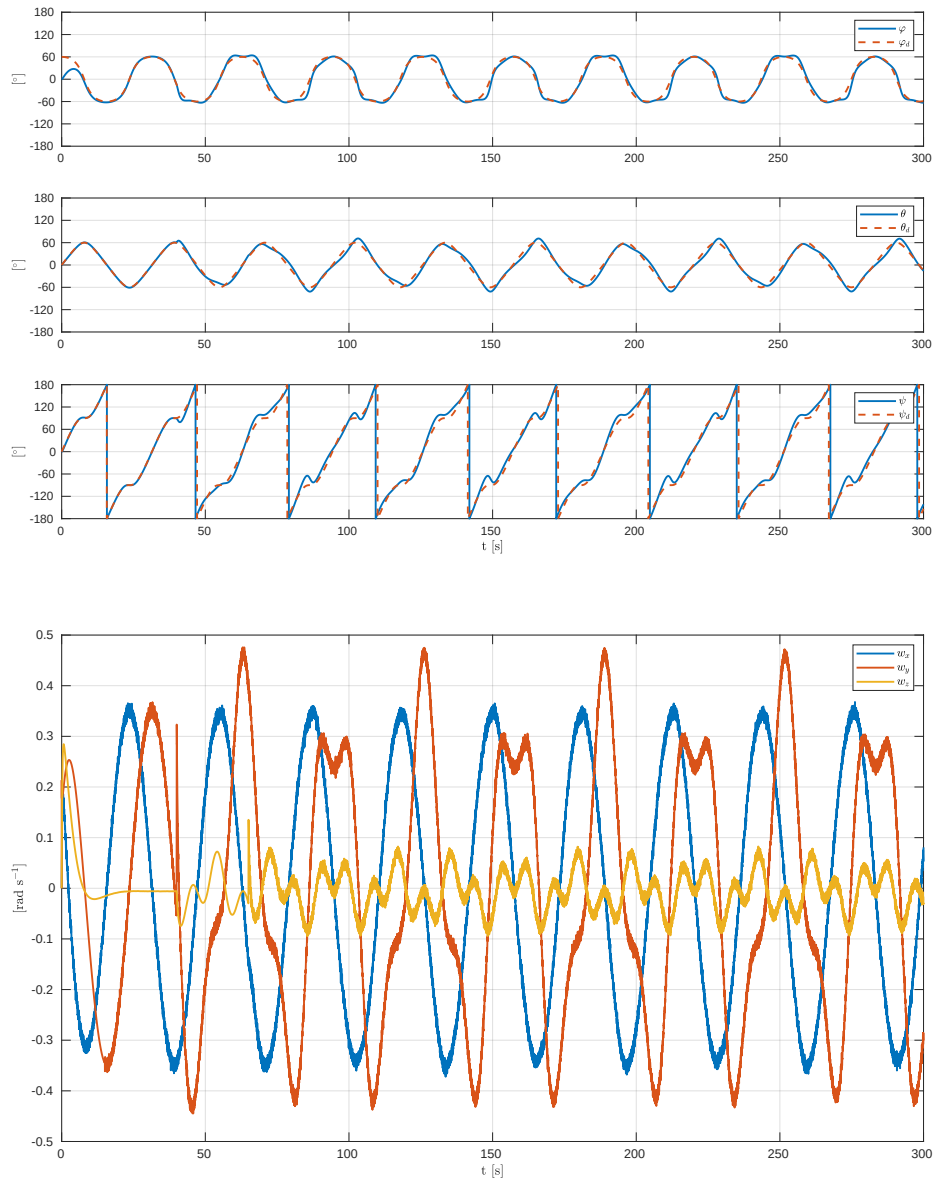


Figure 41 – Attitude Controller simulation I: evolution in time of angular velocity (bottom) and attitude (top) in continuous lines with the reference attitude in dashed lines. Both attitudes were converted to Euler angles for a more readable result.

### 6.3.2 Attitude Controller simulation II: disturbances with nominal controller

An animation video for this simulation is available at [youtu.be/z31OP\\_HVbHI](https://youtu.be/z31OP_HVbHI).

For the second simulation of the Attitude Controller, we removed the robust adaptive control action and repeated the scenario from the previous simulation enabling a comparison between both results. Removing the robust adaptive control action does not mean assigning zero to the robust adaptive gains. In fact, to achieve this we must close the control loop with the practical system and the nominal controller.

To ensure the same scenario as the previous simulation:

- The reference trajectory was given by (6.28);
- The initial conditions were given by (6.29)-(6.30);
- We introduced a parameter estimation error of (6.31);
- The controller gains were given by the equations (6.32) to (6.33). Note that we do not have the robust adaptive gains for this simulation;
- The disturbance generator function was given by (6.40).

Again, the nominal controller is also error-based, so it is expected that it controls the system to track the reference. Figure 42 shows the norm of tracking error between the **practical system** and reference trajectory.

When comparing to the previous result in Figure 37, the controller performed worse without the robust adaptive action, where the disturbances were not mitigated and clearly hampered the system's trajectories.

Figure 43 shows the control inputs. When comparing to the previous one in Figure 38 we can see that both control inputs had a similar shape and magnitude. However, this one performed worse.

The better performance when using the robust adaptive action can also be seen by the evolution in time of the system states. Compare the previous simulation system states in Figure 41 (with robust adaptive action) with this simulation system status in Figure 44 (without robust adaptive action) below. The effects of the disturbances are clearly noted affecting the angular velocity of the simulation without robust action, while on the simulation with robust action, those effects are mitigated.

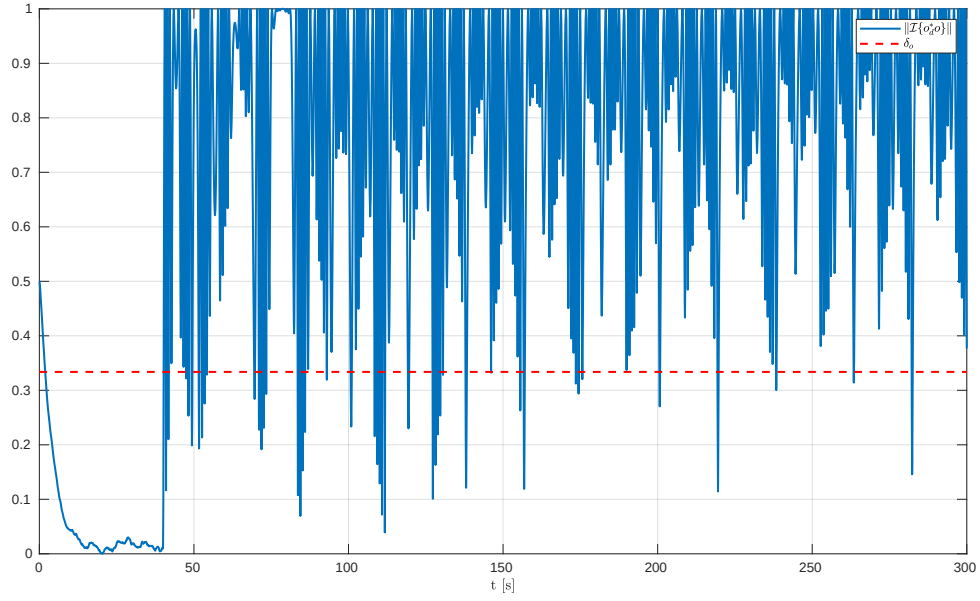


Figure 42 – Attitude Controller simulation II: norm of the attitude error between the **practical system** and the reference trajectory, expressed as  $\|\text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\|$ .

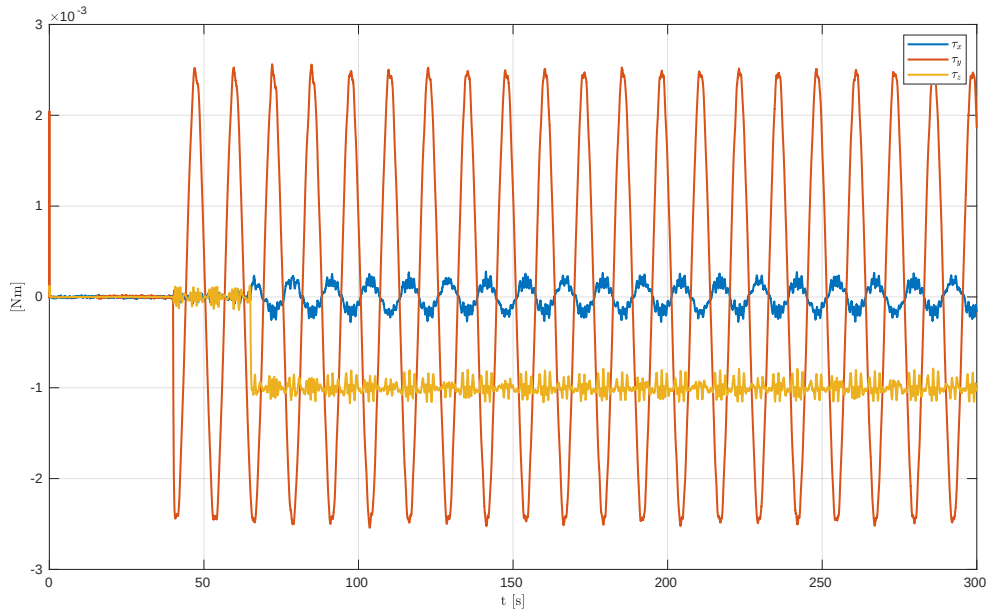


Figure 43 – Attitude Controller simulation II: Evolution in time of control inputs.

For this simulation, it does not make sense to show the error between practical and nominal systems, since we did not have a nominal system in the control loop. Similarly, it does not make sense to show the adaptive gains, since they were not used anywhere here.

The purpose of this simulation is to be compared with the previous, highlighting the superiority of the combined nominal and robust adaptive control scheme.

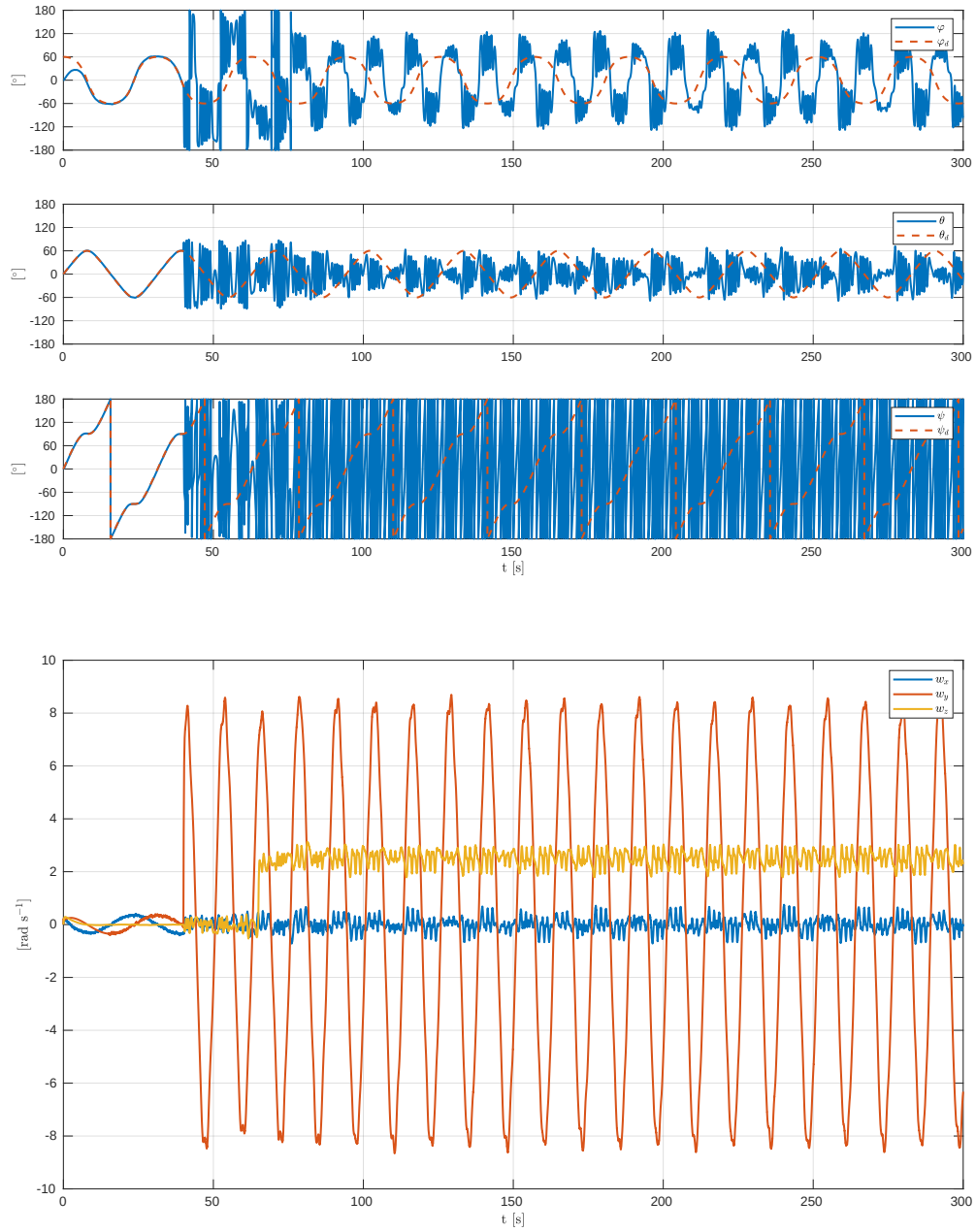


Figure 44 – Attitude Controller simulation II: evolution in time of angular velocity (bottom) and attitude (top) in continuous lines with the reference attitude in dashed lines. Both attitudes were converted to Euler angles for a more readable result.

## 6.4 Attitude experiments

For the three experiments, the Crazyflie was attached to the gimbal ring test bench from Figure 16 to allow it to only orientate itself. Different attitude references were passed to the controller, which successfully controlled the quadcopter system to follow the references, even when it was strongly externally disturbed. Each experiment has a recorded video available.

### 6.4.1 Attitude Controller experiment I: origin regulation

Recording video available at [youtu.be/uJk8t8qGRt](https://youtu.be/uJk8t8qGRt).

For the first one, the orientation reference passed to the Attitude Controller was

$$\mathbf{o}_d = 1, \quad (6.44)$$

meaning an origin regulation problem.

Figure 45 shows the attitude states in continuous lines and the attitude reference in dashed lines. Both were converted to Euler angles. Figure 46 shows the practical-reference attitude error expressed as  $\|\text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\|$ . Analysing both, the system rapidly regulated into the origin where it stayed stable. After approximately two minutes, the system was externally disturbed with high-magnitude disturbances when compared to the magnitude of the inertia tensor and torques produced by the quadcopter, which made the error increase but it rapidly decreased again.

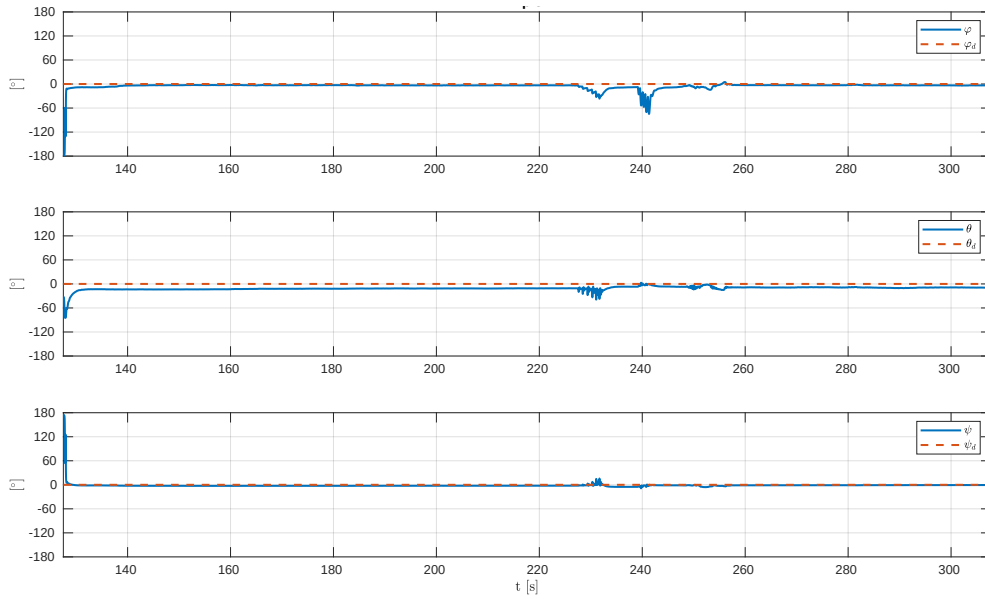


Figure 45 – Attitude Controller experiment I: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles.

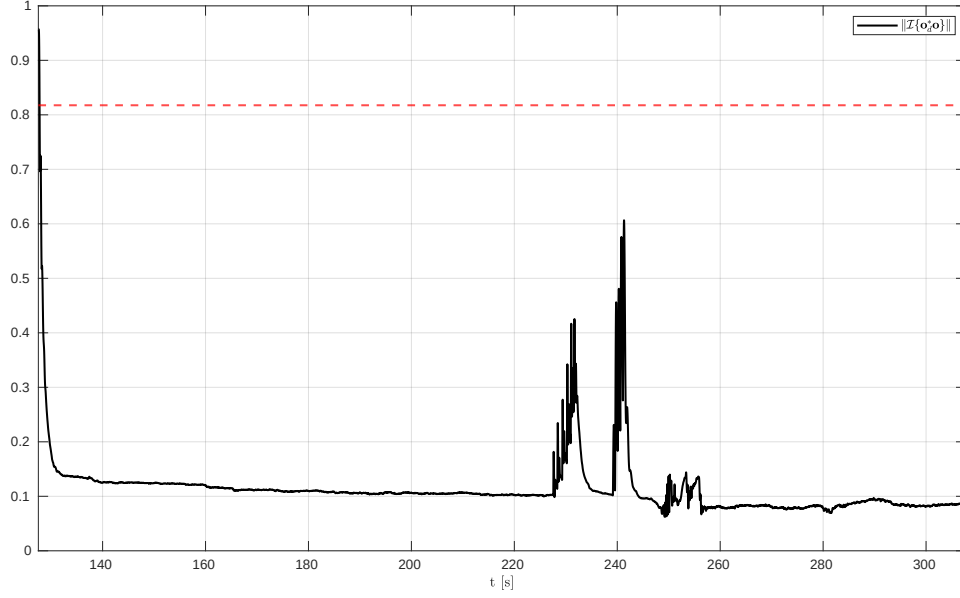


Figure 46 – Attitude Controller experiment I: attitude error expressed as  $\|\text{Im}\{\mathbf{o}_d^* \mathbf{o}\}\|$ .

#### 6.4.2 Attitude Controller experiment II: trajectory-tracking A

Recording video available at [youtu.be/1ggkVky8JjA](https://youtu.be/1ggkVky8JjA).

The next experiment is similar to the previous one, with the difference that now a time-varying reference is passed to the system in the form of

$$\mathbf{o}_d(t) = \cos(t/8) + \hat{\mathbf{i}} \sin(t/8). \quad (6.45)$$

Figure 47 shows the attitude states in continuous lines and the attitude reference in dashed lines. Both were converted to Euler angles. Figure 48 shows the practical-reference attitude error. The system follows the reference. Similar to the previous one, after approximately two minutes, the system was externally disturbed with high-magnitude disturbances when compared to the magnitude of the inertia tensor and torques produced by the quadcopter.

#### 6.4.3 Attitude Controller experiment III: trajectory-tracking B

Recording video available at [youtu.be/qw2jbRyI\\_1M](https://youtu.be/qw2jbRyI_1M).

The last attitude experiment is another time-varying reference, now with a more complex rotation involved. The reference is in the form of

$$\mathbf{o}_d(t) = \cos(\pi/12) + \sin(\pi/12) \left( -\hat{\mathbf{i}} \cos(t/10) + \hat{\mathbf{j}} \sin(t/10) \right). \quad (6.46)$$

Figure 49 shows the attitude states in continuous lines and the attitude reference in dashed lines. Both were converted to Euler angles. Figure 50 shows the practical-reference attitude error. The system followed the reference.

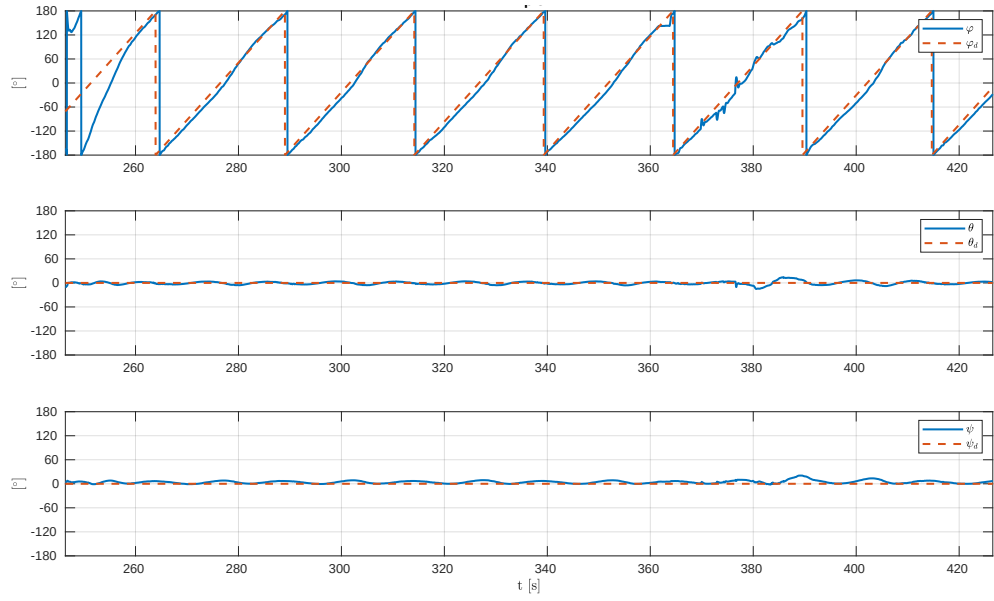


Figure 47 – Attitude Controller experiment II: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles.

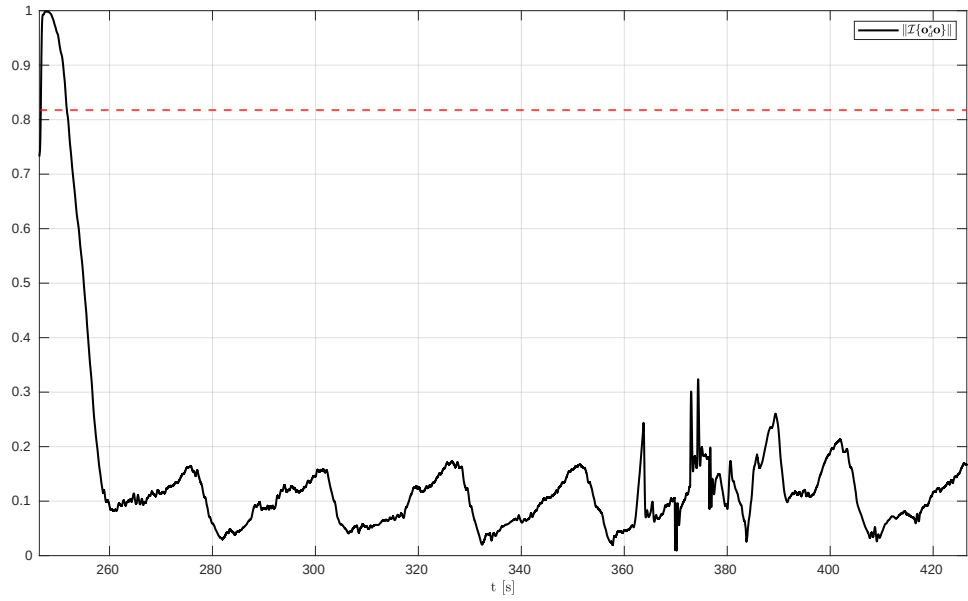


Figure 48 – Attitude Controller experiment II: attitude error expressed as  $\|\text{Im}\{o_d^* o\}\|$ .

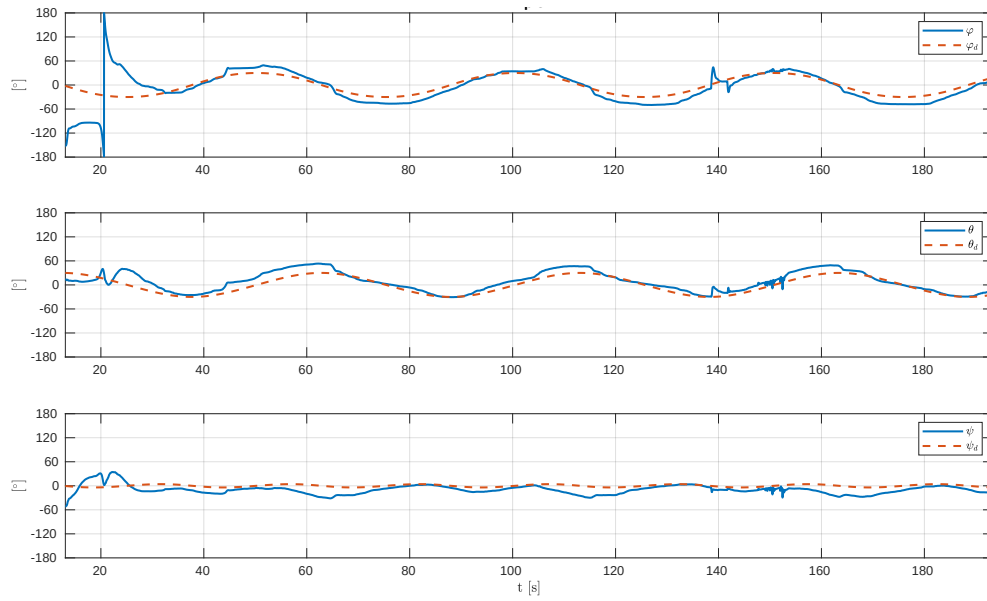


Figure 49 – Attitude Controller experiment III: attitude states in continuous lines and attitude reference in dashed lines. Both are represented as Euler angles.

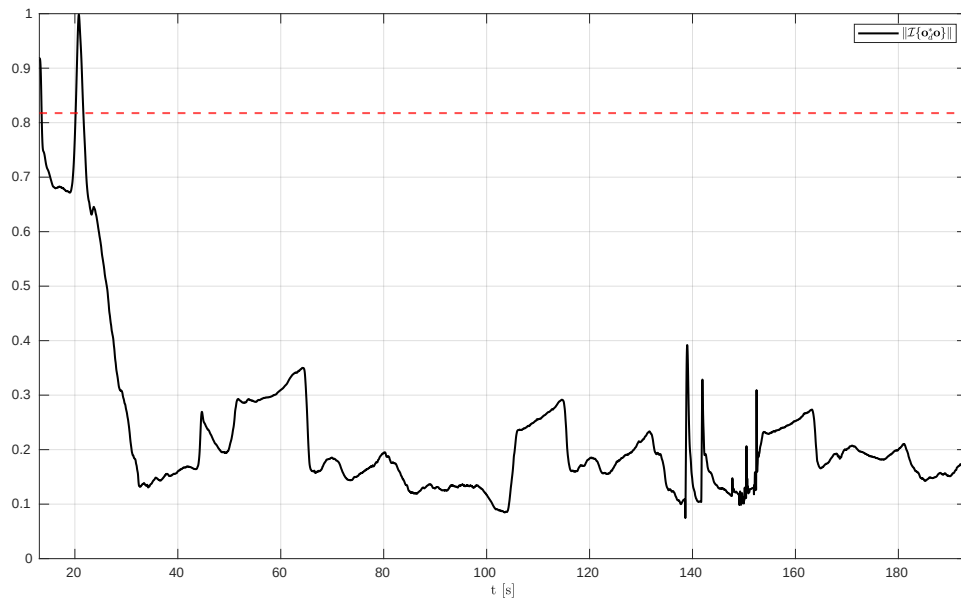


Figure 50 – Attitude Controller experiment III: attitude error expressed as  $\|Im\{o_d^* o\}\|$ .

#### 6.4.4 Attitude Controller experiments summary and comparison

In general, all three experiments accomplished the goal of tracking the reference within predefined error limits. It is important to consider that the system was disturbed with strong disturbances when compared to the magnitude of the motor's torque and system dimensions. Despite the system being heavily disturbed, its attitude stayed within the maximum deviation prescribed during the controller synthesis.

The quadcopter model itself presented some imperfections that can be considered as disturbances. First, the battery cables and mounting produced a force that generated a small torque in the negative  $y$ -axis of the quadcopter, given it a tendency to be slightly rotated on this axis. Furthermore, sometimes during the experiments, some motors presented failures, *i.e.* were trying to rotate but were not able to and therefore just flickered a bit or stayed static.

The ideal 3D Gimbal Ring Test Bench would freely rotate. However, the real prototype disturbed the system to regulate itself. Furthermore, it was observed a higher difficulty for the system to rotate in the world  $z$ -axis when compared to the other axis. This is especially observed in experiment III where it seems like the quadrotor is having difficulties rotating on this axis. This is likely to be due to: (i) the higher demand for control actions to make a quadcopter rotate in its own  $z$ -axis; (ii) mechanical factors such as the poorer bearings in the  $z$ -ring of the 3D Gimbal Ring Test Bench; and (iii) difficulty to estimate the yaw angle, as per what usually happens in estimators that rely on accelerometer and gyroscope data to determine the attitude.

Even in the presence of these imperfections, *i.e.* disturbances, the system performance was as expected, achieving low reference tracking error.

# Chapter 7

## Conclusion and future work

In this work, we developed a controller for quadcopters capable of driving them to follow a given time-varying curve while avoiding obstacles and mitigating unknown disturbances. The constructive design allows separate parts of the controller to be applied to different systems.

The method starts with the vector field guidance path-following controller with obstacle avoidance previously proposed by our group [Rezende et al., 2021, Nunes et al., 2022]. This guidance can be generally applied to any mobile or manipulator robot model. Then, a robust Backstepping Position Controller is developed for second-order integrator systems. Again, this part can be generally applied to any robot model to achieve safe path-following, since it describes the general movement of the double integrator. Separately from the previous controller, a robust rigid-body Attitude Controller for trajectory-tracking is proposed. Different from the Position Controller, the Attitude Controller can be applied to robot models that inherit an attitude part, as many aircrafts and spacecrafts.

The controllers were proven effective under computational simulations for both and real robot experiments for the Attitude Controller. They successfully achieved the goals while bounding the maximum deviation from the nominal trajectory. Our method has some limitations due to: (i) vector field singularities; (ii) the extensive use of derivatives that are inherited from the Backstepping method when compared to other control strategies; and (iii) the adaptive control gains increasing towards infinity.

Our main contribution is extending previous developments by proposing new strategies to control a single quadcopter in path-following tasks with safety (collision avoidance). This is achieved through two new controllers that combine path-following, obstacle avoidance, Backstepping method, and robust adaptive actions. Furthermore, our constructive method allows the controllers to be applied in different scopes as discussed previously.

For future work, we propose five guidelines to continue the research and to continue evolving the methods. Those are summarized below:

- **Bounded control adaptive gains** The first proposal that we aim to evolve our work is to present bounds for the adaptive gains, preventing them to drift towards infinity as it would be always the case in practice. We are considering employing projection modification or similar in the adaptive law [Farrell and Polycarpou, 2006];
- **Bounded control inputs** Similar to the previous proposal, one alternative is also to consider the practical reality of bounded control inputs and the impacts on the system performance;
- **Methods to determine controller gains.** We do not elaborate on how to find the gains. One possible continuity is to propose methods to achieve this;
- **Safety proof for dynamic system.** As discussed in the Position Controller, one contribution of this work is to prove safety, but only for the first-order system. Therefore, we propose as a future work elaborating on the proof to show the safety of the general dynamic system. It might be required to adjust the conditions of the vector field switching or to consider the avoidance region, rather than the avoidance limit  $\lambda$ .
- **Integrate Position and Attitude Controllers.** Our development for quadcopter control is divided into two controllers. One possibility is to integrate both while providing convergence and safety formal proofs;
- **Singularity-free vector field** One possible extension would be to evolve the obstacle avoidance field to be singularity-free and to consider the field combination part;
- **Quadcopter fleet management** Furthermore, we also intend to develop a quadcopter fleet management framework to control multiple quadcopters in cooperation tasks.

# References

- H. Abaunza, P. Castillo, A. Victorino, and R. Lozano. Dual quaternion modeling and control of a quad-rotor aerial manipulator. *Journal of Intelligent & Robotic Systems*, 88(2):267–283, 2017.
- F. F. Afonso Silva, J. José Quiroz-Omaña, and B. Vilhena Adorno. Dynamics of mobile manipulators using dual quaternion algebra. *Journal of Mechanisms and Robotics*, 14(6):061005, 2022.
- A. Aguiar, J. Hespanha, and P. Kokotovic. Path-following for nonminimum phase systems removes performance limitations. *IEEE Transactions on Automatic Control*, 50(2):234–239, 2005. doi: 10.1109/TAC.2004.841924.
- A. P. Aguiar, D. B. Dačić, J. P. Hespanha, and P. Kokotović. Path-following or reference tracking?: An answer relaxing the limits to performance. *IFAC Proceedings Volumes*, 37(8):167–172, 2004.
- Q. Ai, H. Fan, Q. Wang, and S. Li. Spacecraft dynamics modeling and attitude control based on path planning. In *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, pages 303–306, 2021. doi: 10.1109/ICISCAE52414.2021.9590768.
- N. Bao, X. Ran, Z. Wu, Y. Xue, and K. Wang. Research on attitude controller of quadcopter based on cascade PID control algorithm. In *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 1493–1497, 2017. doi: 10.1109/ITNEC.2017.8285044.
- F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999. ISSN 0005-1098. doi: [https://doi.org/10.1016/S0005-1098\(99\)00113-2](https://doi.org/10.1016/S0005-1098(99)00113-2). URL <https://www.sciencedirect.com/science/article/pii/S0005109899001132>.
- S. Bouabdallah and R. Siegwart. Backstepping and sliding-mode techniques applied to an indoor micro quadrotor. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2247–2252, 2005. doi: 10.1109/ROBOT.2005.1570447.

- S. Bouabdallah, A. Noth, and R. Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2451–2456 vol.3, 2004. doi: 10.1109/IROS.2004.1389776.
- W. Cai, X. H. Liao, and Y. D. Song. Indirect robust adaptive fault -tolerant control for attitude tracking of spacecraft. *Journal of Guidance, Control, and Dynamics*, 31(5):1456–1463, Sept. 2008. doi: 10.2514/1.31158. URL <https://doi.org/10.2514/1.31158>.
- H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun. *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke. A prototype of an autonomous controller for a quadrotor UAV. In *2007 European Control Conference (ECC)*, pages 4001–4008, 2007. doi: 10.23919/ECC.2007.7068316.
- M. A. dos Santos, A. Ferramosca, and G. V. Raffo. Set-point tracking MPC with avoidance features. *Automatica*, 159:111390, 2024. ISSN 0005-1098.
- V. Duggal, M. Sukhwani, K. Bipin, G. S. Reddy, and K. M. Krishna. Plantation monitoring and yield estimation using autonomous quadcopter for precision agriculture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5121–5127, 2016. doi: 10.1109/ICRA.2016.7487716.
- S. E. Eid and S. Sham Dol. Design and development of lightweight-high endurance unmanned aerial vehicle for offshore search and rescue operation. In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pages 1–5, 2019. doi: 10.1109/ICASET.2019.8714418.
- J. A. Farrell and M. M. Polycarpou. *Adaptive approximation based control: unifying neural, fuzzy and traditional adaptive approximation approaches*. John Wiley & Sons, 2006.
- V. M. Gonçalves, L. C. Pimenta, C. A. Maia, B. C. Dutra, and G. A. Pereira. Vector fields for robot navigation along time-varying curves in  $n$ -dimensions. *IEEE Transactions on Robotics*, 26(4):647–659, 2010a.
- V. M. Gonçalves, L. C. Pimenta, C. A. Maia, G. A. Pereira, B. C. Dutra, N. Michael, J. Fink, and V. Kumar. Circulation of curves using vector fields: actual robot experiments in 2D and 3D workspaces. In *2010 IEEE International Conference on Robotics and Automation*, pages 1136–1141. IEEE, 2010b.
- M. Heidarian and A. Y. Memon. Attitude control of VTOL-UAVs. In *Proceedings of 2012 UKACC International Conference on Control*, pages 363–368, 2012. doi: 10.1109/CONTROL.2012.6334657.

- M. Huang, B. Xian, C. Diao, K. Yang, and Y. Feng. Adaptive tracking control of underactuated quadrotor unmanned aerial vehicles via backstepping. In *Proceedings of the 2010 American Control Conference*, pages 2076–2081, 2010. doi: 10.1109/ACC.2010.5531424.
- M. Iacono and A. Sgorbissa. Path following and obstacle avoidance for an autonomous UAV using a depth camera. *Robotics and Autonomous Systems*, 106:38–46, Aug. 2018.
- S. Iwata, S. Sahashi, and T. Hasegawa. Verification of visual attitude control for outdoor autonomous mobile robots. In *2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 319–320, 2013. doi: 10.1109/URAI.2013.6677375.
- Y.-B. Jia. Quaternions. *Com S*, 477:577, 2019.
- Y. Jing, S. Xiaoping, L. Long, and L. Hailong. Nonlinear observer based time delay fault-tolerant attitude control for flexible spacecraft during orbit maneuver. In *The 27th Chinese Control and Decision Conference (2015 CCDC)*, pages 2789–2793, 2015. doi: 10.1109/CCDC.2015.7162376.
- H. K. Khalil. Nonlinear systems. *Patience Hall*, 115, 2002.
- J. B. Kuipers. *Quaternions and rotation sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton university press, 1999.
- B. Landry et al. *Planning and control for quadrotor flight through cluttered environments*. PhD thesis, Massachusetts Institute of Technology, 2015.
- D. A. Lawrence, E. W. Frew, and W. J. Pisano. Lyapunov Vector Fields for Autonomous Unmanned Aircraft Flight Control. *Journal of Guidance, Control, and Dynamics*, 31(5): 1220–1229, Sept. 2008.
- T. Lee, M. Leok, and N. H. McClamroch. Geometric tracking control of a quadrotor UAV on  $SE(3)$ . In *49th IEEE Conference on Decision and Control (CDC)*, pages 5420–5425, Dec 2010. doi: 10.1109/CDC.2010.5717652.
- H. Li, G. Wang, D. Cheng, X. Liu, and X. Liu. Attitude control of over-shoulder launched helicopter-borne missile based on control-saturation sliding-mode. In *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1892–1896, 2018. doi: 10.1109/ITOEC.2018.8740392.
- D. Mayne, M. Seron, and S. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, Feb. 2005. doi: 10.1016/j.automatica.2004.08.019. URL <https://doi.org/10.1016/j.automatica.2004.08.019>.

- D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE International Conference on Robotics and Automation*, pages 2520–2525, May 2011. doi: 10.1109/ICRA.2011.5980409.
- V. R. Miranda, A. Rezende, T. L. Rocha, H. Azpúrua, L. C. Pimenta, and G. M. Freitas. Autonomous navigation system for a delivery drone. *Journal of Control, Automation and Electrical Systems*, 33(1):141–155, 2022.
- S.-H. Mok, S. Byeon, H. Bang, and Y. Choi. Attitude dynamics model-based gyroless attitude estimation for agile spacecraft. In *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pages 1429–1434, 2018.
- D. Nelson, D. Barber, T. McLain, and R. Beard. Vector Field Path Following for Miniature Air Vehicles. *IEEE Transactions on Robotics*, 23(3):519–529, June 2007.
- A. H. Nunes, Á. R. Araújo, M. L. Alves, A. M. Rezende, I. B. Silva, and L. C. Pimenta. Controle de robô para serviços de inspeção industrial utilizando campos vetoriais artificiais variantes no tempo. In *Congresso Brasileiro de Automática-CBA*, volume 2, 2020.
- A. H. D. Nunes, A. M. C. Rezende, G. P. Cruz, G. M. Freitas, V. M. Gonçalves, and L. C. A. Pimenta. Vector field for curve tracking with obstacle avoidance. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 2031–2038, 2022. doi: 10.1109/CDC51059.2022.9992435.
- A. H. D. Nunes, G. V. Raffo, and L. C. A. Pimenta. Integrated vector field and backstepping control for quadcopters. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1256–1262, 2023. doi: 10.1109/ICRA48891.2023.10160824.
- J. C. Pereira, V. J. S. Leite, and G. V. Raffo. Nonlinear model predictive control on  $se(3)$  for quadrotor trajectory tracking and obstacle avoidance. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 155–160, 2019. doi: 10.1109/ICAR46387.2019.8981578.
- L. A. Pereira, A. H. Nunes, A. M. Rezende, V. M. Gonçalves, G. V. Raffo, and L. C. Pimenta. Collision-free vector field guidance and MPC for a fixed-wing UAV. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 176–182. IEEE, 2021.
- M. Polycarpou and P. Ioannou. A robust adaptive nonlinear control design. *Automatica*, 32(3):423–427, Mar. 1996. doi: 10.1016/0005-1098(95)00147-6. URL [https://doi.org/10.1016/0005-1098\(95\)00147-6](https://doi.org/10.1016/0005-1098(95)00147-6).
- W. H. Press. *Numerical recipes: The art of scientific computing*. Cambridge university press, 2007.

- M. Rabah, A. Rohan, S. A. S. Mohamed, and S.-H. Kim. Autonomous moving target-tracking for a UAV quadcopter based on fuzzy-pi. *IEEE Access*, 7:38407–38419, 2019. doi: 10.1109/ACCESS.2019.2906345.
- I. A. Raptis, K. P. Valavanis, and W. A. Moreno. A novel nonlinear backstepping controller design for helicopters using the rotation matrix. *IEEE Transactions on Control Systems Technology*, 19(2):465–473, 2011. doi: 10.1109/TCST.2010.2042450.
- A. M. Rezende, V. R. Miranda, H. N. Machado, A. C. Chiella, V. M. Gonçalves, and G. M. Freitas. Autonomous system for a racing quadcopter. In *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 1–6. IEEE, 2019.
- A. M. Rezende, V. M. Gonçalves, A. H. Nunes, and L. C. Pimenta. Robust quadcopter control with artificial vector fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6381–6387. IEEE, 2020.
- A. M. C. Rezende, V. M. Goncalves, and L. C. A. Pimenta. Constructive time-varying vector fields for robot navigation. *IEEE Transactions on Robotics*, 38(2):852–867, 2021. doi: 10.1109/TRO.2021.3093674.
- T. L. Rocha, A. M. Rezende, V. R. Miranda, H. Azpúrua, and G. M. Freitas. Desenvolvimento de um drone autônomo para tarefas de entrega de carga. In *Congresso Brasileiro de Automática-CBA*, volume 2, 2020.
- B. Rubí, R. Pérez, and B. Morcego. A survey of path following control strategies for UAVs focused on quadrotors. *Journal of Intelligent & Robotic Systems*, 98(2):241–265, 2020.
- G. F. Salierno. Whole-body backstepping control of a quadrotor UAV for trajectory tracking. Master’s thesis, Universidade Federal de Minas Gerais, 2018.
- G. F. Salierno and G. V. Raffo. Whole-body backstepping control with integral action of a quadrotor UAV. In *Proceedings of the XIII Brazilian Symposium on Intelligent Automation (SBAI 2017), Porto Alegre, Brazil*, pages 2157–2164, 2017.
- R. Skjetne and T. I. Fossen. On integral control in backstepping: Analysis of different techniques. In *Proceedings of the 2004 American control conference*, volume 2, pages 1899–1904. IEEE, 2004.
- M. W. Spong, S. Hutchinson, M. Vidyasagar, et al. *Robot modeling and control*. Wiley, 2005.
- L. Sun and W. Huo. Robust adaptive attitude tracking for satellites with constrained control inputs. In *Proceedings of the 32nd Chinese Control Conference*, pages 2885–2890, 2013.

- C. K. Tan, J. Wang, Y. C. Paw, and T. Y. Ng. Tracking of a moving ground target by a quadrotor using a backstepping approach based on a full state cascaded dynamics. *Applied Soft Computing*, 47:47–62, 2016. ISSN 1568-4946.
- B. Vilhena Adorno. Robot Kinematic Modeling and Control Based on Dual Quaternion Algebra — Part I: Fundamentals. working paper or preprint, Feb. 2017. URL <https://hal.science/hal-01478225>.
- J. P. Wilhelm and G. Clem. Vector Field UAV Guidance for Path Following and Obstacle Avoidance with Minimal Deviation. *Journal of Guidance, Control, and Dynamics*, 42(8):1848–1856, Aug. 2019.
- A. Williams and O. Yakimenko. Persistent mobile aerial surveillance platform using intelligent battery health management and drone swapping. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, pages 237–246, 2018. doi: 10.1109/ICCAR.2018.8384677.
- B. Yang, F. He, and Y. Yao. MPC-based design of on-off control law of the attitude control thruster. In *2008 27th Chinese Control Conference*, pages 539–543, 2008. doi: 10.1109/CHICC.2008.4605311.
- W. Yao, B. Lin, and M. Cao. Integrated Path Following and Collision Avoidance Using a Composite Vector Field. In *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, Dec. 2019.
- W. Yao, H. G. de Marina, B. Lin, and M. Cao. Singularity-Free Guiding Vector Field for Robot Navigation. *IEEE Transactions on Robotics*, 37(4):1206–1221, Aug. 2021.
- W. Yao, B. Lin, B. D. O. Anderson, and M. Cao. Guiding vector fields for following occluded paths. *IEEE Transactions on Automatic Control*, 67(8):4091–4106, 2022. doi: 10.1109/TAC.2022.3179215.
- L. Zhang, H. Wang, Y. Zhu, and J. Yang. Tube-based attitude control of rigid-bodies with magnitude-bounded disturbances. *Automatica*, 133:109845, Nov. 2021. doi: 10.1016/j.automatica.2021.109845. URL <https://doi.org/10.1016/j.automatica.2021.109845>.
- S. Zhao, X. Wang, Z. Lin, D. Zhang, and L. Shen. Integrating vector field approach and input-to-state stability curved path following for unmanned aerial vehicles. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–8, 2018. ISSN 2168-2216. doi: 10.1109/TSMC.2018.2839840.
- D. Zhou and M. Schwager. Vector field following for quadrotors using differential flatness. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6567–6572, May 2014. doi: 10.1109/ICRA.2014.6907828.

- Z. Zhu, Y. Xia, and M. Fu. Adaptive sliding mode control for attitude stabilization with actuator saturation. *IEEE Transactions on Industrial Electronics*, 58(10):4898–4907, 2011. doi: 10.1109/TIE.2011.2107719.
- N. S. Özbek, M. Önkol, and M. Önder Efe. Feedback control strategies for quadrotor-type aerial robots: a survey. *Transactions of the Institute of Measurement and Control*, 38(5):529–554, 2016. doi: 10.1177/0142331215608427. URL <https://doi.org/10.1177/0142331215608427>.