Czech Technical University in Prague

Faculty of Electrical Engineering
Department of Control Engineering

Master's Thesis

Modeling, Identification and Control of a Quadrotor Aircraft

Marcelo De Lellis Costa de Oliveira

Supervisor:
Ing. Zdeněk Hurák, Ph.D.

Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague

Prague, June 2011

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Control Engineering

DIPLOMA THESIS ASSIGNMENT

Student: **De Lellis Costa De Oliveira Marcelo**

Study programme: Cybernetics and Robotics
Specialisation: Systems and Control

Title of Diploma Thesis: **Modeling, identification and control design for a micro quadrotor.**

Guidelines:

The goal of this project is to design and experimentally verify relevant feedback controllers for a micro quadrotor. The necessary preliminary steps are first-principles modeling and experimental identification so that the control design is based on a mathematical model (implemented in Simulink). The flying device has been designed within another graduation project, hence this work will have to be conducted in close collaboration with that student.

Bibliography/Sources:

[1] Paul Pounds, Robert Mahony, a Joel Gresham, Towards Dynamically Favourable Quad-Rotor Aerial Robots, in Australian Conference on Robotics and Automation, 2004.

[2] P. Castillo, A. Dzul, a R. Lozano, Real-time stabilization and tracking of a four-rotor mini rotorcraft,Control Systems Technology, IEEE Transactions on, roč. 12, čís. 4, s. 510-516, 2004.

[3] S. Bouabdallah, A. Noth, a R. Siegwart, PID vs LQ control techniques applied to an indoor micro quadrotor, in Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on, roč. 3, s. 2451-2456 vol.3, 2004.

Diploma Thesis Supervisor: Ing. Zdeněk Hurák, Ph.D.

Valid until summer semester 2011/2012

L.S.

prof. Ing. Michael Šebek, DrSc.
Head of Department

prof. Ing. Boris Šimák, CSc.
Dean

Prague, December 2, 2010

Abstract

The present work refers to the mathematical modeling, experimental identification and control design of a small unmanned indoors quadrotor aircraft, at low translational speeds around the hovering condition, where the aerodynamic forces on the airframe are disregarded. A Kalman filter is implemented for state estimation and noise filtering. Linear control techniques such as PID, LQ as well as modern robust mixed-sensitivity $\mathcal{H}_\infty$ and μ-synthesis with DK-iteration are employed and compared with each other in terms of flight trajectory reference tracking and parametric and model uncertainty.

Declaration

I hereby attest that all content presented in this work is a result of my own work whereas all used previous material and references are duly indicated.

Prague, May 20, 2011

Marcelo De Lellis Costa de Oliveira

Acknowledgments

First of all I would like to thank the Erasmus Mundus as well as the SpaceMaster program for not only allowing me the opportunity of undertaking this Master course in space science and technology, but also providing me with the necessary financial support throughout my whole studies. I am especially grateful to the all the administrative and teaching staff of the Department of Control Engineering at the Czech Technical University in Prague, in particular to Jana Nováková, for her help with official procedures and arrangements, Ing. Ph.D. Martin Hromčík, Ing. Tomáš Haniš and Ing. Martin Řezáč, for their admirable dedication and effort in teaching, to my colleague Jaromír Dvořák, for the project discussions and collaboration, and to my supervisor Ing. Ph.D. Zdeněk Hurák, for his constant advise and close cooperation, without whom this thesis would not have been possible. I feel very proud of having been part of such a team.

My most sincere gratitude goes also to my Czech friends Petr Beneš and his parents, for kindly providing me with accommodation and helping me to feel at home, among family, to my dear friend Michaela Řehulková, for her friendship and encouragement, especially when times were tough, as well as to my beloved girlfriend Anastasiya Bolekhan, for bringing me a greater meaning for all of this and making it all easier and joyful.

Last but not least I would like to thank all my family and close friends back in Brazil, especially my mother Rejane, for believing in me and encouraging me to set off after my dreams, may they be eleven thousand kilometers away, on the other side of the ocean.

Thank you all,

Marcelo

Contents

List of Figures

List of Tables

Acronyms

ARE Algebraic Riccati Equation. 23, 32

BDCM Brushed DC Motor. 6, 7

BLDCM Brushless DC Motor. 6, 7, 9, 13, 14, 24

DCE Department of Control Engineering. 1, 46

DOF Degree of Freedom. 24, 27, 30, 33, 35–37, 42

GPS Global Positioning System. 24

LS Linear System. 43, 44

LTI Linear Time-Invariant. 23, 24, 42

MEMS Micro-Electromechanical Systems. 1, 20, 22

MIMO Multiple-Inputs-Multiple-Outputs. 27–31, 38

MS Mixed Sensitivity. 38, 39, 41, 42

NLS Non-Linear System. 43, 44

SISO Single-Input-Single-Output. 29, 30, 38

UAV Unmanned Aerial Vehicle. 1, 3

Chapter 1

Introduction

The present work is the realization of the graduate student's special interest on studying aircraft flight dynamics and control combined with the interest from the Department of Control Engineering (DCE) of this university in acquiring expertise on miniaturized Unmanned Aerial Vehicles (UAVs), for future employment in studies of swarm robotics and collective behavior. It also closely relates to the work of Dvořák [2011], another graduate student at the department, who was already working on the assembly of such an aircraft for a private company. In this sense, the author's work could not only collaborate with his colleague's, but also take advantage of the physical real system already built, aiming at comparing and validating his theoretical and simulation results, whenever feasible.

1.1 Project Goals

A fortunate aspect of this work is that it could rely on extensive previous studies and publications, of which a brief bibliographic survey is to be presented next. The author's primordial goal herewith is hence to obtain a thorough understanding of the system's behavior whereas using it as a case study for experimenting with control techniques and other abilities developed throughout his graduate course.

From the DCE's side, the main objective was to obtain a complete and as realistic as possible, yet without unnecessary complexity, simulation model in Matlab/Simulink, with this dissertation as the main project's documentation, allowing the immediate further use of the quadrotor in future studies.

Finally, by means of this work, the author could collaborate with his colleague in the design and tuning of the high-level control loops, while the latter could focus on practical implementation issues with the hardware, such as sensor data fusion and signal filtering, as well as experimenting with different approaches like quaternion-based attitude representation and control with eigenaxis rotations.

1.2 Bibliographic Survey

In the recent years, especially due to advances in Micro-Electromechanical Systems (MEMS), electrical energy accumulators, actuators and smaller integrated micro-controlled boards, a growing number of studies in UAVs such as the quadrotor and related autonomous aerial robots has been carried out, not only by universities and research institutions for private civilian applications but also for military purposes, mainly due to the inherent characteristics of such aircraft, namely high maneuvering at low translational speeds and in small volumes while being able to carry significant payload, thus making them especially adequate for aerial surveillance and monitoring tasks.

The main advantage of rotating-wing over fixed-wing aircraft is the ability of hovering and omni-directional movement. A drawback is, however, a relatively higher power consumption during the flight. Even inside the rotating-wing aircraft classification, a quadrotor is much simpler and easier to build in comparison to a classical helicopter, since the rotors' rotational axis is fixed and there are no moving parts, like aerodynamic control surfaces. Nevertheless, the rotational speed of each rotor needs to be independently controlled in order to achieve the control goals of such a highly unstable open-loop system, what makes it a challenging control engineering problem.

As said, extensive literature has been produced in this field of study. Bouabdallah et al. [2004a] presented a system model with DC motors and, by using the Lyapunov Function's non-linear control technique approach for stabilizing the aircraft's orientation (Euler angles), compare the real system's behavior with a respective simulation. Bouabdallah et al. [2004b] extended their work on the *OS4 project* as they compared classical PD and PID controllers for orientation stabilization with modern LQ adaptive optimal control, despite realizing that the latter one yielded only average results, due to modeling imperfections.

Stepaniak [2008] made a detailed identification work of his built system and model derivation, besides discussing hardware implementation aspects. Despite not focusing on the control loops design, whereby classical control theory was used, his work turned out to be one of the main references hereby used.

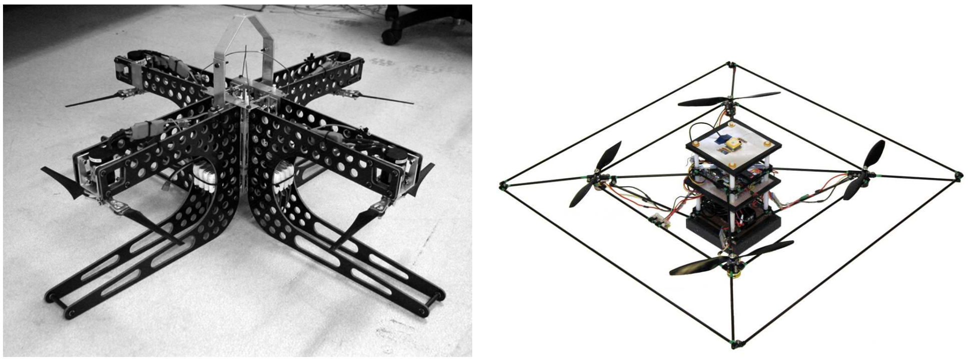
Tayebi and McGilvray [2006], on the other hand, performed a thorough and advanced study on control techniques for attitude stabilization of a quadrotor. It was used quaternions for attitude representation, Lyapunov stability's criterion and a PD^2 feedback structure, with which a model-independent PD controller was compared, achieving with both configurations global asymptotic stability and disturbance rejection in similar fashion. Their experimental results were obtained from a modified version of the *Draganflyer III* aircraft.

Kim et al. [2007] made an interesting performance comparison among four control techniques: LQR, LQR with gain scheduling, feedback linearization and sliding mode control. They experimentally verified that LQR with gain scheduling presented more robustness in light of modeling uncertainties whereas for an accurately modeled system a better performance was achieved with the sliding mode approach.

A meticulous study of usually otherwise disregarded effects such as blade flapping and propeller modeling was done by Pounds et al. [2004]. In fact, the theoretical model for the propeller thrust and torque discussed in their paper was employed in this present work. Later on, Pounds et al. [2006] gave continuity to their work on the *X-4 flyer Mark II* quadrotor implementation (Fig. 1.1a) by designing a discrete-time PID control law to their model including the very fast blade flapping dynamics. The closed-loop behavior, though, turned out to be poor at higher rotor angular speeds ($\omega > 450\,\mathrm{rad/s}$), approaching an unstable behavior, which was attributed to high-frequency noise from the rotors interfering with the accelerometer readings.

Hamel et al. [2002] employed the non-linear Lyapunov functions and the backstepping approach allied to quaternion attitude representation for designing non-linear attitude stabilization controllers. Although presenting a minute theoretical study and proof-based mathematical derivations, their proposal unfortunately was not accompanied by experimental results.

Castillo et al. [2004] proposed a real-time non-linear nested saturation control scheme based on Lyapunov's stability criterion, where each system state is sequentially stabilized following a priority rule, hence allowing a wider stability region and therefore more aggressive maneuvering while maintaining good disturbance rejection capability. Later on, Castillo et al. [2005] compared the performance of their non-linear control with a linear one such as LQR, which presents stability issues when the system is taken far away from its operation point used for the controller design.



(a) X-4 Flyer Mark II.

(b) STARMAC II.

Figure 1.1: Examples of quadrotor implementations found in the literature.

Martínez [2007] performed an extensive identification experimental study on a commercial *Draganfly XPro* quadrotor, including blade flapping and torsion investigations besides wind tunnel tests to identify the aircraft's aerodynamic characteristics. Since his work didn't extend to the design of control loops, its use for this project was limited as another reference for cross-checking the basics aircraft modeling hereby dealt with.

Voos [2009] applied feedback linearization in a nested control loop structure where the inner loop contains the attitude dynamics and the outer, the position one. Their experimental results were obtained also with a *Draganflyer* real model. However, the dynamics of the DC motors used in the rotors was not considered in their work.

Hoffmann et al. [2007] focused their study on the aerodynamic effects on the quadrotor's airframe when operating significantly far from the hover regime, at higher translational speeds, while also considering the very fast dynamics of blade flapping. Their theoretical results were experimented on the *STARMAC II* vehicle.

1.3 Commercially Available Quadrotors

Quadrotor implementations and studies do not limit themselves to the academic environment. Especially in the last decade, several commercially available models have appeared in the market, with a variety of models stretching from mere entertainment up to serious applications.

For example, the German company `Microdrones GmbH`[mic] was established in 2005 and since then has been developing such UAVs for tasks such as aerial surveillance by police and firemen forces, inspection services of power lines, monitoring of nature protection areas, photogrammetry, archeology research, among others. Their smallest model is pictured in Fig. 1.2a and has technical characteristics summarized in table 1.1, illustrating the category of quadrotors that this work relates to.

Another manufacturer of such aircraft is the Canadian `Draganfly Innovations Inc.`[dra]. Their quadrotor models portfolio stretches from the *Draganflyer X4*, with 250 g of payload capacity up to the *Draganflyer X8*, illustrated in Fig. 1.2b, featuring a 8-rotor design, with payload capacity of 1000 g and GPS position hold function.

Still another relevant manufacturer of quadrotors, among other products, is the French company `Parrot SA`[par]. Their *AR.Drone* model, pictures in Fig. 1.2c with a surrounding protective frame, is comparable in size to the *md4-200* of `Microdrone`, however it can fly only for approximately 12 minutes, reaching a top speed of 18 km/h. It was designed for entertainment purposes, including video-gaming and augmented reality, and can be remote-controlled by an iPhone® through a Wi-Fi network. *AR.Drone* is currently available on

Typical take-off weight	1000 g
Diameter (between rotor axes)	70 cm
Flight autonomy	max. 30 min
Flight radius	(500 - 2000) m
Air humidity	max. 80 %
Air temperature	(0 - 40) °C
Wind speed	max. 4 m/s

Table 1.1: Technical characteristics of md4-200 quadrotor from `Microdrones`.

(a) *md4-200* from `Microdrone`.

(b) *Draganflyer X8* from `Draganfly Innovations`.

(c) *AR.Drone* from `Parrot`.

Figure 1.2: Examples of commercially available quadrotors.

`amazon.com` for approximately US$ 300.

1.4 Document Structure

We start the dissertation in chapter 2 by considering the main moments and forces acting on the quadrotor, stating some assumptions, briefly presenting coordinate system transformation and then proposing the mathematical non-linear dynamic model of the quadrotor.

Next, chapter 3 deals with the identification of all modeled system parameters by means of practical experiments with the real aircraft, executed in collaboration with Dvořák [2011]. This allowed validation of the proposed model and finally lead to the full non-linear and linearized model with which further work was done.

Chapter 4 shortly presents two ways of how the aircraft's orientation in Euclidian space can be assessed, followed by an optimal noise filtering of the sensor readings and state estimation by means of Kalman filtering.

Control design constitutes one of the main topics approached in this work and is presented in chapter 6. First, a control architecture is proposed and classical PID control is designed, followed by the optimal LQ-state feedback technique. Robustness to parametric and model uncertainties is an aspect which the previously mentioned control techniques do not deal with, therefore mixed-sensitivity $\mathcal{H}_\infty$ as well as μ-synthesis with DK-itertation based controllers are designed and discussed.

Finally, chapter 7 concludes this author's work by highlighting the main results obtained, proposing some improvements to be done in what has been already achieved as well as outlining the continuation of this project.

Chapter 2

Dynamic Model

In this chapter, the flight dynamics model of the quadrotor will be derived. We will use a top-down approach, starting with the overall rigid-body dynamics, investigating the forces and moments acting on it, and then finally discussing the actuator subsystem (rotor).

2.1 Rigid-Body Dynamics

The first step to achieve the dynamic model of the quadrotor is to define frames of reference, each with its defined right-handed[1] coordinate system, as shown in Figure 2.1. For the body-fixed one, X,Y and Z are its orthogonal axes with its correspondent body linear velocity vector $\vec{V} = {}^{b}\begin{bmatrix} u & v & w \end{bmatrix}^{T}$ and angular rate vector $\vec{\Omega} = {}^{b}\begin{bmatrix} p & q & r \end{bmatrix}^{T}$. The second one is an Earth-fixed inertial (navigation) coordinate system, with which initially the body-fixed coincides. The attitude of the aircraft is assessed by means of successive rotations around each one of the inertial axes, expressed in terms of the *Euler angles* ϕ (roll), θ (pitch) and ψ (yaw).

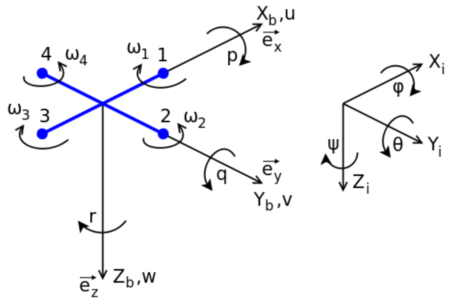


Figure 2.1: Quadrotor's body-fixed and inertial coordinate systems.

In an inertial frame of reference n it is known that the torque (moment) is defined as the time derivative of the angular momentum

$$ {}^{n}\vec{M} \triangleq \frac{d\vec{L}}{dt} = \frac{d}{dt}\left(\vec{I}\cdot\vec{\Omega}\right) \tag{2.1}$$

where $\vec{I}$ is the body's inertia tensor

$$\vec{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.2}$$

[1] Where the right-hand rule applies for determining the direction of a vector cross-product.

where the simplification $\forall i \neq j \Rightarrow I_{ij} = 0$ applies for a well-conditioned (symmetric) mass distribution of the aircraft. In plain words, one can think of an experiment where one axis is freely spinning while the others are still, and a moment applied to the rotating axis generates a variation of angular speed only on itself and not on the others two. This is assumed to be the case here.

In order to make the calculations easier and more intuitive, though, it is better to consider the moments exerted on the quadrotor's airframe directly in the body-fixed rotating frame. By doing so, as presented in details by Stevens and Lewis [2003], the Coriollis effect appears as a second term to be added to (2.1), and considering now $\vec{L} = I_x\, p\, \vec{e}_x + I_y\, q\, \vec{e}_y + I_z\, r\, \vec{e}_z$, it yields

$$^{b}\vec{M} = \left(\frac{d\vec{L}}{dt}\right)_{\text{rot}} + \vec{\Omega} \times \vec{L} \Longleftrightarrow {}^{b}\vec{M} = \vec{I} \cdot \dot{\vec{\Omega}} + \vec{\Omega} \times \left(\vec{I} \cdot \vec{\Omega}\right) \tag{2.3}$$

which is the particular vector form of Euler's equations. By developing the cross-product term its algebraic form is found as the set of equations

$$\begin{aligned} {}^{b}M_x &= I_x\, \dot{p} + (I_z - I_y)\, q\, r \\ {}^{b}M_y &= I_y\, \dot{q} + (I_x - I_z)\, r\, p \\ {}^{b}M_z &= I_z\, \dot{r} + (I_y - I_x)\, p\, q \end{aligned} \tag{2.4}$$

which are also referred to as the *moment equations*. One can note the physical natural sense in these equations: the simultaneous rotation around two axis will generate a torque around a third axis, given that the previous causal two axis don't have the same inertia.

Similarly to the reasoning applied so far to the rotational aspect of the rigid-body dynamics, in the translational case a force is generated in the inertial frame, according to Newton's 2nd law, as

$$^{n}\vec{F} \triangleq \frac{d\vec{P}}{dt} = \frac{d}{dt}\left(m \cdot \vec{V}\right) \tag{2.5}$$

where m is the total mass of the quadrotor in whose center the origin of the aircraft's fixed-body coordinate system is located. Once again turning to the body-fixed rotating frame and defining $\vec{P} = m\, u\, \vec{e_x} + m\, v\, \vec{e_y} + m\, w\, \vec{e_z}$, (2.5) becomes

$$^{b}\vec{F} = \left(\frac{d\vec{P}}{dt}\right)_{\text{rot}} + \vec{\Omega} \times \vec{P} \Longleftrightarrow \vec{F} = m\left(\dot{\vec{V}} + \vec{\Omega} \times \vec{V}\right) \tag{2.6}$$

By solving the cross-product, the set of *force equations* is obtained

$$\begin{aligned} {}^{b}F_x &= m\,(\dot{u} + q\, w - v\, r) \\ {}^{b}F_y &= m\,(\dot{v} + r\, u - w\, p) \\ {}^{b}F_z &= m\,(\dot{w} + p\, v - q\, u) \end{aligned} \tag{2.7}$$

We shall now investigate all those moments and forces which act, respectively, on (2.4) and (2.7). The quadrotor is basically subject to five sources of interactions: gyroscopic effects from the rotors' spin, propeller drag torque, thrust, Earth's gravity and the aerodynamic forces. Assuming low translational speeds of the quadrotor, however, the latter are very small and will thus be disregarded henceforth.

2.2 Rotors

Each of the four rotors comprises a Brushless DC Motor (BLDCM) attached to a two-blade propeller. The BLDCM differs from the conventional Brushed DC Motor (BDCM) in their concept essentially in that the commutation of the input voltage applied to the armature's circuit is done electronically, whereas in the latter, by a mechanical commutator (brush) which, as any rotating mechanical device, suffers wear throughout its operation, and as a consequence, confers the BDCM a significant shorter nominal life time than the newer BLDCM. Detailed information on these and other electric motors are presented by Krishnan [2010].

Despite the extra complexity in its electronic switching circuit, the BLDCM presents several advantages over its counterpart, to name a few: higher torque/weight ratio, less operational noise, longer lifetime, less generation of electromagnetic interference and much more power per volume, practically limited only by its inherent heat generation, whose transfer to the outer environment usually occurs by conduction.

In spite of their performance differences, the BLDCM's dynamic model can be roughly approximated by the well-known BDCM's. Fig. 2.2 shows the basic electrical circuit of such a motor, where u is the voltage applied to its armature, R_a is the armature's resistance, L_a its inductance, $v_b = k_v\, \omega$ is the back-electromotive force induced in the armature, k_v is the speed constant and ω_a is the angular speed.

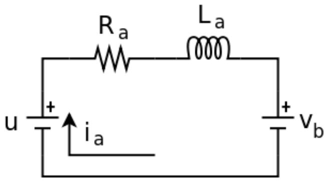


Figure 2.2: Basic electric model of a brushed DC motor.

Applying Kirchhoff's voltage law to the circuit and then the Laplace transform yields

$$i_a = \frac{u - k_v\, \omega}{L_a\, s + R_a} \tag{2.8}$$

Before considering the mechanical aspect of the rotor, we shall first analyze two effects.

2.2.1 Gyroscopic Effect

For a rotor with positive (clockwise) angular speed, namely $j = \{1, 3\}$, considering its rotational frame, which is the same as the quadrotor's body-fixed one with angular rate $\vec{\Omega}$ relative to the inertial frame, the Coriollis effect appears and the gyroscopic (inertial) moment is modeled as

$${}^{b}_{G}\vec{M}_j = \left(\frac{d\vec{L}_j}{dt}\right)_{\text{rot}} + \vec{\Omega} \times \vec{L}_j = \vec{I}_j \cdot \dot{\vec{\omega}}_j + \vec{\Omega} \times \left(\vec{I}_j \cdot \vec{\omega}_j\right) \tag{2.9}$$

where $\vec{I}_j$ is the gyroscopic inertia, namely that of the rotating part of the rotor. By solving the cross-product it comes

$${}^{b}_{G}\vec{M}_j = \begin{bmatrix} I_{j_x}\, \dot{\omega}_{j_x} \\ I_{j_y}\, \dot{\omega}_{j_y} \\ I_{j_z}\, \dot{\omega}_{j_z} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{j_x}\, \omega_{j_x} \\ I_{j_y}\, \omega_{j_y} \\ I_{j_z}\, \omega_{j_z} \end{bmatrix} = \begin{bmatrix} I_{j_x}\, \dot{\omega}_{j_x} + I_{j_z}\, \omega_{j_z}\, q - I_{j_y}\, \omega_{j_y}\, r \\ I_{j_y}\, \dot{\omega}_{j_y} + I_{j_x}\, \omega_{j_x}\, r - I_{j_z}\, \omega_{j_z}\, p \\ I_{j_z}\, \dot{\omega}_{j_z} + I_{j_y}\, \omega_{j_y}\, p - I_{j_x}\, \omega_{j_x}\, q \end{bmatrix} \tag{2.10}$$

However, the direction of $\vec{\omega}_j$ coincides with the Z-axis of the body-fixed coordinate system whereas all its other components are zero, therefore $I_{j_x} = I_{j_y} = 0$ and, assuming that the gyroscopic inertia is the same for all rotors, $I_{j_z} = I_G$, (2.10) is simplified as

$$\begin{aligned} {}^{b}_{G}M_{j_x} &= I_G\,\omega_j\,q \\ {}^{b}_{G}M_{j_y} &= -I_G\,\omega_j\,p \\ {}^{b}_{G}M_{j_z} &= I_G\,\dot{\omega}_j \end{aligned} \tag{2.11}$$

2.2.2 Air Drag on Propeller

It is known that, as the blades of the propeller rotate in the air, they push it into a specific direction, in this case downwards, thus producing the thrust/lift force, however not without being affected by the reaction of the air flow onto them, what is here named as the *drag torque*. Once the propeller's axis of rotation is assumed perfectly aligned with the Z-axis of the body-fixed coordinate system, the drag torque does not affect the other axes. Assuming no torsion effect on the rotor, according to Pounds et al. [2004] the drag torque can be modeled, here again considering the clockwise rotors $j = \{1, 3\}$, as

$${}^{b}_{D}M_{j_z} = c_D\,\rho\,A\,R^2\,(\omega_j\,R)^2 = c_D\,\rho\,\pi\,R^5\,\omega_j^2 = k_D\,\omega_j^2 \tag{2.12}$$

where c_D is the non-dimensional drag torque coefficient, $\rho\,\left[\mathrm{kg/m^3}\right]$ is the air density, $A\,\left[\mathrm{m^2}\right]$ is the area of the propeller disc, $R[\mathrm{m}]$ is the propeller radius and $k_D\,\left[\mathrm{kg\,m^2}\right]$ is the resulting dimensional drag torque coefficient.

Now we can finally assemble the mechanical model of the rotor. Let us first consider its non-linear model. Also, for the sake of notation simplicity, let us drop for a moment the b frame-of-reference and the j rotor index of the moments and assume the latter as vectors having non-zero components only on the body-fixed Z-axis DOF, like ${}^{b}_{D}M_z = \vec{M}_D$, else stated. By assessing the sum of all torques acting on the rotor, namely the electromagnetic $M_E = k_t\,i_a$, the friction $M_B = B_a\,\omega$, the inertial M_G as in (2.11) and a generic load torque M_L, and then applying the Laplace transform, it comes

$$\sum M = I_G\,\dot{\omega} = M_E - M_B - M_L \quad \Rightarrow \quad \omega = \frac{k_t\,i_a - M_L}{I_G\,s + B_a} \tag{2.13}$$

In order to find the linear model, we consider the load torque $M_L = M_D$ as in (2.12), whose linearization around $\omega = \omega_0$ is $M'_D = k_D\,\omega_0^2 + 2\,k_D\,\omega_0\,\Delta\omega = M'_{D_0} + \Delta M'_D$, employ $\Delta M'_{D_0}$ along with $\Delta\omega = \omega - \omega_0$ and $\Delta i_a = i_a - i_{a_0}$ in (2.13), and once again applying the Laplace transform, it yields

$$\sum M = I_G\,\dot{\omega} = M_E - M_B - M'_D \quad \Rightarrow \quad \Delta\omega = \frac{k_t\,\Delta i_a}{I_G\,s + B_a + 2\,k_D\,\omega_0} \tag{2.14}$$

Applying (2.8) to (2.14) results in the linearized 2nd order rotor dynamics $u \to \omega$ described by

$$G_{R_f}(s) = \frac{\Delta\omega(s)}{\Delta u(s)} = \frac{\frac{k_t}{I_G\,L_a}}{s^2 + \frac{I_G\,R_a + (B_a + 2\,k_D\,\omega_0)L_a}{I_G\,L_a}\,s + \frac{(B_a + 2\,k_D\,\omega_0)R_a + k_v\,k_t}{I_G\,L_a}} \tag{2.15}$$

with steady-state ($t \to \infty \Rightarrow s \to 0$) gain

$$\gamma = \left.\frac{\Delta\omega(s)}{\Delta u(s)}\right|_{s \to 0} = \frac{k_t}{(B_a + 2\,k_D\,\omega_0)\,R_a + k_v\,k_t} \tag{2.16}$$

around the linearization point determined by

$$u_0 = \frac{k_D\,R_a}{k_t}\omega_0^2 + \frac{B_a\,R_a + k_v\,k_t}{k_t}\omega_0 \tag{2.17}$$

All torques acting on the rotor are transferred to the aircraft's airframe by means of the electromagnetic torque M_E generated by the BLDCM. Considering all body-fixed coordinate system's components, it is assessed as $\vec{M}_E = \vec{M}_G + \vec{M}_D + \vec{M}_B$, which yields, for rotors $j = \{1, 3\}$ and already with the corrected (inverted) sign for evaluation from the aiframe's perspective

$$\begin{aligned} {}^{b}_{E}M_{j_x} &= -I_G\,\omega_j\,q \\ {}^{b}_{E}M_{j_y} &= I_G\,\omega_j\,p \\ {}^{b}_{E}M_{j_z} &= -\left(I_G\,\dot{\omega}_j + k_D\,\omega_j^2 + B_a\,\omega_j\right) \end{aligned} \tag{2.18}$$

Regarding the other rotors, $j = \{2, 4\}$, the sign in (2.18) needs to be changed due to the inverse rotation direction.

2.2.3 Thrust

The thrust force is generated by the propeller rotation through the viscous air, is used for both lift and translational purposes and its direction is always aligned with the body-fixed Z-axis. Once again referring to the work of Pounds et al. [2004], the thrust force for a given rotor $j = 1 \ldots 4$ can be modeled directly in the body-fixed coordinate system as

$$T_j = {}^{b}_{T}F_z = -c_T\,\rho\,\pi\,R^4\,\omega_j^2 = -k_T\,\omega_j^2 \tag{2.19}$$

Defining l_a as the lever length of each of the quadrotor's arms, i.e. the distance taken in the XY body-fixed plane from the rotor's rotational axis to the aircraft's center of mass, and assuming l_a to be the same for all arms, the difference in thrust produced by the propellers in the X-axis defines a moment around the Y, and vice-versa

$$\begin{aligned} {}^{b}_{T}M_x &= l_a\,(T_4 - T_2) \\ {}^{b}_{T}M_y &= l_a\,(T_1 - T_3) \\ {}^{b}_{T}M_z &= 0 \end{aligned} \tag{2.20}$$

Moreover, considering all rotors, the total thrust force opposite to the aircraft's weight is

$${}^{b}_{T}F_z = \sum_{j=1}^{4} T_j = -k_T \sum_{j=1}^{4} \omega_j^2 \tag{2.21}$$

2.3 Earth's Gravity

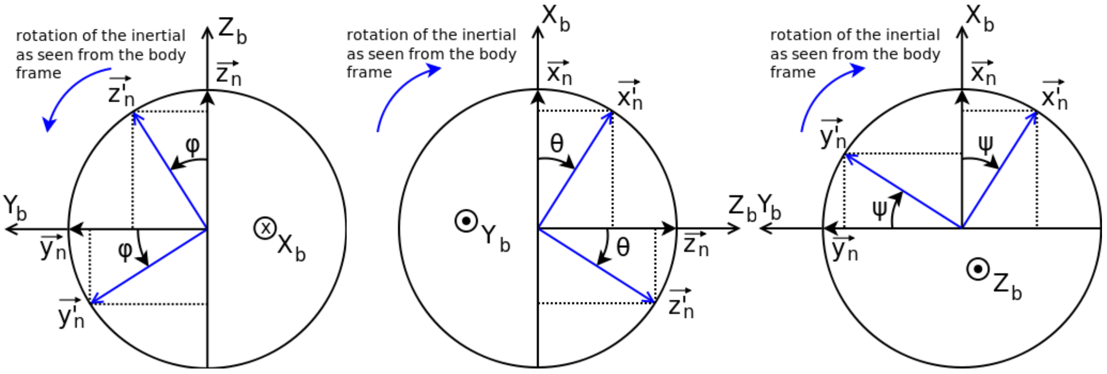
The interaction of the Earth's gravitational field and the quadrotor causes its weight force to act upon it on its center of mass. This is modeled in the inertial (navigation) frame simply by Newton's 2nd law as

$${}^{n}_{W}\vec{F} = \begin{bmatrix} 0 & 0 & m\,g \end{bmatrix}^T \tag{2.22}$$

where $g = 9.81\,\mathrm{m/s^2}$ is the absolute value of Earth's gravity acceleration. However, this force needs to be assessed in the body-fixed coordinate system, therefore the need for a coordinate transformation appears. For the sake of completeness, this will be briefly addressed here, following Stevens and Lewis [2003].

2.3.1 Coordinate System Transformation

Let us consider three basic rotations around each one of the body-fixed coordinate axes, as shown in Fig. 2.3.



(a) X-axis. (b) Y-axis. (c) Z-axis.

Figure 2.3: Basic rotations in the body-fixed frame.

Initially, both coordinate systems are exactly aligned and then the body-fixed is rotated with respect to the inertial according to the right-hand rule. However, from the body-fixed point of view, the inertial frame rotates on the other direction (in blue), which shall be here considered for a transformation from inertial to body-fixed coordinates. Therefore, the vectors $\vec{X}$, $\vec{Y}$ and $\vec{Z}$ with coordinates ${}^{b}\begin{bmatrix} x & y & z \end{bmatrix}^T$ in the body-fixed frame have their coordinates changed according to the respective single rotation matrices

$$
\begin{gathered}
{}^{b}_{n}R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \qquad {}^{b}_{n}R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\
{}^{b}_{n}R_Z(\psi) = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
\end{gathered} \tag{2.23}
$$

In other words, the rotation matrices map the coordinates of the navigation (inertial) frame into the body-fixed (rotating) one. It is known that any point in the *Euclidian* 3D space can be represented by a sequence of these three basic rotations around the Euler angles, and the exact same sequence needs to be applied in order to correctly obtain the 3D rotation. The sequence ${}^{b}_{n}R_Z(\psi) \to {}^{b}_{n}R_Y(\theta) \to {}^{b}_{n}R_X(\phi)$ is chosen, corresponding to the matrix multiplication[2] ${}^{b}_{n}R = {}^{b}_{n}R_X(\phi) \cdot {}^{b}_{n}R_Y(\theta) \cdot {}^{b}_{n}R_Z(\psi)$. The result is

$$
{}^{b}_{n}R = \begin{bmatrix} \cos\theta\,\cos\psi & \cos\theta\,\sin\psi & -\sin\theta \\ -\cos\theta\,\sin\psi{+}\sin\phi\,\sin\theta\,\cos\psi & \cos\phi\,\cos\psi{+}\sin\phi\,\sin\theta\,\sin\psi & \sin\phi\,\cos\theta \\ \sin\phi\,\sin\psi{+}\cos\phi\,\sin\theta\,\cos\psi & -\sin\phi\,\cos\psi{+}\cos\phi\,\sin\theta\,\sin\psi & \cos\phi\,\cos\theta \end{bmatrix} \tag{2.24}
$$

It can be proven that for the inverse mapping, i.e. from the body-fixed to the navigation coordinate system, it holds ${}^{n}_{b}R = {}^{b}_{n}R^T$.

Now, the gravity's force upon the quadrotor can be obtained as

$$
{}^{b}_{W}\vec{F} = {}^{b}_{n}R \cdot {}^{n}_{W}\vec{F} = \begin{bmatrix} -m\,g\,\sin\theta \\ m\,g\,\sin\phi\,\cos\theta \\ m\,g\,\cos\phi\,\cos\theta \end{bmatrix} \tag{2.25}
$$

[2] Note the multiplication order from the right to the left.

Assuming that both the quadrotor's mass and gravitational center coincide, no moment is generated by its weight force.

2.4 Non-Linear Model

Being already assessed the forces and moments acting upon the quadrotor, its non-linear model can now be assembled. However, given the assumptions, so far, of the torque and thrust behaviors, which have not yet been confirmed by identification experiments, we shall leave the inputs of the model expressed in terms of the thrust T_j and torque M_j generated by each rotor $j = 1 \dots 4$. Considering first the moments, we take (2.18), with inverted sign in order to have it as perceived by the quadrotor's airframe, and (2.20), and insert them into the left-hand side of (2.4), while excluding the second term on the right-hand side due to the assumption $I_x \approx I_y$. After rearranging the terms and isolating the angular rates, the *moment equations* are obtained

$$\begin{cases} \dot{p} = \dfrac{l_a}{I_x}\,(T_4 - T_2) + \dfrac{1}{I_x}\displaystyle\sum_{j=1}^{4} M_{j_x} + \dfrac{(I_y - I_z)}{I_x}\,q\,r \\[2ex] \dot{q} = \dfrac{l_a}{I_y}\,(T_1 - T_3) + \dfrac{1}{I_y}\displaystyle\sum_{j=1}^{4} M_{j_y} - \dfrac{(I_x - I_z)}{I_y}\,p\,r \\[2ex] \dot{r} = \dfrac{1}{I_z}\displaystyle\sum_{j=1}^{4} M_{j_z} \end{cases} \tag{2.26}$$

whereas by inserting (2.21) and (2.25) into the left side of (2.7) and isolating the translational accelerations, the complete *force equations* arise as

$$\begin{cases} \dot{u} = v\,r - w\,q - g\,\sin\theta \\ \dot{v} = w\,p - u\,r + g\,\sin\phi\,\cos\theta \\ \dot{w} = u\,q - v\,p + g\,\cos\phi\,\cos\theta + \dfrac{1}{m}\displaystyle\sum_{j=1}^{4} T_j \end{cases} \tag{2.27}$$

This model corroborates the intuitive idea we had about the quadrotor's dynamics in the first place, as illustrated in Fig. 2.4. Given T_j and M_j produced by each rotor, first the aircraft attitude is changed according to (2.26), what determines the inner inputs to the position/altitude dynamics block, corresponding to (2.27). The altitude sub-block, however, can be directly affected by T_j. This brief analysis will be important for, later on, defining a control architecture and criteria to be used.

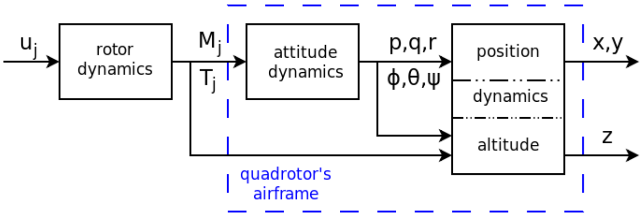


Figure 2.4: Simplified block diagram of the quadrotor's dynamics.

We shall see in chapter 3, as the system's parameters will be identified/dimensioned, that the rotor dynamics is much faster than the airframe attitude one, which in turn is faster than the airframe position.

Chapter 3

Model Identification

We shall start by identifying the airframe rigid-body parameters and then move on to the rotors.

3.1 Airframe

Using a digital scale the aircraft was weighted, resulting in a total mass $m = 0.694\,\mathrm{kg}$. The weight of each single rotor read $m_r = 0.075\,\mathrm{kg}$. For the sake of simplifying the identification process and yet seeking a good approximation of the parameters, let us regard the quadrotor's inertial structure as two perpendicular rods, corresponding to the arms and X,Y axes of the body-fixed coordinate system, with one point-mass on each edge, representing the rotor mass, as depicted in Fig. 3.1.

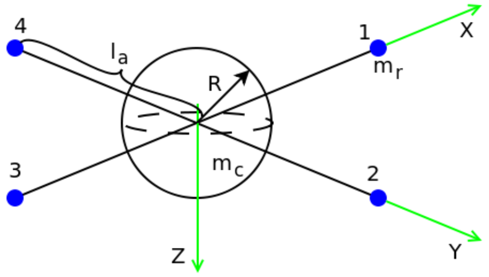


Figure 3.1: Quadrotor's airframe and inertial identification scheme.

The arm length measured is $l_a = 0.18\,\mathrm{m}$. All the aircraft's mass excluding the one of the rotors is assumed to be homogeneously distributed inside the sphere of radius $R = 8\,\mathrm{cm}$, centered in the origin of the axes. Knowing that the moment of inertia of a solid sphere around an axis σ is given by ${}^{s}I_{\sigma} = \frac{2}{5} m_s R^2$ whereas for a point-mass distant l_a from the rotation axis is given by ${}^{r}I_{\sigma} = m_r\, l_a^2$, and having the sphere mass as $m_s = m - 4m_r = 0.394\,\mathrm{kg}$, the moment of inertia around axes X and Y, due to their symmetry, is easily calculated as

$$\begin{aligned} I_x = I_y = {}^{2}I_x + {}^{4}I_x + {}^{s}I_x = {}^{1}I_y + {}^{3}I_y + {}^{s}I_y = 2\left(m_r\, l_a^2\right) + \frac{2}{5} m_s\, R^2 \\ = 5.86864 \cdot 10^{-3}\,\mathrm{kg\,m^2} \end{aligned} \tag{3.1}$$

For the Z-axis the four rotors need to be considered, thus

$$I_z = \sum_{j=1}^{4} {}^{j}I_z + {}^{s}I_z = \sum_{j=1}^{4} \left(m_r\, l_a^2\right) + \frac{2}{5} m_s\, R^2 = 10.72864 \cdot 10^{-3}\,\mathrm{kg\,m^2} \tag{3.2}$$

Table 3.1 summarizes the identified parameters for the quadrotor's airframe, needed later on for simulation and control design purposes.

Symbol	Value	Description
m	$0.694\,\mathrm{kg}$	total aircraft mass
l_a	$0.18\,\mathrm{m}$	rotor lever length to aircraft's CG
$I_x = I_y$	$5.87 \cdot 10^{-3}\,\mathrm{kg\,m^2}$	aircraft's moment of inertia around X and Y axes
I_z	$10.73 \cdot 10^{-3}\,\mathrm{kg\,m^2}$	aircraft's moment of inertia around Z axis

Table 3.1: Identified quadrotor airframe parameters.

3.2 Rotors

The approach employed towards identification of the rotor's parameters was, first, to assume the nominal characteristics from the manufacturer, and then proceed to identify the remaining parameters, considering the value of the ones already available.

3.2.1 Internal Dynamics

The PJS 3D 550E motor data sheet unfortunately did not provide substantial information on the BLDCM dynamics, apart from an armature resistance of $R_a = 250\,\mathrm{m\Omega}$ and a nominal steady-state gain of $k_\infty = \frac{760\,\mathrm{rpm}}{1\,\mathrm{V}} \approx 79.59/(\mathrm{V\,s})$. It also informed that the maximum armature current is $i_{a\max} = 8.5\,\mathrm{A}$, what shall be considered during the control loop design as a control action constraint. However, most of the parameters still had to be determined via identification tests.

First, a steady-state torque experiment ($\omega \to M$) was executed, whose configuration is shown in Fig. 3.2.

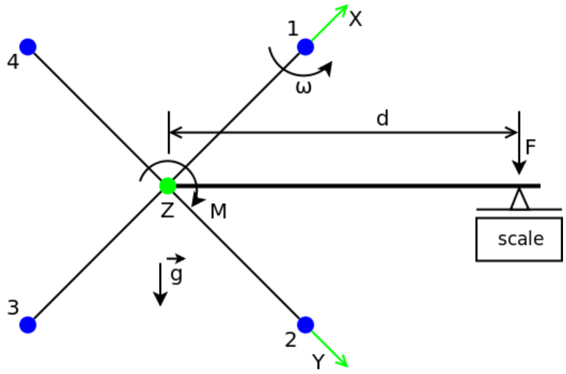


Figure 3.2: Setup for rotor torque identification experiment.

The quadrotor was placed with its Z-axis perpendicular to the gravity vector $\vec{g}$ (i.e. parallel to the ground), whereas the DOFs around the other axes were blocked. A rigid lever with length d was attached to the aircraft's center of mass and its other end applied to the scale, so that the torque M caused on the quadrotor's airframe due to the rotation of a rotor could be assessed as

$$M = \left(\frac{m - m_0}{1000}\right) \mid \vec{g} \mid \; d = I_G\,\dot{\omega} + k_D\,\omega^2 + B_a\,\omega \quad [\mathrm{N\,m}] \tag{3.3}$$

where m [g] was the mass reading of the scale, m_0 [g] was the offset reading at $\omega = 0\,\mathrm{rad/s}$ and $|\ \vec{g}\ | = 9.81\,\mathrm{m/s^2}$ is the gravity's acceleration. The sampled and interpolated curves are shown in Fig. 3.3. It can be seen that the sampled data adheres very well to a parabolic function, as theoretically expected. From the interpolation, it was immediately obtained $k_D = 1.18 \cdot 10^{-7}\,\mathrm{N\,m\,s^2}$ and $B_a = 1.23 \cdot 10^{-6}\,\mathrm{N\,m\,s}$.

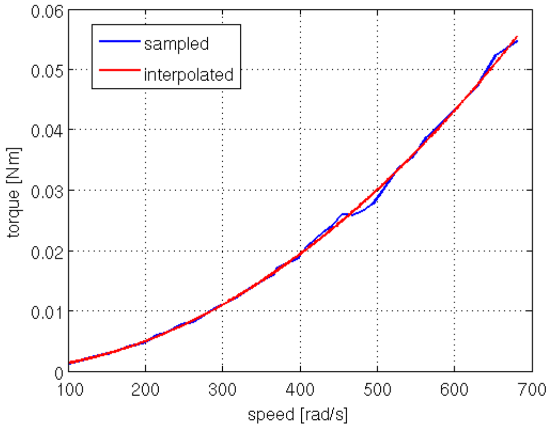
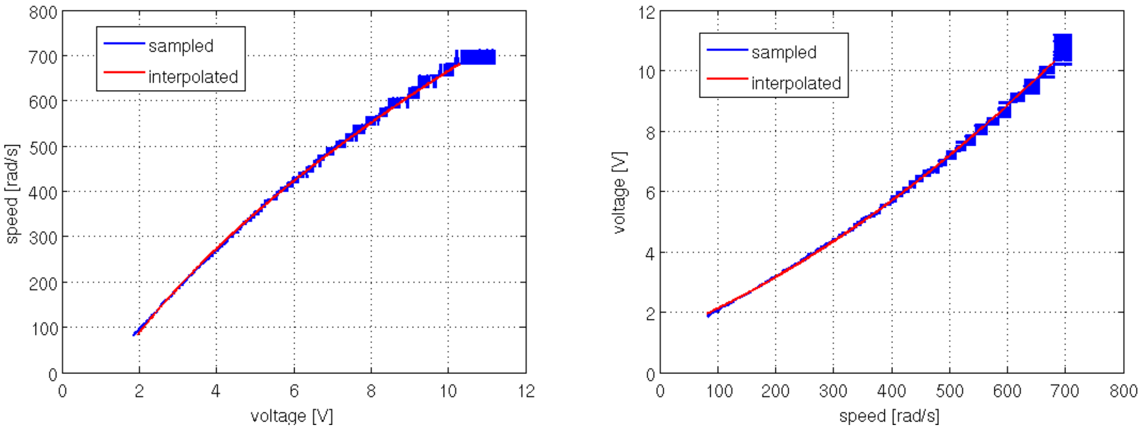


Figure 3.3: Rotor torque identification result.

Following, a steady-state speed ($u \to \omega$) experiment with propeller was undertaken. In Fig. 3.4a it can be seen the expected effect of the drag torque damping, reducing the gain as the operational point (u, ω) increases. It is also noticeable some minimum voltage $u_{\min} > 0\,\mathrm{V}$, below which the rotor stops rotating, possibly due to some internal static friction source or even to the own BLDCM electric driver.



(a) Procedural $u \to \omega$.

(b) Artificial $\omega \to u$ for interpolation.

Figure 3.4: Rotor steady-state experiment in channel $u \to \omega$.

By comparing the rotor steady-state equation as in (2.16) to the interpolated curve as in Fig. 3.4b, and since R_a and k_D were already known, it was obtained $u_{\min} = 1.25\,\mathrm{V}$, $k_t = 3.7 \cdot 10^{-3}\,\mathrm{N\,m/A}$ and $k_v = 7.8 \cdot 10^{-3}\,\mathrm{V\,s}$.

The third test performed was a step input response, without propeller and with an inertia ring attached to the motor shaft, in order to add some known and substantial inertia and thus make the dynamics slower and easier to be detected. The experiment was executed with a sampling rate $f_s = 100\,\mathrm{Hz}$ and its result is shown in Fig. 3.5. Due to the yet fast dynamics of the rotor, a higher sampling rate should have been used in order to yield more accurate data to proceed with the identification. However, due to technical difficulties, that was not possible. Therefore, the last two parameters regarding the rotor dynamics to be identified, namely the armature inductance L_a and the rotor inertia $I_G = I_m + I_p$, are not expected to be identified accurately enough hereby.

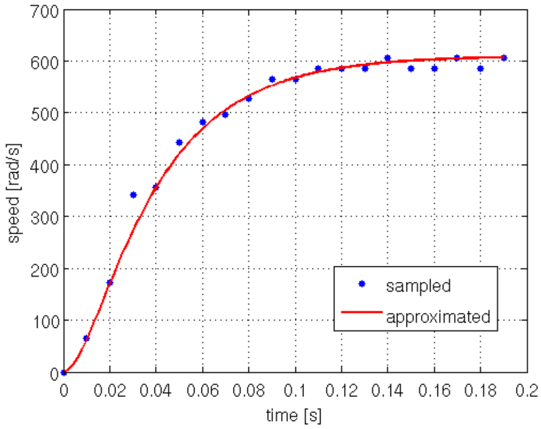


Figure 3.5: Step input response $u \to \omega$ experiment.

Anyhow, by comparing the approximated dynamics from the samples with the theoretical model of the BLDC motor

$$\frac{\omega(s)}{u(s)} = \frac{\frac{k_t}{I_t\,L_a}}{s^2 + \frac{I_t\,R_a + B_a\,L_a}{I_t\,L_a}\,s + \frac{B_a\,R_a + k_v\,k_t}{I_t\,L_a}} = \frac{264360}{s^2 + 133,8\,s + 3115} \tag{3.4}$$

yielded $L_a = 1.9\,\mathrm{mH}$ and $I_t = I_m + I_r = 7.53 \cdot 10^{-6}\,\mathrm{kg\,m^2}$. Knowing the ring inertia $I_r \approx 7 \cdot 10^{-6}\,\mathrm{kg\,m^2}$, the motor inertia is found to be indeed very small, $I_m \approx 5.3 \cdot 10^{-7}\,\mathrm{kg\,m^2}$. Nevertheless, since the propeller inertia I_p was still not known, it was anyway not possible to assess I_G.

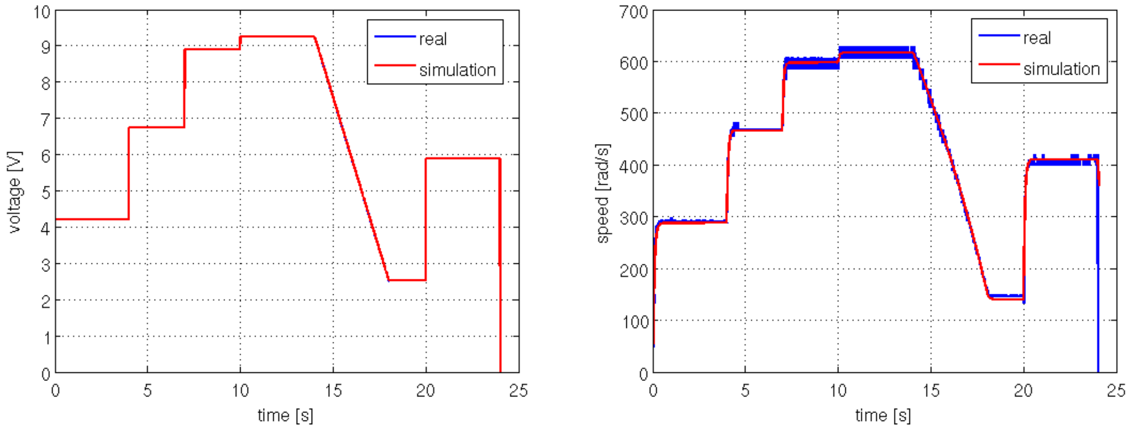
A fourth experiment, consisting of a sequence of voltage steps applied to the rotor, enabled I_G to be detected and fine-tuned until both the model and the real-system response were made to match as close as possible. The result, presented in Fig. 3.6, is quite satisfactory, and served as a validation test to the identification done so far.

In order to achieve this optimal matching between real and simulated data, not only the rotor inertia was identified as $I_G = 1,5 \cdot 10^{-5}\,\mathrm{kg\,m^2}$ but also a small steady-state gain adjustment was made by slightly increasing the armature resistance to $R_a = 260\,\mathrm{m\Omega}$.

3.2.2 Thrust

In order to identify the dependence of thrust T on the rotor speed ω, another steady-state experiment was performed, this time by applying u and observing T, measured on a digital scale. The result can be seen in Fig. 3.7a. The thrust curve $T(u)$ can be very well interpolated by a parabolic function, however with a thrust offset T_{off} at the zero-speed voltage $u_{\min}$. This was due to the dead weight of the rotor itself on the scale, which should not be taken into consideration, therefore yielding the corrected curve in green.

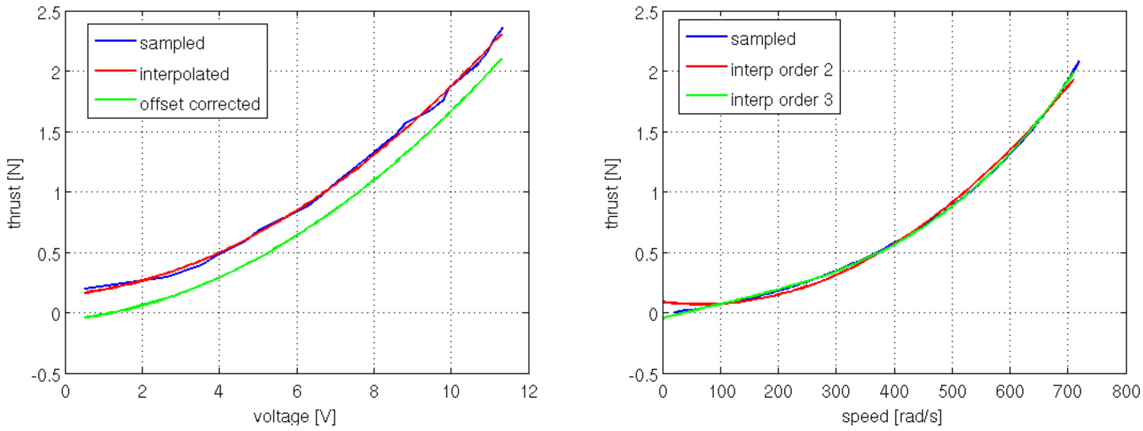
Finally, the function from ω to T was calculated as $T(\omega) = T\left(\omega(u)\right)$, namely by picking values of (u, ω) in function of Fig. 3.4a, finding the respective (u, T) in the offset corrected curve of Fig. 3.7a and then finally assembling the intermediate function (ω, T) as in Fig.



(a) Input $u(t)$.

(b) Output $\omega(t)$.

Figure 3.6: Validation test for rotor dynamics identification.



(a) Steady-state dependence $u \to T$.

(b) Calculated dependence $\omega \to T$.

Figure 3.7: Results of thrust identification experiment.

3.7b. It can be seen, though, that $T(\omega)$ does not adhere sufficiently accurately to a parabolic function, let alone a pure parabolic one as theoretically foreseen by (2.19), i.e. without the lower order coefficients. A cubic function,

$$\begin{aligned} T(\omega) &= \sum_{i=0}^{3} \gamma_i \, \omega^i \\ T(\omega) &= -47.7 \cdot 10^{-3} + 1.3 \cdot 10^{-3} \omega - 1.44 \cdot 10^{-6} \omega^2 + 5.19 \cdot 10^{-9} \omega^3 \quad [\mathrm{N}] \end{aligned} \tag{3.5}$$

matches much better the experimentally observed thrust behavior, and shall therefore be used henceforth.

Table 3.2 encloses all other rotor parameters hereby identified.

Once the thrust function is identified, we can calculate the hovering point by finding the roots of $T(u)$ in Fig. 3.7a for $T = \frac{m\,g}{4}$. Disregarding the negative root, which is of no physical meaning, it yields $(u, \omega)_{\text{hov}} \approx (10.1\,\mathrm{V}; 668\,\mathrm{rad/s})$.

Symbol	Value	Description
k_D	$1.18 \cdot 10^{-7}\,\mathrm{N\,m\,s^2}$	air drag torque coefficient
B_a	$1.23 \cdot 10^{-6}\,\mathrm{N\,m\,s}$	linear friction torque coefficient
k_t	$3.7 \cdot 10^{-3}\,\mathrm{N\,m/A}$	electric torque constant
k_v	$7.8 \cdot 10^{-3}\,\mathrm{V\,s}$	speed constant
L_a	$1.9 \cdot 10^{-3}\,\mathrm{H}$	armature impedance
I_G	$1.5 \cdot 10^{-5}\,\mathrm{kg\,m^2}$	rotor inertia
R_a	$260 \cdot 10^{-3}\,\Omega$	armature resistance

Table 3.2: Identified rotor parameters.

Having all the parameters of the non-linear rotor model properly identified, a Simulink block diagram was built, as shown in Fig. 3.8. This model not only outputs the speed $\omega(t)$, but also the angular momentum $L(t)$, the total (electric) torque $M(t)$ and the generated thrust $T(t)$, which shall be then fed to the quadrotor's non-linear model in (2.26) and (2.27).

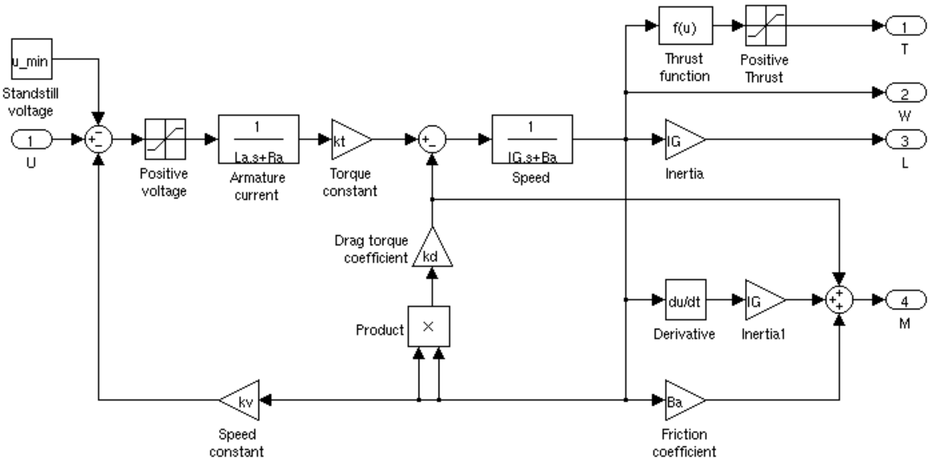


Figure 3.8: Simulink model for non-linear rotor simulation.

3.2.3 Model Simplification

From the rotor identification done so far, the linearized 2nd-order transfer function model obtained, according to (2.15) is

$$G_{R_f}(s) = \frac{3.724 \cdot 10^{-3}}{2.805 \cdot 10^{-8} s^2 + 4.197 \cdot 10^{-6} s + 7.036 \cdot 10^{-5}} \tag{3.6}$$

with poles $\lambda_1 \approx -130$ and $\lambda_2 \approx -19$. The first pole is very fast in comparison to the other one, and thus can be discarded, although being kept its steady-state gain, for the sake of model simplification, yet without hindering its validity. The simplified model is hence

$$G_{R_s}(s) = \frac{K_s}{s + \lambda_s} = \frac{885,6}{s + 16.7} \tag{3.7}$$

As can be seen in the step response plots in Fig. 3.9, the simplification with the 1st-order system still represents sufficiently well the rotor dynamics, and thus shall be used henceforth for the control system design to come.

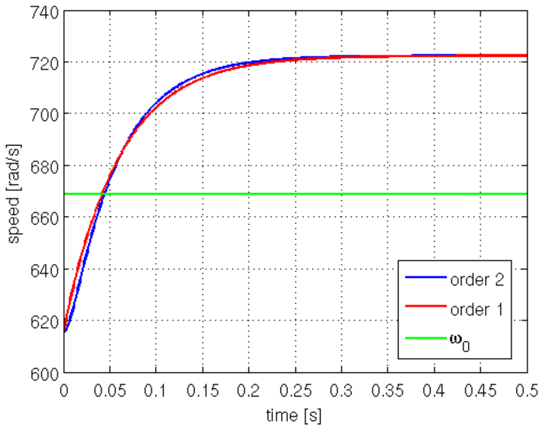


Figure 3.9: Time responses of full 2nd and simplified 1st-order rotor models.

3.3 Identified Quadrotor Dynamics

Once the airframe and rotor parameters have been identified, we can proceed to detail the intermediate non-linear model presented in section 2.4 in terms of the rotor speeds ω_j, and derive a linearized model for enabling the design of linear controllers later in chapter 6.

3.3.1 Non-linear Model

As verified during the identification tests, the thrust behavior cannot satisfactorily be represented by a pure parabolic function as in (2.19), but rather as the complete cubic function in (3.5). Employing this together with the rotor moment in (2.18) yields the *moment equations*

$$\begin{cases} \dot{p} = \dfrac{l_a}{I_x} \displaystyle\sum_{i=0}^{3} \gamma_i \left(\omega_2^i - \omega_4^i\right) + \dfrac{I_G}{I_x} q \displaystyle\sum_{j=1}^{4} \omega_j (-1)^j + \dfrac{(I_y - I_z)}{I_x} q\, r \\ \dot{q} = \dfrac{l_a}{I_y} \displaystyle\sum_{i=0}^{3} \gamma_i \left(\omega_3^i - \omega_1^i\right) - \dfrac{I_G}{I_y} p \displaystyle\sum_{j=1}^{4} \omega_j (-1)^j - \dfrac{(I_x - I_z)}{I_y} p\, r \\ \dot{r} = \dfrac{1}{I_z} \displaystyle\sum_{j=1}^{4} \left(I_G\, \dot{\omega}_j + k_D\, \omega_j^2 + B_a\, \omega_j\right) (-1)^j \end{cases} \tag{3.8}$$

and the *force equations*

$$\begin{cases} \dot{u} = v\, r - w\, q - g\, \sin\theta \\ \dot{v} = w\, p - u\, r + g\, \sin\phi\, \cos\theta \\ \dot{w} = u\, q - v\, p + g\, \cos\phi\, \cos\theta - \dfrac{1}{m} \displaystyle\sum_{j=1}^{4} \sum_{i=0}^{3} \gamma_i\, \omega_j^i \end{cases} \tag{3.9}$$

3.3.2 Linearized Model

By considering the derivative of the thrust produced by each rotor at its operational speed ω_{j_0} as $\dot{T}_{j_0} = \gamma_1 + 2\,\gamma_2\,\omega_{j_0} + 3\,\gamma_3\,\omega_{j_0}^2$, the correspondent airframe[1] linearized model at the operational point (p_0, q_0, r_0) is represented by the *linearized moment equations*

[1] Disregarding the rotors.

$$\begin{cases} \Delta\dot{p} = \dfrac{l_a}{I_x}\left(\dot{T}_{2_0}\Delta\omega_2 - \dot{T}_{4_0}\Delta\omega_4\right) + \dfrac{I_G}{I_x}\left[q_0 \displaystyle\sum_{j=1}^{4}\Delta\omega_j(-1)^j + \Delta q\sum_{j=1}^{4}\omega_{j_0}(-1)^j\right] + \dfrac{I_y - I_z}{I_x}\left(r_0\,\Delta q + q_0\,\Delta r\right) \\ \Delta\dot{q} = \dfrac{l_a}{I_y}\left(\dot{T}_{3_0}\Delta\omega_3 - \dot{T}_{1_0}\Delta\omega_1\right) - \dfrac{I_G}{I_y}\left[p_0 \displaystyle\sum_{j=1}^{4}\Delta\omega_j(-1)^j + \Delta p\sum_{j=1}^{4}\omega_{j_0}(-1)^j\right] - \dfrac{I_x - I_z}{I_y}\left(r_0\,\Delta p + p_0\,\Delta r\right) \\ \Delta\dot{r} = \dfrac{1}{I_z}\displaystyle\sum_{j=1}^{4}\left[I_G\,\Delta\dot{\omega}_j + (2\,k_D\,\omega_{j_0} + B_a)\Delta\omega_j\right](-1)^j \end{cases} \tag{3.10}$$

Having also at the operational point (u_0, v_0, w_0) and (ϕ_0, θ_0), the *linearized force equations* appear as

$$\begin{cases} \Delta\dot{u} = v_0\,\Delta r + r_0\,\Delta v - w_0\,\Delta q - q_0\,\Delta w - g\,\cos\theta_0\,\Delta\theta \\ \Delta\dot{v} = w_0\,\Delta p + p_0\,\Delta w - u_0\,\Delta r - r_0\,\Delta u + g\,\cos\phi_0\,\cos\theta_0\,\Delta\phi - g\,\sin\phi_0\,\sin\theta_0\,\Delta\theta \\ \Delta\dot{w} = u_0\,\Delta q + q_0\,\Delta u - v_0\,\Delta p - p_0\,\Delta v - g\,\sin\phi_0\,\cos\theta_0\,\Delta\phi - g\,\cos\phi_0\,\sin\theta_0\,\Delta\theta - \dfrac{1}{m}\displaystyle\sum_{j=1}^{4}\dot{T}_{j_0}\Delta\omega_j \end{cases} \tag{3.11}$$

The full linearized aircraft dynamics can be obtained by simply substituting the rotor dynamics $\Delta u_j \to \Delta\omega_j$ (3.7) into (3.10) and (3.11).

Chapter 4

Attitude Estimation

The aircraft's attitude with respect to the inertial frame can be represented by the Euler angles $\phi(t)$, $\theta(t)$ and $\psi(t)$[1]. For its assessment, gyroscopes in a MEMS sensory unit provide readings of p, q and r in the body-fixed frame. These angular rates cannot be simply integrated in order to obtain the Euler angles, since the latter are defined in the inertial (navigation) frame. Therefore a function $J : \{p, q, r, \phi, \theta, \psi\} \rightarrow \left\{\dot{\phi}, \dot{\theta}, \dot{\psi}\right\}$ is needed.

4.1 Euler Kinematic Equations

The classical and rigorous full derivation of such a mapping can be found in the literature, as in Stevens and Lewis [2003], but in short words, similarly as for the Euler angles defined previously in Fig. 2.3, we can find J^{-1} by solving the Euler angular rates into the body-fixed frame by applying the same sequence of rotations defined in section 2.3.1, therefore

$$
{}^{b}\begin{bmatrix} p \\ q \\ r \end{bmatrix} = {}^{b}_{n}R_1 \; {}^{n}\begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + {}^{b}_{n}R_2 \; {}^{n}\begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + {}^{b}_{n}R_3 \; {}^{n}\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = J^{-1} \; {}^{n}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{4.1}
$$

where, given that $\dot{\phi}$ corresponds already to the last rotation, no other needs to be done and so ${}^{b}_{n}R_1 = 1$. $\dot{\theta}$ is the second and thus still needs to be rotated around X, hence ${}^{b}_{n}R_2 = {}^{b}_{n}R_X(\phi)$, whereas $\dot{\psi}$ is the very first rotation and consequently ${}^{b}_{n}R_3 = {}^{b}_{n}R_X(\phi) \cdot {}^{b}_{n}R_Y(\theta)$ still need to be applied. Finally, by finding J^{-1} and taking its inverse, the transformation from body-fixed to inertial (Euler) angular rates is achieved by

$$
{}^{n}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \, \tan\theta & \cos\phi \, \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix} {}^{b}\begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.2}
$$

which are known as the *Euler kinematic equations.*

4.2 Rotation Matrix

Although relatively simple, the classical method for attitude evaluation described by (4.2) is subject to the known problem of singularity when $\theta \rightarrow 90^o$. Even if we assume that such conditions are unlikely to happen during operation of the quadrotor, which is quite reasonable for non-acrobatic applications of the quadrotor, still the computational effort for working with trigonometric functions is relatively high for the embedded microprocessor onboard the aircraft. Moreover, eventual and unavoidable approximation errors in the calculation of

[1] For the sake of notation simplification we shall leave their domain (t) implicit.

such functions would propagate in time through the integration of the Euler angular rates, yielding a drift (random walk) in the true values of the Euler angles. Therefore the need of an alternative, simpler evaluation method arises. The complete and rigorous derivation of such can be found in Stepaniak [2008], yet, for the sake of clarity, it will be briefly addressed here as well.

The idea is to extract the Euler angles from the total rotation matrix ${}^n_bR$ while propagating it through time. Again, ${}^n_bR = {}^b_nR^T$, which was presented in (2.24), so the Euler angles can be easily calculated as

$$\phi = \arctan\left(\frac{{}^n_bR(3,2)}{{}^n_bR(3,3)}\right) \quad \theta = -\arcsin\left({}^n_bR(3,1)\right) \quad \psi = \arctan\left(\frac{{}^n_bR(2,1)}{{}^n_bR(1,1)}\right) \tag{4.3}$$

As said, this requires the propagation of the rotation matrix through time, therefore we shall look for an expression of its time derivative. For a time interval Δt small enough to be shorter than the fastest airframe attitude dynamics, we define

$${}^n_b\dot{R} = \lim_{\Delta t \to 0} \frac{\Delta {}^n_bR(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{{}^n_bR\left(t+\Delta t\right) - {}^n_bR(t)}{\Delta t} \tag{4.4}$$

where the rotation decomposition

$${}^n_bR\left(t+\Delta t\right) = {}^n_nR^{(t+\Delta t)}_{(t)} \cdot {}^n_bR^{(t)}_{(t)} \cdot {}^b_bR^{(t)}_{(t+\Delta t)} \tag{4.5}$$

applies. Since, by definition, there is no rotation in the inertial frame, ${}^n_nR^{(t+\Delta t)}_{(t)} = I$, where I is the identity matrix. Now, in the body frame, a rotation occured between the time instants t and $t+\Delta t$ is solely due to Δt, as if the previous body frame had become the navigation frame in $t = 0$. Moreover, a rotation backwards in time is calculated by the transposed matrix of that in forward time, hence

$${}^b_bR^{(t)}_{(t+\Delta t)} = {}^b_bR^{(t+\Delta t)^T}_{(t)} = {}^b_nR^T\left(\Delta t\right) = {}^n_bR\left(\Delta t\right) =$$
$$\begin{bmatrix} \cos\Delta\theta\cos\Delta\psi & -\cos\Delta\theta\sin\Delta\psi+\sin\Delta\phi\sin\Delta\theta\cos\Delta\psi & \sin\Delta\phi\sin\Delta\psi+\cos\Delta\phi\sin\Delta\theta\cos\Delta\psi \\ \cos\Delta\theta\sin\Delta\psi & \cos\Delta\phi\cos\Delta\psi+\sin\Delta\phi\sin\Delta\theta\sin\Delta\psi & -\sin\Delta\phi\cos\Delta\psi+\cos\Delta\phi\sin\Delta\theta\sin\Delta\psi \\ -\sin\Delta\theta & \sin\Delta\phi\cos\Delta\theta & \cos\Delta\phi\cos\Delta\theta \end{bmatrix} \tag{4.6}$$

Assuming that in such short Δt only small angle variations $\Delta\alpha : \{\Delta\phi, \Delta\theta, \Delta\psi\}$ are generated, it can be simplified $\Delta\alpha \to 0 \Rightarrow \sin\Delta\alpha \approx \Delta\alpha$ and $\cos\Delta\alpha \approx 1$, therefore

$${}^b_bR^{(t)}_{(t+\Delta t)} = \begin{bmatrix} 1 & -\Delta\psi & \Delta\theta \\ \Delta\psi & 1 & -\Delta\phi \\ -\Delta\theta & \Delta\phi & 1 \end{bmatrix} \tag{4.7}$$

Defining $\Delta\Theta = \begin{bmatrix} \Delta\phi & \Delta\theta & \Delta\psi \end{bmatrix}^T$, (4.7) can be rewritten as

$${}^b_bR^{(t)}_{(t+\Delta t)} = \begin{bmatrix} 0 & -\Delta\psi & \Delta\theta \\ \Delta\psi & 0 & -\Delta\phi \\ -\Delta\theta & \Delta\phi & 0 \end{bmatrix} + I = \Delta\Theta_\times + I \tag{4.8}$$

where $\Delta\Theta_\times$ is the skew-symmetric cross-product representation of $\Delta\Theta$. Inserting this into (4.5) and the result into (4.4) yields

$$
\begin{aligned}
{}^{n}_{b}\dot{R} &= \lim_{\Delta t \to 0} \frac{{}^{n}_{b}R(t)\left(\Delta\Theta_{\times} + I\right) - {}^{n}_{b}R(t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{{}^{n}_{b}R(t) \cdot \Delta\Theta_{\times}}{\Delta t} = {}^{n}_{b}R(t) \lim_{\Delta t \to 0} \frac{\Delta\Theta_{\times}}{\Delta t} \\
&= {}^{n}_{b}R(t) \begin{bmatrix} 0 & -\lim_{\Delta t \to 0} \frac{\Delta\psi}{\Delta t} & \lim_{\Delta t \to 0} \frac{\Delta\theta}{\Delta t} \\ \lim_{\Delta t \to 0} \frac{\Delta\psi}{\Delta t} & 0 & -\lim_{\Delta t \to 0} \frac{\Delta\phi}{\Delta t} \\ -\lim_{\Delta t \to 0} \frac{\Delta\theta}{\Delta t} & \lim_{\Delta t \to 0} \frac{\Delta\phi}{\Delta t} & 0 \end{bmatrix} \\
&= {}^{n}_{b}R(t) \begin{bmatrix} 0 & -r & q \\ r & 0 & p \\ -q & p & 0 \end{bmatrix} = {}^{n}_{b}R(t) \cdot \Omega_{\times}
\end{aligned} \tag{4.9}
$$

where p, q, r are the body-fixed angular rates read from the MEMS unit. Discretizing (4.9) produces

$$
{}^{n}_{b}R\left(k+1\right) = {}^{n}_{b}R(k) \cdot e^{\int_{0}^{T} \Omega_{\times} dt} = {}^{n}_{b}R(k) \cdot e^{\Omega_{\times}\,\Delta t} = {}^{n}_{b}R(k) \cdot e^{\Theta_{\times}} \tag{4.10}
$$

Once again assuming a Δt much smaller[2] than the fastest airframe attitude dynamics, we can approximate $e^{\Theta_{\times}} = I + \Theta_{\times}$ and finally the discrete equation of the rotation matrix propagation through time is obtained as

$$
{}^{n}_{b}R\left(k+1\right) = {}^{n}_{b}R(k)\left(I + \Theta_{\times}\right) \tag{4.11}
$$

[2] This assumption becomes valid by using sufficiently high sampling rates from the sensors.

Chapter 5

Noise Filtering and State Estimation

In chapter 6, all the control techniques discussed assume that all state variables, or at least a significant number of them, are available through sensor readings and, moreover, such measurements are not affected by any noise. When it comes to the implementation on the physical hardware, such assumptions are not true since usually it is impossible or even not feasible to measure certain variables of the state vector whereas, for those measured, along with the pure signal comes noise originated from the sensor - *output noise*, and possibly even noise which affects directly the system dynamics, so called *process noise*, usually as a result of stochastic disturbances. Hence, in order to achieve an enhanced control performance, such implementation issues need to be dealt with accordingly. One way to do that is by using a *Kalman filter*.

5.1 Kalman Filter Design

Extensive literature on this filtering technique is available. A detailed derivation and study on this topic was carried out, for example, by Anderson and Moore [1979]. In short words, though, the Kalman filter is a state observer for the stochastic case, where the system's dynamics are subject to a Gaussian zero-mean noise $\vec{v}$ and the outputs also to some Gaussian zero-mean noise $\vec{e}$, hence yielding the new system linear model

$$\begin{aligned} \dot{\vec{x}} &= A\,\vec{x} + B\,\vec{u} + \vec{v} \\ \vec{y} &= C\,\vec{x} + D\,\vec{u} + \vec{e} \end{aligned} \tag{5.1}$$

The filter, whose linear model[1] is described in (5.2), can be seen as a Linear Time-Invariant (LTI) system operating in parallel to the real physical system in order to generate an *optimal* estimate $\dot{\hat{x}}$ of all states whereas compensating for the noise effects, as shown in Fig. 5.1, where it forms the well-known LQG structure along an LQ-controller.

$$\begin{aligned} \dot{\hat{x}} &= A\,\hat{x} + B\,u + L\,(y - C\,\hat{x}) \\ \hat{y} &= C\,\hat{x} \end{aligned} \tag{5.2}$$

where L is the steady-state Kalman gain given by (5.3) wherein P is the steady-state estimation error covariance obtained by solving the Algebraic Riccati Equation (ARE) in (5.4).

[1] We drop the $\vec{\cdot}$ index for notation simplicity.

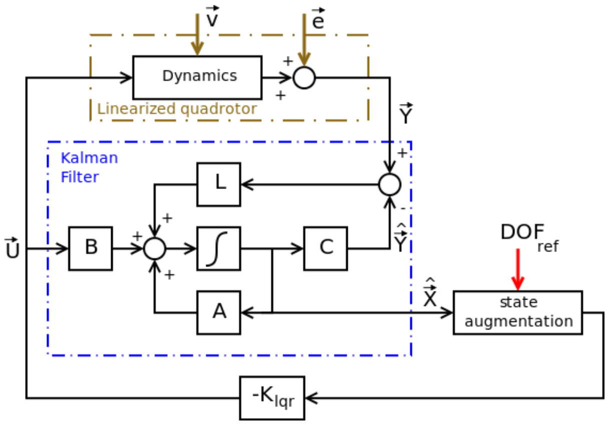


Figure 5.1: LQG composed of continuous-time Kalman Filter and LQ control.

$$L = P\,C^T\,R^{-1} \tag{5.3}$$

$$A\,P + P\,A^T + Q - P\,C^T\,R^{-1}\,C\,P = 0 \tag{5.4}$$

For the quadrotor, the 3D position $\vec{X}$ can be determined either by Global Positioning System (GPS)[2], for outdoors applications, or some infrared or even vision system for the indoors case. Assuming the latter one, a sensor output noise standard deviation of $\sigma_n = 5\,\mathrm{cm}$ is assumed for each component. Other sensor measurements are the angular rates vector $\vec{\Omega}$, by means of the on-board gyros and the angular speed ω_j of each rotor, obtained from the BLDCM's driver, possibly through some *Hall effect* based sensor.

Initially it was assumed that all body-fixed speeds $\vec{V}$ and Euler angles $\vec{\Theta}$ would not be measured in this simulation, but then problems in the design of the Kalman filter LTI system arose due to non-observability of one mode, namely ψ. Therefore we assume it is measurable through some magnetic sensor that perceives the orientation of Earth's magnetic field, for the outdoors case, or, for the indoors scenario, again with some vision system, which is the assumption here. The output vector is then $\vec{y} = \begin{bmatrix} x & y & z & p & q & \psi & r & \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T$

In addition to the sensor (output) noises, some process noise disturbing the aircraft's force equations is assumed, hence affecting $\vec{V}$ and consequently $\vec{X}$ after integration. Table 5.1 summarizes all noise sources and their characteristics.

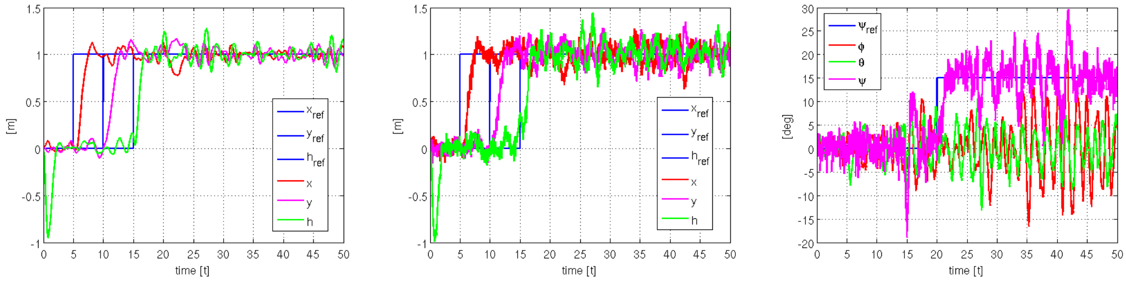
Type	Affected variable	Index	Standard deviation
Process $\vec{v}$	$\vec{V}$	2,4,6	1 m/s
Output $\vec{e}$	$\vec{X}$	$1\dots3$	5 cm
	$\vec{\Omega}$	4,5,7	1 deg/s
	ψ	6	2 deg
	$\vec{\omega}$	$8\dots11$	5 rad/s

Table 5.1: Zero-mean Gaussian-distributed noises acting on quadrotor.

The simulation of the noisy system and experimentation with the Kalman filter to be presented now will make use of an LQ control which will be approached later on in section 6.2. The effect of such noises on the closed-loop LQ-controlled non-linear model is shown

[2] Resolution of $\approx 1\,m$ might be an issue, though.

in Fig. 5.2, where the same Degree of Freedom (DOF) reference vector as for Fig. 6.8 and 6.9 was applied. The quite wide oscillations on ϕ and θ are not due to the noise sources, but as a consequence of the controller trying to stabilize the aircraft in the desired attitude reference. Despite these noises, the LQ controller manages to maintain the aircraft stable, but for more demanding maneuvers, like the ones used for Figs. 6.10 and 6.11, its stability might be significantly hindered, as it was the case[3], motivating the employment of such a filtering technique.



(a) Process noise measured on $\vec{X}$. (b) Total noise measured on $\vec{X}$. (c) Total noise measured on $\vec{\Theta}$.

Figure 5.2: Effect of noises on the LQ-controlled non-linear quadrotor model.

Tuning of the Kalman filter occurs by means of choosing the elements of the diagonal matrices Q_k and R_k in (5.4). In fact, if one knew *a priori* the exact covariance of the noises $\vec{v}$ and $\vec{e}$, the best possible tuning would be achieved, by definition, as $Q_k = \mathcal{E}\left\{v\, v^T\right\}$ and $R_k = \mathcal{E}\left\{e\, e^T\right\}$, with $\mathcal{E}\{g\}$ being the expected value of a random variable $g(t)$. However, both matrices are taken as relative to each other, and assuming that the precise characteristics of the noises are not known, the tuning is done by an iterative process of changing the matrix elements and verifying the filtering results. The bigger $Q_{k_{ii}}$, the faster is the filter on following the process noise, which is a desired characteristic, but the less output noise from the sensors is filtered away. On the other hand, by increasing $R_{k_{ii}}$ the opposite is achieved: better output noise filtering but with slower convergence, i.e. less process noise tracking.

5.2 Filtering Results

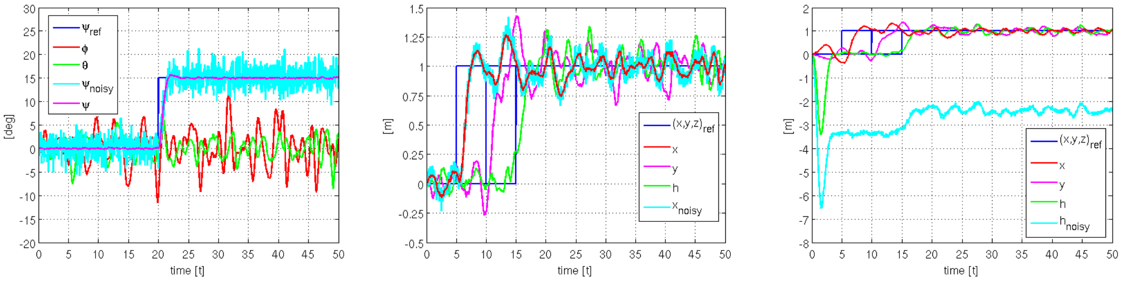
For the noise scenario hereby presented, a satisfying filtering was obtained with the settings

$$\begin{aligned} Q_k &= \operatorname{diag}\left(\left[\begin{array}{cccccccccccccccc} 1 & 5000 & 1 & 5000 & 1 & 5000 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]\right) \\ R_k &= 50 \cdot \operatorname{diag}\left(\left[\begin{array}{ccccccccccc} 3 & 3 & 3 & 1 & 1 & 2 & 1 & 300 & 300 & 300 & 300 \end{array}\right]\right) \end{aligned} \tag{5.5}$$

Having the LQ controller tuned as in (6.12), the Kalman filter was applied to both the linear and the non-linear models of the aircraft, whose results can be seen in Fig. 5.3.

We can observe that the output noise on ψ is very well filtered, while the oscillatory behavior of ϕ and θ was slightly reduced due to a more efficient control action enabled by the filtering. Noise on $\vec{X}$ was also efficiently filtered out, as illustrated for the component x in Fig. 5.3b. However, in Fig. 5.3c the real practical issue with the employment of the Kalman filter on a non-linear system can be verified: due to the system's non-linearity, the filtered and estimated states are generated based on a linearized model. In the beginning of the simulation, when the controller is trying to get the aircraft into the hovering condition, the linearized model does not apply, which yields the steady-state discrepancy on the filtered

[3] Even though not only due to the noise, but also because of the system's non-linearities.



(a) Linear model $\vec{\Theta}$.

(b) Linear model $\vec{X}$.

(c) Non-linear model $\vec{X}$.

Figure 5.3: Kalman filtering results.

altitude h, when compared to the measured noisy signal. However, for any states on which the linearization does not depend, as $x_{11} = \psi$, such discrepancy problem does not appear[4].

Two ways for coping with this issue can be promptly suggested: either the Kalman filter is calculated at several linearization points, and some switching to the closest model is done *on-line*, or the so-called *Extended Kalman Filter* can be implemented, whose main feature is precisely to constantly linearize the system, however not without some drawbacks, especially of not being an *optimal* estimator and having a higher possibility of divergence if the filter is not configured to be fast enough. Nevertheless, it is a broadly adopted solution for the case of non-linear systems, usually yielding satisfactory results.

[4] The linear model does not depend on a ψ_0.

Chapter 6

Control Design

Having the quadrotor model already properly identified, we proceed to the design of its control loops. In fact, the quadrotor, as an open-loop system, is highly non-linear and unstable, therefore the need for designing an efficient and reliable control.

The quadrotor can be regarded as a Multiple-Inputs-Multiple-Outputs (MIMO) system composed of the rotor plus the airframe dynamics, as depicted in Fig. 6.1.

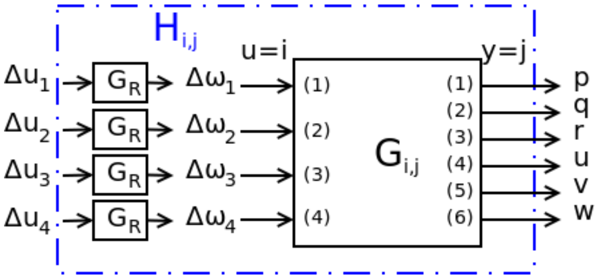


Figure 6.1: Quadrotor's airframe $G_{i,j}(s)$ and total $H_{i,j}(s)$ MIMO system.

All other outputs and/or states such as the Euler angles $\vec{\Theta}$ and the translational speed $\vec{V}$, XY-position $\vec{P}$ and altitude h in the navigation frame can be obtained by manipulating the fundamental outputs represented in this scheme. The ultimate control goal is to follow a position $\vec{P}_{\text{ref}}$ and altitude h_{ref} references on the navigation frame. As can be seen from (3.9), they can be achieved by properly choosing $\vec{\Theta}$. Moreover, as it will be demonstrated next, by linearizing the quadrotor's model around the hovering condition, the vertical speed component w of the body-fixed coordinate system becomes independent of $\vec{\Theta}$, thus depending only on the total thrust T, which in turn depends on the speed of each rotor. As a consequence, an extra 4^{th} DOF appears, for the independent altitude control.

Hereby four approaches shall be addressed: first, the classical PID with nested control loops will be employed, followed by state-feedback LQ-optimal control and then arriving at modern robustness-oriented control techniques as mixed-sensitivity H_∞ and μ-synthesis with DK-iteration for dealing with parametric and model uncertainties.

6.1 Classical (PID) Control

As it was discussed in chapter 4, the Euler angles can be obtained by properly evaluating the angular rates $\vec{\Omega}$. These, in turn, are determined by the thrust (hence speed) of each rotor as well as by their gyroscopic and the Coriollis effect of the airframe. Finally, the speed of each rotor depends on the armature voltage applied to them. This analysis enables us to propose the control architecture shown in Fig. 6.2, which handles the four DOFs of the closed-loop system: XY-position $\vec{P}$, altitude h and heading ψ, each respectively represented by the boxes in colors blue, red and green.

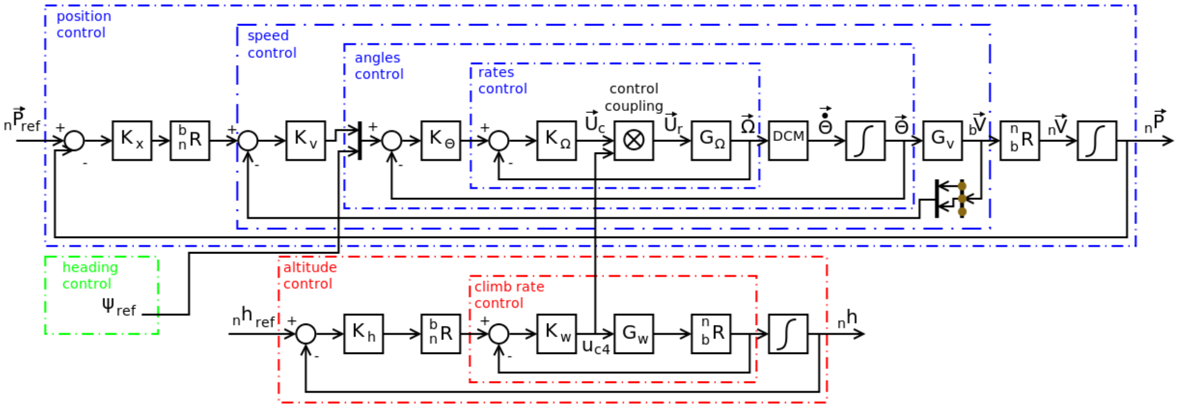


Figure 6.2: Proposed nested classical PID control architecture.

First of all it is important highlighting that the speed control loop refers to the body-fixed coordinate system whereas the position is resolved in the navigation one, therefore the need for applying the coordinate system transformation ^b_nR. This also holds for the altitude and climb rate control loops.

The tuning of each control loop is achieved by using the *root locus* method for finding a stable and desired time (step) response. For such, the quadrotor's model was linearized at the hovering situation, namely where $(p_0, q_0, r_0) = (0, 0, 0)$, $(u_0, v_0, w_0) = (0, 0, 0)$ and $(\phi_0, \theta_0) = (0, 0)$. Also, by disregarding the gyroscopic effect as the 2$^{\text{nd}}$ term in equations $\Delta\dot{p}$ and $\Delta\dot{q}$ of (3.10), having the same speed ω_0 and consequently the same thrust derivative $\dot{T}_0$ for all rotors, and defining the constants $\alpha = \frac{l_a \dot{T}_0}{I_x}$, $\beta = 2\, k_D\, \omega_0 + B_a$ and $\sigma = \frac{\dot{T}_0}{m}$, it yields

$$\left\{\begin{array}{rcl} \dot{p} & = & \alpha\,(\Delta\omega_2 - \Delta\omega_4) \\ \dot{q} & = & \alpha\,(\Delta\omega_3 - \Delta\omega_1) \\ \dot{r} & = & \frac{1}{I_z}\sum_{j=1}^{4}\left[I_G\,\Delta\dot{\omega}_j + (\beta + B_a)\Delta\omega_j\right](-1)^j \end{array}\right. \qquad \left\{\begin{array}{rcl} \dot{u} & = & -g\,\Delta\theta \\ \dot{v} & = & g\,\Delta\phi \\ \dot{w} & = & -\frac{\dot{T}_0}{m}\sum_{j=1}^{4}\Delta\omega_j \end{array}\right. \tag{6.1}$$

By applying the Laplace transform to (6.1), considering the rotor dynamics $\Delta\omega = G_{R_s}\,\Delta u_r$ as in (3.7) and defining the output $\vec{Y} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ and input vector $\vec{U}_r = \begin{bmatrix} \Delta u_{r1} & \Delta u_{r2} & \Delta u_{r3} & \Delta u_{r4} \end{bmatrix}^T$, the MIMO matrix of transfer functions is obtained as

$$H_{i,j}(s) = G_{i,j}(s)\cdot G_{R_s}(s) = \begin{bmatrix} \frac{g\,\alpha}{s^3} & 0 & -\frac{g\,\alpha}{s^3} & 0 \\ 0 & \frac{g\,\alpha}{s^3} & 0 & -\frac{g\,\alpha}{s^3} \\ -\frac{\sigma}{s} & -\frac{\sigma}{s} & -\frac{\sigma}{s} & -\frac{\sigma}{s} \\ 0 & \frac{\alpha}{s} & 0 & -\frac{\alpha}{s} \\ -\frac{\alpha}{s} & 0 & \frac{\alpha}{s} & 0 \\ -\frac{I_G\,s+\beta}{I_z\,s} & \frac{I_G\,s+\beta}{I_z\,s} & -\frac{I_G\,s+\beta}{I_z\,s} & \frac{I_G\,s+\beta}{I_z\,s} \end{bmatrix} \cdot G_{R_s}(s) \tag{6.2}$$

To tune such system in this configuration with the classical transfer function approach is not straightforward, since the transfer, input and output poles and zeros have to be considered. However, an advantageous effect of the linearization done here is the decoupling of the outputs p and q. Moreover, due to the equality of some elements of $G_{i,j}$ in (6.2), by defining a new input vector as $\vec{U}_c = \begin{bmatrix} u_{c1} & u_{c2} & u_{c3} & u_{c4} \end{bmatrix}^T$ where $u_{c1} = \Delta u_{r2} - \Delta u_{r4}$, $u_{c2} = \Delta u_{r3} - \Delta u_{r1}$, $u_{c3} = \Delta u_{r2} + \Delta u_{r4} - \Delta u_{r1} - \Delta u_{r3}$ and $u_{c4} = \sum_{j=1}^{4}\Delta u_j$, a decoupling also of the inputs is achieved, and the new configuration of the MIMO system is

$$H_{i,j}(s) = \begin{bmatrix} 0 & -\frac{g\,\alpha\,K_s}{s^3(s+\lambda_s)} & 0 & 0 \\ \frac{g\,\alpha\,K_s}{s^3(s+\lambda_s)} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{\sigma\,K_s}{s(s+\lambda_s)} \\ \frac{\alpha\,K_s}{s(s+\lambda_s)} & 0 & 0 & 0 \\ 0 & \frac{\alpha\,K_s}{s(s+\lambda_s)} & 0 & 0 \\ 0 & 0 & \frac{(I_G\,s+\beta)K_s}{I_z\,s(s+\lambda_s)} & 0 \end{bmatrix} \tag{6.3}$$

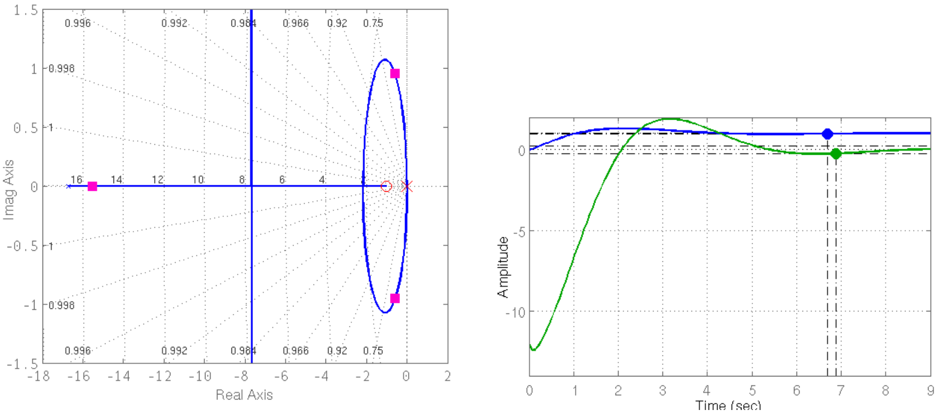
Now the control problem is simplified by approaching the MIMO system as three independent Single-Input-Single-Output (SISO) systems by means of simply solving the system of control input equations

$$\vec{U}_c = W\,\vec{U}_r = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 1 & 0 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \vec{U}_r \quad \Rightarrow \quad \vec{U}_r = W^{-1}\,\vec{U}_c \tag{6.4}$$

where u_{c4} is the output of the climb rate controller, as represented in Fig. 6.2.

6.1.1 Tuning of Control Loops

The first controller to be tuned is the climb rate. Given the integrator in $H_{3,4}$, a P-controller would suffice for stabilization and achieving a satisfactory step response. However, this model only applies precisely at the hovering condition, which is not the case when the Simulink simulation starts, when all rotors have zero initial speed. Hence, by using a mere P-controller, a constant offset is propagated between reference and measured climb rate. For solving this, a PI was tuned as $K_w(s) = -\frac{12(s+1)}{s}$, whose root locus and step response can be seen in Fig. 6.3, yielding a settling time of approx. 6.7 s. The constraint here was the control action, which should obey $\mid \Delta u_{c_4} \mid \leq 12$, given that the remaining voltage to be applied to each of the four rotors is approximately 3 V.



(a) Root locus.

(b) Step response.

Figure 6.3: Tuning of climb rate controller $K_w(s)$.

The next step was to tune the altitude controller K_h, thus completing the vertical control (red boxes in Fig. 6.2). For the horizontal level, the tuning started, as usual, by the innermost loop, the rates controller K_Ω. Again, due to an integrator in the plant, a P-controller already produced satisfactory results. In fact, given the presence of an integrator as part of the subplant in every control loop, P-controllers were designed overall in the control system. Table 6.1 summarizes the setting of such controllers for every loop.

Loop	Controller	Settling time (s)
Climb rate	$K_w(s) = -\frac{12(s+1)}{s}$	6.7
Altitude (height)	$K_h(s) = 2$	2.29
Angular rates	$\left[\begin{array}{ccc} K_p & K_q & K_r \end{array}\right] = \left[\begin{array}{ccc} 0.66 & 0.66 & 22.4 \end{array}\right]$	$\left[\begin{array}{ccc} 0.34 & 0.34 & 0.34 \end{array}\right]$
Attitude (angles)	$\left[\begin{array}{ccc} K_\phi & K_\theta & K_\psi \end{array}\right] = \left[\begin{array}{ccc} 2.85 & 2.85 & 7.3 \end{array}\right]$	$\left[\begin{array}{ccc} 0.7 & 0.7 & 0.36 \end{array}\right]$
Speed	$\left[\begin{array}{cc} K_u & K_v \end{array}\right] = \left[\begin{array}{cc} -0.129 & -0.129 \end{array}\right]$	$\left[\begin{array}{cc} 1.54 & 1.54 \end{array}\right]$
Position	$\left[\begin{array}{cc} K_x & K_y \end{array}\right] = \left[\begin{array}{cc} 0.56 & 0.56 \end{array}\right]$	$\left[\begin{array}{cc} 3.45 & 3.45 \end{array}\right]$

Table 6.1: Controller settings for the classical PID architecture.

Once the controllers were found in the 3-SISO systems approach, the respective controller matrices were assembled for the real MIMO system as

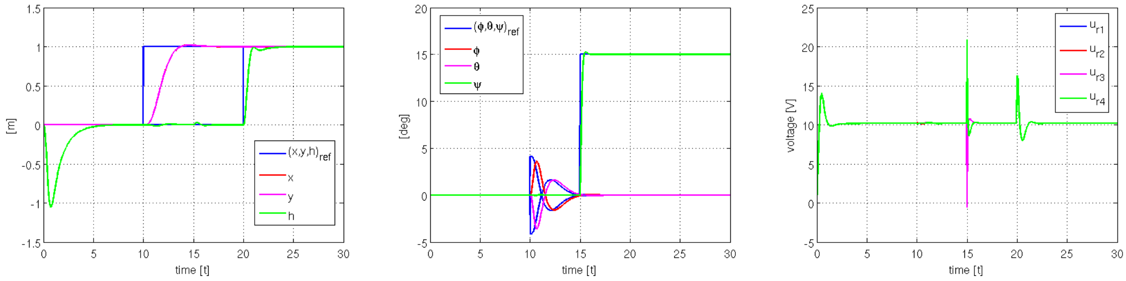
$$K_\Omega = \left[\begin{array}{ccc} K_p & 0 & 0 \\ 0 & K_q & 0 \\ 0 & 0 & K_r \end{array}\right] \; K_\Theta = \left[\begin{array}{ccc} K_\phi & 0 & 0 \\ 0 & K_\theta & 0 \\ 0 & 0 & K_\psi \end{array}\right] \; K_V = \left[\begin{array}{cc} 0 & K_v \\ K_u & 0 \end{array}\right] \; K_X = \left[\begin{array}{cc} K_x & 0 \\ 0 & K_y \end{array}\right] \quad (6.5)$$

where the output of each control loop k is given by $\vec{U}_k = K_k \, \vec{E}_k$ with $\vec{E} = \vec{Y}_{\text{ref}} - \vec{Y}$ being the control error input.

6.1.2 Simulation and Results

Each control loop was independently tested for verifying its stability and reference tracking characteristics. However, for the sake of analysis briefness, we shall here focus on the results of the four closed-loop DOFs previously mentioned.

First, small step references were applied to each DOF channel, whose responses can be seen in Fig. 6.4. After reaching the hovering condition[1] at $t \approx 6\,$s, a XY-position reference $\vec{P}_{\text{ref}} = (1, 1)$ was applied at $t = 10\,$s, resulting in a settling time of $\approx 3.5\,$s with an overshoot of $\approx 2\%$, a very similar performance to the one obtained with the linearized model.



(a) XY-position and altitude.

(b) Attitude $\vec{\Theta}$.

(c) Control action $\vec{U}_r$.

Figure 6.4: Step response around linearization point with PID control.

At $t = 15\,$s a heading reference $\psi_{\text{ref}} = 15^{\text{o}}$ was applied, resulting in a settling time of $\approx 1\,$s with overshoot of $\approx 4\%$. It can be noticed a very small disturbance on the altitude during the yawing[2] maneuver. This is due to the non-linearity of the thrust function being dealt

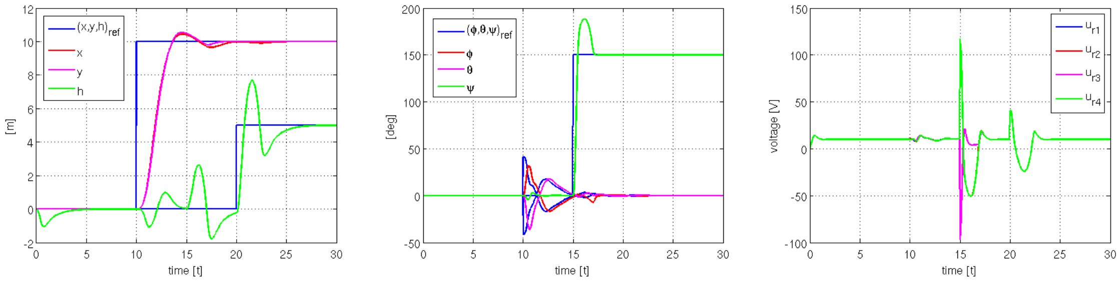
[1] Altitude stabilization at $h_{\text{ref}} = 0\,$m.

[2] Rotation around the body-fixed Z-axis through a ψ angle.

by the linear classical controllers. Nevertheless, the closed-loop performance of the system is quite good. Finally, at $t = 20\,\mathrm{s}$ a step on the altitude reference $h_{\mathrm{ref}} = 1\,\mathrm{m}$ is applied. The settling time is $\approx 3\,\mathrm{s}$, similar to the foreseen during the respective controller tuning.

By observing Fig. 6.4c we note that an upper-limit saturation of the control action occurs at times during this flight simulation, given that the maximum voltage input to any rotor is $u_{r_{\max}} = 12.6\,\mathrm{V}$, especially due to the heading control, which is quite fast. However, for this flight path reference, no significant performance deterioration is observed.

In order to verify the performance of the linear controllers against the system non-linearities, the same sequence of maneuvers was applied, yet this time with bigger step reference magnitudes, as shown in Fig. 6.5. At $t = 10\,\mathrm{s}$ the new XY-position reference was $\vec{P}_{\mathrm{ref}} = (10, 10)$, resulting in a bit longer settling time ($\approx 5\,\mathrm{s}$) as well as a slightly bigger overshoot ($\approx 2.5\%$). When the bigger step on the heading reference $\psi_{\mathrm{ref}} = 150^{\mathrm{o}}$ was applied, the disturbance caused in the altitude regulation was much bigger, reaching a peak of $\approx 2.6\,\mathrm{m}$, but still being compensated. The yaw angle also successfully followed the reference, however with a bigger overshoot of $\approx 25.6\%$ and a bit longer settling time of $\approx 2\,\mathrm{s}$.



(a) XY-position and altitude.

(b) Attitude $\vec{\Theta}$.

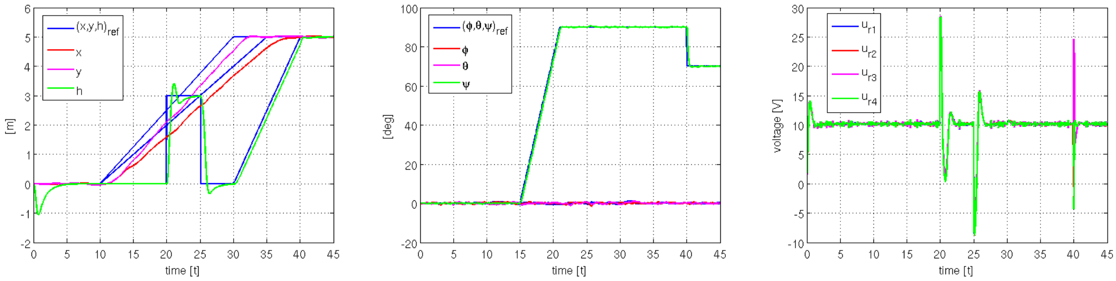
(c) Control action $\vec{U}_r$.

Figure 6.5: Step response further from linearization point with PID control.

Again, we verify control action saturation, this time on both upper and lower limits, as shown in Fig. 6.5c. Due to the larger step references, the saturation also lasts longer, which adds up to the model non-linearity in deteriorating the control performance in terms of the much bigger overshoots observed.

In fact, during the simulation experiments it was observed, for instance, that if the altitude step reference is even increased to $h_{\mathrm{ref}} = 10$ m the system becomes unstable. This problem, however, is significantly minimized if the references are given as ramps rather then steps, as illustrated in Fig. 6.6, where also output noise on the angular rate sensors is simulated. Also, for broadening the stability region of the classical linear controllers, if necessary, their gain could be decreased, despite making the closed-loop system slower. Another way would be to implement the *gain scheduling* technique, whereby the controller gains are fine tuned at specific linearization points (linearization grid) and then an interpolation is executed among these pre-calculated gains at the current operational point of the system.

Still, for the proposed classical control architecture, the overall closed-loop performance is satisfactory, especially if one considers that the pilot would be defining a speed reference to the system, which would then be integrated as a ramp position reference, which in turn is quite nicely followed, as can be observed in the plots of Fig. 6.6, thus avoiding the instability threats of large step references. It can also be noted that the control efficiently rejects the disturbances induced by the noisy sensor readings.



(a) XY-position and altitude. (b) Attitude $\vec{\Theta}$. (c) Control action $\vec{U}_r$.

Figure 6.6: Flight trajectory and heading reference tracking with PID control and noisy $\vec{\Omega}$ sensors.

6.2 LQ Control

An appealing alternative for the control of MIMO systems is the *linear quadratic regulator*, usually referred to by the abbreviation LQ or even LQR. A detailed study of this topic can be found in Lewis and Syrmos [1995]. In summary, the LQ is a kind of state-feedback control that works with the state-space description of the plant

$$\begin{aligned} \dot{\vec{x}} &= A\,\vec{x} + B\,\vec{u} \\ \vec{y} &= C\,\vec{x} + D\,\vec{u} \end{aligned} \tag{6.6}$$

The state-feedback matrix[3] of gains is determined by the LQ approach as

$$K_{\text{lqr}}(\infty) = R^{-1}\,B^T\,S(\infty) \tag{6.7}$$

where $S(\infty) = S$ is the solution of the ARE

$$0 = A^T\,S + S\,A - S\,B\,R^{-1}\,B^T\,S + Q \tag{6.8}$$

which minimizes the linear quadratic cost function (criterion)

$$J_\infty = \frac{1}{2}\int\limits_0^\infty \left(\vec{x}^T\,Q\,\vec{x} + \vec{u}\,R\,\vec{u}\right)dt \tag{6.9}$$

considering the *infinite horizon* situation. For the sake of notation simplicity we shall drop the ∞ index from now on.

The LQ can be applied as a controller in the sense of not only regulating the system's states to the state-space origin (zero), but also for reference tracking. For such, artificial states need to be created as the integral of the control error, for guaranteeing asymptotic tracking. This results in augmented A and B matrices of the system in (6.6) for the purpose of controller design. Before, however, we shall consider the state-space model of the quadrotor, obtained by linearization at a general operational point defined by $(u_0, v_0, w_0) \in \mathbb{R}_3$, $(\phi_0, \theta_0) \in \mathbb{R}_2 \wedge \{-\pi < \phi_0, \theta_0 \geq \pi\}$, $(p_0, q_0, r_0) \in \mathbb{R}_3$ and $\omega_{j_0} \in \mathbb{R}\ \forall j = 1 \ldots 4$. By defining the state vector as $\vec{x} = \begin{bmatrix} x & \Delta u & y & \Delta v & z & \Delta w & \Delta\phi & \Delta p & \Delta\theta & \Delta q & \psi & \Delta r & \Delta\omega_1 & \Delta\omega_2 & \Delta\omega_3 & \Delta\omega_4 \end{bmatrix}^T$ and the linearization constants as shown in table 6.2, from (3.10) and (3.11) it comes

[3] Also known as the *Kalman gain*.

$\omega_{R_0} = \sum_{j=1}^{4} \omega_{j_0} (-1)^j$	$\alpha_{u_1} = -g\ \cos\theta_0$	$\alpha_{v_1} = g\ \cos\phi_0\ \cos\theta_0$	$\alpha_{v_2} = -g\ \sin\phi_0\ \sin\theta_0$
$\alpha_{w_1} = -g\ \sin\phi_0\ \cos\theta_0$	$\alpha_{w_2} = -g\ \cos\phi_0\ \sin\theta_0$	$\alpha_{\phi_1} = \cos\phi_0\ \tan\theta_0\, q_0$	$\alpha_{\phi_2} = \frac{\sin\phi_0}{\cos^2\theta_0} q_0$
$\alpha_{\phi_3} = \sin\phi_0\ \tan\theta_0$	$\alpha_{\phi_4} = -\sin\phi_0\ \tan\theta_0\, r_0$	$\alpha_{\phi_5} = \frac{\cos\phi_0}{\cos^2\theta_0} r_0$	$\alpha_{\phi_6} = \cos\phi_0\ \tan\theta_0$
$\alpha_{p_1} = \frac{I_G\, \omega_{R_0} + (I_y - I_z) r_0}{I_x}$	$\alpha_{p_2} = \frac{(I_y - I_z) q_0}{I_x}$	$\alpha_{p_3} = -\frac{I_G\, q_0}{I_x}$	$\alpha_{p_4} = \frac{l_a\, \dot{T}_{2_0} + I_G\, q_0}{I_x}$
$\alpha_{p_5} = \frac{-l_a\, \dot{T}_{4_0} + I_G\, q_0}{I_x}$	$\alpha_{\theta_1} = -\sin\phi_0\, q_0$	$\alpha_{\theta_2} = \cos\phi_0$	$\alpha_{\theta_3} = -\cos\phi_0\, r_0$
$\alpha_{\theta_4} = -\sin\phi_0$	$\alpha_{q_1} = -\frac{I_G\, \omega_{R_0} + (I_x - I_z) r_0}{I_y}$	$\alpha_{q_2} = -\frac{(I_x - I_z) p_0}{I_y}$	$\alpha_{q_3} = \frac{-l_a\, \dot{T}_{1_0} + I_G\, p_0}{I_y}$
$\alpha_{q_4} = -\frac{I_G\, p_0}{I_y}$	$\alpha_{q_5} = \frac{l_a\, \dot{T}_{3_0} + I_G\, p_0}{I_y}$	$\alpha_{\psi_1} = \frac{\cos\phi_0}{\cos\theta_0} q_0$	$\alpha_{\psi_2} = \frac{\sin\phi_0\ \sin\theta_0}{\cos^2\theta_0} q_0$
$\alpha_{\psi_3} = \frac{\sin\phi_0}{\cos\theta_0}$	$\alpha_{\psi_4} = -\frac{\sin\phi_0}{\cos\theta_0} r_0$	$\alpha_{\psi_5} = \frac{\cos\phi_0\ \sin\theta_0}{\cos^2\theta_0} r_0$	$\alpha_{\psi_6} = \frac{\cos\phi_0}{\cos\theta_0}$
$\alpha_{r_j} = \frac{I_G\, \lambda_s - (2\, k_D\, \omega_{j_0} + B_a)}{I_z} (-1)^{j+1}\ \forall\, j = 1 \dots 4$		$\dot{T}_{j_0} = \gamma_1 + 2\, \gamma_2\, \omega_{j_0} + 3\, \gamma_3\, \omega_{j_0}^2\ \forall\, j = 1 \dots 4$	

Table 6.2: State-space linearization constants for quadrotor at generic operational point.

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & r_0 & 0 & -q_0 & 0 & 0 & \alpha_{u_1} & -w_0 & 0 & v_0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -r_0 & 0 & 0 & 0 & p_0 & \alpha_{v_1} & w_0 & \alpha_{v_2} & 0 & 0 & -u_0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & q_0 & 0 & -p_0 & 0 & 0 & \alpha_{w_1} & -v_0 & \alpha_{w_2} & u_0 & 0 & 0 & -\frac{\dot{T}_{1_0}}{m} & -\frac{\dot{T}_{2_0}}{m} & -\frac{\dot{T}_{3_0}}{m} & -\frac{\dot{T}_{4_0}}{m} \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_{\phi_1} + \alpha_{\phi_4} & 1 & \alpha_{\phi_2} + \alpha_{\phi_5} & \alpha_{\phi_3} & 0 & \alpha_{\phi_6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{p_1} & 0 & \alpha_{p_2} & \alpha_{p_3} & \alpha_{p_4} & \alpha_{p_3} & \alpha_{p_5} \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_{\theta_1} + \alpha_{\theta_3} & 0 & 0 & \alpha_{\theta_2} & 0 & \alpha_{\theta_4} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{q_1} & 0 & 0 & 0 & \alpha_{q_2} & \alpha_{q_3} & \alpha_{q_4} & \alpha_{q_5} & \alpha_{q_4} \\
0 & 0 & 0 & 0 & 0 & 0 & \alpha_{\psi_1} + \alpha_{\psi_4} & 0 & \alpha_{\psi_2} + \alpha_{\psi_5} & \alpha_{\psi_3} & 0 & \alpha_{\psi_6} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha_{r_1} & \alpha_{r_2} & \alpha_{r_3} & \alpha_{r_4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_s & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_s & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_s & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\lambda_s
\end{bmatrix}$$

$$B = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
-\frac{I_G\, K_s}{I_z} & \frac{I_G\, K_s}{I_z} & -\frac{I_G\, K_s}{I_z} & \frac{I_G\, K_s}{I_z} \\
K_s & 0 & 0 & 0 \\
0 & K_s & 0 & 0 \\
0 & 0 & K_s & 0 \\
0 & 0 & 0 & K_s
\end{bmatrix} \tag{6.10}$$

with $C = \text{eye}(16)$ being an identity matrix, assuming all states are directly and with unitary gain measured, and $D = \text{zeros}(16, 4)$ being a 16-4 null matrix.

As previously discussed, the quadrotor dynamics offers four DOFs, namely the three-dimensional position (XY-position and altitude) plus a heading coordinate (yaw angle). Therefore, to the 16 states of the model (6.10), we add the 4 new ones

$$\begin{aligned}
x_{17} &= \int (X_{\text{ref}} - X) \Rightarrow \dot{x}_{17} = x_{1_{\text{ref}}} - x_1 \\
x_{18} &= \int (Y_{\text{ref}} - Y) \Rightarrow \dot{x}_{18} = x_{3_{\text{ref}}} - x_3 \\
x_{19} &= \int (Z_{\text{ref}} - Z) \Rightarrow \dot{x}_{19} = x_{5_{\text{ref}}} - x_5 \\
x_{20} &= \int (\psi_{\text{ref}} - \psi) \Rightarrow \dot{x}_{20} = x_{20_{\text{ref}}} - x_{20}
\end{aligned} \tag{6.11}$$

in order to obtain the augmented system with matrices A_g, B_g with which to design the LQ controller, whose structure is represented in Fig. 6.7.

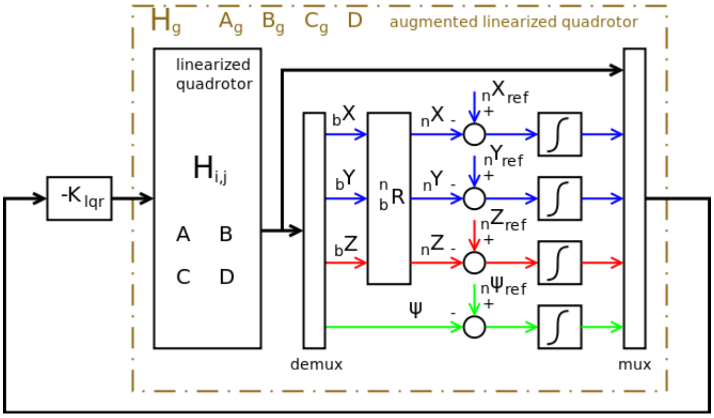


Figure 6.7: State-feedback LQ control structure for the quadrotor.

6.2.1 Tuning of the Kalman Gain

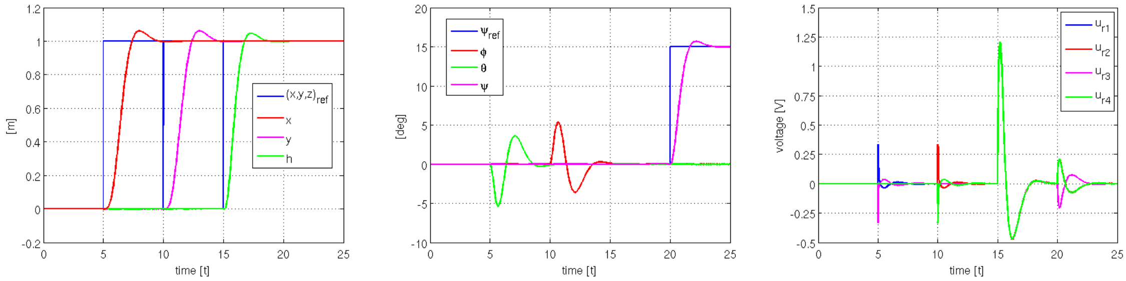
Differently from the classical control, where one has several degrees of freedom for tuning the control loops, such as the control structure (P, PI, PD, PID, . . .) and the position of each zero/pole and consequently the steady-state gain, always looking for a resulting stable feedback-loop which satisfies some time or frequency-domain requirements, the LQ technique requires only the adjustment of the weighting matrices Q and R in the minimization criterion (6.9).

Here also, the fundamental hovering operational point was considered for a first tuning of the feedback matrix K_{lqr}. Q and R are diagonal matrices and can be initially instantiated as identity matrices. By increasing the weight of each $q_{i,i}$ element, the regulation (or reference tracking) on that particular state becomes faster, usually requiring a stronger control action, whereas precisely the penalization of the latter is obtained by increasing the weight of $r_{j,j}$. Also, what matters for the tuning is the relative values between both matrices, so we start with $R = \text{eye}(2)$ and, after an iterative process of changing $q_{i,j}$ and checking the time responses, a satisfactory closed-loop system performance was obtained with

$$\begin{aligned} Q &= \text{diag}\left(\begin{bmatrix} 0 & 20 & 0 & 20 & 0 & 1 & 10 & 40 & 10 & 40 & 0 & 1 & 0 & 0 & 0 & 0 & 200 & 200 & 20 & 20 \end{bmatrix}\right) \cdot 50 \\ R &= \text{eye}(2) \end{aligned} \tag{6.12}$$

which can be seen in the plots of Fig. 6.8.

At time $t = 5\,\text{s}$ $X_{\text{ref}} = 1\,\text{m}$ was applied, resulting in a settling time of $\approx 5\,\text{s}$ with an overshoot of $\approx 6\%$. The same was observed for the Y state, whereas for the altitude the control achieved was a bit faster, with settling time of $\approx 4\,\text{s}$, and with a slightly less overshoot of $\approx 4.7\%$. At time $t = 20\,\text{s}$ $\psi_{\text{ref}} = 15^\circ$ was applied, yielding a settling time of $\approx 4\,\text{s}$ and overshoot of $\approx 4.5\%$. Concerning the control action, which is the limiting criterion for the controller tuning, at least when the linearized system alone is considered, the control peak reached $\Delta u_{\text{peak}} \approx 1.2\,\text{V}$, considerably below the limit $\Delta u_{\max} = u_{\max} - u_{\text{hov}} = 12.6 - 10.1 = 2.5\,[\text{V}]$. It means that, theoretically, we could push the control action up a bit more, to make the control faster, but the result would be a more oscillatory response. Later, when applying the control to the non-linear system, this stronger control could result in a smaller stability region, which is obviously undesired, so a trade-off between control speed and non-linear stability has to be met. In fact, as demonstrated in Lewis and Syrmos [1995], if Q is selected



(a) XY-position and altitude.

(b) Attitude $\vec{\Theta}$.

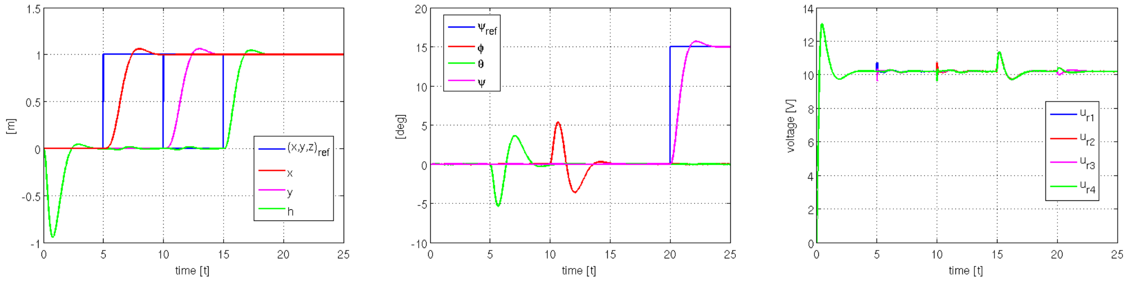
(c) Control action $\vec{U}_r$.

Figure 6.8: LQ control performance on quadrotor's linearized model.

so that $\left(A, \sqrt{Q}\right)$ is observable, then the closed-loop system with K_{lqr} is asymptotically stable. This, however, does not hold to the non-linear plant, therefore the motivation for avoiding too strong control actions.

6.2.2 Simulation and Results

The effect of the system's non-linearities can be seen in the plots of Fig. 6.9, where exactly the same sequence of DOF references was applied to the system as for the linearized version in section 6.2.1.



(a) XY-position and altitude.

(b) Euler angles.

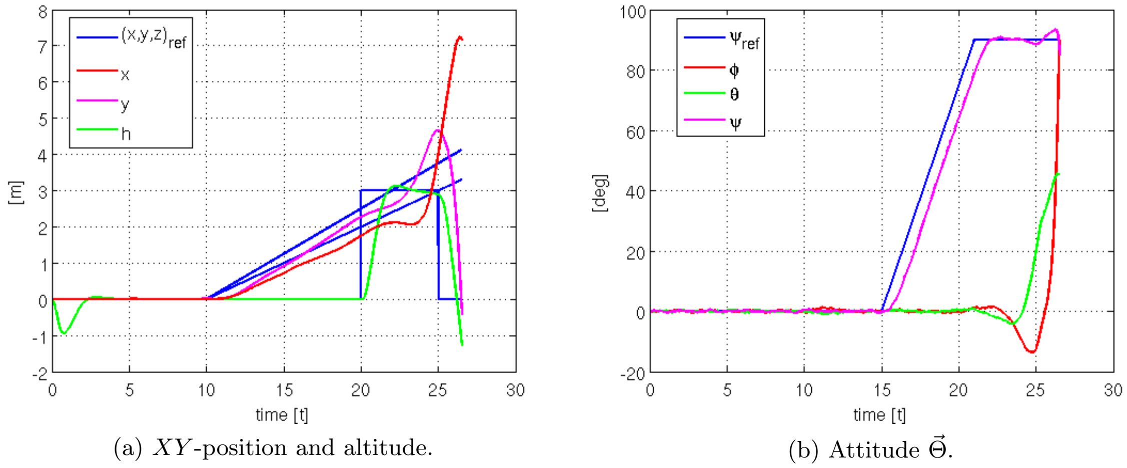
(c) Control action.

Figure 6.9: LQ control performance on quadrotor's non-linear model.

After reaching the hovering condition, the step references on X and Y are achieved with practically the same settling times and overshoots, yet temporarily and slightly disturbing the altitude control, due to the effect of p and q on $\dot{w}$ in (3.9). Nevertheless, asymptotic stability was again observed in all states, with ψ_{ref} being achieved in approximately the same conditions as in the linear case. Regarding the control action, it also behaved quite similarly to the linear design case, except for the first 5 seconds, when the hovering condition was being met.

In order to compare the LQ and PID controllers' performance with each other, the same set of DOF references as in Fig. 6.6, including the sensor (output) noise on $\vec{\Omega}$, was applied to the the LQ-controlled plant. The result is shown in Fig. 6.10. The simulation was supposed to run until $t_{\text{end}} = 50\,\text{s}$ but at $t \approx 26\,\text{s}$ the aircraft went unstable. This proves once more the instability threat when taking the linear closed-loop control system too far from its linerization point.

As an attempt to solve this problem, at least for the given maneuver reference, re-tuning of the LQ controller by experimenting directly on the non-linear model was undertaken.



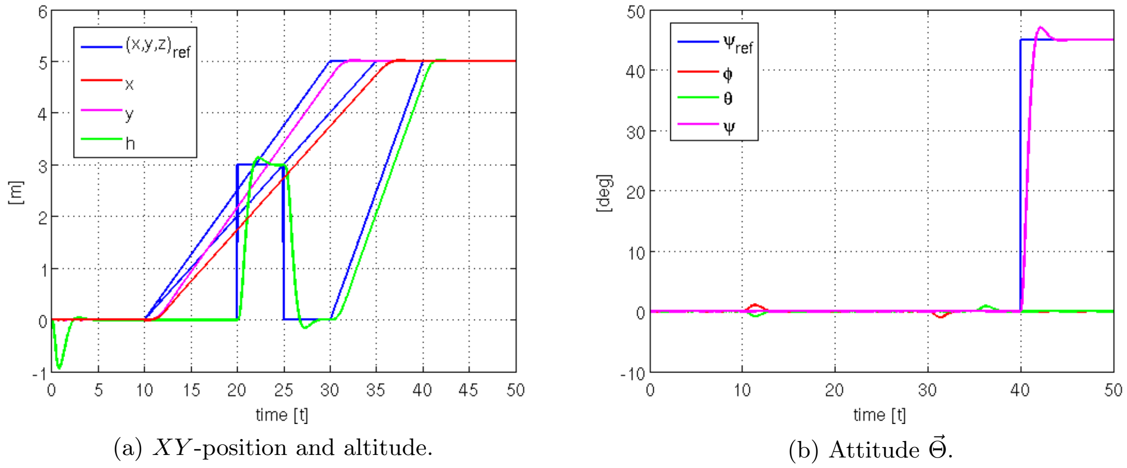
(a) XY-position and altitude.

(b) Attitude $\vec{\Theta}$.

Figure 6.10: Instability of LQ controller tuning for general maneuver and with noisy $\vec{\Omega}$ sensors.

Nevertheless, after numerous trials, unfortunately no stabilizing controller could be found. In fact, by analyzing the plots of Fig. 6.10, one can notice that the X and Y trajectories start to go unstable when $\psi \to \psi_{\text{ref}}$. This leads to the assumption that the conclusion of a yaw maneuver while a 3D-position is still being tracked reduces the stability of the overall non-linear closed-loop system. Or, as a weaker assumption, simply the tracking of ψ affects the oder DOFs when they are occurring simultaneously.

This seems to be the case, as indicate the results shown in Fig. 6.11. There, the 3D maneuver reference remains the same whereas the ramp on ψ was replaced by a step and applied at $t = 40\,\text{s}$ so that, when $\psi \to \psi_{\text{ref}}$, all other DOFs had already met their references. As can be seen, no instability was generated, and the controller performance was comparable to the one obtained previously with the PID. However, no noise was used in this new simulation. If the same noise as for the PID in Fig. 6.6 is applied, the system again goes unstable, despite the separation of the yaw maneuver from the other DOFs.



(a) XY-position and altitude.

(b) Attitude $\vec{\Theta}$.

Figure 6.11: Good performance of LQ controller for isolated yaw maneuver.

Again, this demonstrates the importance of implementing some filtering technique, as discussed in chapter 5, for conferring more stability to the closed-loop control.

Clearly, the results obtained with the ordinary one-level LQ control, in which all DOF references are attempted to be controlled at one single loop, are not as good as the ones obtained with the nested PID control (section 6.1). In fact, a better tuning of the LQ controller might still be possible to achieve, but the procedure gets quite inefficient since the tuning of one DOF ends up affecting, usually in a degrading fashion, the others. This problem does not arise in the nested loops architecture, being the motivation for attempting its use also with the LQ, especially made possible, in the case of the quadrotor, due to the its subdivision into attitude and translation subsystems. An attempt to employ such strategy is depicted in Fig. 6.12. Unfortunately it did not yield better results, possibly due to implementation/calculation mistakes, leaving room for further investigation.

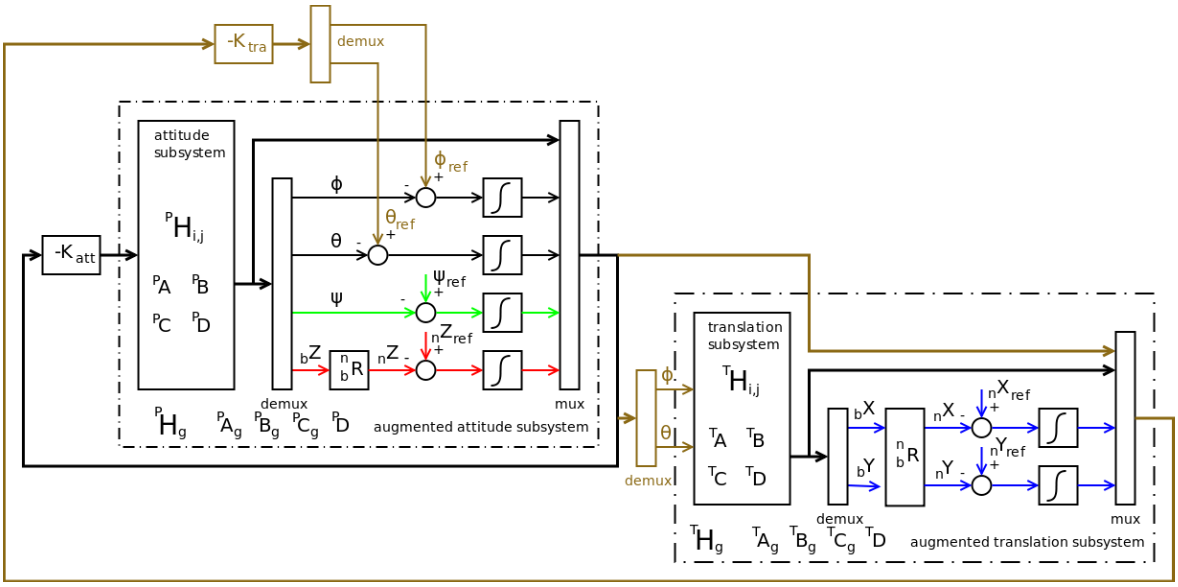


Figure 6.12: Attempt of nested LQ implementation for the quadrotor.

6.3 Robust Control Techniques

The control techniques employed so far assume that the plant's model is perfectly known, i.e. all significant system dynamics is modeled and, moreover, all parameters describing such dynamics are correctly identified. This is frequently an underestimated issue in control design, since sometimes modeling the system's dynamics can be quite a complex task, and thus many assumptions are made in order to simplify the model and make it feasible to work with for later stages of system analysis and control design.

This *unmodeled dynamics* issue appears especially at higher frequencies. However, even when this is not an issue, running identification tests on the real plant might also be not straightforward, yielding not precise identified parameters as desired. Nevertheless, although an initial identification is successful, the plant's parameters may vary in time as a consequence of mechanical wear - friction coefficients increase, for example - or even due to changes in operational conditions, e.g. if a payload device like a camera is attached to the quadrotor's airframe, changing its moments of inertia and total mass.

All these model and parameter uncertainties can easily deteriorate the performance of control loops or even make them go unstable. In the aerospace sector this robustness aspect, which directly affects stability, is an even more crucial aspect of control design, given the human safety and high financial investment usually involved in such applications. Hence the need for system's robustness analysis and control design. The theoretical background for the calculations in the following sections can be found in details in Skogestad and Postlethwaite [2005].

6.3.1 Mixed-Sensitivity $\mathcal{H}_\infty$

This is a control technique based on the frequency domain that aims at finding a stable controller K that minimizes the $\mathcal{H}_\infty$ norm of some system configuration. In plain words, the $\mathcal{H}_\infty$ norm of a SISO system can be thought of as the peak in the frequency domain of its open-loop transfer function L. For the MIMO case, the peak is to be considered in the *singular values* plot, that is, the maximum open-loop amplification from any input to any output considering all frequencies.

Moreover, the Mixed Sensitivity (MS) term refers to the way in wich the $\mathcal{H}_\infty$ norm is to be minimized, namely according to the system's configuration as shown in Fig. 6.13, i.e. between the *exogenous inputs* $\vec{w}$, which in the case of reference tracking correspond to the references, and the *performance outputs* $\vec{z}$. The tuning parameters are the weighting filters W_1, W_2 and W_3. Considering the SISO case, as it will be approached henceforth, W_1 defines how the sensitivity transfer function $S = 1/(1 + H)$, between w and u is to be shaped, a low-pass filter, if we consider that S also corresponds to the channel $d \to y$ where d is a disturbance acting on the system's output ($w = d$) and having a limited bandwidth ω_d. Therefore the shaping of S determines how effectively the closed-loop system will reject output disturbances.

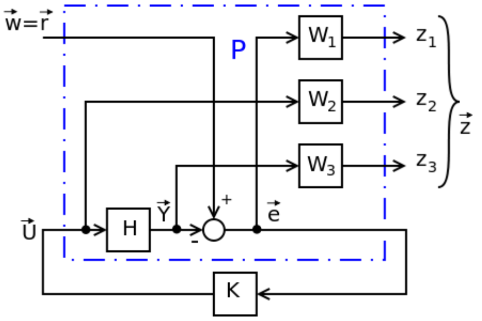


Figure 6.13: $S/K\,S/T$ mixed-sensitivity control design configuration for reference tracking.

In its turn, W_3 determines the complementary sensitivity function $T = H/(1 + H)$, representing the channel $r \to y$, is to be shaped. It is related to reference tracking and also models how a multiplicative uncertainty acts on the system's input u. This is precisely where we want to focus in this robustness approach. Given that at low frequencies the system's model is usually known, whereas at higher frequencies it grows to high values, W_3 is designed as a high-pass filter. For both W_1 and W_3 filters, their design must be so that, for any of the closed-loop transfer functions F after closing the loop with the controller, it obeys

$$\mid W_i\, F \mid \leq 1 \Rightarrow \mid F \mid \leq \frac{1}{\mid W_i \mid} \tag{6.13}$$

which, in plain words, means that the curve of S or T should be below, or at most superpose itself to the respective filter inverse. Once the filters are designed, the MS $\mathcal{H}_\infty$ algorithm can be expressed as finding a stabilizing controller K, over all possible frequencies ω, which minimizes

$$\left\| \begin{bmatrix} W_1\,S \\ W_2\,K\,S \\ W_3\,T \end{bmatrix} \right\|_\infty \tag{6.14}$$

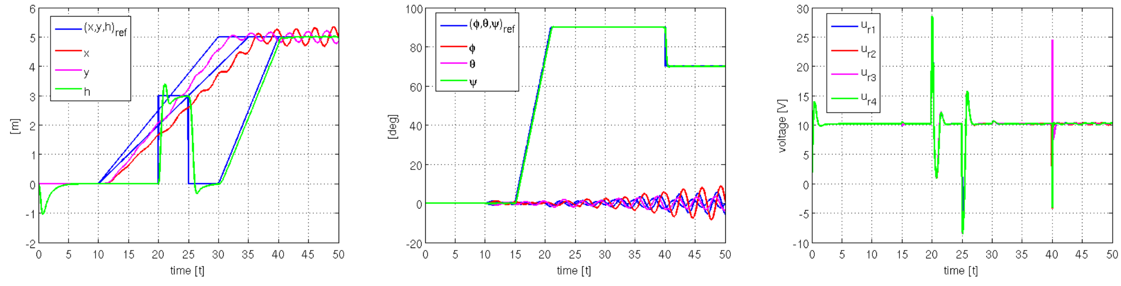
The W_2 filter must not necessarily be included in the calculations. When informed, it defines how the control action $K\,S$ is to be minimized over the frequency range. This could be used if the controller obtained with (6.14) would require a bandwidth higher than what the actuator reaches, therefore W_2 would be modeled as a high-pass filter as well. For the sake of design simplicity though, we shall first attempt a control design without this extra filter.

One practical aspect to take into consideration when designing the weighting filters is that the MS $\mathcal{H}_\infty$ algorithm requires all of them to be proper and stable transfer functions. Therefore, for the case of W_1, one would probably like to have an integrator as pole, in order to maximize disturbance rejection as much as possible in lower frequencies. In practice, it needs to be replaced by a very slow pole, also in order to limit the filter's steady-state gain.

Airframe Uncertainty and Controller Design

As a study-case, let us consider that a camera as payload has been attached to the quadrotor's airframe. It can be modeled as a point-mass of $m_p = 150\,\mathrm{g}$ aligned with the body-fixed Z-axis and distant $l_p = 11.73\,\mathrm{cm}$ of the coordinate system's origin. This has the effect of adding $\Delta I_x = m_p\,l_p^2 = 17.6 \cdot 10^{-3}\,\mathrm{kg\,m^2} = 3\,I_{x_0}$ to the moments of inertia around the X and Y-axes, thus $I'_x = I'_y = 4\,I_{x_0} = 23.5 \cdot 10^{-3}\mathrm{kg\,m^2}$. Such payload scenario is realistic, since the maximum lift-off mass of the quadrotor is $m_{\max} = 1062\,\mathrm{g}$ whereas, with the camera coupled, the aircraft's total mass becomes $m' = 844\,\mathrm{g}$. Nevertheless, for the sake of analysis simplification, let us consider first only the effects on the moments of inertia.

With the nested PID control design, the same trajectory reference as used in Fig. 6.6 was applied, but this time without the $\vec{\Omega}$ noises, yet with the increased I'_x, I'_y. The results can be seen in the plots of Fig. 6.14. The p and q components of the angular rates controller are directly affected, as expected, given their tuning dependence on $\alpha = \frac{l_a\,\dot{T}_0}{I_x}$ of $H_{4,1}$ and $H_{5,2}$ (from now on H_{nom}) in (6.3), and the whole closed-loop system becomes unstable.



(a) XY-position and altitude. (b) Attitude $\vec{\Theta}$. (c) Control action $\vec{U}_r$.

Figure 6.14: Instability on PID closed-loop system induced by payload coupling.

We first proceed to tune W_1 with aid of the template

$$W_1(s) = \frac{s/M_S + \omega_S}{s + A_S\,\omega_S} \tag{6.15}$$

where M_S defines the magnitude in high frequencies. Since we know that oscillations in the frequency domain imply in oscillations in the time domain response, we choose the usual value of $M_S = 2 \approx 6\,\text{dB}$. ω_S is the desired closed-loop bandwidth. Given that we do not have information on the disturbance's bandwidth, we push it as far to the right side as possible while obeying (6.13), thus $\omega_S = 2.5\,\text{rad/s}$. Finally, A_S defines the steady-state gain at low frequencies: the smaller this parameter is made, the less steady-state control error e_∞ and the more disturbance rejection (attenuation) is obtained. We find enough to choose $A_S = 0.001$. The resulting $1/W_1$ and S behavior is presented in Fig. 6.15a, where each blue curve corresponds to S for a given value of I_x in the interval $I_{x0} \leq I_x \leq 4\,I_{x0}$.

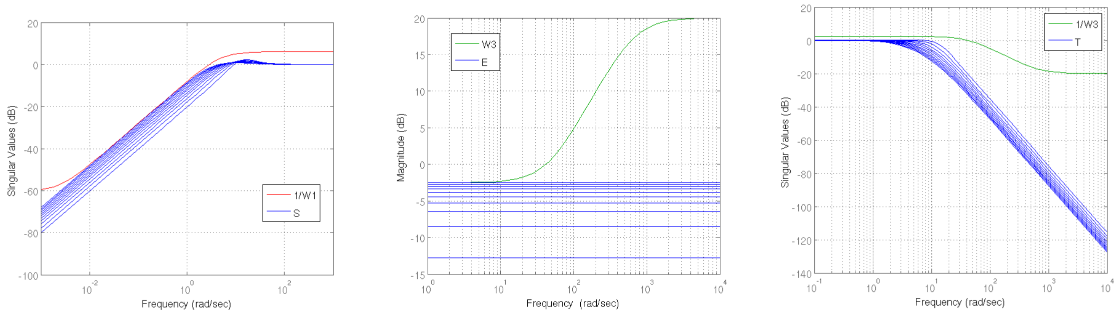
Now we turn to W_3, which shall model the parameter uncertainty related to the changes in I_x, I_y. For this, a relative modeling error $\mathbb{E}$ is obtained as

$$\mathbb{E}(s) = \frac{H_{\text{var}(s)} - H_{\text{nom}}(s)}{H_{\text{nom}}(s)} \tag{6.16}$$

which is also calculated for every variation of I_x, I_y, corresponding to the blue curves in Fig. 6.15b. *A priori* we could define W_3 superposing to the uppermost $\mathbb{E}(j\omega)$. However, with this filter we also aim at representing unmodeled dynamics, such that the modeling error increases at high frequencies. Therefore we use the template

$$W_3(s) = \frac{\tau\, s + r_0}{(\tau/r_\infty)\, s + 1} \tag{6.17}$$

where we choose $r_0 = 0.75 =\approx -2.5\,\text{dB}$ of relative uncertainty at steady-state (low frequencies). $1/\tau$ corresponds approximately to the frequency at which the uncertainty reaches 100%, hence $\tau = 0.016$. Finally, r_∞ represents the uncertainty at high frequencies, which can be made arbitrarily high, but not so much as to make the filter lose its proper condition (pole at $-\infty$). It proved to be enough $r_\infty = 10$.



(a) $1/W_1$ (red) and S (blue).

(b) W_3 and $\mathbb{E}$.

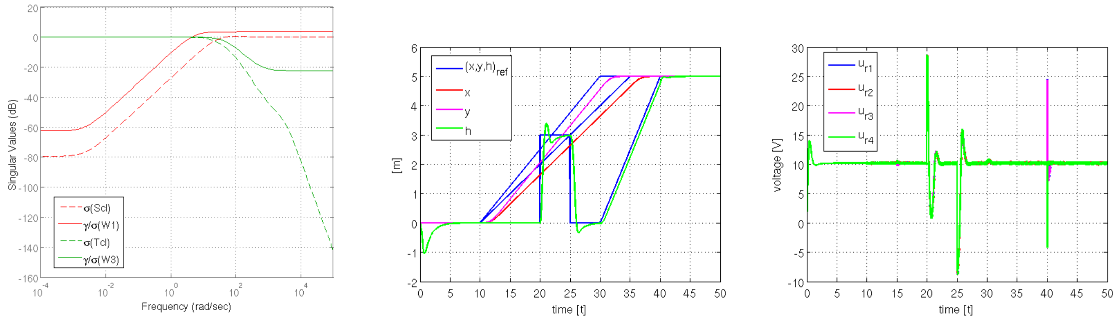
(c) $1/W_3$ and T.

Figure 6.15: Weighting filters for tuning of mixed-sensitivity $\mathcal{H}_\infty$ controller.

After running the algorithm in Matlab by means of the function *mixsyn*, it was obtained a controller of the 4th order, which then replaced K_p and K_q in (6.5). Simulation results with this robust control technique are shown in Fig. 6.16.

We can see that the constraints (6.13) were obeyed after closing the loop with the $\mathcal{H}_\infty$ controller, yielding an excellent performance and robustness even in extreme parameter uncertainty situations, as exemplified in Fig. 6.16b, where $I'_x = I'_y = 4\,I_{x0}$ and the pure PID control goes unstable. It is also interesting to observe that only the innermost control loop of angular rates had to be made robust, since it is directly affected by the uncertainty, whereas all others remained in the original PID tuning.

The results seem perhaps too good to be true, and, in fact, Fig. 6.16c shows the drawback of the tuned MS $\mathcal{H}_\infty$ controller: a quite high frequency behavior on the control action with $10\,\text{V} \leq u_c \leq 10.4\,\text{V}$ at steady-state. This also happens in the case of the nominal system

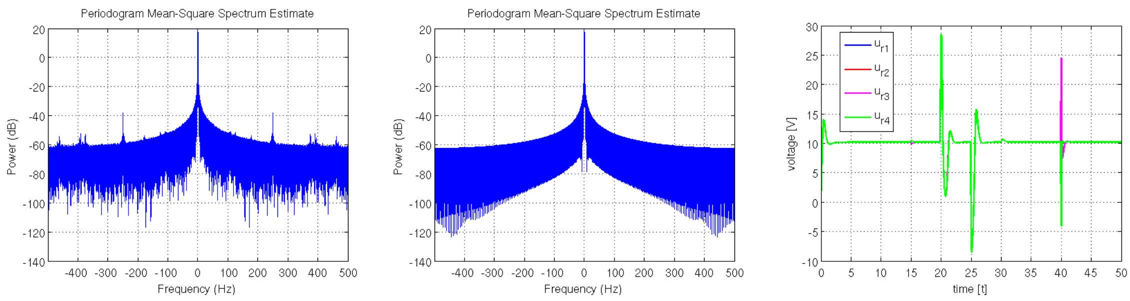


(a) Filters and shaped functions. (b) XY-position and altitude. (c) Control action $\vec{U}_r$.

Figure 6.16: System performance with mixed-sensitivity $\mathcal{H}_\infty$ control for $I'_x = I'_y = 4\, I_{x0}$.

parameters. Depending on the system, the actuator might not have enough bandwidth for following such high-frequency switching, and the desired robustness is not possible to be achieved. Even though in the mathematical model used in this simulation the rotors seemed to have coped well with the high-frequency content of u_r, unmodeled effects such as wire capacitance may, in practice, hinder the foreseen excellent robustness results obtained via simulation.

For coping with such an issue, we now employ W_2. In theory we would like to have it as a high-pass filter for penalizing the control action only at higher frequencies, but by doing so it becomes a hard task for the algorithm to attend all minimization criteria. It turns out to be enough setting the filter to be a constant, say $W_2 = 0.1$. After re-running the MS $\mathcal{H}_\infty$ algorithm, still a controller of order 4 is obtained, but now with a different dynamics so to avoid the high-frequency behavior on the control action, as can be seen in the plots of Fig. 6.17, where some frequency peaks around 250 Hz, 380 Hz and 450 Hz were removed, results obtained also by simulating the closed-loop system with $I'_x = I'_y = 4\, I_{x0}$. If W_2 is made smaller, less high-frequency content is damped. On the other hand, the higher W_2 is taken, the less robustness is achieved. For example, for $W_2 = 1$ the closed-loop control goes unstable for this uncertainty scenario.



(a) Power spectrum of u_{r_1} without W_2. (b) Power spectrum of u_{r_1} with $W_2 = 0.1$. (c) u_{r_1} with $W_2 = 0.1$.

Figure 6.17: Improvement in mixed-sensitivity $\mathcal{H}_\infty$ control performance with W_2.

6.3.2 μ-Synthesis with DK-Iteration

Another modern control technique for taking into consideration model and parameter uncertainty is the so-called *μ-synthesis with DK-iteration*. μ is the structured singular value, a non-negative function which provides a generalization of the singular value $\bar{\sigma}$ of a system, useful for analyzing robust stability and performance conditions, and is defined as

$$\mu(M) \triangleq \frac{1}{\min\left\{k_m \mid \det\left(I - k_m M\Delta\right) = 0 \text{ for structured } \Delta, \bar{\sigma}(\Delta) \leq 1\right\}} \tag{6.18}$$

where $\Delta = \operatorname{diag}\{\Delta_i\}$ is the block-diagonal structured matrix of norm-bounded perturbations $\bar{\sigma}(\Delta) \leq 1$, with which uncertainties in the system can be modeled.

Again, detailed explanation on how this control method works can be found in Skogestad and Postlethwaite [2005]. In general words, though, it combines $\mathcal{H}_\infty$-synthesis and μ-analysis and can be summarized in the following algorithm:

1. **K**-step: synthesize an $\mathcal{H}_\infty$ controller for the scaled problem, $\min_K \parallel DN(K)D^{-1} \parallel_\infty$ with fixed $D(s)$.

2. **D**-step: find $D(j\omega)$ to minimize at each frequency $\bar{\sigma}\left(DND^{-1}(j\omega)\right)$ with fixed N.

3. Fit the magnitude of each element of $D(j\omega)$ to a stable and minimum-phase transfer function $D(s)$. Go to step 1.

The iterations continue until sufficient performance is achieved or the $\mathcal{H}_\infty$-norm no longer decreases. Convergence to a local minimum may happen, however the method works well in most cases. One drawback is the order of the resulting controller K, which might be too high, since it is equal to the number of states in the nominal plant $G(s)$, plus that of the weighting filters, plus twice the number of states in $D(s)$. Therefore, depending on the case, an order reduction of the synthesized controller K might be necessary to allow practical implementation.

Rotor Uncertainty and Controller Design

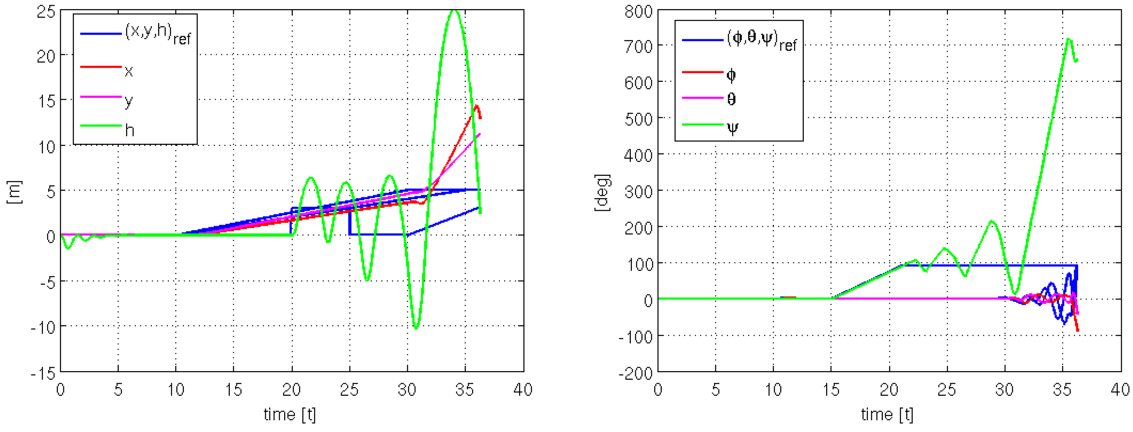
As a study-case for this controller, let us assume now some uncertainty in the rotor. More precisely, let us consider the case where the rotor inertia can vary up to 300% of the nominal value, thus $I_{G_\text{nom}} \leq I_G \leq 4I_{G_\text{nom}}$. The effect is that the rotors become slower[4] than what the pre-tuned control system expects. Nevertheless, this does not affect the XY-position neither the ψ DOF control, at least for the linear model, since the change in one rotor dynamics is compensated by the others, as can be inspected in (6.10). For the non-linear model in (3.8), only minor effects occur in the control of these DOFs. However, in both cases, the h control is significantly affected. In fact, it was verified that for the case when $I'_G = 3.1I_{G_\text{nom}}$, the non-linear system controller by the PID in section 6.1 goes unstable, as illustrated in Fig. 6.18.

Hence, the need for designing a robust controller also for the altitude control. Again, it will be attempted to re-tune only the innermost loop of this DOF, namely the climb rate control with K_w, taking advantage of the nested control architecture. For this, a control design scheme was proposed as depicted in Fig. 6.19.

First, the augmented plant P needs to be instantiated. Since we are only interested in the channel $u_{c4} \rightarrow w$, P contains the nominal transfer function of each rotor G_{r_i} as in (2.15) whose output is summed and fed to the transfer function $\frac{w}{\omega_\text{tot}} = G_w(s) = -\frac{\dot{T}_0}{s\,m}$ as in $G_{3,j}$ of (6.2), with $\omega_\text{tot} = \sum_{j=1}^{4} \omega_j$.

As for the MS $\mathcal{H}_\infty$ control, here we also have to define some filters for the control design. $W_p(s)$ is the performance filter and defines how the minimization of the $\mathcal{H}_\infty$-norm in the

[4] Note that not only the I_G parameter is changed, but also λ_s and K_s in (3.7), since they are results of the first-order approximation of the general second-order model (2.15), in which the poles depend on I_G, but not the steady-state gain.



(a) XY-position and altitude. (b) Euler angles.

Figure 6.18: Instability in PID-controlled non-linear system induced by $I'_G = 3.1 I_{G_{\text{nom}}}$.

channel $w_r \to z$ is to be carried out. In our case of reference tracking, the signal to be shaped is the control error $e = w_r - w$, therefore we choose to set this as a low-pass filter, with aid of the same template (6.15), in order to minimize the error at lower frequencies, where also eventual perturbations might occur. Again here, all filters must be stable, proper LTI systems, therefore W_p must have a bounded steady-state gain. Since this control synthesis has the drawback of not allowing integrators in the design, the greater this gain is made, the less will be the steady-state control error e_∞, therefore we want A_S to be small. The crossover frequency ω_S determines how fast the control action is. We cannot push it too high otherwise u_{c_4} saturates and the closed-loop system goes unstable. Last, M_S determines the filter gain at high frequencies. We leave it as $M_S = 2$ in order to avoid peaks in the sensitivity function S. After multiple trials, the best setting found for the other parameters was $A_S = 10^{-4}$ and $\omega_S = 130\,\text{rad/s}$. The behavior of W_p can be seen in the red plots of Fig. 6.20.

The four W_{I_j} filters model the parametric uncertainty in the rotors. Here, again, by varying I_G in the proposed interval, a set of relative modeling errors as described by (6.16) is obtained, and we choose the filter parameters so as to make the filter magnitude curve in the bode plot to at least limit the set of modeling errors at the top. Again, after multiple tuning attempts, the desired robustness could only be achieved by adding some *extra* uncertainty to the design, i.e. by shifting the crossover frequency of the filter to the left, thus detaching the filter curve from the relative errors. The author believes this was necessary in order to somehow include in the calculations the discrepancy of the non-linear model as some model uncertainty. As expected and shown in Fig. 6.20a, $\mathbb{E}$ increases in higher frequencies, so we set W_I as a high-pass filter whose parameters, again using the template (6.17), are $r_0 = 10^{-6}$, $\tau = 25$ and $r_\infty = 0.81$.

Once having the augmented plant P built and connected to the diagonal-structured uncertainty matrix Δ, the augmented uncertain plant P_{unc} is obtained and, by running the Matlab function *dksyn*, a stabilizing robust controller of the 28$^{\text{th}}$ order is found. In Fig. 6.21 the time plots of the closed-loop system for nominal rotor inertia with this new controller can be seen. Analyzing the altitude control, for the Linear System (LS) there is some significant oscillation during the settling time of $\approx 5\,\text{s}$, with overshoot of $\approx 30\%$. For the Non-Linear System (NLS), the settling time is a bit longer, around $7\,\text{s}$, with a slight bigger overshoot of $\approx 33\%$. We can observe in Fig. 6.21b a small non-zero steady-state error, given the lack of an integrator in both the climb rate and altitude control loops. Also, it is noticeable that

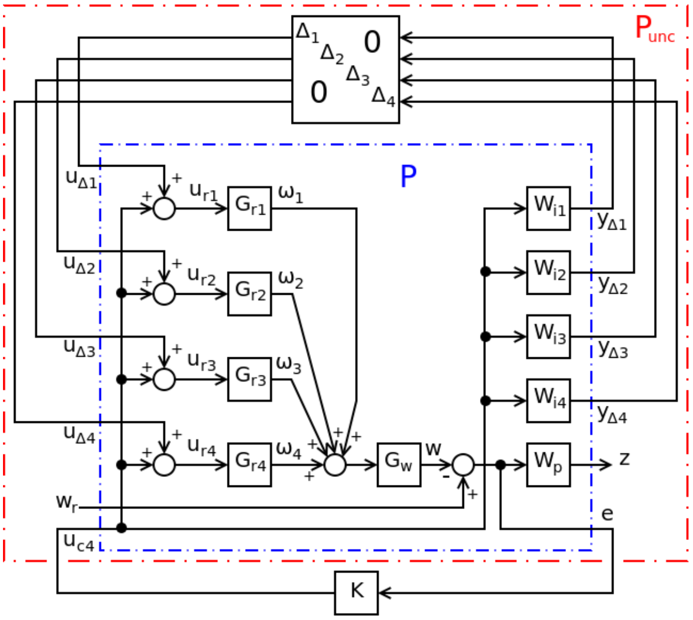
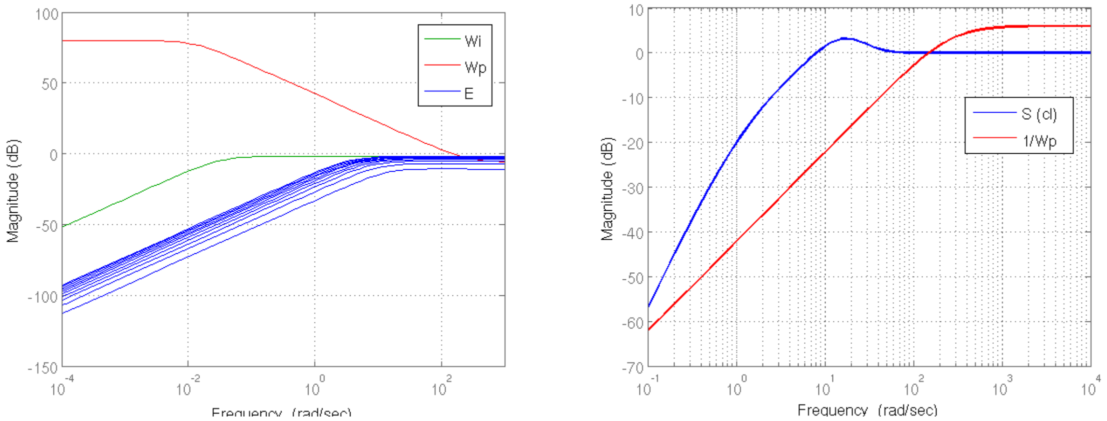


Figure 6.19: Block diagram scheme for synthesis of μ-synthesis with DK-iteration based controller.

the big oscillations occur due to the step inputs on the reference; when following a ramp, the oscillations are significantly less. Also, the control action suffers some brief saturation when the reference changes are applied as well as when the tracking is concluded, but without major consequences for the control.

Now analyzing the uncertainty scenario of $I'_G = 3.1 I_{G\text{nom}}$, as depicted in Fig. 6.22, the performance in both LS and NLS is deteriorated in terms of oscillations in the reference tracking and control action saturation. Nevertheless, the important here is that the altitude control loop remains stable. Concerning the control action, in Fig. 6.22c we see that it suffers more saturation, however not so critically as to hamper the robust stability achieved.

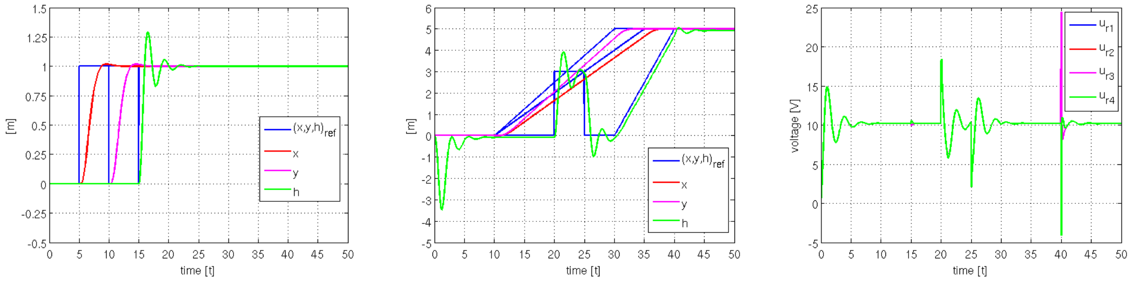
It can be concluded that this control technique indeed offers the advantages of robust stability, however no really adequate tuning for the quadrotor could be achieved, at least for this scenario of I_G parametric uncertainty. The oscillations on h deteriorate the nominal and robust performances, but while trying to eliminate them, the robust stability characteristic was lost. However, one needs to consider that such uncertainty scenario is quite extreme, given that I_G is not a parameter which is expected to vary with time, i.e. it does not suffer wear. In any case, other arrangements of P, like choosing output multiplicative or additive uncertainty for the rotors might produce better results, or perhaps by trying to mix both climb rate and altitude nested loops into a single one. Hence, possibilities for further investigation and improvement in this area remain open.



(a) W_p, W_I and $\mathbb{E}$.

(b) $1/W_p$ and S with K.

Figure 6.20: Weighting filters for μ-synthesis with DK-iteration based controller design.

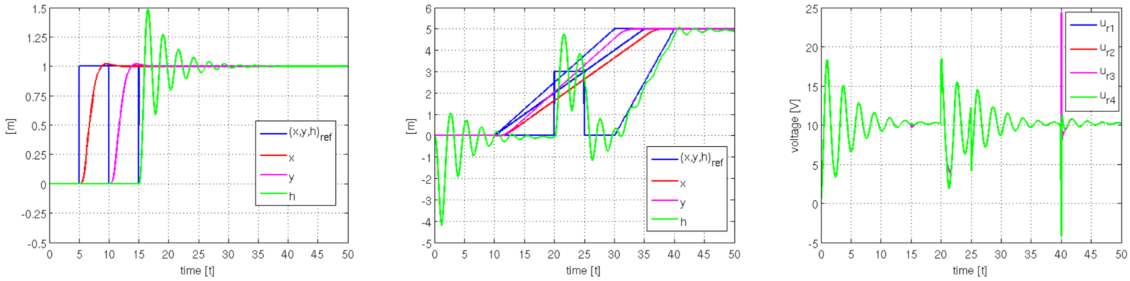


(a) XY-position and h for LS.

(b) XY-position and h for NLS.

(c) $\vec{U}_r$ for NLS.

Figure 6.21: Performance of μ-synthesis with DK-iteration based climb rate controller in the altitude control loop for $I_{G_{\text{nom}}}$.



(a) XY-position and h for LS.

(b) XY-position and h for NLS.

(c) $\vec{U}_r$ for NLS.

Figure 6.22: Performance of μ-synthesis with DK-iteration based climb rate controller in the altitude control loop for $I'_G = 3.1 I_{G_{\text{nom}}}$.

Chapter 7

Final Considerations

As it could be seen throughout this thesis, the proposed goals of this work were achieved. In summary, the fundamentals of flight dynamics of the quadrotor aircraft were investigated and a full mathematical non-linear model was derived using classical mechanics. Furthermore, a linearized model for a general operational point was also obtained, enabling the later use of linear control techniques. Following, an experimental identification of some of the system's parameters was carried out, like the moments of inertia $I_x = I_y$,I_z and I_G, air drag torque coefficient k_D and thrust polynomial γ_i. All others, like the linear friction torque coefficient B_a and armature impedance L_a were indirectly assessed by mathematical manipulation of the previously identified parameters with the system's model. Once the identification was complete, multiple control techniques were employed and their performances compared with each other, especially in terms of reference tracking. Among them, robust control techniques like mixed-sensitivity $\mathcal{H}_\infty$ and μ-synthesis with DK-iteration were also considered for coping with parametric and model uncertainty. Finally, all project files were made available to the author's supervisor at DCE, thus collaborating for future studies which can rely on the results already obtained herein.

However, in order to make all these results feasible to be achieved in the time span of approximately six months, many assumptions and significant simplifications were made. One of them is the disregard of the aerodynamic drag force acting on the quadrotor airframe when it translates in the air. For simulation of the system near the hovering condition this is a reasonable assumption, nevertheless, for more realistic studies, when considering significant translational speeds of the aircraft, by still disregarding such air drag effect one runs into the problem of not being able to use the accelerometer readings $\vec{a} = (a_x, a_y, a_z)$ together with data from the gyros in some data fusion algorithm for obtaining the aircraft attitude. The accelerometer reads a combination of Earth's gravity $\vec{g}$ and the translational acceleration $\vec{a}_T$. The drag force opposes itself to the thrust force which translates the aircraft. When steady-state is achieved, i.e. a constant translational speed is met, the Euler angles can be obtained by measuring the components of the accelerometer readings, which then correspond solely to $\vec{g}$. However, while this steady-flight condition has not yet been met, a reasonable estimation of the attitude can only be obtained if the air drag force is considered in the sum of all forces and moments acting on the aircraft.

With respect to the real system built by Dvořák [2011], which the identification hereby made relates to, a discrepancy was identified in the armature current i_a. According to the data sheet of the PJS 3D 550E motor, its maximum value should be 8.5 A, however, applying $u_{\max} = 12.6\,\text{V}$ to the armature in the identified rotor model yields $\approx 20\,\text{A}$. This suggests that the rotor parameters do not correspond to the real system's ones, even though their combination produces equivalent results in term of outputs (thrust, torque and angular speed). Therefore the results of this work should still remain valid for immediate application to the real aircraft. Yet, before doing so, proper discretization of the controllers hereby calculated should be done, among other practical implementation considerations.

Another suggestion of future work to be done at this project is to further experiment with the controllers, especially in terms of disturbance rejection. During the simulations it was verified that all controllers designed with the different techniques coped well, at least up to some point, in rejecting disturbances on the output of the control loops, but the limits of disturbance rejection were not determined. This would be an important issue to consider before employing the controllers in practical applications with the real aircraft in environments where, for example, wind is present.

Also, one very important issue when applying the designed controllers to the real system is the stability, which is not ensured for any operational point, as it is in the case of the linearized model. However, one way of augmenting the stability bounds of the non-linear system is by employing the *gain scheduling* technique, either to the P-controllers of the nested architecture of Fig. 6.2 as well as to the gains of the LQ in the scheme of Fig. 6.7. By establishing a *scheduling grid*, one can tune these controllers at each grid point and then use some interpolation to find an *adaptive* controller for the given operational point, with which one expects to obtain better results in terms of stability and even performance.

One last improvement suggestion for future work is the fragmentation of the LQ-control structure in Fig. 6.7 into three independent control loops: XY-position, altitude h and heading ψ. This control loop segmentation is quite straightforward to be implemented when considering the linearization around the hovering condition.

Bibliography

Draganfly Innovations Inc. `www.draganfly.com`.

Microdrones GmbH. `www.microdrones.com`.

Parrot SA. `www.parrot.com`.

Brian D.O. Anderson and John B. Moore. *Optimal Filtering.* Dover Publications, 1979.

S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, page 4393–4398, 2004a. ISBN 0780382323.

Samir Bouabdallah, André Noth, and Roland Siegwart. PID vs LQ control techniques applied to an indoor micro quadrotor. Technical report, Swiss Federal Institute of Technology, 2004b.

P. Castillo, A. Dzul, and R. Lozano. Real-Time stabilization and tracking of a Four-Rotor mini rotorcraft. *IEEE Transactions on Control Systems Technology*, 12(4):510–516, 2004. ISSN 1063-6536. doi: 10.1109/TCST.2004.825052. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1308180`.

Pedro Castillo, Rogelio Lozano, and Alejandro Dzul. Experimental implementation of linear and nonlinear control laws. *IEEE Control Systems Magazine*, page 11, December 2005.

Jaromír Dvořák. Micro quadrotor: Design, modelling, identification and control. Master's thesis, Czech Technical University in Prague, May 2011.

T. Hamel, R. Mahony, R. Lozano, and J. Ostrowski. DYNAMIC MODELLING AND CONFIGURATION STABILIZATION FOR AN X4-FLYER. *IFAC 15th Triennial World Congress*, page 6, 2002.

G. M Hoffmann, H. Huang, S. L Waslander, and C. J Tomlin. Quadrotor helicopter flight dynamics and control: Theory and experiment. Technical report, American Institute of Aeronautics and Astronautics, 2007.

T. Kim, K. Stol, and V. Kecman. Control of 3 DOF quadrotor model. *Robot Motion and Control 2007*, page 29–38, 2007.

Ramu Krishnan. *Permanent Magnet Synchronous and Brushless DC Motor Drives.* CRC Press, 1st edition, 2010.

Frank L. Lewis and Vassilis L. Syrmos. *Optimal Control.* John Wiley & Sons, 2nd edition, 1995.

Vicente Martínez Martínez. Modelling of the flight dynamics of a quadrotor helicopter. Master's thesis, Cranfield University, September 2007.

P. Pounds, R. Mahony, J. Gresham, P. Corke, and J. Roberts. Towards dynamically-favourable quad-rotor aerial robots. In *Australian Conference on Robotics and Automation*, page 10, 2004.

Paul Pounds, Robert Mahony, and Peter Corke. Modelling and control of a Quad-Rotor robot. In *Australian Conference on Robotics and Automation*, page 10, 2006.

Sigurd Skogestad and Ian Postlethwaite. *Multivariable Feedback Control Analysis and Design.* John Wiley & Sons, 2^{nd} edition, 2005.

Michael J. Stepaniak. *A Quadrotor Sensor Platform.* PhD thesis, Ohio University, November 2008.

Brian L. Stevens and Frank L. Lewis. *Aircraft Control and Simulation.* John Wiley & Sons, 2^{nd} edition, 2003.

A. Tayebi and S. McGilvray. Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006. ISSN 1063-6536. doi: 10.1109/TCST.2006.872519. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1624481`.

Holger Voos. Nonlinear control of a quadrotor Micro-UAV using Feedback-Linearization. page 1–6, Málaga, Spain, April 2009.

Appendix A

Simulink Diagrams

Here the most relevant Simulink diagrams are shown, for illustration purposes.

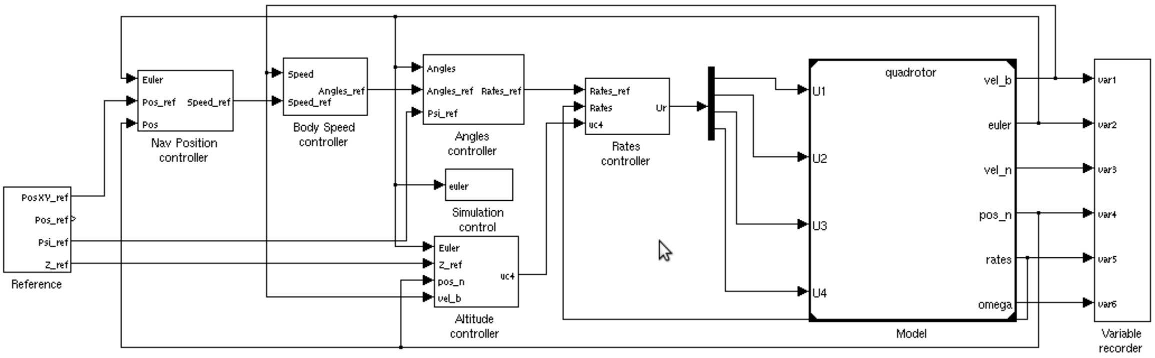


Figure A.1: Simulink diagram of nested PID control applied to the quadrotor's non-linear model.

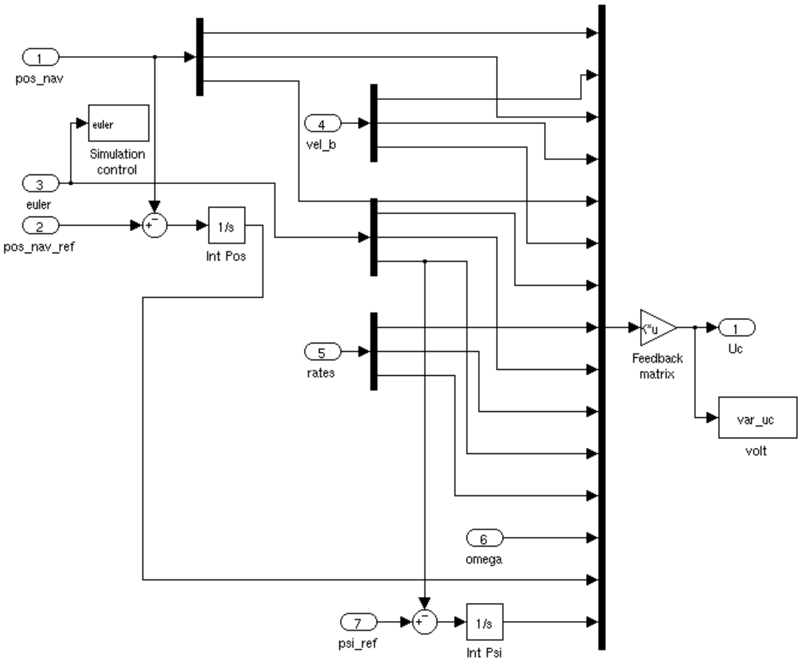


Figure A.2: Simulink diagram of LQ control state-feedback assembly.

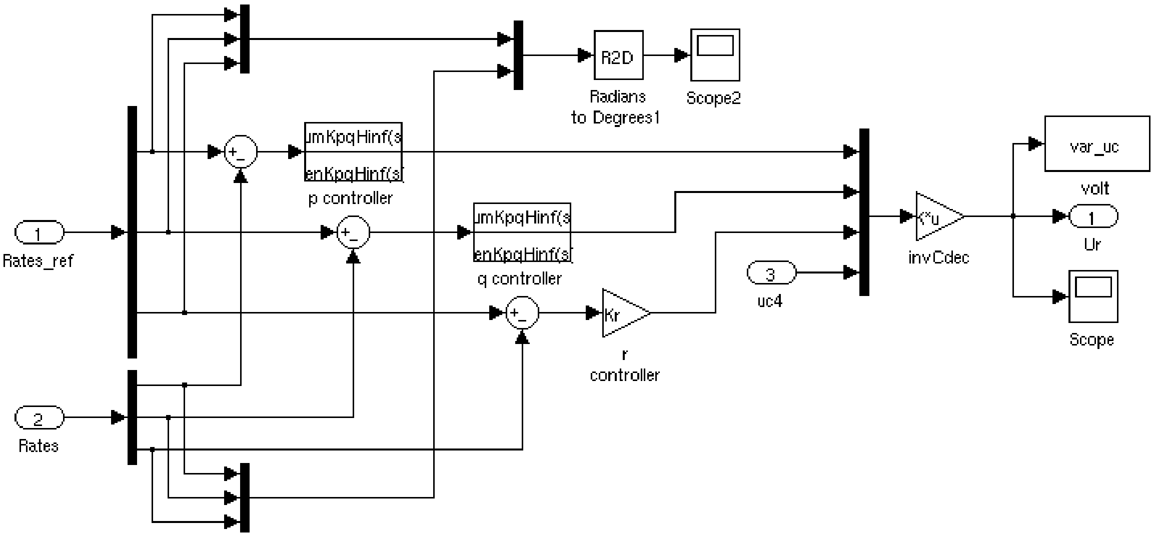


Figure A.3: Simulink diagram of MS $\mathcal{H}_\infty$ controller for p and q in the $\vec{\Omega}$ loop.

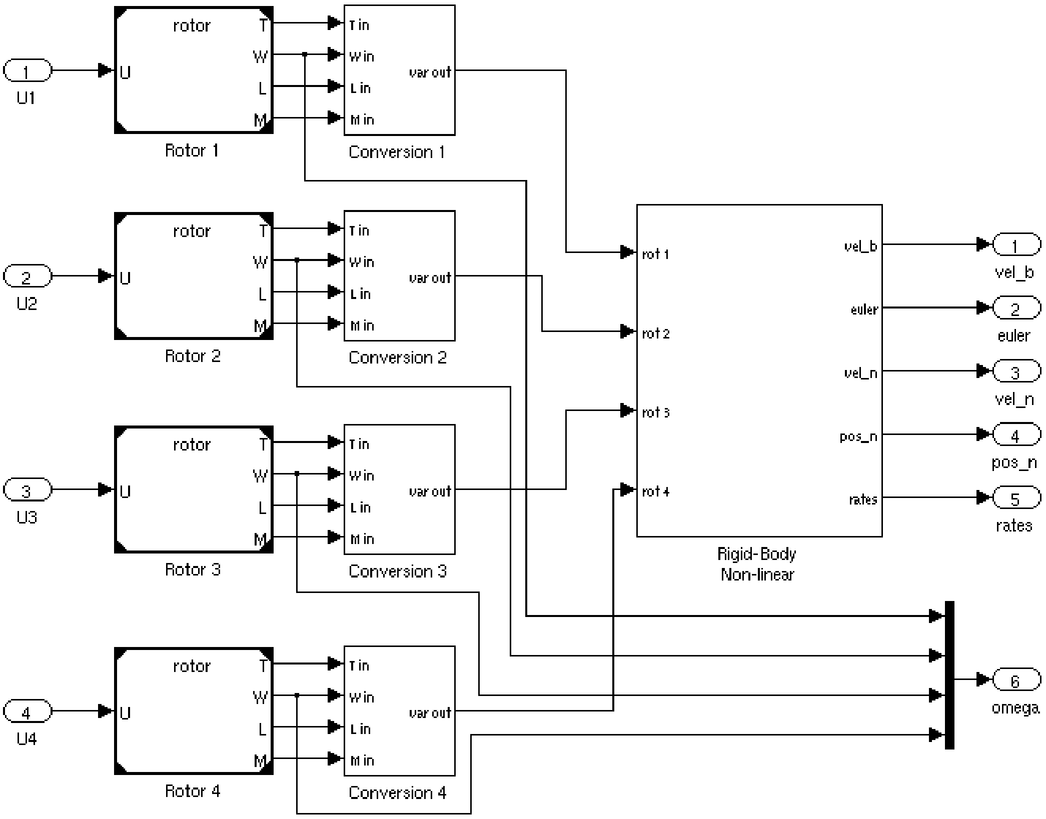


Figure A.4: Simulink diagram of quadrotor's non-linear model.

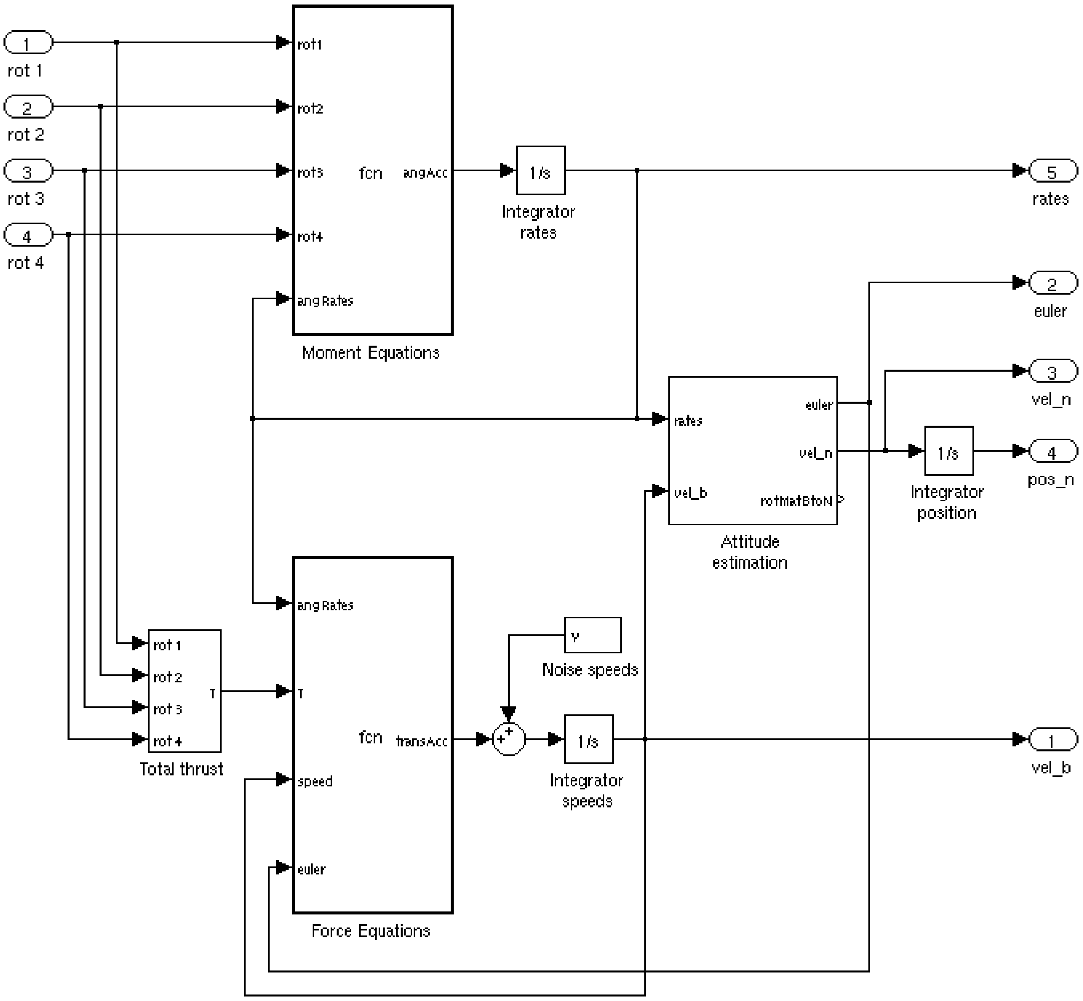


Figure A.5: Simulink diagram of quadrotor's non-linear model rigid body dynamics.

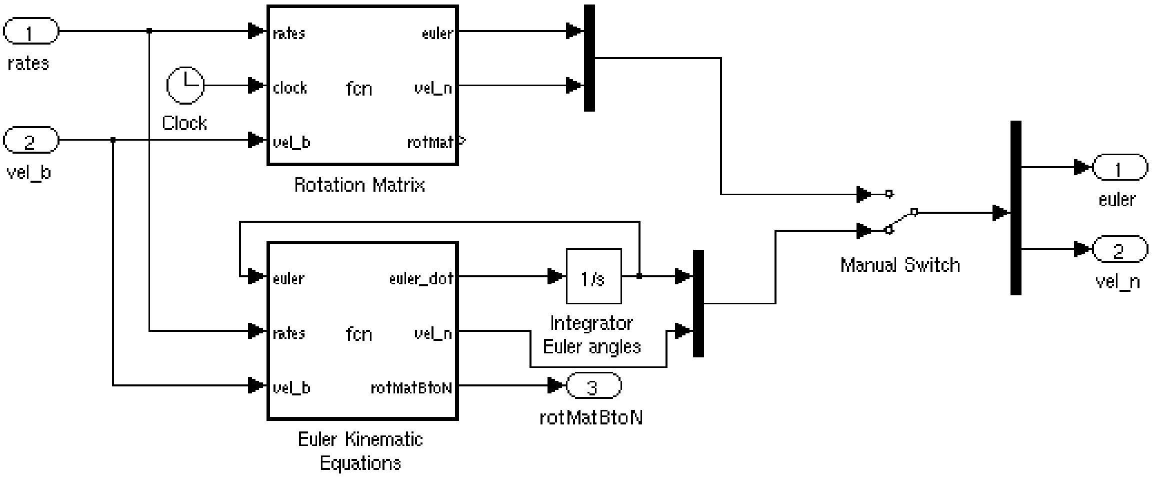


Figure A.6: Simulink diagram of quadrotor's attitude estimation.