Quadrotor Modeling and Control

Stig Espen Krisitiansen

SHO 6300

Master Thesis, Satellite Engineering
Department of Electrical Engineering
University of Tromsoe campus Narvik,
Narvik,
Norway

June 24, 2016

Master thesis project 2016
for
Stig Espen Kristiansen

Quadrotor Modeling and Control

Figure 1: Image of a quadrotor. Illustrated by Tom Stian Andersen

Since the first flight of an unmanned aerial vehicle (UAV) in 1804 by George Cayley, and especially in the last two decades, a considerable effort has been made to improve UAV technologies aiming at safety and reliability of unmanned aviation. The result is seen today as a growing use of UAV systems to perform a variety of tasks, such as military reconnaissance, geological surveys, environmental monitoring, and time-optimal search and rescue operations.

UAV systems include various types of unmanned aerial vehicles, including e.g fixed wing aircrafts, helicopters and quadrocopters. Where fixed wing aircrafts have requirements of sufficient flight speed to avoid stalling, helicopters and quadrocopters have the possibility to perform hovering operations and may therefore be used for extended surveillance of a particular area. However, the latter will typically have a shorter operating range and more complex dynamics. It is therefore important to derive a detailed nonlinear mathematical model of the system for control purposes.

Main Task

The main task of this project is to derive a detailed mathematical model of a quadrocopter as shown in the picture above. The model should be implemented in Matlab/Simulink, together with a simple control solution.

Technical specifications for the system are chosen by the student in cooperation with the supervisor.

Subtasks

1. Study previous work on quadrocopters and their applications, including cascaded systems modeling and passivity-based control.
2. Derive a mathematical model of quadrocopter dynamics using a cascaded framework, including actuation for rotation and translation of the system, and implement the model in Matlab/Simulink.
3. Derive a passivity-based control solution for the system, and show the performance of the controller through simulations.

Abstract

In this thesis a mathematical quadrotor model based on cascade modeling theory, using a passivity-based control solution is derived and presented. The rotation and translation cascades are first decoupled, modeled mathematically and simulated separately to test individual control solutions and stability, followed by total system modeling and simulations to verify the control solutions and stability of the total system. Both the decoupled systems and the total system are able to track both fixed positions and positions that change with time, such as the circle, helix and waypoint tracking. The simulation figures for the spiral trajectory tracking shows a growth in position error as the radius of the circle increases, indicating that the control solutions are struggling with increase in acceleration (*jerk*), but is believed to be rectified by additional compensation terms. Overall the system performs well, and can be presumed stable.

Preface

This thesis was written in the spring of 2016 as the concluding work of the master of technology study program in 'Satellite Engineering' at the Department of Electrical Engineering, University of Tromsoe campus Narvik.

In the beginning when I started working on the thesis, I had limited knowledge about quadrotors in general, not to mention quadrotor dynamics and kinematics. Along the course of the spring a greater understanding emerged, although some "headaches" troubled me until the end of the project. I appreciate the knowledge and understanding I have attained regarding quadrotor workings and mathematical modeling these past 8 months, although frustrating at times. I am proud of the work I have produced, and I have proved to myself that I am able to overcome large obstacles.

I would like to thank my supervisor Associate Professor Raymond Kristiansen for his support and guidance during the project. I would like to thank Tor-Aleksander Johansen for answering the questions I had for him as best he could, Espen Oland for pointing me in the right direction at the start of the project, my mother and sister for believing in me all the way, and Martin Johansen for many good discussions and for being my classmate these past two years.

Finally I would like to extend my immense gratitude to Ph.D student Tom Stian Andersen, whose help has been invaluable regarding questions ranging from Latex trouble and Matlab work to quadrotor mathematics.

University of Tromsoe campus Narvik, June 24, 2014

Stig Espen Krsitiansen

Contents

Chapter 1

Introduction

Since man first gazed upon the sky with awe at its beauty and grandeur, discovering creatures with the ability to soar free and untethered from the surface, he has yearned for the skies and the freedom it represents. For centuries thinkers, scientists and adventurers have made attempts of flight using various methods, but it would take over 2000 years before the first modern flying machine would see the light of day.

The Beginning
It is believed that kites may have been the first form of man-made aircraft and was invented in China, possibly as early as the 5th century BC. It is also believed that man-carrying kites were used in ancient China, for both military and civil purposes, even as a punishment. An early recording of such a flight was that of prisoner Yuan Huangtou, a Chinese prince in the 6th century AD. The Chinese also invented a toy in the form of the bamboo-copter which made use of rotor wings like a helicopter and has existed since around 400 BC.

The Renaissance and Rational Designs
From the Period of Chinese kites and up until the rational aircraft designs of the renaissance era, tower jumping was the most common experimental method of flight, using feathers glued to make-shift wings or wings made out of wood and cloth. But with the coming of one of the renaissance's most famous thinkers, namely Leonardo da Vinci, a new approach to aircraft design was used: Rational design. Da Vinci studied bird flight and while analyzing it, anticipated many principles of aerodynamics. At the end of the 15th century he sketched and designed several flying machines and contraptions, such as ornithopters, fixed-wing gliders, rotor-craft and parachutes. Although rational, the designs were not based on good science, and since his work remained unknown until 1797, his designs did not influence the development over the next three hundred years.

Lighter than Air: Balloons
In the 17th and 18th century it was suggested that lighter than air flight was possible. In 1670 a man named Francesco Lana de Terzi proposed a theory, known as *Vacuum Airship*, that it would be possible to lift an airship using copper foil spheres containing a vacuum, which would be lighter than the displaced air. Of course this theory is not feasible with any current materials due to the fact that the surrounding pressure would crush the spheres.

Instead methods including hot air and hydrogen were used, and during the late 18th century five aviation firsts were achieved in France:

1. Demonstration of unmanned hot air balloon in Annonay by the Montgolfier brothers.
2. The launch of the first unmanned hydrogen-filled balloon from Champ de Mars, Paris by Jacques Charles and the Robert brothers.
3. The launch of the first manned flight, a tethered balloon with humans on board at the Folie Titon in

Paris by the Montgolfier brothers. The aviators were Jean-Francois Piltre de Rozier, Jean-Baptiste Rveillon and Giroud de Villette.

4. First free flight with human passengers launched by the Montgolfier brothers and piloted by Jean-Francois Piltre de Rozier and Marquis Francois d'Arlandes.

5. Launch of the first manned hydrogen balloon by Jacques Charles and Nicolas-Louis Robert from Jardin des Tuileries in Paris.

The most common type of balloons from the 1790s to the 1960s were gas balloons. These were replaced by hot air balloons with the invention of a more sophisticated heat source, making it easier to control ascent and decent. Today ballooning is associated with recreational activities and it is believed that some 7500 balloons are operating in the United States alone.

Lighter than Air: Airships
Airships, or dirigible balloons as they were originally called were mainly developed from the late 18th, to the early 20th century. The design comprised of three different types of airships: *Non-rigid* (blimps), which rely on internal pressure to maintain the shape of the airship, semi-rigid, which relies on internal pressure as well but has some sort of supporting structure, such as a fixed keel, and *rigid*, which has an outer structural framework, which carries all the load and maintains the shape.

The most famous airships throughout history are the Zeppelins, rigid airship designs pioneered by the German count Ferdinand von Zeppelin. At the time rigid airships had far superior lifting capabilities compared to the fixed-wing aircraft, and resulted in passenger transport zeppelins like the "Hindenburg". Although used extensively in the early 20th century, through World War I and II, the airship was subdued by the advancements in heavier-than-air crafts, and are today mostly used as advertising, sightseeing and surveillance.

Heavier than Air: Fixed-Wing
Along side the development of lighter-than-air crafts, studies in heavier-than-air flight were also carried out. In the 18th century, the first paper on aviation was published by Emanuel Swedenborg in 1716 and suggested a design for an aircraft with wings working similar to a birds wings. Although he knew it would not fly, he was confident that the problem would be solved.

In the late 18th century, George Cayley, also known as the "father of the aeroplane", began the first rigorous study of the physics of flight and would later design the first modern heavier-than-air craft (HTAC). Some of Cayley's contributions to aeronautics include:

- Clarifying our ideas and laying down the principles of heavier-than-air flight
- Reaching a scientific understanding of the principles of bird flight
- Conducting scientific aerodynamic experiments demonstrating drag and streamlining, movement of the center of pressure, and the increase in lift from curving the wing surface
- Defining modern aeroplane configuration comprising a fixed wing, fuselage and tail assembly
- Demonstration of manned, gliding flight
- Setting out the principles of power to-weight-ration in sustained flight

In 1848 Cayley constructed a triplane glider large and safe enough to carry a child, where a local unknown boy was chosen, and in 1852 he published the design of a full-sized manned glider the "governable parachute", which carried the first adult aviator across Brompton Dale in 1853.

In the late 19th century Samuel Pierpoint Langley started an investigation into aerodynamics and later turned to building his designs. His Aerodrome No. 5 made the first successful sustained flight of an unmanned engine-driven HTAC on May 6, 1896 followed by Aerodrome No. 6 in November 28, 1896 but never

managed to design a successful model scaled for manned flight.

The names that may be the most well known in the history of aeronautics are that of Wilbur and Orville Wright, who made the first known sustained, controlled powered HTAC flights, which were undertaken at Kill Devil Hills, North Carolina on December 17, 1903 in their craft 'The Wright Flyer'. Following the success of the flights in 1903 the Wrights continued to fly in 1904-05 and introduced the Flyer II in 1904, which was an improved version of the original Flyer and in 1905 Flyer III was launched, which became the first practical aircraft, though without wheels and needing a launching device.

Following the success of the Wright brothers and technology advances in European flight pioneering, flight became an established technology ca. 1908 and has since then been subjected to huge technological advancement from the simple designs using wood, canvas and modest internal combustion engines to the metal, electronics, computers and jet propulsion that is common today.

Heavier Than Air: Rotorcraft
References of vertical flight can be dated as far back as 5th century BC, and found in China, in the form of a toy called the bamboo-copter, and in the 4th century book Baopuzi by Ge Hong, which reportedly describes some of the ideas inherent to rotary wing aircraft.

During the Renaissance the Chinese helicopter toy was introduced in paintings and other works. It was in this era that Leonardo da Vinci created a design for a machine, described as an aerial screw. Until he did this there had not been any recorded advancement made towards vertical flight. As scientific knowledge increased, man continued to pursue vertical flight, and many of these machines would more closely resemble the ancient bamboo top with spinning wings rather than Leonardo's screw.

In the 18th century some coaxials modelled after the Chinese top were developed. One in 1754 by Mikhail Lomonosov powered by a spring, and one in 1783 by Christian de Launoy and his mechanic, Bienvenu, consisting of a contra-rotating of turkey flight feathers as rotor blades. Even George Cayley developed a model similar to that of Launoy and Bienvenu, but powered by rubber bands, and later progressed to using sheets of tin for rotor blades and springs for power.

The term "helicopter" was coined by Gustave de Ponton d'Amécourt in 1861, a French inventor who demonstrated a small steam- powered model. In 1878 the first vehicle of its kind, designed by the Italian Enrico Forlanini, rose to a hight of 12ft and hovered for some 20 seconds after a vertical take-off, unmanned and powered by a steam engine. In 1906 the brothers Jacques and Louis Breguet experimented with aerofoils for helicopters, and those experiments resulted in the Gyroplane No. 1, possibly the earliest known example of a quadcopter. Sometime between August and September 1907, the Gyroplane No. 1 lifted its pilot about 0.6m into the air for one minute. Although considered to be the first manned flight of a helicopter, but not free and untethered flight because one man had to stand and hold it in each corner to stabilize the vehicle. Instead the first truly free flight was conducted by Paul Cornu in his Cornu helicopter November 13. the same year.

The early 20th century saw increasing development in rotorcraft technology and experimentation, and leaps were taken during the 1920s and 30s. The first mass produced helicopter was the Sikorsky R-4, designed by Igor Sikorsy and were produced between 1942-44. The 1950s saw the first turbine engine-powered helicopter K-255 designed by Charles Kaman, which spurred the further development of helicopters along with advancements within other technologies, such as electronics, and much like with the development of fixed-wing aircraft, into the the direction of today's modern helicopters.(Crouch, 2004)

Unmanned Aerial Vehicles(UAVs)
UAVs have always been a part of aviation history, in the sense that, generally back in the experimentation and pioneering era of flight technology, first flights of prototypes, ranging from kites to rotorcrafts, were

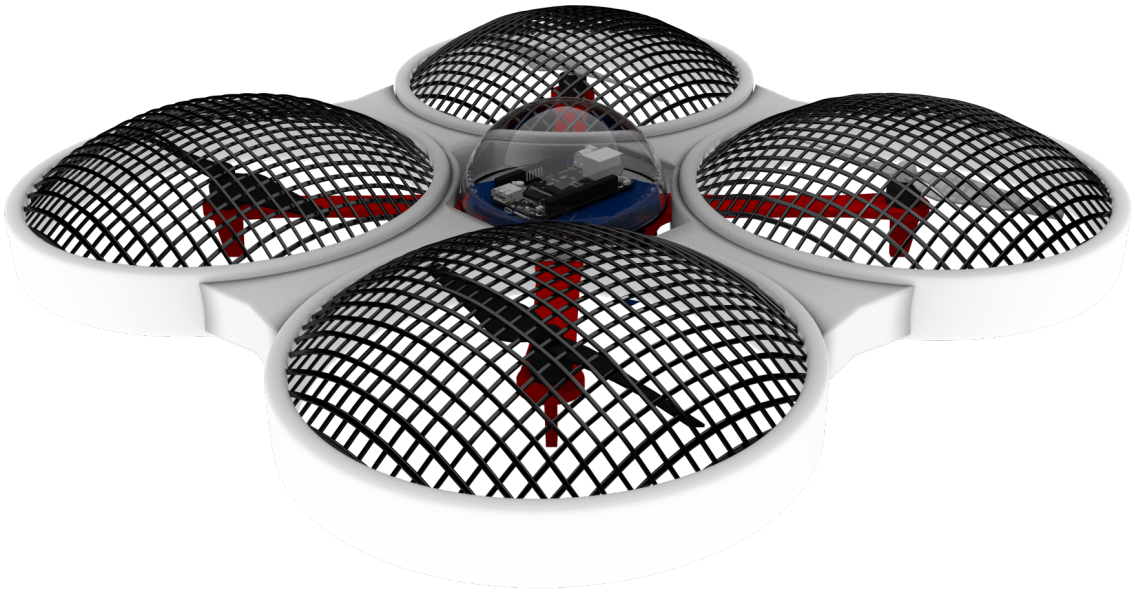


Figure 1.1: Configuration of a quadrotor (Illustration by Tom Stian Andersen).

almost always conducted without pilots. Today however, when regarding UAVs, most think of military drones such as the Predator drone used by the United States Air Force, which is a form of fixed-wing UAV, or remote controlled quadcopters or quadrotors, which is a form of rotorcraft UAV, and this thesis' subject of focus.

Quadrotor UAVs
A quadrotor is a multi rotor helicopter propelled by four (two pairs) of identical fixed pitched rotors, where one pair rotates clockwise and the other pair rotates counter-clockwise. These use independent variation of the speed of each rotor to achieve control and movement in the form of total thrust and total torque. Quadrotors were seen as a possible solutions to persistent problems in vertical flight such as torque-induced control issues, which can be eliminated with counter-rotation and the short blades are easier to construct. Early prototypes suffered from poor performance and later prototypes required too much pilot work, due to poor stability and limited control authority. However, advancements in technology in the late 2000s, namely micro electro-mechanic systems MEMS, allowed the production of cheap and lightweight sensor components, such as accelerometers, GPS, barometers and cameras. The quadrotor has become a popular research platform for new ideas within a number of fields, including flight control theory, navigation, real time systems and robotics, due to its small size and mechanical simplicity. The quadrotor can be applied in many scenarios, ranging from military use such as surveillance and reconnaissance, commercial use within aerial imagery, and humanitarian use like search and rescue.(Hoffmann et al., 2004)

1.1 Background and the Purpose of the Thesis

The background for this thesis is, that it is desired to investigate if a cascaded quadrotor system model always becomes ideally UGAS or ULES regardless of which type of control solution that is chosen.

The purpose of the thesis is to verify if a passivity based control solution can be applied to a cascaded quadrotor model system.

1.2 Previous Work

1.2.1 Previous Work in General

When it comes to previous work within quadrotor modeling the dynamics of a quadrotor using i.e. Euler angles or quaternions are well developed and much of the work done related to quadrotors have to do with improved control under various conditions.

In Beul et al. (2014) nonlinear model-based position control for a micro aerial vehicle (MAV)is proposed, where the model makes use of a variety of sensors to navigate to desired way-points created by a planning algorithm. The paper studies both simulation data for the control algorithm and real flight data for the vehicle MAV 1, with the intention of studying a second vehicle MAV 2 with an array of more advanced sensors to give faster and more accurate feedback.

In Alexis et al. (2011) a model for predictive indoor control is proposed for environments where absolute-localization data, like GPS or positioning from external cameras are inadequate, by the use of an internal measurement unit (IMU) and an optical flow sensor. The dynamics $\dot{\mathbf{X}}$ of the quadrotor is presented in its augmented form and comprise of a $\mathbb{R}^{12}$ matrix that contains translation, rotation and input components summed together with the additive disturbance vector. The model predictive control (MPC) in this case uses the estimated rotational components provided by the IMU, the calibrated translational acceleration measurements expressed in the Earth-fixed frame, and a data fusion with the ground distance measurement of a sonar through a 2-state Extended Kalman Filter to estimate the altitude and change in altitude. the control-scheme itself is based on three cascade switching model predictive controllers applied on the multiple Piecewise Affine representation of the augmented dynamic system and the error dynamics modeling for vertical and planar motions.

Other methods for control of a quadrotor can be found in, for example, Mian and Daobo (2008) which proposes altitude, position and a backstepping-based PID control technique for the rotation subsystem, and in Pounds et al. (2010) where control of a large quadrotor (e.g $>$ 3kg) is proposed.

1.2.2 Cascade Modeling

Cascade modeling (CM) is a form of mathematical modeling used in control engineering to prove stability of a certain system, where the system itself is divided into a cascade interconnected system instead of a feedback interconnected system. The general idea behind CM is to decouple a system that has multiple distinctive subsystem functions, i.e. rotation and translation, and try to prove that each subsystem is by itself stable. If it can be proven that *subsystem 1* is stable, and that *subsystem 2* is stable, combined with restrictions of the growth term $g(t,x)x_2$ represented in (1.1), it follows that *stable + stable* $\Rightarrow$ *stable*, meaning that the total system consequently becomes stable. For large and complicated systems CM has its advantages due to the fact that smaller, simpler systems are easier to prove stable, thus lengthy and complicated calculations can be avoided. Cascade theories and work is presented in Lamnabhi-Lagarrigue et al. (2004); Loría (2004a,b), where different forms and configurations of cascade interconnected systems and stability are discussed. For non-linear time-varying systems (NLTV) a general expression is described as

$$\Sigma_1 : \dot{x}_1 = f_1(t, x_1) + g(t, x)x_2 \tag{1.1}$$
$$\Sigma_2 : \dot{x}_2 = f_2(t, x_2) \tag{1.2}$$

where x_2 becomes the input of system Σ_1 and x_1 becomes the total output of the system. It is stated in Khalil (2002) that the term $g(t,x)$ represents perturbation in the system that could result from modeling errors, aging, or uncertainties and disturbances, which is inherent in any realistic problem. A generic representation of a cascaded system can be as shown in Figure 1.2 , but cascaded structures can also be constructed as shown in Figure 1.3, Figure 1.4 and Figure 1.5.

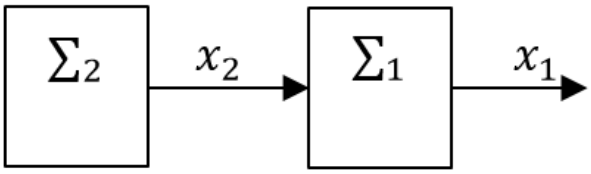


Figure 1.2: Block diagram example of a cascade interconnected system inspired by (Loría, 2004b)

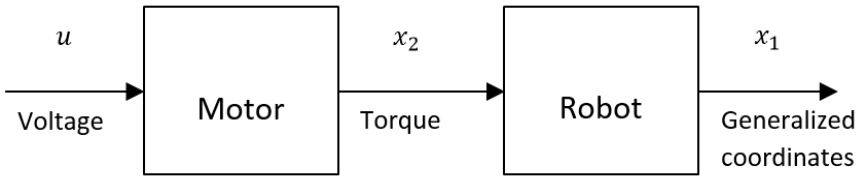


Figure 1.3: Case 1, the plant itself has a cascaded structure, example robot with a motor. inspired by (Loría, 2004b)

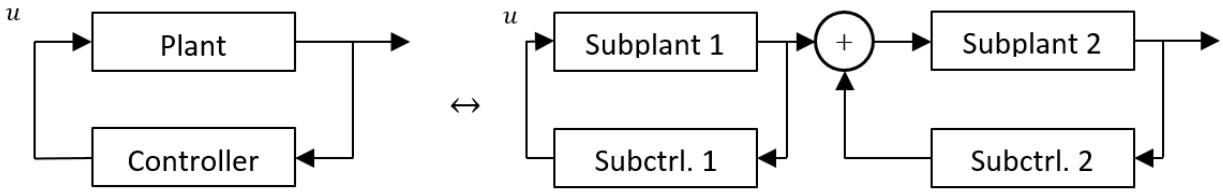


Figure 1.4: Case 2, appeared cascaded structure when applying a control input. inspired by (Loría, 2004b)

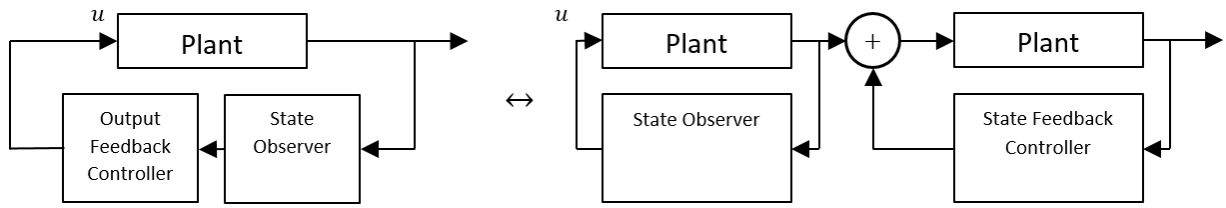


Figure 1.5: Case 3, output-feedback control design that ideally leads to a separation principle. inspired by (Loría, 2004b)

How much the perturbation affects the system as a whole depends on the robustness of the stability in the system. The goal in general is to control the system such that the the growth term $g(t, x)$ does not grow faster than the function term $f_1(t, x_1)$, thus causing instabilities in the total system. Several proofs for ensuring stability for NLTV cascaded systems are presented in Khalil (2002); Loría (2004a); Lamnabhi-Lagarrigue et al. (2004); Loría (2004b), where it is stated that from a "*robustness viewpoint*", the most useful forms of stabilities are uniform global asymptotic stability UGAS, and uniform local exponential stability ULES.

1. UGAS: *A system that behaves exactly the same independent of time, that can exist everywhere in relation to the desired stability point and stabilizes asymptotically towards the desired stability point.*

2. ULES: *A system that behaves exactly the same independent of time, that must exist within a local boundary of the desired stability point and stabilizes exponentially towards the desired stability point.*

From a practical point of view these are considered as mathematically ideal, and therefore not possible to achieve in real practical systems.Practical stabilities as introduced in Kristiansen (2008) are described as uniform practical asymptotical stability UPAS, and uniform semi-global practical asymptotical stability USPAS. Practical stability can in a sense be described as illustrated in Figure 1.6, where the desired position P_d is enveloped by a sphere or "ball" of a certain size, where the size of the sphere is dependant on the control gains introduced in the system. The sphere represents the area of position that is actually attainable for the quadrotor, referenced with P_d. This is due to the unknown conditions that surrounds P_d which from a reality viewpoint makes it impossible for the controllers to obtain position convergence, and consequently the quadrotor will hover around P_d within the sphere. However, in this thesis the stability criteria will be viewed as ideal, i.e. UGAS and ULES.

1.2.3 PD+ Passivity Based Control

In Ortega et al. (1998) its is stated that the term passivity-based control, (PBC) was first introduced in Ortega and Spong (1988) to "define a control methodology whose aim is to render the closed-loop passive". The GAS PD+ control scheme was introduced in Paden and Panja (1988) as an extension of the position controller presented in Takegaki and Arimoto (1981), Figure 1.7 and is stated as passive. The idea behind the PD+ is to decompose the controller into an inner PD loop and an outer dynamic compensation loop, made possible because the inertia matrix is *outside* the feedback loops, and by doing so the structure allows the simple PD computations to be run at a higher speed than the dynamic compensation loop in digital implementations. The control schemes' "+" term is the reference term denoted $\ddot{\boldsymbol{\theta}}_D M$, which is considered a reference signal. Examples of applications for the PD+ controller can be found in (Oland, 2014; Kristiansen, 2008; Schlanbusch, 2012).

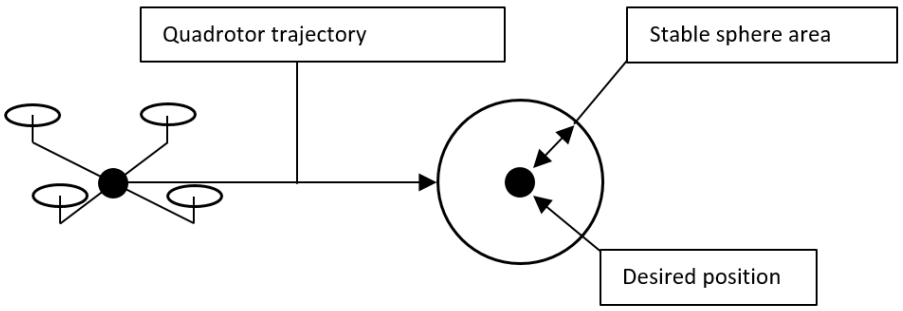


Figure 1.6: Illustration of practical stability for UPAS, Inspired by (Kristiansen, 2008)

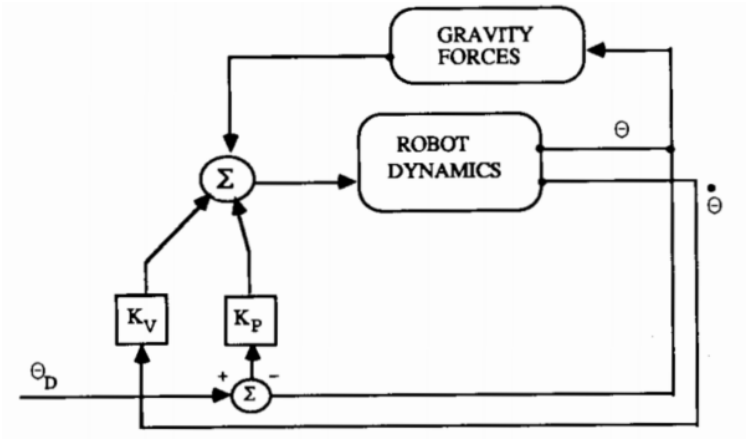


Figure 1.7: Takegaki and Arimoto's control scheme (Paden and Panja, 1988)

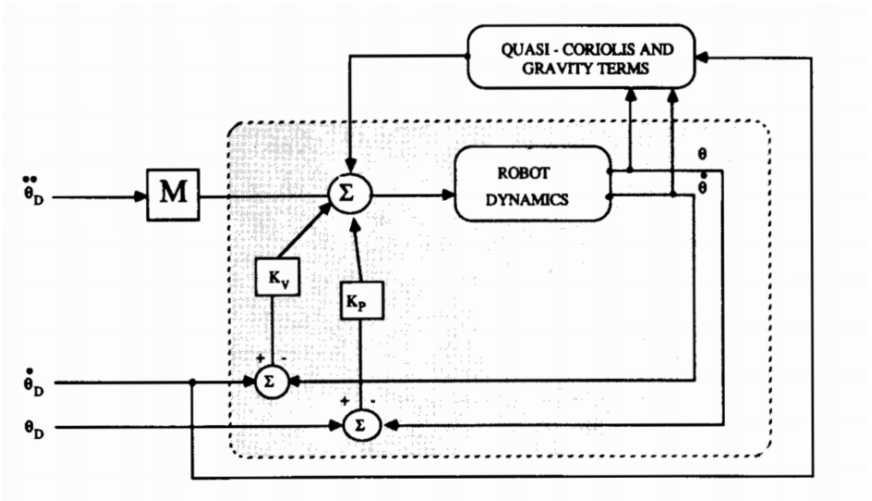


Figure 1.8: PD+ control scheme (Paden and Panja, 1988)

1.3 Contribution

In this thesis, a mathematical quadrotor model based on cascade modelling theory is presented. The cascaded interconnected system is proposed as a rotational cascade and a translational cascade with separate control solutions, where both are based on the passive PD+ control scheme. The basis for the control solutions in the cascaded systems are models of the the translation and rotation of a quadrotor in flight.

Initially, the rotation and translation of the quadrotor is presented decoupled from one another as stand-alone systems, where the control solutions are based on the assumptions made in accordance with the desired behavior of the systems. Then the two cascades are integrated into one cascaded interconnected system, where the final control solutions are presented such that the rotation and translation cascades work in union, allowing the quadrotor to be pointed in the desired flight direction and fly towards a desired position. A guidance generator is also presented such that the desired rotation, desired angular velocity and reference signal can be constructed.

1.4 Outline

This section provides a brief presentation of the contents of the thesis.

Chapter 2 contains mathematical definitions and notations, introduction to quaternion mathematics, preliminary rotational and translational kinematics and dynamics and explanations of the reference frames.

Chapter 3 has three main sections, where the first contains the main mathematical results of the kinematics, dynamics and control solution for the rotational cascade, the second contains the main mathematical results of the kinematics, dynamics and control solution for the translational cascade, and the third contains the main mathematical results of the kinematics, dynamics and control solution for the total cascaded interconnected system and the guidance generator.

Chapter 4 contains the simulation results for the rotation and translation cascades, and the total cascaded interconnected system.

Chapter 5 contains the concluding remarks and suggestions for future work.

1.5 Delimitations

This thesis does not consider actuator dynamics in the quadrotor model, and stability proofs for the controllers and the cascade system growth term has not been conducted.

Chapter 2

Preliminaries

This chapter contains descriptions of notations, definitions, introduction of quaternion mathematics, reference frames, and descriptions of dynamics and kinematics of the rotation and translation of a quadrotor.

2.1 Notations and Definitions

In this paper the notations that are used are inspired by the notations used in Oland (2014), where bold small case letters are vectors $\mathbf{x} \in \mathbb{R}^n$, bold capital letters are matrices $\mathbf{X} \in \mathbb{R}^{n\times m}$, and scalar values are represented as non-bold. The time derivatives of vectors and matrices are represented in a similar fashion to the representation in linear systems and control engineering in general. Where the time derivative of a vector is denoted as $\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt}$ and $\dot{\mathbf{X}} = \frac{d\mathbf{X}}{dt}$ for matrices, and the length of a vector is represented as $\|\mathbf{x}\| = \sqrt{\mathbf{x}^T\mathbf{x}}$. Superscripts when used with vectors denote the reference frame in which the vector exist such that $\mathbf{x}^B$ is a vector in frame B. For rotation matrices the notation is written as $\mathbf{R}_A^B \in \mathcal{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^T\mathbf{R} = \mathbf{I},\ det(\mathbf{R}) = 1\}$, which rotates a vector from frame A to frame B and where $\mathbf{I}$ is the identity matrix of sufficient dimensions depening on the context. The vector representing the angular velocity is denoted $\boldsymbol{\omega}_{B,A}^C \in \mathbb{R}^3$, which represents the angular velocity of frame A. relative to frame B referenced in frame C. Adding together angular velocities from different frames can be done as $\boldsymbol{\omega}_{A,D}^B = \boldsymbol{\omega}_{A,C}^B + \boldsymbol{\omega}_{C,D}^B$ (Egeland and Gravdahl, 2002). The time derivative of the rotation matrix is found as $\dot{\mathbf{R}}_A^B = \mathbf{R}_A^B\mathbf{S}(\boldsymbol{\omega}_{B,A}^A)$, where the cross product operator $\mathbf{S}(\cdot)$ is a skew symmetric matrix, and is such that for two arbitrary vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$, $\mathbf{S}(\mathbf{v}_1)\mathbf{v}_2 = \mathbf{v}_1 \times \mathbf{v}_2$, $\mathbf{S}(\mathbf{v}_1)\mathbf{v}_2 = -\mathbf{S}(\mathbf{v}_2)\mathbf{v}_1$, $\mathbf{S}(\mathbf{v}_1)\mathbf{v}_1 = 0$ and $\mathbf{v}_1^T\mathbf{S}(\mathbf{v}_2)\mathbf{v}_1 = 0$ and with $\mathbf{v}_1 = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}^T$ the skew symmetric cross-product operator matrix is defined as

$$\mathbf{S}(\mathbf{v}_1) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}. \tag{2.1}$$

When it comes to the parametrization of the rotation matrix $\mathbf{R}$, it is shown in Egeland and Gravdahl (2002) that this can be done by using a *unit quaternion vector* as invented by Hamilton in 1844 (Oland, 2014). A quaternion is represented by a vector

$$\mathbf{q} = \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{2.2}$$

where $\mathbf{q} \in \mathbb{R}^4$, η is the scalar part and $\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T$ is the vector part of the quaternion. A quaternion rotation from frame A to frame B is denoted

$$\mathbf{q}_{B,A} = \begin{bmatrix} \eta_{B,A} & \boldsymbol{\epsilon}^T \end{bmatrix}^T \tag{2.3}$$

where $\mathbf{q}_{B,A}\mathbf{q}_{B,A}^T = 1$, and η and $\boldsymbol{\epsilon}$ are the Euler parameters

$$\eta_{B,A} = \cos\frac{\theta_{B,A}}{2}, \qquad \boldsymbol{\epsilon} = \boldsymbol{k}\sin\frac{\theta_{B,A}}{2}, \tag{2.4}$$

which performs a rotation of an angle θ around the unit vector $\boldsymbol{k}_{B,A}$, and where the inverse quaternion of $\mathbf{q}_{B,A}$ is defined as

$$\mathbf{q}_{A,B} = \bar{\mathbf{q}}_{B,A}, \tag{2.5}$$

where

$$\bar{\mathbf{q}}_{B,A} = \begin{bmatrix} \eta_{B,A} & -\boldsymbol{\epsilon}_{B,A}^T \end{bmatrix}^T. \tag{2.6}$$

In terms of Euler parameters this enables the rotation matrix to be constructed as

$$\mathbf{R}_A^B = \mathbf{I} + 2\eta_{B,A}\mathbf{S}(\boldsymbol{\epsilon}_{B,A}) + 2\mathbf{S}^2(\boldsymbol{\epsilon}_{B,A}), \tag{2.7}$$

and for a general quaternion denoted $\mathbf{q} = \begin{bmatrix} \eta & \epsilon_1 & \epsilon_2 & \epsilon_3 \end{bmatrix}^T$ a rotation matrix can be written as

$$\mathbf{R} = \begin{bmatrix} \eta^2 + \epsilon_1^2 - \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_1\epsilon_2 - \eta\epsilon_3) & 2(\epsilon_1\epsilon_3 + \eta\epsilon_2) \\ 2(\epsilon_1\epsilon_2 + \eta\epsilon_3) & \eta^2 - \epsilon_1^2 + \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_2\epsilon_3 - \eta\epsilon_1) \\ 2(\epsilon_1\epsilon_3 - \eta\epsilon_2) & 2(\epsilon_2\epsilon_3 + \eta\epsilon_1) & \eta^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2 \end{bmatrix}. \tag{2.8}$$

It is important that the resulting quaternion maintains the unit length property , which is ensured by normalizing the composite quaternion rotations using first the quaternion product presented in Egeland and Gravdahl (2002) as

$$\mathbf{q}_{A,C} = \mathbf{q}_{A,B} \otimes \mathbf{q}_{B,C}, \tag{2.9}$$

or

$$\mathbf{q}_{A,C} = \mathbf{T}(\mathbf{q}_{A,B})\mathbf{q}_{B,C}, \tag{2.10}$$

then normalizing the composite rotation as

$$\mathbf{q}_{d,b} = \|\mathbf{q}_{d,b}\|^{-1}\mathbf{q}_{d,b} \tag{2.11}$$

where the operator $\mathbf{T}(\cdot)$ is defined as

$$\mathbf{T}(\mathbf{q}_{A,B}) = \begin{bmatrix} \eta_{A,B} & -\boldsymbol{\epsilon}_{A,B}^T \\ \boldsymbol{\epsilon}_{A,B} & \eta_{A,B}\mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}_{A,B}) \end{bmatrix}, \tag{2.12}$$

and the quaternion kinematics is given as

$$\dot{\mathbf{q}}_{A,B} = \frac{1}{2}\mathbf{q}_{A,B} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{A,B}^B \end{bmatrix} \tag{2.13}$$

or as

$$\dot{\mathbf{q}}_{A,B} = \frac{1}{2}\mathbf{T}(\mathbf{q}_{A,B}) \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{A,B}^B \end{bmatrix}. \tag{2.14}$$

2.2 Reference frames

In order to describe the rotational and translational dynamics of a quadrotor a set of basic reference frames are required.

North East Down (NED): The NED Frame is denoted $\mathcal{F}^n$, and is treated as the inertial frame. The x^n axis points North, y^n points East, while z^n completes the right handed reference system by pointing downward to the center of the Earth. The NED frame is chosen as the inertial frame due to the fact that flying with low speed in a local region, the centripetal and Coriolis effects of the Earth can be ignored (Oland, 2014), enabling the Laws of Newton to become valid. This method is called flat-Earth approximation.

Body Frame: The Body Frame is denoted $\mathcal{F}^b$, and is the origin in the center of mass of the rigid body. The x^b axis is aligned with rotor 2 and 4, the y^b axis is aligned with rotor 1 and 3, and to complete the body reference system the z^b axis points down through the center of mass, as shown in Figure 2.1.

Desired Frame: The Desired Frame is denoted $\mathcal{F}^d$, and describes the orientation that is desired for the quadrotor to obtain.

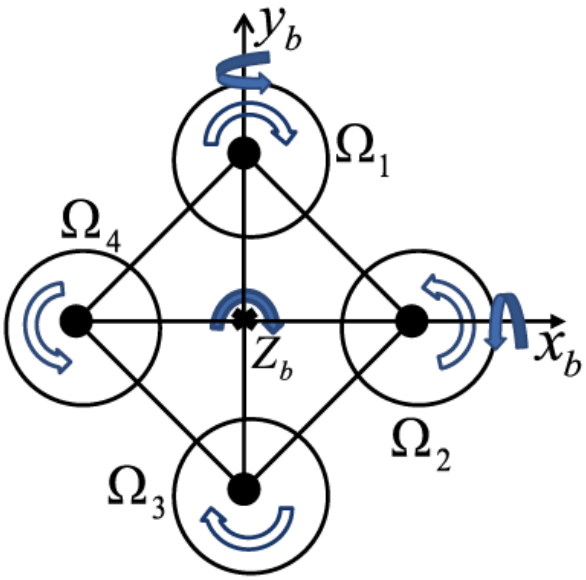


Figure 2.1: Scheme of quadrotor body axes, (Ailon and Arogeti, 2015)

2.3 Translation Dynamics and Kinematics

The translational dynamics and kinematics of a quadrotor is represented in Mahoney et al. (2012) with modified notation as

$$\dot{\mathbf{p}}^n = \mathbf{v}^n \tag{2.15}$$

$$\dot{\mathbf{v}}^n = -\frac{1}{m}\mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ \Upsilon \end{bmatrix}^b + g^n, \tag{2.16}$$

where $\mathbf{p}^n$ is the position vector, $\mathbf{v}^n$ is the velocity vector and $\dot{\mathbf{v}}^n$ is the acceleration in the NED frame. $\mathbf{R}_b^n$ represents the rotation matrix from Body Frame to NED frame, Υ^b is the total thrust along the z^b-axis, g^n is the gravitational acceleration constant along the z^n-axis and m is the mass of the quadrotor. The total thrust Υ^b can be written as

$$\Upsilon = m\ddot{z}_d - k_p(z_n - z_d) - k_d(\dot{z}_n - \dot{z}_d) + mg \tag{2.17}$$

where k_p and k_d are gains, z_n is the z-component of the position in NED frame, z_d is the z-component of the desired position and mg is the compensation term for the gravitational effects on the quadrotor. The position and velocity errors are found as

$$\tilde{z} = z_d - z_n \tag{2.18}$$

$$\dot{\tilde{z}} = \dot{z}_d - \dot{z}_n \tag{2.19}$$

inserted into (2.17) giving the total thrust as

$$\Upsilon = m\ddot{z}_d - k_p\tilde{z} - k_d\dot{\tilde{z}} + mg. \tag{2.20}$$

Υ is presented negative as a consequence of the z^b-axis' positive direction is defined downwards.

2.4 Rotation Dynamics and Kinematics

2.4.1 Dynamics and Kinematics

The rotational kinematics are described as the quaternion rotation from Body Frame to NED Frame (Oland, 2014; Egeland and Gravdahl, 2002) as

$$\mathbf{q}_{n,b}, \tag{2.21}$$

where the rotation dynamics are found using Euler's momentum equation as (Oland, 2014; Stengel, 2014)

$$\mathbf{J}\dot{\boldsymbol{\omega}}^b_{n,b} = -\mathbf{S}(\boldsymbol{\omega}^b_{n,b})\mathbf{J}\boldsymbol{\omega}^b_{n,b} + \boldsymbol{\tau}^b, \tag{2.22}$$

where $\boldsymbol{\tau}^b$ is a control signal.

2.4.2 Error Dynamics and Kinematics

The rotational error kinematics are found as the composite quaternion rotation from Body Frame to Desired Frame as described in (2.9), (2.13) and (2.14) but rewritten with the correct notations as

$$\mathbf{q}_{d,b} = \mathbf{q}_{d,n} \otimes \mathbf{q}_{n,b}, \tag{2.23}$$

also known as the error quaternion. If the system is asymptotically stable the error quaternion will go towards unity, i.e.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T,$$

however, to avoid misinterpretations in simulation plots, the error quaternion can be defined as for positive rotation(*clockwise*)

$$\mathbf{e}_{q+} = \begin{bmatrix} 1 - \eta_{d,b} & \boldsymbol{\epsilon}^T_{d,b} \end{bmatrix}^T \tag{2.24}$$

and similarly for negative rotation(*counter clockwise*)

$$\mathbf{e}_{q-} = \begin{bmatrix} 1 + \eta_{d,b} & \boldsymbol{\epsilon}^T_{d,b} \end{bmatrix}^T \tag{2.25}$$

assuring that the the quaternion error $\rightarrow \mathbf{0}$. The error kinematics are found as

$$\dot{\mathbf{q}}_{d,b} = \frac{1}{2}\mathbf{q}_{d,b} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{d,b} \end{bmatrix} \tag{2.26}$$

or as

$$\dot{\mathbf{q}}_{d,b} = \frac{1}{2}\mathbf{T}(\mathbf{q}_{d,b}) \begin{bmatrix} 0 \\ \boldsymbol{\omega}^b_{d,b} \end{bmatrix}, \tag{2.27}$$

where the angular velocity error $\boldsymbol{\omega}^b_{d,b}$ is described as

$$\boldsymbol{\omega}^b_{d,b} = \boldsymbol{\omega}^b_{d,n} + \boldsymbol{\omega}^b_{n,b} = \boldsymbol{\omega}^b_{n,b} - \mathbf{R}^b_d\boldsymbol{\omega}^d_{n,d}. \tag{2.28}$$

The angular acceleration in regard to the rotational dynamics is found using Euler's momentum equation as (Oland, 2014; Stengel, 2014)

$$\mathbf{J}\dot{\boldsymbol{\omega}}^b_{d,b} = -\mathbf{S}(\boldsymbol{\omega}^b_{n,b})\mathbf{J}\boldsymbol{\omega}^b_{n,b} + \boldsymbol{\tau}^b + \mathbf{J}\mathbf{S}(\boldsymbol{\omega}^b_{d,b})\mathbf{R}^b_d\boldsymbol{\omega}^d_{n,d} - \mathbf{J}\mathbf{R}^b_d\dot{\boldsymbol{\omega}}^d_{n,d} \tag{2.29}$$

where $\boldsymbol{\tau}^b$ is the control signal and kan be written as

$$\boldsymbol{\tau}^b = \mathbf{S}(\boldsymbol{\omega}^b_{n,b})\mathbf{J}\boldsymbol{\omega}^b_{n,b} + \mathbf{J}\mathbf{R}^b_d\dot{\boldsymbol{\omega}}^d_{n,d} - k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}^b_{d,b}, \tag{2.30}$$

and $\mathbf{J}^b$ is the inertia matrix defined as

$$\mathbf{J}^b = \begin{bmatrix} J_{xx} & 0 & 0 \\ 0 & J_{yy} & 0 \\ 0 & 0 & J_{zz} \end{bmatrix} \tag{2.31}$$

where J_{xx}, J_{yy} and J_{zz} represents the constant inertia components along the $\{x^b, y^b, z^b\}$-axes of the quadrotor.

Chapter 3

Main Results

In this chapter the main mathematical results of the quadrotor cascaded interconnected system is presented. The chapter contains three main sections, where the first addresses the decoupling of the quadrotor's rotational motion, the second deals with the decoupling of the quadrotor's translational motion, and the third where the two cascades are joined together to form the total cascaded interconnected system. The main reason for using a cascaded approach when it comes to mathematical modeling is, as stated in section 1.4, that by breaking a larger system into smaller subsystems (or cascades), proving system stability becomes much easier in terms of mathematical calculations. By proving that cascades are stable it follows, in accordance with cascade theory, that if every cascade in a system is stable then the whole system becomes stable.

3.1 Rotational Cascade

In this section the rotation of the quadrotor is addressed. The goal is to derive a mathematical model describing the rotation and prove its stability. The problem is to control the quadrotor relative to an inertial frame. The Body Frame is defined to coincide with the moments of inertia with an origin in the quadrotors center of mass, while the NED Frame is treated as the inertial frame. By utilizing the the equations (2.23), (2.27), (2.28), (2.29), (2.30), (2.31) the stand-alone rotation can be, with modification, obtained.

3.1.1 Kinematics and Dynamics

If it is assumed that no translational motion occurs during the rotation and optimal conditions, (i.e. no wind, blade flapping etc.) the desired quaternion $\mathbf{q}_{n,d}$ can be defined as *constant*. First, the expressions for the quaternion and angular velocity error are obtained. Using (2.10) the quaternion error is written as

$$\mathbf{q}_{d,b} = \mathbf{T}(\mathbf{q}_{d,n})\mathbf{q}_{n,b}$$

with the desired quaternion defined as (2.5) and (2.6) and the quaternion operator $\mathbf{T}(\mathbf{q}_{d,n})$ as (2.12). With these results the error kinematics can be constructed according to (2.27) and combined with (2.12) as

$$\dot{\mathbf{q}}_{d,b} = \frac{1}{2}\begin{bmatrix} -\boldsymbol{\epsilon}_{d,b}^T\boldsymbol{\omega}_{d,b}^b \\ (\eta_{d,b}\mathbf{I} - \mathbf{S}(\boldsymbol{\epsilon}_{d,b}))\boldsymbol{\omega}_{d,b}^b \end{bmatrix}. \tag{3.1}$$

The angular velocity error can be obtained from (2.28), which is defined as

$$\boldsymbol{\omega}_{d,b}^b = \boldsymbol{\omega}_{n,b}^b - \mathbf{R}_d^b\boldsymbol{\omega}_{n,d}^d.$$

However, since $\mathbf{q}_{n,d}$ is considered constant, the rotation

$$\mathbf{R}_d^b\boldsymbol{\omega}_{n,d}^d = 0$$

and the angular velocity error can then be written as

$$\boldsymbol{\omega}_{d,b}^{b} = \boldsymbol{\omega}_{n,b}^{b}.$$

Knowing this the rotation dynamics presented in (2.29) can be rewritten as

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{d,b}^{b} = -\mathbf{S}(\boldsymbol{\omega}_{n,b}^{b})\mathbf{J}\boldsymbol{\omega}_{n,b}^{b} + \boldsymbol{\tau}^{b} \tag{3.2}$$

where the derivative of the angular velocity error can be presented as

$$\dot{\boldsymbol{\omega}}_{d,b}^{b} = \dot{\boldsymbol{\omega}}_{n,b}^{b}$$

enabling the rotational dynamics in (3.2) to be rewritten as

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{n,b}^{b} = -\mathbf{S}(\boldsymbol{\omega}_{n,b}^{b})\mathbf{J}\boldsymbol{\omega}_{n,b}^{b} + \boldsymbol{\tau}^{b}. \tag{3.3}$$

and the error dynamics as

$$\dot{\boldsymbol{\omega}}_{n,b}^{b} = \mathbf{J}^{-1}(-\mathbf{S}(\boldsymbol{\omega}_{n,b}^{b})\mathbf{J}\boldsymbol{\omega}_{n,b}^{b} + \boldsymbol{\tau}^{b}). \tag{3.4}$$

3.1.2 Control Design

The controller is designed according to the principle of PD+ which is described in section 1.6.1, where instead of a standard PD control scheme, the PD+ offers a reference in that the PD-response is to follow. By using the rotation dynamics found in (3.3) a control law is constructed as a function of the quaternion and angular velocity errors as

$$\boldsymbol{\tau}^{b} = \mathbf{S}(\boldsymbol{\omega}_{n,b}^{b})\mathbf{J}\boldsymbol{\omega}_{n,b}^{b} - k_p\mathbf{q}_{d,b} - k_d\boldsymbol{\omega}_{n,b}^{b}, \tag{3.5}$$

where the error quaternion is represented by the vector part $\boldsymbol{\epsilon}_{d,b}$ and the control law is rewritten as

$$\boldsymbol{\tau}^{b} = \mathbf{S}(\boldsymbol{\omega}_{n,b}^{b})\mathbf{J}\boldsymbol{\omega}_{n,b}^{b} - k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}_{n,b}^{b}. \tag{3.6}$$

Inserting the control signal (3.6) into (3.3) the control law is obtained on the form $\dot{x} = -k_p x_1 - k_d x_2$ as

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{n,b}^{b} = -k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}_{n,b}^{b} \tag{3.7}$$

where k_p and k_d are gains. The PD+ approach requires a reference for the system to follow, which usually comes in the form of the desired angular acceleration vector. However, since the system only rotates around its own body axes with a constant velocity, the acceleration is, in this case 0, and subsequently the controller can be represented as written in (3.7).

3.1.3 Block Diagram Model

With the kinematics, dynamics and controller in place a block diagram representation of the quadrotor model is made to visually show the rotation cascade of the quadrotor.

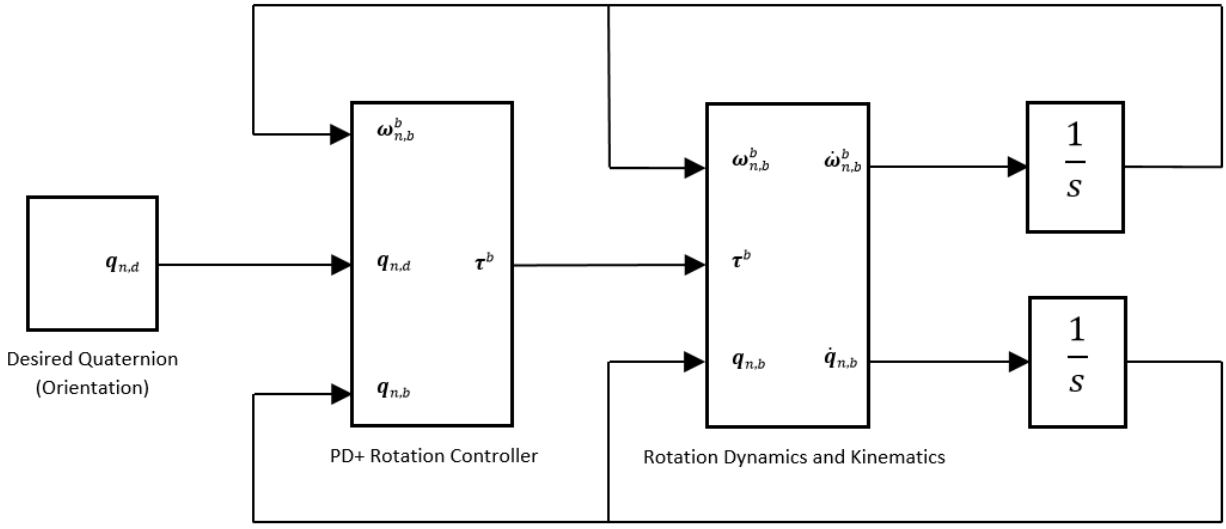


Figure 3.1: Illustration of the Rotation Cascade System.

3.2 Translational Cascade

In this section the control and stability of the translational motion of the quadrotor is addressed. As in section 3.1.1 a mathematical model based on its translational motion is derived and the stability of the controller is verified.

3.2.1 Kinematics and Dynamics

As described in (2.15) and (2.16) the translation kinematics and dynamics are written as

$$\begin{aligned}\dot{\mathbf{p}}^n &= \mathbf{v}^n \\ \dot{\mathbf{v}}^n &= -\frac{1}{m}\mathbf{R}^n_b \begin{bmatrix}0\\0\\\Upsilon\end{bmatrix}^b + \begin{bmatrix}0\\0\\g\end{bmatrix}^n.\end{aligned}$$

Decoupled from the rotation, translational motion will only be achievable along the z-axis of the Body Frame where the dynamics are represented as

$$\dot{\mathbf{v}}^n = -\frac{1}{m}\mathbf{I} \begin{bmatrix}0\\0\\\Upsilon\end{bmatrix}^b + \begin{bmatrix}0\\0\\g\end{bmatrix}^n \tag{3.8}$$

However, if it is assumed that the rotation matrix $\mathbf{R}^n_b$ is perfect, the rotation control can be performed directly by the thrust vector $\boldsymbol{\Upsilon}^b$, allowing the necessary rotation to occur. The dynamics can then be rewritten as

$$\dot{\mathbf{v}}^n = -\frac{1}{m}\mathbf{I} \begin{bmatrix}\Upsilon_x\\\Upsilon_y\\\Upsilon_z\end{bmatrix}^n + \begin{bmatrix}0\\0\\g\end{bmatrix}^n \tag{3.9}$$

3.2.2 Control Design

As in section 3.1.2 the translational controller is also constructed as a PD+ control scheme. The control of the quadroor will be divided into two parts, where the control signals of the translation are the total thrust Υ and the thrust vector $\boldsymbol{\Upsilon}^b$ and is constructed as a function of the position, velocity and acceleration of the

translational motion in NED frame $\mathbf{p}^n$ and the desired position, velocity and acceleration in NED frame $\mathbf{p}_d^n$. The desired position as a function of the time t is defined as

$$\mathbf{p}_d^n(t) = \begin{bmatrix} r\sin(\omega t) \\ r\cos(\omega t) \\ z_d \end{bmatrix} \tag{3.10}$$

where r is the radius (in meters), ω is the angular velocity of the *sine*, and *cosine* functions (rad/s) and z is the desired altitude (in meters). The desired velocity and acceleration is found as the first and second derivative of $\mathbf{p}_d^n$ as

$$\dot{\mathbf{p}}_d^n(t) = \begin{bmatrix} \omega r\cos(\omega t) \\ -\omega r\sin(\omega t) \\ 0 \end{bmatrix} \tag{3.11}$$

$$\ddot{\mathbf{p}}_d^n(t) = \begin{bmatrix} -\omega^2 r\sin(\omega t) \\ -\omega^2 r\cos(\omega t) \\ 0 \end{bmatrix}. \tag{3.12}$$

For a Helix trajectory the desired position, velocity and acceleration can be defined as

$$\mathbf{p}_d^n(t) = \begin{bmatrix} r\sin(\omega t) \\ r\cos(\omega t) \\ -t \end{bmatrix}$$

$$\dot{\mathbf{p}}_d^n(t) = \begin{bmatrix} \omega r\cos(\omega t) \\ -\omega r\sin(\omega t) \\ -1 \end{bmatrix}$$

$$\ddot{\mathbf{p}}_d^n(t) = \begin{bmatrix} -\omega^2 r\sin(\omega t) \\ -\omega^2 r\cos(\omega t) \\ 0 \end{bmatrix},$$

and a spiral trajectory can be represented as

$$\mathbf{p}_d^n(t) = \begin{bmatrix} (1+t)\sin(\omega t) \\ (1+t)\cos(\omega t) \\ -t \end{bmatrix}$$

$$\dot{\mathbf{p}}_d^n(t) = \begin{bmatrix} \sin(\omega t) + \omega(1+t)\cos(\omega t) \\ \cos(\omega t) - \omega(1+t)\sin(\omega t) \\ -1 \end{bmatrix}$$

$$\ddot{\mathbf{p}}_d^n(t) = \begin{bmatrix} 2\omega\cos(\omega t) - \omega^2(1+t)\sin(\omega t) \\ -2\omega\sin(\omega t) - \omega^2(1+t)\cos(\omega t) \\ 0 \end{bmatrix}.$$

Total Thrust Control

Using $\mathbf{p}_d^n(t)$, $\dot{\mathbf{p}}_d^n(t)$ and $\ddot{\mathbf{p}}_d^n(t)$ the total thrust Υ^b for the decoupled translational motion is found as a modification of (2.17) as

$$\Upsilon = m\ddot{z}_d - k_p(p_{d_z}^n - p_z^n) - k_d(\dot{p}_{d_z}^n - \dot{p}_z^n) + mg, \tag{3.13}$$

where k_p and k_d are gains, p_z^n is the z-component of $\mathbf{p}^n$, $p_{d_z}^n$ is the z-component of $\mathbf{p}_d^n(t)$. Inserting the position and velocity errors into (2.18) and (2.19), the errors are rewritten as

$$\tilde{z} = p_{d_z}^n - p_z^n$$
$$\dot{\tilde{z}} = \dot{p}_{d_z}^n - \dot{p}_z^n,$$

with the acceleration $\ddot{z}_d$ as

$$\ddot{z}_d = \ddot{p}_{d_z}^n \tag{3.14}$$

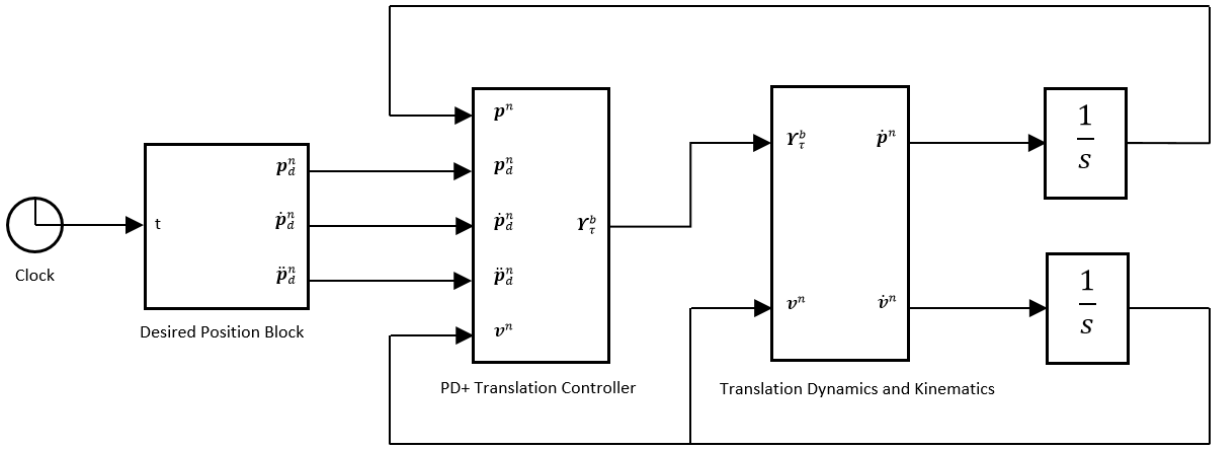


Figure 3.2: Illustration of the Translation Cascade system

Using these alterations and inserting them into (2.17), the total thrust is found as

$$\Upsilon = m\ddot{z}_d - k_p\tilde{z} - k_d\dot{\tilde{z}} + mg.$$

Thrust Vector Control
The thrust vector $\boldsymbol{\Upsilon}^b$ is found by inserting $\mathbf{p}_d^n(t)$, $\dot{\mathbf{p}}_d^n(t)$ and $\ddot{\mathbf{p}}_d^n(t)$ into (2.17) as

$$\boldsymbol{\Upsilon}_\tau^b = m\ddot{\mathbf{p}}_d^n - k_p(\mathbf{p}_d^n - \mathbf{p}^n) - k_d(\dot{\mathbf{p}}_d^n - \dot{\mathbf{p}}^n) + mg_z^n, \tag{3.15}$$

where the position error, velocity error and acceleration is defined as

$$\mathbf{p}_{err} = \mathbf{p}_d^n - \mathbf{p}^n \tag{3.16}$$
$$\mathbf{v}_{err} = \dot{\mathbf{p}}_d^n - \dot{\mathbf{p}}^n \tag{3.17}$$
$$\mathbf{a}_d^n = \ddot{\mathbf{p}}_d^n. \tag{3.18}$$

Inserting (3.16),(3.17),(3.18) into (3.15) the thrust vector is finally written as

$$\boldsymbol{\Upsilon}_\tau^b = m\mathbf{a}_d^n - k_p\mathbf{p}_{err} - k_d\mathbf{v}_{err} + mg \tag{3.19}$$

3.2.3 Block Diagram Model

Having found expressions for the translational dynamics and kinematics and a control law, a block diagram model is made to show the translation cascade system.

3.3 Total Cascade Interconnected System

In this section the decoupled cascades are joined together. where the rotation, translation and controller designs are reconstructed, such that, when given a desired position in $\mathbb{R}^3$ the quadrotor will rotate the translational motion along the desired trajectory to obtain the desired position, where the desired rotations, angular velocities and angular accelerations are generated by a guidance system.

3.3.1 Rotation Kinematics and Dynamics

Unlike the assumption for the decoupled rotation in section 3.1.1, the desired quaternion $\mathbf{q}_{n,d}$ for the total system can not be considered constant. The definition of the error quaternion is still as presented in (2.23)

$$\mathbf{q}_{d,b} = \mathbf{q}_{d,n} \otimes \mathbf{q}_{n,b},$$

with the desired rotation from NED Frame to Desired Frame written as (2.5)

$$\mathbf{q}_{d,n} = \bar{\mathbf{q}}_{n,d},$$

where $\bar{\mathbf{q}}_{n,d}$ is found as (2.6)

$$\bar{\mathbf{q}}_{n,d} = \begin{bmatrix} \eta_{n,d} & -\boldsymbol{\epsilon}_{n,d}^T \end{bmatrix}^T$$

The error dynamics however will not be represented as in the decoupled system. With $\mathbf{q}_{n,d}$ as a variable, the angular velocity error must be represented as shown in (2.28) as

$$\boldsymbol{\omega}_{d,b}^b = \boldsymbol{\omega}_{n,b}^b - \mathbf{R}_d^b \boldsymbol{\omega}_{n,d}^d$$

with the error dynamics as (2.29)

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{d,b}^b = -\mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{J}\boldsymbol{\omega}_{n,b}^b + \boldsymbol{\tau}^b + \mathbf{J}\mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{R}_d^b\boldsymbol{\omega}_{n,d}^d - \mathbf{J}\mathbf{R}_d^b\dot{\boldsymbol{\omega}}_{n,d}^d.$$

3.3.2 Rotation Controller Design

Although the error dynamics are represented with the control signal $\boldsymbol{\tau}^b$, the rotation that requires a control solution is the rotation $\mathbf{q}_{n,b}$, due to the fact that the quadrotor has to rotate from Body Frame to NED frame. which is represented by the rotation dynamics of $\boldsymbol{\omega}_{n,b}^b$ described in (3.3) as

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{n,b}^b = -\mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{J}\boldsymbol{\omega}_{n,b}^b + \boldsymbol{\tau}^b.$$

The control law vector $\boldsymbol{\tau}^b$ as a function of the error quaternion and rotation dynamics is chosen as a modification of (3.6) as

$$\boldsymbol{\tau}^b = \mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{J}\boldsymbol{\omega}_{n,b}^b - k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}_{d,b}^b,$$

and to keep a PD+ structure the reference $\mathbf{J}\mathbf{R}_d^b\dot{\boldsymbol{\omega}}_{n,d}^d$ found in the error dynamic equation (2.29) is added, reforming the control law to

$$\boldsymbol{\tau}^b = \mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{J}\boldsymbol{\omega}_{n,b}^b + \mathbf{J}\mathbf{R}_d^b\dot{\boldsymbol{\omega}}_{n,d}^d - k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}_{d,b}^b. \tag{3.20}$$

Inserting the control law into (3.3) the rotation dynamics of the total system is found as

$$\mathbf{J}\dot{\boldsymbol{\omega}}_{n,b}^b = \mathbf{J}\mathbf{R}_d^b\dot{\boldsymbol{\omega}}_{n,d}^d - k_p\boldsymbol{\epsilon}_{d,b} - k_d\boldsymbol{\omega}_{d,b}^b \tag{3.21}$$

With $\mathbf{q}_{n,b}$ controlled, the rotation matrix $\mathbf{R}_b^n$ can be constructed as (2.7)

$$\mathbf{R}_b^n = \mathbf{I} + 2\eta_{n,b}\mathbf{S}(\boldsymbol{\epsilon}_{n,b}) + 2\mathbf{S}^2(\boldsymbol{\epsilon}_{n,b}).$$

3.3.3 Translation Kinematics and Dynamics

The translational kinematics and dynamics are essentially the same as in section 3.2.1. However in the total system the translation is not decoupled from the rotation. The rotation matrix $\mathbf{R}_b^n$, which is produced in the rotation cascade is fed into the translation cascade and acts as a control signal that enables the thrust vector to be pointed in the desired direction such that the Body Frame aligns with the Desired Frame, and subsequently allows the quadrotor to fly towards the desired position $\mathbf{p}_d^n$, The kinematics and dynamics are represented as (2.15) and (2.16)

$$\begin{aligned} \dot{\mathbf{p}}^n &= \mathbf{v}^n \\ \dot{\mathbf{v}}^n &= -\frac{1}{m}\mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ \Upsilon \end{bmatrix}^b + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}^n. \end{aligned}$$

3.3.4 Translation Controller Design

Knowing that the rotation matrix $\mathbf{R}_b^n$ is not a perfect rotation, i.e. no instant rotation, the translation can not be controlled by the thrust vector that can be constructed using the desired position vector $\mathbf{p}_d^n$. As a result the translation controller for the total cascade interconnected system is represented as described in (3.13), (2.18), (2.19) and (2.20) as an altitude controller on the form

$$\Upsilon = m\ddot{z} - k_p\tilde{z} - k_d\dot{\tilde{z}} + mg.$$

However to obtain the correct thrust vector, the error position, error velocity and the reference acceleration (2.18), (2.19), (3.18) is rotated to the Body Frame where the errors

$$\mathbf{e}^b = (\mathbf{R}_b^n)^T(\mathbf{p}_d^n - \mathbf{p}^n) \tag{3.22}$$
$$\dot{\mathbf{e}}^b = (\mathbf{R}_b^n)^T(\dot{\mathbf{p}}_d^n - \dot{\mathbf{p}}^n) \tag{3.23}$$
$$\ddot{\mathbf{e}}^b = (\mathbf{R}_b^n)^T\ddot{\mathbf{p}}_d^n \tag{3.24}$$

where

$$\tilde{z}^b = \mathbf{e}_z^b \tag{3.25}$$
$$\dot{\tilde{z}}^b = \dot{\mathbf{e}}_z^b \tag{3.26}$$
$$\ddot{z}^b = \ddot{\mathbf{e}}_z^b \tag{3.27}$$

When considering rotation, the thrust needs to compensate for the rotation angles or else the quadrotor will fall to the ground. This can be done by modifying the thrust controller by dividing (3.13) with the zz-component of the rotation matrix $\mathbf{R}_b^n$, and finally inserting (3.25), (3.26), (3.27) into (3.13), reconstructing the controller as

$$\Upsilon = \frac{1}{\mathbf{R}_{b_{zz}}^n}(m\ddot{z}^b - k_p\tilde{z}^b - k_d\dot{\tilde{z}}^b + mg). \tag{3.28}$$

3.3.5 Guidance

To achieve the desired rotation in accordance with the change in position, three values must be generated, namely the desired rotation quaternion $\mathbf{q}_{n,d}$, the desired angular velocity $\boldsymbol{\omega}_{n,d}^d$, and the angular acceleration $\dot{\boldsymbol{\omega}}_{n,d}^d$. These are needed such that the quadrotor rotates correctly and points the thrust vector in the desired direction. This is accomplished by constructing a waypoint tracking system that creates the desired values using the position error between $\mathbf{p}^n$ and $\mathbf{p}_d^n$. As stated in (Oland, 2014) the position error in the Desired Frame for fixed-wing UAVs can be defined as

$$\mathbf{e}^d := \begin{bmatrix} \|\mathbf{e}^n\| \\ 0 \\ 0 \end{bmatrix} = \mathbf{R}_n^d\mathbf{e}^n = \mathbf{R}_n^d(\mathbf{p}_d^n - \mathbf{p}^n), \tag{3.29}$$

where the objective is to make $\mathbf{e}^d \to 0$. However, unlike a fixed-wing UAV which has its thrust vector aligned with its x^b-axis, a quadrotor UAV has its thrust vector aligned along its z^b-axis, and the waypoint error $\mathbf{e}^d$ is as a consequence redefined as

$$\mathbf{e}^d := \begin{bmatrix} 0 \\ 0 \\ -\|\mathbf{e}^n\| \end{bmatrix} = \mathbf{R}_n^d\mathbf{e}^n = \mathbf{R}_n^d(\mathbf{p}_d^n - \mathbf{p}^n), \tag{3.30}$$

where $\mathbf{p}_d^n - \mathbf{p}^n = \mathbf{e}^n$ is the position error between the desired and actual position in NED Frame and $\|\mathbf{e}^n\|$ is its norm. The desired quaternion rotation $\mathbf{q}_{n,d}$ is then obtained as (2.3), (2.4)

$$\mathbf{q}_{n,d} = \begin{bmatrix} \cos\left(\frac{\vartheta_{n,d}}{2}\right) & \mathbf{k}_{n,d}^T \sin\left(\frac{\vartheta_{n,d}}{2}\right) \end{bmatrix}^T, \tag{3.31}$$

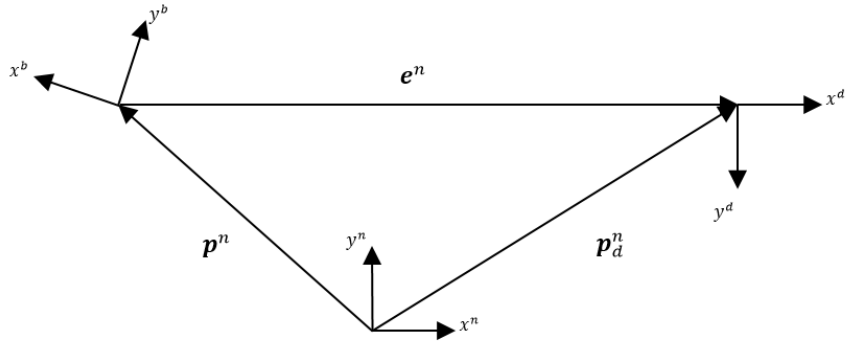


Figure 3.3: Illustration of the Position vectors in the xy-plane. Inspired by (Oland, 2014)

where the rotation angle $\vartheta_{n,d}$ and the quaternion vector $\mathbf{k}_{n,d}$ is constructed as

$$\vartheta_{n,d} = \frac{1}{3}\tan^{-1}\left(k\| \begin{bmatrix} e_x^n & e_y^n & 0 \end{bmatrix}^T \|\right) \qquad \mathbf{k}_{n,d} = \frac{\mathbf{e}^d \times \mathbf{e}^n}{\|\mathbf{e}^d \times \mathbf{e}^n\|}, \tag{3.32}$$

where k is a constant. Having obtained a solution for $\mathbf{q}_{n,d}$, an expression for the desired angular velocity must be attained. As shown in (3.29), the waypoint error is defined as $\mathbf{e}^d = \mathbf{R}_n^d \mathbf{e}^n$, and by differentiating $\mathbf{e}^d$ a preliminary expression containing the desired angular velocity is found as

$$\dot{\mathbf{e}}^d = \mathbf{R}_n^d \mathbf{S}(\boldsymbol{\omega}_{d,n}^n)\mathbf{e}^n + \mathbf{R}_n^d \dot{\mathbf{e}}^n. \tag{3.33}$$

To solve for the angular velocity the equation needs to be modified slightly and is rewritten as

$$\dot{\mathbf{e}}^d = -\mathbf{S}(\boldsymbol{\omega}_{d,n}^d)\mathbf{e}^d + \mathbf{R}_n^d \dot{\mathbf{e}}^n$$

and lastly with the angular velocity outside of the skew-symmetric matrix as

$$\dot{\mathbf{e}}^d = \mathbf{S}(\mathbf{e}^d)\boldsymbol{\omega}_{n,d}^d + \mathbf{R}_n^d \dot{\mathbf{e}}^n. \tag{3.34}$$

With an expression containing $\boldsymbol{\omega}_{n,d}^d$ obtained, the equation is then solved for the desired angular velocity, rewriting the expression as

$$\mathbf{S}(\mathbf{e}^d)\boldsymbol{\omega}_{n,d}^d = \dot{\mathbf{e}}^d - \mathbf{R}_n^d \dot{\mathbf{e}}^n. \tag{3.35}$$

Knowing that the skew-symmetric matrix with the components from the waypoint error does not have full rank and is constructed by its z-component alone the matrix is written as

$$\mathbf{S}(\mathbf{e}^d) = \begin{bmatrix} 0 & -\|\mathbf{e}^n\| & 0 \\ \|\mathbf{e}^n\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

This means that to solve for $\boldsymbol{\omega}_{n,d}^d$ the Moore-Penrose pseudo-inverse $\mathbf{S}^\dagger(\cdot)$ must be used, where

$$\mathbf{S}^\dagger(\mathbf{e}^d) = \begin{bmatrix} 0 & \frac{1}{\|\mathbf{e}^n\|} & 0 \\ -\frac{1}{\|\mathbf{e}^n\|} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and by inserting the the pseudo-inverse into (3.35) the expression becomes

$$\mathbf{S}^\dagger(\mathbf{e}^d)\mathbf{S}(\mathbf{e}^d)\boldsymbol{\omega}_{n,d}^d = \mathbf{S}^\dagger(\mathbf{e}^d)(\dot{\mathbf{e}}^d - \mathbf{R}_n^d \dot{\mathbf{e}}^n),$$

where $\mathbf{S}^\dagger(\mathbf{e}^d)\mathbf{S}(\mathbf{e}^d) = \mathbf{I}$ and

$$\mathbf{S}^\dagger(\mathbf{e}^d)\mathbf{S}(\dot{\mathbf{e}}^d) = \begin{bmatrix} 0 & \frac{1}{\|\mathbf{e}^n\|} & 0 \\ -\frac{1}{\|\mathbf{e}^n\|} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \|\dot{\mathbf{e}}^n\| \end{bmatrix} = 0$$

making the desired angular velocity

$$\boldsymbol{\omega}_{n,d}^d = -\mathbf{S}^\dagger(\mathbf{e}^d)\mathbf{R}_n^d\dot{\mathbf{e}}^n. \tag{3.36}$$

In addition to $\boldsymbol{\omega}_{n,d}^d$, the guidance system must also provide the rotational control reference, which is found by either differentiating $\boldsymbol{\omega}_{n,d}^d$ or by low pass filtering. Since differentiating $\boldsymbol{\omega}_{n,d}^d$ can be quite difficult, a filter solution is applied to approximate $\dot{\boldsymbol{\omega}}_{n,d}^d$ as

$$\frac{\tilde{\boldsymbol{\omega}}_{n,d}^d}{\boldsymbol{\omega}_{n,d}^d} = \frac{1}{Ts+1}. \tag{3.37}$$

Solving for $\boldsymbol{\omega}_{n,d}^d$ gives

$$\tilde{\boldsymbol{\omega}}_{n,d}^d(Ts+1) = \boldsymbol{\omega}_{n,d}^d,$$

rewriting to the extended form as

$$\tilde{\boldsymbol{\omega}}_{n,d}^d Ts + \tilde{\boldsymbol{\omega}}_{n,d}^d = \boldsymbol{\omega}_{n,d}^d,$$

and finally performing inverse laplace and solving for the derivative yields the filtered expression

$$\dot{\tilde{\boldsymbol{\omega}}}_{n,d}^d = \frac{\boldsymbol{\omega}_{n,d}^d - \tilde{\boldsymbol{\omega}}_{n,d}^d}{T} \tag{3.38}$$

where $\dot{\tilde{\boldsymbol{\omega}}}_{n,d}^d$ is the desired angular acceleration and T can be considered the sampling rate of the filter.

3.3.6 Total Cascade Interconnected System Model

Having found the modified expressions and control laws necessary to construct the final versions of the rotational and translational cascades, a model for the total system can be constructed. The cascade interconnected system can then be expressed as a modification of (1.1) and (1.2) on the form

$$\Sigma_1 : \dot{\mathbf{v}}^n = -\frac{1}{m}\mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ \Upsilon \end{bmatrix}^b + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}^n \tag{3.39}$$

$$\Sigma_2 : \mathbf{J}\dot{\boldsymbol{\omega}}_{n,b}^b = -\mathbf{S}(\boldsymbol{\omega}_{n,b}^b)\mathbf{J}\boldsymbol{\omega}_{n,b}^b + \boldsymbol{\tau}^b. \tag{3.40}$$

A block diagram model is made to show the total cascaded interconnected system.

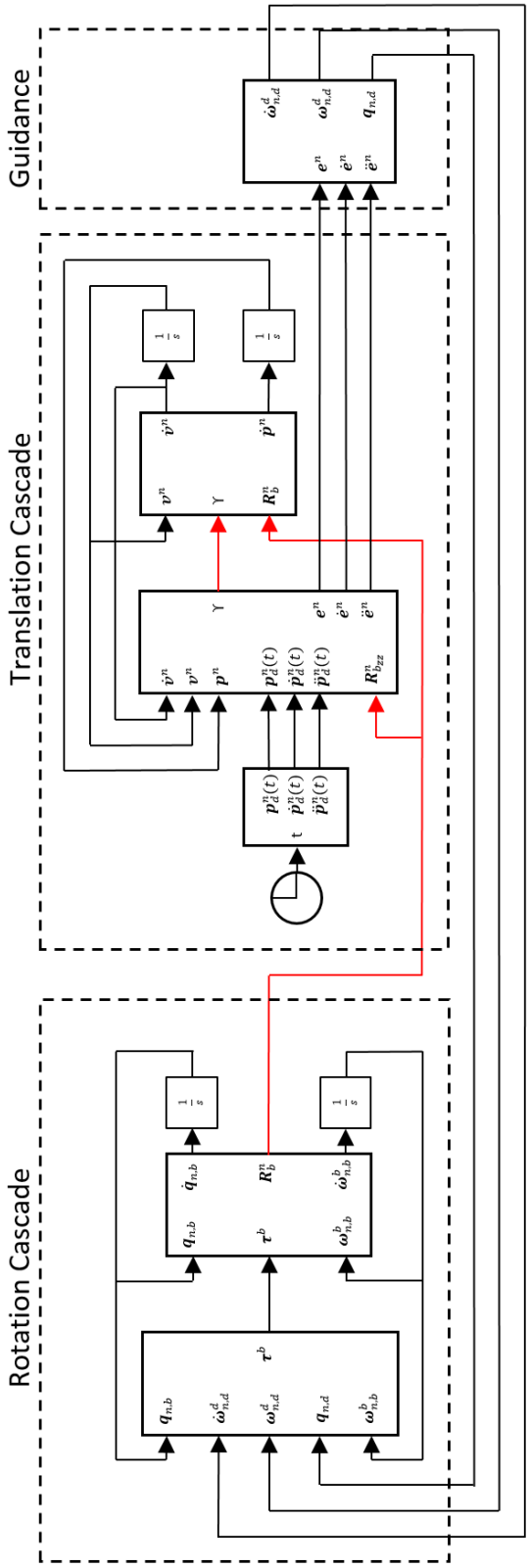


Figure 3.4: Block Model of the Total Cascade Interconnected System

Chapter 4

Simulation Results

To verify the control solutions that have been proposed in sections 3.1.2, 3.2.2 and 3.3.2, several simulations have been conducted. The quadrotor model that is used is a prototype that is under development in the University of Tromsoe (UiT) campus Narvik by Tor-Aleksander Johansen, shown in Figure 4.1, who has provided the inertia matrix values and mass of the quadrotor. The simulations are divided into three sections in accordance with chapter 3, where section 4.1 and 4.2 respectively holds the simulation results of the Rotation and Translation Cascades, and section 4.3 presents the simulation results of the total interconnected system. As a note, the z-axes in both NED and Body Frame are defined with positive direction pointing down, meaning that in the simulation results, positive movement along the z-axis shows the quadrotor flying downwards, and negative movement shows the quadrotor flying upwards.

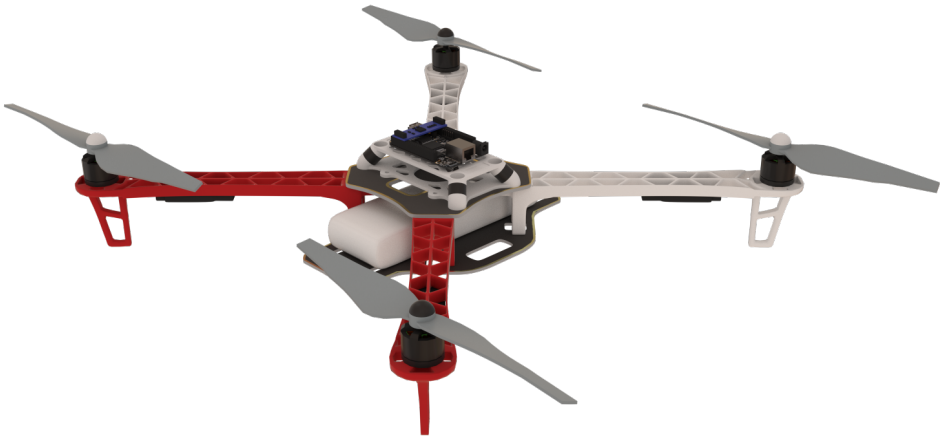


Figure 4.1: HiNrotor Prototype Used For Simulation, Illustration by Tom Stian Andersen

4.1 Rotational Cascade Simulation Results

Let the initial states be given as $\mathbf{q}_{n,b}(0) = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$, and $\boldsymbol{\omega}^b_{n,b}(0) = \mathbf{0}\ rad/s$, indicating that the quadrotor is *standig still* where the quadrotors inertia matrix is defined as

$$\mathbf{J} = \begin{bmatrix}0.047316 & 0 & 0\\ 0 & 0.048898 & 0\\ 0 & 0 & 0.539643\end{bmatrix}.$$

Three simulations are performed to test that for different rotations the controller holds. The three desired orientations corresponding with the angles $O_1 = \begin{bmatrix}30^\circ & 30^\circ & 30^\circ\end{bmatrix}$, $O_2 = \begin{bmatrix}30^\circ & 30^\circ & -30^\circ\end{bmatrix}$, $O_3 = \begin{bmatrix}-30^\circ & -30^\circ & -30^\circ\end{bmatrix}$ are

$$\mathbf{q}_{n,d}(1) = \begin{bmatrix}0.918558653543692 & 0.176776695296637 & 0.306186217847897 & 0.176776695296637\end{bmatrix}^T$$
$$\mathbf{q}_{n,d}(2) = \begin{bmatrix}0.883883476483185 & 0.306186217847897 & 0.176776695296637 & -0.306186217847897\end{bmatrix}^T$$
$$\mathbf{q}_{n,d}(3) = \begin{bmatrix}0.883883476483185 & -0.306186217847897 & -0.176776695296637 & -0.306186217847897\end{bmatrix}^T.$$

The control gains are set to $k_p = 20$ and k_d=12. In these simulations the error dynamics are denoted $\boldsymbol{\omega}^b_{n,b}$ since $\boldsymbol{\omega}^b_{d,b} = \boldsymbol{\omega}^b_{n,b}$ for the decoupled rotational cascade.

4.1.1 Simulation Case 1

This section contains the simulation results for the desired quaternion rotation $\mathbf{q}_{n,d}(1)$

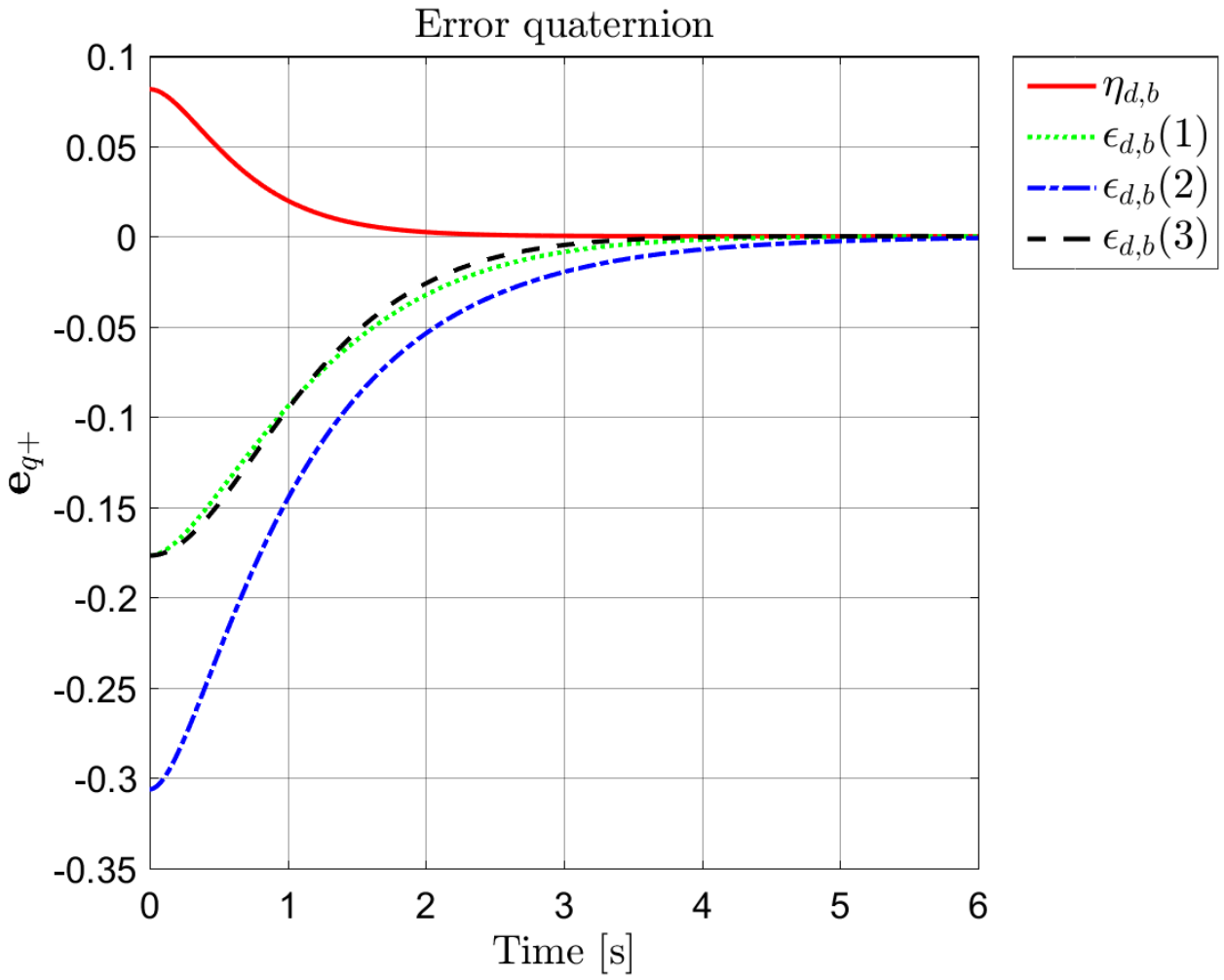


Figure 4.2: Quaternion Error for Simulation $\mathbf{q}_{n,d}(1)$

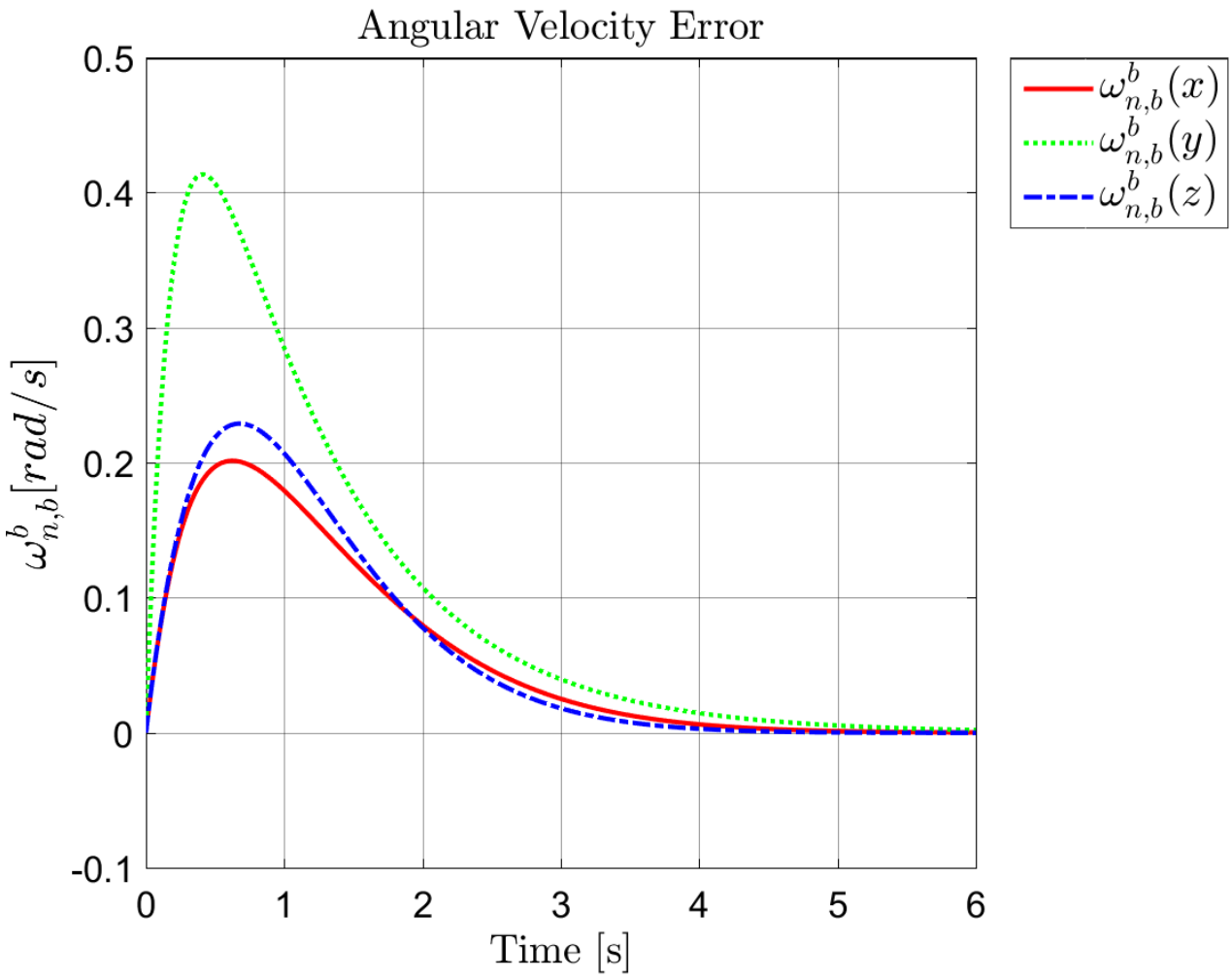
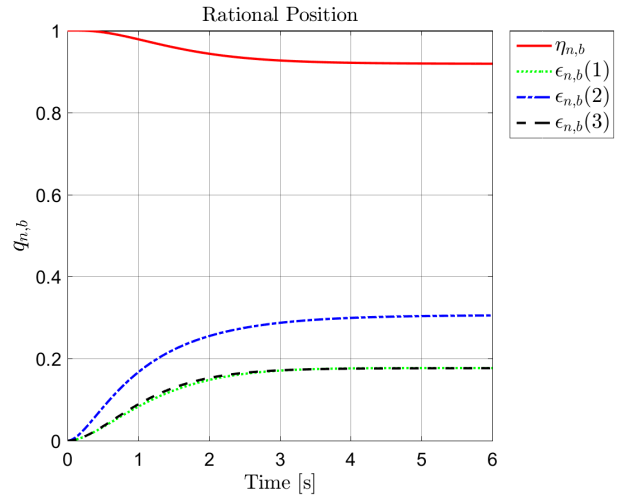
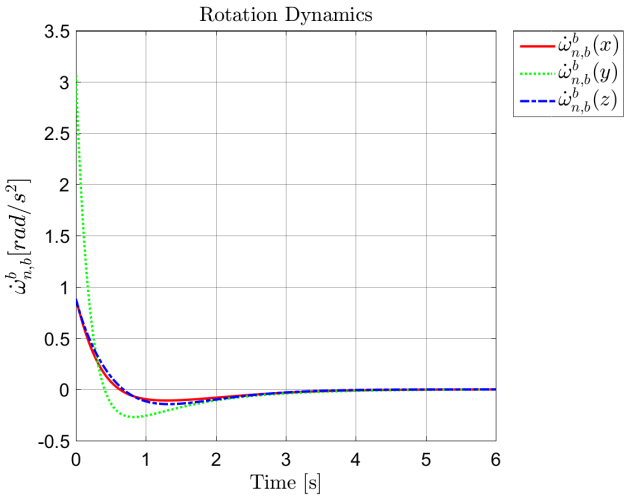


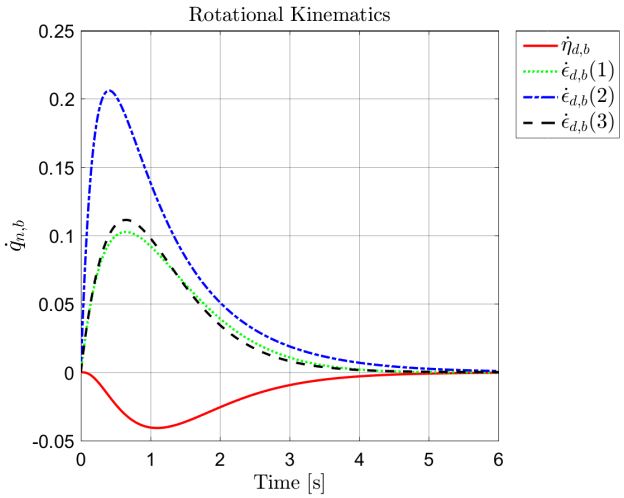
Figure 4.3: Angular Velocity Error for Simulation $\mathbf{q}_{n,d}(1)$



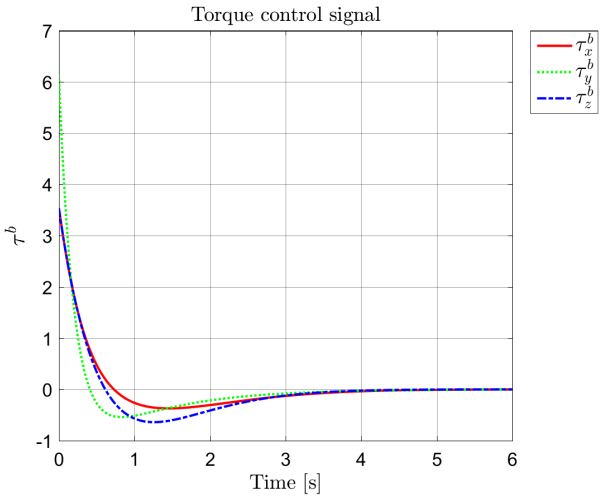
(a) Quaternion Rotation



(b) Angular Acceleration



(c) Quaternion Kinematics



(d) Quadrotor Torques

Figure 4.4: Additional Simulation Results for $\mathbf{q}_{n,d}(1)$

4.1.2 Simulation Case 2

This section contains the simulation results for the desired quaternion rotation $\mathbf{q}_{n,d}(2)$

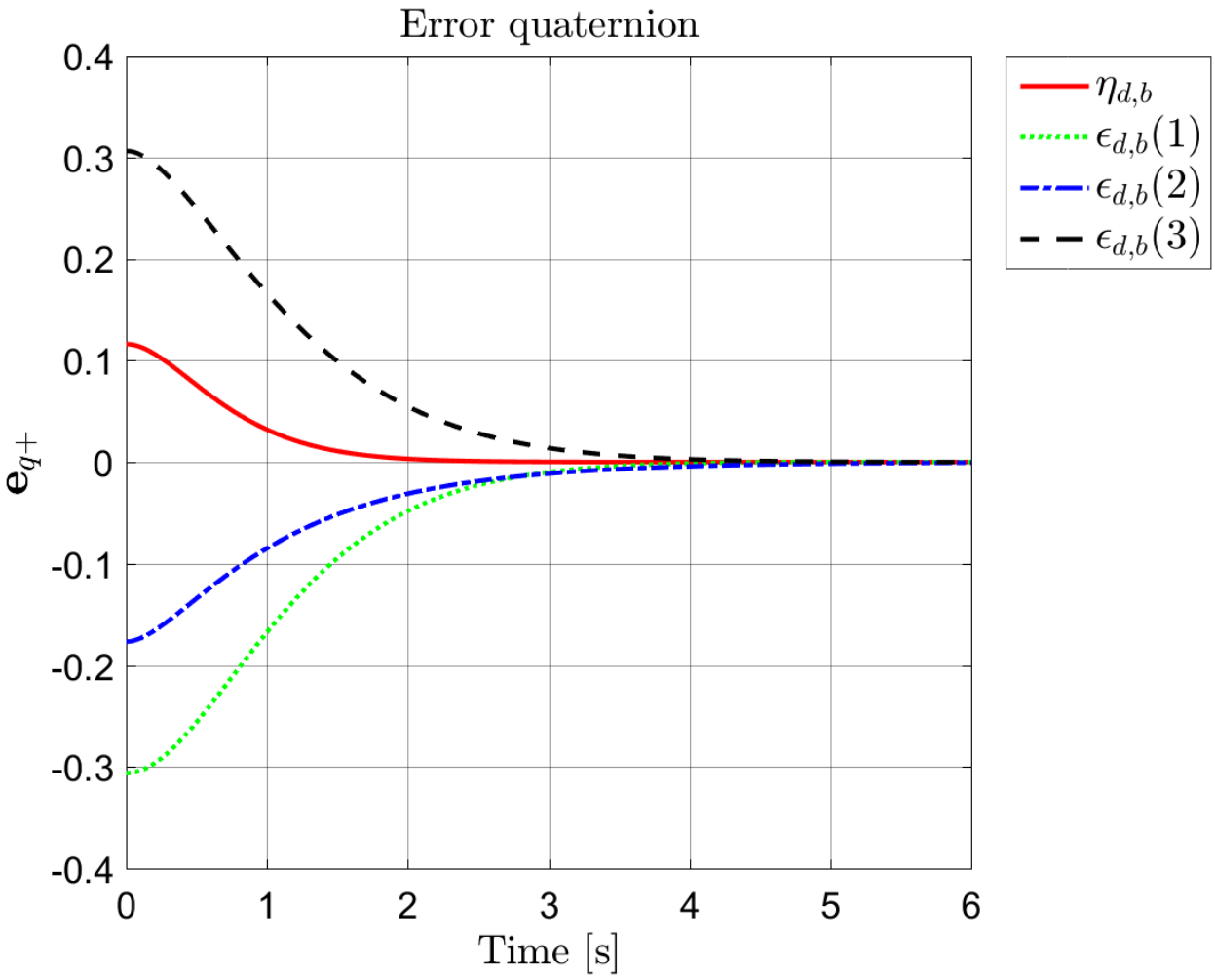


Figure 4.5: Quaternion Error for Simulation $\mathbf{q}_{n,d}(2)$

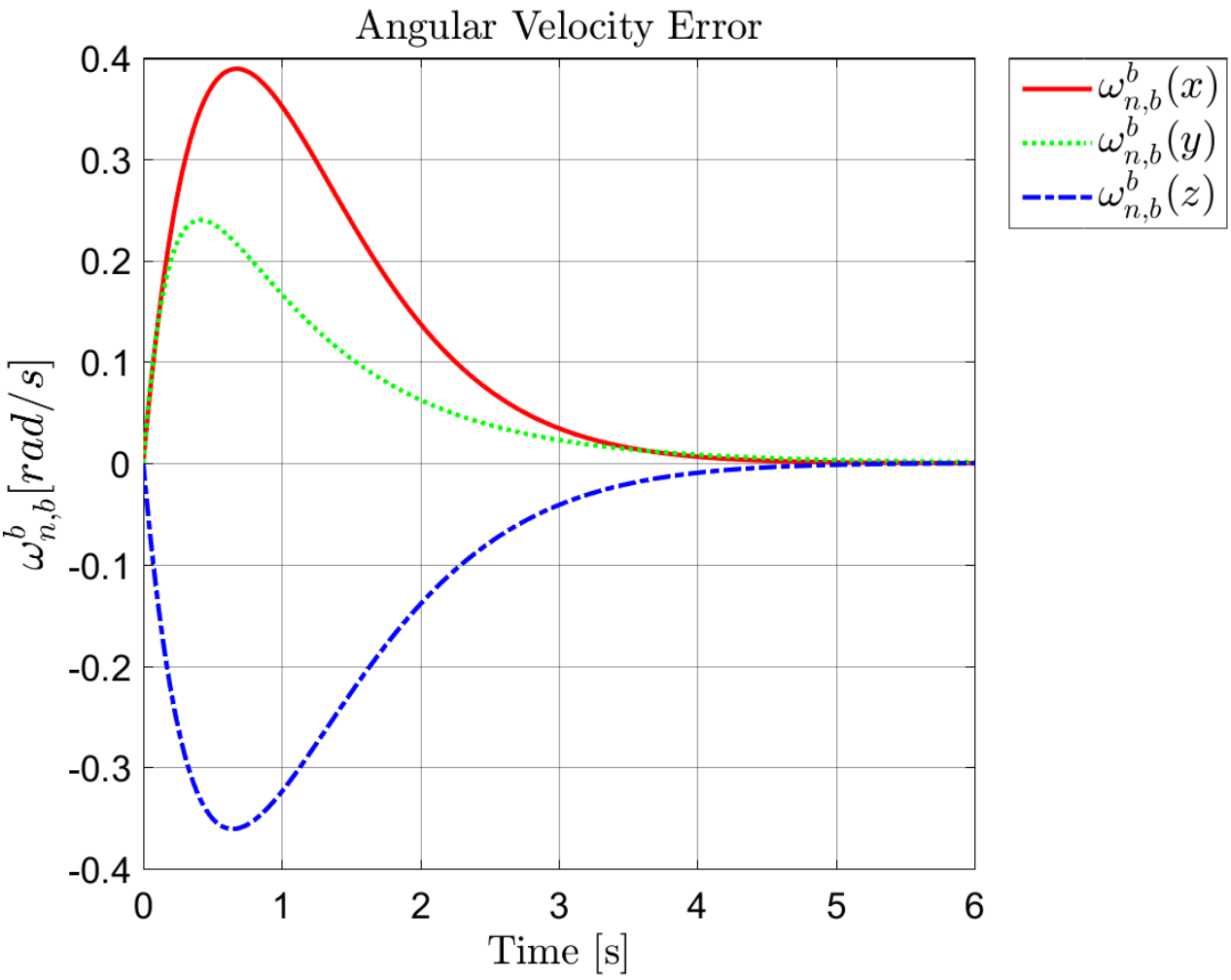
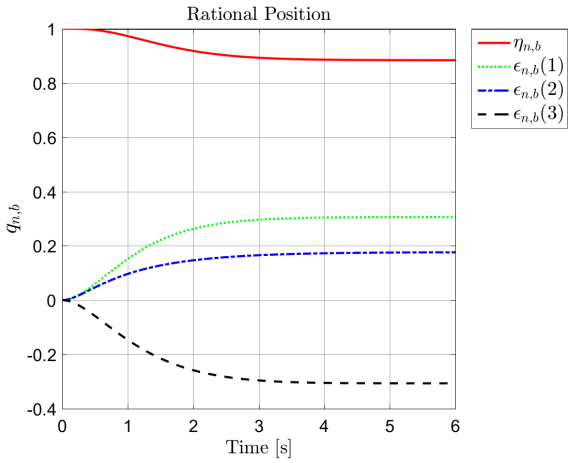
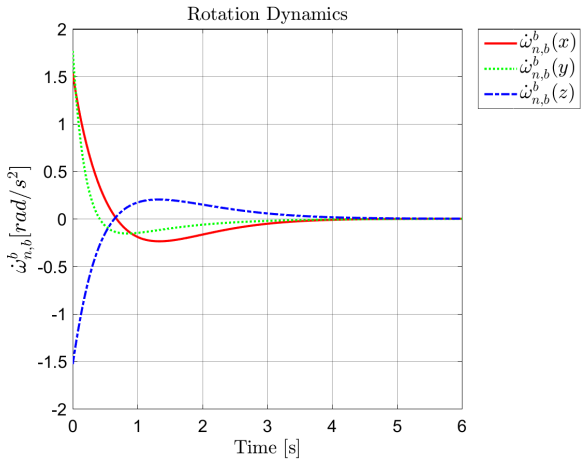


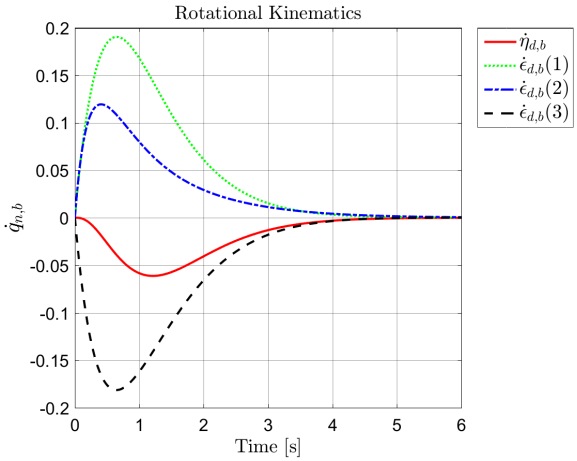
Figure 4.6: Angular Velocity Error for Simulation $\mathbf{q}_{n,d}(2)$



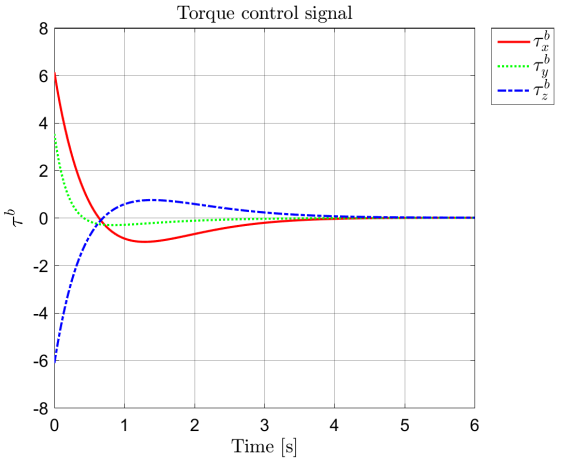
(a) Quaternion Rotation



(b) Angular Acceleration



(c) Quaternion Kinematics



(d) Quadrotor Torques

Figure 4.7: Additional Simulation Results for $\mathbf{q}_{n,d}(2)$

4.1.3 Simulation Case 3

This section contains the simulation results for the desired quaternion rotation $\mathbf{q}_{n,d}(3)$

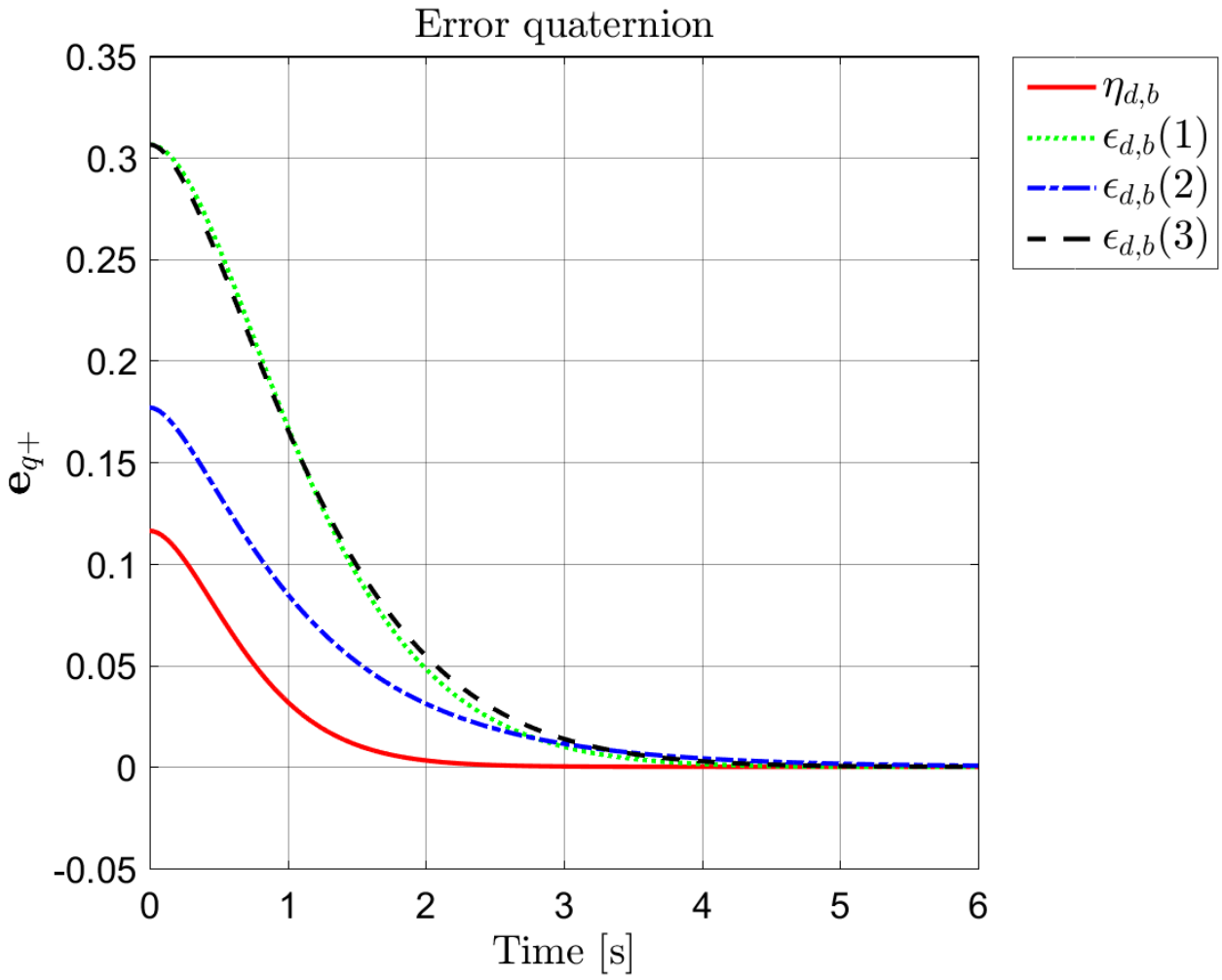


Figure 4.8: Quaternion Error for Simulation $\mathbf{q}_{n,d}(3)$

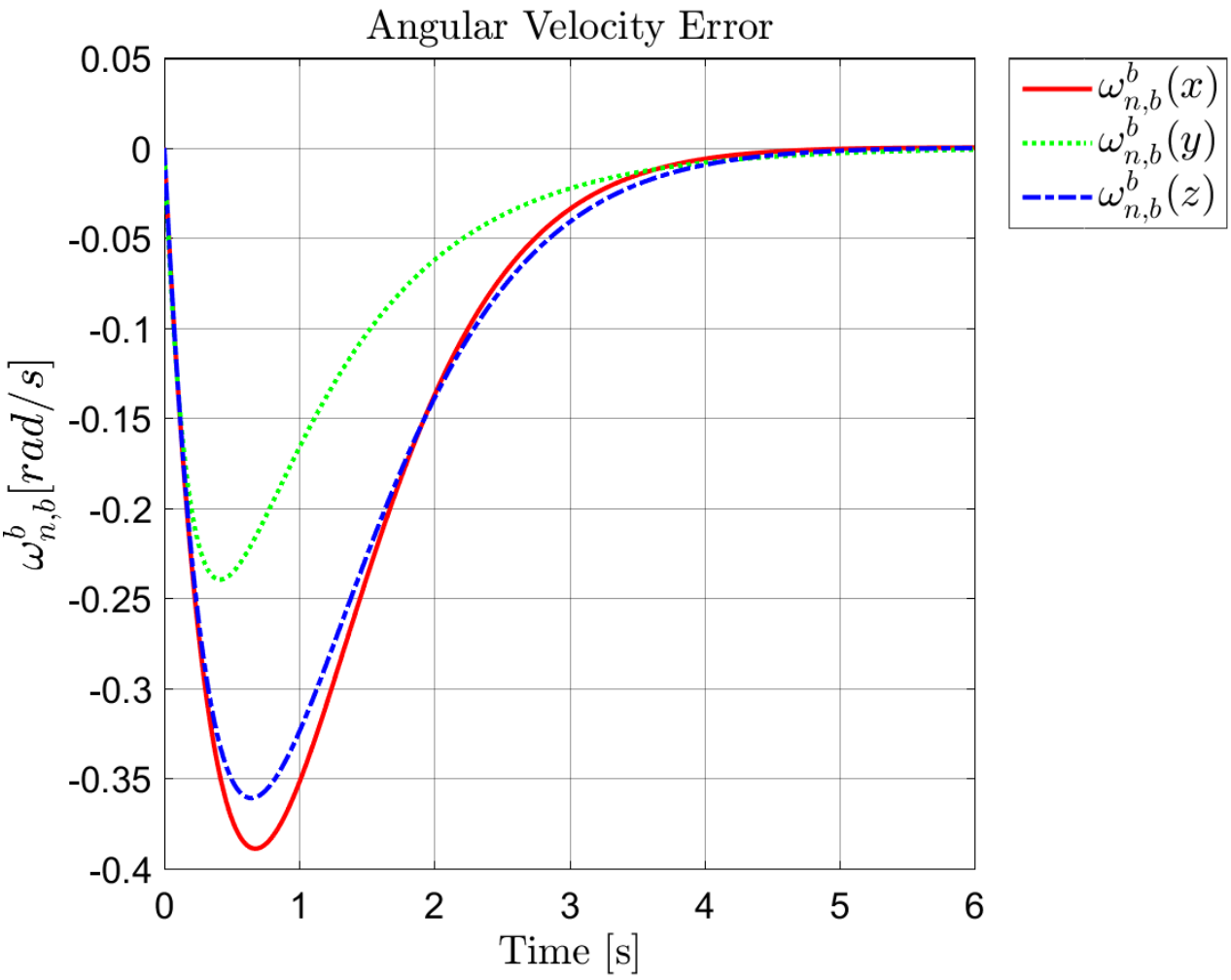
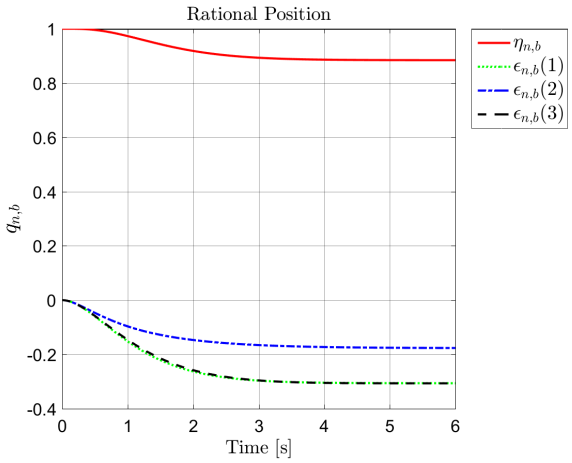
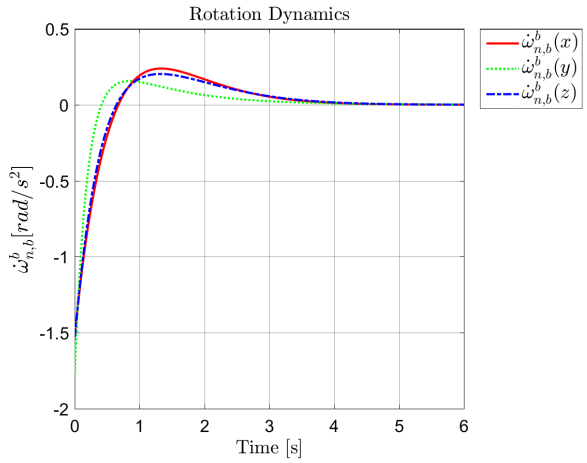


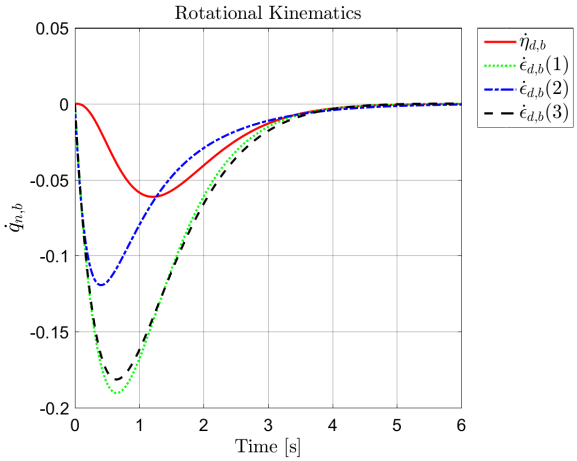
Figure 4.9: Angular Velocity Error for Simulation $\mathbf{q}_{n,d}(3)$



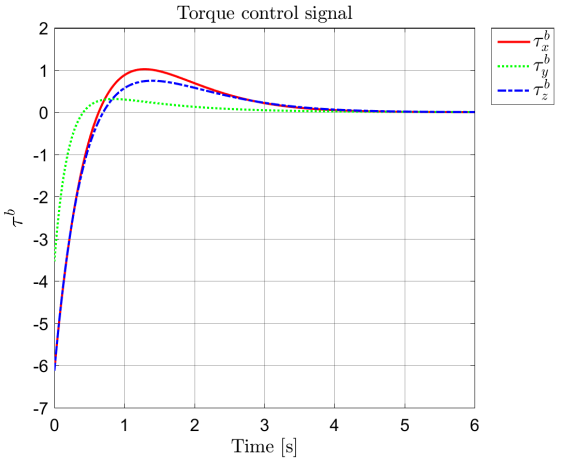
(a) Quaternion Rotation



(b) Angular Acceleration



(c) Quaternion Kinematics



(d) Quadrotor Torques

Figure 4.10: Additional Simulation Results for $\mathbf{q}_{n,d}(3)$

4.1.4 Simulation Comments

As the Simulation figures for all three cases clearly shows, the control solution for the decoupled rotation of the quadrotor presented in section 3.1 makes all errors $\rightarrow 0$, thus verifying the control solution.

4.2 Translational Cascade Simulation Results

Let the initial states be given as $\mathbf{p}^n(0) = \mathbf{0}$ m and $\mathbf{v}^n(0) = \mathbf{0}$ m/s, indicating that the start position of the quadrotor is in origin and that it is *standing still*. Three simulations are performed to test both the decoupled thrust controller and the ideal thrust vector controller. The first simulation features the thrust controller, while the second and third utilizes the thrust vector controller. Let a fixed desired position be given by the coordinates $\mathbf{p}_d^n = \begin{bmatrix} 20 & 15 & 50 \end{bmatrix}$ m, and a desired position as a function of time presented in (3.10) as

$$\mathbf{p}_d^n(t) = \begin{bmatrix} r\sin(\omega t) \\ r\cos(\omega t) \\ z_d \end{bmatrix}.$$

The following scenarios are to be simulated:

1. Thrust controller with desired position as a function of time, with $r = 25$ m, $\omega = 0.01$ rad/s and $z_d = 50$ m.
2. Thrust vector controller using the fixed desired position.
3. Thrust vector controller with trajectory tracking of a circle, with $r = 25$ m, $\omega = 0.01$ rad/s and $z_d = 50$ m, a helix, with $r = 25$ m, $\omega = 0.01$ rad/s and $z_d t$ m and a spiral, with $r(t) = 2t$ m, $\omega = 0.01$ rad/s and $z_d t = 1t$ m.

The mass of the quadrotor is 1.22463 kg and the control gains are set to $k_p = 20$ and $k_d = 15$.

4.2.1 Simulation Case 1

This section contains the simulation results using the thrust controller to track a desired trajectory.

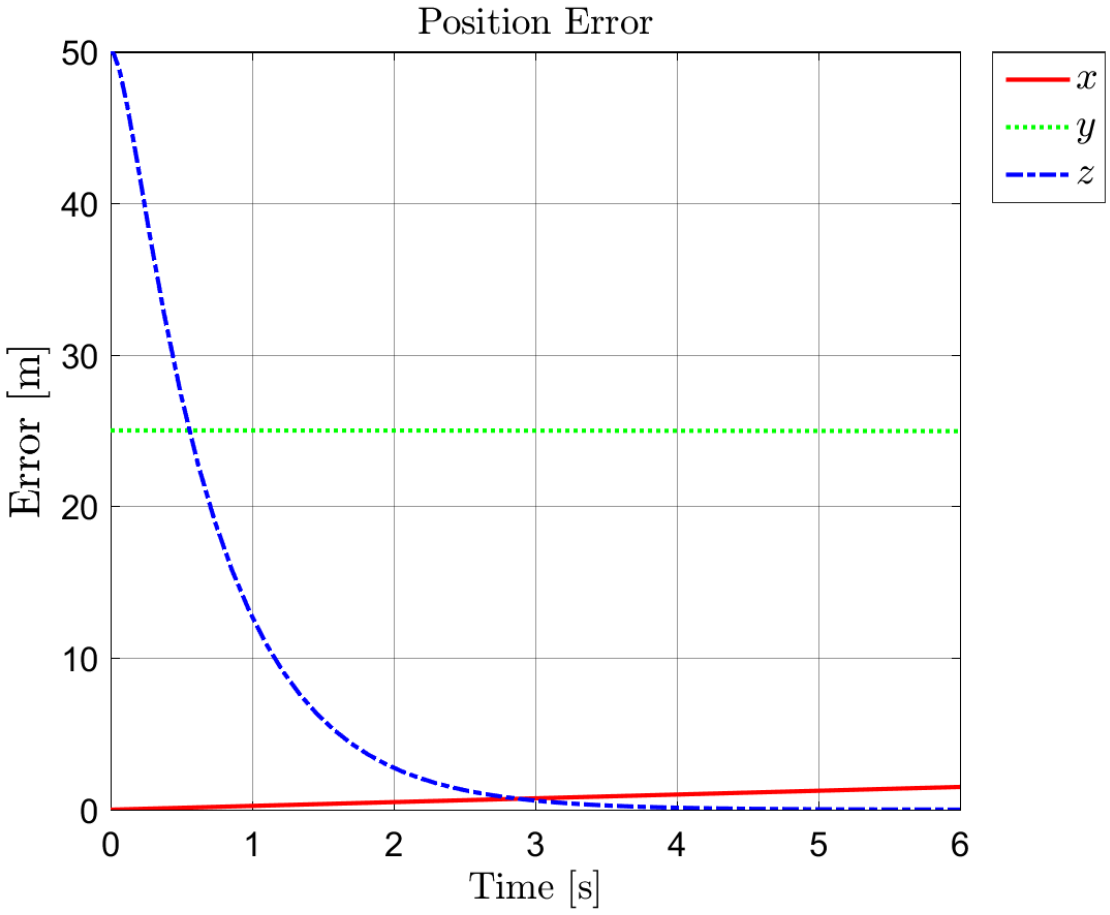
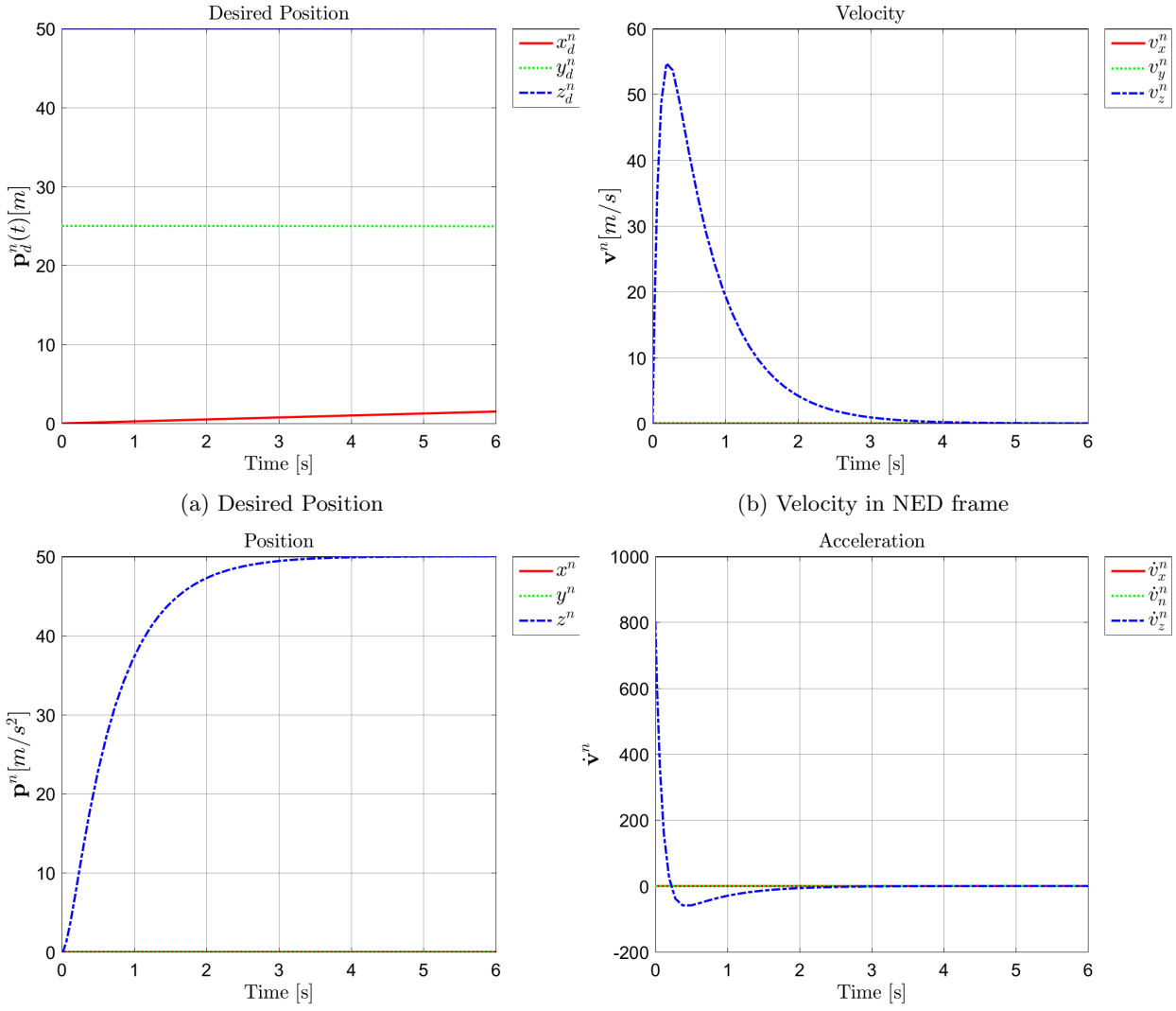


Figure 4.11: Position Error with Thrust Controller Υ^b



(a) Desired Position

(b) Velocity in NED frame

(c) Position in NED frame

(d) Acceleration in NED frame

Figure 4.12: Additional Simulation Results with Thrust Controller Υ^b

4.2.2 Simulation Case 2

This section contains the simulation results using the thrust vector to reach a fixed position.

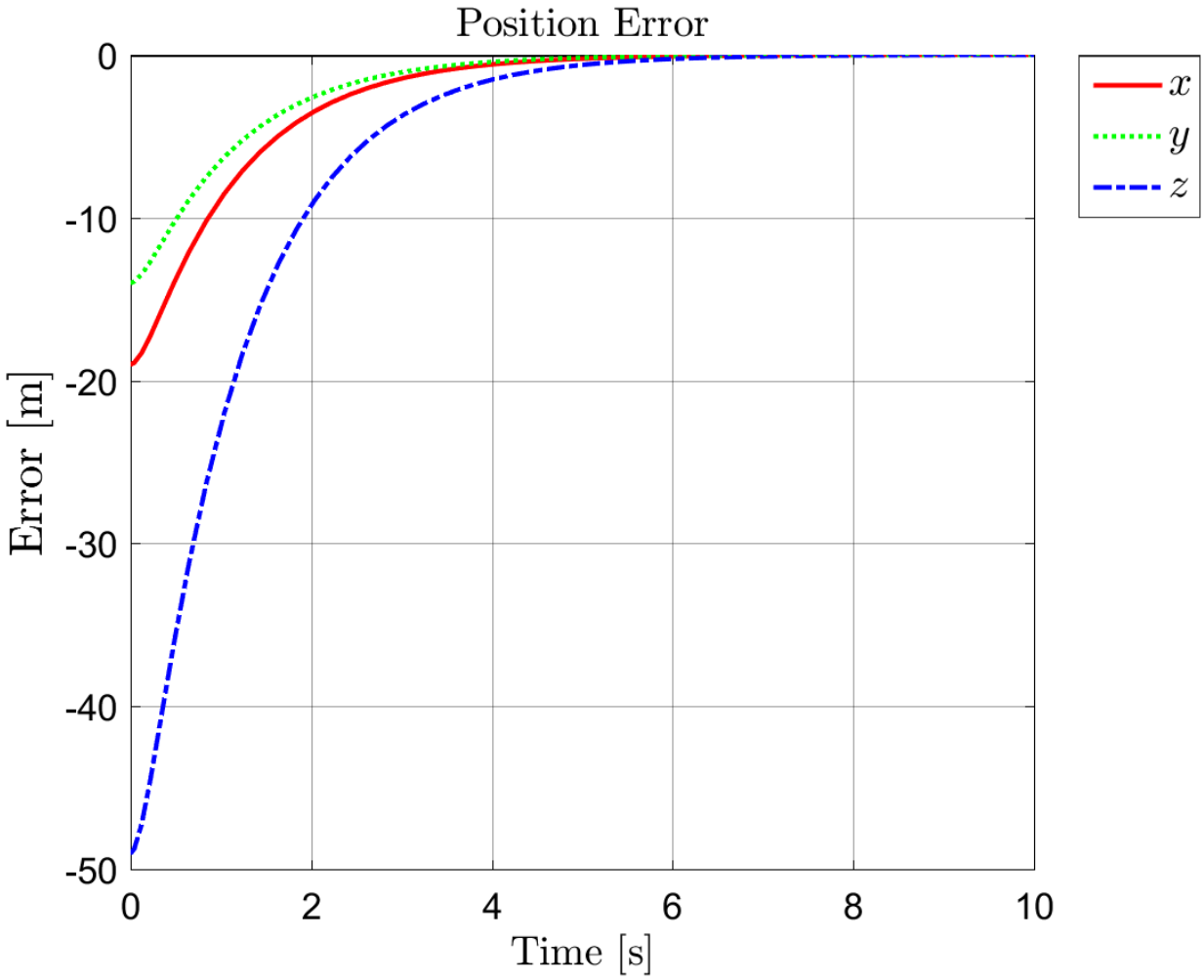
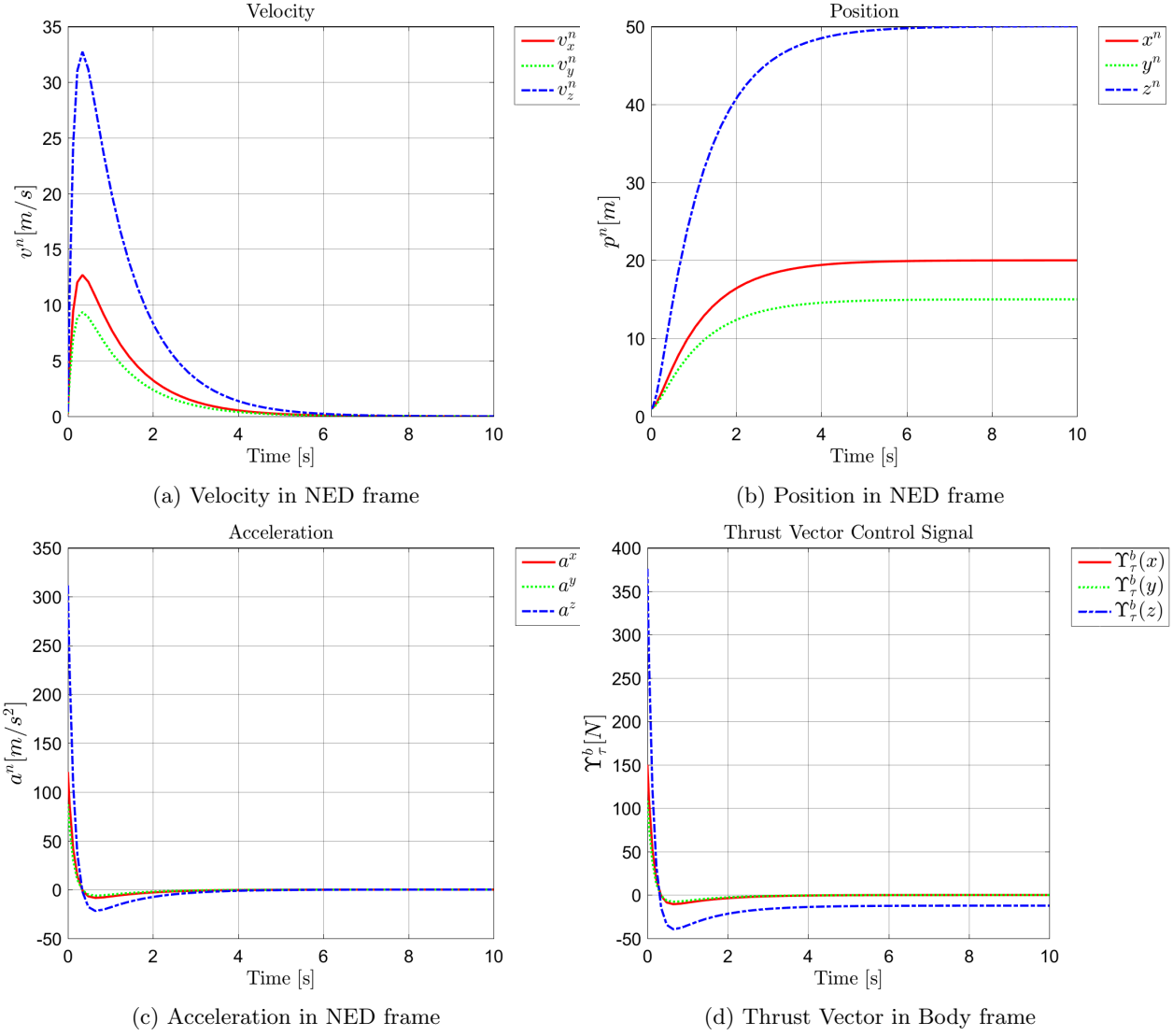


Figure 4.13: Position Error in Fixed coordinates Simulation with Thrust Vector controller $\boldsymbol{\Upsilon}^b$



(a) Velocity in NED frame

(b) Position in NED frame

(c) Acceleration in NED frame

(d) Thrust Vector in Body frame

Figure 4.14: Additional Simulation Results for Fixed Coordinates with Thrust Vector Controller $\boldsymbol{\Upsilon}^b$

4.2.3 Simulation Case 3

This section contains the simulation results using the thrust vector to track a desired trajectory, accompanied by $3D$ figures showing the trajectory tracking of a circle, helix and a spiral.

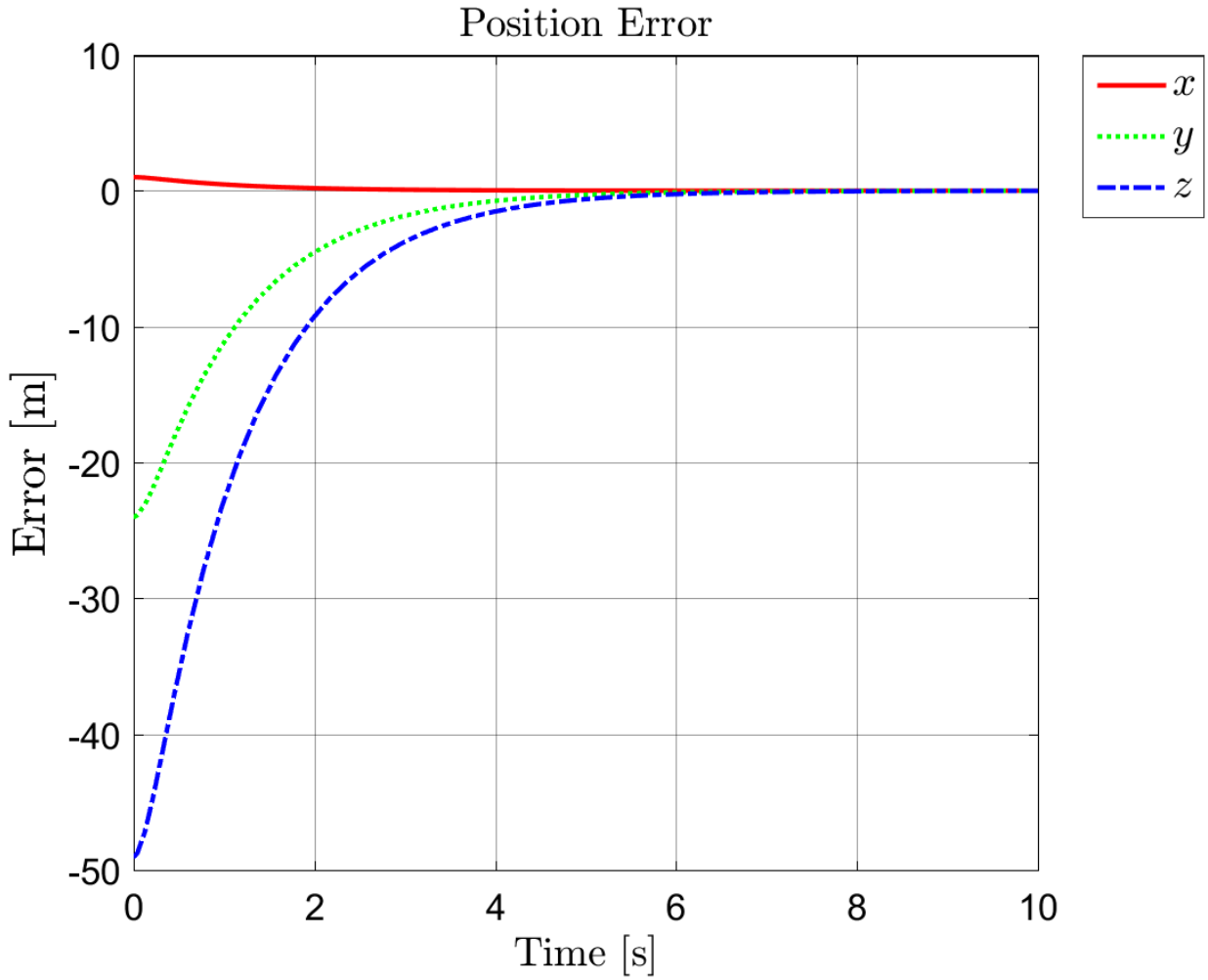
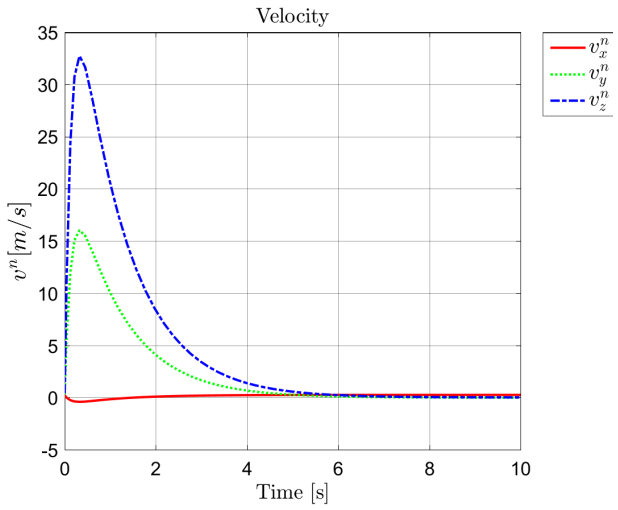
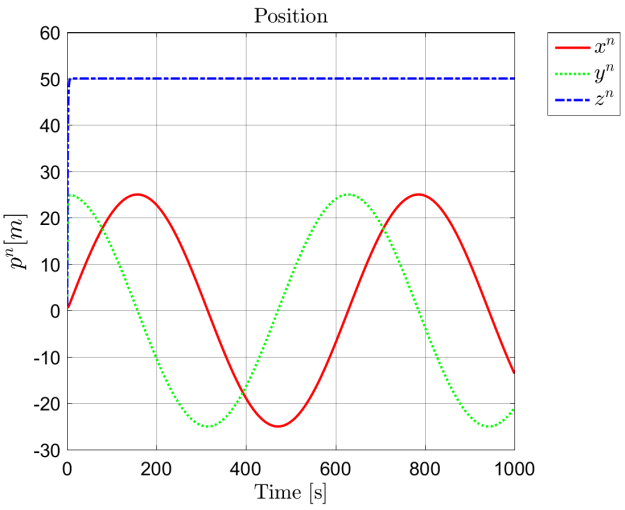


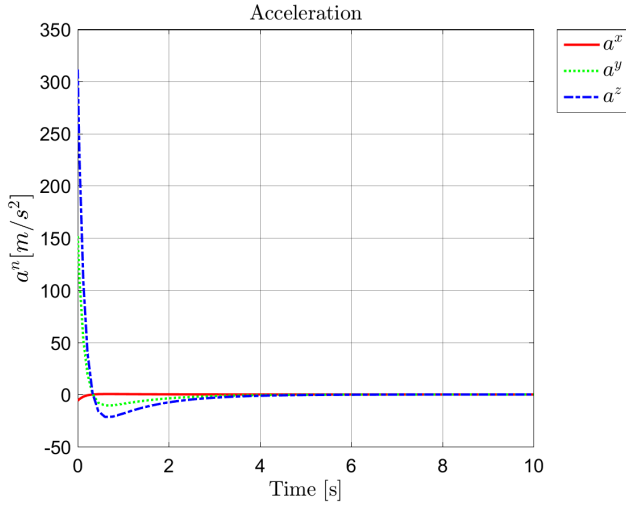
Figure 4.15: Position Error



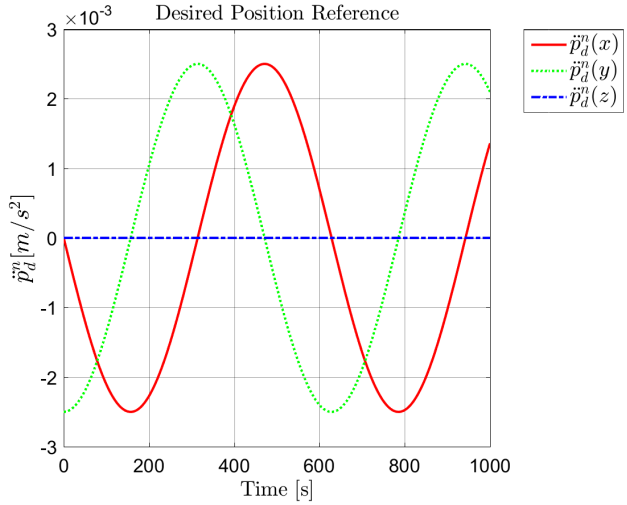
(a) Velocity in NED frame



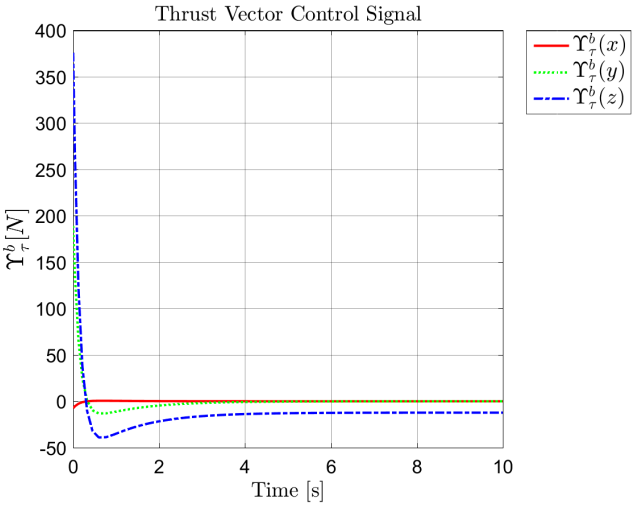
(b) Position in NED frame



(c) Acceleration in NED frame

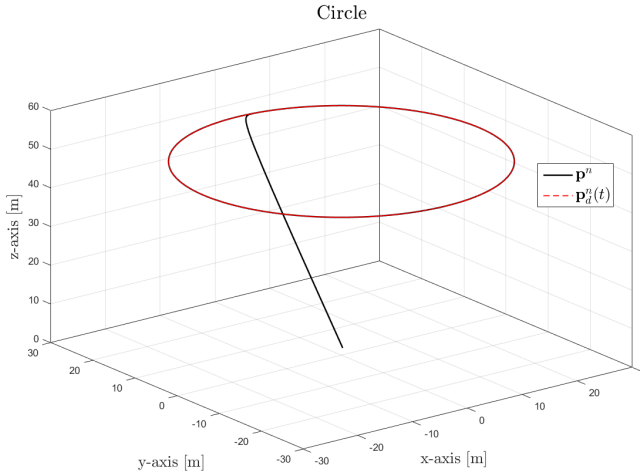


(d) Desired Position Reference

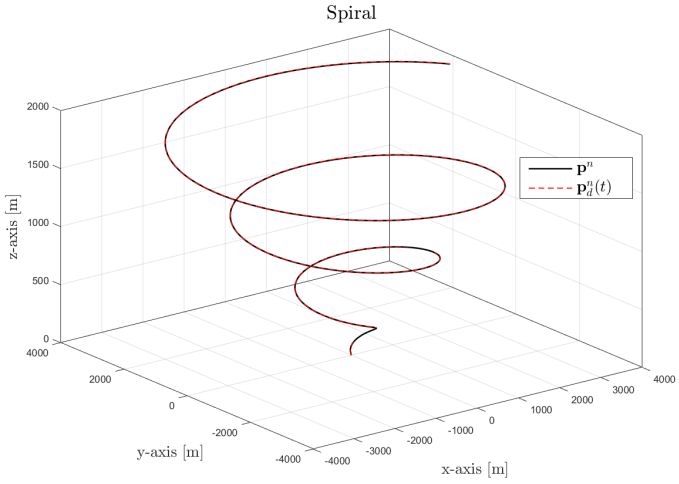


(e) Thrust Vector in Body frame

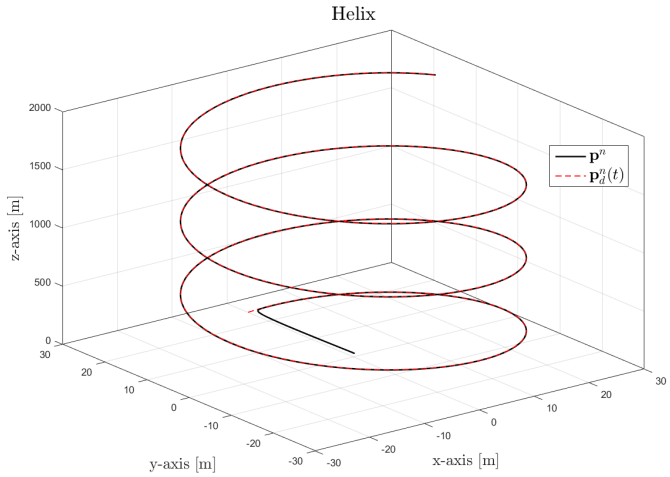
Figure 4.16: Additional Simulation Results for Trajectory Tracking with Thrust vector Controller $\boldsymbol{\Upsilon}^b$



(a) Trajectory tracking of a Circle with Altitude 50 m



(b) Trajectory Tracking of a Spiral



(c) Trajectory Tracking of a Helix

Figure 4.17: 3D-plots of Trajectory Tracking with Thrust Vector Controller $\boldsymbol{\Upsilon}^b$

4.2.4 Simulation Comments

As the simulation figures shows, the translation system behaves as expected according to the assumptions made in section 3.2, both when it comes to the thrust controller and the perfect rotational response from the thrust control vector. As the figures of the circle, helix and spiral trajectory tracking shows, the quadrotor follows the desired positions accurately.

4.3 Total Cascade Interconnection Simulations

Let the initial states be given as $\mathbf{q}_{n,b} = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}^T$, $\mathbf{p}^n(0) = \mathbf{0}\ m$ and $\mathbf{v}^n(0) = \mathbf{0}\ m/s$, indicating that the quadrotor is *standing still* and has the starting position in origin with no initial rotation. The thrust of the quadrotor has been limited to 30 N following the quadrotor specification given, and the rotation angles have been limited to $\vartheta = \pm 30°$. The following scenarios are to be simulated:

1. Fixed Desired position, with $\mathbf{p}_d^n = \begin{bmatrix}10 & -10 & 15\end{bmatrix}^T\ m$, to test if the rotation and translation controllers are working, and to test if the cascades are working in union. The rotation controller gains are set to $k_p = 4$ and $k_d = 1$, and the translation controller gains are set to $k_p = 2$ and $k_d = 2$.

2. Trajectory tracking of a circle, with $r = 25\ m$, $\omega = 0.1\ rad/s$ and $z_d = 50$ m, where the rotation controller gains are set to $k_p = 5$ and $k_d = 1$, and the translation controller gains are set to $k_p = 1$ and $k_d = 5$.

3. Trajectory tracking of a helix, with $r = 25\ m$, $\omega = 0.1\ rad/s$ and $z_d(t) = -1t\ m$, where the rotation controller gains are set to $k_p = 5$ and $k_d = 1$, and the translation controller gains are set to $k_p = 1$ and $k_d = 5$.

4. Trajectory tracking of a spiral, with $r(t) = t + 1\ m$, $\omega = 0.1\ rad/s$ and $z_d(t) = -1t\ m$, where the rotation controller gains are set to $k_p = 20$ and $k_d = 1$, and the translation controller gains are set to $k_p = 1$ and $k_d = 5$.

5. Waypoint tracking, where the desired positions are set to: $wp_1 = \begin{bmatrix}5 & 5 & 0\end{bmatrix}^T$, $wp_2 = \begin{bmatrix}5 & 5 & -5\end{bmatrix}^T$, $wp_3 = \begin{bmatrix}-5 & 15 & -5\end{bmatrix}^T$, $wp_1 = \begin{bmatrix}-7 & -8 & -20\end{bmatrix}^T$ and $wp_1 = \begin{bmatrix}15 & -10 & -5\end{bmatrix}^T$. the rotation controller gains are set to $k_p = 10$ and $k_d = 2$, and the translation controller gains are set to $k_p = 2$ and $k_d = 5$.

4.3.1 Simulation Case 1

This section contains the simulation results for the total cascaded system tracking a fixed desired position.

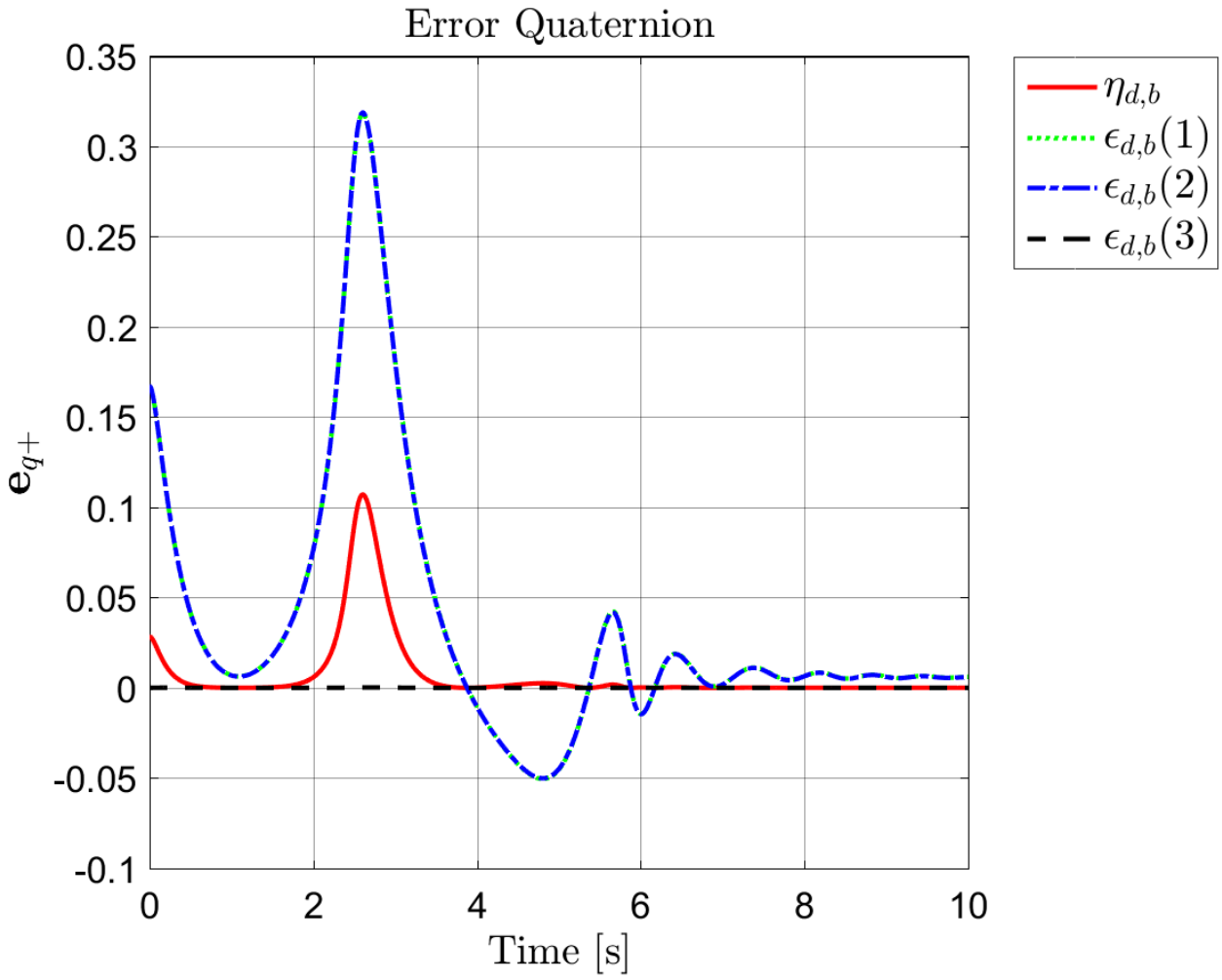


Figure 4.18: Quaternion Error, Fixed position Simulation

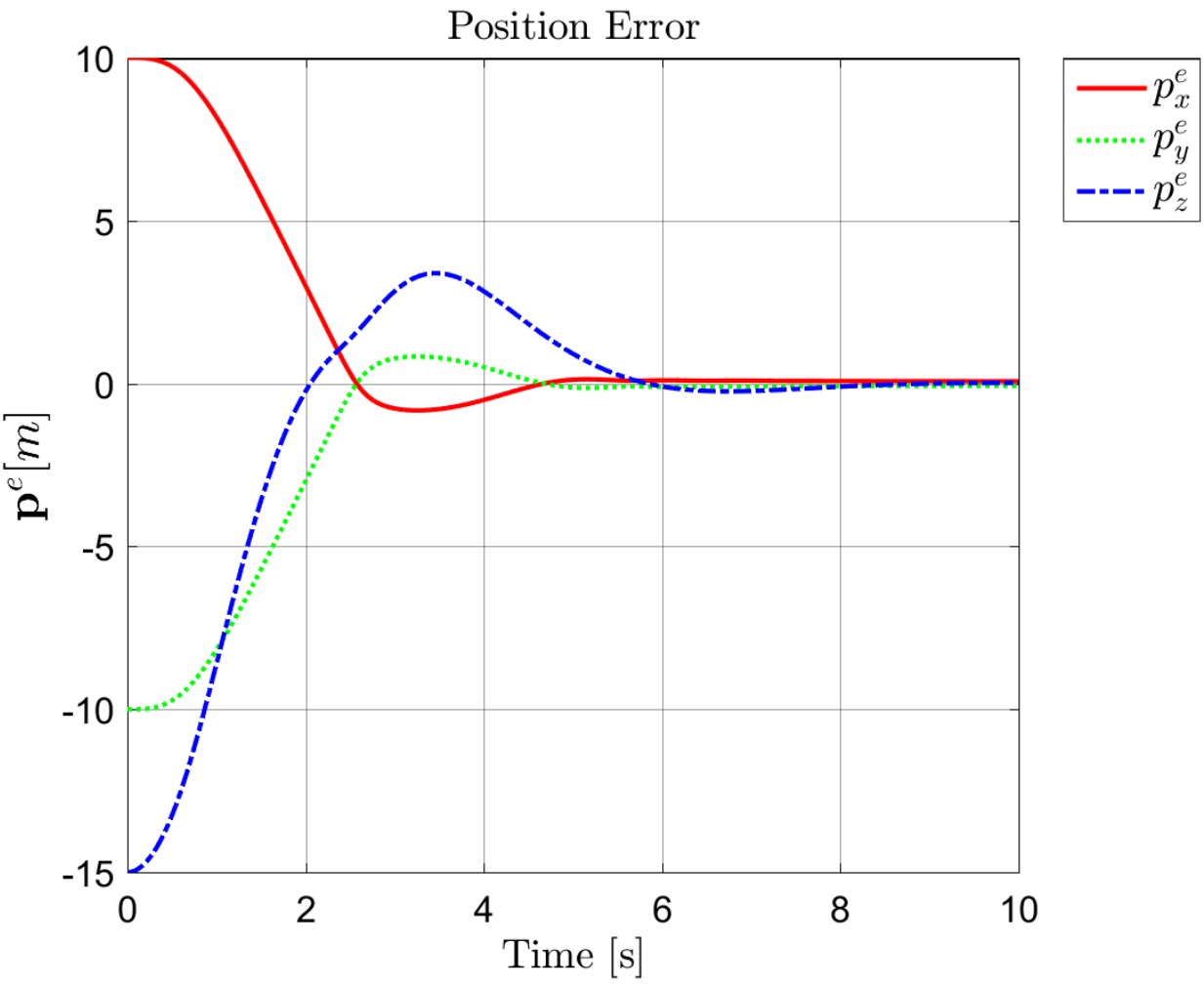


Figure 4.19: Position error, Fixed Position Simulation

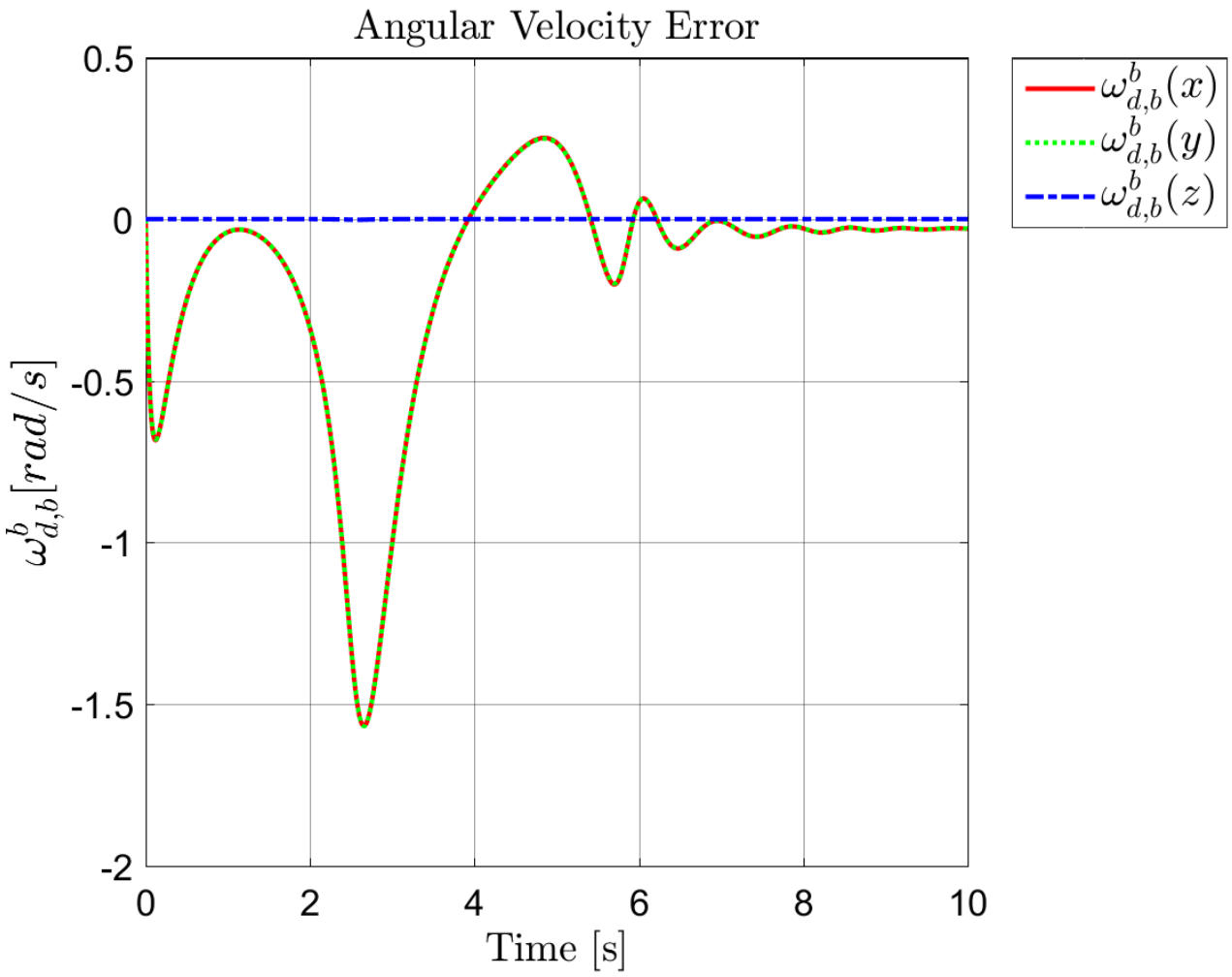
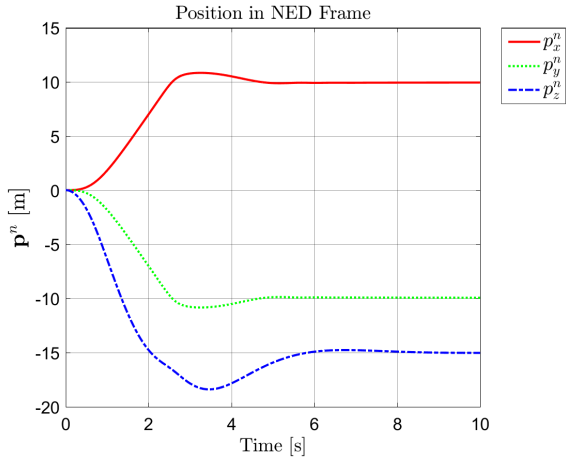
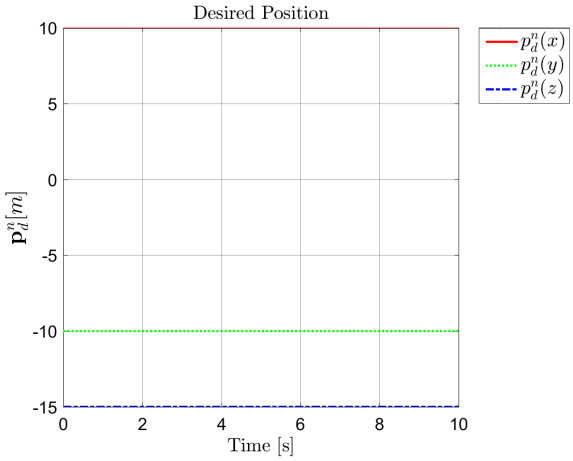


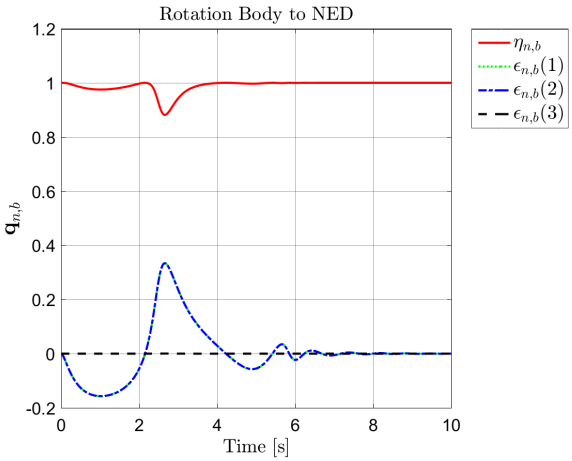
Figure 4.20: Angular Velocity error, fixed Position Simulation



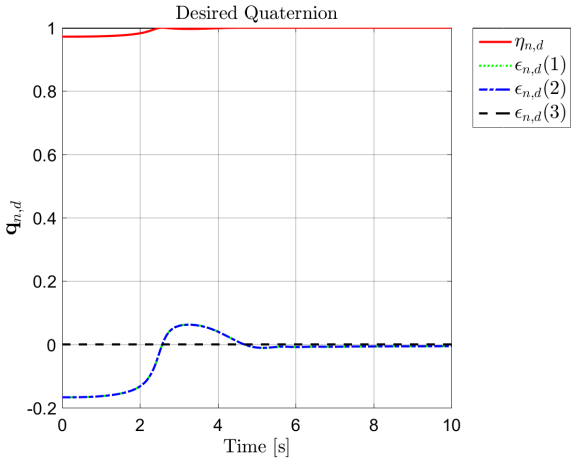
(a) Position in NED Frame



(b) Desired Position in NED Frame



(c) Quaternion Rotation from Body to NED Frame



(d) Desired Quaternion Rotation from Desired to NED Frame

Figure 4.21: Additional Simulation results, Fixed Position Simulation

4.3.2 Simulation Case 2

This section contains the simulation results for the total cascaded system tracking a circle trajectory.

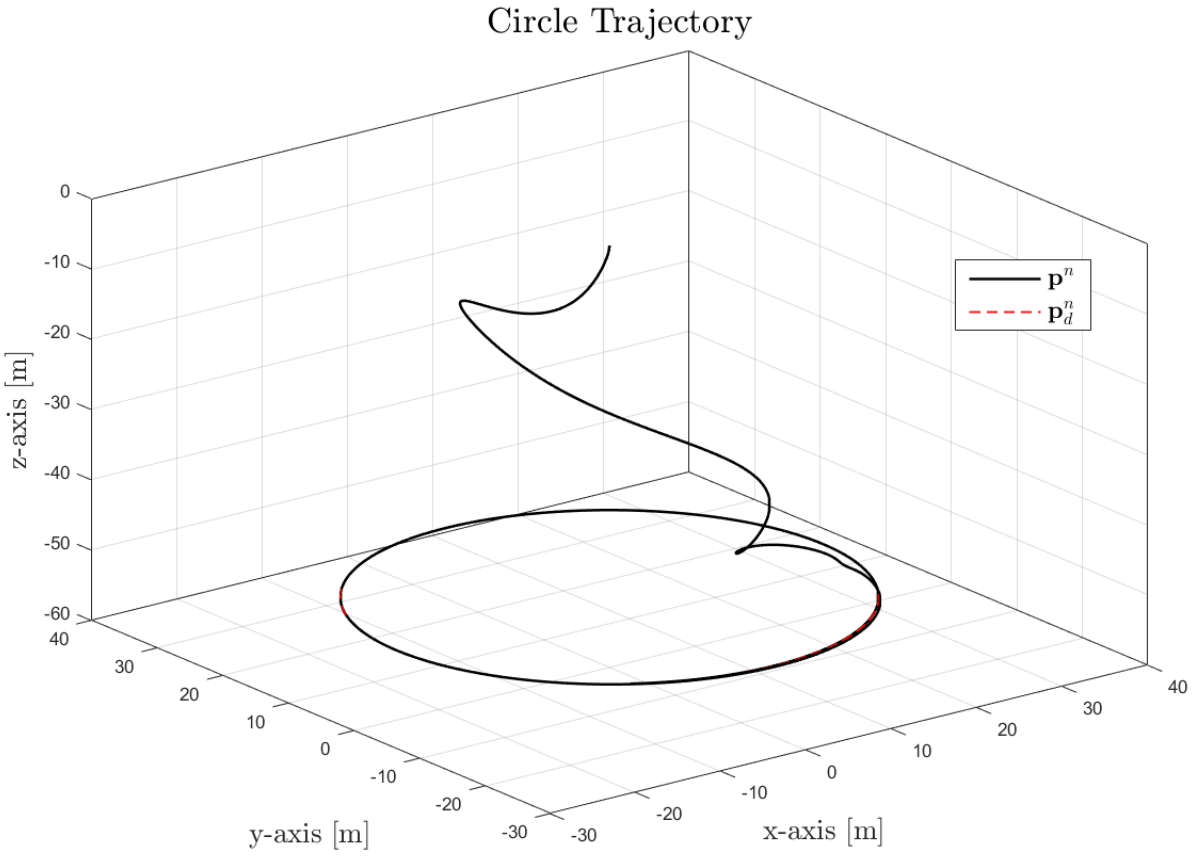


Figure 4.22: Trajectory Tracking of a Circle

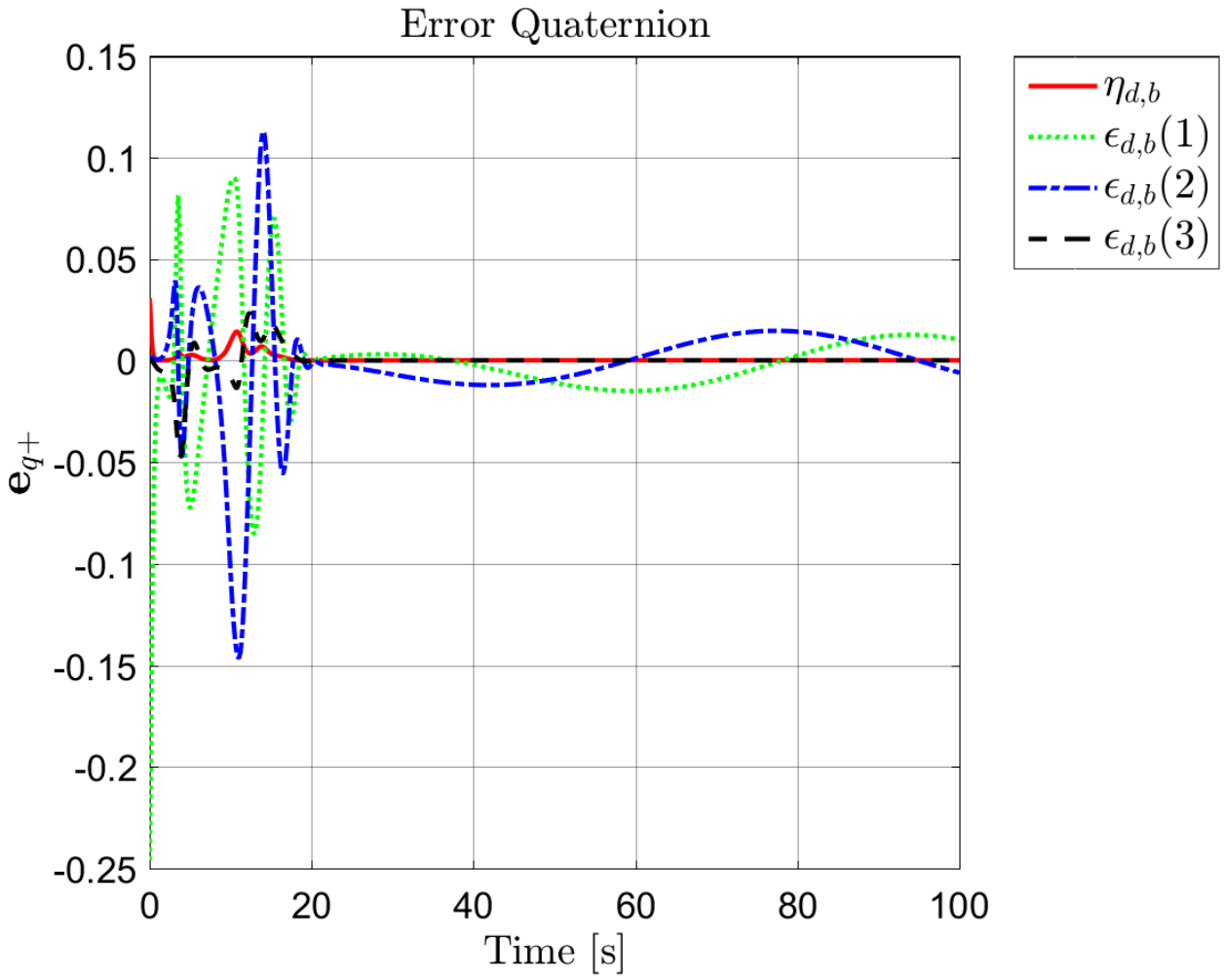


Figure 4.23: Quaternion Error, Trajectory Tracking of a Circle

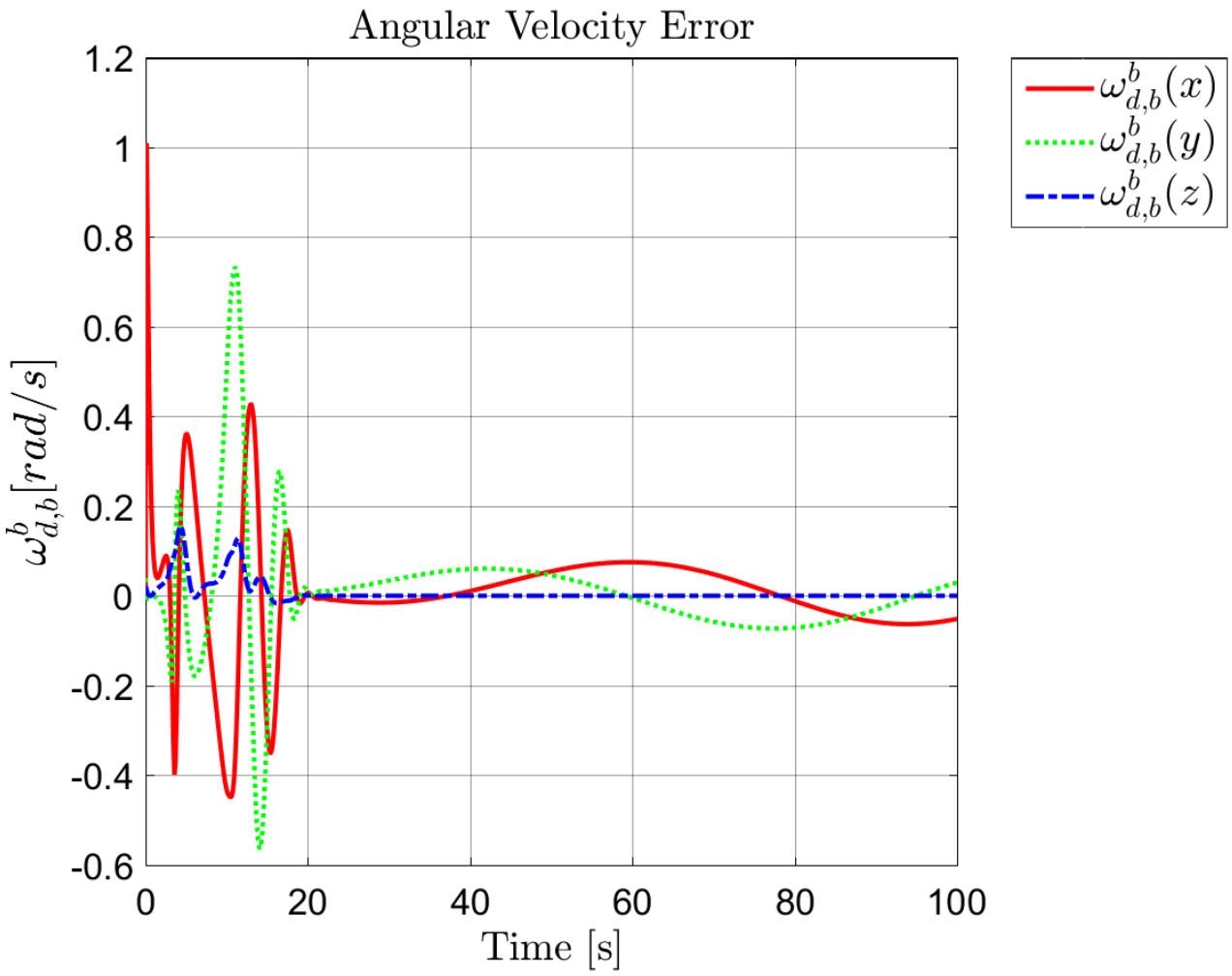


Figure 4.24: Angular Velocity Error, Trajectory Tracking of a Circle

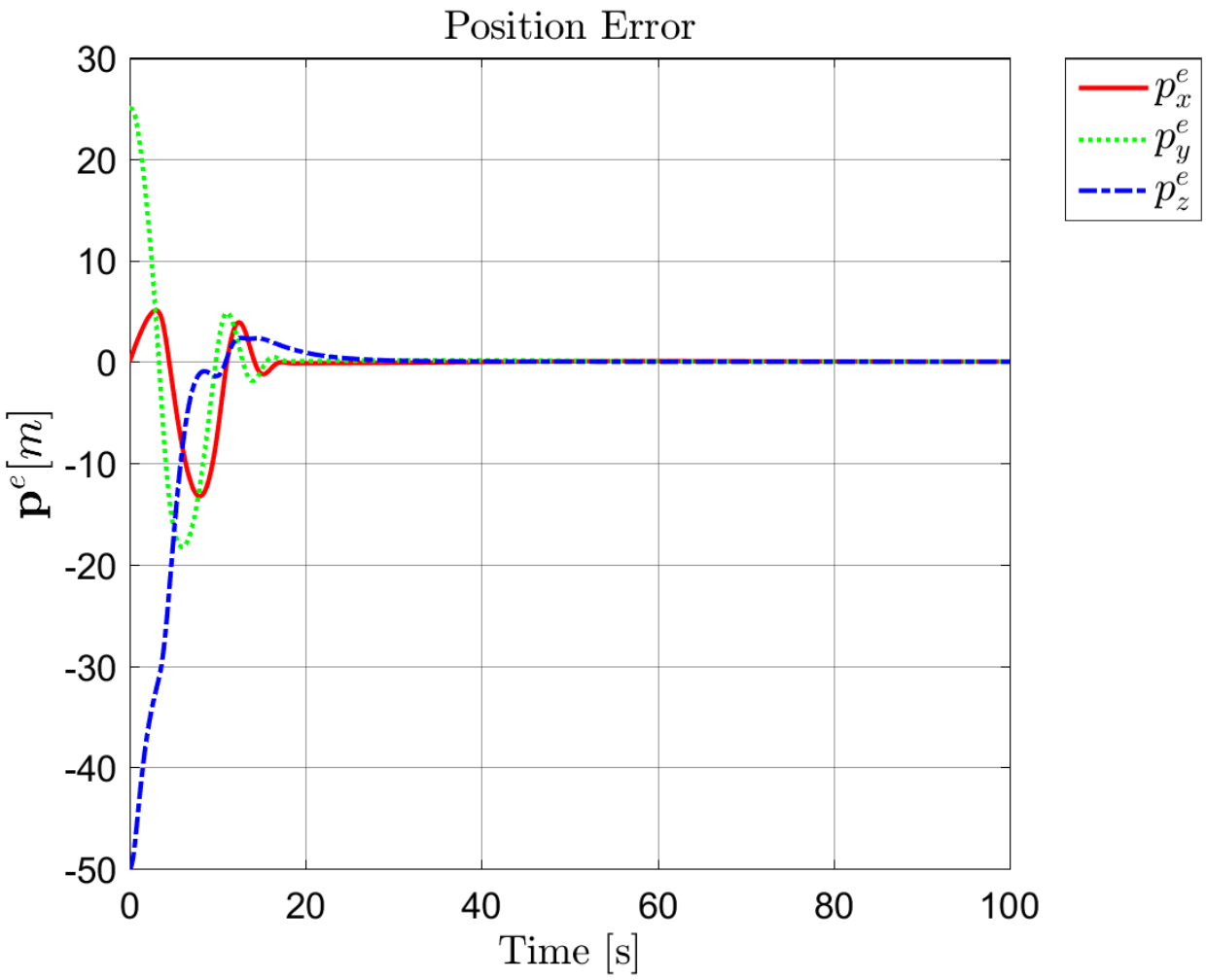
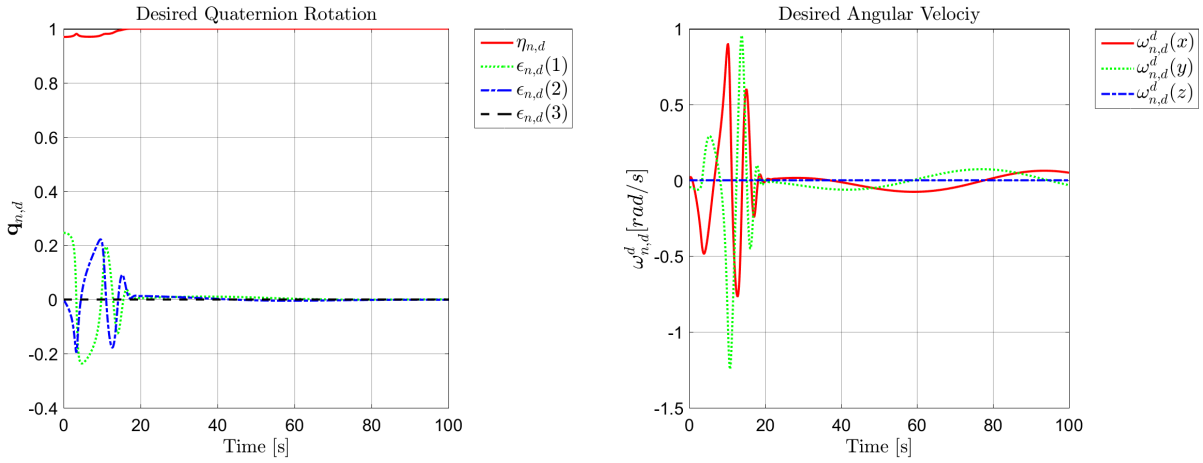


Figure 4.25: Position Error, Trajectory Tracking of a Circle



(a) Desired Quaternion Rotation

(b) Desired Angular Velocity

Figure 4.26: Additional Simulation Results, Trajectory Tracking of a Circle

4.3.3 Simulation Case 3

This section contains the simulation results for the total cascaded system tracking a helix trajectory.

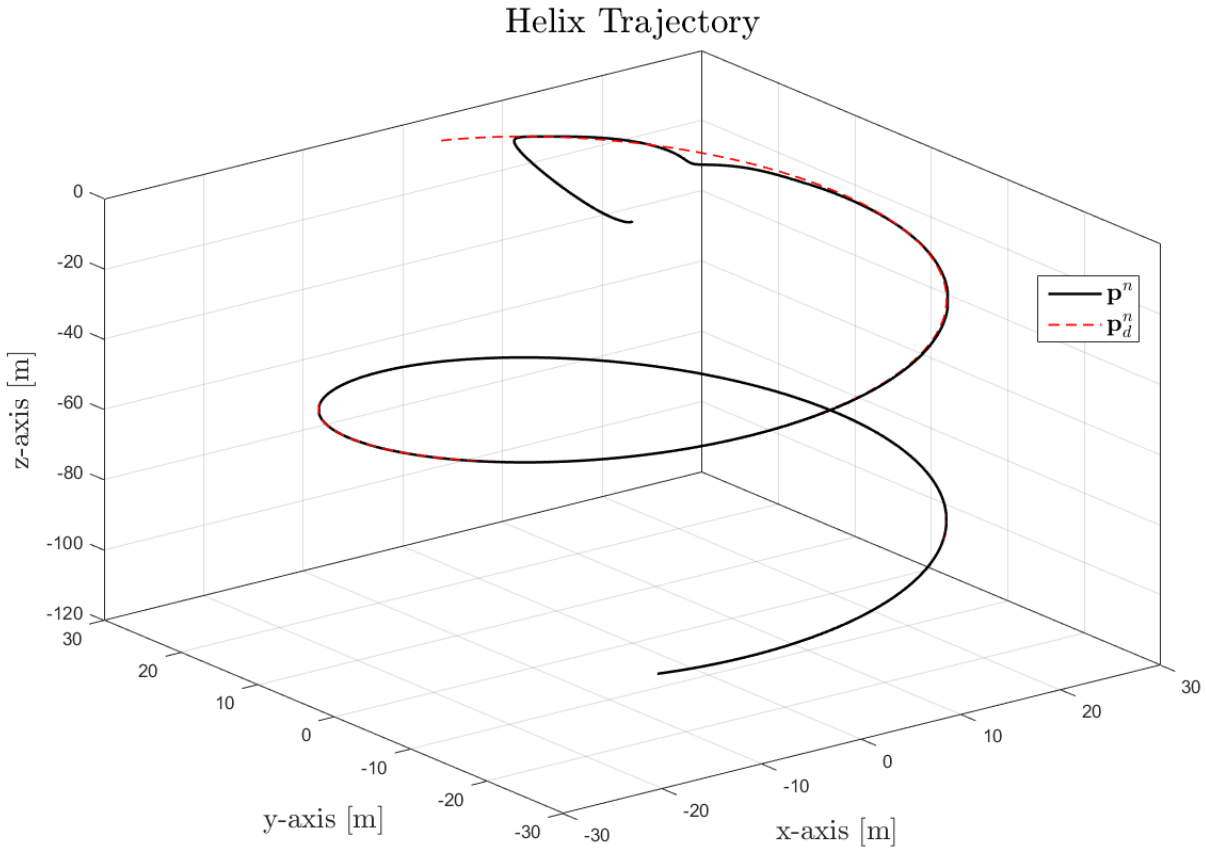


Figure 4.27: Trajectory Tracking of a Helix

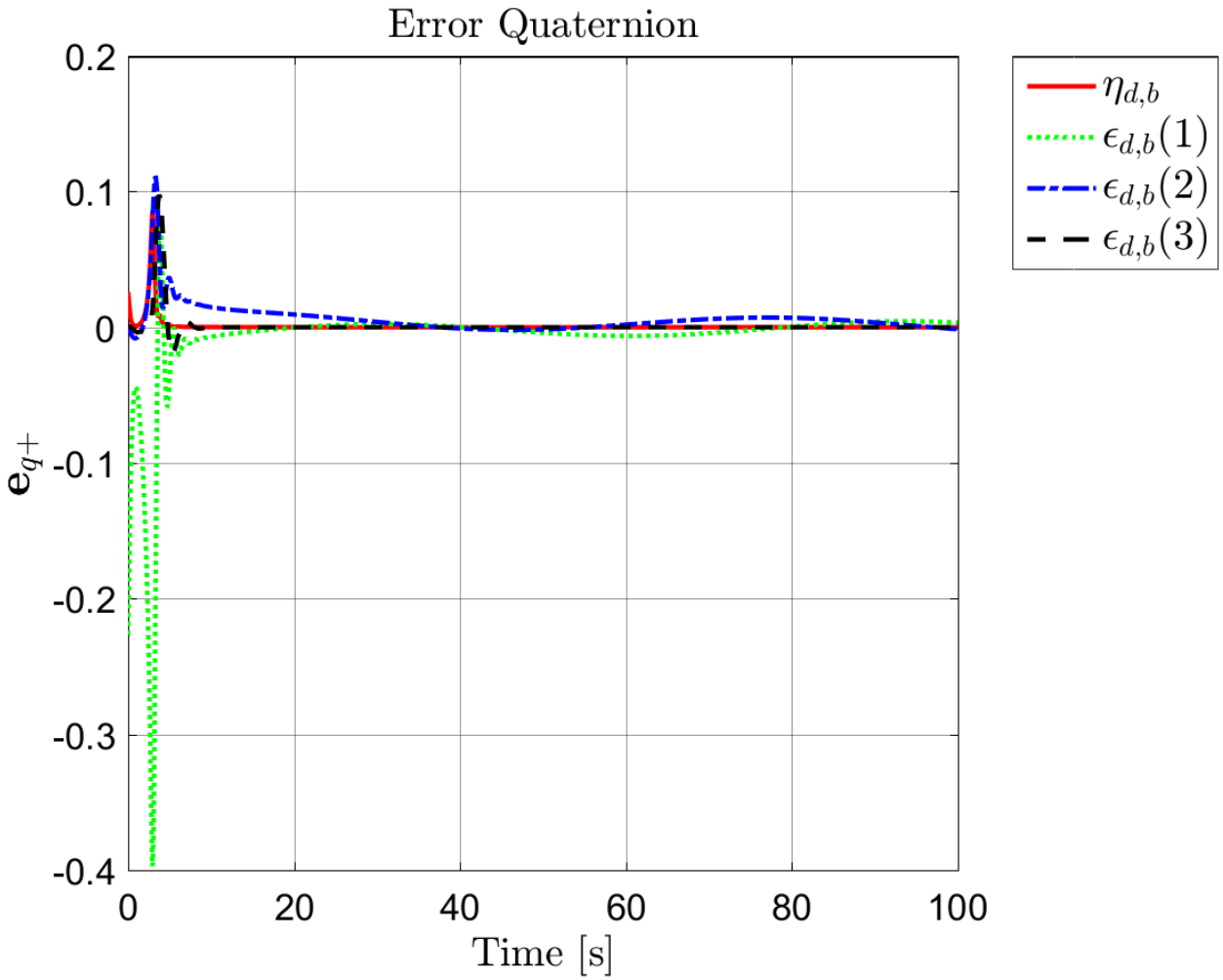


Figure 4.28: Quaternion Error, Trajectory Tracking of a Helix

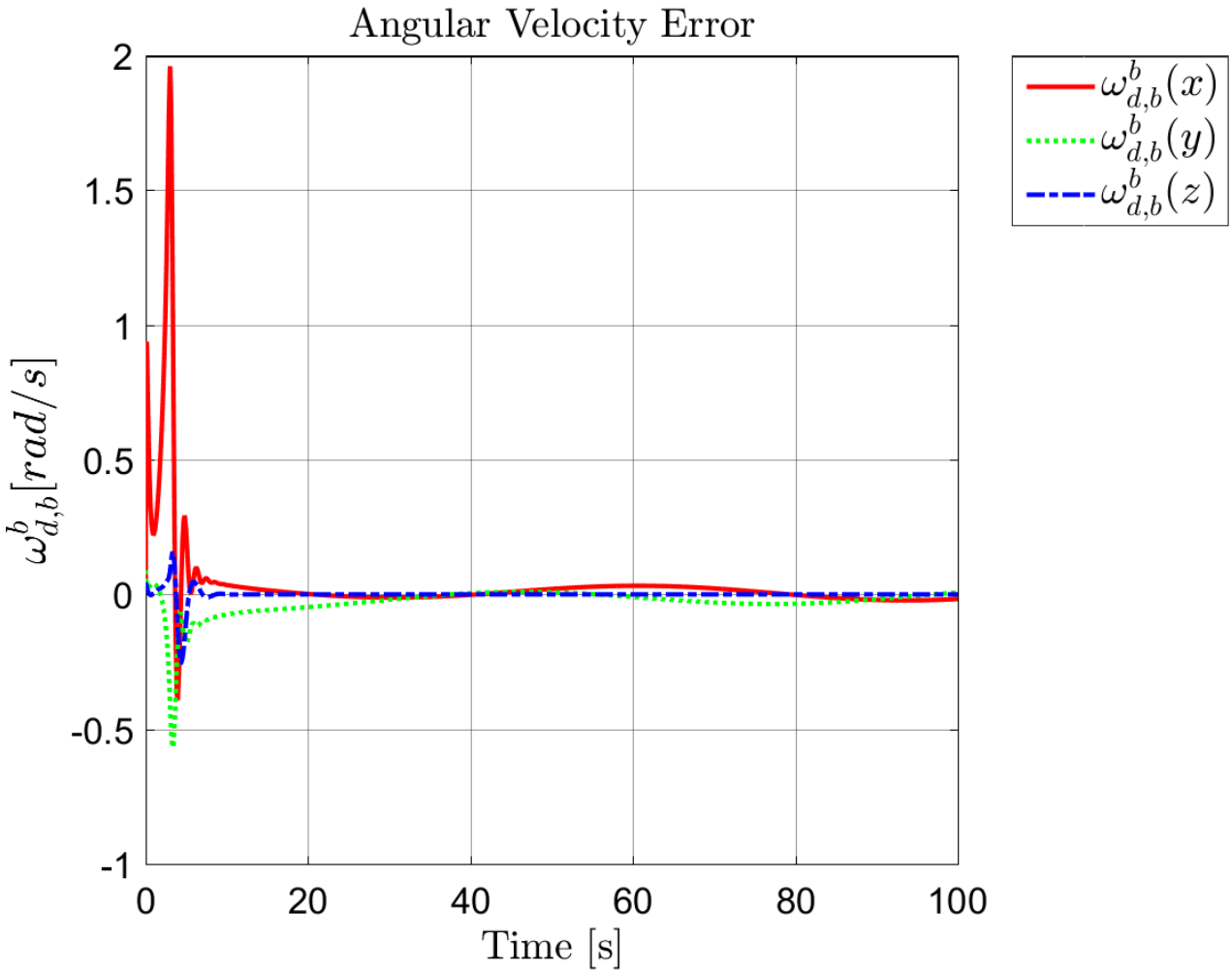


Figure 4.29: Angular Velocity Error, Trajectory Tracking of a Helix

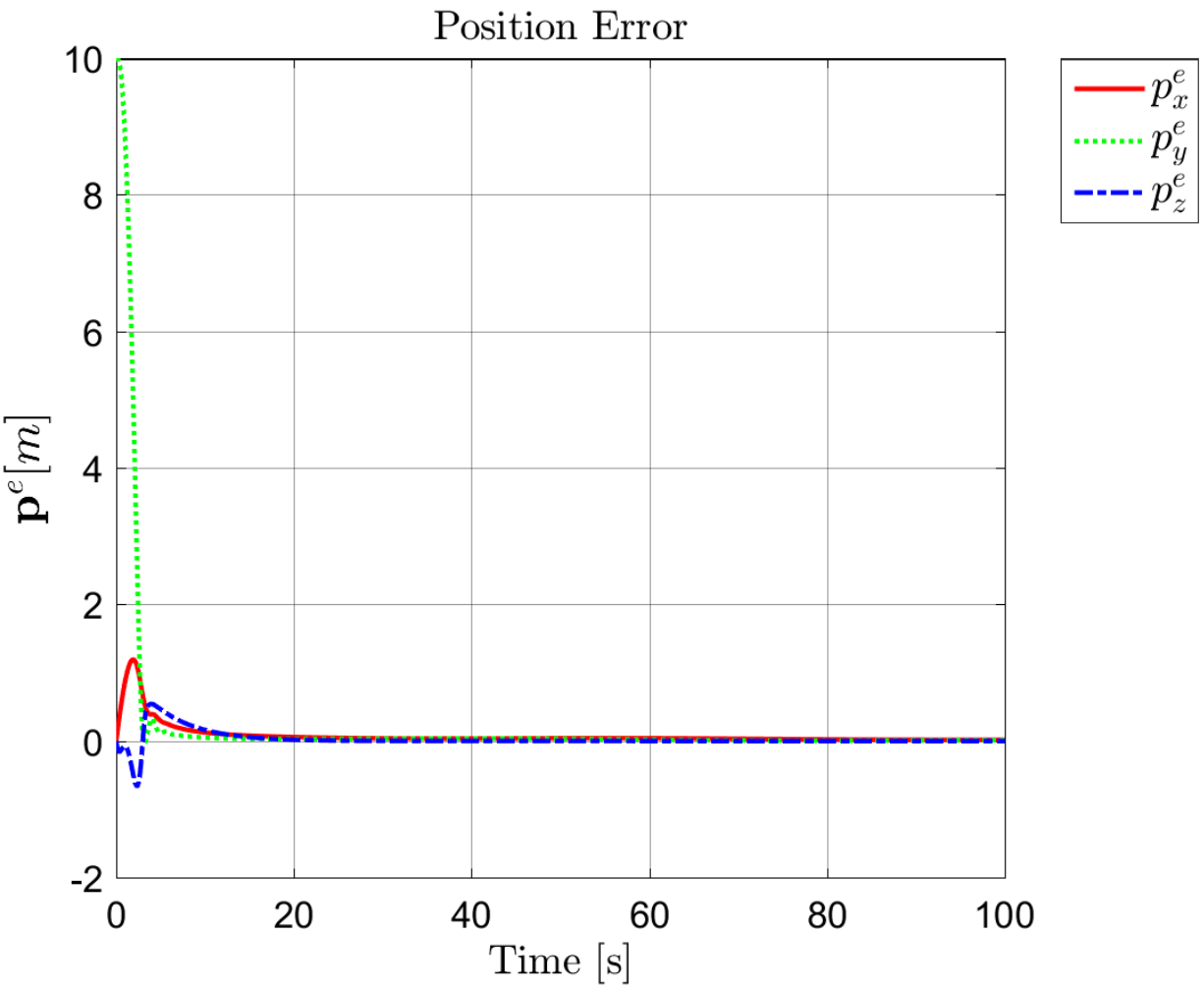
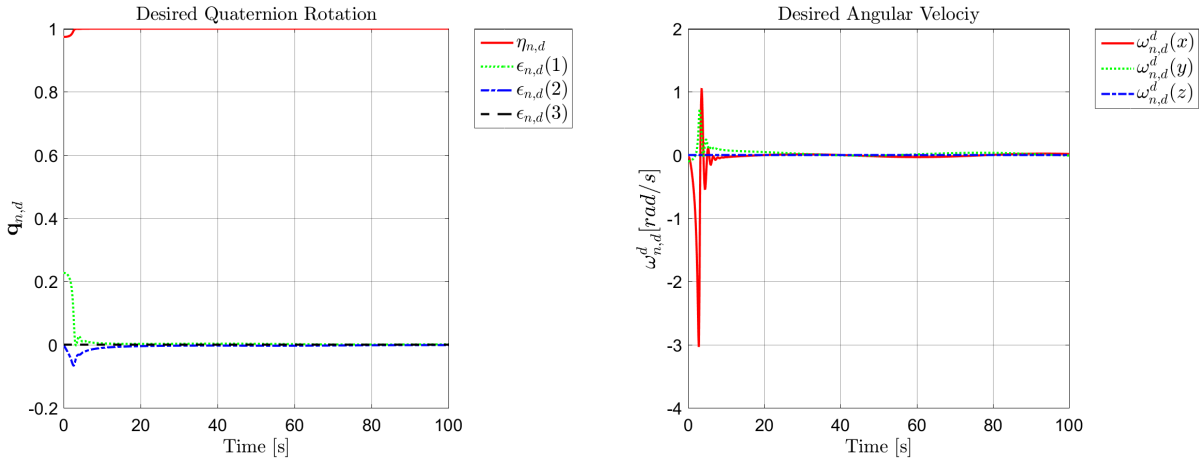


Figure 4.30: Position Error, Trajectory Tracking of a Helix



(a) Desired Quaternion Rotation

(b) Desired Angular Velocity

Figure 4.31: Additional Simulation Results, Trajectory Tracking of a Helix

4.3.4 Simulation Case 4

This section contains the simulation results for the total cascaded system tracking a spiral trajectory.

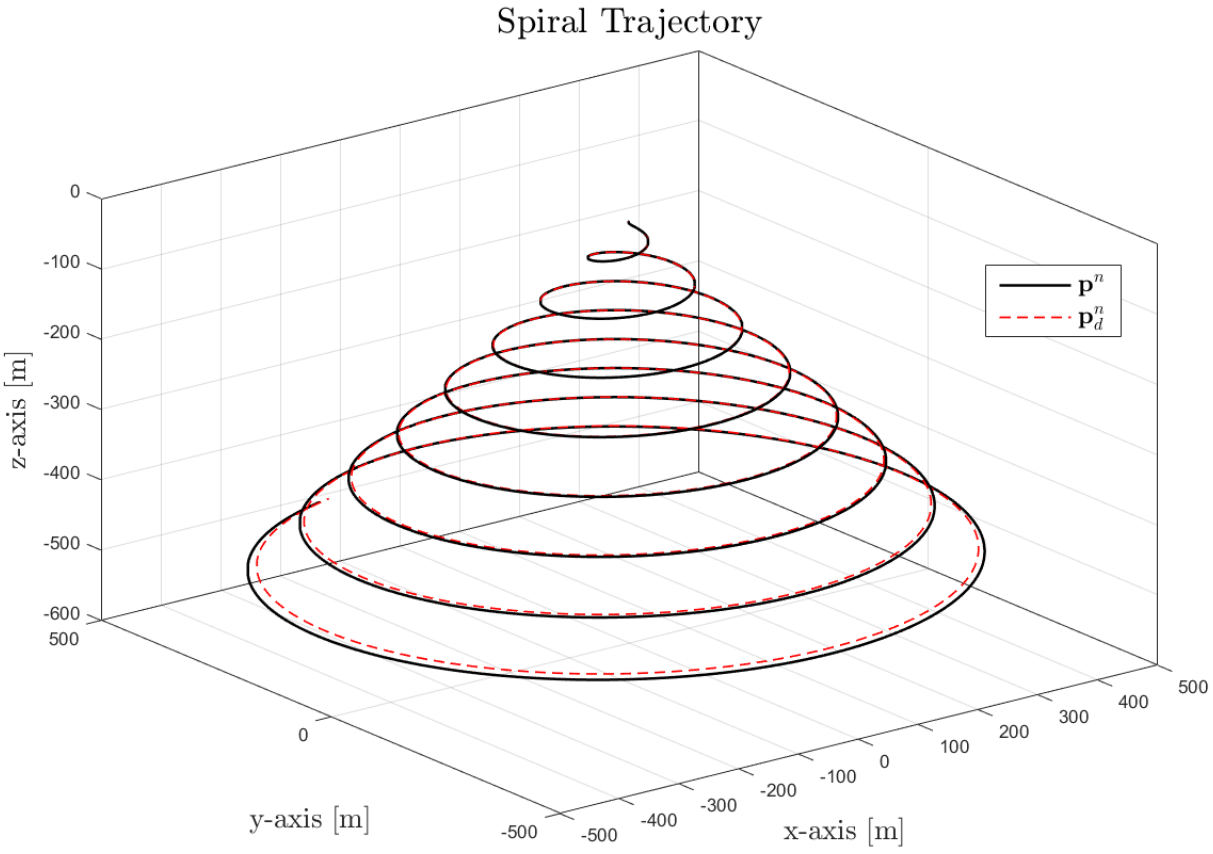


Figure 4.32: Trajectory Tracking of a Spiral

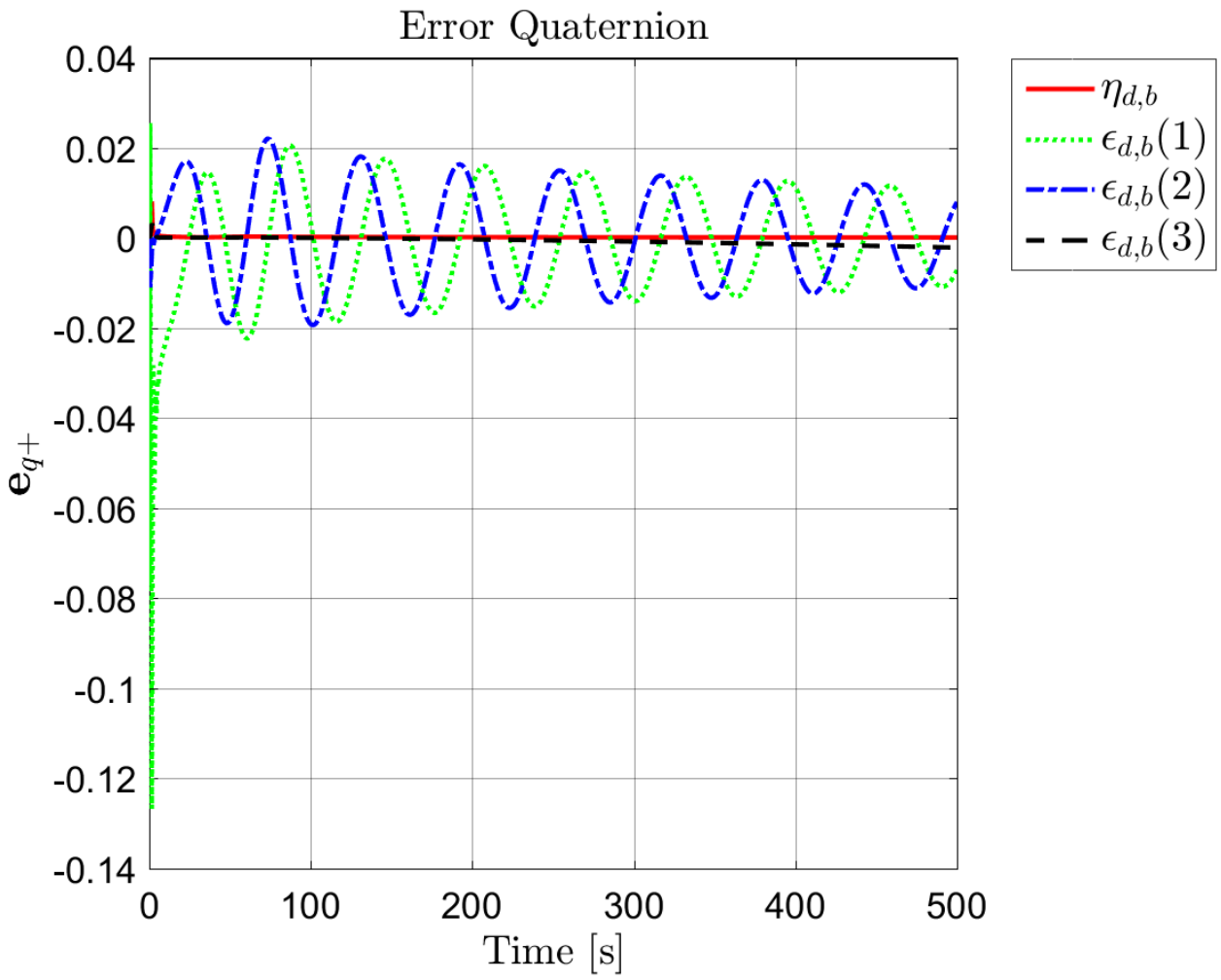


Figure 4.33: Quaternion Error, Trajectory Tracking of a Spiral

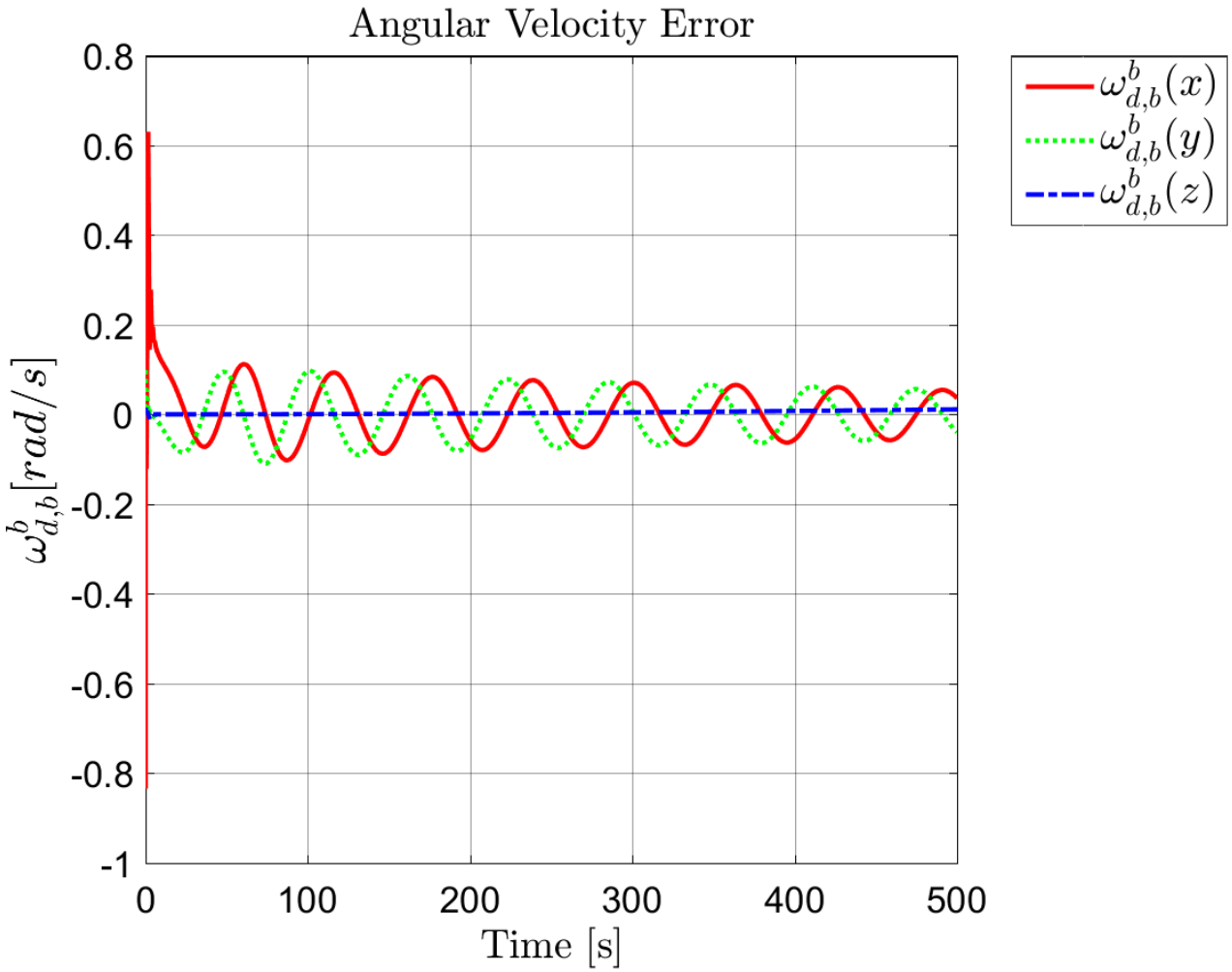


Figure 4.34: Angular Velocity Error, Trajectory Tracking of a Spiral

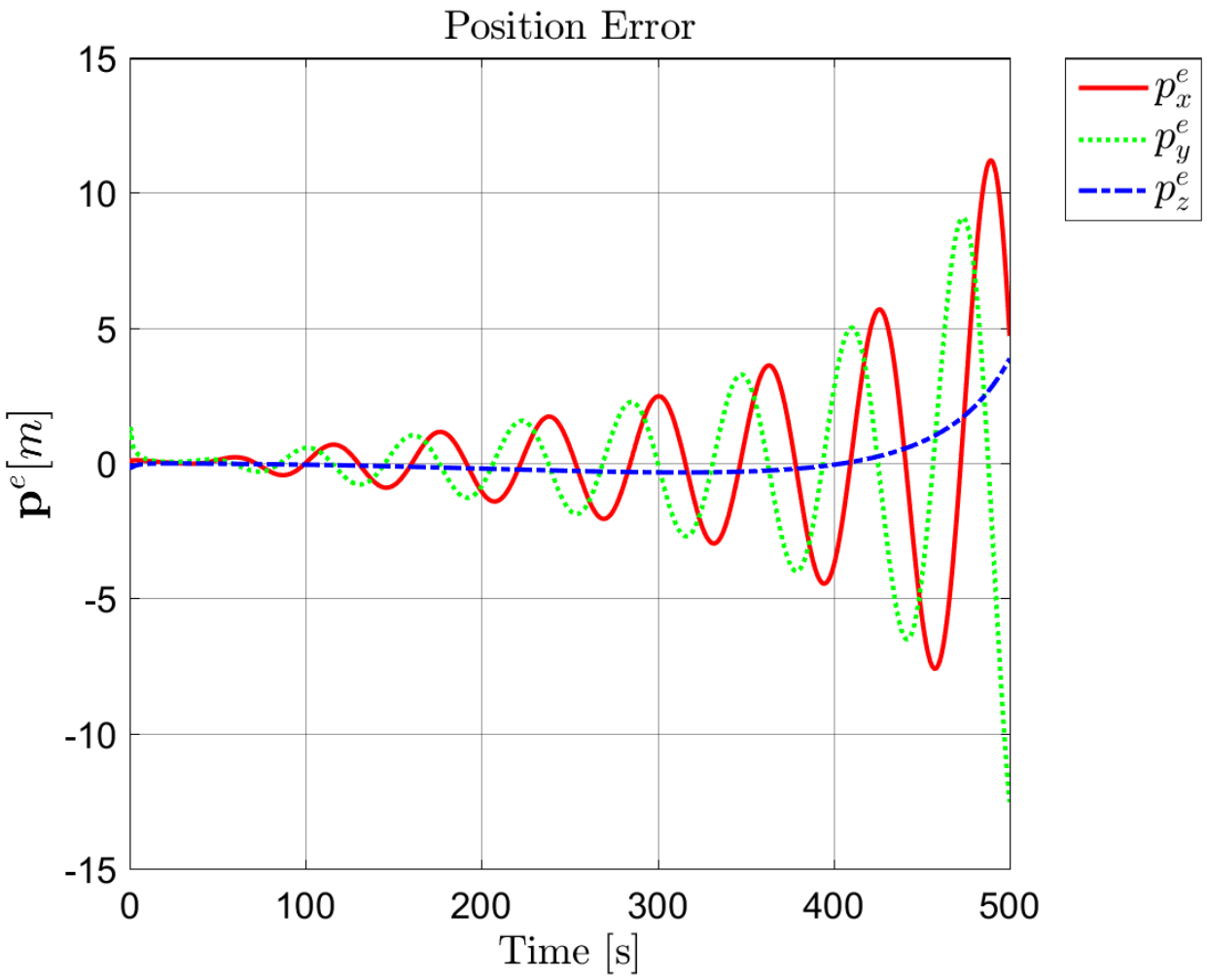
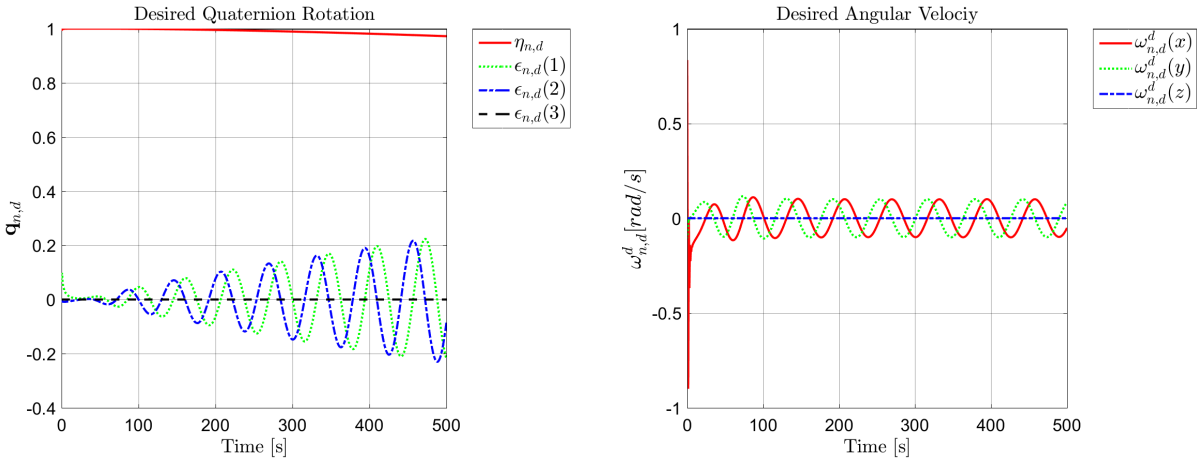


Figure 4.35: Position Error, Trajectory Tracking of a Spiral



(a) Desired Quaternion Rotation

(b) Desired Angular Velocity

Figure 4.36: Additional Simulations, Trajectory Tracking of a Spiral

4.3.5 Simulation Case 5

This section contains the simulation results for the total cascaded system tracking waypoints.

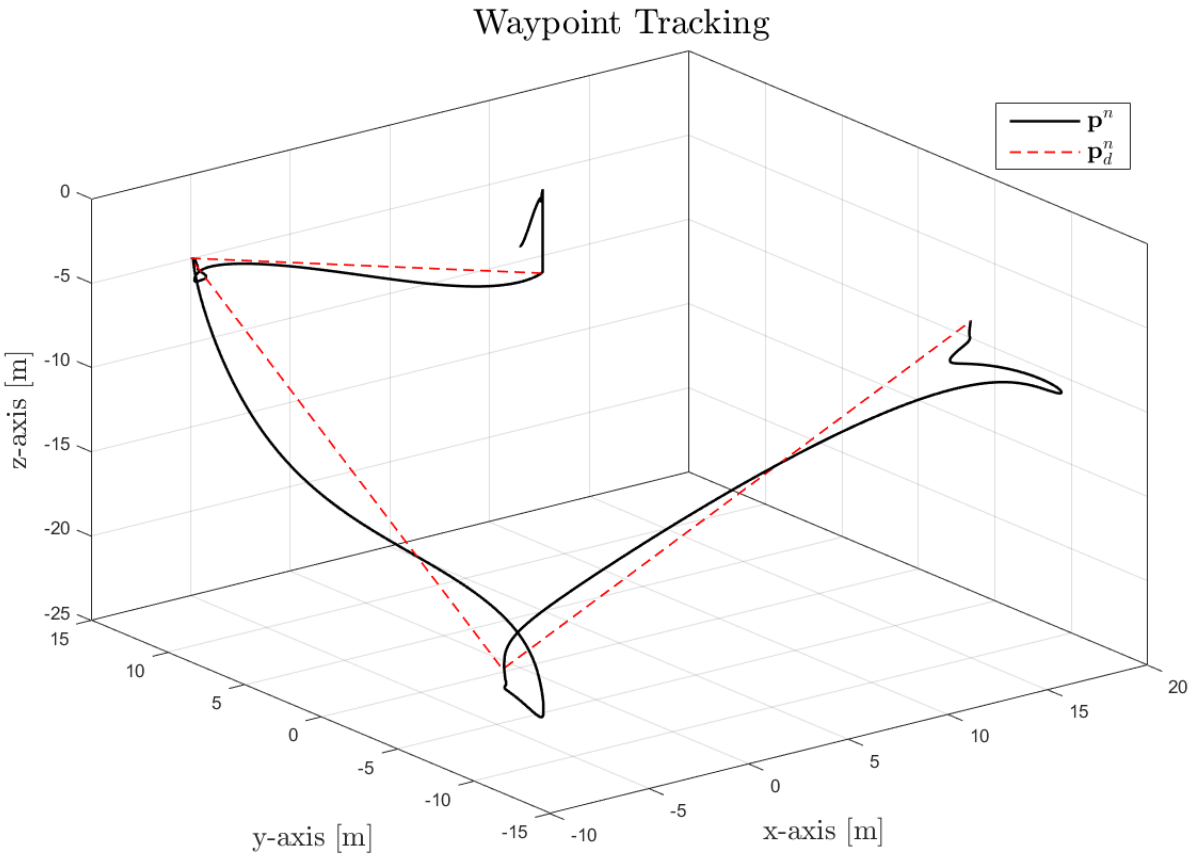


Figure 4.37: Waypoint Tracking

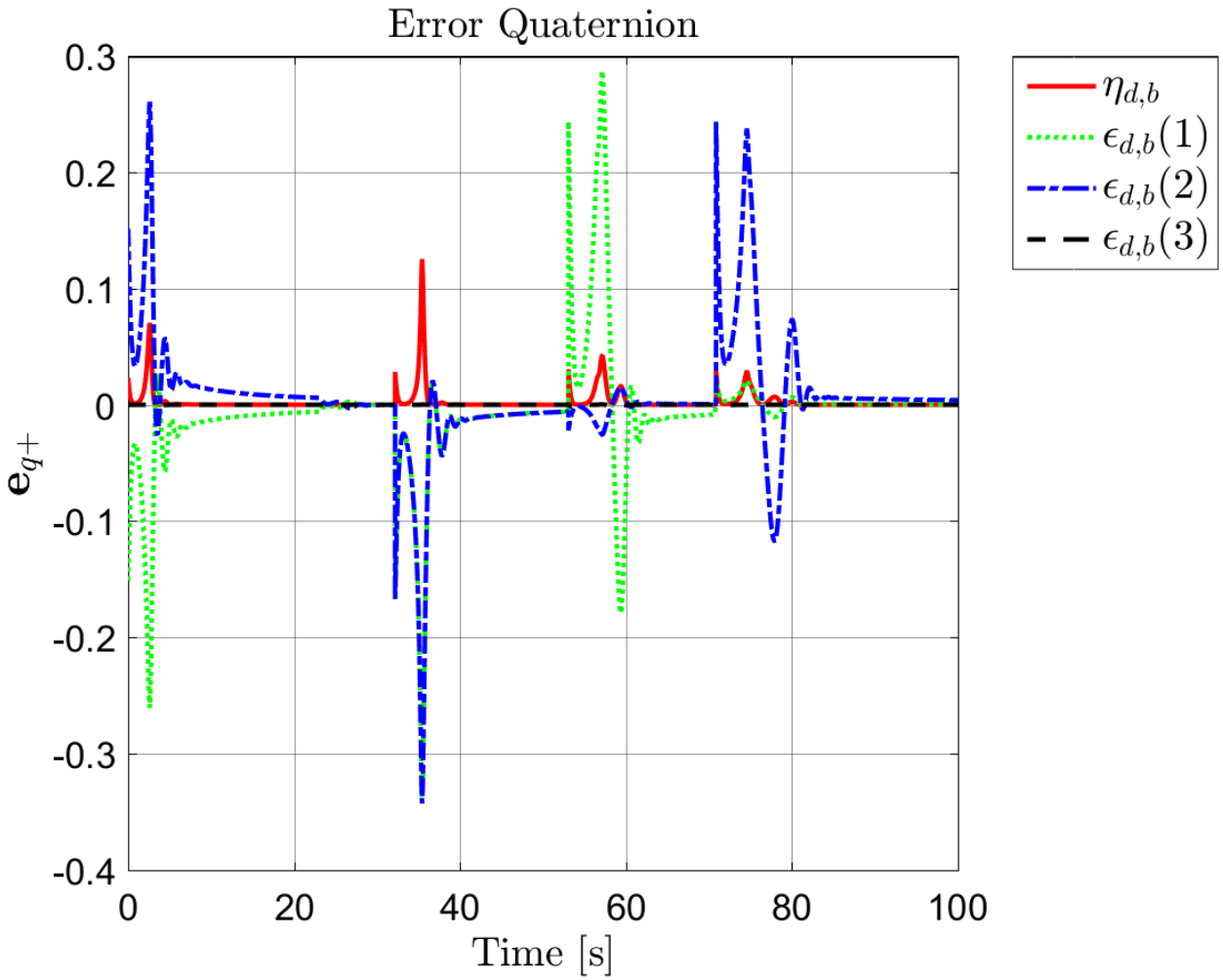


Figure 4.38: Quaternion Error, Waypoint Tracking

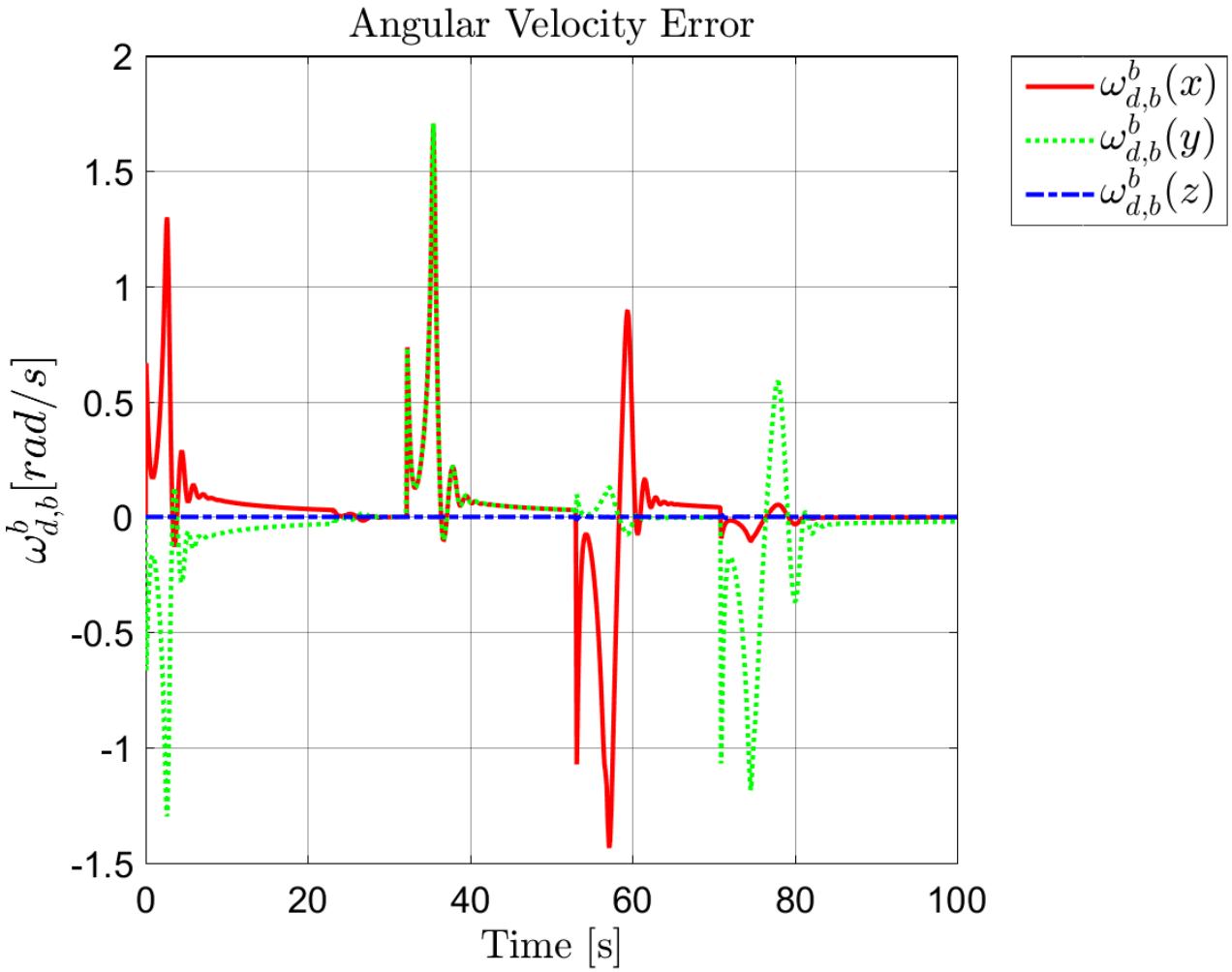


Figure 4.39: Angular Velocity Error, Waypoint Tracking

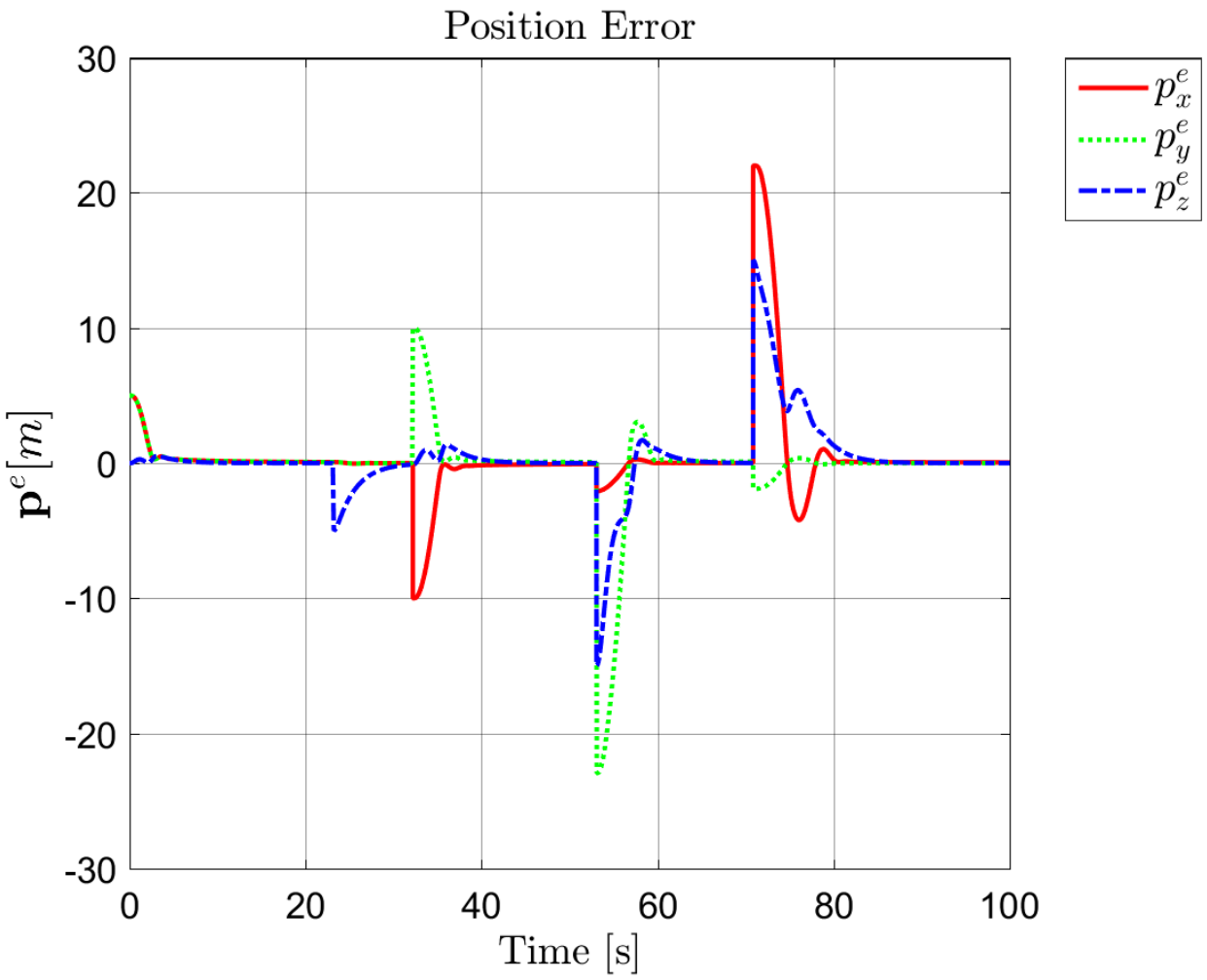
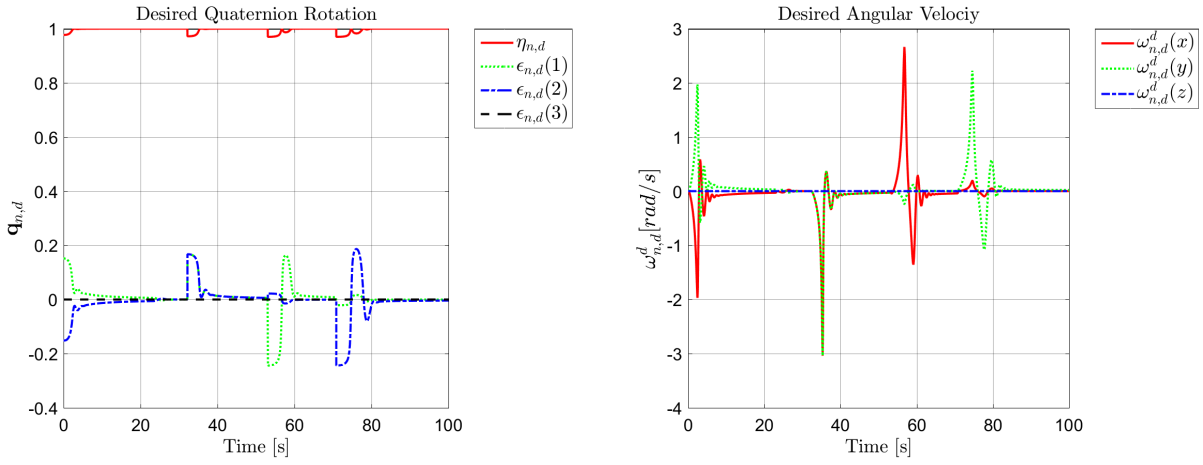


Figure 4.40: Position Error, Waypoint Tracking



(a) Desired Quaternion Rotation

(b) Desired Angular Velocity

Figure 4.41: Additional Simulations, Waypoint Tracking

4.3.6 Simulation Comments

As Figure 4.19, Figure 4.22, Figure 4.27, Figure 4.32 and Figure 4.37 shows, it can be observed that the quadrotor is tracking both the desired fixed position, the circle, helix and spiral trajectories, as well as the waypoints quite smoothly. The figures representing the error measurements show some anomalies, in the form of oscillatory movements where it is expected convergence towards zero. For the spiral trajectory, the position error increases with time. A reason for this might be because the magnitude of the desired acceleration increases over time, a solution to this could be to add more compensation terms.

Chapter 5

Concluding Remarks

5.1 Discussion

In this thesis a mathematical cascaded quadrotor model with passivity-based control has been presented. The model could be considered as a rotational cascade and a translational cascade, where the quaternion rotation from the Body Frame to NED Frame within the rotation cascade generated a rotation matrix, $\mathbf{R}_b^n$, which served as an input to the translation cascade, enabling the quadrotor to point the thrust in a desired direction, causing the quadrotor to fly towards a desired position. Both the rotational and translational controllers have been developed according to the passive GAS PD+ control scheme. The cascades were first decoupled, such that the individual stability of the two cascades could be tested in accordance with cascade stability theories, before they were joined together and the whole cascaded system was tested.

The simulation figures for both the decoupled rotation and translation cascades, along with the mathematical descriptions and results of the systems shows that the rotation and translation in principle are stable. In the rotation cascade, the quadrotor rotates fast and stable towards the desired quaternion, and in the translation cascade, the quadrotor reaches the desired position fast and robustly using both the altitude controller and the vector controller.

Following the theory of cascade interconnections the total system joined together should be stable. The simulation results suggests that the system is stable, nonetheless they give no indications as to what type of stability the system has, which can only be verified by stability proofs.

5.2 Conclusions

As the simulation results clearly show, the cascade interconnected quadrotor system is able to track both fixed positions and positions that change with time, such as the circle, helix and waypoint tracking. The simulation figures for the spiral trajectory tracking shows a growth in position error as the radius of the circle increases, indicating that the control solutions are struggling with increase in acceleration (*jerk*), but is believed to be rectified by additional compensation terms. Overall the system performs well, and can be presumed stable.

5.3 Recommendations for Future Work

The following list includes recommendations for future work as an extension of the work done in this thesis.

- Develop a different guidance system, perform stability proofs for the obtained control solutions and the growth term $g(t, x)x_2$ of the cascaded system.
- Perform detailed simulations by taking atmospherical perturbations into account, and assess the controllers developed in this thesis performances with the perturbing forces present.

- Expand on the model by taking actuator dynamics into account, suggest control solutions based on the added dynamics, and perform simulations of the system.
- Expand on the model by developing alternative control solutions and perform simulations of the system.

Bibliography

Ailon, A. and Arogeti, S. (2015). Closed-form nonlinear tracking controllers for quadrotors with model and input generator uncertainties. *Automatica*, 54:317 – 324.

Alexis, K., Papachristos, C., Nikolakopoulos, G., and Tzes, A. (2011). Model predictive quadrotor indoor position control. In *19th Mediterranean Conference on Control Automation (MED), 2011*, pages 1247–1252.

Beul, M., Worst, R., and Behnke, S. (2014). Nonlinear model-based position control for quadrotor uavs. In *Proceedings of ISR/Robotik 2014; 41st International Symposium on Robotics;*, pages 1–6.

Crouch, T. (2004). *A History of Aviation from Kites to the Space Age*. W.W. Norton & co.

Egeland, O. and Gravdahl, J. T. (2002). *Modeling and Simulation for Automatic Control*. Marine Cybernetics.

Hoffmann, G., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Jang, J. S., and Tomlin, C. J. (2004). The stanford testbed of autonomous rotorcraft for multi agent control (starmac). In *Digital Avionics Systems Conference, 2004. DASC 04. The 23rd*, volume 2, pages 12.E.4–121–10 Vol.2.

Khalil, H. K. (2002). *Nonlinear Systems Third Edition*. Prentice Hall.

Kristiansen, R. (2008). *Dynamic Synchronization of Spacecraft Modeling and Coordinated Control of Leader-Follower Spacecraft Formations*. PhD thesis, NTNU.

Lamnabhi-Lagarrigue, F., Lora, A., and Panteley, E. (2004). Advanced topics in control systems theory. Lecture notes.

Loría, A. (2004a). Cascaded nonlienar time-varying systems: Analysis and design. Lecture notes from FAP 2004 European Control Training Site Gif sur Yvette, France, March, 5 2004.

Loría, A. (2004b). Cascaded systems: Stability and stabilization. Lecture notes.

Mahoney, R., Kumar, V., and Corke, P. (2012). Multirotor aerial vehicles, modeling, estimation and control of quadrotor. *IEEE Robotics &Automation Magazine*.

Mian, A. A. and Daobo, W. (2008). Modeling and backstepping-based nonlinear control strategy for a 6 dof quadrotor helicopter. *Chinese Journal of Aeronautics*, 21(3):261 – 268.

Oland, E. (2014). *Nonlienar Control of Fixed-Wing Unmanned Aerial Vehicles*. PhD thesis, NTNU.

Ortega, R., Loria, A., Nicklasson, P. J., and Sira-Ramirez, H. (1998). *Passivity-based Control of Euler-Lagrange Systems*. Springer-Verlag London.

Ortega, R. and Spong, M. W. (1988). Adaptive motion control of rigid robots: a tutorial. In *Decision and Control, 1988., Proceedings of the 27th IEEE Conference on*, pages 1575–1584 vol.2.

Paden, B. and Panja, R. (1988). Globally asymptotically stable 'pd+' controller for robot manipulators. *Int J. Control vol. 47:1697-1712*.

Pounds, P., Mahony, R., and Corke, P. (2010). Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691 – 699. Special Issue on Aerial Robotics.

Schlanbusch, R. (2012). *Control of Rigid Bodies with Applications to Leader-Follower Spacecraft Formations*. PhD thesis, NTNU.

Stengel, R. (2014). Aircraft equations of motion - 2. Aircraft Flight Dynamics MAE, 2014.

Takegaki, M. and Arimoto, S. (1981). A new feedback method for dynamic control of manipulators. *J. Dun. Sys., Meas., Control 103(2) : 119-125.*

Appendix A

CD index

In the following directories on the attached CD you may find:

- THESIS: The thesis in .pdf format
- FIGURS: Folder containing all figures used in the thesis
- MATLAB AND SIMULINK: The Simulink models used for simulations and Matlab scripts used to calculate quaternions and create figures.
- PAPERS: A selection of references