Politecnico di Milano

School of Industrial and Information Engineering

Master of Science in
Automation and Control Engineering

Nonlinear control of a tilt-arm quadrotor UAV

Advisor: Prof. Marco LOVERA
Co-Advisors: Eng. Mattia GIURATO
Eng. Davide INVERNIZZI

Thesis by:
Paolo GATTAZZO Matr. 841497

Academic Year 2016–2017

alla mia famiglia

Ringraziamenti

La prima persona che tengo a ringraziare per l'enorme opportunitá datami di svolgere questa tesi é il Professor Lovera Marco, non soltanto per avermi permesso di lavorare ad una tematica di mio grande interesse, ma soprattutto per la disponibilitá e la grande competenza dimostrate.
Un sentito ringraziamento per l'assistenza, i consigli e la pazienza portata va a Mattia Giurato e Davide Invernizzi, che mi hanno seguito e aiutato durante tutto il percorso di tesi. Ringrazio anche tutto il gruppo di dottorandi Pietro, Aureliano, Simone e Matteo, per il supporto e la compagnia durante questi mesi.
Voglio inoltre ringraziare Davide, Fabrizio, Dario, Jacopo, Daniele, Emanuele, Marco, Andrea, Luca, Mattia, Riccardo, amici e compagni di studi e di fatiche nei quasi sei anni che mi hanno accompagnato a questo importante traguardo.
Ringrazio gli amici con cui ho convissuto in questi anni, Mussie, Federico, Davide, Lorenzo, Davide, Matteo, Luca, grazie per aver arricchito questa fase della mia vita con la vostra presenza e aver reso felice la trasferta di un veneto fuori sede.
Un grazie lo voglio rivolgere ai compagni del liceo e agli amici di una vita, Christopher, Sebastiano, Matteo, Andrea, Marco, Alberto, Filippo, Federica, Laura, che mi hanno sempre sostenuto e spronato a dare il meglio.
Un grande ringraziamento va alla persona che mi ha sostenuto, aiutato e sopportato nei momenti felici e in quelli difficili, grazie Nikla per il tuo prezioso supporto.
Infine il ringraziamento piú sentito va a tutta la mia famiglia, zii, cugini e soprattutto ai miei genitori Arcangelo e Daniela e a mio fratello Alberto per il sostegno continuo e per avermi messo nelle condizioni migliori per portare a termine questo lungo e prezioso lavoro e percorso di studi. Semplicemente e profondamente grazie.

Abstract

Nowadays, the Unmanned Aerial Vehicles (UAVs) are more and more popular thanks to the broad class of applications in which they can be employed. When referring to an UAV, generally called drone, usually one can refer to a category of multi-rotor Vertical Take-Off and Landing (VTOL) vehicles provided with four, six, or eight motors, of small/medium size and remote controlled. Everyday these aircrafts are used in new fields of applications, from the entertainment to the professional purposes, up to the military missions.
The increasing interest in the UAVs and, in particular, in their capabilities, is pushing the commercial and research communities towards new challenges. The development of new configurations for multi-rotor UAVs is essential to improve the maneuverability and the operational range of the well-established co-planar platforms, in which the thrust can be produced along a single direction.
The work conducted within this thesis is focused on a particular class of UAVs: the quad-copters with tilt-arm capabilities, called tilt-rotors. This kind of quadrotors, unlike a standard quad-copter, have an over-actuated structure that allows ideally to independently control the six Degrees Of Freedom (DOFs) of a rigid-body in space. This is possible thanks to the use of eight actuators: four motors at which the propellers are fixed and four servo-actuators that can change the thrust direction of four propellers by tilting the four arms around their longitudinal axes. This makes the platform capable to perform complex maneuvers, impossible for a fixed-arm quadrotor and potentially useful in some special operations.
Starting from these premises, the purpose of this thesis is to analyze, to implement and to simulate nonlinear control strategies for an existing tilt-rotor prototype, designed and realized during a previous thesis developed at Aerospace Science and Technologies Department in Politecnico di Milano.
Initially, such prototype under exam is described, its mathematical model is presented and all the formalisms adopted in the following chapters are defined. A simplified nonlinear model is proposed to address the trajectory tracking control problem. Two control strategies are explored: feedback linearization and Lyapunov-based geometric control.
The first family of controllers taken into account is the feedback linearization controllers, of which two different control laws and their respective control parameters tuning are presented. Moreover, two control laws are presented in the

framework of geometric control theory, which is particularly suitable to tackle the control problem for the tilt-rotor UAV. In particular, the first geometric control law solves the control problem in the case that full-actuation is assumed. It constitutes a base case from which the second control law is derived. Restricting the maximum tilt-angle of the servo-actuators, the first control law is properly modified in order to guarantee position over orientation tracking and be compatible with an approximated form of the constraints. Numerical simulations are performed to assess the performance of the different control laws.
Finally, a robustness analysis for the implemented controllers is presented in order to evaluate the possibility of implementing them on the real prototype.

Sommario

Al giorno d'oggi, gli aeromobili a pilotaggio remoto (APR) sono sempre piú diffusi grazie alle notevoli applicazioni in cui possono essere impiegati. Quando ci si riferisce ad un APR, comunemente chiamato drone, si fa riferimento ad una categoria di velivoli multi-rotore a decollo ed atterraggio verticale generalmente dotati di quattro, sei oppure otto rotori, di piccole/medie dimensioni e pilotati da remoto. L'utilizzo di questa categoria di velivoli trova di giorno in giorno nuovi campi di applicazione che spaziano dall'intrattenimento a scopi professionali, fino a missioni di tipo militare.
Il crescente interesse per gli APR ed in particolare per le loro capacitá di missione, guida la ricerca scientifica ed ingegneristica verso nuove sfide ed orizzonti. Lo sviluppo di nuove piattaforme multi-rotore a pilotaggio remoto é essenziale per migliorare la manovrabilitá e le capacitá di impiego degli ormai affermati droni quadri-rotore, in grado di generare spinta solamente lungo il loro asse verticale.
Questo lavoro di tesi si concentra su una particolare tipologia di APR: i quadricotteri dotati di braccia inclinabili, detti tilt-rotor. Questa tipologia di droni quadri-rotore, al contrario di un quadricottero standard, possiede una struttura sovra-attuata che gli permette di controllare completamente i sei gradi di libertá di un corpo rigido nello spazio. Questo é possibile grazie all'utilizzo di otto attuatori: quattro motori a cui sono fissate le eliche e quattro servo-motori in grado di modificare la direzione della spinta prodotta grazie all'inclinazione delle quattro braccia, che rendono il tilt-rotor un drone in grado di eseguire manovre complesse potenzialmente utili in alcune operazioni speciali.
Da queste premesse, il presente lavoro di tesi propone l'analisi, l'implementazione e la simulazione di strategie di controllo non lineari per un prototipo esistente di tilt-rotor, progettato e realizzato in un precedente lavoro di tesi presso il Dipartimento di Scienze e Tecnologie Aerospaziali del Politecnico di Milano.
Inizialmente viene descritto il prototipo in questione, é presentato il suo modello matematico e vengono definiti tutti i formalismi utilizzati nei capitoli successivi. Viene proposto, inoltre, un modello non lineare semplificato del tilt-rotor, utilizzato per risolvere il problema di inseguimento di traiettoria. Vengono analizzate due strategie di controllo: il controllo in feedback linearization e il controllo geometrico basato sulla teoria di Lyapunov. La prima famiglia di controllori presa in esame é quella dei controllori in feedback linearization, di cui sono presentate

due differenti leggi di controllo e i relativi metodi di taratura. Successivamente, sono presentate due leggi di controllo del mondo dei controllori geometrici, che si presta bene per affrontare il problema di controllo del tilt-rotor. In particolare, le prima legge di controllo geometrico risolve il problema di controllo assumendo che il tilt-rotor sia in grado di produrre forze arbitrarie nello spazio senza vincoli. Essa costituisce un caso base dalla quale la seconda legge di controllo geometrico é sviluppata. Limitando gli angoli di inclinazione massimi dei servomotori, la prima legge di controllo viene modificata in modo da rendere prioritario l'inseguimento di posizione rispetto all'inseguimento di assetto ed essere compatibile con una forma approssimata dei vincoli di attuazione. Sono eseguite diverse simulazioni numeriche per verificare le performance delle differenti leggi di controllo.
Infine, é presentata un'analisi di robustezza dei controllori mostrati in modo di poter valutare la possibilitá di implementarli sul prototipo esistente.

Contents

List of Figures

List of Tables

Introduction

An Unmanned Aerial Vehicle (UAV) is an aircraft without a pilot aboard, which is able to fly autonomously or could be driven by remote. Usually called drones, in recent years this type of vehicles has met with great interest both in civil and military fields thanks to their wide range of applications, including precision agriculture, photography, policing and surveillance, search and rescue, entertainment, product delivery, aerial inspection and many others. For example, a Non Governative Organization (NGO) in Malta, the Migrant Aid Offshore, uses drones to locate and rescue migrants in the Mediterranean Sea, some towns in Italy find illegal buildings or measure the quality of the air through those vehicles, that are equipped with appropriate sensors (Botta [2015]).
In this thesis the focus will be on a particular class of UAVs: the multi-rotors, in particular the four-rotors configuration, named quadrotors or quad-copters. The simplest kind of quadrotor is composed by a central body, that comprehends the Flight Control Unit (FCU) with all sensors, the battery pack, the radio receiver and the Electronic Speed Controllers (ESCs) to supply the motors, and four fixed arms at the end of which the motor plus propeller groups are placed. Controlling the thrust generated by each rotor, the drone is able to reach arbitrary positions and yaw configuration in space by varying its roll and pitch angles. The standard quadrotor is in fact intrinsically under-actuated, *i.e.*, it isn't able to follow a trajectory with decoupled attitude and position setpoint.
The tilt-rotor is another kind of quadrotor that has an over-actuated structure, in particular it has tilting rotor capabilities. Thanks to four servo-motors, each arm with motor plus propeller group can be tilted in order to produce not only a vertical force, but also translational forces. This capability lets the tilt-rotor to reach a full position/attitude decoupling: for example, it is able to hover keeping non-null roll/pitch angles. This kind of platform paves the way for more complex maneuvers and more operational scenarios, but also to more sophisticated control strategies able to exploit the eight actuators to fully control the six degrees of freedom of a rigid body in space.
A tilt-rotor prototype has been designed and realized in Micheli [2016]. In that work, linear controllers have been implemented in order to control the platform. Dealing with the analysis, the implementation and the numerical simulation of nonlinear control strategies for such tilt-rotor prototype, this thesis represents a

further progress in solving the control problem. The present thesis begins by describing the tilt-rotor prototype and presenting its mathematical model along with all the formalisms necessary to avoid any ambiguity. The core of this thesis relies on the implementation on Matlab Simulink environment of four different control laws, that span into two control strategy families: feedback linearization control and Lyapunov-based geometric control.
Two feedback linearization control laws are implemented. The first one, presented in Ryll et al. [2015], resorts to the system dynamic extension in order to perform the dynamic inversion, while the second control law works at a lower differential level. Numerical simulations are performed in order to assess the feedback linearization controllers performance and some considerations are exposed. Then, the control problem is addressed in the geometric control framework, implementing two control laws, which has been presented in Invernizzi and Lovera [2017a] and Invernizzi and Lovera [2017b]. Also for the geometric control laws, numerical simulations are performed in order to evaluate the controllers performance and limitations. The last part of the thesis deals with a qualitative robustness analysis of the implemented controllers.
The arguments treated in this thesis are presented in this order:

- Tilt-rotor description: prototype, model and formalisms
- Feedback linearization control and relative simulations
- Geometric control and relative simulations
- Qualitative robustness analysis

Chapter 1

Modeling of the tilt-rotor quad-copter

This first chapter introduces the tilt-rotor quad-copter, object of this thesis. Starting from the description of the existing prototype designed and built in Micheli [2016], a mathematic model is then presented. Initially, all the mathematical formalisms adopted are explained in order to avoid ambiguities, then the dynamic model is obtained studying the tilt-rotor kinematics and equations of motion. Moreover, the Simulink implementation of the model is presented and a simplified model for control purposes is shown. Finally, the computation of the nonlinear mixer matrix is exposed.

1.1 The prototype

1.1.1 Prototype description

The tilt-rotor prototype is shown in Figure 1.1 and is composed by a central body to which four arms are connected. On the central body are mounted the Flight

Figure 1.1: Tilt-rotor prototype

Control Unit (FCU), the servo-motors, the battery and the Electronic Speed Controllers (ESCs) to supply the motors. The four arms are connected to the central body through the servo-motors, thanks to which they can tilt around their longitudinal axis. At the end of each arm the motors with the corresponding propellers are mounted. For the sake of simplicity, from now on the set composed by a motor and its propeller is called "propeller group".
The FCU has the important role to control and stabilize the tilt-rotor and is in turn composed by Rapid Robot Prototyping (R2P) boards. R2P is a framework that allows to implement real time architectures for robotic systems using the composition of different basic modules. Each module provides some specific functions: the modules can communicate with each other through a middleware that integrates a publish/subscribe communication protocol and allow to set up distributed control loops in a flexible way. In order to communicate wireless with the ground station, a Raspberry board is mounted over the FCU and connected through a USB cable to R2P.
Specifically, four R2P modules are used:

- USB module: provides serial communication with a computer or other devices and has an SD slot to expand the memory

- RC module: receives radio messages from the radio-controller and allows the PWM communication with the motors controllers
- IMU module: provides attitude and position estimation
- Proximity module: altitude estimation and position/attitude control.

Unlike Micheli [2016], the control computation has been moved from the IMU module to the proximity module in order to have a better distribution of the computation load.

1.1.2 Actuators

Figures 1.2 and 1.3 show the actuators mounted on the tilt-rotor: respectively the motors and the servo-motors.
The motors are DC Brushless type, model HP2814 from RCTimer High Performance Series. The main parameter that describes their behavior is Kv, that represents the number of revolutions per minute (rpm) for each Volt applied to the motor without any load. In this case $710Kv$ means that, having a 3 cells LiPo battery with 11.1 rated Volts, the maximum achievable speed is 7881 rpm. These

Figure 1.2: The motor RCTimer HP2814

motors require a three-phase modulated power supplier: the four ESCs receive as input a PWM signal from the controller and produce the AC three-phase signals. Thanks to the identification campaigns performed in Micheli [2016], the static relation between throttle (%) and rotational speed Ω is expressed as

$$\Omega = \hat{m}\ Th\% + \hat{q} \tag{1.1}$$

where $\hat{m} = 6.031$ and $\hat{q} = 80.49$.
The dynamic model of the motors is expressed by a first order transfer function

$$G(s) = \frac{\Omega(s)}{Th\%(s)} = \frac{\hat{\mu}}{1 + s\hat{\tau}} \tag{1.2}$$

where $\hat{\mu} = 5.2$ and $\hat{\tau} = 55 \cdot 10^{-3}$. The servo-motor model is the HS-485HB,

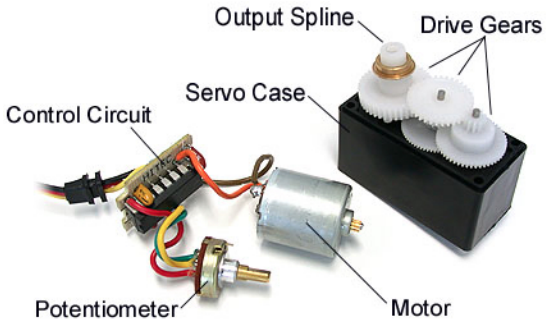

Figure 1.3: Servomotor

provided with an internal control loop that lets them to track an external set-point. In order to characterize their behavior, in Micheli [2016] an identification campaign has been performed and the identified model is given by:

$$F(s) = \frac{\phi(s)}{\phi_{ref}(s)} = \frac{b_0}{b_1 s^3 + b_2 s^2 + b_3 s + b_4} \tag{1.3}$$

where the estimated parameters values are shown in Table 1.1:

Parameter	Value
b_0	4670.2519
b_1	1
b_2	28.355992
b_3	598.45913
b_4	4650.2325

Table 1.1: Servo-motors identified parameters

The actuators bandwidth is around 20 rad/s for both motors and servomotors. However, the servo-motors transfer function is a third order: it causes a significant phase displacement that limits their available bandwidth. In order to complete the actuators characterization and to evaluate the available bandwidth, the Bode diagrams of the actuators are shown in Figure 1.4.

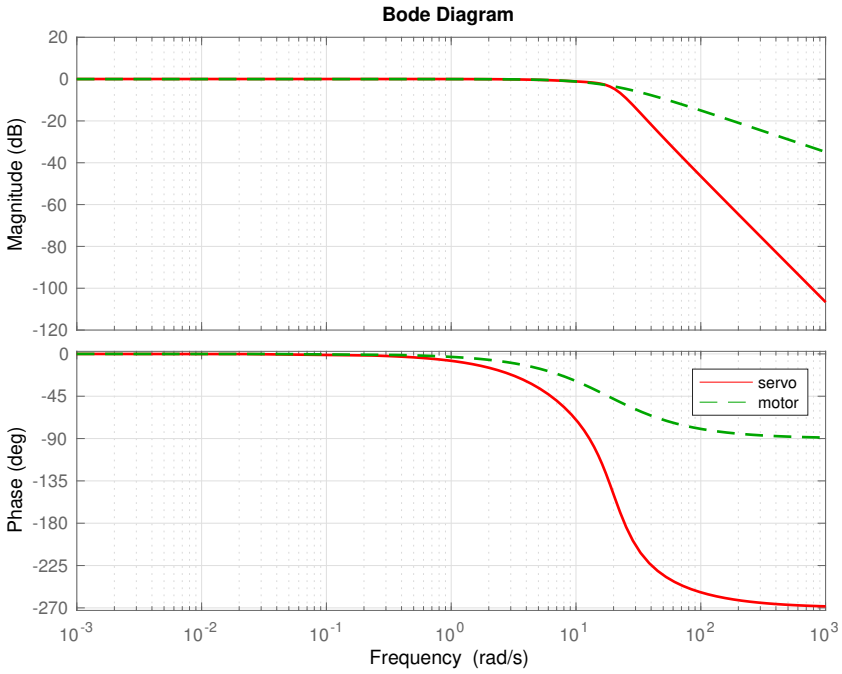

Figure 1.4: Bode diagrams of the actuators

1.2 Formalisms

1.2.1 Reference frames and axes

The motion of a body in space is described thanks to reference systems, that need to be properly chosen. Many conventions are known in the literature, but, since this work is the continuation of Micheli [2016], the chosen reference systems follows in the same way the NED standard.
The Earth fixed frame, called *inertial frame*, is defined as $\mathcal{F}_E = \{O_E, N, E, D\}$, where O_E is a point on the Earth surface. The N-axis and the E-axis are chosen to point respectively North and East and the D-axis completes the right-hand rule pointing downward.
The second frame to be defined is the *body frame* $\mathcal{F}_B = \{O_B, X_B, Y_B, Z_B\}$, where O_B corresponds to the body center of mass. The X-body axis points forward in the direction of the propeller group labelled with number 1, the Y-body axis points right, in direction of the second propeller group. Finally the Z-body axis points downward to satisfy the right-hand rule.
It is necessary also to define other four *propeller frames*, $\mathcal{F}_{P_i} = \{O_{P_i}, X_{P_i}, Y_{P_i}, Z_{P_i}\}$ $(i = 1, 2, 3, 4)$, each attached to a propeller group. In the same way, each X-propeller axis points outward, lying on the same direction of the arm sustaining the propeller group. Each Z-propeller axis points downward and each Y-propeller axis completes the right-hand rule. Figure 1.5 shows the chosen reference frames and axes.

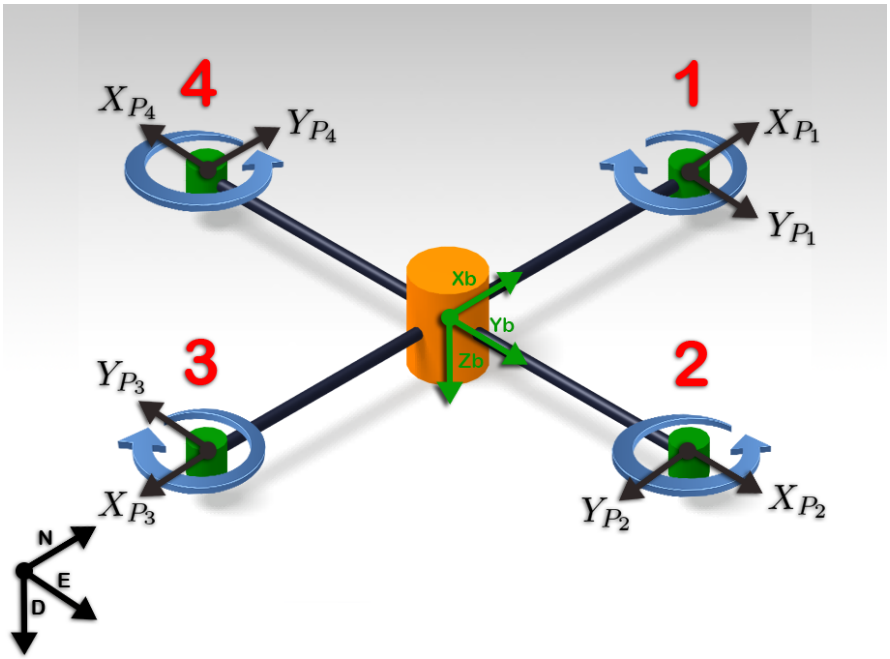

Figure 1.5: Reference frames and axes

1.2.2 Rotation matrices and Euler angles

A rotation matrix is a matrix $\in$ SO(3), the group of orthogonal matrices under matrix multiplication with determinant equal to 1. A rotation matrix could expresses three different meanings:

- the orientation of a frame with respect to another frame
- the transformation that relates the coordinates of a point in two different frames
- the rotation of a vector in a coordinate frame.

A frame or a vector could be rotated around different axes many times: in order to perform this operation, it is useful to define the elementary rotations around a coordinate axis, in particular:

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{1.4}$$

$$R_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \tag{1.5}$$

$$R_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{1.6}$$

The rotation direction is positive counterclockwise, following the right-hand rule. Since a rotation matrices is orthonormal, the inverse rotation is obtained by transposing it. In fact:

$$R_x^{-1} = R_x^T \tag{1.7}$$

$$R_y^{-1} = R_y^T \tag{1.8}$$

$$R_z^{-1} = R_z^T. \tag{1.9}$$

Many consecutive rotations could be performed multiplying the rotation matrices, noting that rotations performed in different order produce different results ($R_x(\alpha)R_y(\beta) \neq R_y(\beta)R_x(\alpha)$). When a vector or a frame is rotated an arbitrary number of times, the final attitude vector could be represented by a minimal representation, that consists in performing just three consecutive rotations around R_z, R_y, R_x.
A minimal representation is a parameterization of the attitude with respect to three parameters, called Euler angles $\Phi = [\phi\ \theta\ \psi]^T$. They represents the three angles of the rotations performed around R_x, R_y, R_z, also called *roll, pitch, yaw*.

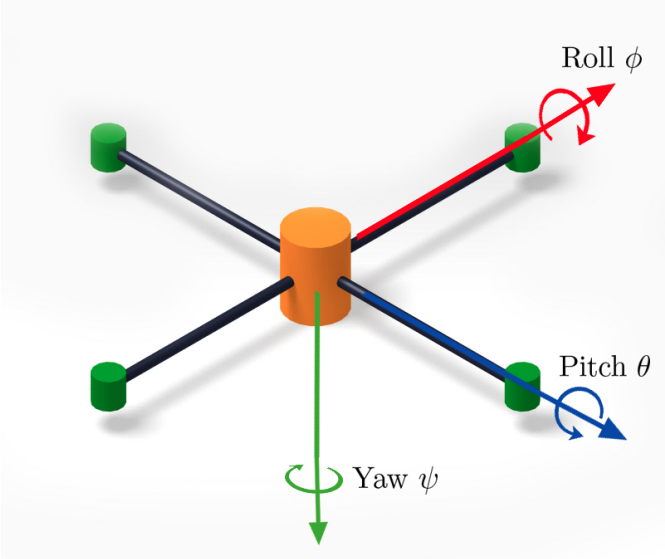

Figure 1.6: Euler angles

The rotation from the world frame to the body frame, *e.g.*, could be expressed in this form

$$^{B}R_{W}(\phi,\theta,\psi) = R_x(\phi)R_y(\theta)R_z(\psi) \tag{1.10}$$

where B and W stands for "body" and "world" respectively and $^{B}R_{W}(\phi,\theta,\psi)$ is the matrix that describes the rotation. For the sake of completeness,

$$^{B}R_{W}(\phi,\theta,\psi) = \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & s_\phi c_\theta \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\phi c_\theta \end{bmatrix}, \tag{1.11}$$

where to simplify the notation, $s_\phi = \sin(\phi)$ and $c_\phi = \cos(\phi)$.
To obtain $^{W}R_{B}(\phi,\theta,\psi)$, the rotation matrix from the body frame to the world frame, it is sufficient to transpose $^{B}R_{W}(\phi,\theta,\psi)$. In fact:

$$^{W}R_{B} = {}^{B}R_{W}^{T}. \tag{1.12}$$

Thanks to rotation matrices and Euler angles, it is possible to express the vectors of kinematic quantities with respect to the body frame or the inertial frame:

$$u_W = {}^{W}R_{B}u_B \tag{1.13}$$

$$u_B = {}^{B}R_{W}u_W \tag{1.14}$$

where u is a generic vector expressed in the inertial frame (u_W) and in the body frame (u_B).

1.2.3 Algebra for matrices

Some useful notions about matrix algebra are reported here.

Skew-symmetric matrix

A matrix $A \in \mathbb{R}^{3\times3}$ is skew symmetric (or anti-symmetric) if it satisfies the condition:

$$A^T = -A. \tag{1.15}$$

The $\hat{}$ operator represents the map transforming a vector $a \in \mathbb{R}^3$ into the associated skew-symmetric matrix $\hat{a} \in \mathfrak{so}(3)$:

$$\hat{a} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{1.16}$$

Cross product

The transformation of a vector in its corresponding skew-symmetric matrix is useful also to compute the cross product. In fact, given two vectors $a, b \in \mathbb{R}^3$:

$$a \times b = \hat{a}\ b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}. \tag{1.17}$$

Matrix derivative

The derivative of a rotation matrix is defined as:

$$\dot{R} = R\ \hat{\omega} \tag{1.18}$$

where $R \in \mathrm{SO}(3)$ is a standard rotation matrix that expresses the transformation from a reference frame to another and ω is the vector of angular rates resolved to the first frame.

Moore-Penrose pseudoinverse

Given a matrix $A \in \mathbb{R}^{n\times m}$, the Moore-Penrose pseudo-inverse of matrix A is defined as a matrix $A^\dagger \in \mathbb{R}^{m\times n}$ that always exists and satisfies the following properties:

- $AA^\dagger A = A$
- $A^\dagger AA^\dagger = A^\dagger$
- $(AA^\dagger)^T = AA^\dagger$
- $(A^\dagger A)^T = A^\dagger A$

If A is full rank, its pseudo-inverse can be computed as:

$$A^{\dagger} = A^{T}(AA^{T})^{-1}. \tag{1.19}$$

If A is not full rank, the pseudo-inverse can be computed numerically using the Singular Value Decomposition (SVD). An important property is that, for linear systems $Ax = b$ with non-unique solution, the pseudo-inverse can be used to find the minimum Euclidean norm solution.

1.3 Mathematical model

A good mathematical model of the tilt-rotor is essential to perform valid simulations. As in Micheli [2016] and Ryll et al. [2015], the tilt-rotor model is derived thanks to a proper kinematic description and then the use of Newton-Euler method by considering:

- forces/moments generated by the propellers
- cross coupling due to gyroscopic and inertial effects
- aerodynamic damping and forces.

In derivating the model, the following assumptions have been considered:

- neglected the arms inertiæ
- simplified thrust and torque generation of rotary propellers

1.3.1 Kinematics

The tilt-rotor model is considered composed by five rigid bodies: the central body and the four propeller groups P_i.

Central body kinematics

The tilt-rotor center of mass is described by:

$$p = \begin{bmatrix} n \\ e \\ d \end{bmatrix} \tag{1.20}$$

$$v = \dot{p} = \begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix} \tag{1.21}$$

$$v_b = {}^{B}R_{W} v = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{1.22}$$

where p and v are the position and velocity vectors of the center of mass O_B expressed in inertial frame, v_b is the linear velocity vector resolved to body axes. The attitude of the tilt-rotor could be expressed in many ways thanks to different parameterizations. One could be the rotation matrix ${}^{W}R_{B}(\phi, \theta, \psi)$, another could be using a quaternion, otherwise it is possible to use the attitude vector whose elements are the Euler angles:

$$\Phi = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}. \tag{1.23}$$

It is possible to define the Euler rates as

$$\omega_e = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{1.24}$$

and the body angular velocity as

$$\omega_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{1.25}$$

The transformation between ω_e and ω_b is sophisticated, in fact it depends not only on the Euler rates, but also on the Euler angles (see Giurato [2015]):

$$\omega_b = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R_x(\phi) \left(\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R_y(\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \right) = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\phi & s_\phi c_\theta \\ 0 & -s_\phi & c_\phi c_\theta \end{bmatrix} \omega_e = G(\phi, \theta)\omega_e. \tag{1.26}$$

The inverse relation cannot be obtained by simply transposing the $G(\phi, \theta)$ matrix (since it is not orthonormal) but it is necessary to invert it:

$$G^{-1}(\phi, \theta) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix}, \tag{1.27}$$

obtaining:

$$\omega_e = G^{-1}(\phi, \theta)\omega_b. \tag{1.28}$$

When the angle θ reaches $\pm\frac{\pi}{2}$, matrix G^{-1} becomes singular: this particular case is called *gimbal lock* and can be avoided using other attitude parameterizations.
Many frames have been defined in order to describe the motion of the tilt-rotor in space: the inertial frame $\mathcal{F}_\mathcal{W}$, the body frame $\mathcal{F}_\mathcal{B}$ and propeller groups frames $\mathcal{F}_{P_i}$ for $i = 1, 2, 3, 4$.
When dealing with rotating frames, in order to have a correct kinematic and dynamic description, it is necessary to use the Coriolis equations (see Giurato [2015]).
Consider p (1.20) the inertial position vector of a body in space that is rotating with angular body rate ω_b (1.25). The velocity vector in the inertial frame is

$$\dot{p} = \begin{bmatrix} \dot{n} \\ \dot{e} \\ \dot{d} \end{bmatrix} = v_b + \omega_b \times p. \tag{1.29}$$

Differentiating again with respect to time and after some expansions, the acceleration is:

$$\ddot{p} = \dot{v}_b + 2\ \omega_b \times v_b + \dot{\omega}_b \times p + \omega_b \times (\omega_b \times p). \tag{1.30}$$

The obtained equation expresses the total linear acceleration acting on a body in space with respect to the inertial fixed frame $\mathcal{F}_{\mathcal{W}}$.

Propeller groups kinematics

The position vectors of the propeller groups with respect to body center of mass are defined:

$$^{B}O_{P_i} = R_z^T\left((i-1)\frac{\pi}{2}\right)\begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix},\ i = 1,2,3,4 \tag{1.31}$$

where b is the arm length, *i.e.*, the distance between the body center of mass and a propeller group.
Each propeller group is able to tilt by an angle α_i around the X_{P_i} axis. In order to express the tilting action, other rotation matrices are adopted:

$$^{P_i}R_B = R_x(\alpha_i)\ R_z\left((i-1)\frac{\pi}{2}\right) \tag{1.32}$$

$$^{B}R_{P_i} = {}^{P_i}R_B^T \tag{1.33}$$

where $^{B}R_{P_i}$ are the matrices that express the rotation of the frames $\mathcal{F}_{P_i}$ with respect to the body frame $\mathcal{F}_{\mathcal{B}}$.
The angular velocities of the propeller groups are given by:

$$\omega_{P_i} = {}^{P_i}R_B\omega_b + \begin{bmatrix} \dot{\alpha}_i \\ 0 \\ \Omega_i \end{bmatrix} \tag{1.34}$$

where $\dot{\alpha}_i$ is the tilting velocity around the X_{P_i} axis and Ω_i is the spinning velocity about Z_{P_i}.
After a time derivation, the angular acceleration is obtained

$$\dot{\omega}_{P_i} = {}^{P_i}R_B\dot{\omega}_b + {}^{P_i}\dot{R}_B\omega_b + \begin{bmatrix} \ddot{\alpha}_i \\ 0 \\ \dot{\Omega}_i \end{bmatrix}, \tag{1.35}$$

where

$$^{P_i}\dot{R}_B = {}^{P_i}R_B\hat{\omega}_\alpha \tag{1.36}$$

$$\omega_\alpha = \begin{bmatrix} \dot{\alpha} \\ 0 \\ 0 \end{bmatrix}. \tag{1.37}$$

Table 1.2 recaps the kinematic quantities of the tilt-rotor.

Symbol	Description	Unit
n	inertial north position of the tilt-rotor	m
e	inertial east position if the tilt-rotor	m
d	altitude of the aircraft	m
u	velocity in body frame around local N	m/s
v	velocity in body frame around local E	m/s
w	velocity in body frame around local D	m/s
ϕ	roll angle	rad
θ	pitch angle	rad
ψ	yaw angle	rad
p	roll rate in body frame	rad/s
q	pitch rate in body frame	rad/s
r	yaw rate in body frame	rad/s

Table 1.2: Kinematic quantities

1.3.2 Equations of motion

Translational motion

The translational motion is referred to the body frame $\mathcal{F}_\mathcal{B}$. Let v_b the velocity vector of the origin of $\mathcal{F}_\mathcal{B}$ in the body frame, applying the Newton's law:

$$m\dot{v}_b + \omega_b \times (mv_b) = F_{prop} + F_{ext} \tag{1.38}$$

$$F_{prop} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{1.39}$$

$$F_{ext} = F_g + F_{aero} \tag{1.40}$$

$$F_g = m\ {}^{B}R_W \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{1.41}$$

where m is the mass of the tilt-rotor, g is the gravity acceleration, F_g is the gravity vector rotated from inertial to body frame and F_{prop} includes the forces generated by the propellers. The vector F_{aero} includes the aerodynamic forces related to the motion of the tilt-rotor in the space. Since the drone is supposed to fly at low speed and indoor, these forces are considered negligible.
The F_{prop} vector is defined as:

$$F_{prop} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \sum_{i=1}^{4} {}^{B}R_{P_i} T_{P_i} \tag{1.42}$$

where T_{P_i} is the thrust vector and is described in the following paragraphs. It

is important to note that unlike a standard quadrotor, the tilt-rotor is able to produce forces not only along the Z_b axis but also along X_b and Y_b axes thanks to its rotors tilting capabilities.
Rewriting equation (1.38) with respect to the inertial frame, it results:

$$m\ddot{p} = m \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + {}^{W}R_B F_{prop}. \tag{1.43}$$

Rotational motion

The rotational equation of motion of the propeller group is obtained applying the Euler's rotation equation:

$$\tau_{P_i} = I_{P_i}\dot{\omega}_{P_i} + \omega_{P_i} \times I_{P_i}\omega_{P_i} - \tau_{prop_i} \tag{1.44}$$

where τ_{P_i} is the vector of total torque acting on the propeller group, I_{P_i} is the inertia matrix of the propeller group and τ_{prop_i} is the counter rotating torque of the i_{th} propeller acting along Z_{P_i} axis.
Applying Newton's second law, the rotational dynamic equilibrium is:

$$I_b\dot{\omega}_b + \omega_b \times I_b\omega_b = \sum_{i=1}^{4}\Big({}^{B}O_{P_i} \times {}^{B}R_{P_i}T_{P_i} - {}^{B}R_{P_i}\tau_{P_i}\Big) \tag{1.45}$$

where I_b is the body inertia tensor and T_{P_i} are the thrust vectors. Due to the symmetry of the tilt-rotor structure the inertia tensor could be considered a diagonal matrix:

$$I_b = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \tag{1.46}$$

The total moments generated by the actuators are given by

$$M_{props} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \sum_{i=1}^{4}\Big({}^{B}O_{P_i} \times {}^{B}R_{P_i}T_{P_i} - {}^{B}R_{P_i}\tau_{P_i}\Big). \tag{1.47}$$

Taking into account also the aerodynamic damping M_{damp}, that can be assumed proportional to the angular body rates ω_b, equation (1.47) becomes:

$$M_{props} + M_{damp} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \sum_{i=1}^{4}\Big({}^{B}O_{P_i} \times {}^{B}R_{P_i}T_{P_i} - {}^{B}R_{P_i}\tau_{P_i}\Big) \tag{1.48}$$

where M_{damp} can be expressed as

$$M_{damp} = \begin{bmatrix} \frac{dL}{dp} & 0 & 0 \\ 0 & \frac{dM}{dq} & 0 \\ 0 & 0 & \frac{dN}{dq} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{1.49}$$

In equation (1.49) the derivatives $\frac{dL}{dp}$, $\frac{dM}{dq}$, $\frac{dN}{dr}$ are called *stability derivatives*. It is possible to derive their analytical form, but since it is not of interest in this thesis, the reader can refer to Micheli [2016].

Forces and moments generation

As explained in Mahony et al. [2012], the effects of a rotary propeller can be approximated as:

- a vertical force along the Z_{P_i} axis
- a counter-rotating torque around the Z_{P_i} axis.

The vector of forces produced by the propeller in the local frame can be approximated as:

$$T_{P_i} = \begin{bmatrix} 0 \\ 0 \\ -K_t \Omega_i^2 \end{bmatrix} \tag{1.50}$$

where K_t is called *thrust coefficient*. It can be experimentally identified and is specific for each kind of propeller. The counter-rotating torque has always a sign that is opposite to the rotation direction of the propeller and can be approximated as:

$$\tau_{prop_i} = \begin{bmatrix} 0 \\ 0 \\ -K_q \Omega_i |\Omega_i| \end{bmatrix}. \tag{1.51}$$

Figure 1.7 shows the propeller frame and the direction of the thrust generated by the propeller according to the tilting action.
Table 1.3 summarizes the physical quantities of the tilt-rotor prototype.

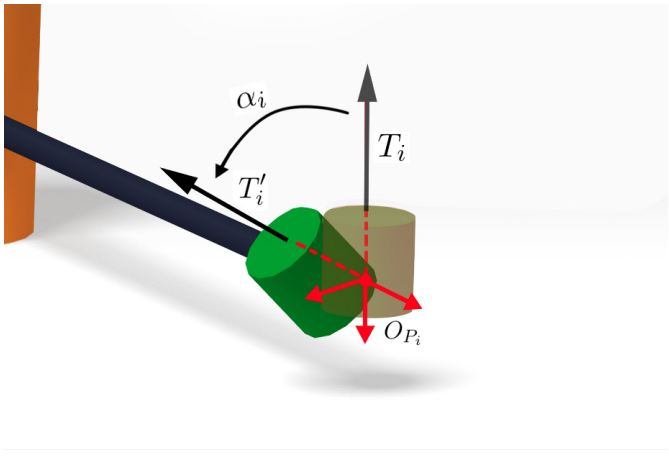

Figure 1.7: Propeller frame and thrust generated according to tilting angle

Description	Symbol	Quantity	Unit
Total mass of tilt-rotor	m	1.9	kg
Arm length	b	0.275	m
Inertia around X_b axis	I_{xx}	0.0074	$kg \cdot m^2$
Inertia around Y_b axis	I_{yy}	0.0074	$kg \cdot m^2$
Inertia around Z_b axis	I_{zz}	0.05	$kg \cdot m^2$
Stability derivative of vehicle roll	dL/dp	-0.046271	$N \cdot m \cdot s$
Stability derivative of vehicle pitch	dM/dq	-0.046271	$N \cdot m \cdot s$
Stability derivative of vehicle yaw	dN/dr	-0.0185	$N \cdot m \cdot s$
Thrust coefficient	K_t	$2.4619 \cdot 10^{-5}$	$kg \cdot m$
Torque coefficient	K_q	$2.8893 \cdot 10^{-7}$	$kg \cdot m^2$

Table 1.3: Physical quantities of the tilt-rotor prototype

1.4 Model implementation in Simulink

Starting from the dynamic model described in the previous sections, the Simulink implementation for simulation purposes of the tilt-rotor model is presented. This model is an update of the one presented in Micheli [2016] and, in particular, it has been modified in order to have a structure similar to the quadrotor model designed in Giurato [2015]: in this way, the implementations of two different multi-copters share the same kernel.

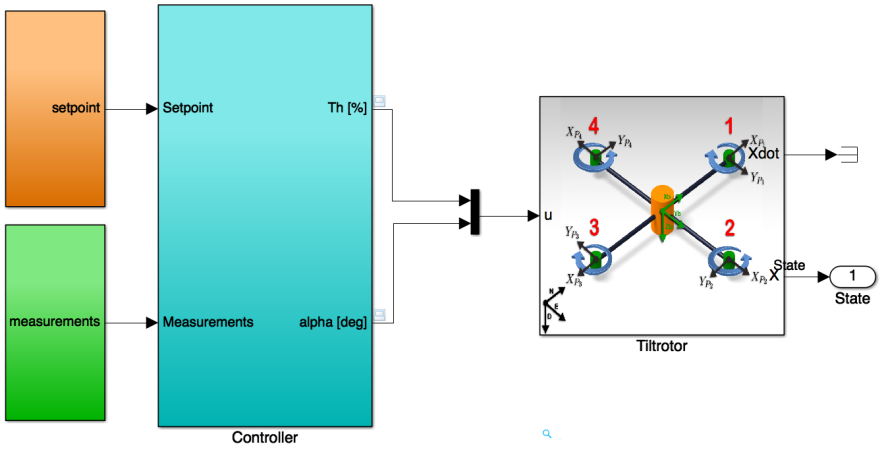

Figure 1.8: Simulink model

Figure 1.8 shows the main blocks that compose the model:

- Tiltrotor: this block contains all the dynamic equations presented in section 1.3. It receives as input the thrusts requested to the four motors in %, $Th\%_i$, and the tilting angles of the four arms α_i for $i = 1, 2, 3, 4$. The outputs of this block are the position, velocity, attitude and angular rates vectors.

- Controller: it is the key block of this thesis. Many nonlinear control strategies have been implemented, all receive as inputs the measurements and produce as outputs the eight control variables $Th\%_i$, α_i for $i = 1, 2, 3, 4$.

- Measurements: this block reads the output of the tilt-rotor block and transforms the signals by discretizing them. This is a way to take into account the sampling time of an hypothetical hardware on which the control system should be implemented. The considered working frequency is $100Hz$.

- Setpoint: this block generates the trajectory that the tilt-rotor is supposed to follow. Due to the complexity of the adopted nonlinear controllers, the set-point specifies not only the required position and the attitude in function

of time, but also velocity, acceleration, jerk, angular rates, angular accelerations and angular jerks are generated. Fortunately, not all the control strategies require acceleration, jerk, angular accelerations and angular jerks.

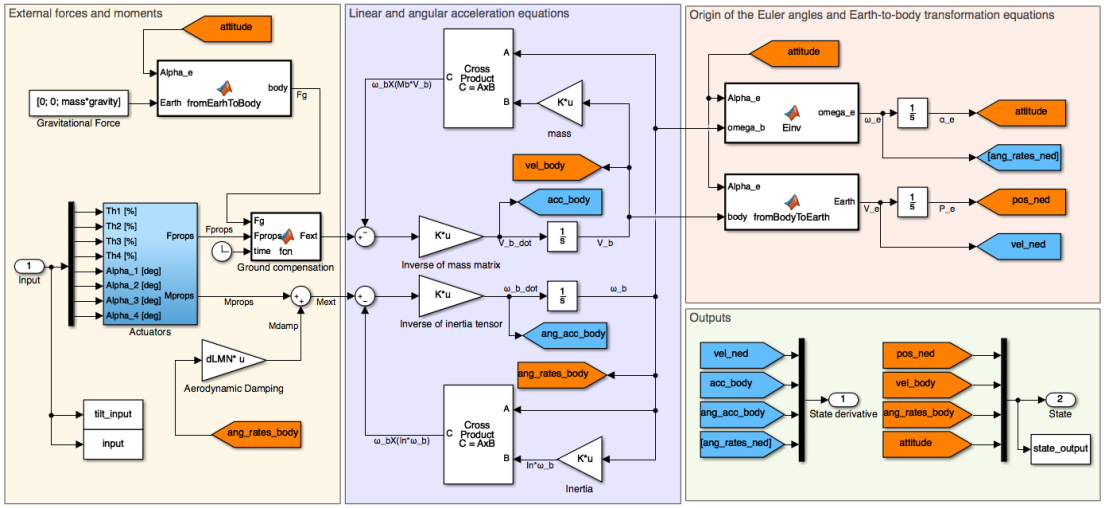

Figure 1.9: Tilt-rotor Simulink dynamic model

Figure 1.9 shows the implementation in Matlab Simulink of the dynamic model described in Section 1.3. The actuators block contains the dynamic equations of motors and servo-motors whose parameters have been identified in Micheli [2016]. The left rectangle contains forces and moments acting on the system in body frame: gravity force, propellers forces, propellers moments and aerodynamic damping. The central section contains the linear and angular equations counting for the inertial and gyroscopic effects. In the right upper rectangle the frames transformations are executed and finally, in the right lower rectangle, the output of the model simulation are produced.

1.5 Model for control

The model described in Section 1.3 is a mathematical approximation of the tilt-rotor prototype. It is useful to perform simulations before executing tests on the real system since it captures the main effects of the tilt-rotor motion in the space. Each model has uncertainties that can be present in different magnitude depending on the assumptions and the simplifications made. For example, our model considers the tilt-rotor composed by only five rigid bodies, in reality they are a lot more and they are of course non ideally rigid. The practice is to create a model making the less simplifications, letting it to be as close as possible to the reality: this way the simulations should have a"realistic" meaning.
The simplification of a model could be useful for control purposes: since in Section 1.3 the model is complex, the design of a controller for the complete model could be hard to be done because of the high number of effects to take into account. In order to make the design of a controller for the tilt-rotor model feasible, some simplifications are assumed. Therefore, a controller is designed specifically for the simplified model: this control should be robust enough to compensate for the model complexity reduction even when it is acting on the complete model. The simplifications made are:

- neglect motors and servomotors dynamics
- neglect aerodynamic damping and forces.

The simplified model for control purposes is:

$$\begin{cases} \ddot{p} & = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} {}^{W}R_B F_{props} \\ \dot{\omega}_b & = I_b^{-1} M_{props} - I_b^{-1}(\omega_b \times I_b\ \omega_b) \\ {}^{W}\dot{R}_B & = {}^{W}R_B \hat{\omega}_b. \end{cases} \tag{1.52}$$

1.6 Mixer matrix

In this section the relation between control variables, *i.e.*, $Th\%_i$ and α_i, and forces/moments acting on the tilt-rotor is treated.
The control law needs as inputs the set-points of position, velocity, attitude and angular rates of the quad-copter and produces as outputs forces and moments requested to actuators. These quantities can be expressed as function of the propellers spinning speeds Ω_i and the tilt angles α_i $(i = 1, 2, 3, 4)$:

$$F_x = -K_t \sin(\alpha_2)\omega_2^2 + K_t \sin(\alpha_4)\omega_4^2 \tag{1.53}$$
$$F_y = K_t \sin(\alpha_1)\omega_1^2 - K_t \sin(\alpha_3)\omega_3^2 \tag{1.54}$$
$$F_z = -K_t \cos(\alpha_1)\omega_1^2 - K_t \cos(\alpha_2)\omega_2^2 - K_t \cos(\alpha_3)\omega_3^2 - K_t \cos(\alpha_4)\omega_4^2 \tag{1.55}$$
$$L = K_q \sin(\alpha_2)\omega_2^2 - K_q \sin(\alpha_4)\omega_4^2 - K_t b \cos(\alpha_2)\omega_2^2 + K_t b \cos(\alpha_4)\omega_4^2 \tag{1.56}$$
$$M = K_q \sin(\alpha_1)\omega_1^2 - K_q \sin(\alpha_3)\omega_3^2 + K_t b \cos(\alpha_1)\omega_1^2 - K_t b \cos(\alpha_3)\omega_3^2 \tag{1.57}$$
$$\begin{aligned} N = &-K_q \cos(\alpha_1)\omega_1^2 + K_q \cos(\alpha_2)\omega_2^2 - K_q \cos(\alpha_3)\omega_3^2 + K_q \cos(\alpha_4)\omega_4^2 \\ &+ K_t b \sin(\alpha_1)\omega_1^2 + K_t b \sin(\alpha_2)\omega_2^2 + K_t b \sin(\alpha_3)\omega_3^2 + K_t b \sin(\alpha_4)\omega_4^2 \end{aligned} \tag{1.58}$$

where b is the arm length, K_t the thrust coefficient and K_q the torque coefficient. What is needed at this point is a matrix form of the previous equations in order to obtain this kind of relation:

$$\begin{bmatrix} F_{orces} \\ M_{oments} \end{bmatrix}_{6\times 1} = MM_{6\times 8}\ u_{8\times 1} \tag{1.59}$$

and then to obtain the input vector u

$$u_{8\times 1} = MM^{\dagger}_{8\times 6} \begin{bmatrix} F_{orces} \\ M_{oments} \end{bmatrix}_{6\times 1} \tag{1.60}$$

where MM is the so called *mixer matrix*, that relates the control variables vector to forces and moments generated by the actuators (vectors and matrices dimensions are shown for the sake of clarity). $MM^{\dagger}$ is the Moore-Penrose pseudo-inverse of matrix MM, treated in Section 1.2.3.
Since it is impossible to rewrite equations from (1.53) to (1.58) in matrix form pointing out both Ω_i^2 and angles α_i $(i = 1, 2, 3, 4)$, what is needed in order to obtain an 8×1 input vector are some mathematical transformations.
The equations from (1.53) to (1.58) are rewritten this way:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -b & 0 & b & 0 & \sigma & 0 & -\sigma \\ b & 0 & -b & 0 & \sigma & 0 & -\sigma & 0 \\ -\sigma & \sigma & -\sigma & \sigma & b & b & b & b \end{bmatrix} \begin{bmatrix} f_{p_1}\cos(\alpha_1) \\ f_{p_2}\cos(\alpha_2) \\ f_{p_3}\cos(\alpha_3) \\ f_{p_4}\cos(\alpha_4) \\ f_{p_1}\sin(\alpha_1) \\ f_{p_2}\sin(\alpha_2) \\ f_{p_3}\sin(\alpha_3) \\ f_{p_4}\sin(\alpha_4) \end{bmatrix} \tag{1.61}$$

where

$$\sigma = \frac{K_q}{K_t} \tag{1.62}$$

$$f_{p_i} = K_t \Omega_i^2. \tag{1.63}$$

Equation (1.61) in compact form becomes

$$\begin{bmatrix} F_{orces} \\ M_{oments} \end{bmatrix}_{6\times 1} = W_{6\times 8}\ f_{u\ 8\times 1} \tag{1.64}$$

where W is the new mixer matrix and f_u is a new vector whose first four elements correspond to the horizontal components of the thrust generated by propellers and the last four to the vertical components.
As in equation (1.60), the vector f_u is computed as

$$f_u = W^{\dagger} \begin{bmatrix} F_{orces} \\ M_{moments} \end{bmatrix} \tag{1.65}$$

$$\begin{bmatrix} f_{p_1}\cos(\alpha_1) \\ f_{p_2}\cos(\alpha_2) \\ f_{p_3}\cos(\alpha_3) \\ f_{p_4}\cos(\alpha_4) \\ f_{p_1}\sin(\alpha_1) \\ f_{p_2}\sin(\alpha_2) \\ f_{p_3}\sin(\alpha_3) \\ f_{p_4}\sin(\alpha_4) \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sigma}{2b} & -\frac{1}{4} & 0 & \frac{1}{2b} & -\frac{\sigma}{4(b^2+\sigma^2)} \\ -\frac{\sigma}{2b} & 0 & -\frac{1}{4} & -\frac{1}{2b} & 0 & \frac{\sigma}{4(b^2+\sigma^2)} \\ 0 & \frac{\sigma}{2b} & -\frac{1}{4} & 0 & -\frac{1}{2b} & -\frac{\sigma}{4(b^2+\sigma^2)} \\ \frac{\sigma}{2b} & 0 & -\frac{1}{4} & \frac{1}{2b} & 0 & \frac{\sigma}{4(b^2+\sigma^2)} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{b}{4(b^2+\sigma^2)} \\ -\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{b}{4(b^2+\sigma^2)} \\ 0 & -\frac{1}{2} & 0 & 0 & 0 & \frac{b}{4(b^2+\sigma^2)} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{b}{4(b^2+\sigma^2)} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \\ L \\ M \\ N \end{bmatrix}. \tag{1.66}$$

In order to explicit Ω_i^2 and α_i $(i = 1, 2, 3, 4)$, some transformations are performed:

$$f_{p_i} = \sqrt{f_u^2(i) + f_u^2(i+4)} \tag{1.67}$$

$$\Omega_i^2 = \frac{f_{p_i}}{K_t} \tag{1.68}$$

$$\alpha_i = \arctan2(f_u(i+4), f_u(i)). \tag{1.69}$$

In this way, thanks to some mathematical transformations, the exact (nonlinear) mixer matrix of the tilt-rotor is derived. It is important to note that the mixer matrix W is constant and depends only from physical quantities and coefficients of the tilt-rotor: thanks to this property, its pseudo-inverse needs to be computed just once and can be stored on-board as part of the controller implementation.

Chapter 2

Feedback linearization control

The tilt-rotor quad-copter is an over-actuated system, in fact it has eight control inputs to control the six degrees of freedom of a body in space. It is capable to track both desired position and orientation and has also two degrees of actuation available to perform additional tasks. These tasks can be achieved by exploiting the internal motions of system, without affecting the tracking capabilities.
The first control strategy adopted to fully control the tilt-rotor quad-copter motion in this thesis is feedback linearization. It is a common control approach used for nonlinear systems: it consists in transforming the nonlinear system to an equivalent linear one through a change of variables and a suitable feedback control law.
Consider the input-affine nonlinear system

$$\dot{x} = f(x) + g(x)u \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^p$ is the vector of inputs. $f(x), g(x)$ are nonlinear functions.
In order to feedback linearize, a change of variables $z = T(x)$, defined for all $x \in D \subset \mathbb{R}^n$, transforms the system into the controller form

$$\dot{z} = Az + B[\alpha(x) + \beta(x)u] \tag{2.2}$$

where (A, B) is controllable and $\beta(x)$ is non singular for all $x \in D$.
Selecting the control law as

$$u = \beta^{-1}(x)[-\alpha(x) + v], \tag{2.3}$$

the equivalent linear system results

$$\dot{z} = Az + Bv \tag{2.4}$$

where v is the new input vector, which can be used to implement the control law

$$v = -Kz. \tag{2.5}$$

In order to have an asymptotically stable closed-loop system, the closed-loop eigenvalues are assigned by defining the gain K such that $(A - BK)$ is Hurwitz. For insights about feedback linearization theory, the reader could refer to Seifried [2013].
This chapter deals with the description of two different feedback-linearization controllers designed for the tilt-rotor model presented in Section 1.3. The first one has been developed in Ryll et al. [2015] and is based on the extension of the system dynamics. The second controller presented is similar to the first one but has a simpler structure thanks to the use of the nonlinear mixer matrix shown in Section 1.6.
After the implementation of these controllers on the Simulink platform, some simulations are performed in order to evaluate their performance.

2.1 Feedback linearization with dynamic extension

The paper Ryll et al. [2015] is one of the first nonlinear control applications for a tilt-rotor quad-copter. Since the presented solution is a working and already tested control system, in this section it is analyzed and implemented in order to understand its functioning and highlight its pros and cons. It is the starting point from which other control strategies will be developed in the thesis.

2.1.1 Equations of motion

In order to show the development of the controller designed in Ryll et al. [2015], the mathematical model equations are written in accordance with the cited paper. The equations of motion are the same presented in Section 1.3: equations (1.38) and (1.47) are translational and rotational equations of motion.
The reduced model for control purposes makes the same simplifications shown in Section 1.5 but, in addition, it neglects also the inertial and gyroscopic effects. Equations (1.52) are written as

$$\begin{cases} \ddot{p} &= \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} {}^{W}R_B F(\alpha)\omega \\ \dot{\omega}_b &= I_b^{-1}\tau(\alpha)\omega \\ {}^{W}\dot{R}_B &= {}^{W}R_B\hat{\omega}_b \end{cases} \tag{2.6}$$

where $F(\alpha)$ and $\tau(\alpha)$ are the coupling matrices that relate the vector ω of propellers spinning signed and squared velocities Ω_i^2 (for $i = 1, 2, 3, 4$) to the thrusts and torques. These matrices have a dependency on the tilting angles, in fact

$$F(\alpha) = \begin{bmatrix} 0 & K_t s_2 & 0 & -K_t s_4 \\ K_t s_1 & 0 & -K_t s_3 & 0 \\ -K_t c_1 & K_t c_2 & -K_t c_3 & K_t c_4 \end{bmatrix} \tag{2.7}$$

$$\tau(\alpha) = \begin{bmatrix} 0 & K_t b c_2 - K_q s_2 & 0 & -K_t b c_4 + K_q s_4 \\ K_t b c_1 + K_q s_1 & 0 & -K_t b c_3 - K_q s_3 & 0 \\ K_t b s_1 - K_q c_1 & -K_t b s_2 - K_q c_2 & K_t b s_3 - K_q c_3 & -K_t b s_4 - K_q c_4 \end{bmatrix} \tag{2.8}$$

where s_1 and c_1 stand for $\sin(\alpha_1)$ and $\cos(\alpha_1)$.
Recalling the notation of Section 1.6:

$$F_{orces} = F(\alpha)\omega \tag{2.9}$$

$$M_{oments} = \tau(\alpha)\omega. \tag{2.10}$$

The first two equations in (2.6) can be also written as

$$\begin{aligned}\begin{bmatrix}\ddot{p}\\ \dot{\omega}_b\end{bmatrix} &= \begin{bmatrix}\begin{bmatrix}0\\0\\g\end{bmatrix}\\ \mathbf{0}_{3\times1}\end{bmatrix} + \begin{bmatrix}\frac{1}{m}{}^{W}R_B & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & I_b^{-1}\end{bmatrix}\begin{bmatrix}F(\alpha) & \mathbf{0}_{3\times4}\\ \tau(\alpha) & \mathbf{0}_{3\times4}\end{bmatrix}\begin{bmatrix}\omega\\ \dot{\alpha}\end{bmatrix} \\ &= f + J_R\begin{bmatrix}\bar{J}_\alpha(\alpha) & \mathbf{0}_{6\times4}\end{bmatrix}\begin{bmatrix}\omega\\ \dot{\alpha}\end{bmatrix} = f + J_R J_\alpha(\alpha)\begin{bmatrix}\omega\\ \dot{\alpha}\end{bmatrix} \\ &= f + J(\alpha)\begin{bmatrix}\omega\\ \dot{\alpha}\end{bmatrix}.\end{aligned} \tag{2.11}$$

It is important to note that in (2.11), the angular tilting velocities $\dot{\alpha}$ are used instead of the tilting angles. Matrix J_α has four null columns, that means

$$rank(J_\alpha) \leq 4. \tag{2.12}$$

Since

$$rank(J) = rank(J_R J_\alpha) \tag{2.13}$$

and J_R is a non-singular full rank matrix ($rank(J_R) = 4$), that means

$$rank(J) = rank(J_\alpha) = rank(\bar{J}_\alpha) \leq 4 < 6. \tag{2.14}$$

Matrix J is singular and since this feedback linearization scheme is based on dynamic inversion, it is not possible to invert the equations of motion expressed in (2.11). The presence of the four null columns in matrix J_α is due to the fact that inputs $\dot{\alpha}$ affect the output dynamics at a higher differential level compared to inputs ω (Ryll et al. [2015]).

2.1.2 Dynamic extension

In this Section the dynamic extension approach will be employed in order to make the input-output map invertible. In particular, it will be shown that the input-output map can be inverted at a higher differential level where inputs $\dot{\alpha}$ explicitly appear.
This is possible expanding the term

$$\bar{J}_\alpha(\alpha)\omega = \sum_{i=1}^{4} \bar{j}_i(\alpha)\Omega_i^2. \tag{2.15}$$

The dynamic extension is obtained by differentiation of equation (2.11) with respect to time:

$$\begin{aligned}\begin{bmatrix}\dddot{p}\\ \ddot{\omega}_b\end{bmatrix} &= J_R\bar{J}_\alpha(\alpha)\dot{\omega} + J_R\sum_{i=1}^{4}\frac{\partial\bar{j}_i(\alpha)}{\partial\alpha}\dot{\alpha}\Omega_i^2 + \dot{J}_R\bar{J}_\alpha(\alpha)\omega \\ &= J_R\begin{bmatrix}\bar{J}_\alpha(\alpha) & \sum_{i=1}^{4}\frac{\partial\bar{j}_i(\alpha)}{\partial\alpha}\Omega_i^2\end{bmatrix}\begin{bmatrix}\dot{\omega}\\ \dot{\alpha}\end{bmatrix} + \begin{bmatrix}\frac{1}{m}{}^W\dot{R}_B F(\alpha)\omega\\ \mathbf{0}_{3\times1}\end{bmatrix} \\ &= J_R J'_\alpha(\alpha,\omega)\begin{bmatrix}\dot{\omega}\\ \dot{\alpha}\end{bmatrix} + d(\alpha,\omega,\omega_b) \\ &= A(\alpha,\omega)\begin{bmatrix}\dot{\omega}\\ \dot{\alpha}\end{bmatrix} + d(\alpha,\omega,\omega_b).\end{aligned} \tag{2.16}$$

Matrix $A(\alpha,\omega)$ is the new input-output mixer matrix and has $rank(A) = 6$ as long as $\Omega_i \neq 0$ for $i = 1,2,3,4$ as shown in Ryll et al. [2015]. Matrix A can be assured to be full rank by preventing the propellers from stopping.

2.1.3 Control law

Since $rank(A) = 6$, equation (2.16) can be inverted obtaining

$$\begin{bmatrix}\dot{\omega}\\ \dot{\alpha}\end{bmatrix} = A^\dagger\Big(\begin{bmatrix}\dddot{p}_r\\ \ddot{\omega}_r\end{bmatrix} - d\Big) + (I_8 - A^\dagger A)z \tag{2.17}$$

where $A^\dagger$ is the Moore-Penrose pseudo-inverse of matrix A, $\dddot{p}_r$ and $\ddot{\omega}_r$ are virtual inputs and the last term $(I_8 - A^\dagger A)z$ corresponds to the null space projection of matrix A. The presence of a 2-dimensional null space for matrix A is a consequence of the system over-actuation, that has eight control inputs and six controlled degrees of freedom.
In order to achieve the full input-output linearization it is required that

$$\begin{bmatrix}\dddot{p}\\ \ddot{\omega}_b\end{bmatrix} = \begin{bmatrix}\dddot{p}_r\\ \ddot{\omega}_r\end{bmatrix}. \tag{2.18}$$

The control inputs are obtained integrating the vector computed in (2.17):

$$\Omega_i = \int \dot{\Omega}_i \tag{2.19}$$

$$\alpha_i = \int \dot{\alpha}_i \tag{2.20}$$

for $i = 1,2,3,4$.
Once the system is feedback linearized, the dynamics of the closed-loop system can be assigned by selecting

$$\dddot{p}_r = \dddot{p}_d + K_{p_1}(\ddot{p}_d - \ddot{p}) + K_{p_2}(\dot{p}_d - \dot{p}) + K_{p_3}(p_d - p) \tag{2.21}$$

$$\ddot{\omega}_r = \ddot{\omega}_d + K_{\omega_1}(\dot{\omega}_d - \dot{\omega}_b) + K_{\omega_2}(\omega_d - \omega_b) + K_{\omega_3}e_R \tag{2.22}$$

where p_d and ω_d are the desired position and attitude, ω_d is the desired body angular velocity and e_R is the orientation error. It is defined as

$$e_R = \frac{1}{2}[{}^W R_B^T R_d - R_d^T {}^W R_B]^\vee \tag{2.23}$$

where $[.]^\vee$ is the map from SO(3) to $\mathbb{R}^3$ and R_d is the desired orientation matrix, defined as the desired rotation between the body frame and the world frame. $K_{p_1}, K_{p_2}, K_{p_3}$ and $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$ are positive definite gain matrices that define Hurwitz polynomials.

2.1.4 Optimization of the control law

As already stated, the tilt-rotor is an over-actuated system and the two redundant degrees of actuation can be exploited to perform additional tasks. The vector z is projected onto the null space of matrix A and does not produce actions interfering with the output tracking objectives.
In this case, the control law is optimized in order to minimize the energy consumption, *i.e.*, to minimize the norm of the vector of the squared and signed spinning velocities ω. This is achieved by minimizing the cost function

$$H(\omega) = \sum_{i=1}^{4} h(\Omega_i^2) \tag{2.24}$$

where

$$h(\Omega_i^2) = \begin{cases} k_{h_1} \tan^2(\gamma_1 |\Omega_i^2| + \gamma_2) & \text{if } \Omega_{min}^2 < |\Omega_i^2| \leq \Omega_{hover}^2 \\ k_{h_2}(|\Omega_i^2| - \Omega_{hover}^2)^2 & \text{if } |\Omega_i^2| > \Omega_{hover}^2 \end{cases} \tag{2.25}$$

$$\gamma_1 = \frac{\pi}{2(\Omega_{hover}^2 - \Omega_{min}^2)} \tag{2.26}$$

$$\gamma_2 = -\gamma_1 \Omega_{hover}^2 \tag{2.27}$$

and $k_{h_1} > 0, k_{h_2} > 0$ suitable scalar gains. The parameter Ω_{min} is defined as the minimum value for the propeller spinning speed and Ω_{hover} is the spinning speed required to each propeller in order to let the drone to hover, computed as:

$$\Omega_{hover} = \sqrt{\frac{mg}{4K_t}}. \tag{2.28}$$

The minimization of $H(\omega)$ is obtained by setting

$$z = -k_H \begin{bmatrix} \nabla_\omega H(\omega) \\ \mathbf{0} \end{bmatrix} \tag{2.29}$$

where $k_H > 0$ is a suitable gain. It is important to note that with this minimization function, no optimizations are made on the alpha angles α_i $(i = 1, 2, 3, 4)$. A second cost function $H_\alpha(\alpha)$ could be implemented in order to have a full optimization action. Figure 2.1 shows the function $h_i(\Omega_i^2)$ and Table 2.1 recaps the parameters to characterize this function.

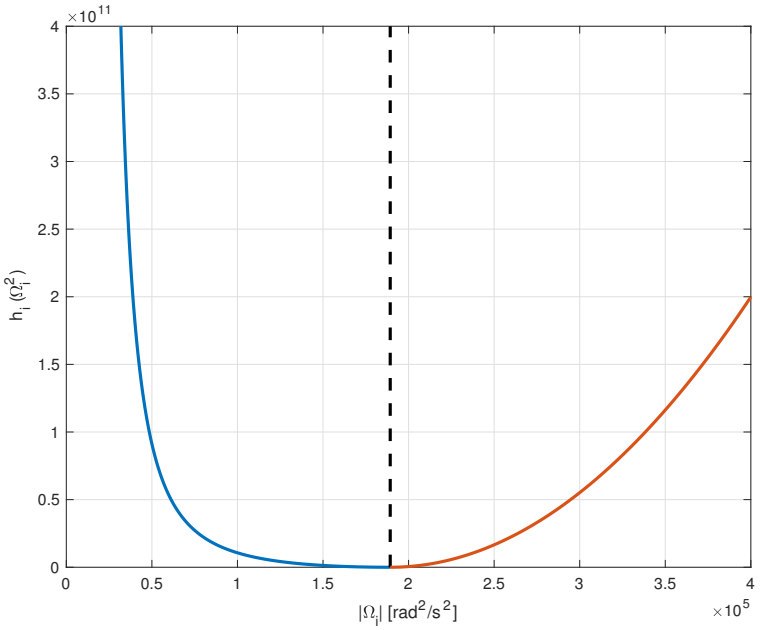

Figure 2.1: $h_i(\Omega_i^2)$ function

Parameter	Value
Ω_{min}	120 rad/s
k_{h_1}	10^{10}
k_{h_2}	4.5
k_H	0.0001

Table 2.1: Parameters to characterize $H(\omega)$

2.1.5 Tuning of control parameters

Since the feedback linearization scheme makes the system linear, the control parameters $K_{p_1}, K_{p_2}, K_{p_3}$ and $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$ have the important role to assign the closed-loop dynamics of the overall system. These gains are computed with an eigenvalues assignment strategy.
Rewriting equations (2.21) and (2.22)

$$(\dddot{p}_d - \dddot{p}_r) + K_{p_1}(\ddot{p}_d - \ddot{p}) + K_{p_2}(\dot{p}_d - \dot{p}) + K_{p_3}(p_d - p) = 0 \tag{2.30}$$

$$(\ddot{\omega}_d - \ddot{\omega}_r) + K_{\omega_1}(\dot{\omega}_d - \dot{\omega}_b) + K_{\omega_2}(\omega_d - \omega_b) + K_{\omega_3} e_R = 0 \tag{2.31}$$

and having $(p_d - p) = e_p$, equation (2.30) is written in state space form considering $e_p = x_1$:

$$\begin{cases} \dot{x}_1 = x_2 = \dot{e}_p \\ \dot{x}_2 = x_3 = \ddot{e}_p \\ \dot{x}_3 = -K_{p_3} x_1 - K_{p_2} x_2 - K_{p_1} x_3. \end{cases} \tag{2.32}$$

The dynamic matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -K_{p_3} & -K_{p_2} & -K_{p_1} \end{bmatrix} \tag{2.33}$$

whose characteristic polynomial is

$$\lambda^3 + K_{p_1}\lambda^2 + K_{p_2}\lambda + K_{p_3} = 0. \tag{2.34}$$

It is possible to assign the three eigenvalues of the system p_1, p_2, p_3 by choosing

$$(\lambda + p_1)(\lambda + p_2)(\lambda + p_3) = 0 \tag{2.35}$$

$$\lambda^3 + (p_1 + p_2 + p_3)\lambda^2 + (p_1 p_2 + p_2 p_3 + p_3 p_1)\lambda + p_1 p_2 p_3 = 0. \tag{2.36}$$

The diagonal gain matrices are obtained from the assigned eigenvalues by

$$\begin{cases} K_{p_1} = (p_1 + p_2 + p_3) I_3 \\ K_{p_2} = (p_1 p_2 + p_2 p_3 + p_3 p_1) I_3 \\ K_{p_3} = (p_1 p_2 p_3) I_3. \end{cases} \tag{2.37}$$

The same is done for the rotational dynamics, obtaining $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$.
The eigenvalues of translational and rotational dynamics are placed in

$$\begin{cases} p_1 = -0.5 \\ p_2 = -1 \\ p_3 = -1.2 \end{cases} \tag{2.38}$$

$$\begin{cases} r_1 = -0.5 \\ r_2 = -3 \\ r_3 = -5. \end{cases} \tag{2.39}$$

All eigenvalues are expressed in rad/s. These assignments are made considering the bandwidth of the actuators. Since no standard procedures are defined to tune a nonlinear controller, the eigenvalues are assigned in order to keep the translational and rotational dynamics at least one decade below the actuators bandwidth. Figure 2.2 shows the Bode diagrams of actuators and of the assigned translational and rotational dynamics.

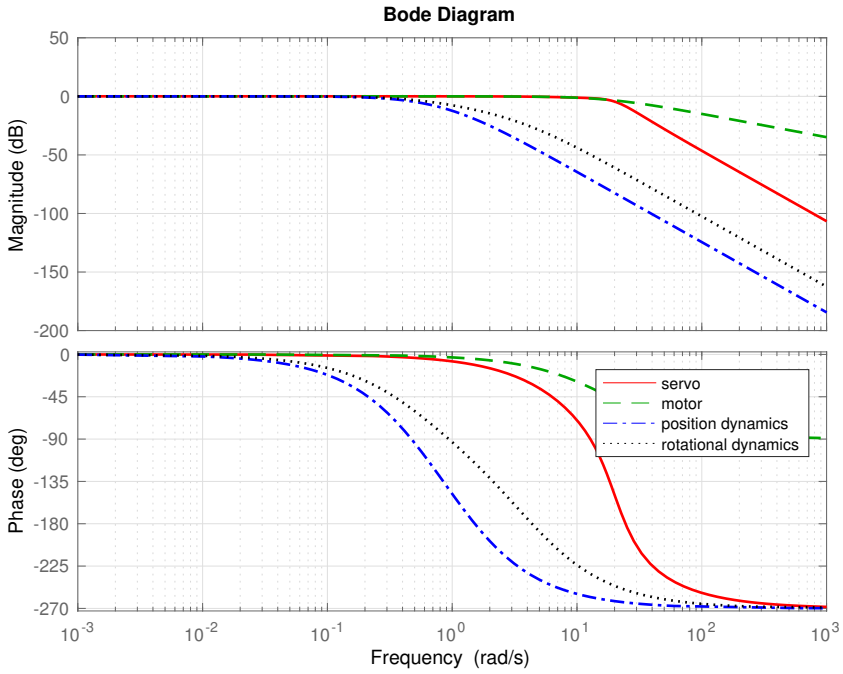

Figure 2.2: Bode diagram of the assigned translational and rotational dynamics compared to actuators bandwidth

2.1.6 Simulation results

The controller based on dynamic extension has been implemented in the Simulink environment in order to perform simulations.
The complete model of the tilt-rotor is tested along with the controller to perform a tracking problem. Two smooth set-points $(x_d, v_d, a_d, j_d, R_d, \omega_d, \dot{\omega}_d, \ddot{\omega}_d)$ are assigned as a function of time. The first one describes an eight-shape trajectory whereas the second one is composed by fifth order polynomials and describes a hovering condition with a non-null roll angle. Thanks to the use of fifth order polynomials, the overall desired position and attitude could be defined combining several trajectories, specifying for each one its initial and final positions, velocities, accelerations and attitudes, angular speeds and angular accelerations. This

leads to a set-point with continuous accelerations and angular accelerations. The jerk and the angular jerk have discontinuities that do not affect the overall performance: the trajectory definition is made in order to keep them small.

Polynomial trajectory

This first set-point shows the capability of the tilt-rotor to track position and attitude separately, thanks to its over-actuated structure.
The simulation starts with the drone in hover with attitude $\Phi = [0^\circ\ 0^\circ\ 0^\circ]^T$. At time $t = 25s$ the first attitude variation is requested: $+90^\circ$ yaw in three seconds. The adjustment time after this rotation is quite short and after that, at time $t = 38s$, the second attitude variation: $+15^\circ$ roll in five seconds, keeping the same position. From this point until the end of the simulation the drone keeps this attitude with non null roll angle remaining in the same position.
Figures 2.3, 2.4, 2.5 show position, position error, attitude, attitude error and the actuators commands of the tilt-rotor during the simulation where the polynomial trajectory is tested. The velocity and angular speed plots are not shown since they are not necessary to understand the behavior of the system with this set-point trajectory.
Looking at the simulation results it is appreciable that the controlled system is able to track position and attitude set-points keeping the errors small. In fact, the position error shows a peak of around 0.04 m during the pitch variation and the maximum attitude error is around 3°, letting to consider this behavior satisfiable.

Eight-shape trajectory

The eight-shape trajectory adopted is the Lemniscate of Bernoulli. It is defined as

$$p_d = \begin{bmatrix} \frac{\sin(2\omega t)}{3-\cos(2\omega t)} \\ \\ \frac{2\cos(\omega t)}{3-\cos(2\omega t)} \\ \\ height \end{bmatrix} \tag{2.40}$$

where t is the simulation time, ω is the linear speed and $height$ is the altitude at which the eight-shape trajectory is performed. The velocity, acceleration and jerk set-points v_d, a_d, j_d are obtained differentiating with respect to time equation (2.40). Besides the position set-point, the drone is required also to keep the roll and pitch angles equal to zero and to follow the direction of the trajectory with

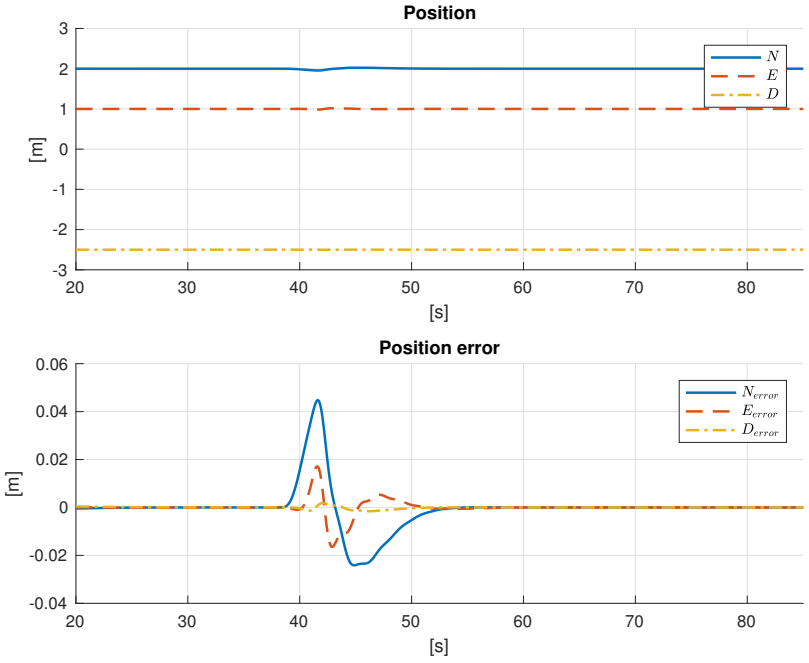

Figure 2.3: Dynamic extension: position and position error in polynomial trajectory simulation

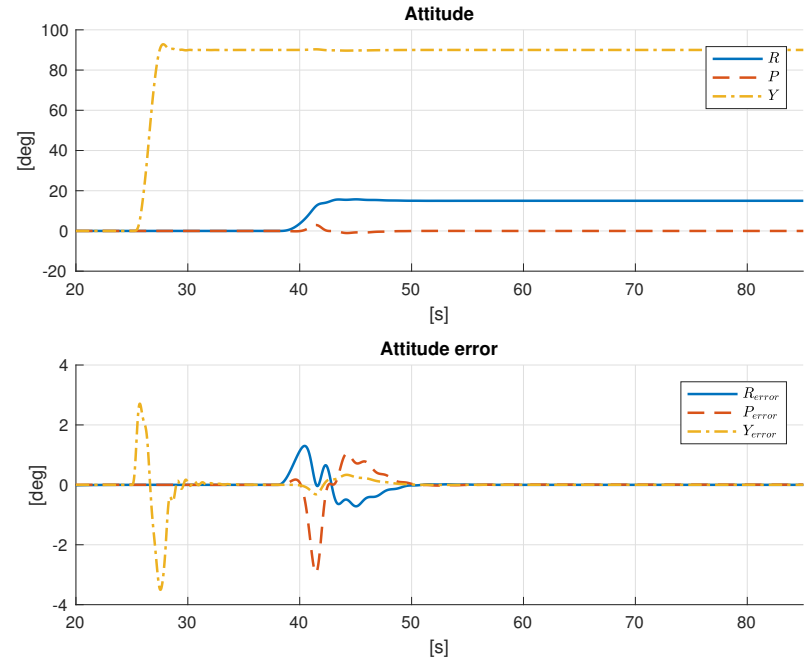

Figure 2.4: Dynamic extension: attitude and attitude error in polynomial trajectory simulation

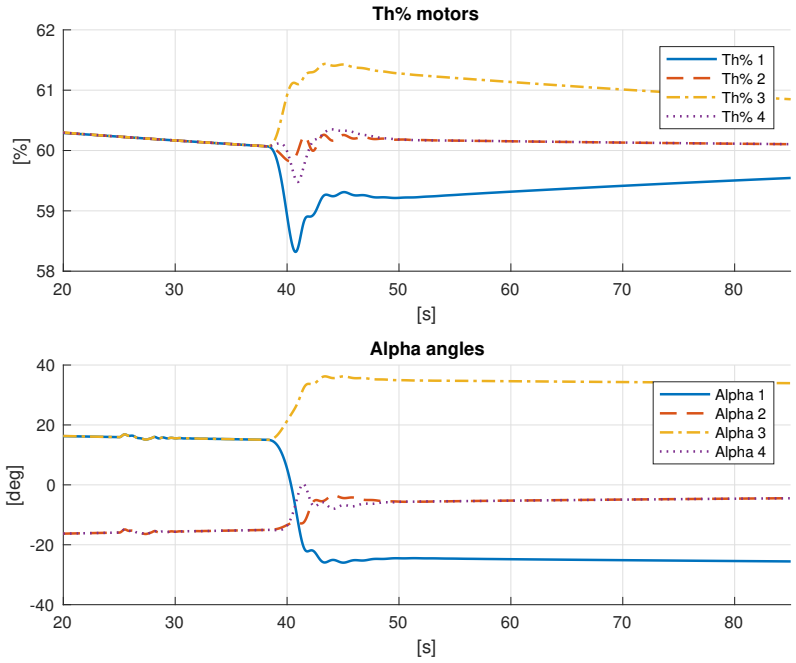

Figure 2.5: Dynamic extension: throttle percentages and tilting angles in polynomial trajectory simulation

the yaw angle. This is achieved by imposing

$$\psi = \arctan2(v_d(2), v_d(1)) \tag{2.41}$$

$$\dot{\psi} = \frac{a_d(2)v_d(1) - a_d(1)v_d(2)}{v_d(1)^2 + v_d(2)^2}. \tag{2.42}$$

Figures 2.6, 2.7, 2.8, 2.9, 2.10 show position and position error, velocity and velocity error, attitude and attitude error, angular speed and angular speed error and the actuators commands of the tilt-rotor during the simulation where the eight-shape trajectory is tested.
Looking at the simulation results, the errors are kept very small except for the yaw angle that has peaks of around 10°. The yaw angle error is maximum when the yaw rate reaches its maximum (or minimum) value: it is considered acceptable since the trajectory is tracked and the yaw angle error has a stable behavior.
In Figures 2.5 and 2.10 the action of the energy optimization function is visible. In hover the best configuration achievable by the tilt-rotor is the one with the tilt angles close to zero and the spinning speed close to the hovering speed. The controller instead computes a solution that is one of the infinite possible solutions (pseudo-inverting a 6×8 matrix is equal to solve an under-determined system with infinite solutions) and then the optimization function acts in order to reduce the spinning speeds, as explained in Subsection 2.1.4, working on the null space of

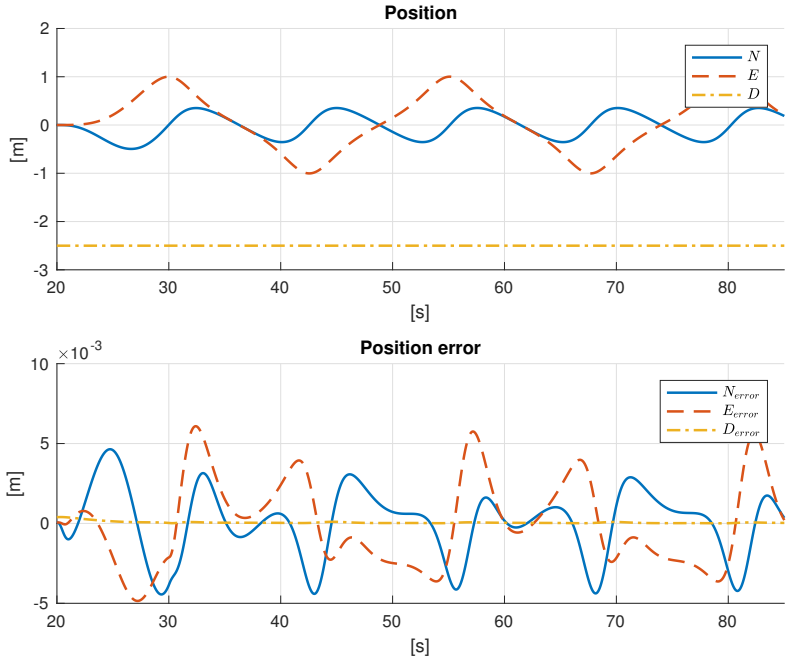

Figure 2.6: Dynamic extension: position and position error in eight-shape trajectory simulation

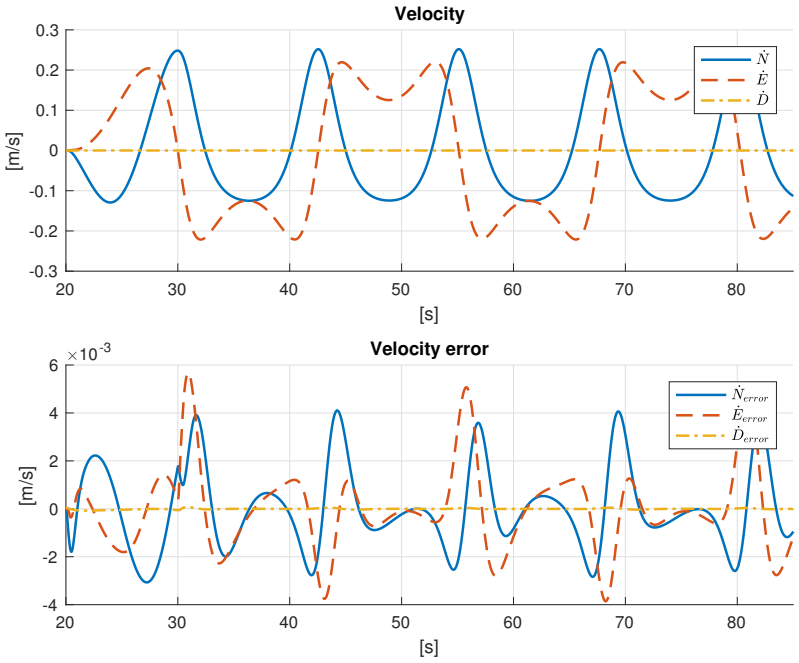

Figure 2.7: Dynamic extension: velocity and velocity error in eight-shape trajectory simulation

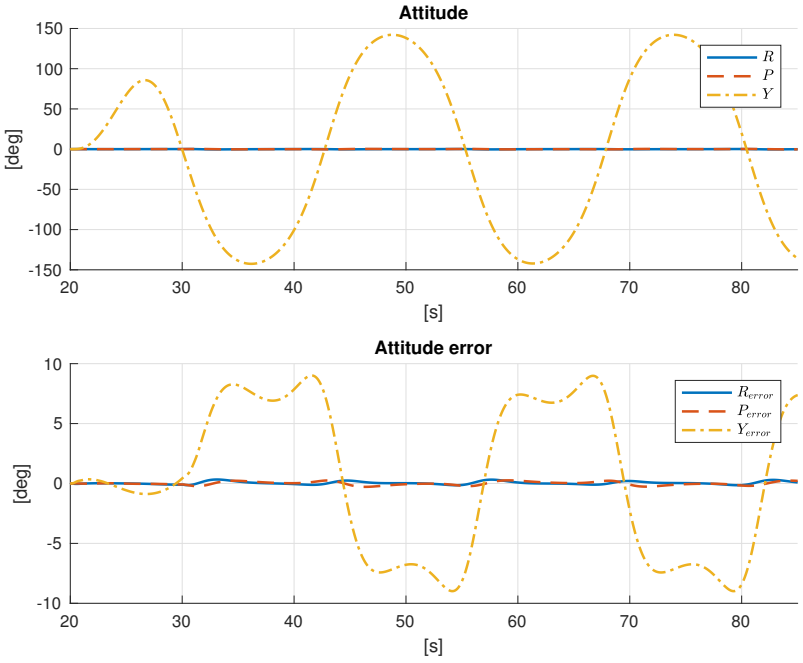

Figure 2.8: Dynamic extension: attitude and attitude error in eight-shape trajectory simulation

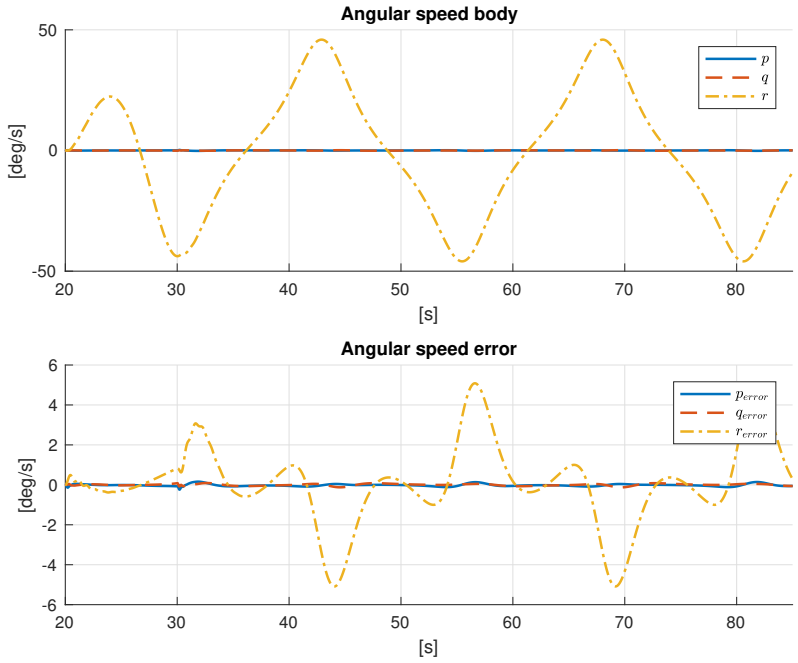

Figure 2.9: Dynamic extension: angular speed and error in eight-shape trajectory simulation

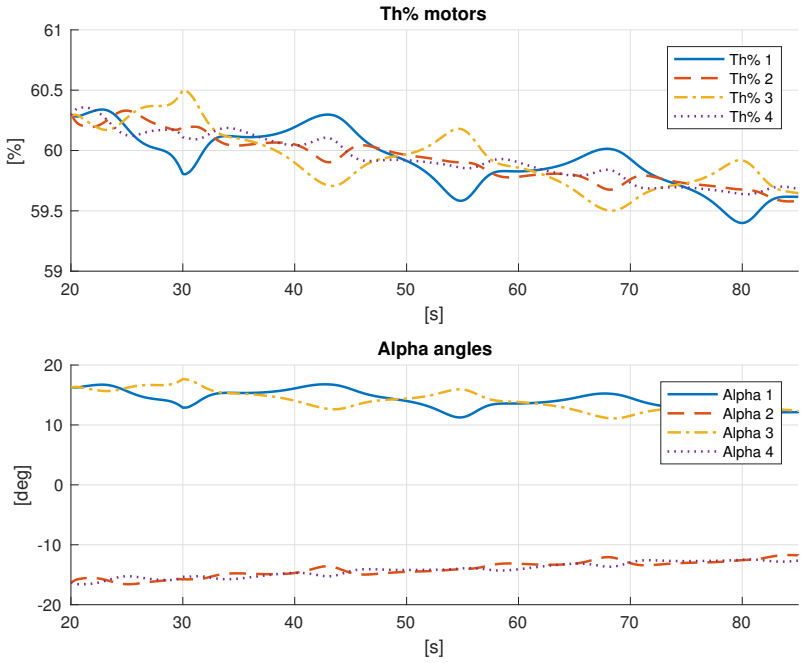

Figure 2.10: Dynamic extension: throttle percentages and tilting angles in eight-shape trajectory simulation

matrix A. The propeller spinning speeds are reduced by a very small percentage during the simulation (around 0.5% in 60 seconds). This is necessary because, after several simulations, it has been noticed that an aggressive action of the optimization strategy would lead to destabilize the system.

2.1.7 Considerations

The control solution described in this section presents several issues that limit its capabilities. In fact:

- The control law requires the first, second and third derivative of the set-point trajectory. This requires a complex analytic set-point generation.
- The control law needs also acceleration and angular acceleration measurements. Usually the acceleration feedback is given by the accelerometer but the angular acceleration is given by an observer.
- The pseudo-inversion of the mixer matrix A needs to be computed at every control cycle, causing a really heavy computational load. In fact, due to the dynamic extension shown in Section 2.16 the mixer matrix A and its Moore-

Penrose pseudo-inverse are time-dependent since $A(\omega, \alpha)$ and $\omega(t), \alpha(t)$. This operation is the critical point of the overall control system.

- The controller is sensitive to initial conditions and set-points, that needs to be continuous and well defined.
- The energy optimization action must be kept very small in order to keep the system stable.

2.2 Feedback linearization

The second control strategy developed in this thesis is similar to the one presented in the previous Section 2.1: it is a feedback linearization control with the assignment of the closed-loop dynamics, but it introduces several improvements. The dynamic extension of the system requires an awkward operation that is the differentiation with respect to time of the equations of motion. This step is hereby avoided through the use of the invertible mixer matrix presented in Section 1.6.

2.2.1 Control law

Recalling the equations of motion of the reduced model (1.52) and the mixer matrix formulation (1.64) and (1.61), the system can be written as

$$\begin{bmatrix} \ddot{p} \\ \dot{\omega}_b \end{bmatrix} = \left(\begin{bmatrix} 0 \\ 0 \\ g \\ \mathbf{0} \end{bmatrix} \right) + \begin{bmatrix} \frac{1}{m}{}^W R_B & \mathbf{0} \\ \mathbf{0} & I_b^{-1} \end{bmatrix} W f_u \tag{2.43}$$

$$\begin{bmatrix} \ddot{p} \\ \dot{\omega}_b \end{bmatrix} = f + J_R W f_u. \tag{2.44}$$

In order to feedback linearize it, the control law is computed as

$$f_u = W^\dagger J_R^{-1} \left(\begin{bmatrix} \ddot{p}_r \\ \dot{\omega}_r \end{bmatrix} - f \right). \tag{2.45}$$

The control inputs Ω_i and α_i $(i = 1, 2, 3, 4)$ are computed thanks to the relations (1.67), (1.68), (1.69).
In order to achieve the full input-output linearization, it is required that

$$\begin{bmatrix} \ddot{p} \\ \dot{\omega}_b \end{bmatrix} = \begin{bmatrix} \ddot{p}_r \\ \dot{\omega}_r \end{bmatrix}. \tag{2.46}$$

The closed-loop dynamics can be assigned by selecting

$$\ddot{p}_r = \ddot{p}_d + K_{p_1}(\dot{p}_d - \dot{p}) + K_{p_2}(p_d - p) + K_{p_3} \int (p_d - p) \tag{2.47}$$

$$\dot{\omega}_r = \dot{\omega}_d + K_{\omega_1}(\omega_d - \omega_b) + K_{\omega_2} e_R + K_{\omega_3} \int e_R \tag{2.48}$$

where p_d and ω_d are the desired position and attitude, ω_d is the desired body angular velocity and e_R is the orientation error. $K_{p_1}, K_{p_2}, K_{p_3}$ and $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$ are positive definite gain matrices that define Hurwitz polynomials.

2.2.2 Tuning of control parameters

The procedure to tune the control parameters is the same as in Subsection 2.1.5, but computed at a lower differential level. In fact, the control parameters $K_{p_1}, K_{p_2}, K_{p_3}$ and $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$ are computed with the same eigenvalues assignment strategy. Equations (2.47) and (2.48) are written as

$$(\ddot{p}_d - \ddot{p}_r) + K_{p_1}(\dot{p}_d - \dot{p}) + K_{p_2}(p_d - p) + K_{p_3}\int(p_d - p) = 0 \tag{2.49}$$

$$(\dot{\omega}_d - \dot{\omega}_r) + K_{\omega_1}(\omega_d - \omega_b) + K_{\omega_2}e_R + K_{\omega_3}\int e_R = 0 \tag{2.50}$$

and having $(p_d - p) = e_p$, equation (2.49) is written in state space form considering $\int e_p = x_1$:

$$\begin{cases} \dot{x}_1 = x_2 = e_p \\ \dot{x}_2 = x_3 = \dot{e}_p \\ \dot{x}_3 = -K_{p_3}x_1 - K_{p_2}x_2 - K_{p_1}x_3. \end{cases} \tag{2.51}$$

The dynamic matrix is:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -K_{p_3} & -K_{p_2} & -K_{p_1} \end{bmatrix} \tag{2.52}$$

whose characteristic polynomial is

$$\lambda^3 + K_{p_1}\lambda^2 + K_{p_2}\lambda + K_{p_3} = 0. \tag{2.53}$$

It is possible to assign the three eigenvalues of the system p_1, p_2, p_3 by

$$(\lambda + p_1)(\lambda + p_2)(\lambda + p_3) = 0 \tag{2.54}$$

$$\lambda^3 + (p_1 + p_2 + p_3)\lambda^2 + (p_1p_2 + p_2p_3 + p_3p_1)\lambda + p_1p_2p_3 = 0. \tag{2.55}$$

The control diagonal matrices are obtained from the assigned eigenvalues by

$$\begin{cases} K_{p_1} = (p_1 + p_2 + p_3)I_3 \\ K_{p_2} = (p_1p_2 + p_2p_3 + p_3p_1)I_3 \\ K_{p_3} = (p_1p_2p_3)I_3. \end{cases} \tag{2.56}$$

The same is done for the rotational dynamics, obtaining $K_{\omega_1}, K_{\omega_2}, K_{\omega_3}$.
The eigenvalues of translational and rotational dynamics are placed in

$$\begin{cases} p_1 = -0.3 \\ p_2 = -0.5 \\ p_3 = -1.5 \end{cases} \tag{2.57}$$

$$\begin{cases} r_1 = -0.1 \\ r_2 = -3 \\ r_3 = -7. \end{cases} \tag{2.58}$$

All eigenvalues are expressed in rad/s. These assignments are made through the same bandwidth considerations made in Subsection 2.1.5. Figure 2.11 shows the Bode diagrams of actuators and of the assigned translational and rotational dynamics. Comparing Figure 2.11 with the Bode diagram of Figure 2.2, it is easy

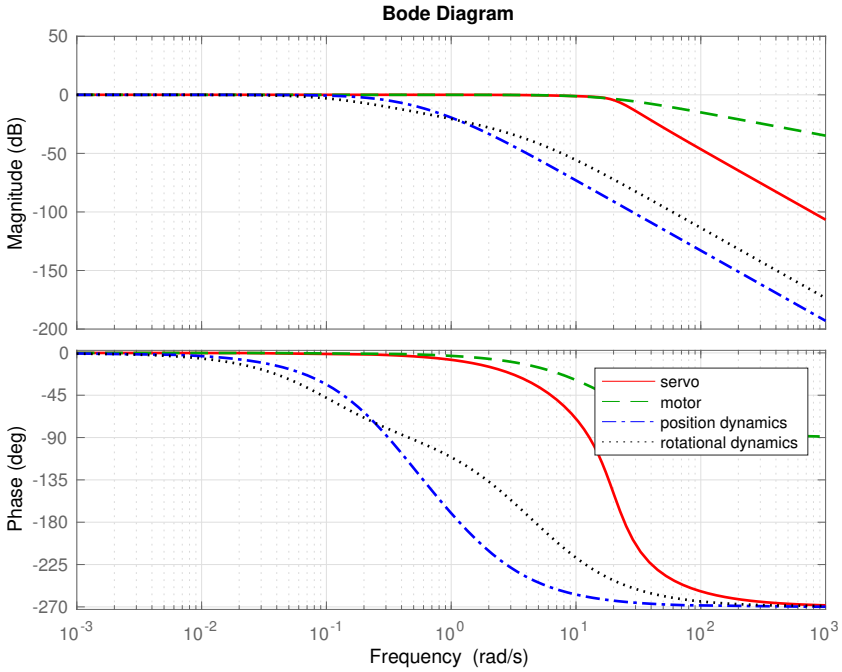

Figure 2.11: Bode diagram of the assigned translational and rotational dynamics compared to actuators bandwidth with feedback linearization controller

to see that the assigned dynamics are similar and the respective bandwidths are almost the same.

2.2.3 Simulation results

The developed controller is much simpler than the one presented in Section 2.1 and provides better performance if tested with the same trajectories of Subsection 2.1.6. This is the reason why, from this point until the end of the work, much complex trajectories are tested in order to push to the limits the controlled system. In particular, the following set-points describe the full motion of a flight simulation: starting from the ground, the drone takes off, performs a trajectory and then lands.

Polynomial trajectory

Initially the trajectory describes the take off maneuver from the ground, moving the tilt-rotor from a generic point on the ground with non null yaw angle to the hovering point in $p = [0\ 0\ -2.5]^T$ with attitude $\Phi = [0°\ 0°\ 0°]^T$. Then, at time $t = 20s$ the first attitude change is requested: $+90°$ of yaw angle in five seconds; at time $t = 25s$ the second attitude change: $+25°$ of pitch angle in ten seconds and, after that, the set-point remains constant for other ten seconds. At time $t = 45s$ a very challenging attitude variation: the rotation of $-180°$ around the Z_b axis is performed, taking the drone to the final pose with $\Phi = [0°\ -25°\ -90°]^T$. From this position and attitude, the drone lands in the origin with attitude $\Phi = [0°\ 0°\ 0°]^T$. Figures 2.12, 2.13, 2.14 show position, position error, attitude, attitude error and the actuators commands of the tilt-rotor during the simulation where the polynomial trajectory is tested. The velocity and angular speed plots are not shown for this simulation since they are not necessary to understand the behavior of the system with this set-point trajectory.
Looking at the simulation results, all the errors are kept small enough to be considered satisfiable. It is important to note that the maneuver of rotating around the Z_b axis when the pitch angle is not $0°$ is really complicated and a maximum position error of $0.1\ m$ is considered acceptable.

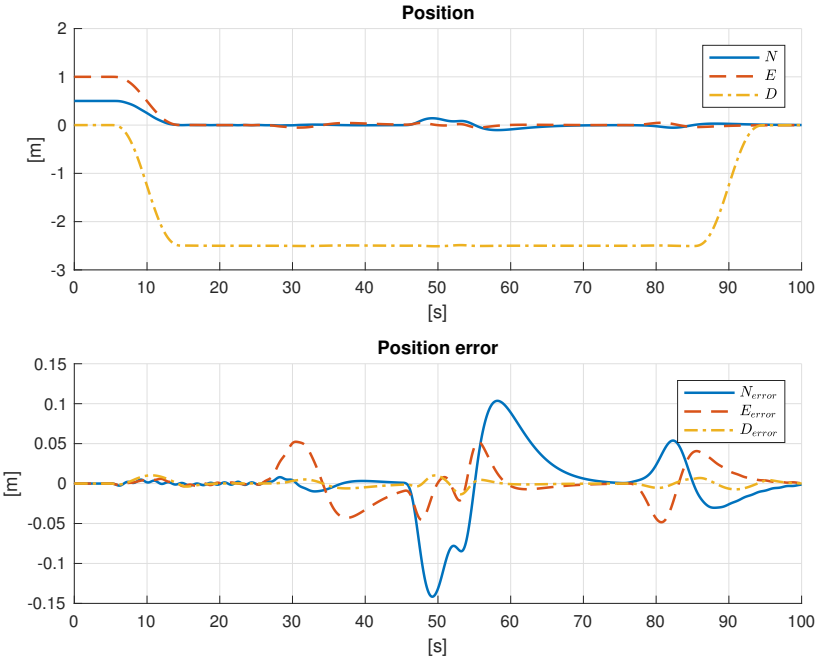

Figure 2.12: Feedback linearization: position and position error in polynomial trajectory simulation

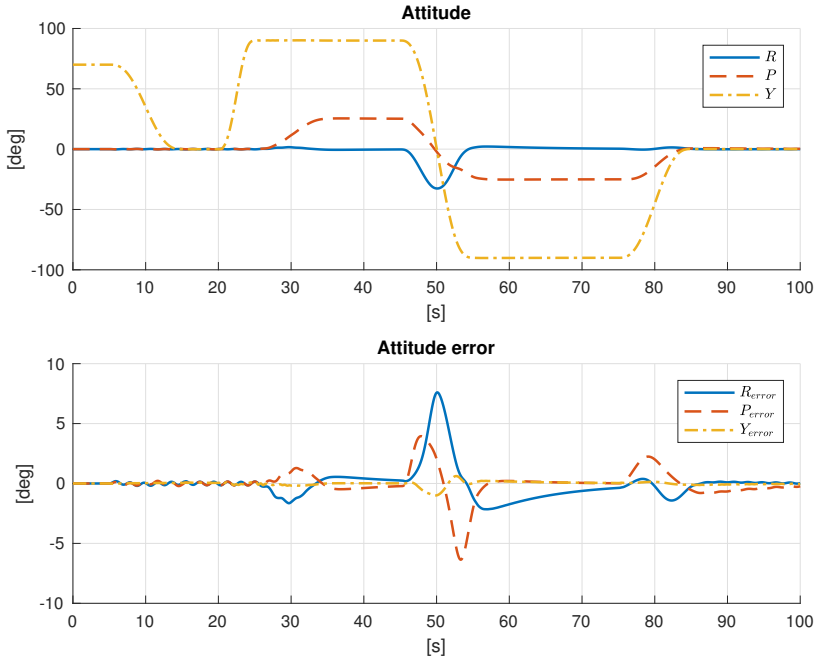

Figure 2.13: Feedback linearization: attitude and attitude error in polynomial trajectory simulation

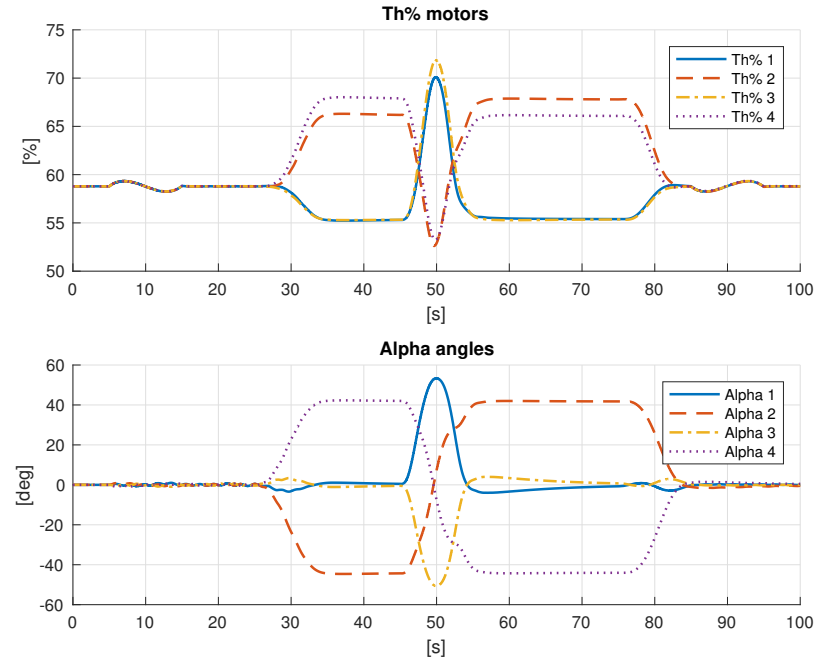

Figure 2.14: Feedback linearization: throttle percentages and tilting angles in polynomial trajectory simulation

Eight shape trajectory

Besides the take-off maneuver, that is the same of the polynomial trajectory, the set-point describes an eight-shape trajectory as in Subsection 2.1.6. In addition, the yaw angle follows the direction of the velocity angle and a superimposed sinusoidal roll movement is requested, with an amplitude of $\pm 15°$. Finally the drone is requested to land in the origin with attitude $\Phi = [0° \ 0° \ 0°]^T$.
Summarizing, the requested trajectory is composed by:

- Take-off from a generic point ($p = [0.5 \ 1 \ 0]^T$) with non null yaw angle on the ground ($\psi = +70°$)
- Hover in $p = [0 \ 0 \ -2.5]^T$ with attitude $\Phi = [0° \ 0° \ 0°]^T$
- Perform the eight-shape trajectory and:
  - the direction of the first arm follows the velocity vector direction, *i.e.*, the direction of the eight, acting on the yaw angle
  - a sinusoidal roll movement of $\pm 15°$ is superimposed to the motion
- Land in the origin $p = [0 \ 0 \ 0]^T$ with attitude $\Phi = [0° \ 0° \ 0°]^T$.

Figures 2.15, 2.16, 2.17, 2.18, 2.19 show position and position error, velocity and velocity error, attitude and attitude error, angular speed and angular speed error and the actuators commands of the tilt-rotor during the simulation where the eight-shape trajectory is tested.
Also for the eight-shape trajectory the overall behavior of the controlled system is considered satisfiable. The errors are, in fact, kept small even if the tested trajectory is complicated, in particular due to the sinusoidal roll movement superimposed.

2.2.4 Considerations

The presented control strategy is much simpler than the one in Ryll et al. [2015], analyzed in Section 2.1 because of several reasons.
Firstly, the adopted mixer matrix is a constant matrix: this means that it is no more necessary to compute it at any iteration but it is computed once. The computational load becomes lighter.
Secondly, no dynamic extension is needed and the optimization strategy for energy saving shown in Subsection 2.1.4 is no more necessary. In fact, when the pseudo-inverse of the mixer matrix W is computed, the solution found is the one that minimizes the norm of the control input vector. This lead to a vector with minimized propeller spinning speeds and tilting angles. This property was not valid for the controller shown in Section 2.1 because the pseudo-inversion was executed at a higher differential level, where the vector minimized is the one with

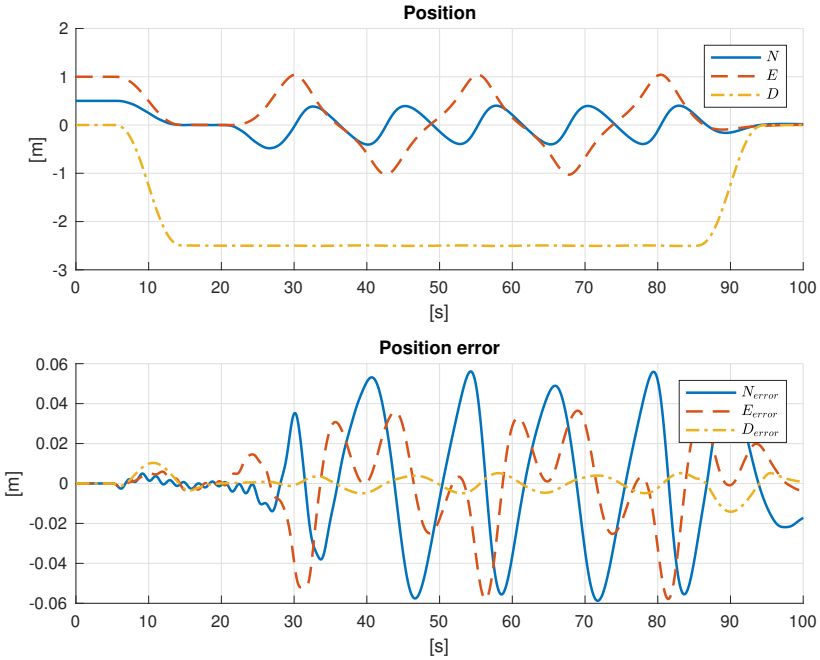

Figure 2.15: Feedback linearization: position and position error in eight-shape trajectory simulation

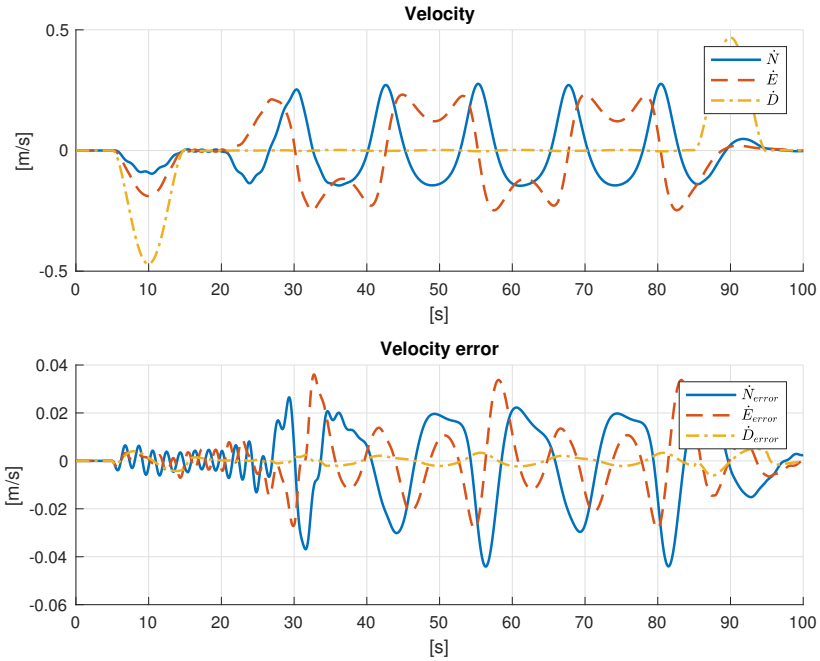

Figure 2.16: Feedback linearization: velocity and velocity error in eight-shape trajectory simulation

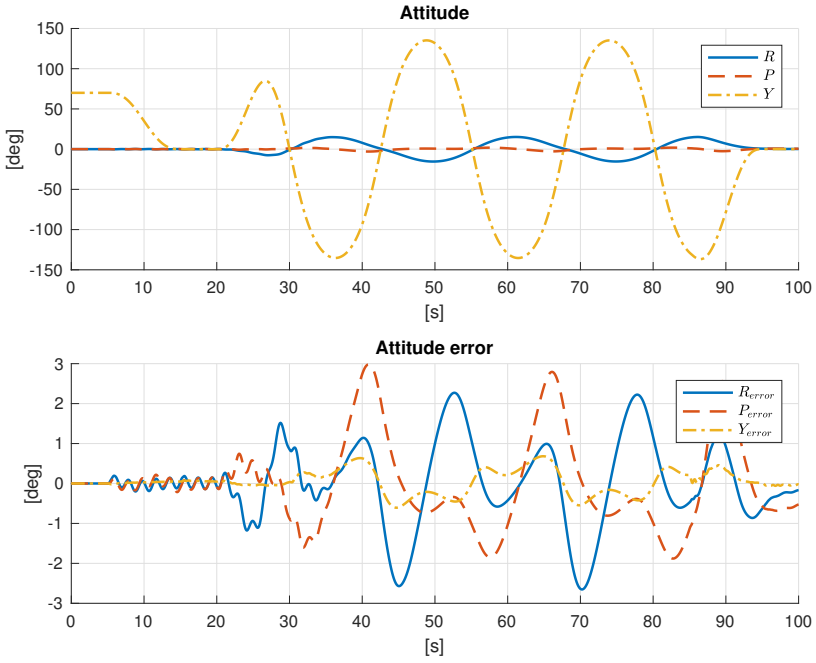

Figure 2.17: Feedback linearization: attitude and attitude error in eight-shape trajectory simulation

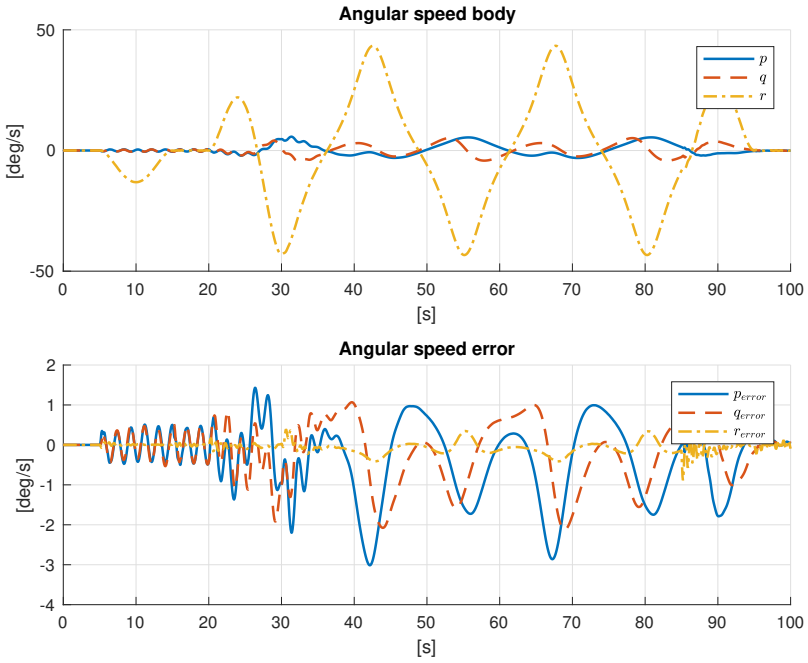

Figure 2.18: Feedback linearization: angular speed and error in eight-shape trajectory simulation

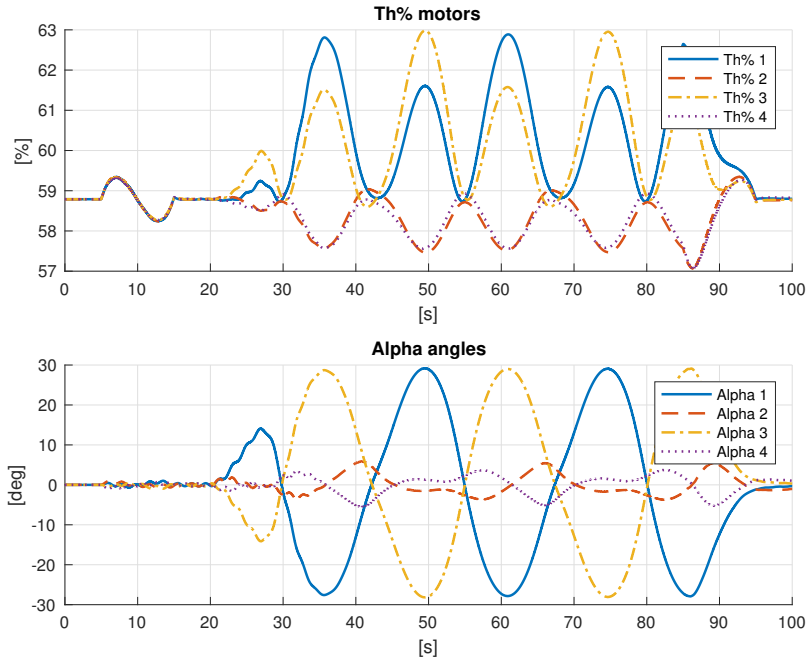

Figure 2.19: Feedback linearization: throttle percentages and tilting angles in eight-shape trajectory simulation

propeller spinning accelerations and tilting rates: the minimization of these quantities have no energy saving effects.
The null space projection of the mixer matrix W can be exploited to perform additional tasks, *e.g.*, a recovery action in case of failure of one or more actuators. This topic will not be investigated further in the following , but it is a valid improvement for future developments.

Chapter 3

Geometric control

The motion of the tilt-rotor quad-copter evolves in the nonlinear manifold of rigid-body displacement SE(3). In order to address the tilt-rotor control problem to its natural framework, the geometric control theory is adopted. This choice lets to exploit a full attitude representation in SO(3) without resorting to attitude parameterizations, that introduce singularities or ambiguities like the Euler angles or the quaternions, and to overcome to topological obstructions, that can be taken into account by design.
In this chapter two Lyapunov-based geometric controllers for the tilt-rotor quad-copter are shown and implemented: their design and all the mathematical demonstrations are presented and proved in Invernizzi and Lovera [2017a] and Invernizzi and Lovera [2017b].

3.1 Geometric control law for the fully actuated case

The first controller presented in this section is a basic kind of geometric controller, useful to test the stability and the capabilities of this kind of control strategy. In the following, the geometric control law is presented and tested through simulations. It is hereby assumed that the tilt-rotor quadcopter is capable to produce forces and moments in the whole space, since no constraints for the maximum tilting angles are introduced. This control law represents the theoretical starting point from which, in the next Section, an improved geometric control law is developed: it is able to take into account for maximum tilting limitations and, as a consequence, it constrains the generated control force within a cone region.

3.1.1 Tracking errors

In order to perform the computation of the geometric control law, the tracking errors need to be defined. Considering the simplified model for control purposes shown in equations (1.52), the position and velocity error resolved to the inertial frame are expressed as

$$e_p = p - p_d \tag{3.1}$$

$$e_v = v - v_d. \tag{3.2}$$

Since in this control architecture the attitude is given directly in SO(3), the attitude error in SO(3) is obtained according to the left error representation, that is defined as

$$R_e = {}^W R_B R_d^T \tag{3.3}$$

where $R_d \in \mathrm{SO}(3)$ is the desired orientation matrix

$$R_d = \begin{bmatrix} b_{d_1} & b_{d_2} & b_{d_3} \end{bmatrix}. \tag{3.4}$$

The use of the left error instead of the right error leads to a simplified controller (see Bullo and Murray [1999]).
The left angular velocity error is computed comparing the current orientation tangent vector with the desired tangent vector in the tangent space. Since the left error representation represents the map τ_l between SO(3) and its tangent space,

$${}^W\dot{R}_B - \tau_l(\dot{R}_d) = {}^W\dot{R}_B - R_e\dot{R}_d = {}^W R_B(\hat{\omega}_b - \hat{\omega}_d) \tag{3.5}$$

from which the left angular velocity error is

$$e_\omega = \omega_b - \omega_d. \tag{3.6}$$

Another quantity that is used in the following is the error navigation function, defined as

$$\Psi = \frac{1}{2} tr(K_R(I - R_e)) \tag{3.7}$$

where I is the 3×3 identity matrix and K_R is a positive definite gain matrix:

$$K_R = \begin{bmatrix} k_{R_1} & 0 & 0 \\ 0 & k_{R_2} & 0 \\ 0 & 0 & k_{R_3} \end{bmatrix}. \tag{3.8}$$

Recalling Invernizzi and Lovera [2017a], the main properties of the error navigation function are:

1. Ψ is locally positive definite about $R_e = I_{3\times 3}$
2. the left trivialized derivative of Ψ, *i.e.*, the gradient, is

$$T_I^* L_{R_e}(d_{R_e}\Psi) = skew(K_R R_e)^\vee = e_R \tag{3.9}$$

3. the four critical points of Ψ, for distinct $k_{R_1}, k_{R_2}, k_{R_3}$ and for which $e_R = 0$, are $\{R \in \mathrm{SO}(3) : {}^W R_B = R_d \bigcup {}^W R_B = \exp(\pi \hat{e}_i) R_d\}$
4. for $\Psi < \eta < c_1$ is locally quadratic

$$h_1 ||e_R||^2 \leq \Psi \leq h_2 ||e_R||^2 \tag{3.10}$$

where

$$h_1 = \frac{c_1}{c_2 + c_3^2}, \; h_2 = \frac{c_3}{c_1(c_1 - \eta)} \tag{3.11}$$

$$c_1 = min\{k_{R_1} + k_{R_2}, k_{R_2} + k_{R_3}, k_{R_3} + k_{R_1}\} \tag{3.12}$$
$$c_2 = max\{(k_{R_1} - k_{R_2})^2, (k_{R_2} - k_{R_3})^2, (k_{R_3} - k_{R_1})^2\} \tag{3.13}$$
$$c_3 = max\{k_{R_1} + k_{R_2}, k_{R_2} + k_{R_3}, k_{R_3} + k_{R_1}\} \tag{3.14}$$

and the only critical point is

$$R_e = I_3. \tag{3.15}$$

3.1.2 Control law

Recalling the position, velocity, attitude and angular speed errors from equations (3.1), (3.2), (3.9), (3.6), the adopted geometric control law is

$$f_c = {}^W R_B^T f_c^d \tag{3.16}$$
$$\tau_c = -R_d^T e_R - K_\omega e_\omega + I_b \dot{\omega}_d + \omega_d \times I_b \omega_b \tag{3.17}$$

where the ideal control action expressed in the inertial frame is defined as

$$f_c^d = -K_p e_p - K_v e_v + m(\dot{v}_d - g e_3). \tag{3.18}$$

The control torque computed in (3.17) is Almost Globally Exponentially Stable (AGAS). In fact, recalling Invernizzi and Lovera [2017a], considering the attitude kinematics and dynamics given by equations (1.52) and the torque control law defined in (3.17), for any positive definite matrix K_ω and K_R, and a constant $0 < \eta < c_1$, if the initial conditions satisfy

$$\frac{1}{2} e_\omega^T(0) I_b e_\omega^T + \Psi(R_e(0)) < \eta, \tag{3.19}$$

the zero equilibrium of the closed-loop tracking errors (e_R, e_ω) is exponentially stable. Therefore, the attitude error converges to zero, that implies

$$b_i \to b_{d_i}$$
$${}^W R_B \to R_d.$$

This also implies that the translational equation of motion shown in (1.52) becomes

$$\ddot{p} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} R_d R_d^T f_c^d = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} f_c^d \tag{3.20}$$

$$= \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m}(-K_p e_p - K_v e_v) + \dot{v}_d - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.21}$$

$$= \frac{1}{m}(-K_p e_p - K_v e_v) + \dot{v}_d. \tag{3.22}$$

Selecting $\dot{e}_v = \ddot{p} - \dot{v}_d$, it becomes

$$m\dot{e}_v = -K_p e_p - K_v e_v \tag{3.23}$$

that is Globally Asymptotically Stable (GAS) in the origin $(e_p = 0, e_v = 0)$.

3.1.3 Simulation results

As in Chapter 2, the presented control strategy has been implemented in the Simulink environment to be tested along with the model of the tilt-rotor. The set-point trajectories for the simulations are the same shown in Subsection 2.2.3:

- a polynomial trajectory involving attitude changes
- an eight-shape trajectory with a superimposed sinusoidal roll movement.

The control parameters adopted in the numerical simulations are summarized in Table 3.1.

Parameter	Value
K_p	12 I_3
K_v	8 I_3
K_R	0.4 I_3
K_ω	0.15 I_3

Table 3.1: Geometric controller parameters

Polynomial trajectory

The polynomial trajectory describes the take off and landing maneuvers at the beginning and at the end of the simulation, and the following attitude changes:

- at time $t = 20s$, $+90°$ of yaw angle in five seconds
- at time $t = 25s$, $+25°$ of pitch angle in ten seconds
- at time $t = 45s$, $-180°$ around the Z_b axis, taking the tilt-rotor to the final attitude $\Phi = [0° \ -25° \ -90°]^T$
- at time $t = 75s$, the tilt-rotor returns to the initial attitude $\Phi = [0° \ 0° \ 0°]^T$ in ten seconds.

The set-point tracking can be appreciated in Figures 3.1, 3.2, 3.3.
Looking at the simulation results, the polynomial trajectory is well tracked. Both position and attitude errors have peaks during the rotation around the Z_b axis that are smaller if compared to the peaks given by the feedback linearization controller. Also, the tilting action of the servo-motors is a bit smaller, leading to a faster and more precise response of the controlled system.

Eight-shape trajectory

The eight-shape trajectory, as in Subsection 2.2.3, starts describing the take-off maneuver, then the the eight-shape defined thanks to the Lemniscate of Bernoulli, and finally the landing maneuver. A sinusoidal roll movement of $\pm 15°$ is superimposed during the eight-shape part of the trajectory. Figures 3.4, 3.5, 3.6, 3.7, 3.8 show the simulation of the tilt-rotor model tested along with the geometric controller.
Comparing the simulation results with those obtained using the feedback linearization controller, the overall tracking errors are two/three times smaller when the same trajectory is performed. It is possible to state that the exponentially convergent control law of the geometric controller presented in this section has better results than the feedback linearization control law of Section 2.2 in terms of set-point tracking.

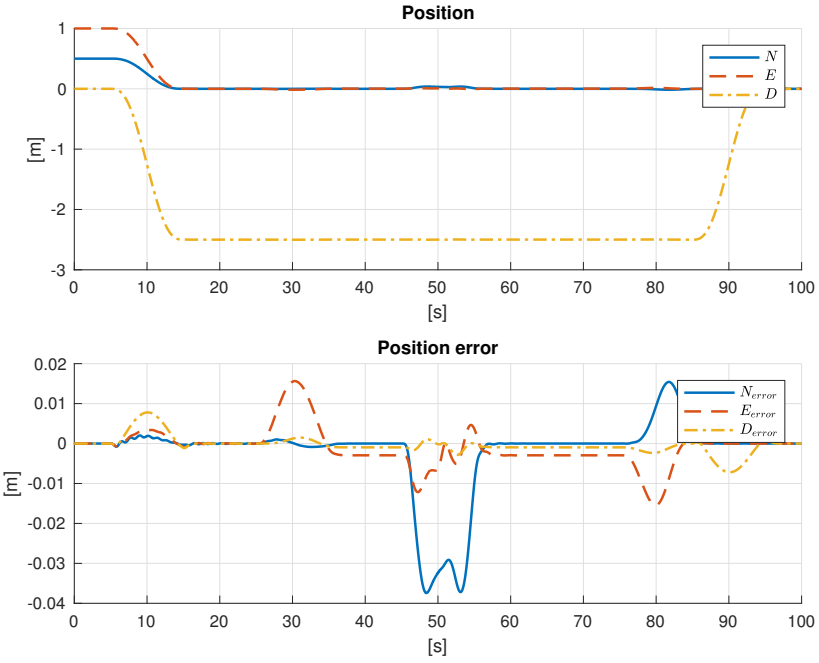

Figure 3.1: Geometric control: position and position error in polynomial trajectory simulation

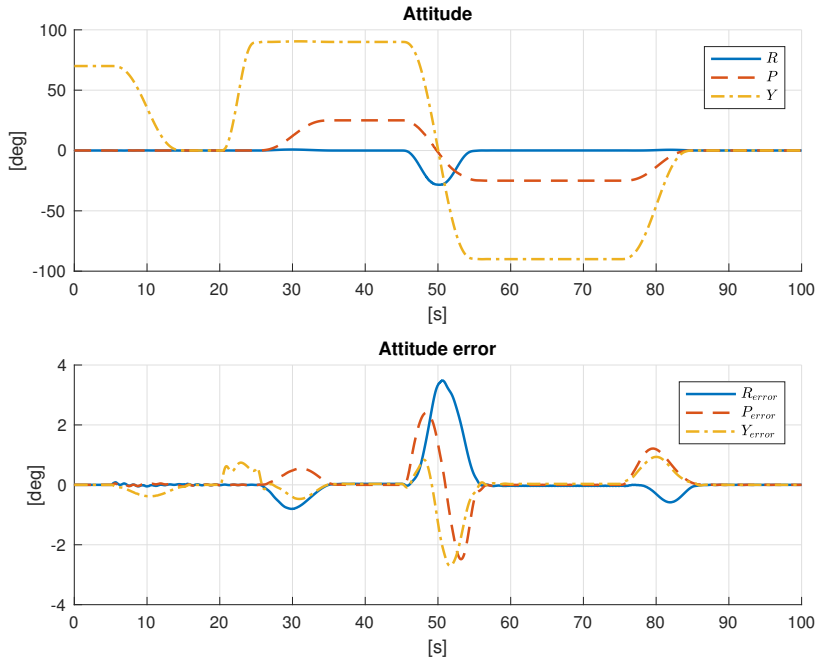

Figure 3.2: Geometric control: attitude and attitude error in polynomial trajectory simulation

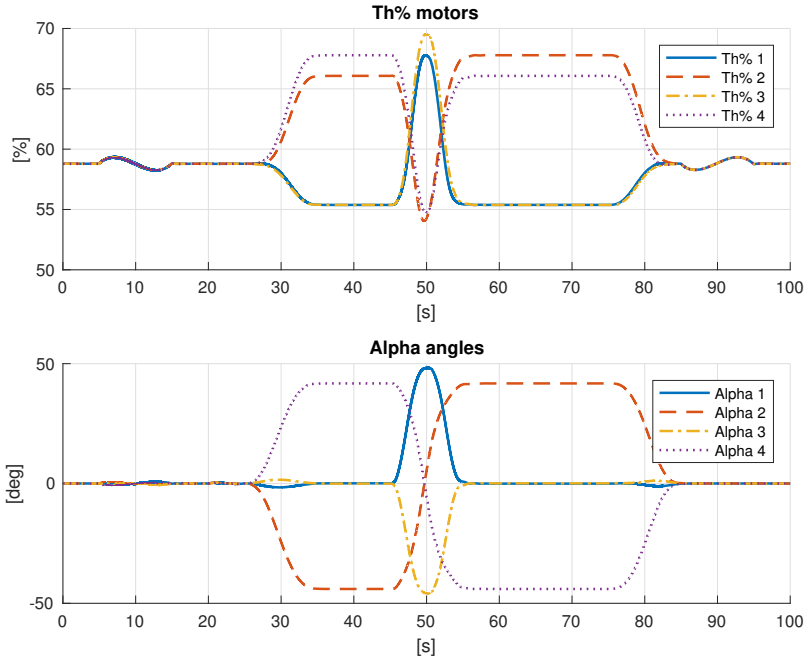

Figure 3.3: Geometric control: throttle percentages and tilting angles in polynomial trajectory simulation

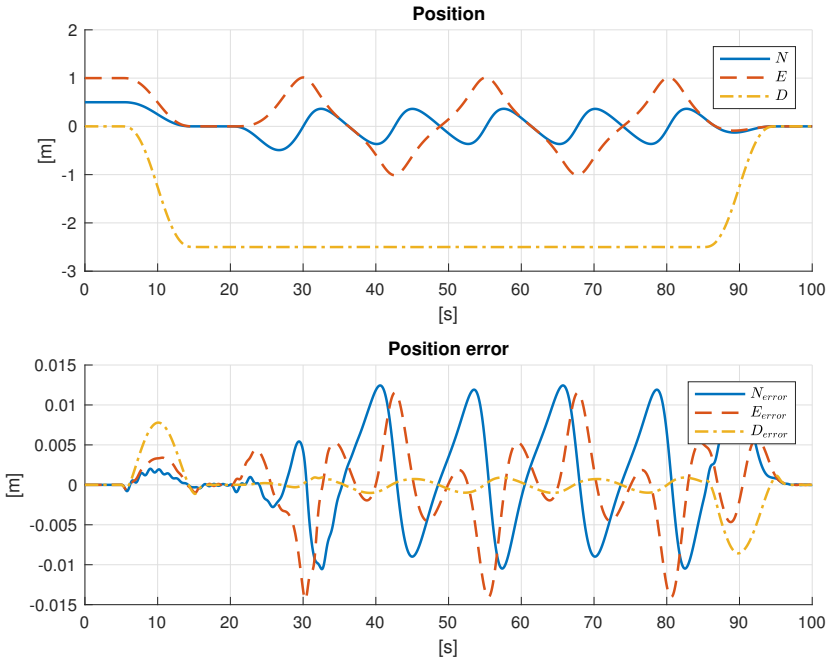

Figure 3.4: Geometric control: position and position error in eight-shape trajectory simulation

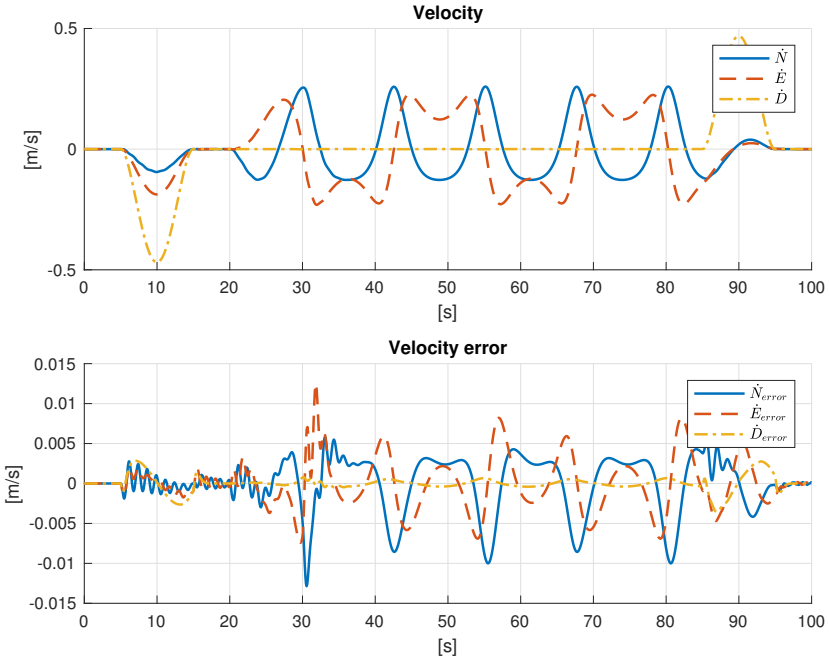

Figure 3.5: Geometric control: velocity and velocity error in eight-shape trajectory simulation

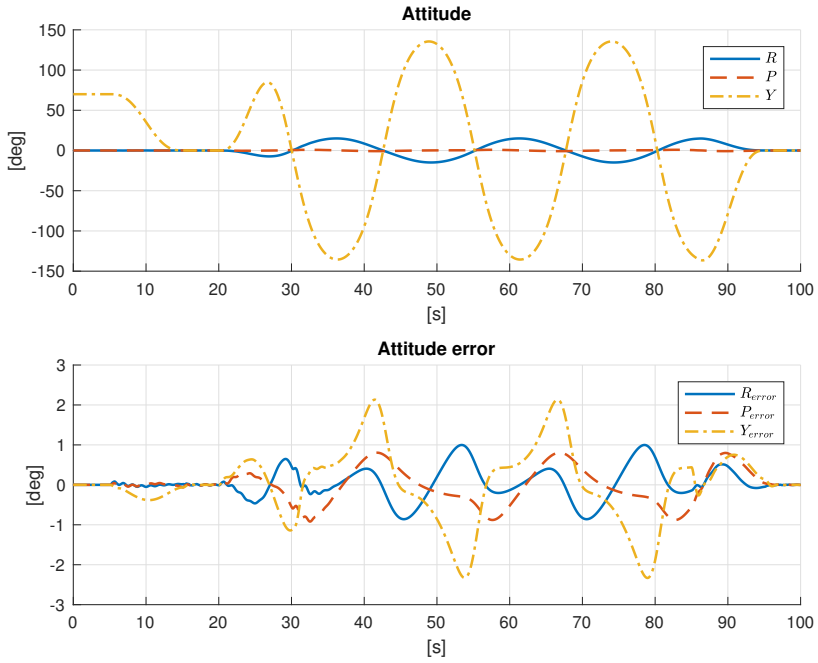

Figure 3.6: Geometric control: attitude and attitude error in eight-shape trajectory simulation

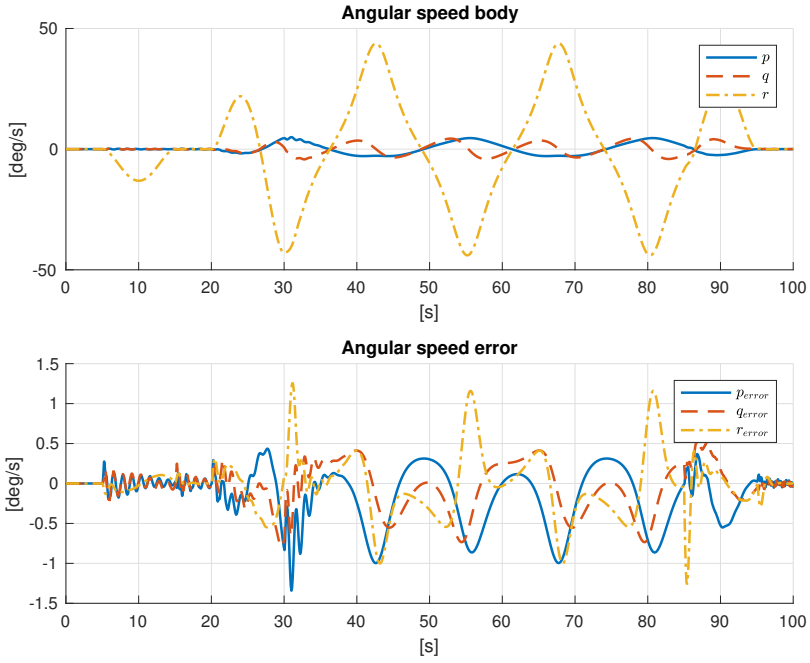

Figure 3.7: Geometric control: angular speed and error in eight-shape trajectory simulation

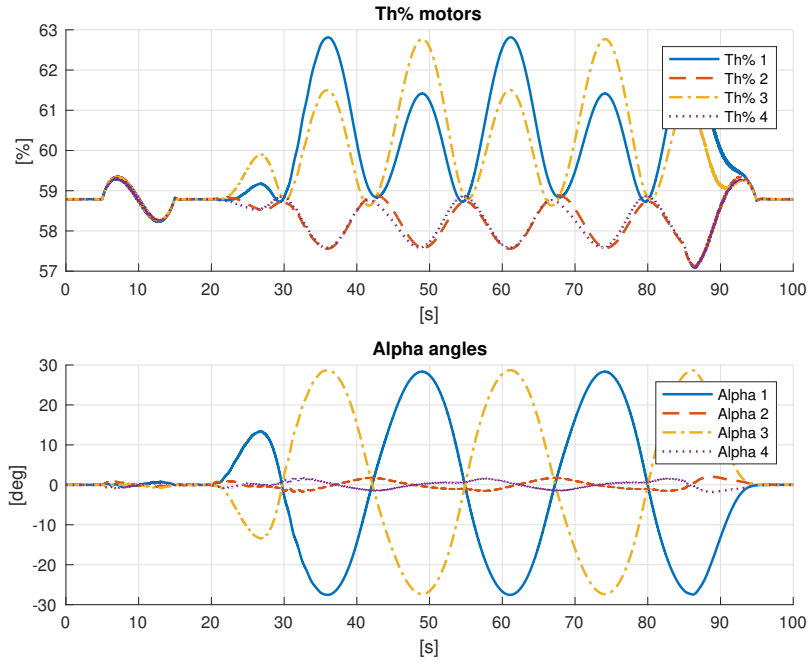

Figure 3.8: Geometric control: throttle percentages and tilting angles in eight-shape trajectory simulation

3.1.4 Considerations

Given a smooth set-point trajectory $(p_d, \dot{p}_d, \ddot{p}_d, \Phi_d, \omega_d, \dot{\omega}_d)$, the controlled system is able to track both position and attitude set-points. Since no constraints are defined for the force control law, in order to obtain a stable behavior of the system, the position and orientation set-points must be feasible, *i.e.*, they must not require control actions that the actuators are not able to fulfill. In order to overcome this, in the next section a dynamic geometric control law is proposed, capable to take into account for actuation limitations by constraining the desired control force within an admissible cone region.

3.2 Geometric tracking control with dynamic reference attitude

The second geometric controller presented has been designed in Invernizzi and Lovera [2017b], to which the reader could refer for further investigations. As for the controller presented in Section 3.1, it resorts to geometric control theory, setting the tilt-rotor tracking control problem on the group of rigid displacement SE(3) in order to overcome parameterization issues and to take into account by design for topological obstructions.
This control law has been designed to guarantee position tracking even in case the control force cannot span the 3D space. This is obtained by properly modifying the attitude set-point, which can be exactly tracked only if compatible with the position tracking requirements.

3.2.1 Actuation constraints

Considering that the servo-motors have tilting limitations, it is reasonable to state that the control force f_c computed by the controller lies approximately in a spherical sector defined around the third body axis b_3, as Figure 3.9 shows.
The spherical sector is defined as

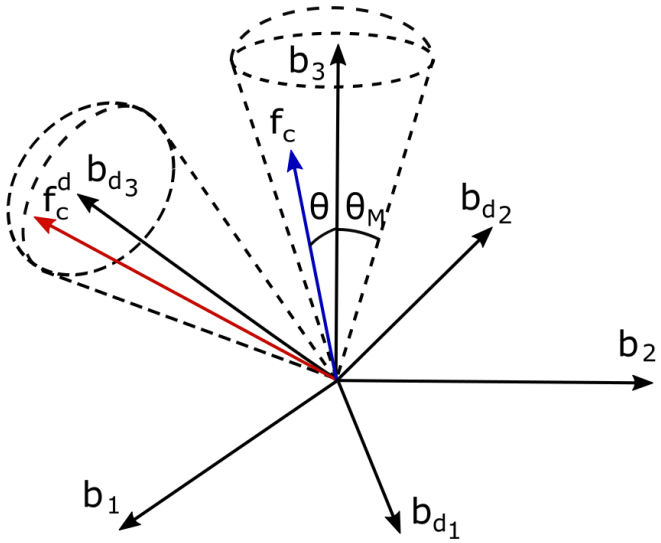

Figure 3.9: Spherical sector definition

$$0 < \cos(\theta_M \leq \frac{f_c^T e_3}{||f_c||}) = \cos(\theta) \tag{3.24}$$

$$||f_c|| < f_M \tag{3.25}$$

where

- θ_M is the maximum tilting angle of the servo-motors
- θ is the angle between the delivered force and the vertical body axis b_3
- f_M is the maximum deliverable control force.

In order to comply with the constraints shown in (3.24) and (3.25), the control force computed in (3.16) is modified as follows:

$$f_c = c(\Psi) R_d^T f_c^d \tag{3.26}$$

where the scaling function $c(\Psi)$ is

$$c(\Psi) = \frac{\Psi_M - \Psi}{\Psi_M} \ , \Psi_M > \eta \ . \tag{3.27}$$

The control force vector computed in equation (3.26) has the same components of the desired control force vector of the equation (3.18) resolved to the desired body frame and scaled by a term dependent on the navigation function Ψ.
The scaling function satisfies the conditions:

$$\lim_{\Psi \to 0} c(\Psi) = 1 \tag{3.28}$$

$$0 < c(\Psi) \leq 1 \tag{3.29}$$

and thanks to this function, the delivered control force f_c is kept within a cone region, defined in (3.24) and shown in Figure 3.10, as long as the desired force f_c^d is kept inside the cone defined around the desired third axis b_{d_3}. Additionally, it holds true that

$$||f_c|| = c(\Psi)||R_d^T f_c^d|| \leq ||f_c^d||, \tag{3.30}$$

that means that the constraint in (3.25) is satisfied as long as

$$||f_c^d|| \leq f_M. \tag{3.31}$$

The expression of the control force suggests a possible strategy to take into account the cone region constraint. Indeed, by expecting equation (3.26), the desired control force f_c^d is rotated by R_d^T. Hence, by relaxing the requirement on the attitude tracking, it is always possible to select R_d such that the corresponding control force is compatible with the constraint.
The Local Exponential Stability (LES) of the geometric control law is proven in Invernizzi and Lovera [2017a] in two steps: after showing the exponential stability of the torque control law, the convergence of the force control law is derived. The control torque computed in (3.17) has, in fact, the Almost Global Asymptotic Stability (AGAS) property. In fact, if the conditions in equation (3.19) hold, the zero equilibrium of the closed-loop tracking errors (e_R, e_ω) is asymptotically stable, given the attitude kinematics and dynamics in equations (1.52), the torque control

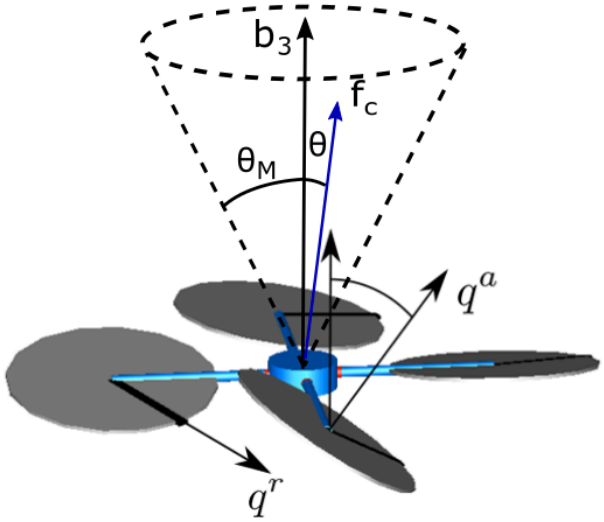

Figure 3.10: Cone region constraint

law defined in (3.17), any positive definite matrix K_ω and K_R and a constant $0 < \eta < c_1$.
To show the rationale behind the proposed control law, let's assume that the attitude error converges to zero, namely $b_i \to b_{d_i}$, then:

$$\begin{aligned} R_e &\to I_{3\times 3} \\ c(\Psi) &\to 1 \end{aligned}$$

and the control force tends to the desired force

$$^W R_B f_c = c(\Psi)\, ^W R_B R_d f_c^d = c(\Psi) R_e f_c^d = f_c^d, \tag{3.32}$$

necessary to track the desired position.

3.2.2 Tracking errors

The next step to define the geometric controller shown in Invernizzi and Lovera [2017b], is to compute the modified attitude reference that let to take into account for the tilting limitation of the actuators. In order to do so and to compute the control law, the tracking errors need to be defined. In particular, since the modified attitude reference is computed dynamically by the controller, all the quantities that have a dependency from the desired attitude or the desired angular speed need to be redefined and are called with the adjective "modified".
The desired attitude matrix R_d is changed with the modified desired attitude matrix R_{dc} (shown in the next subsection) and, as a consequence, the left attitude error in SO(3) shown in equation (3.3) becomes the modified attitude error in

SO(3)

$$R_e^{dc} = R_{dc} R_d^T. \tag{3.33}$$

Recalling the equation (3.9), the attitude error in $\mathbb{R}^3$ is defined as

$$e_R = \text{skew}(K_R R_e)^\vee \tag{3.34}$$

and the modified attitude error in $\mathbb{R}^3$ is

$$e_R^{dc} = \text{skew}(K_R^{dc} R_e^{dc})^\vee \tag{3.35}$$

where K_R^{dc} a suitable diagonal gain matrix. The modified angular velocity error is

$$e_\omega^{dc} = \omega_b - \omega_{dc} \tag{3.36}$$

where ω_{dc} is the modified desired angular speed expressed in the body frame. The error navigation function shown in (3.7) becomes the modified error navigation function:

$$\Psi_{dc} = \frac{1}{2}\text{tr}(K_R^{dc}(I - R_e^{dc})). \tag{3.37}$$

On the contrary, position and velocity errors resolved to the inertial frame are the same shown in equations (3.1) and (3.2).

3.2.3 Reference attitude computation

The control system receives as input the desired orientation matrix $R_d \in \text{SO}(3)$ defined in equation (3.4). Since the desired orientation may be not compatible with the position tracking, the modified reference orientation $R_{dc} \in \text{SO}(3)$ is computed in order to be feasible and as close as possible to the desired attitude. In particular, the matrix of the modified desired attitude R_{dc} is computed thanks to the composition of two rotation matrices:

$$R_{dc} = R_c R_r \tag{3.38}$$

where R_c is the rotation matrix required to track the trajectory in the under-actuated (co-planar) case, hence R_r is a relative rotation matrix. In particular,

$$R_c = \begin{bmatrix} b_{c_1} & b_{c_2} & b_{c_3} \end{bmatrix} \tag{3.39}$$

and is selected among the set of rotations around the axis

$$b_{c_3} = -\frac{f_c^d}{||f_c^d||} \tag{3.40}$$

where

$$f_c^d = -K_x e_x - K_v e_v + m \left(\ddot{p}_d - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \right). \tag{3.41}$$

Then b_{c_2} and b_{c_1} are computed as

$$b_{c_2} = \frac{b_{c_3} \times b_{d_1}}{||b_{c_3} \times b_{d_1}||} \tag{3.42}$$

$$b_{c_1} = b_{c_2} \times b_{c_3}. \tag{3.43}$$

In order to compute the modified desired angular speed and accelerations, the first and second derivative of matrix R_c need to be computed. Therefore,

$$\dot{R}_c = \begin{bmatrix} \dot{b}_{c_1} & \dot{b}_{c_2} & \dot{b}_{c_3} \end{bmatrix} \tag{3.44}$$

$$\ddot{R}_c = \begin{bmatrix} \ddot{b}_{c_1} & \ddot{b}_{c_2} & \ddot{b}_{c_3} \end{bmatrix} \tag{3.45}$$

where $\dot{b}_{c_1}, \dot{b}_{c_2}, \dot{b}_{c_3}, \ddot{b}_{c_1}, \ddot{b}_{c_2}, \ddot{b}_{c_3}$ computations are shown in Appendix A.
The kinematic evolution of the modified reference trajectory is computed from equation (3.38):

$$\dot{R}_{dc} = R_{dc}\hat{\omega}_{dc} \tag{3.46}$$

$$\omega_{dc} = R_r^T(\omega_c + \omega_r) \tag{3.47}$$

$$\dot{\omega}_{dc} = -R_r^T\hat{\omega}_r(\omega_c + \omega_r) + R_r^T(\dot{\omega}_c + \dot{\omega}_r) \tag{3.48}$$

where

$$\omega_c = (R_c^T\dot{R}_c)^\vee \tag{3.49}$$

$$\dot{\omega}_c = (R_c^T\ddot{R}_c - ((\omega_c)^\wedge)^2)^\vee. \tag{3.50}$$

The matrix

$$R_r = \begin{bmatrix} b_{r_1} & b_{r_2} & b_{r_3} \end{bmatrix} \tag{3.51}$$

is derived from the kinematics of the relative attitude motion

$$\dot{R}_r = \hat{\omega}_r R_r \tag{3.52}$$

where

$$\omega_r = b_{r_3} \times \dot{b}_{r_3} + (b_{r_3}^T\omega_r^d)b_{r_3} \tag{3.53}$$

$$\omega_r^d = R_r\omega_d - \omega_c - R_r R_d^T e_R^{dc}. \tag{3.54}$$

The third axis b_{r_3} of the differential equation (3.52) is modified as follows:

$$\dot{b}_{r_3}^p = \text{Proj}(b_{r_3}^p, \dot{b}_{r_3}^{pd}) \tag{3.55}$$

$$\dot{b}_{r_3}^{(3)} = \frac{(b_{r_3}^{(1)})^T\dot{b}_{r_3}^{(1)} + (b_{r_3}^{(2)})^T\dot{b}_{r_3}^{(2)}}{\sqrt{1 - (b_{r_3}^{(1)})^2 - (b_{r_3}^{(2)})^2}} \tag{3.56}$$

where

$$b_{r_3}^p = \begin{bmatrix} b_{r_3}^{(1)} \\ b_{r_3}^{(2)} \end{bmatrix} \tag{3.57}$$

is the projection of the vector b_{r_3} in the plane spanned by $\{b_{c_1}, b_{c_2}\}$. The parameter $\delta = \sin(\theta_M)$ defines the maximum value of $||b_{r_3}^p||$ in order to keep b_{r_3} inside the cone region and θ_M corresponds to the angle of the cone defined around the axis b_{c_3}.
The differential equation that describes the relative desired third body axis is

$$\dot{b}_{r_3}^d = \omega_r^d \times b_{r_3} \tag{3.58}$$

whose first two elements are grouped in the vector

$$\dot{b}_{r_3}^{pd} = \begin{bmatrix} \dot{b}_{r_3}^{d(1)} \\ \dot{b}_{r_3}^{d(2)} \end{bmatrix}. \tag{3.59}$$

The projection operator adopted in equation (3.55) is defined as the function that smoothly removes the radial components of $\dot{b}_{r_3}^p$, keeping the modulus of $b_{r_3}^p$ within the maximum admissible value according to the cone region constraint.
In particular (see Lavretsky and Wise [2013]),

$$\text{Proj}(b_{r_3}^p, \dot{b}_{r_3}^{pd}, f) = \begin{cases} \dot{b}_{r_3}^{pd} - f(b_{r_3}^p) \frac{\Gamma \nabla f(b_{r_3}^p) \nabla f(b_{r_3}^p)^T}{||\nabla f(b_{r_3}^p)||_\Gamma^2} \dot{b}_{r_3}^{pd} & \text{if} f(b_{r_3}^p) > 0 \wedge \left(\dot{b}_{r_3}^{pd}\right)^T \nabla f(b_{r_3}^p) > 0 \\ \dot{b}_{r_3}^{pd} & \text{otherwise} \end{cases} \tag{3.60}$$

where $f(b_{r_3}^p)$ is a convex continuously differentiable function and $\nabla : \mathbb{R} \to \mathbb{R}^2$ is the gradient operator. Since the projection operator defined in equation (3.60) is continuos but not differentiable, the relative angular acceleration $\dot{\omega}_r$ cannot be defined and used in the feedforward term of equation (3.48). The function $f(b_{r_3}^p)$ is defined as

$$f(b_{r_3}^p) = \frac{(1+\varepsilon)||b_{r_3}^p||^2 - \delta^2}{\varepsilon \delta^2} \tag{3.61}$$

with $\varepsilon \in (0, 1)$ the constant that defines the smoothness of the projection operator. It can be proven that, when

$$||b_{r_3}^p|| \in \left[\frac{\delta}{\sqrt{1+\varepsilon}}, \delta\right], \tag{3.62}$$

the projection operator starts to work enabling a smooth transition towards the bound δ. Hence, the b_{r_3} axis is kept inside the cone defined by the angle θ_M around b_{c_3} assuring that

$$\cos(\theta) \geq \cos(\theta_M). \tag{3.63}$$

Finally, when

$$||b_{r_3}^p|| < \frac{\delta}{\sqrt{1+\varepsilon}}, \tag{3.64}$$

it follows that

$$\dot{b}_{r_3} = \omega_r^d \times b_{r_3} = \dot{b}_{r_3}^d, \tag{3.65}$$

which in turn implies that

$$\omega_r = \omega_r^d \tag{3.66}$$

matching exactly the desired angular velocity and, as a consequence, the desired orientation.

3.2.4 Control law

Considering the control law shown in equations (3.26), (3.17), (3.18), the control law for the geometric controller with modified reference attitude is:

$$f_c = c(\Psi_{dc}) R_{dc}^T f_c^d \tag{3.67}$$

$$\tau_c = -R_{dc}^T e_R - K_\omega e_\omega^{dc} + I\dot{\omega}_{dc} + \omega_{dc} \times I\omega_b \tag{3.68}$$

where the ideal control action expressed in the inertial frame is defined as

$$f_c^d = -K_p e_p - K_v e_v + m(\dot{v}_d - g e_3) \tag{3.69}$$

and the function $c(\Psi_{dc})$ is

$$c(\Psi_{dc}) = \frac{\Psi_M - \Psi_{dc}}{\Psi_M}. \tag{3.70}$$

As in Subsection 3.2.1, the scaling function $c(\Psi_{dc})$ scales the control force computed in equation (3.67) by a term that depends by the modified error navigation function Ψ_{dc} in order to keep the delivered control force f_c inside the cone region defined around the axis b_3. In practice, when the attitude error becomes high, *i.e.*, when the modified error navigation function Ψ_{dc} is close to its maximum value Ψ_M, the value of the scaling function becomes $c(\Psi_{dc}) \ll 1$ and the computed control force is scaled, letting the platform to follow the position tracking prior to the attitude tracking. This leads to reduce the position overshoot.

3.2.5 Simulation results

In order to test the capabilities of the presented geometric controller with modified reference attitude, numerical simulations are performed. In particular, the same trajectories of the previous Subsections 2.2.3, 3.1.3 are tested: the polynomial and the eight shape trajectory. Additionally, in order to appreciate the action of the projection operator that limits the tilting actuation, an eight shape trajectory

Parameter	Value
K_p	16 I_3
K_v	4 I_3
K_R	0.75 I_3
K_ω	0.1875 I_3
Ψ_M	40
K_R^{dc}	$2K_R$
ε	0.1
θ_M	30°
Γ	10 I_2

Table 3.2: Control parameters of the geometric controller with modified reference attitude

with a superimposed roll sinusoidal movement of $\pm 30°$ and θ_M limited to 15° is tested. The control parameters adopted in these simulations are shown in Table 3.2.

Polynomial trajectory

Figures 3.11, 3.12, 3.13 show the behavior of the system, controlled thanks to the geometric controller.

Eight-shape trajectory

Figures 3.14, 3.15, 3.16, 3.17, 3.18 show the simulation of the tilt-rotor performing an eight-shape trajectory. The simulations highlight that the behavior of the controlled system is really close to the one shown in Subsection 3.1.3. This means that when the trajectory to be tracked is smooth and feasible, the two geometric controllers presented in this thesis behaves in a similar way.

Eight-shape trajectory with limited θ_M

The parameter θ_M limits the region in which the thrust vector can lie, but imposing the angle θ_M is not equivalent to limit the maximum servo-motors tilting angles. In fact, the angle θ_M is used to define the angle of the cone region, in which the vector f_c lies, defined around the third body axis b_3. The thrust vector f_c is close to the vertical inertial axis when the platform is near to hovering conditions, but if the translational tracking requires great forces in order to be performed, the thrust vector can assume great angles with respect to the vertical inertial axis.
In the simulation hereby described, the controlled system is asked to track an

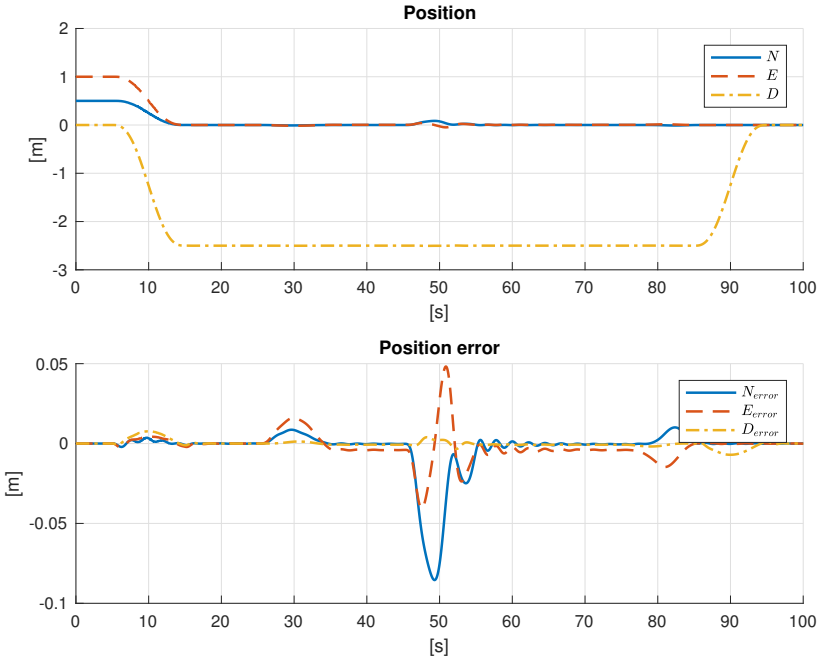

Figure 3.11: Geometric control with modified reference attitude: position and position error in polynomial trajectory simulation

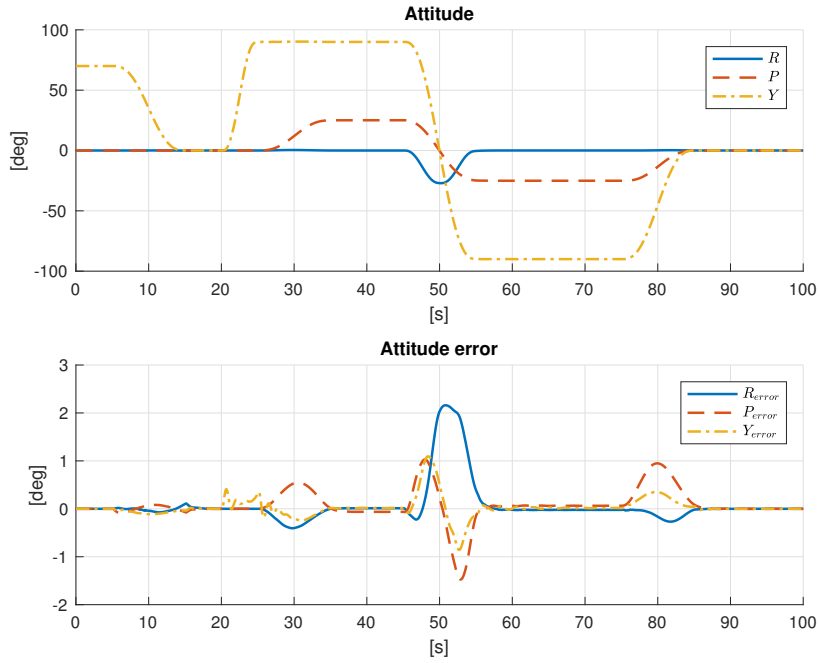

Figure 3.12: Geometric control with modified reference attitude: attitude and attitude error in polynomial trajectory simulation

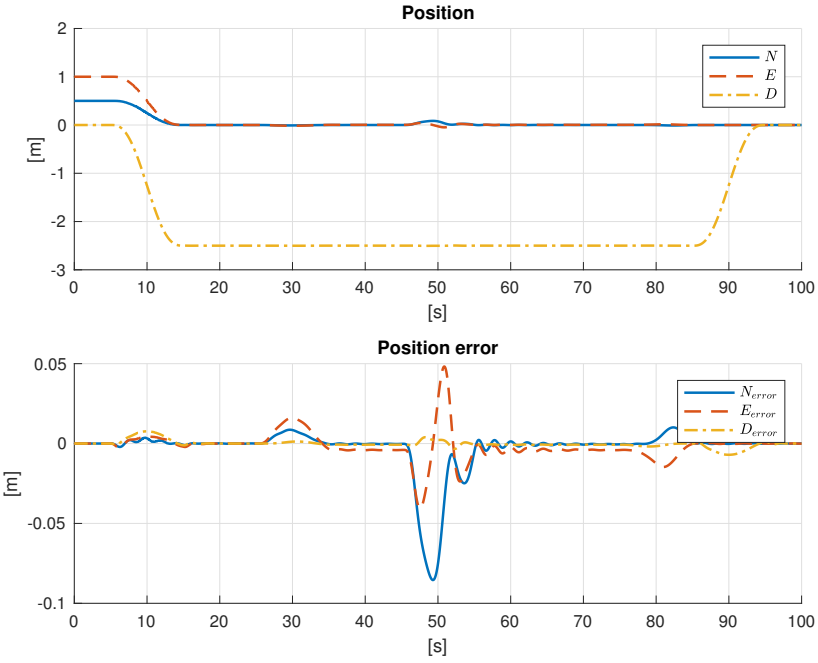

Figure 3.13: Geometric control with modified reference attitude: throttle percentages and tilting angles in polynomial trajectory simulation

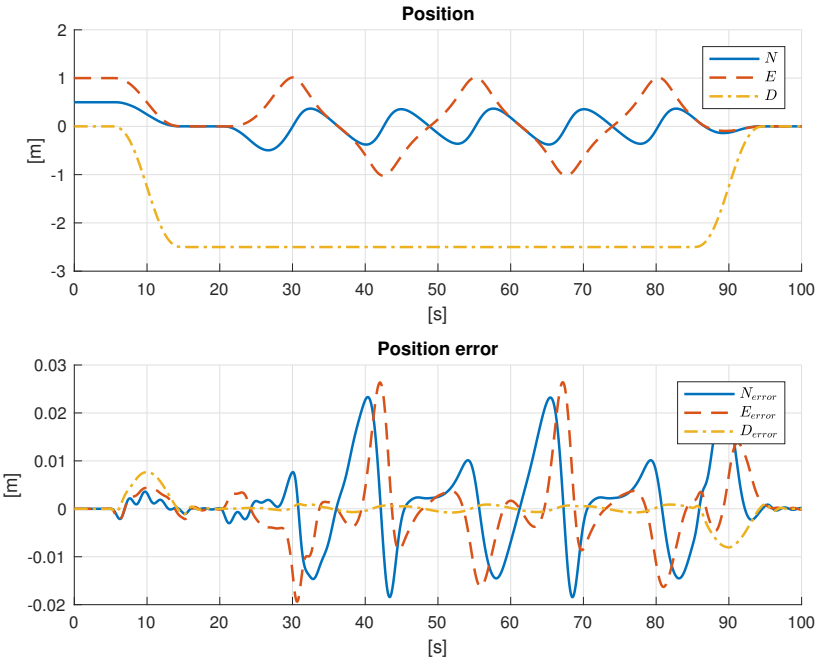

Figure 3.14: Geometric control with modified reference attitude: position and position error in eight-shape trajectory simulation

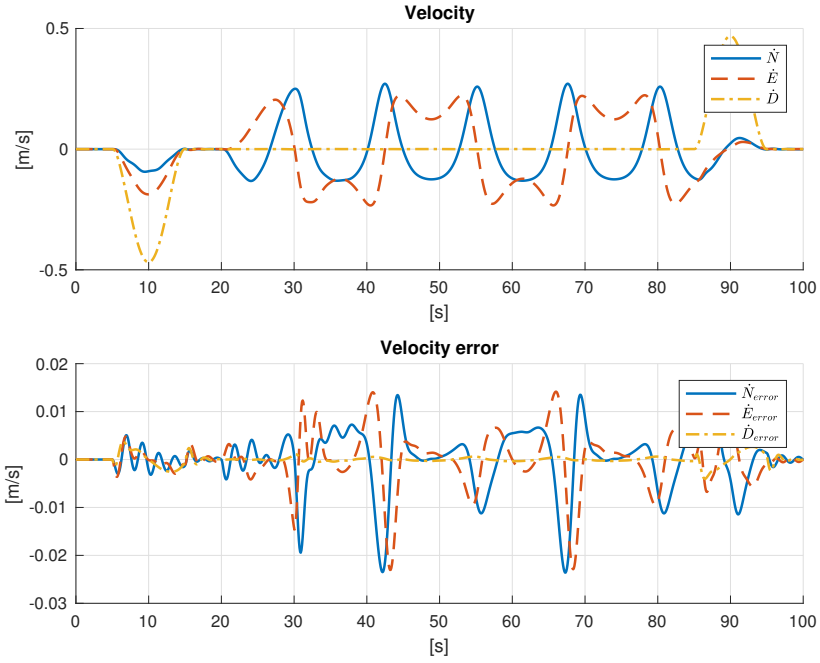

Figure 3.15: Geometric control with modified reference attitude: velocity and velocity error in eight-shape trajectory simulation

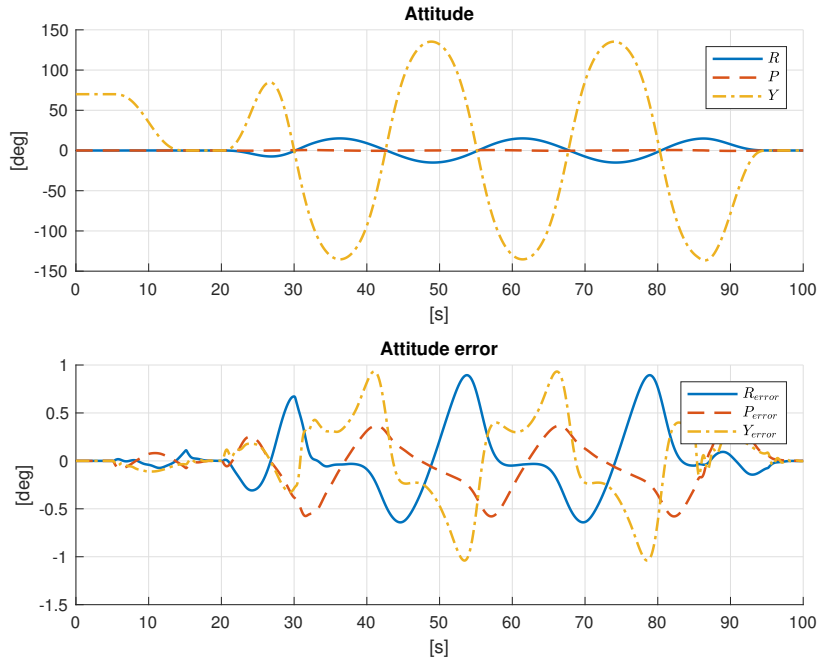

Figure 3.16: Geometric control with modified reference attitude: attitude and attitude error in eight-shape trajectory simulation

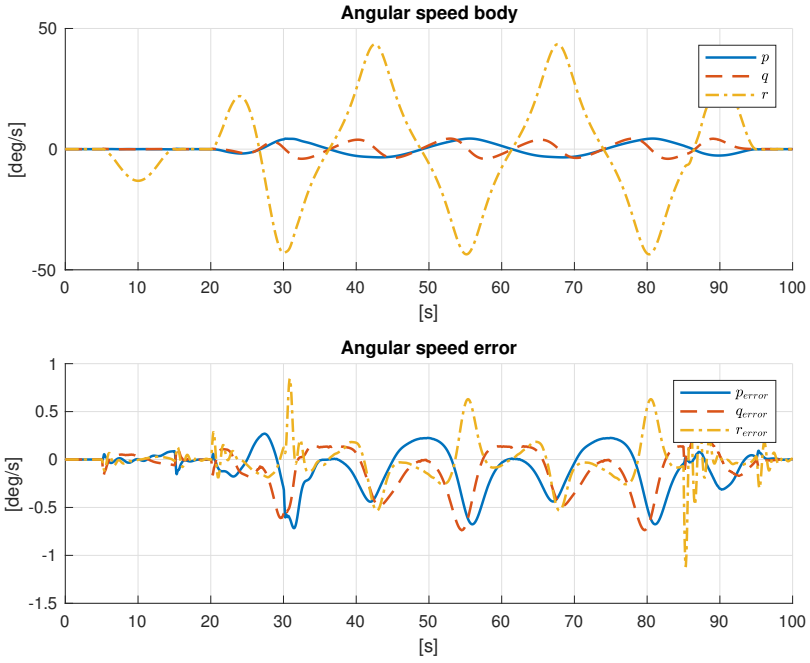

Figure 3.17: Geometric control with modified reference attitude: angular speed and error in eight-shape trajectory simulation

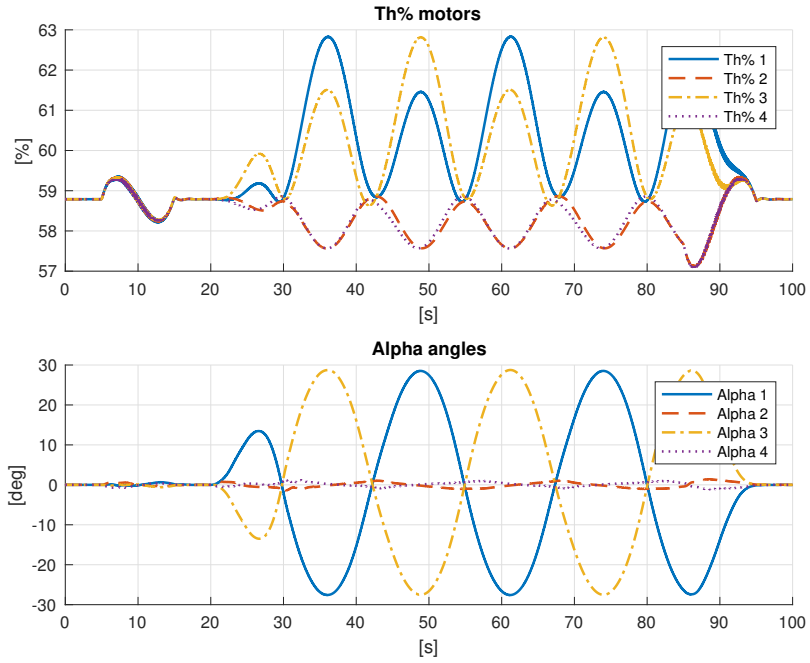

Figure 3.18: Geometric control with modified reference attitude: throttle percentages and tilting angles in eight-shape trajectory simulation

eight-shape trajectory with a superimposed sinusoidal roll movement of ±30°, while limiting $\theta_M = 15°$. Since the trajectory is performed at low speeds, the required translational forces are small and impose this limitation is almost equivalent to ask to the control system not to tilt the platform over the angle θ_M.
Looking to Figures 3.21, 3.22 and 3.23, it is clearly visible the action of the pro-

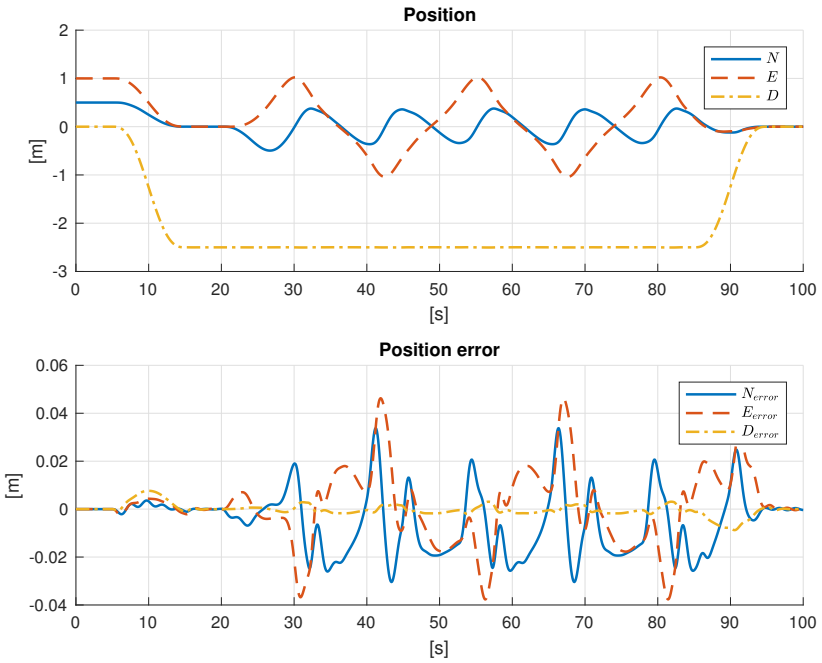

Figure 3.19: Geometric control with modified reference attitude: position and position error in eight-shape trajectory simulation with limited $\theta_M = 15°$

jection operator: when the desired roll angle is greater than $\theta_M = 15°$, it keeps the vector b^p_{r3} inside the cone region, constraining the modified desired attitude. It also visible that the servo-motors tilting angles reaches their maximum values of around 30°, that is greater than θ_M. This, as already introduced, is due to the fact that there is not a direct relation between the servo-motors maximum tilting angles and the cone dimension since the angle θ_M only defines the cone angle. It is noticed that, in the performed simulations, imposing $\theta_M = 15°$ implies that the servo-motors tilting angles remain in the limit of around ±30°.
Looking to Figures 3.19 and 3.20, the position and velocity simulation results are similar to the non limited case, where the projection operator do not act, and there are no great differences between the desired attitude R_d and the modified desired attitude R_{dc} computed by the controller.

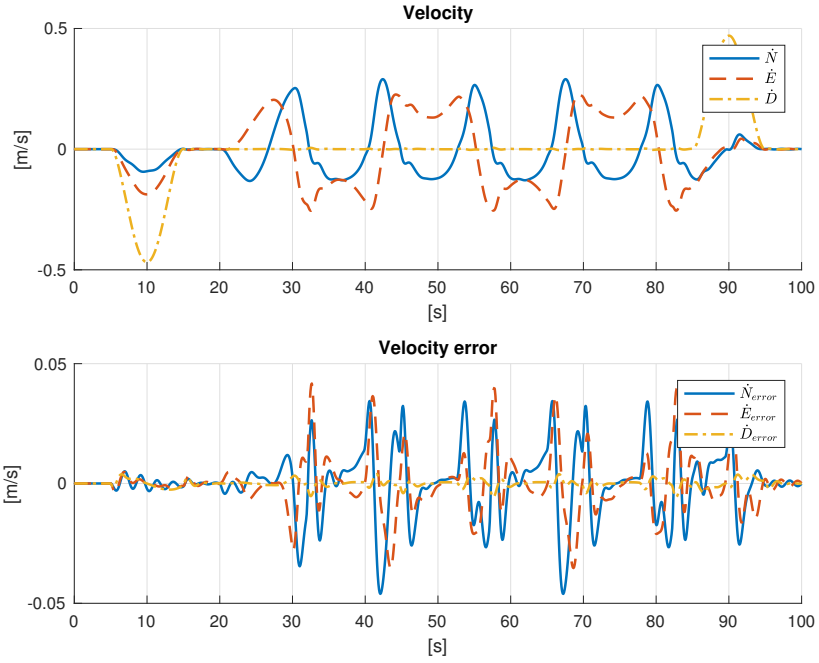

Figure 3.20: Geometric control with modified reference attitude: velocity and velocity error in eight-shape trajectory simulation with limited $\theta_M = 15°$

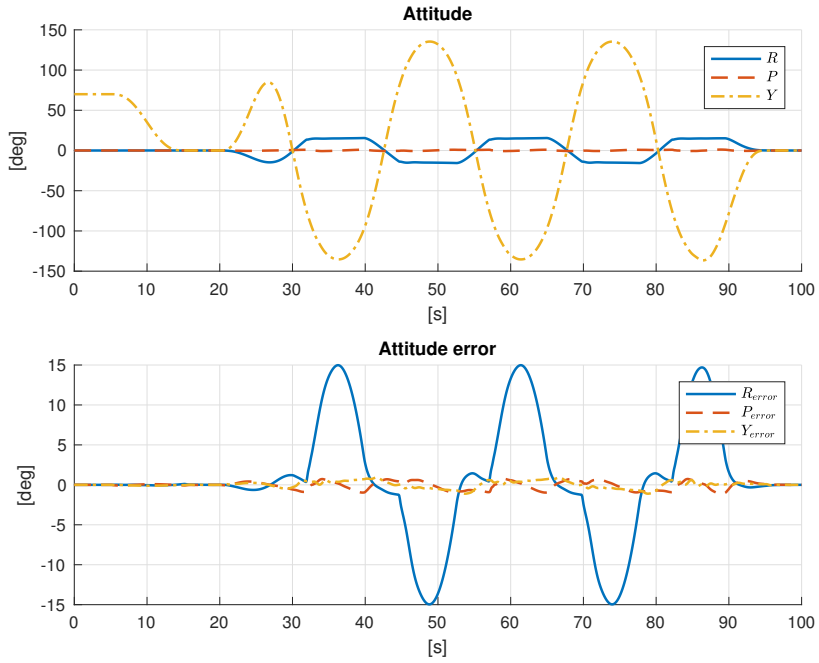

Figure 3.21: Geometric control with modified reference attitude: attitude and attitude error in eight-shape trajectory simulation with limited $\theta_M = 15°$

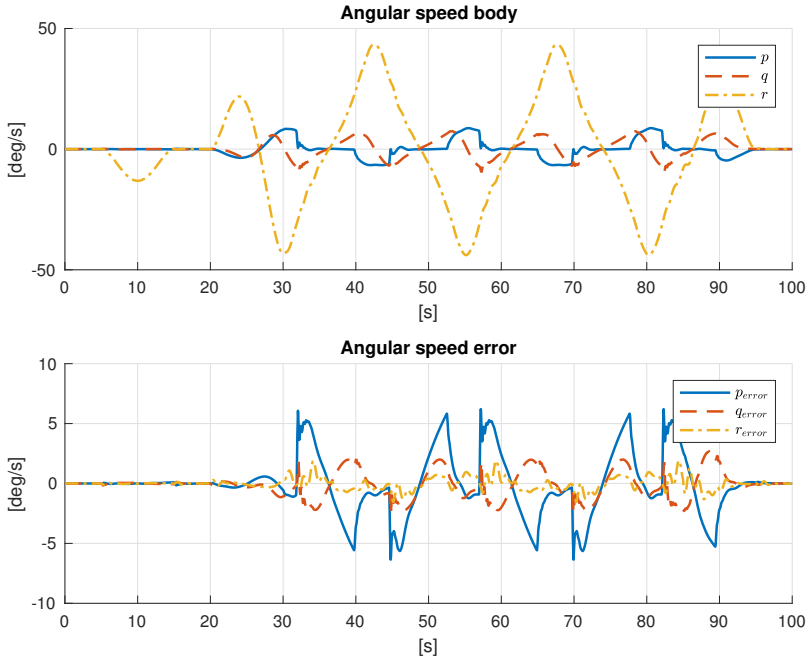

Figure 3.22: Geometric control with modified reference attitude: angular speed and error in eight-shape trajectory simulation with limited $\theta_M = 15°$

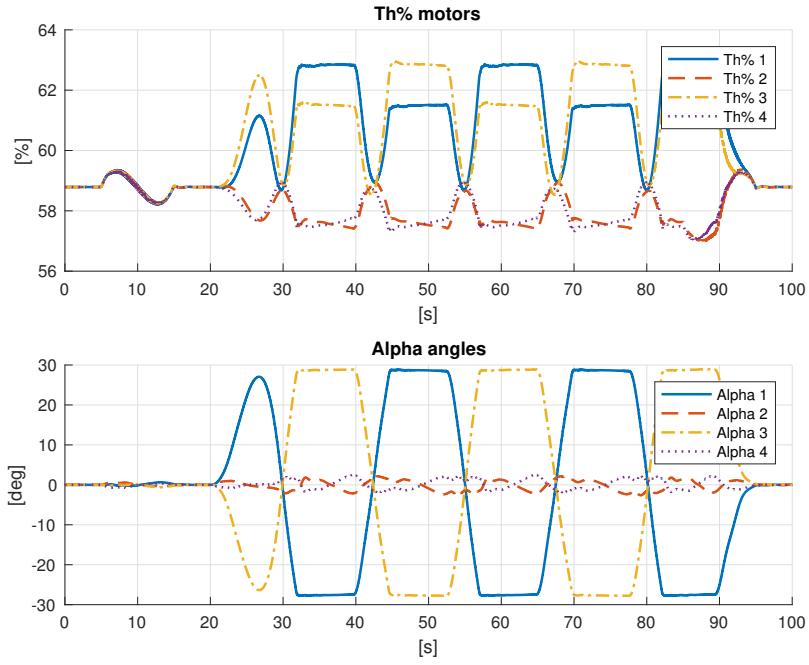

Figure 3.23: Geometric control with modified reference attitude: throttle percentages and tilting angles in eight-shape trajectory simulation with limited $\theta_M = 15°$

3.2.6 Considerations

When the desired trajectories are smooth and feasible, the platform behaves very similarly with both the geometric controllers presented in this chapter. This means that the computation of the modified reference attitude do not affect the overall system until the required thrust vector reaches the limit of the predefined cone region.
In Chapter 1 the body frame is defined according to NED standard but, for the sake of simplicity and clarity, in this section all the pictures and the rationales are shown adopting the vertical body axis pointing upward. This is due to better understanding of the physical meaning of the presented control laws, that would otherwise be difficult to visualize. All the computations are in NED frame.

Chapter 4

Robustness analysis

In Chapter 1 the dynamic model of the tilt-rotor quad-copter is shown underlying the assumptions and the approximations made. The physical quantities of the tilt-rotor used to characterize this model are inherited from Micheli [2016]. Shown in Table 1.3, these quantities are in some cases directly measured, *e.g.*, the mass and the arm length, in other cases are estimated thanks to some identification campaigns. Their values, in particular for the estimated ones, are intrinsically uncertain and the control system must be able to work properly even with these uncertainties.
The robustness is the property of a controller, designed for a particular set of parameters, to work well even if under a different set of parameters or in presence of disturbances. In this thesis, the tilt-rotor body inertia tensor is the main parameter that is considered uncertain, of which a variation may cause important effects on the overall controlled system. In the following, the robustness of the controllers presented in Sections 2.2, 3.1 and 3.2 is evaluated simulating the controlled system with a perturbed inertia tensor. Additionally, the controllers capability to reject measurement disturbances by adding estimated white noises to the simulation feedbacks is analyzed. Finally, the controllers capability to reject external constant disturbances is evaluated.
The first controller presented in Section 2.1 is not analyzed in this chapter. This choice arises from the fact that, as already discussed in Subsection 2.1.7, this controller presents several numerical issues.

4.1 Simulations with measurement noise

The implementation on Simulink platform of the overall controlled system has been presented in Section 1.4. Hereby, in order to perform simulations that are as close as possible to the reality, measurement noise is added to the feedback. This way the feedback signals are not only discretized through a Zero Order Hold (ZOH) at $100Hz$, but have also noisy components, in order to simulate the non-idealities of the Inertial Measurement Unit (IMU). Table 4.1 shows the estimated standard deviation of the measurement noise, modeled as a white noise disturbance.
In order to assess the effects of the measurement noise on the controlled system,

Quantity measured	Standard deviation	Unit
n	0.0011	m
e	0.0009	m
d	0.0013	m
$\dot{n}$	0.001	m/s
$\dot{e}$	0.001	m/s
$\dot{d}$	0.01	m/s
ϕ	0.0076	rad
θ	0.0118	rad
ψ	0.0115	rad
p	0.1	rad/s
q	0.1	rad/s
r	0.1	rad/s

Table 4.1: Estimated standard variation of measurement noise

the eight-shape trajectory presented in Subsection 2.2.3 is tested along with the controllers. For the sake of clarity, the position error, velocity error, attitude error, angular speed error and the control input signals are plotted in order to compare them with the results obtained in Chapters 2 and 3. The plots of position, velocity, attitude and angular speed are not shown since the effects of the measurement noise is not appreciable in those plots with big scale factors.

Feedback linearization

The feedback linearization controller presented in Section 2.2 is capable to perform the tracking of the eight-shape trajectory. The behavior of the controlled system subject to measurement noise is shown in Figures 4.1, 4.2, 4.3, 4.4, 4.5, where it is visible that the controller do not completely rejects the measurement noise. In particular, in velocity error and angular speed error plots (Figures 4.2 and 4.4) are noticed the presence of undesirable oscillations, that in Figures 2.16 and 2.18, *i.e.*, without measurements noise, are present but with a significantly smaller

magnitude. This effect can be explained recalling the fact that the feedback linearization controller is not a really robust control strategy. Figure 4.5 shows the control input signals: due to the effect of measurement noise and the signal discretization, the throttle percentages commands of the motors become more oscillatory, without destabilizing the system.
In this thesis, this controller is the one on which the measurement noise has the worst impact.

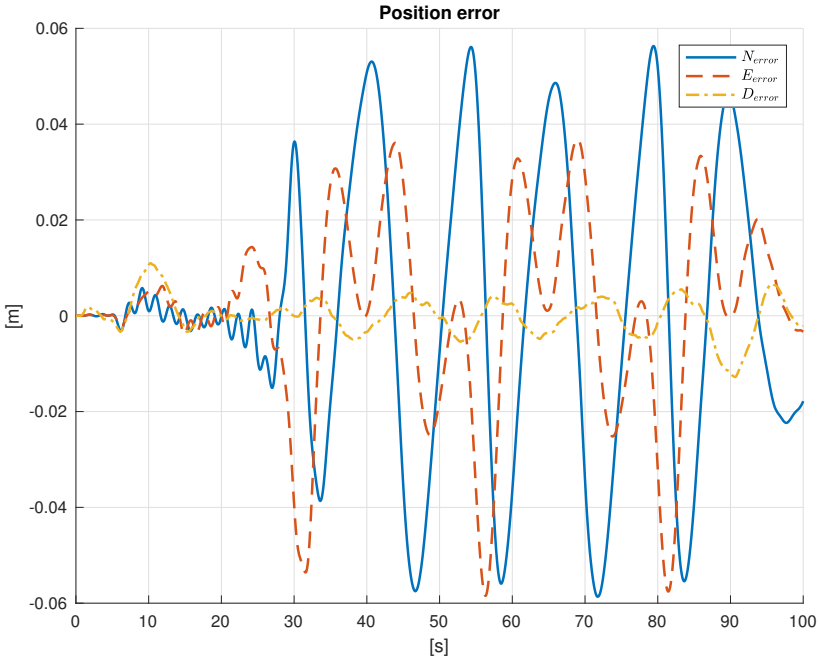

Figure 4.1: Feedback linearization: position error in eight-shape trajectory simulation with measurement noise

Geometric control for the fully actuated case

The geometric controller, presented in Section 3.1, is capable to reject the measurement noise. The behavior of the system is shown in Figures 4.6, 4.7, 4.8, 4.9, 4.10. In particular, the position and orientation tracking are performed as well, while it is noticeable small and acceptable oscillations in velocity and angular speed plots. As for the feedback linearization controller, the throttle percentages commands of the motors have a fast oscillatory behavior, caused by the measurement noise and the feedback signal discretization.

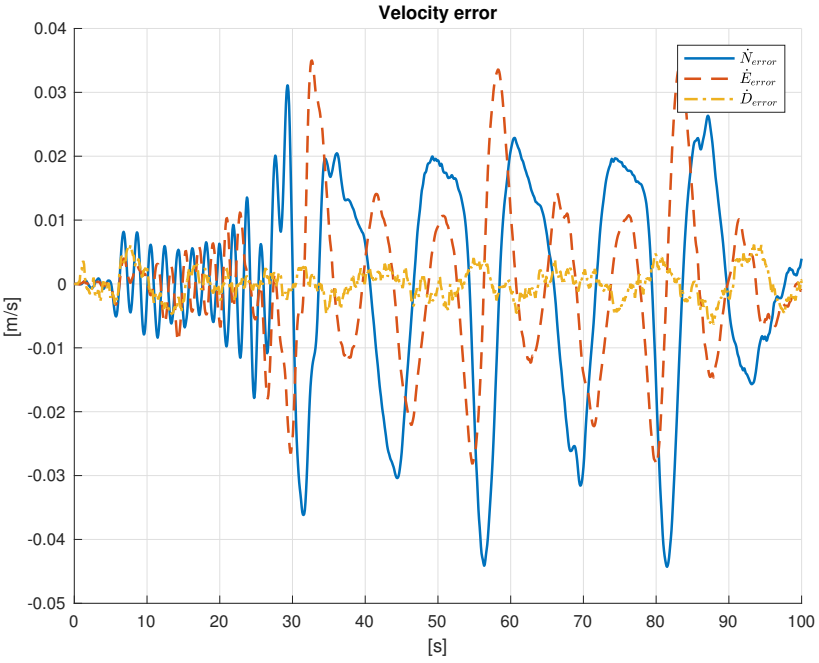

Figure 4.2: Feedback linearization: velocity error in eight-shape trajectory simulation with measurement noise

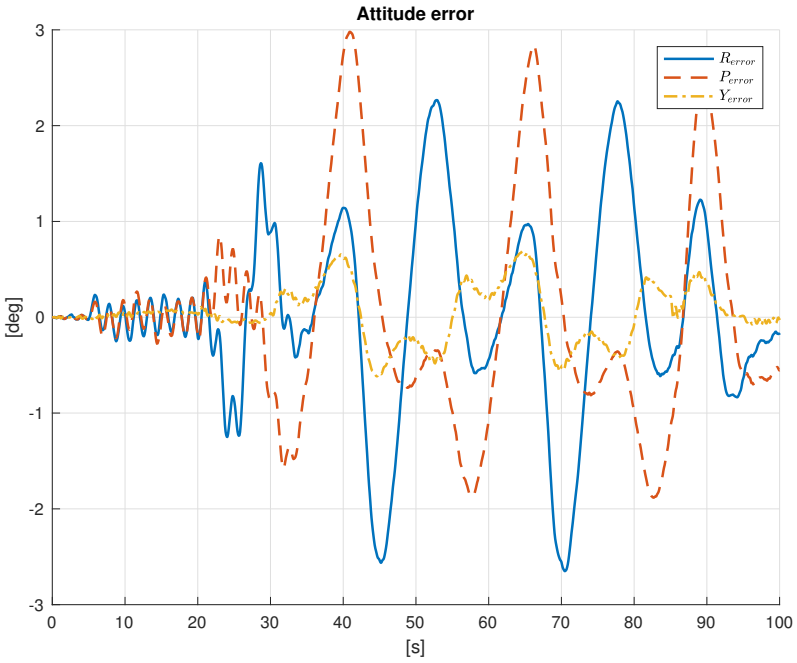

Figure 4.3: Feedback linearization: attitude error in eight-shape trajectory simulation with measurement noise

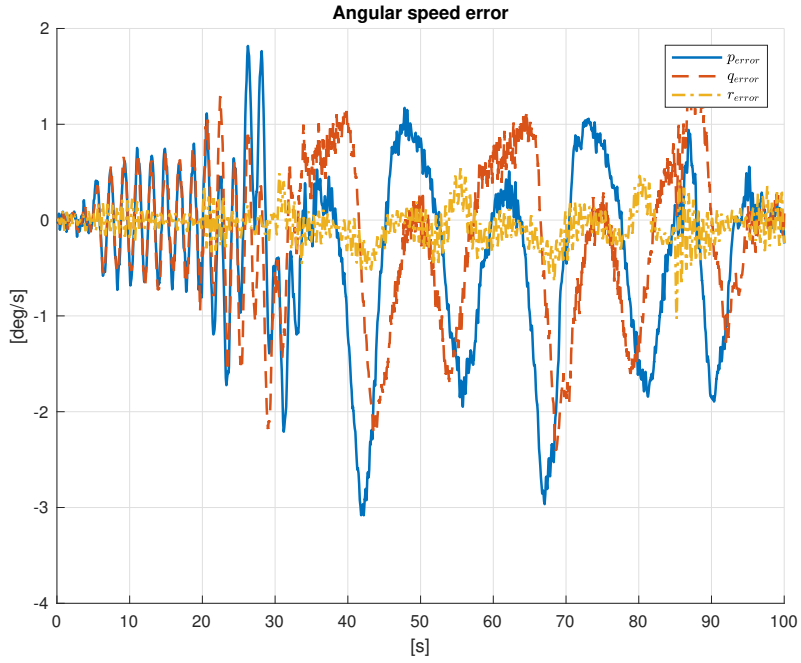

Figure 4.4: Feedback linearization: Angular speed error in eight-shape trajectory simulation with measurement noise

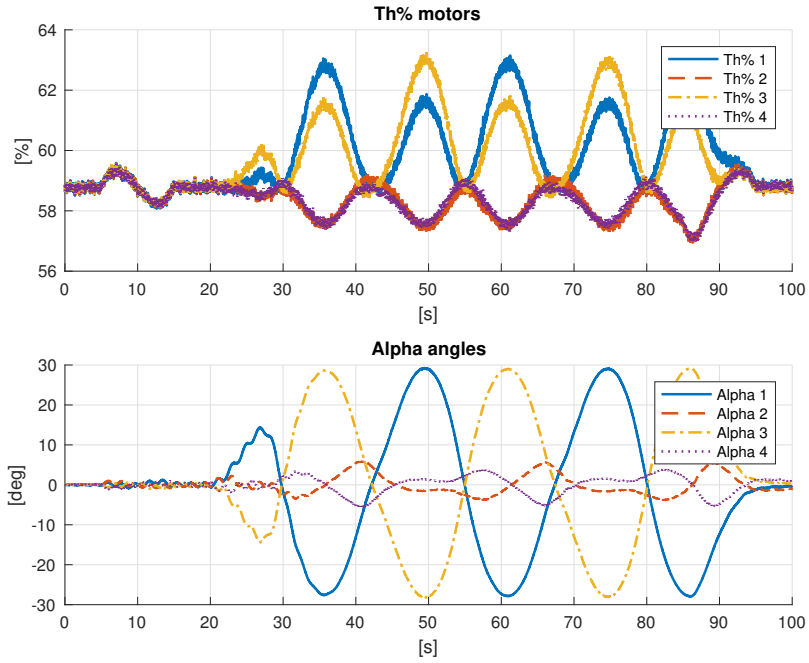

Figure 4.5: Feedback linearization: throttle percentages and tilting angles in eight-shape trajectory simulation with measurement noise

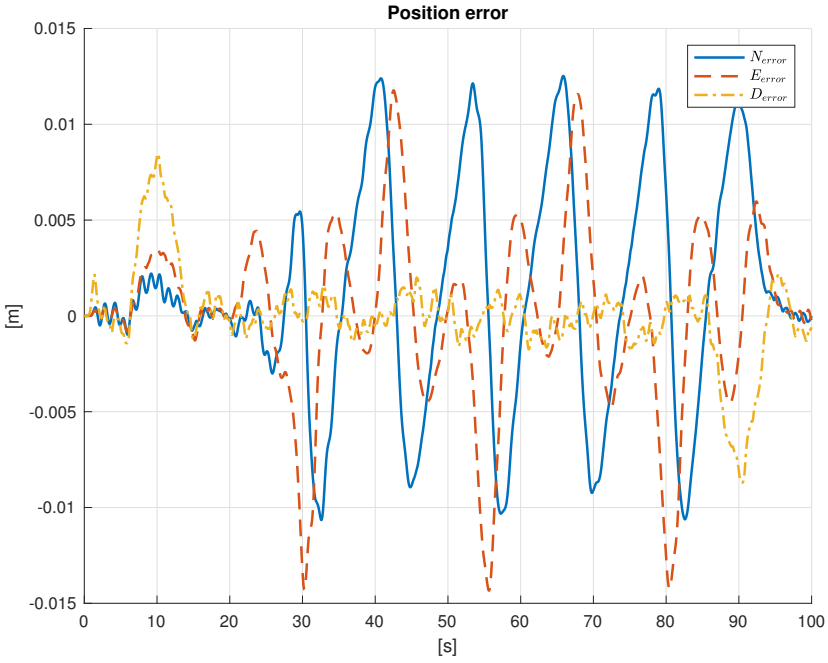

Figure 4.6: Geometric control: position error in eight-shape trajectory simulation with measurement noise

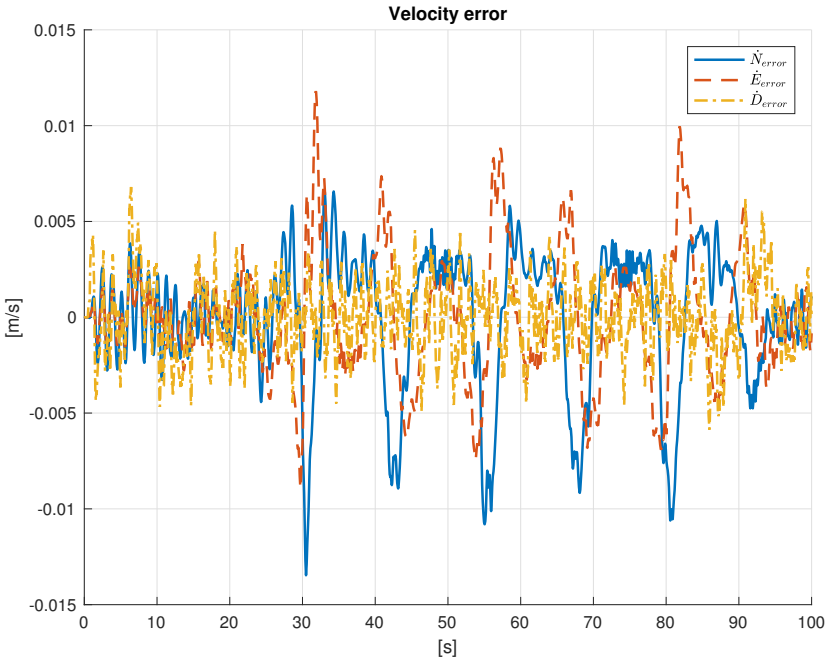

Figure 4.7: Geometric control: velocity error in eight-shape trajectory simulation with measurement noise

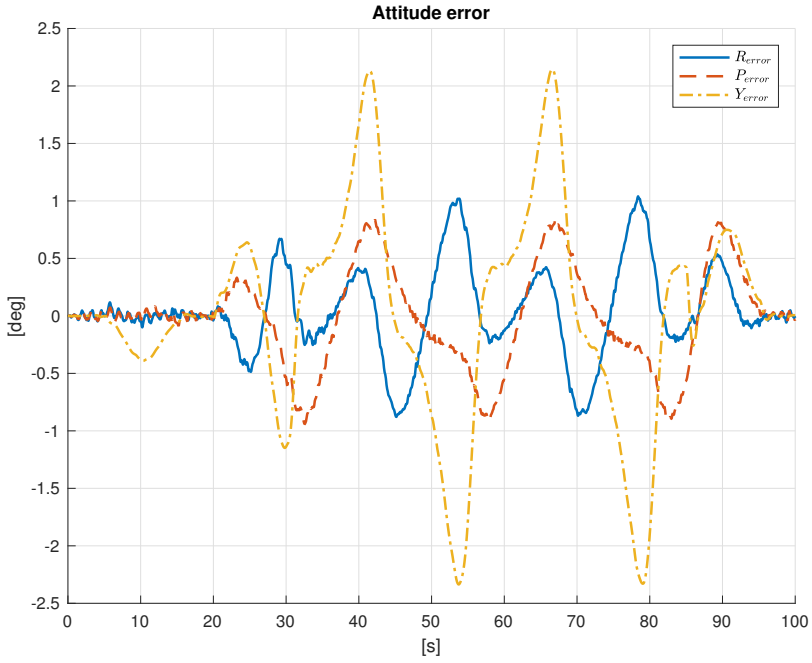

Figure 4.8: Geometric control: attitude error in eight-shape trajectory simulation with measurement noise

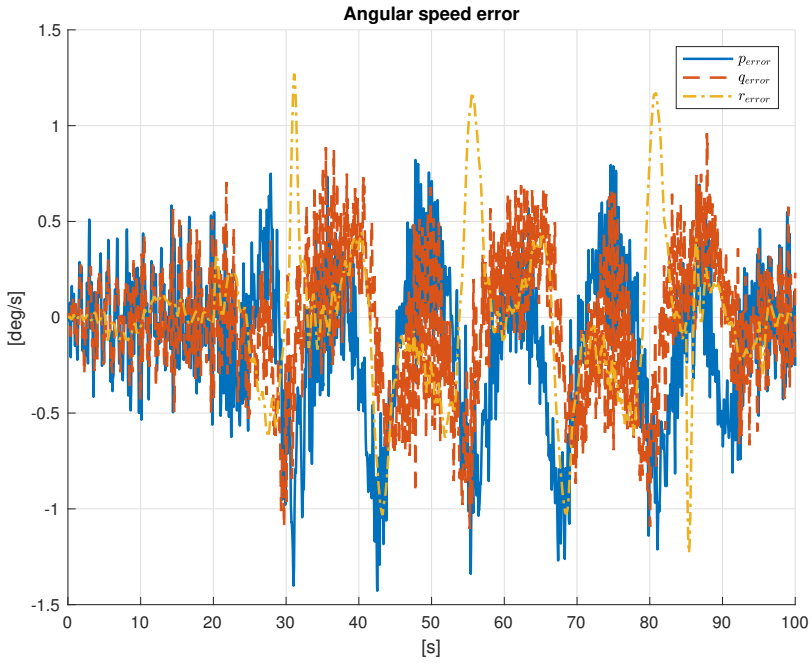

Figure 4.9: Geometric control: Angular speed error in eight-shape trajectory simulation with measurement noise

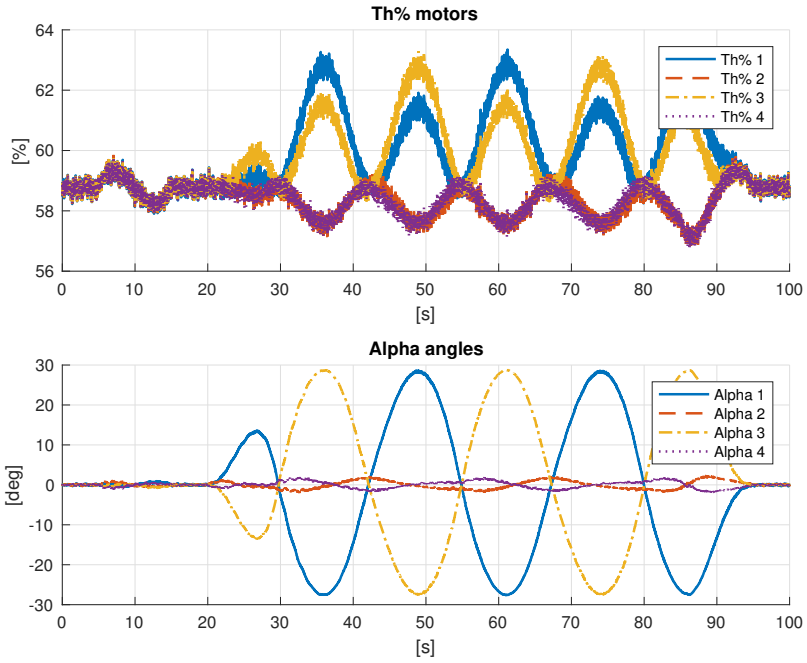

Figure 4.10: Geometric control: throttle percentages and tilting angles in eight-shape trajectory simulation with measurement noise

Geometric control with dynamic reference attitude

The geometric controller with dynamic reference attitude has almost the same behavior of the standard geometric controller. It is visible in Figures 4.11, 4.12, 4.13, 4.14, 4.15, and the same considerations can be done.

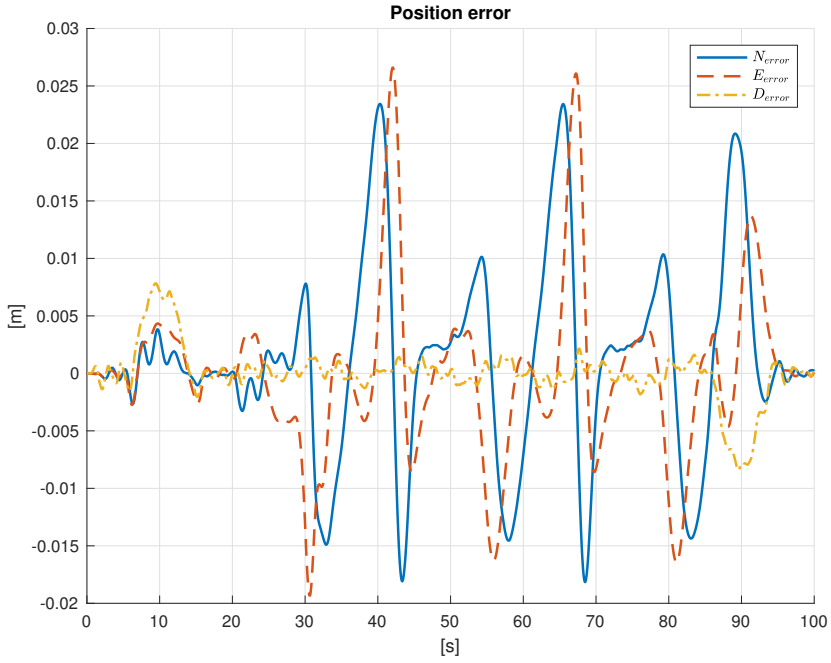

Figure 4.11: Geometric control with modified attitude reference: position error in eight-shape trajectory simulation with measurement noise

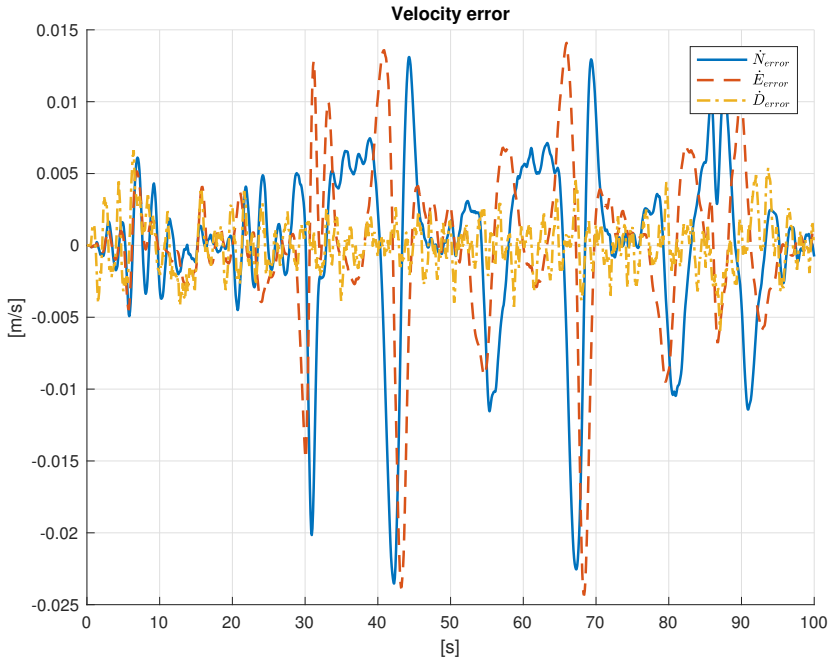

Figure 4.12: Geometric control with modified attitude reference: velocity error in eight-shape trajectory simulation with measurement noise

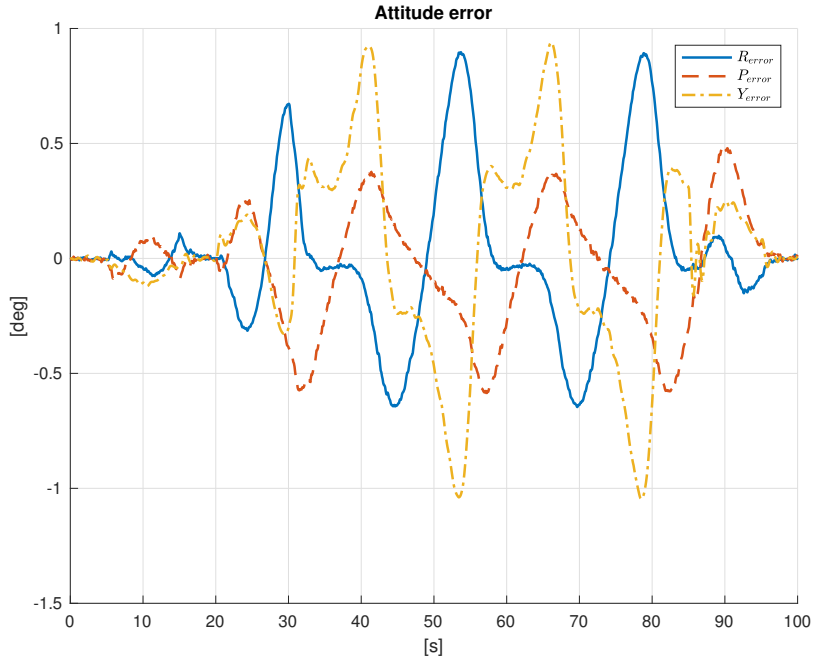

Figure 4.13: Geometric control with modified attitude reference: attitude error in eight-shape trajectory simulation with measurement noise

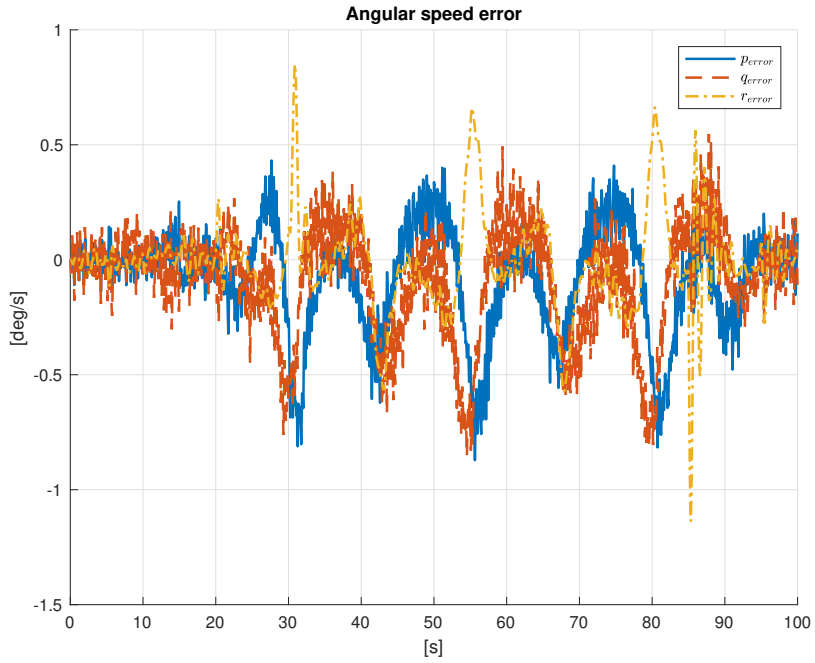

Figure 4.14: Geometric control with modified attitude reference: Angular speed error in eight-shape trajectory simulation with measurement noise

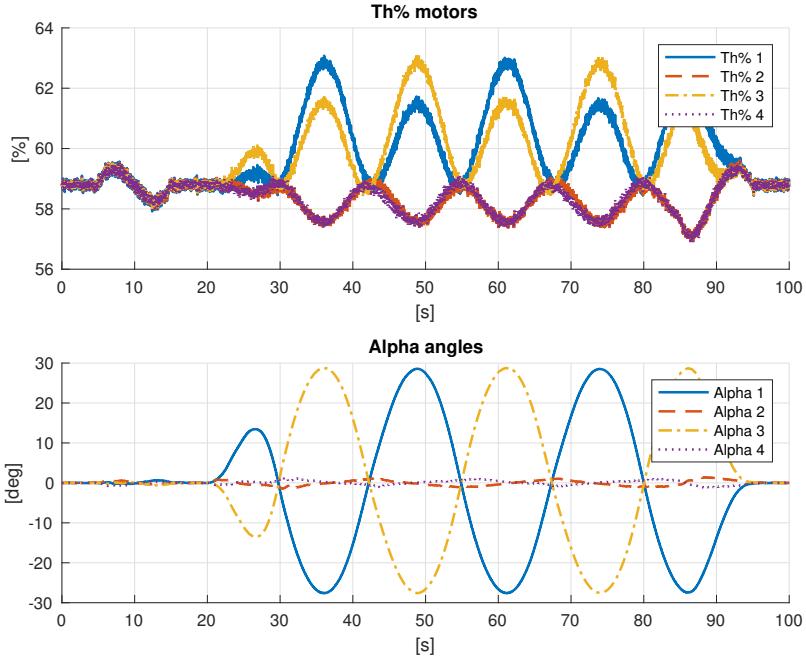

Figure 4.15: Geometric control with modified attitude reference: throttle percentages and tilting angles in eight-shape trajectory simulation with measurement noise

4.2 Parametric uncertainty

As already introduced, the sensitivity analysis of the controllers presented in Sections 2.2, 3.1, 3.2 with respect to the body inertia tensor is evaluated. The body inertia tensor, defined in equation 1.46 as

$$I_b = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix},$$

is perturbed in order to verify the effects of its variation on the controlled system. Thanks to the intrinsic symmetry of the tilt-rotor structure, the body inertia tensor is considered as a diagonal matrix and the only parameters that are subject of this sensitivity study are those on the diagonal.
In order to define a unique test case useful for all the studied controllers, three arrays (P_{10}, P_{15}, P_{20}), each containing twenty perturbed body inertia tensors, have been created: the parameters I_{xx}, I_{yy} and I_{zz} are perturbed around their nominal values (that are shown in Table 1.3), adopting a uniform probability density function, of

- $\pm 10\%$ in P_{10}
- $\pm 15\%$ in P_{15}
- $\pm 20\%$ in P_{20}.

Finally, each controller is tested performing sixty simulations, twenty per each perturbation array of the body inertia tensor. All the perturbations used for the simulations are shown in Appendix B.
In the following are not shown all the plots of the simulations since it would lead to show an excessive number of figures, but only the most significant plots per each controller are presented.

Feedback linearization

The feedback linearization controller proposed in Section 2.2 is sensitive to body inertia tensor variations. Important oscillations are present when the body inertia tensor I_b is perturbed of $\pm 15\%$ around its nominal value. Figures 4.16, 4.17, 4.18, 4.19, 4.20 show the behavior of the controlled system under twenty different body inertia tensors that span in the set of $\pm 15\%$ around its nominal value. The response is unsatisfactory, since big and fast oscillations are present with a significant amplitude.
The simulations performed with a body inertia tensor that varies of $\pm 20\%$ around its nominal values lead to a strongly oscillating and unacceptable response of the controlled system. For the sake of brevity, those plots are not shown since they do not represent meaningful results besides the instability of the system.

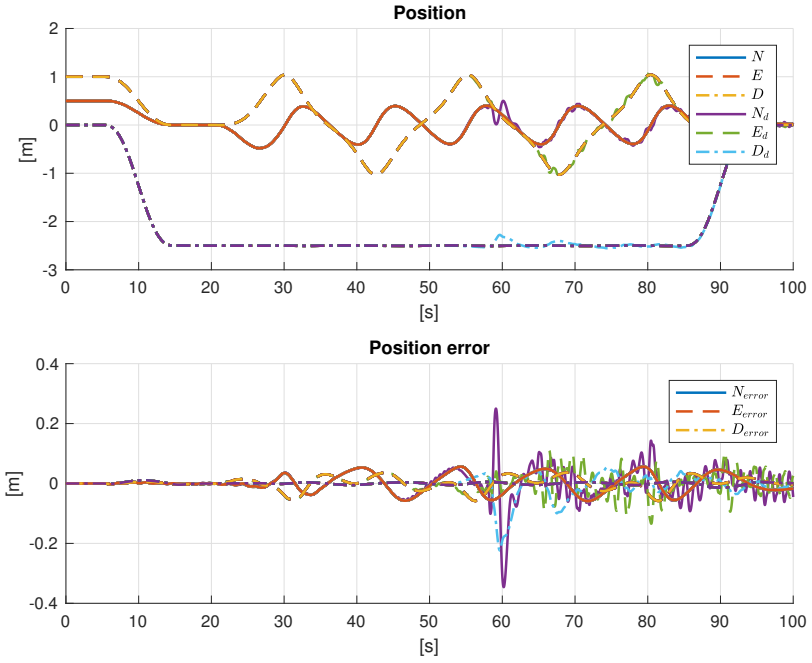

Figure 4.16: Feedback linearization control: position and position error in eight-shape trajectory simulation with ±15% variation of I_b

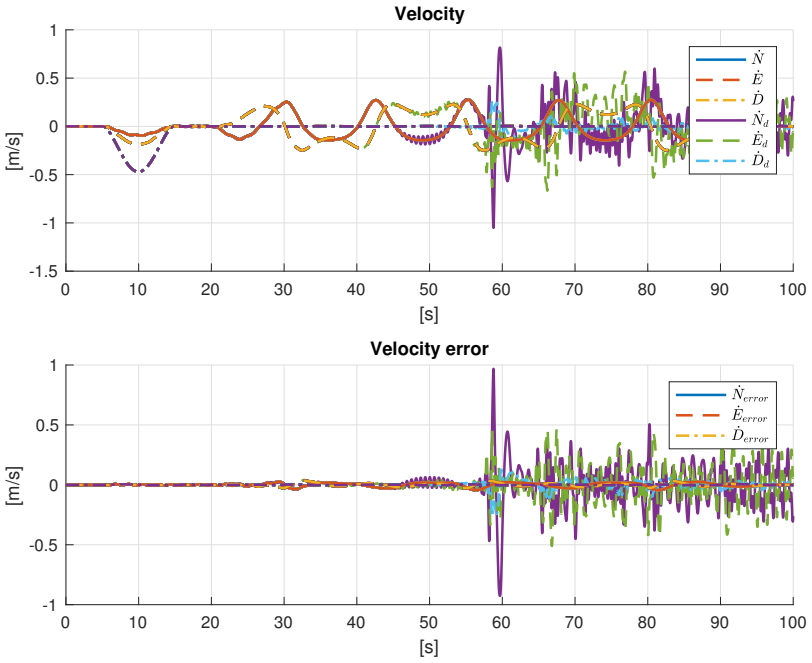

Figure 4.17: Feedback linearization control: velocity and velocity error in eight-shape trajectory simulation with ±15% variation of I_b

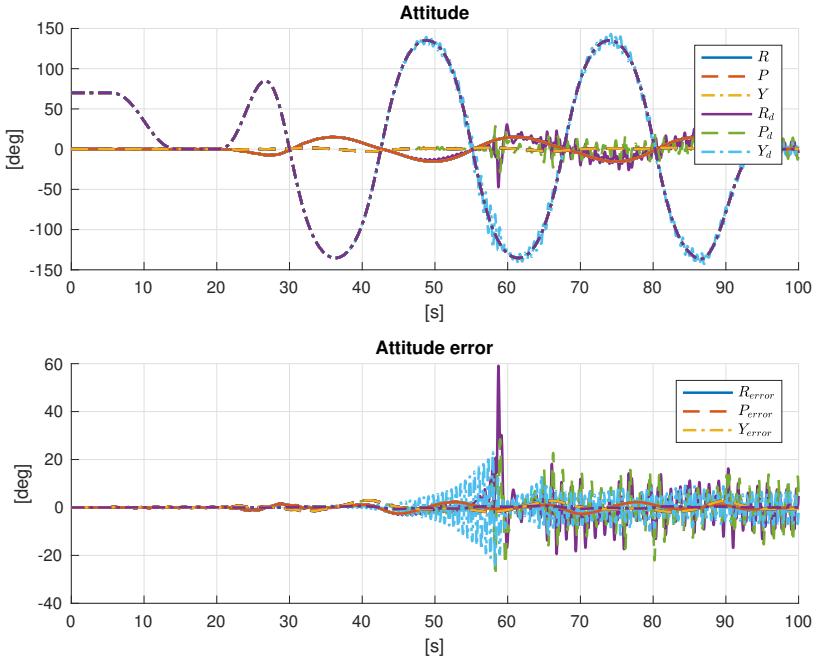

Figure 4.18: Feedback linearization control: attitude and attitude error in eight-shape trajectory simulation with ±15% variation of I_b

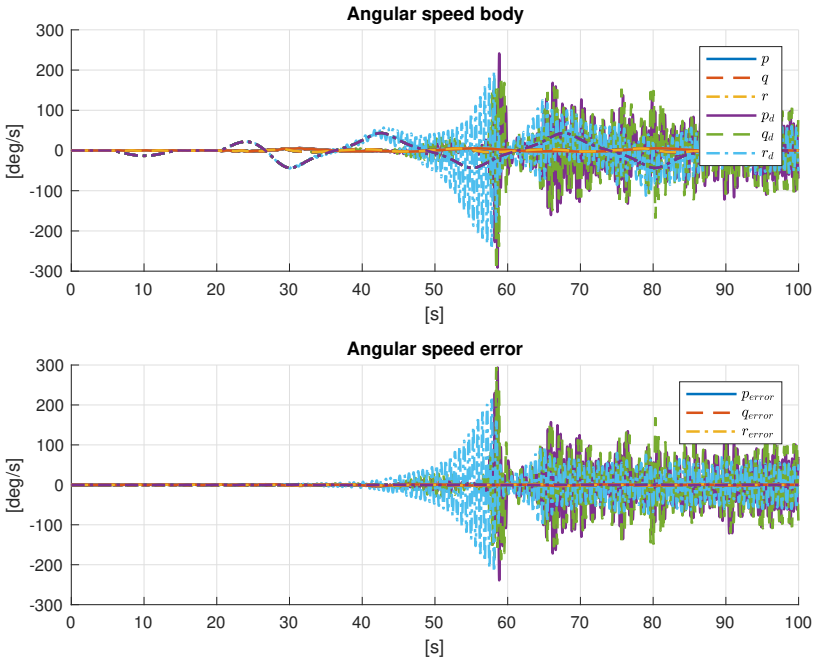

Figure 4.19: Feedback linearization control: Angular speed and error in eight-shape trajectory simulation with ±15% variation of I_b

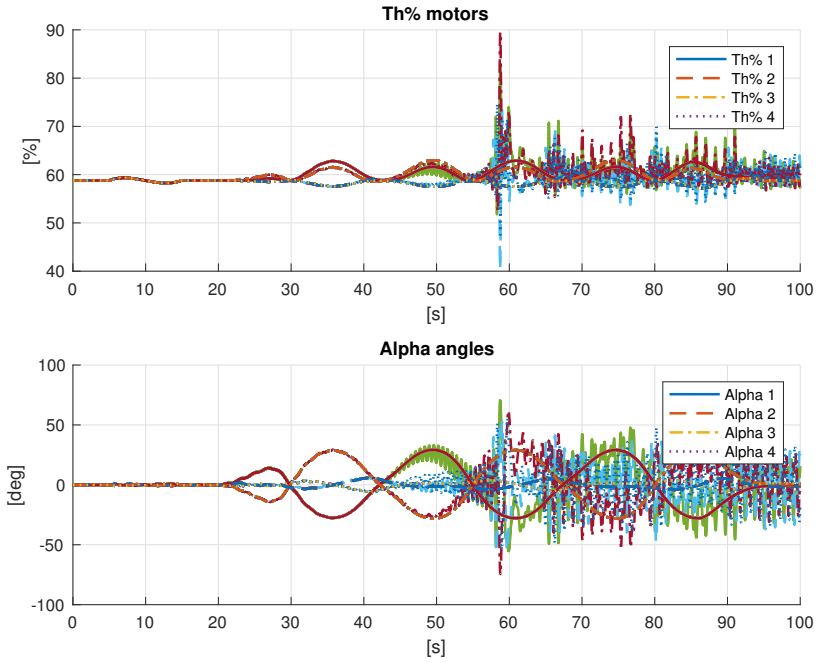

Figure 4.20: Feedback linearization control: throttle percentages and tilting angles in eight-shape trajectory simulation with ±15% variation of I_b

Geometric control for the fully actuated case

Figures 4.21, 4.22, 4.23, 4.24, 4.25 show the response of the system controlled by the first developed geometric controller (in Section 3.1). It is easy to see that, even if the variation of ±20% of I_b is considerable, the controlled system responds very well in all the plotted results. The geometric controller results to be, in fact, really robust with respect to parametric uncertainty of the body inertia tensor.

Geometric control with dynamic reference attitude

Similarly to the simplest geometric controller, the geometric controller with dynamic reference attitude responds very to well to parametric variation of ±20% of the body inertia tensor. Figures 4.26, 4.27, 4.28, 4.29, 4.30 show the behavior of the controlled system, that is really satisfactory and confirm that the geometric controllers developed in Chapter 3 are really robust with respect to parametric uncertainty of I_b.

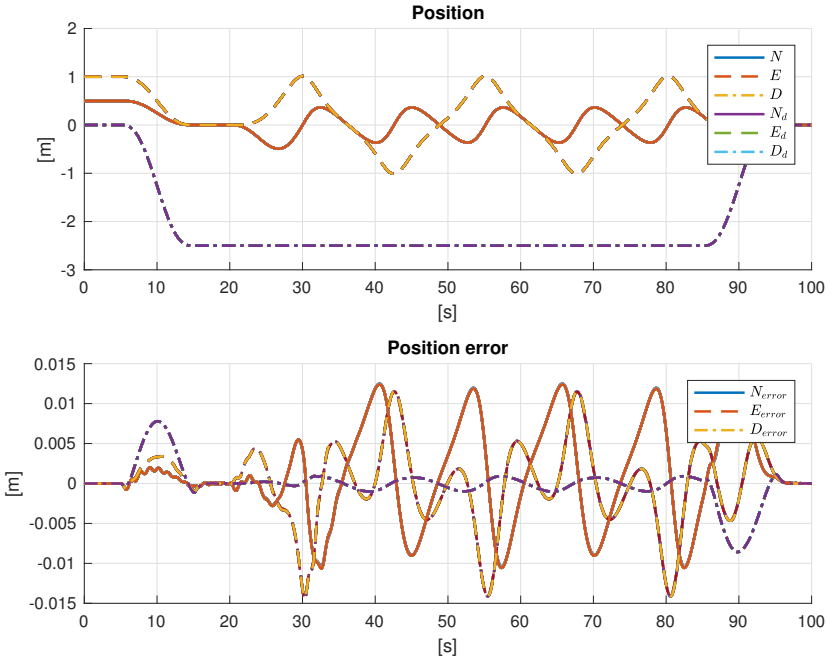

Figure 4.21: Geometric control: position and position error in eight-shape trajectory simulation with ±20% variation of I_b

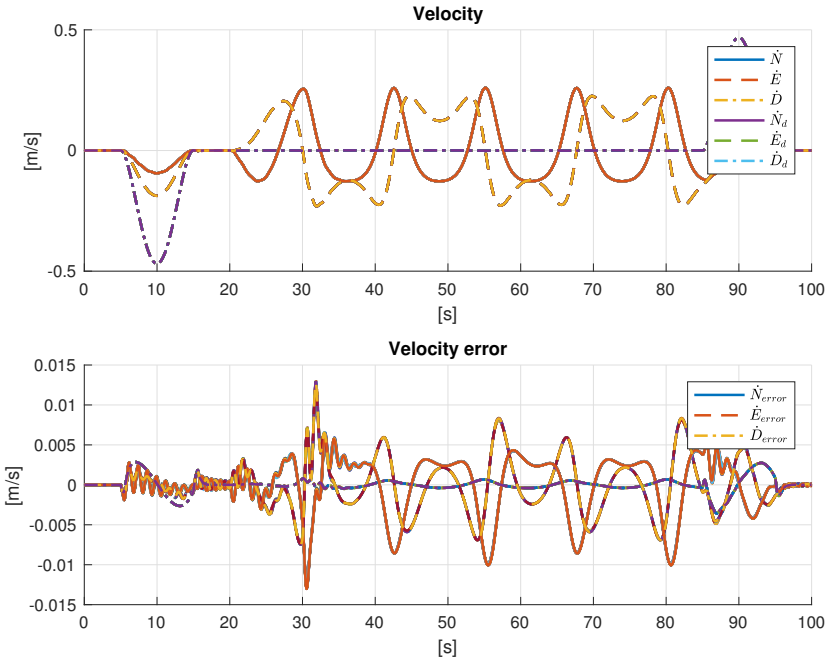

Figure 4.22: Geometric control: velocity and velocity error in eight-shape trajectory simulation with ±20% variation of I_b

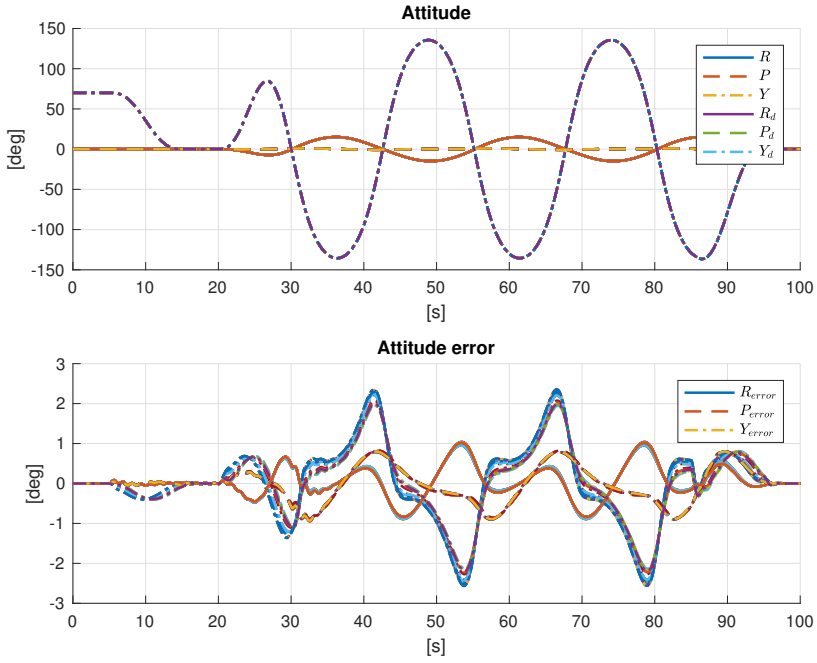

Figure 4.23: Geometric control: attitude and attitude error in eight-shape trajectory simulation with ±20% variation of I_b

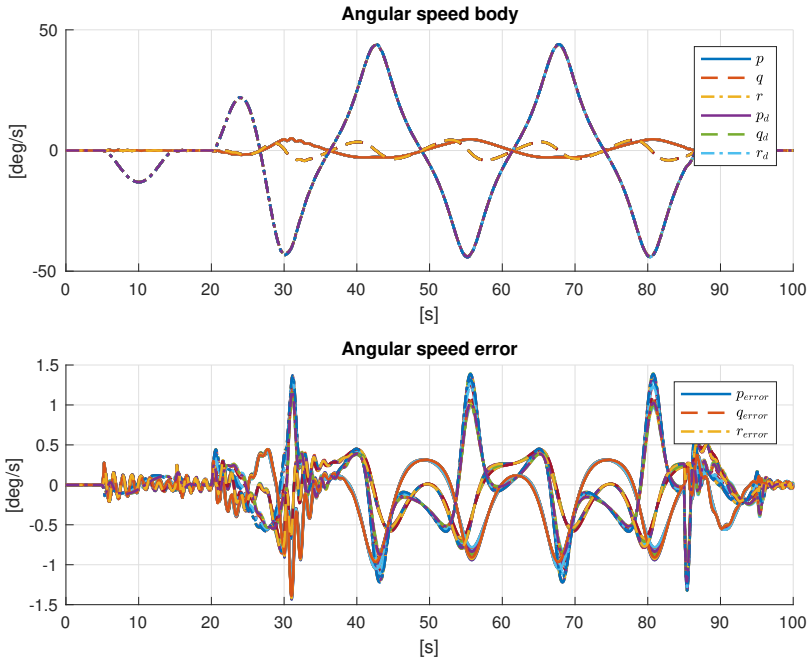

Figure 4.24: Geometric control: Angular speed and error in eight-shape trajectory simulation with ±20% variation of I_b

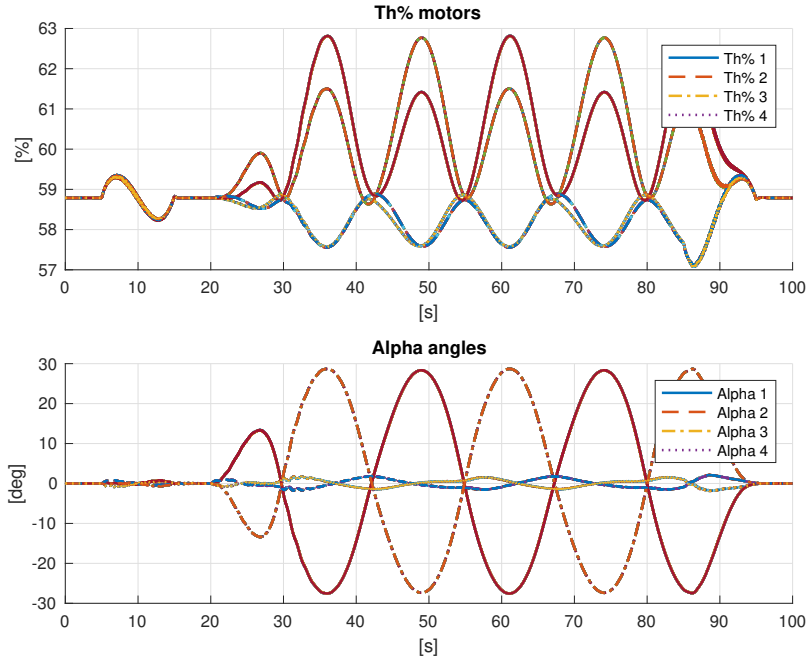

Figure 4.25: Geometric control: throttle percentages and tilting angles in eight-shape trajectory simulation with ±20% variation of I_b

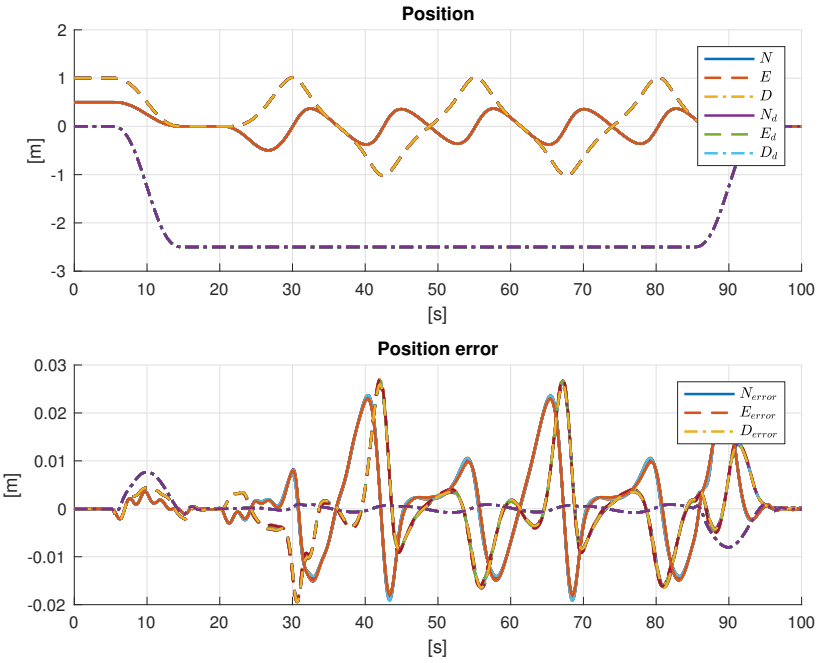

Figure 4.26: Geometric control with modified attitude reference: position and position error in eight-shape trajectory simulation with ±20% variation of I_b

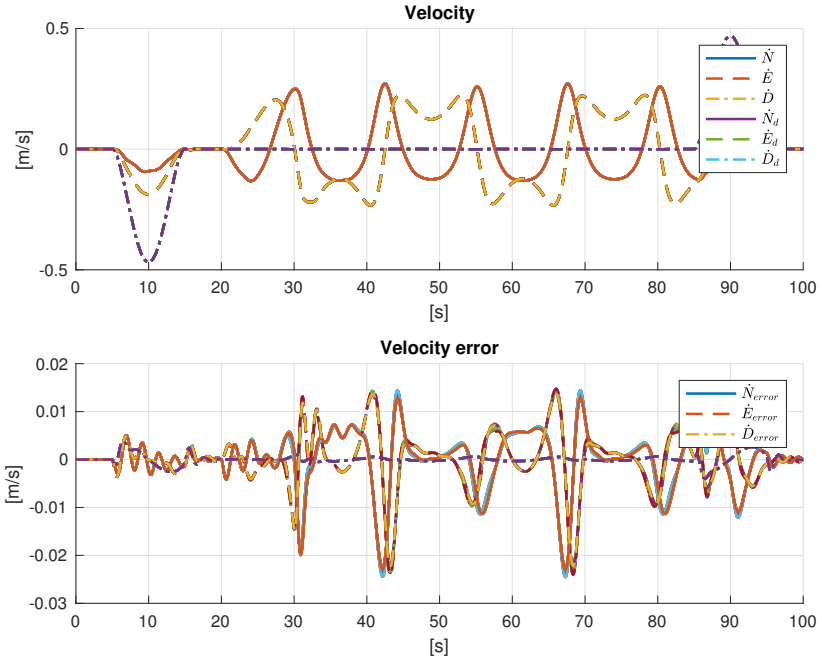

Figure 4.27: Geometric control with modified attitude reference: velocity and velocity error in eight-shape trajectory simulation with ±20% variation of I_b

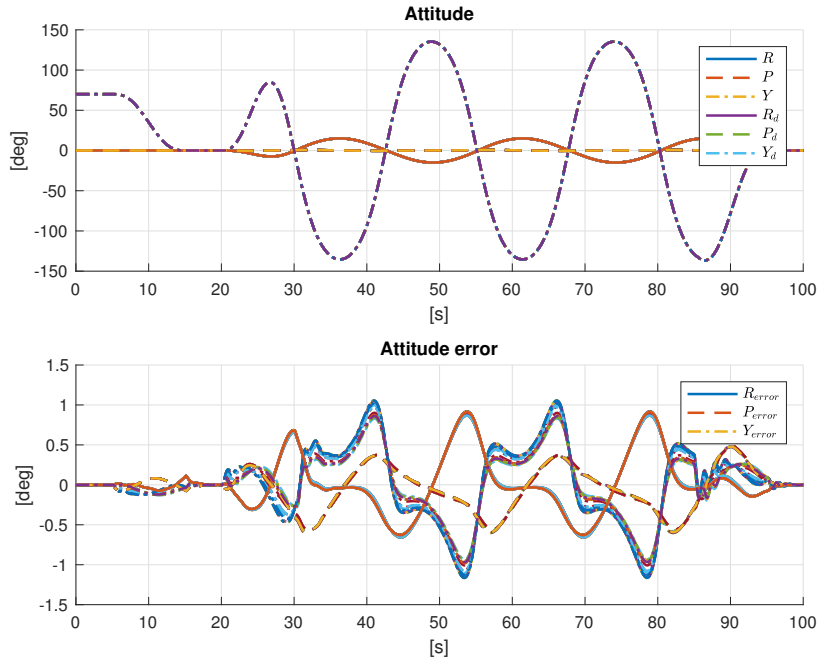

Figure 4.28: Geometric control with modified attitude reference: attitude and attitude error in eight-shape trajectory simulation with ±20% variation of I_b

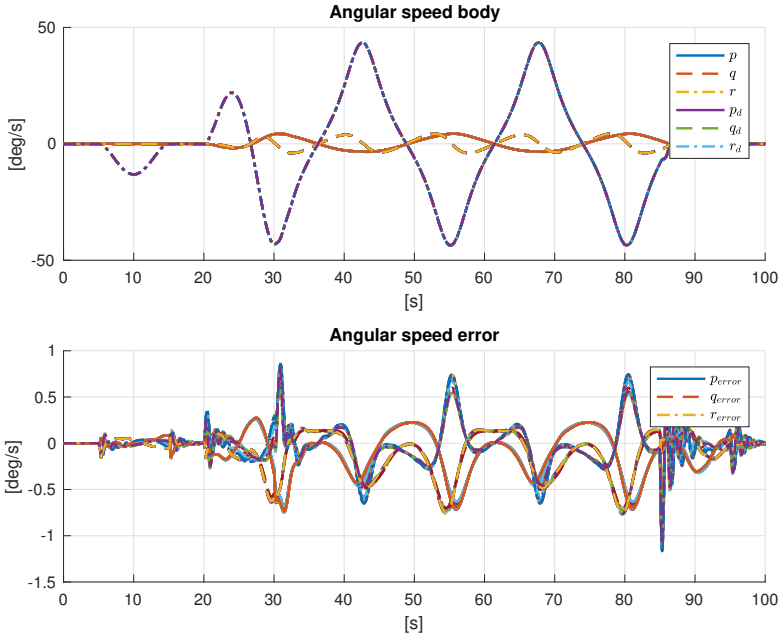

Figure 4.29: Geometric control with modified attitude reference: Angular speed and error in eight-shape trajectory simulation with ±20% variation of I_b

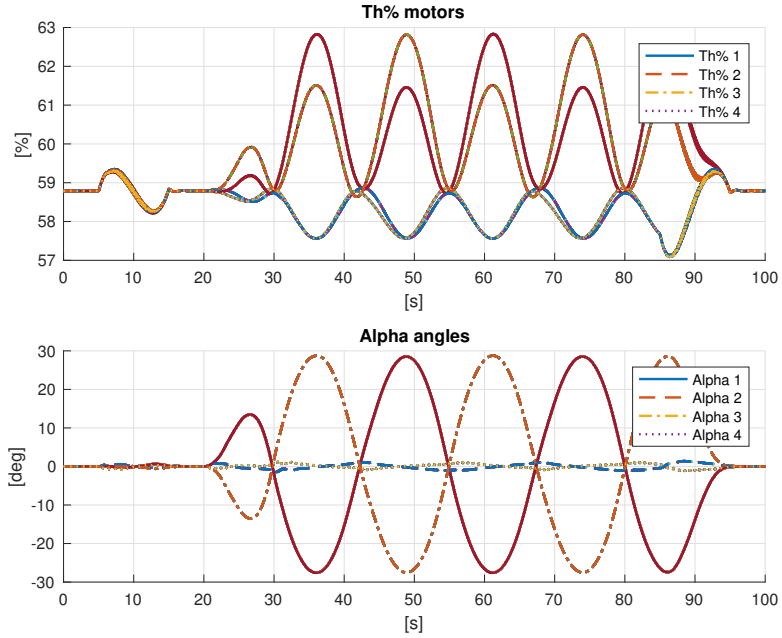

Figure 4.30: Geometric control with modified attitude reference: throttle percentages and tilting angles in eight-shape trajectory simulation with ±20% variation of I_b

4.3 Constant disturbances rejection

In Section 4.1, the controllers capability to reject the measurement noise is evaluated by adding to the feedback signals white noises, that have zero mean value. In order to evaluate the effects of constant disturbances and to simulate the presence of whatever external force/torque that can act on the system, a simulation for each controller is performed. In particular, the platform is asked to hover in $p_d = [0.5\ 1\ -1]^T$ with attitude $\Phi = [0^\circ\ 0^\circ\ 0^\circ]^T$ and

- at time $t = 30s$, a constant force disturbance $F_{dist} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ is injected
- at time $t = 60s$, a constant torque disturbance $\tau_{dist} = \begin{bmatrix} 0 \\ 0.1 \\ 0 \end{bmatrix}$ is injected.

Feedback linearization

The feedback linearization controller presented in Section 2.2 is able to reject constant forces and torques disturbances. Figures 4.31 and 4.32 show that even in presence of constant disturbances the steady state position and attitude errors converges to zero thanks to the integral action of the controller.

Geometric control for the fully actuated case

Figures 4.33 and 4.34 show the behavior of the controlled system adopting the geometric controller presented in Section 3.1 when constant disturbances are injected in the system. Looking at the results, the constant force disturbance along the N and E axes causes a constant error in the position tracking, while the constant torque disturbance around the Y_b axis causes constant position and attitude errors. The errors do not converge to zero, since the geometric controller do not have integral action on the errors.
The fact that this controller works very well in the simulations performed in Section 3.1, where the steady state errors are zero, is due to the fact that the simulated model is almost perfect, in the sense that no external unbalanced forces of any nature is present.

Geometric control with dynamic reference attitude

Figures 4.35 and 4.36 show the simulation results when the system, controlled by the geometric control presented in Section 3.2, is subject to constant disturbances. As in the previous case, this geometric controller do not let to reach the zero steady state error due to the lack of the controller integral action. In this case, the constant force disturbance injected at time $t = 30s$ implies not only the presence

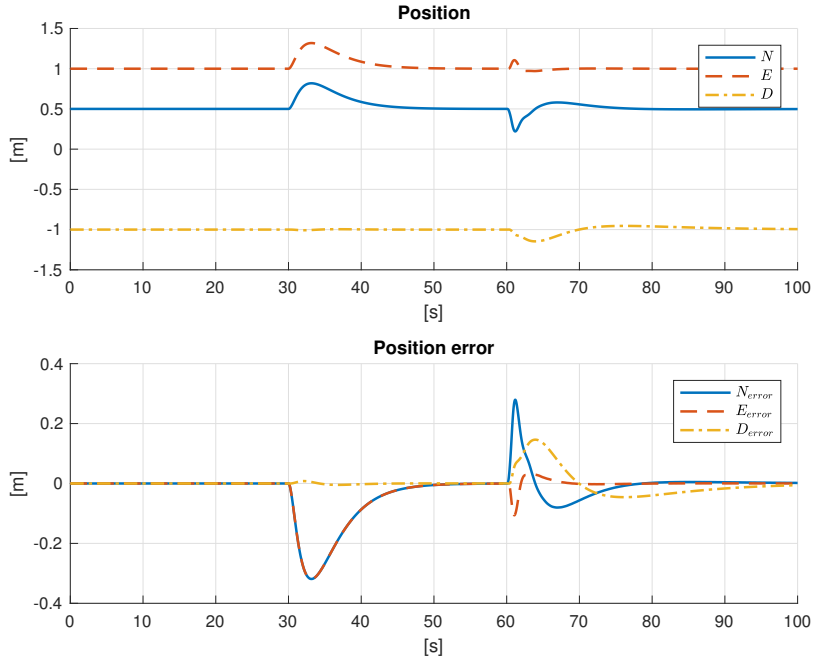

Figure 4.31: Feedback linearization control: position and position error with constant force and torque disturbances

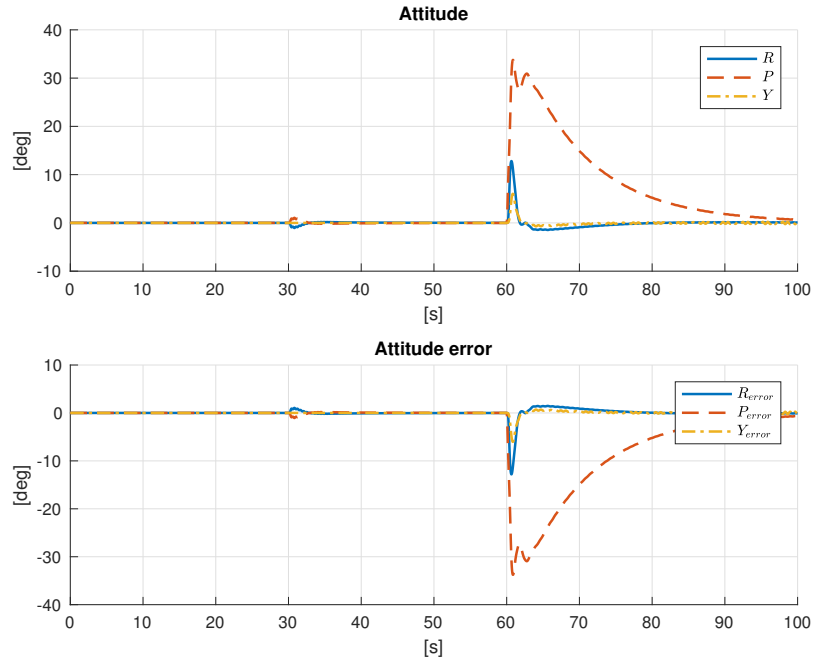

Figure 4.32: Feedback linearization control: attitude and attitude error with constant force and torque disturbances

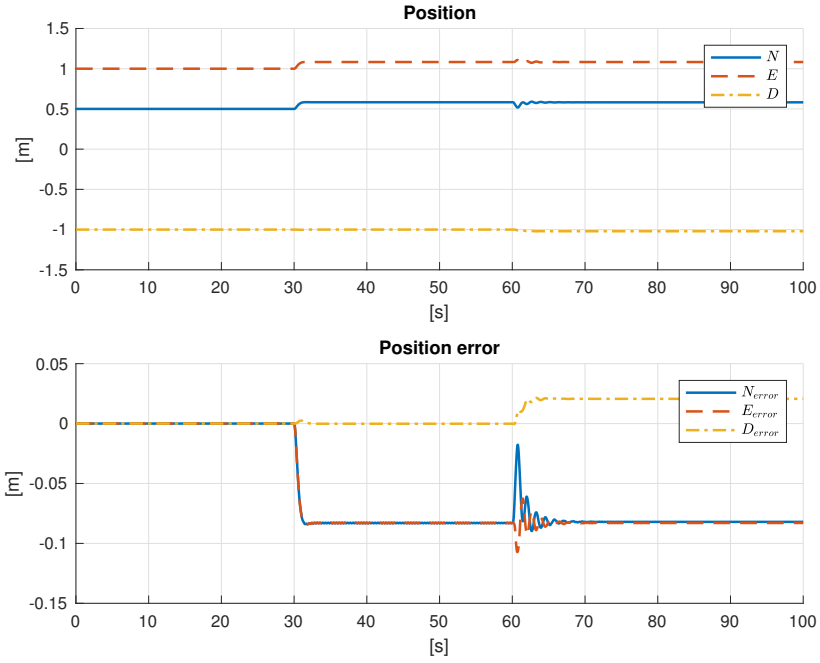

Figure 4.33: Geometric control: position and position error with constant force and torque disturbances

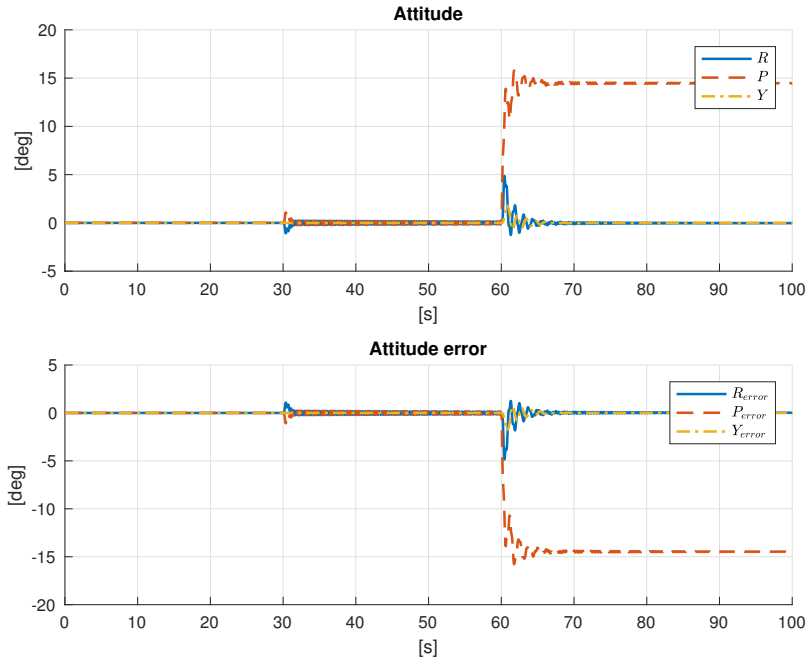

Figure 4.34: Geometric control: attitude and attitude error with constant force and torque disturbances

of a constant position error, but also a constant attitude error. This is due the fact that, thanks to the R_c orientation matrix computation shown in equation (3.39), the desired vertical axis has the same direction of the desired control force, shown in equation (3.41), that is pointing towards the desired position $p_d = [0.5\ 1\ -1]^T$.

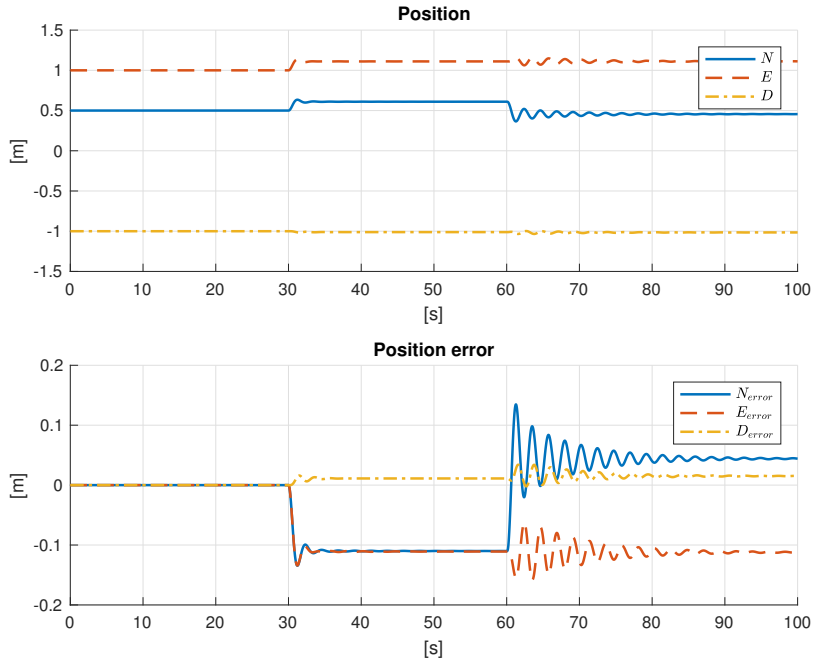

Figure 4.35: Geometric control with modified attitude reference: position and position error with constant force and torque disturbances

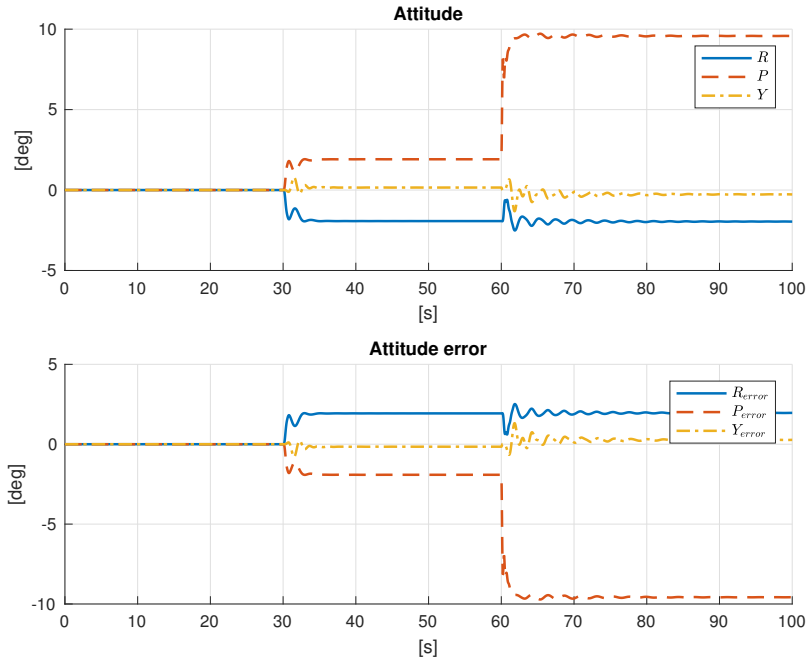

Figure 4.36: Geometric control with modified attitude reference: attitude and attitude error with constant force and torque disturbances

4.4 Considerations

The geometric control laws presented in Chapter 3, as shown in the previous Section 4.1, are really robust with respect to measurement noise on the feedback signals. Considering that a variation of $\pm 20\%$ around the nominal value is a significant one, these controllers are also robust with respect to body inertia tensor uncertainty (Section 4.2). The lack of an integral action makes them difficult to be implemented in the non-ideal real world, where unconsidered external forces and moments act on the system. The rationale behind the development of these geometric controllers, as explained in Invernizzi and Lovera [2017b], is the capability to take into account for intrinsic tilting limitations of the actuators, and in the future the insertion of the integral action will be certainly studied and performed. The feedback linearization controller presented in Section 2.2 is able to reject the measurements noise but, on the contrary, it is not really robust with respect to I_b variations. A variation of $\pm 15\%$ around its nominal value is enough to obtain a non satisfactory, strongly oscillating behavior. However, the presence of the integral action in the feedback linearization controller makes it suitable for a real application, assuming that the system parameter uncertainty is small enough to guarantee the stability.

Conclusions

The purposes of this thesis was the analysis, the implementation and the numerical simulation of nonlinear control laws for the tilt-rotor quad-copter designed and realized in Micheli [2016].
The conducted activities start with an introductory first chapter, in which, after the prototype description, the tilt-rotor mathematical model is shown along with all the mathematical and geometrical formalisms to avoid any ambiguities. Given that the complete mathematical model of the tilt-rotor is a complex model, a reduced model for control purposes is also presented. The first introductory chapter ends with the description of the nonlinear mixer matrix that maps the output of the controller with the input of the tilt-rotor model.
In the second chapter, the control problem of the tilt-rotor platform is addressed through two feedback linearization controllers. The first control law resorts to the dynamic extension of the system in order to perform the dynamic inversion, necessary for the feedback linearization control law. Since this controller requires to perform the dynamic inversion at every control cycle, leading to a computationally heavy control law, the second feedback linearization control law solves this issue through the use of the nonlinear matrix presented in the first chapter. In order to assess the stability and the capabilities of the presented feedback linearization controllers, numerical simulations have been performed and commented.
The third chapter let to exploit the natural framework of the tilt-rotor control problem by addressing it in the geometric control framework. Two geometric Lyapunov-based control laws are presented, showing the numerical simulations performed. The first geometric control law has been designed assuming a fully actuated system, *i.e.*, considering the tilt-rotor capable to direct the generated thrust in every direction, while the second geometric control law consists in an evolution of the first one. It is in fact capable to take into consideration tilting limitation of the servo-actuators.
The final chapter deals with a qualitative robustness analysis of the four presented control laws. In particular, the effects due to the body inertia tensor variations are evaluated by performing a simulation campaign. Furthermore, the capability of the control laws to reject the measurement noise and constant disturbances are also evaluated through other simulations. Finally some considerations about the control laws implementability on the real prototype are exposed.

Concluding, some indications of future developments and improvements for the present thesis work are left:

- Add the integral action in the geometric control laws
- Evaluate the effects of performing the weighted pseudo-inversion of the mixer matrix
- Evaluate the possibility to exploit the two redundant degrees of actuation in order to perform additional tasks

Bibliography

Botta. Droni, é boom: ecco tutti gli usi, 2015. URL `http://espresso.repubblica.it/attualita/2015/03/19/news/droni-e-boom-ecco-tutti-gli-usi-1.204995`.

Bullo and Murray. Tracking for fully actuated mechanical systems: a geometric framework. *Automatica*, 35(1):17–34, 1999.

Giurato. Design, integration and control of a multirotor uav platform. *Master thesis*, 2015.

Invernizzi and Lovera. Geometric tracking control of a quadcopter tiltrotor uav. 20^{th} *IFAC World Congress, Toulouse, France. Accepted*, 2017a.

Invernizzi and Lovera. Geometric tracking control of thrust vectoring uavs. *arXiv preprint arXiv:1703.06443*, 2017b.

Lavretsky and Wise. Robust and adaptive control with aerospace applications, 2013.

Mahony, Kumar, and Corke. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics Automation Magazine*, 19(3):20–32, Sept 2012. ISSN 1070-9932. doi: 10.1109/MRA.2012.2206474.

Micheli. Design, identification and control of a tiltrotor quadcopter uav. *Master thesis*, 2016.

Ryll, Bülthoff, and Giordano. A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Transactions on Control Systems Technology*, 23(2):540–556, 2015.

Seifried. *Dynamics of underactuated multibody systems: modeling, control and optimal design*, volume 205. Springer Science & Business Media, 2013.

Appendix A

Computation of $\dot{R}_C$ and $\ddot{R}_C$

The matrices $\dot{R}_c$ and $\ddot{R}_c$ are defined as

$$\dot{R}_c = \begin{bmatrix} \dot{b}_{c_1} & \dot{b}_{c_2} & \dot{b}_{c_3} \end{bmatrix} \tag{A.1}$$
$$\ddot{R}_c = \begin{bmatrix} \ddot{b}_{c_1} & \ddot{b}_{c_2} & \ddot{b}_{c_3} \end{bmatrix} \tag{A.2}$$

where

$$\dot{b}_{c_3} = \frac{\dot{a}}{||a||} - (a^T \dot{a}) \frac{a}{||a||^3} \tag{A.3}$$
$$\dot{b}_{c_2} = \frac{\dot{c}}{||c||} - (c^T \dot{c}) \frac{c}{||c||^3} \tag{A.4}$$
$$\dot{b}_{c_1} = \dot{b}_{c_2} \times b_{c_3} + b_{c_2} \times \dot{b}_{c_3} \tag{A.5}$$
$$\ddot{b}_{c_3} = \frac{\ddot{a}}{||a||} - 2(a^T \dot{a}) \frac{\dot{a}}{||a||^3} - (||\dot{a}||^2 + (a^T \ddot{a})) \frac{a}{||a||^3} + 3(a^T \dot{a})^2 \frac{a}{||a||^5} \tag{A.6}$$
$$\ddot{b}_{c_2} = \frac{\ddot{c}}{||c||} - 2(c^T \dot{c}) \frac{\dot{c}}{||c||^3} - (||\dot{c}||^2 + (c^T \ddot{c})) \frac{c}{||c||^3} + 3(c^T \dot{c})^2 \frac{c}{||c||^5} \tag{A.7}$$
$$\ddot{b}_{c_1} = \ddot{b}_{c_2} \times b_{c_3} + 2\dot{b}_{c_2} \times \dot{b}_{c_3} + b_{c_2} \times \ddot{b}_{c_3} \tag{A.8}$$

and

$$a = -f_c^d = -\left(-K_x e_x - K_v e_v + m(\ddot{p}_d - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}\right) \tag{A.9}$$

$$\dot{a} = -\left(\frac{K_x K_v}{m} e_x + \left(\frac{K_v^2}{m} - K_x\right) e_v - \frac{K_v}{m} Df + m\dddot{p}_d\right) \tag{A.10}$$

$$\ddot{a} = -\left(\frac{K_x K_v}{m} e_v + \left(\frac{K_v^2}{m} - K_x\right)\left(-\frac{K_x}{m} e_x - \frac{K_v}{m} e_v \frac{D_f}{m}\right) - \frac{K_v}{m}\dot{D}_f + m\ddddot{p}_d\right) \tag{A.11}$$

$$c = b_{c_3} \times b_{d_1} \tag{A.12}$$

$$\dot{c} = \dot{b}_{c_3} \times b_{d_1} + b_{c_3} \times \dot{b}_{d_1} \tag{A.13}$$

$$\ddot{c} = \ddot{b}_{c_3} \times b_{d_1} + 2\dot{b}_{c_3} \times \dot{b}_{d_1} + b_{c_3} \times \ddot{b}_{d_1} \tag{A.14}$$

$$D_f = {}^W R_B f_c^{dc} - a \tag{A.15}$$

$$\dot{D}_f = bR_e^{dc}(R_d e_\omega^{dc})^\wedge a - e_R^{dc^T} R_{dc} \frac{e_\omega^{dc}}{\Psi_M} R_e^{dc} a + bR_e^{dc}\dot{a} - \dot{a} \tag{A.16}$$

$$b = \frac{\Psi_M - \Psi_{dc}}{\Psi_M}. \tag{A.17}$$

The operator $^\wedge$ is the map from $\mathbb{R}^3$ to $\mathfrak{so}(3)$.

Appendix B

Perturbed inertia tensors values

The body inertia tensor is defined as

$$I_b = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}$$

and its nominal values (in $kg \cdot m^2$) are

$$\begin{cases} I_{xx} = 0.0074 \\ I_{yy} = 0.0074 \\ I_{zz} = 0.05. \end{cases} \tag{B.1}$$

As explained in Section 4.2, the vector P_{10} contains twenty sets of perturbed values of I_b of $\pm 10\%$ about its nominal values, P_{15} of $\pm 15\%$ and P_{20} of $\pm 20\%$. Tables B.1, B.2, B.3 show the perturbed values of the inertia tensors used in Section 4.2.

Parameter	I_b 1	I_b 2	I_b 3	I_b 4	I_b 5	I_b 6	I_b 7
I_{xx}	0.0079	0.0080	0.0071	0.0081	0.0081	0.0069	0.0078
I_{yy}	0.0080	0.0076	0.0075	0.0069	0.0074	0.0073	0.0081
I_{zz}	0.0463	0.0460	0.0546	0.0547	0.0530	0.0542	0.0516
	I_b 8	I_b 9	I_b 10	I_b 11	I_b 12	I_b 13	I_b 14
I_{xx}	0.0067	0.0077	0.0072	0.0077	0.0067	0.0077	0.0067
I_{yy}	0.0079	0.0078	0.0076	0.0067	0.0068	0.0071	0.0073
I_{zz}	0.0543	0.0524	0.0467	0.0478	0.0532	0.0545	0.0488
	I_b 15	I_b 16	I_b 17	I_b 18	I_b 19	I_b 20	
I_{xx}	0.0078	0.0074	0.0077	0.0077	0.0068	0.0072	
I_{yy}	0.0078	0.0073	0.0078	0.0076	0.0074	0.0075	
I_{zz}	0.0469	0.0515	0.0478	0.0466	0.0546	0.0472	

Table B.1: Perturbations of ±10% of body inertia tensor

Parameter	I_b 1	I_b 2	I_b 3	I_b 4	I_b 5	I_b 6	I_b 7
I_{xx}	0.0075	0.0067	0.0071	0.0065	0.0074	0.0070	0.0081
I_{yy}	0.0069	0.0078	0.0077	0.0084	0.0073	0.0074	0.0081
I_{zz}	0.0537	0.0453	0.0542	0.0541	0.0492	0.0502	0.0522
	I_b 8	I_b 9	I_b 10	I_b 11	I_b 12	I_b 13	I_b 14
I_{xx}	0.0071	0.0071	0.0075	0.0068	0.0068	0.0068	0.0073
I_{yy}	0.0081	0.0084	0.0077	0.0070	0.0082	0.0067	0.0070
I_{zz}	0.0505	0.0556	0.0513	0.0496	0.0454	0.0459	0.0564
	I_b 15	I_b 16	I_b 17	I_b 18	I_b 19	I_b 20	
I_{xx}	0.0072	0.0085	0.0069	0.0069	0.0068	0.0070	
I_{yy}	0.0067	0.0073	0.0072	0.0076	0.0066	0.0072	
I_{zz}	0.0561	0.0442	0.0514	0.0532	0.0470	0.0501	

Table B.2: Perturbations of ±15% of body inertia tensor

Parameter	I_b 1	I_b 2	I_b 3	I_b 4	I_b 5	I_b 6	I_b 7
I_{xx}	0.0083	0.0086	0.0067	0.0088	0.0088	0.0063	0.0083
I_{yy}	0.0086	0.0078	0.0075	0.0064	0.0074	0.0072	0.0088
I_{zz}	0.0425	0.0420	0.0592	0.0594	0.0560	0.0583	0.0531
	I_b 8	I_b 9	I_b 10	I_b 11	I_b 12	I_b 13	I_b 14
I_{xx}	0.0060	0.0079	0.0071	0.0080	0.0061	0.0080	0.0060
I_{yy}	0.0084	0.0082	0.0079	0.0060	0.0062	0.0069	0.0072
I_{zz}	0.0587	0.0549	0.0434	0.0455	0.0565	0.0590	0.0476
	I_b 15	I_b 16	I_b 17	I_b 18	I_b 19	I_b 20	
I_{xx}	0.0082	0.0074	0.0080	0.0079	0.0063	0.0069	
I_{yy}	0.0083	0.0072	0.0082	0.0079	0.0074	0.0077	
I_{zz}	0.0437	0.0529	0.0455	0.0433	0.0592	0.0445	

Table B.3: Perturbations of ±20% of body inertia tensor