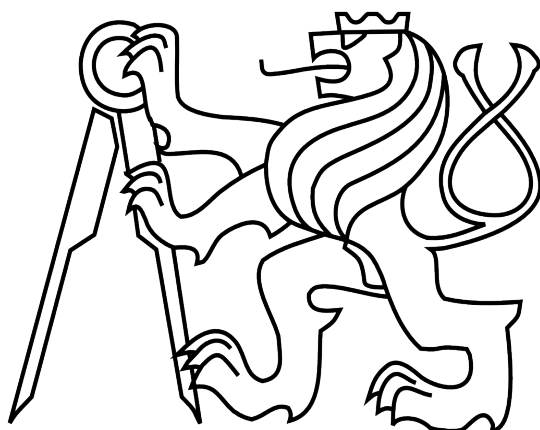


CZECH TECHNICAL UNIVERSITY IN PRAGUE

FACULTY OF ELECTRICAL ENGINEERING

DEPARTMENT OF CYBERNETICS



## **MASTER'S THESIS**

**Stabilization and control of unmanned quadcopter**

Prague, May 30, 2011

Tomáš Jiřinec



Space Master is a Joint European  
Master in Space Science and  
Technology.



This thesis is supported by the  
European Space Agency.



Agent Technology Center,  
Department of Cybernetics,  
Faculty of Electrical Engineering,  
Czech Technical University,  
Prague



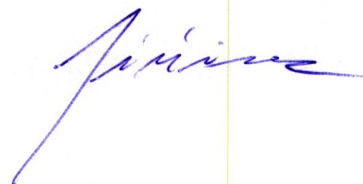
Luleå University of Technology,  
the main partner of the  
Space Master consortium.

## **Declaration by candidate**

I hereby declare that this thesis is my own work and effort and I have not used any sources or resources other than stated. I further declare that I have clearly indicated all direct and indirect quotations.

Prague, May 30, 2011

Tomáš Jiřinec



## **Acknowledgments**

First of all I would like to thank my family which supported me during all my studies, without them I would have had much more difficulties finishing my education.

Then to my supervisor Ing. Milan Rollo, Ph.D. whose support was instrumental for this thesis and who was available to me all the time. My gratitude goes also to my teachers Ing. Martin Hromčík, Ph.D. and Ing. Tomáš Haniš for their valuable advices and help.

I would also like to thank all my friends who supported me and made me feel happy especially during the difficult times.



## Abstract

Recent development in the fields of MEMS sensors, miniature, energy efficient and very powerful microcontrollers and microprocessors has given the opportunity to build small autonomous flying vehicles.

This thesis is studying one type of these vehicles, so called quadrotor. The quadrotor is flying vehicle similar to helicopter but having four rotors which are situated in one plane. Each propeller is actuated by an independently controlled motor. This feature allows to control Euler angles of the quadrotor such as yaw, pitch and roll thus giving the possibility to control quadrotor's position and velocity.

The CTU Department of Cybernetics bought such a quadrotor prototype called Linkquad. The goal is to use this prototype for indoor flights using camera vision as a navigation reference. Unfortunately the prototype has no autonomous control loops implemented.

The focal point of this thesis is design of a control law for the already mentioned prototype. This leads to other tasks such as an identification of the models 's parameters, creating mathematical nonlinear model, linearization of this model in certain trim points, design and testing of the linear controller and finally implementation and testing using the real system.

Czech Technical University in Prague  
Faculty of Electrical Engineering

Department of Cybernetics

## DIPLOMA THESIS ASSIGNMENT

Student: **Bc. Tomáš Jiřinec**

Study programme: Cybernetics and Robotics  
Specialisation: Systems and Control

Title of Diploma Thesis: **Stabilization and control of unmanned quadcopter**

### Guidelines:

1. Analyze the hardware equipment of the LinkQuad aerial vehicle
2. Identify system parameters
3. Create a linear model of the system
4. Design a state regulator for given model
5. Implement the designed regulator
6. Experimentally evaluate features of the regulator

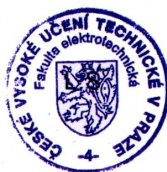
### Bibliography/Sources:

Will be provided by the supervisor

Diploma Thesis Supervisor: Ing. Milan Rolló, Ph.D.

Valid until summer semester 2011/2012

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



  
prof. Ing. Boris Šimák, CSc.  
Dean

Prague, December 14, 2010



## Table of Contents

Picture Index.....	vii
Figure Index.....	viii
Table Index.....	ix
Table of notations.....	x
1 Introduction.....	1
1.1 Motivation.....	1
1.2 Contribution.....	1
1.3 Structure of this thesis.....	1
2 Modeling the quadrotor.....	3
2.1 Overview.....	3
2.2 Equations of motion – basic derivation.....	3
2.2.1 Direction cosine matrix.....	4
2.2.2 Angular rates transformation.....	5
2.2.3 Linear acceleration.....	6
2.2.4 Angular acceleration.....	8
2.3 Expanding the equations of motion.....	10
2.3.1 Gyroscopic moments of the propellers.....	10
2.3.2 Engine dynamics.....	11
3 Control techniques overview.....	12
3.1 PID control.....	12
3.2 LQR control.....	12
3.3 H infinity control .....	13
3.4 Nonlinear control.....	13
4 Identification of the quadrotor.....	14
4.1 Engine and propeller.....	14
4.1.1 Moment of inertia of the propeller.....	14
4.1.2 Engine dynamics identification, introduction.....	17
4.1.3 Measurement of the engine's angular rate.....	17
4.1.4 Engine dynamics identification, measurement and results.....	20
4.2 Calculating the moment of inertia of the quadrotor.....	22
4.3 Aerodynamic coefficients identification.....	24
4.3.1 Thrust coefficient.....	24
4.3.2 Drag coefficient.....	25
5 Nonlinear mathematical model.....	29
5.1 6DOF rigid body subsystem.....	31
5.2 Gravity subsystem.....	31
5.3 Engines subsystem.....	31
5.4 FlightGear subsystem.....	32
6 Designing the control loops.....	34
6.1 Firmware limitations.....	34
6.2 Control strategy.....	34
6.3 Channel Mixer.....	36
6.4 Linearized state equations.....	37
6.5 Control system design.....	38

6.5.1 The vertical movement.....	39
6.5.1.1 Linear controllers' design.....	40
6.5.1.2 Verification on nonlinear model.....	45
6.5.2 The yawing movement.....	48
6.5.2.1 Linear controller design.....	49
6.5.2.2 Verification on nonlinear model .....	53
6.5.3 The longitudinal movement.....	55
6.5.3.1 Linear controller design.....	56
6.5.3.2 Verification on nonlinear model .....	64
6.5.4 The lateral movement.....	66
6.6 Flight simulation.....	68
7 Testing using the real model.....	73
7.1 Linkboard configuration.....	75
7.1.1 Mixer configuration.....	75
7.1.2 Inner loops configuration.....	77
7.1.3 Flight modes – Outer loops.....	80
7.2 Flight test.....	82
7.2.1 Indoors tests.....	83
7.2.2 Outdoor test.....	84
8 Conclusions and future work.....	87
9 Enclosed CD's content.....	89
10 Bibliography.....	90

## Picture Index

Picture 2.1: Top view of the quadrotor, engine numbering and their sense of rotation.....	3
Picture 2.2: Body and inertial frame comparison. taken from (Kıvrak, A., Ö. 2006).....	4
Picture 4.1: Propeller divided into three parts.....	15
Picture 4.2: Parallel axis theorem schematic.....	16
Picture 4.3: Thin plate rotating around its side schematic.....	16
Picture 4.4: Custom optical incremental encoder, side view.....	18
Picture 4.5: Custom optical incremental encoder, top view.....	18
Picture 4.6: Arduino Duemilanove board.....	19
Picture 4.7: Dividing the quadrotor into reasonable small parts.....	22
Picture 4.8: Loading mass attached to the motor.....	27
Picture 5.1: Nonlinear model of the quadrotor.....	30
Picture 5.2: Engine subsystem from the nonlinear model.....	32
Picture 5.3: FlightGear subsystem from the nonlinear model.....	33
Picture 5.4: FlightGear visualization demonstration.....	33
Picture 6.1: Control loops schematic.....	35
Picture 6.2: Channel mixer schematic.....	36
Picture 6.3: Vertical speed control loop.....	40
Picture 6.4: Vertical position control loop.....	42
Picture 6.5: Yaw rate control loop.....	50
Picture 6.6: Yaw angle control loop.....	52
Picture 6.7: Pitch rate control loop.....	57
Picture 6.8: Pitch angle control loop.....	59
Picture 6.9: Longitudinal speed control loop.....	61
Picture 6.10: Longitudinal position control loop.....	62
Picture 6.11: Complete control scheme used in the simulation.....	69
Picture 7.1: Mixer configuration in LinkGS.....	75
Picture 7.2: Roll rate controller configuration in LinkGS.....	77
Picture 7.3: Yaw rate controller configuration in LinkGS.....	78
Picture 7.4: Vertical speed feedforward setting in LinkGS.....	79
Picture 7.5: Roll angle controller configuration in LinkGS.....	80
Picture 7.6: Pitch angle controller configuration in LinkGS.....	81
Picture 7.7: Yaw angle controller configuration in LinkGS.....	82

## Figure Index

Figure 4.1: RPS measurement example.....	19
Figure 4.2: Comparison of the real and modeled motor dynamics.....	20
Figure 4.3: Comparison of the DC gains.....	21
Figure 4.4: Comparison of the proper-less motor dynamics.....	28
Figure 6.1: Root locus and bode plots for the vertical speed loop.....	41
Figure 6.2: Step response of the vertical speed loop.....	41
Figure 6.3: Root locus and bode plots for the vertical position loop.....	43
Figure 6.4: Step response of the vertical position loop.....	44
Figure 6.5: Comparison of the P and LQ controller using linear system.....	45
Figure 6.6: Vertical speed step responses comparison, linear and nonlinear plant control.....	46
Figure 6.7: Vertical position responses comparison, linear and nonlinear plant control.....	46
Figure 6.8: Vertical position step responses comparison, nonlinear plant, different steps, P controller..	47
Figure 6.9: Vertical position step responses comparison, nonlinear plant, different steps, LQ controller.	47
Figure 6.10: Vertical position step responses comparison with the maximum vertical speed limit.....	48
Figure 6.11: Root locus and bode plots for the yaw rate loop.....	50
Figure 6.12: Step response of the yaw rate loop.....	51
Figure 6.13: Root locus and bode plots for the yaw angle loop.....	52
Figure 6.14: Step response of the yaw angle loop.....	53
Figure 6.15: Yaw rate step responses comparison, linear and nonlinear plant control.....	54
Figure 6.16: Yaw angle step responses comparison, linear and nonlinear plant control.....	54
Figure 6.17: Root locus and bode plots for the pitch rate loop.....	57
Figure 6.18: Step response of the pitch rate loop.....	58
Figure 6.19: Root locus and bode plots for pitch angle loop.....	59
Figure 6.20: Step response of the pitch angle loop.....	60
Figure 6.21: Root locus and bode plots for the longitudinal speed loop.....	61
Figure 6.22: Step response of the longitudinal speed loop.....	62
Figure 6.23: Root locus and bode plots for longitudinal position loop.....	63
Figure 6.24: Step response of the longitudinal position loop.....	63
Figure 6.25: Pitch rate step responses comparison, linear and nonlinear plant control.....	64
Figure 6.26: Pitch angle step responses comparison, linear and nonlinear plant control.....	65
Figure 6.27: Longitudinal speed step responses comparison, linear and nonlinear plant control for different steps.....	65
Figure 6.28: Longitudinal position step responses comparison, linear and nonlinear plant control for different steps.....	66
Figure 6.29: Quadrotor's trajectory, projection on the xy plane.....	70
Figure 6.30: Quadrotor's trajectory, projection on the xz plane.....	71
Figure 6.31: Quadrotor's position in time.....	71
Figure 6.32: Quadrotor's speed in Earth's frame in time.....	72
Figure 6.33: Quadrotor's attitude in time.....	72
Figure 6.34: Quadrotor's heading in space, projection on xy plane.....	73
Figure 7.1: Roll channel's response in time.....	83
Figure 7.2: Pitch channel's response in time.....	83
Figure 7.3: Yaw channel's response in time.....	84

Figure 7.4: Roll channel's response in time.....	85
Figure 7.5: Pitch channel's response in time.....	85
Figure 7.6: Yaw channel's response in time.....	86

## Table Index

Table 4.1: Propeller's parts properties.....	15
Table 4.2: List of the parts and their parameters.....	23
Table 6.1: Comparison of the controllers' gains.....	68
Table 6.2: Mission plan.....	70
Table 7.1: Mixer configuration.....	76



## Table of notations

In this thesis there is a large number of symbols used. The meaning of each symbol is usually clear from the context. The most used symbols and their meanings are listed below.

$b$	Thrust constant
$d$	Drag factor of the rotating propeller
$CG$	Center of gravity
$D$	Direction cosine matrix
$g$	Acceleration due to gravity
$I_x$	Moment of inertia in roll
$I_y$	Moment of inertia in pitch
$I_z$	Moment of inertia in yaw
$l$	Lever length
$m$	Mass of the quadrotor
$mm$	Mixing ratio for the yaw channel
$n$	Mixing ratio for the pitch channel
$o$	Mixing ratio for the roll channel
$p$	Roll rate
$q$	Pitch rate
$r$	Yaw rate
$T$	Thrust force
$u$	Longitudinal speed in body-fixed frame
$U_0$	Engine's input during the hover.
$v$	Lateral speed in body-fixed frame
$w$	Vertical speed in body-fixed frame
$x$	Longitudinal coordinate in Earth-fixed frame
$y$	Lateral coordinate in Earth-fixed frame
$z$	Lvertical coordinate in Earth-fixed frame

$\theta$	Pitch angle
$\phi$	Roll angle
$\psi$	Yaw angle
$\Omega_1$	Angular rate of first engines' propeller
$\Omega_2$	Angular rate of second engines' propeller
$\Omega_3$	Angular rate of third engines' propeller
$\Omega_4$	Angular rate of fourth engines' propeller
$\Omega_0$	Angular rate of the propeller during the hover.



# 1 Introduction

## 1.1 Motivation

As has been already stated in the abstract, this thesis is revolving around an unmanned flying vehicle called quadrotor. The aim of this thesis is to find a suitable mathematical model for such a device and then develop a complete control architecture which will allow the quadrotor to fly autonomously. This feature can be then exploited by some higher level planning algorithms that can use this UAV, or more of them, for observation and scouting missions for civilian or even military personnel.

## 1.2 Contribution

This thesis is focusing on identification of parameters of the already built quadrotor, development of mathematical model of this UAV and then design of the linear controllers for this system. The main contributions are:

- Identified nonlinear mathematical model. This model can be further used as the simulation and development testbed for modified or new controllers.
- Linear models of the quadrotor, This systems can be further exploited for development of new linear controllers.
- Development of the controllers which enables the quadrotor to be flown and easily controlled.

## 1.3 Structure of this thesis

The thesis begins with the mathematical modeling of the quadrotor and the derivation of its equations of motion. This is the the content of the following chapter 2 .

Consequently a brief overview of the already developed controllers and control techniques is presented in chapter 3 .

The chapter 4 describes the procedure of identifying the physical properties of the quadrotor.

The chapter 5 introduces the nonlinear mathematical model created in the Simulink which is further used for simulations and testing.

In the chapter 6 the linear models of the quadrotor are derived and various controllers are designed and tested.

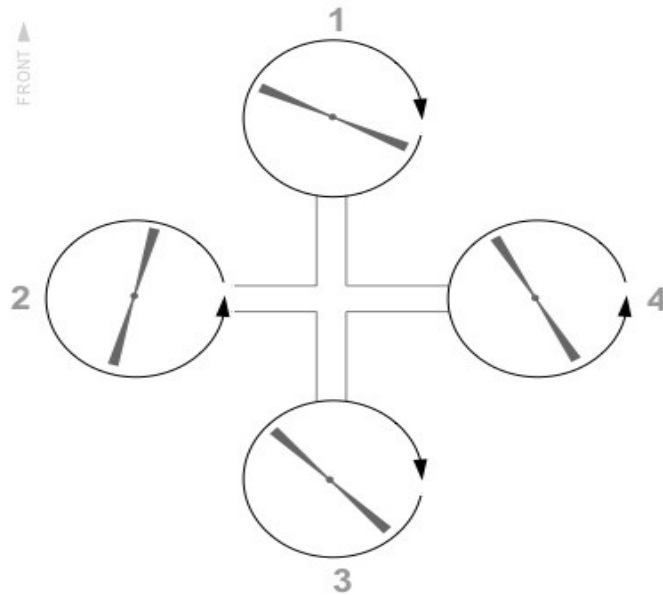
The chapter 7 describes the problematic of implementing the controllers into the real model and shows the results of the performed tests using the real quadrotor.

Finally conclusions and considerations are made in the chapter 8 .

## 2 Modeling the quadrotor

### 2.1 Overview

As it has been already said, the quadrotor has four rotors located in one plane.



*Picture 2.1: Top view of the quadrotor, engine numbering and their sense of rotation.*

When all of the rotors have same speed, then the overall moment produced is zero and the quadrotor is holding its attitude. When an appropriate speed is set up, then the rotors provide balancing thrust against the gravitational force hence the quadrotor holds its altitude. The position when attitude and altitude are kept is called hover.

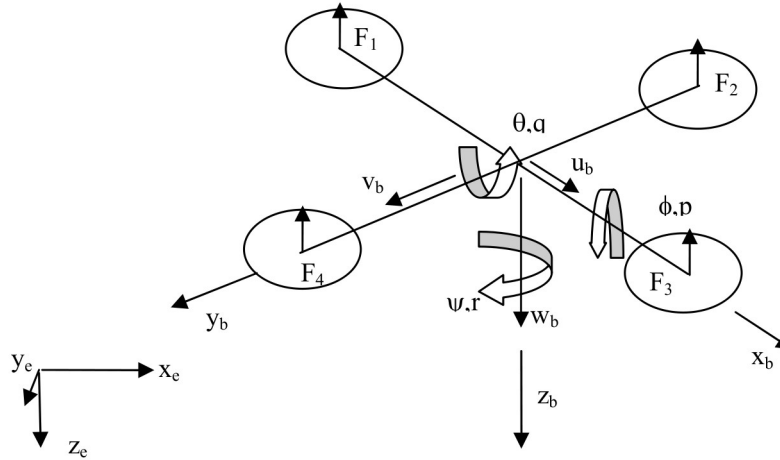
### 2.2 Equations of motion – basic derivation

The quadrotor is a system with 6 DOF, hence it is described using twelve states. The first six states represents the attitude and its change. These are the variables  $\phi$ ,  $\theta$  and  $\psi$  (roll, pitch and yaw angles are Euler angles between body-fixed frame and the Earth-fixed frame) and  $p$ ,  $q$ , and  $r$  (roll rate, pitch rate and yaw rate with respect to the body-fixed frame). The sense of the Euler angles is determined using the right hand rule.

The other six states are positions with respect to the Earth  $x$ ,  $y$  and  $z$  described in North-East-

Down (NED) coordinate system and the speeds in the body frame  $u$ ,  $v$ , and  $w$ .

Any movement of the quadrotor is achieved by changing the angular rates of its rotors. Increasing and decreasing the speed of all rotors leads to ascend and descend. Yaw rate (rotation around  $z$  axis) is obtained by changing speed of pair (1,3) or (2,4). Pitch rate (rotation around  $y$  axis) is achieved when the speed balance of rotors 1 and 3 is changed. Change in pitch angle then leads to longitudinal acceleration. Roll rate is obtained in similar way as pitch rate, the only difference is that rotors 2 and 4 are used instead rotors 1 and 3. Changing the roll angle leads to lateral acceleration.



Picture 2.2: Body and inertial frame comparison. taken from (Kivrak, A., Ö. 2006).

Because of the existence of two different coordinates frames one needs the transformation matrix between these two systems.

### 2.2.1 Direction cosine matrix

Vector rotation around axis  $x$  can be described as matrix

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (2.1)$$

Around axis  $y$  as

$$R_y = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.2)$$

And around axis  $z$  as

$$R_z = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

Then the consequent rotations around  $z \rightarrow y \rightarrow x$  axes can be described by the direction cosine matrix

$$D = R_x R_y R_z = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (2.4)$$

The  $D$  matrix describes the transformation Earth-fixed coordinates  $\rightarrow$  body-fixed coordinates.

The first set of state equations is describing the change of position according to quadrotor's attitude in its velocity measured in the body frame:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = D^{-1} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.5)$$

### 2.2.2 Angular rates transformation

The transformation between angular rates in Earth-fixed frame to body-fixed frame is given by equation (Cook 1997, p.23)

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = E \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \quad (2.6)$$

where



$$E = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (2.7)$$

Then the second set of state equations describing change of attitude according to rotation in the body frame is

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = E^{-1} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.8)$$

### 2.2.3 Linear acceleration

The linear acceleration in Earth-fixed frame is described by the Newton's Second Law

$$F = m \dot{V} \quad (2.9)$$

where  $m$  is the quadrotor's mass which is constant and  $V$  is the velocity vector in the body frame.

The speeds  $u, v$  and  $w$  are measured in body-fixed coordinates and the body frame velocity vector can rotate and change its magnitude at the same time. This leads to total derivative of vector  $V$  (Blakelock 1991, p.10)

$$F = m \dot{V} + \omega \times mV \quad (2.10)$$

Then

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + m \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.11)$$

After expanding the cross product and reorganizing

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.12)$$

Neglecting the aerodynamic forces then the external forces acting in the quadrotor's body are thrust of the propellers  $T$  and weight force  $W$ .

Thrust is always acting in the body  $z$  axis while the weight force is projected according the attitude of the quadrotor

$$\begin{bmatrix} W_x \\ W_y \\ W_z - T \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (2.13)$$

The weight force is always acting in the Earth's frame  $z$  axis. Conversion to body-fixed frame is done by the direction cosine matrix (2.4).

$$D \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} . \quad (2.14)$$

After reorganizing

$$\begin{aligned} \dot{u} &= rv - qw - g \sin \theta \\ \dot{v} &= pw - ru + g \cos \theta \sin \phi \\ \dot{w} &= qu - pv + g \cos \phi \cos \theta - \frac{T}{m} \end{aligned} \quad (2.15)$$

Considering no motor dynamics the thrust of all rotors is (thrust is proportional to the square of the propeller's angular rate (Hoffmann et al. 2007, p.8)

$$T = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (2.16)$$

Where  $b$  is a thrust coefficient and  $\Omega_i$  is speed of each rotor.

This leads to another set of state equations

$$\begin{aligned}\dot{u} &= rv - qw - g \sin \theta \\ \dot{v} &= pw - ru + g \cos \theta \sin \phi \\ \dot{w} &= qr - pv + g \cos \phi \cos \theta - \frac{b}{m} (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)\end{aligned}\quad (2.17)$$

## 2.2.4 Angular acceleration

Application of an external torque will change the angular momentum of the quadrotor

$$M = \dot{H} \quad (2.18)$$

But the angular momentum vector changes its direction hence the total derivative of vector  $H$  is applied (Blakelock 1991, p.13)

$$M = \dot{H} + \omega \times H \quad (2.19)$$

And

$$H = I \omega, \quad (2.20)$$

where  $\omega$  is the change of the attitude and  $I$  is the moment of inertia of the quadrotor.

The quadrotor is a rigid body symmetric about its  $xz$  and  $yz$  plane, and the rotation axes coincidences with the principal axes, then the moment of inertia tensor is

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \quad (2.21)$$

Then

$$M = I \dot{\omega} + \omega \times I \omega \quad (2.22)$$

After expanding

$$\begin{aligned}M_x &= \dot{p} I_x + qr (I_z - I_y) \\ M_y &= \dot{q} I_y + pr (I_x - I_z) \\ M_z &= \dot{r} I_z + pq (I_y - I_x)\end{aligned}\quad (2.23)$$

And because of the  $xz$  and  $yz$  symmetry

$$I_x \approx I_y, \quad (2.24)$$

the equations can be simplified to

$$\begin{aligned} M_x &= \dot{p} I_x + q r (I_z - I_y) \\ M_y &= \dot{q} I_y + p r (I_x - I_z) \\ M_z &= \dot{r} I_z \end{aligned} \quad (2.25)$$

The external torques are produced by the thrust and drag of the propellers. Neglecting the propeller's inertia and aerodynamic torques, then the external torques can be written as

$$\begin{aligned} M_x &= l b (\Omega_2^2 - \Omega_4^2) \\ M_y &= l b (\Omega_1^2 - \Omega_3^2) \\ M_z &= d (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{aligned}, \quad (2.26)$$

where  $d$  is the drag factor of the rotors and  $l$  is the distance of the propeller from the CG.

Then the last set of equations of motion is

$$\begin{aligned} \dot{p} &= \frac{l b}{I_x} (\Omega_2^2 - \Omega_4^2) - q r \frac{I_z - I_y}{I_x} \\ \dot{q} &= \frac{l b}{I_y} (\Omega_1^2 - \Omega_3^2) - p r \frac{I_x - I_z}{I_y} \\ \dot{r} &= \frac{d}{I_z} (\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2) \end{aligned} \quad (2.27)$$

## 2.3 Expanding the equations of motion

### 2.3.1 Gyroscopic moments of the propellers

The previous equation of motion are simplified. They do not take into account aerodynamic and gyroscopic forces and moments and the motor dynamics. In this chapter I am going to expand the already derived equations of motion by the terms associated with motor dynamics and gyroscopic moments.

Appending the gyroscopic moments to the moment equations leads to (Stepaniak 2008, p.65)

$$\begin{aligned} M_x &= \dot{p} I_x + qr(I_z - I_y) + \dot{H}_x + H_z q - H_y r \\ M_y &= \dot{q} I_y + pr(I_x - I_z) + \dot{H}_y + H_x r - H_z p \\ M_z &= \dot{r} I_z + \dot{H}_z + H_y p - H_x q \end{aligned} \quad (2.28)$$

where  $H_x, H_y, H_z$  are total angular momentums of spinning masses with angular rates in  $x, y$  and  $z$  direction in the body frame.

$$H_x = \sum_{i=1}^4 I_{xi} \omega_{xi} \quad (2.29)$$

$$H_y = \sum_{i=1}^4 I_{yi} \omega_{yi} \quad (2.30)$$

$$H_z = \sum_{i=1}^4 I_{zi} \omega_{zi} \quad (2.31)$$

The angular rates of the rotors are present only in the  $z$  axis (in the body frame) and there are no more rotating masses than them, the equations can be simplified to

$$\begin{aligned} M_x &= \dot{p} I_x + qr(I_z - I_y) + H_z q \\ M_y &= \dot{q} I_y + pr(I_x - I_z) - H_z p \\ M_z &= \dot{r} I_z + \dot{H}_z \end{aligned} \quad (2.32)$$

The state equations for the angular rates with the propellers' gyroscopic moments added are

$$\begin{aligned}
 \dot{p} &= \frac{lb}{I_x}(\Omega_2^2 - \Omega_4^2) - qr \frac{I_z - I_y}{I_x} - \frac{H_z}{I_x} q \\
 \dot{q} &= \frac{lb}{I_y}(\Omega_1^2 - \Omega_3^2) - pr \frac{I_x - I_z}{I_y} + \frac{H_z}{I_y} p \\
 \dot{r} &= \frac{d}{I_z}(\Omega_2^2 + \Omega_4^2 - \Omega_1^2 - \Omega_3^2)
 \end{aligned} \tag{2.33}$$

### 2.3.2 Engine dynamics

The motors propelling the quadrotor have their own dynamics. The equations of motions are the well known equations of motion of DC motor with the aerodynamic damping added.

$$\begin{aligned}
 L \frac{di}{dt} &= u - Ri - k_e \omega_m \\
 J_r \dot{\omega}_m &= k_i i - d_m \omega_m - f(\omega_m)
 \end{aligned} \tag{2.34}$$

Where

- $L$  - inductance of the coil in the engine
- $i$  – current flowing through the engine
- $u$  – voltage across the engine
- $R$  – resistance of the coil and wirings
- $k_e$  – back EMF constant
- $\omega_m$  - motor angular rate
- $d_m$  - bearing damping constant
- $J_r$  – moment of inertia of the rotor
- $k_i$  – torque constant
- $f(\omega_m)$  - nonlinear drag torque function for the given propeller

### **3 Control techniques overview**

Lot of controllers have been already developed for quadrotor system. In this chapter I am going to mention some of them and provide short summary.

#### **3.1 PID control**

(Bouabdallah et al. 2005) have used this controller to stabilize the attitude of the quadrotor around the hover position.

The controller was designed using linearized model of the quadrotor in the hover trim point. The controller was developed using the nonlinear Simulink model and it was verified on the physical system. The resulting controller was able to stabilize the physical system within three seconds.

The linearity of the controller constraints its use only around the hover trim point. Strong perturbation from this positions leads to loss of control.

(Hoffmann et al. 2007) have used PID control for controlling attitude, altitude and position. Results were satisfactory, but the quadrotor has not performed any aggressive maneuvers and the disturbance rejection of the control system was not very good.

#### **3.2 LQR control**

(Castillo et al. 2005) have implemented this kind of controller. During simulation the controller has performed satisfactory. When strong perturbation was introduced the controller due to its linearity was not able to stabilize the system. On the physical model, this controller was not able to stabilize the system at all.

(Bouabdallah et al. 2005) have implemented LQR controller using multiple trim points. Unfortunately they have not implemented the motor dynamics into the model. This lead to worse performance than their already mentioned PID controller.

### **3.3 *H infinity control***

(Chen & Huzmezan 2003) have used the  $H^\infty$  approach to design 2DOF controller combined with MPC (model predictive control) controller for position control. The simulation gave satisfactory results. The controller provided robust reference tracking and disturbance rejection.

(De Lellis, Marcelo 2011) has used the mixed sensitivity  $H^\infty$  and  $\mu$ -synthesis with DK iteration algorithms to obtain robustly stable controller with robust performance. The controllers were tested using the nonlinear Simulink mathematical model.

### **3.4 *Nonlinear control***

(Castillo et al. 2005) have used nonlinear control. Authors compared the nonlinear controller with linear LQR controller. The nonlinear controller has performed better when the perturbations were very high hence leading the LQR controller far away from its trim point.



## **4 Identification of the quadrotor**

To be able to find an appropriate control law the physical parameters of the real model have to be measured or estimated. This chapter is dealing with this problem and is divided into three parts: the engine and propeller subsystem identification, measurement of moment of inertia of the model's body and identification of the aerodynamic parameters of the propellers.

### ***4.1 Engine and propeller***

As stated in previous chapter this thesis considers the engine dynamics and gyroscopic effect of the propellers. This means the identification of the engine parameters and the moment of inertia of the propeller is needed.

#### **4.1.1 Moment of inertia of the propeller**

I was first considering to measure this parameter using a small already identified engine with the propeller attached to it and introducing some step changes at the engine input. Measuring the response of the angular rate would enable me to calculate the moment of inertia of the propeller. The engine from the quadrotor would do this task but the control electronics inside the quadrotor does not allow to feed the engines with small voltages. This means only higher voltages and bigger speeds are achievable. The increasing speed introduces the unknown drag force acting on the propeller making this kind of measurement very imprecise. Other engines which I had at my disposal were capable of running slowly thus avoiding the aerodynamic effects but these engines were too big and heavy thus the presence of the light propeller would have no significant influence on the step response of angular rate.

I decided to estimate the moment of inertia of the propeller using the direct approach. I used laboratory scales and measuring cylinder to measure the propeller's weight and density. Then I divided one half of the propeller into three parts.



Picture 4.1: Propeller divided into three parts.

Then I measured the value of these parts. Knowing the volume of these parts and the density of the propeller (homogeneous body) I calculated the weight of these parts. Knowing the mass, dimensions and the distance from the rotation axis of these parts I was able to calculate their moments of inertia which contributes to the overall moment of inertia of the whole propeller.

part	mass (g)	width (cm)	height (cm)	r (cm) (distance from the shaft)
I	1.7	3	2.5	9
II	2.6	4.4	2.8	5
III	1.2	5	1.2	0

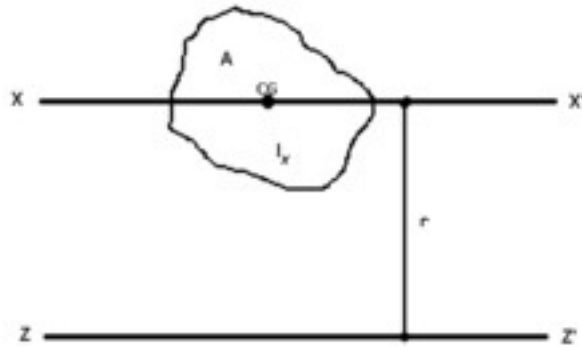
Table 4.1: Propeller's parts properties.

The moment of inertia of a thin rectangular plate of height  $h$  and of width  $w$  and mass  $m$  (axis of rotation in the CG) is (Serway & Jewett n.d., p.304)

$$I = m \left( \frac{h^2}{12} + \frac{w^2}{12} \right) \quad (4.1)$$

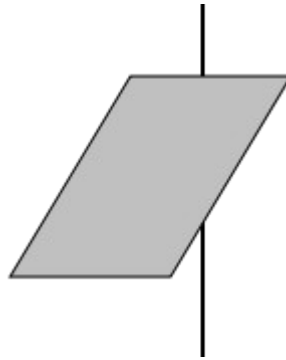
Using the parallel axis theorem it is possible to calculate the moment of inertia of the body of mass  $m$  with already known moment of inertia  $I_0$  around an arbitrary axis. The new rotation axis is at distance  $r$  from the original axis and is parallel to it.

$$I = I_0 + mr^2$$



Picture 4.2: Parallel axis theorem schematic.

When the axis of rotation is moved to the end of the plate



Picture 4.3: Thin plate rotating around its side schematic.

Then

$$I = I_0 + mr^2 = m \left( \frac{h^2}{12} + \frac{w^2}{12} \right) + m \frac{h^2}{4} = m \left( \frac{h^2}{3} + \frac{w^2}{12} \right) \quad (4.2)$$

Using the values from table and the equation (4.2) to calculate moment of inertia of each part, the moment of the inertia of the whole propeller is

$$I_p = 2(I_I + I_{II} + I_{III}) = 4.439 \cdot 10^{-5} \text{ kg m}^2 \quad (4.3)$$

### 4.1.2 Engine dynamics identification, introduction

In the chapter 2.3.2 I introduced equations describing a general DC electric engine. It is a set of two first order equation which means the overall system is of order two. The engines used in the quadrotor are very small thus having small inductance and back EMF constant. This means that I can discard the dynamics of the current without any bigger impact of the model precision. This leads to only one differential equation

$$0 = u - Ri$$

$$J_r \dot{\omega}_m = k_i i - d_m \omega_m - f(\omega_m) = -d_m \omega_m - f(\omega_m) + \frac{k_i}{R} u \quad (4.4)$$

A general LTI (linear time invariant) system of first order can be described by the following transfer function

$$G(s) = \frac{K}{(\tau s + 1)} \quad , \quad (4.5)$$

where  $K$  stands for the DC (steady-state) gain and  $\tau$  for time constant of the system.

The differential equations of the engine are nonlinear, which means that the linear transfer function will only be valid in close vicinity of an arbitrary trim point. The most convenient trim point is the hovering quadrotor.

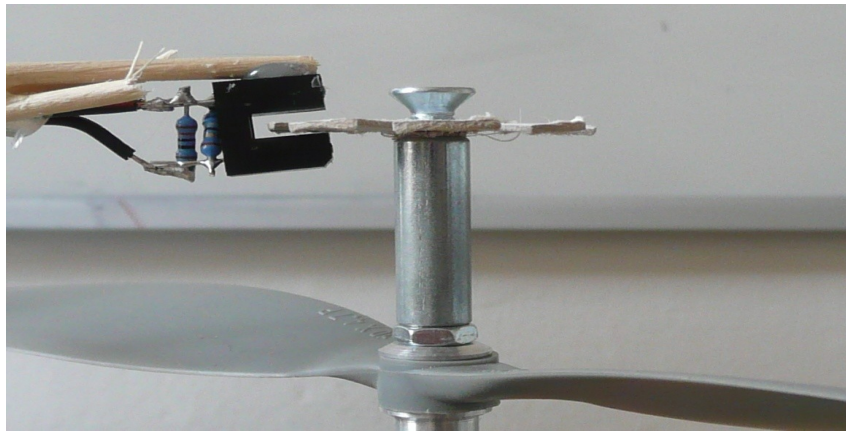
To find out the actual parameters of the transfer function I needed to measure the angular rate of the engine. To do this I have built a simple incremental optical sensor, which is described in the next chapter.

### 4.1.3 Measurement of the engine's angular rate

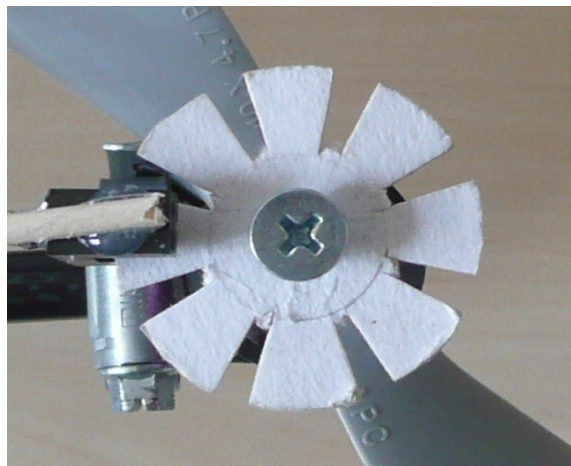
To do this measurement I decided to use the optical measurement approach which has no direct contact with the engine thus not influencing its dynamics at all. The conventional optical incremental sensors are expensive. On the other hand their precision is very high but for this application this precision is not needed.

I decided to build my own sensor. For that I used an paper circle with eight regularly spaced slits and an opto-interrupter. The circle is attached to the rotor and rotating with it. the opto-interrupter creates a

falling edge every time when there is a transition from the the solid part of the circle to the slit.



*Picture 4.4: Custom optical incremental encoder, side view.*

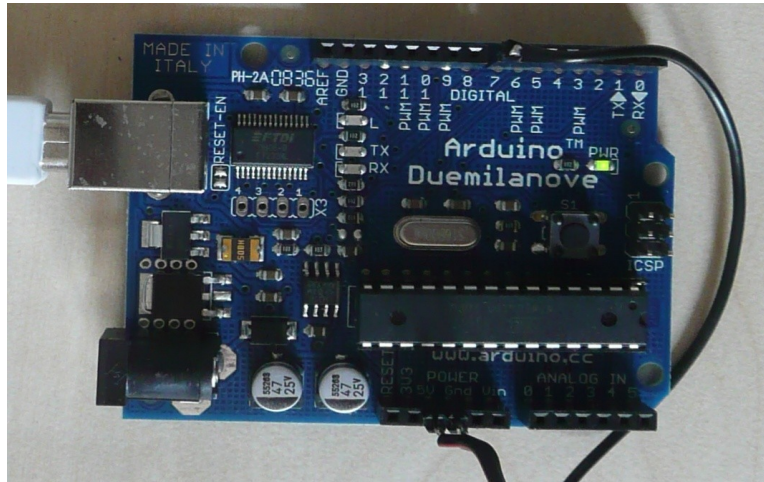


*Picture 4.5: Custom optical incremental encoder, top view*

The opto-interrupter is connected to the Arduino board which supplies it with 5V and processes the pulses. The Arduino Duemilanove is basically a board with ATmega168 MCU and some supporting facilities.

The MCU uses its comparator subsystem to convert the analog pulses to their digital representations. Then the capture functionality of TIMER1 is used to measure the clock ticks between each eight pulses (one rotation of the propeller). Then this value is coded into its ASCII representation and send via USB to the computer where it is logged to the CSV file. The capture function of TIMER1 is handled using an interrupt, hence it is not blocked by the sending process running in the main thread.

In the computer the CSV file is processed using MATLAB, which converts the clock ticks to the time, calculates and plots the RPS (rotations per second) values.



Picture 4.6: Arduino Duemilanove board.

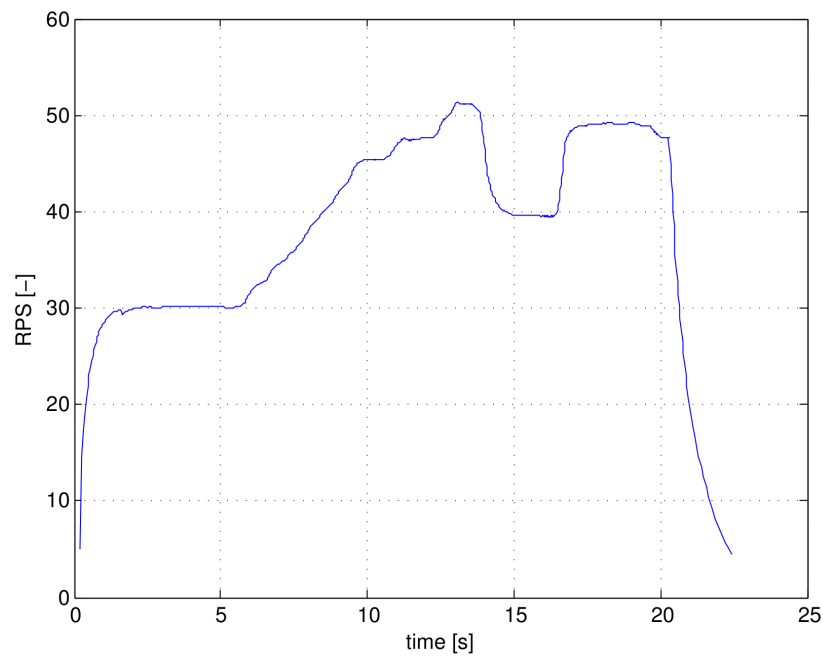
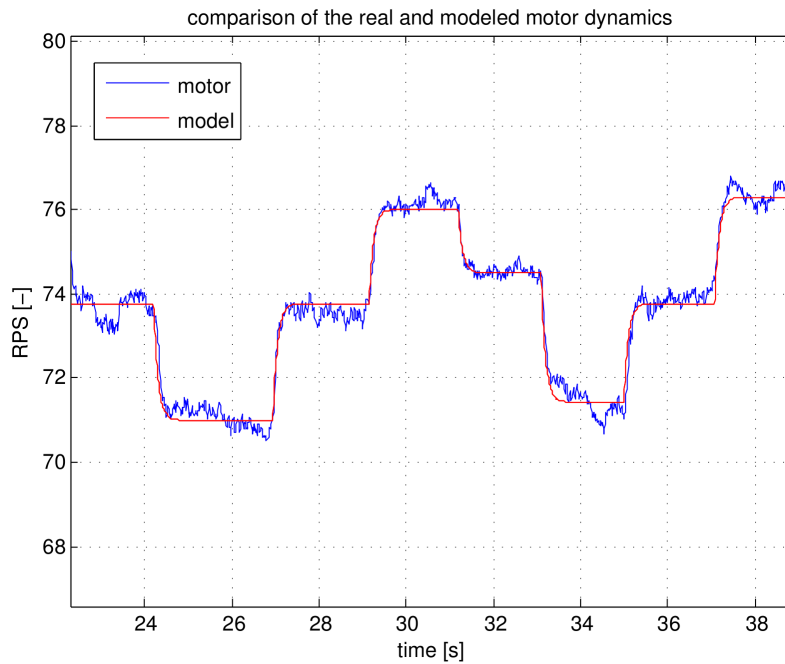


Figure 4.1: RPS measurement example.

#### 4.1.4 Engine dynamics identification, measurement and results

To identify the time constant of the motor it is necessary to find its trim point first. The trim point is its angular rate during the hover of the quadrotor. To find this values I manually piloted the quadrotor into the hover position and used the internal software to log the input to the engines (it was not viable to fly the quadrotor with opto-interrupter connected and running). With the knowledge of engines' input during the hover I connected the sensor and performed some steps around this value. After analyzing the responses from the engine I estimated the time constant of the motor to be  $\tau=0.1$  . The measured and estimated responses can be seen in the following figure 4.2.



*Figure 4.2: Comparison of the real and modeled motor dynamics.*

The last thing is finding the DC gain of the motor. For that I used the log from quadrotor again. In the log it is possible to find values referencing to the PPM signal sent to the motor. After analyzing the the signals I estimated the DC gain of the engine to be  $K=0.7$

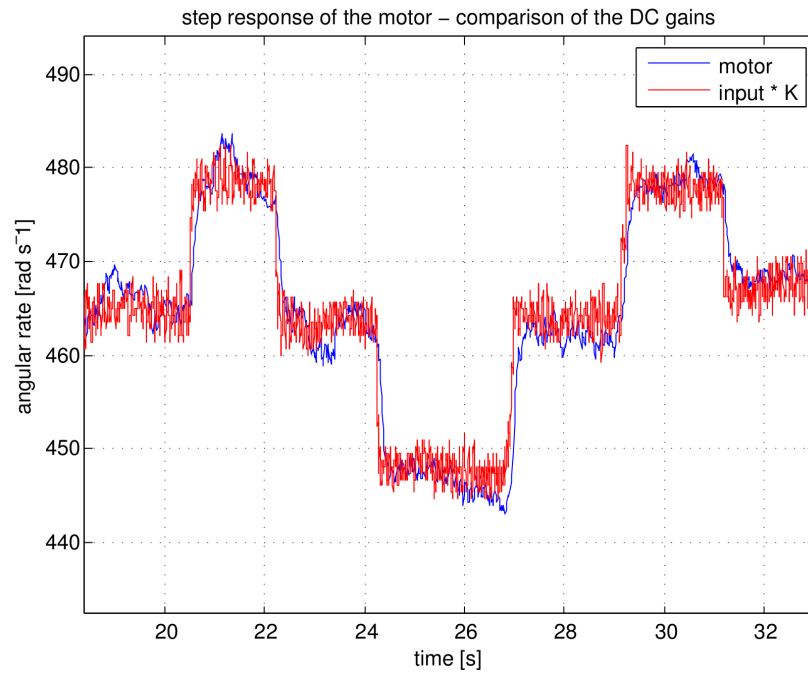


Figure 4.3: Comparison of the DC gains.

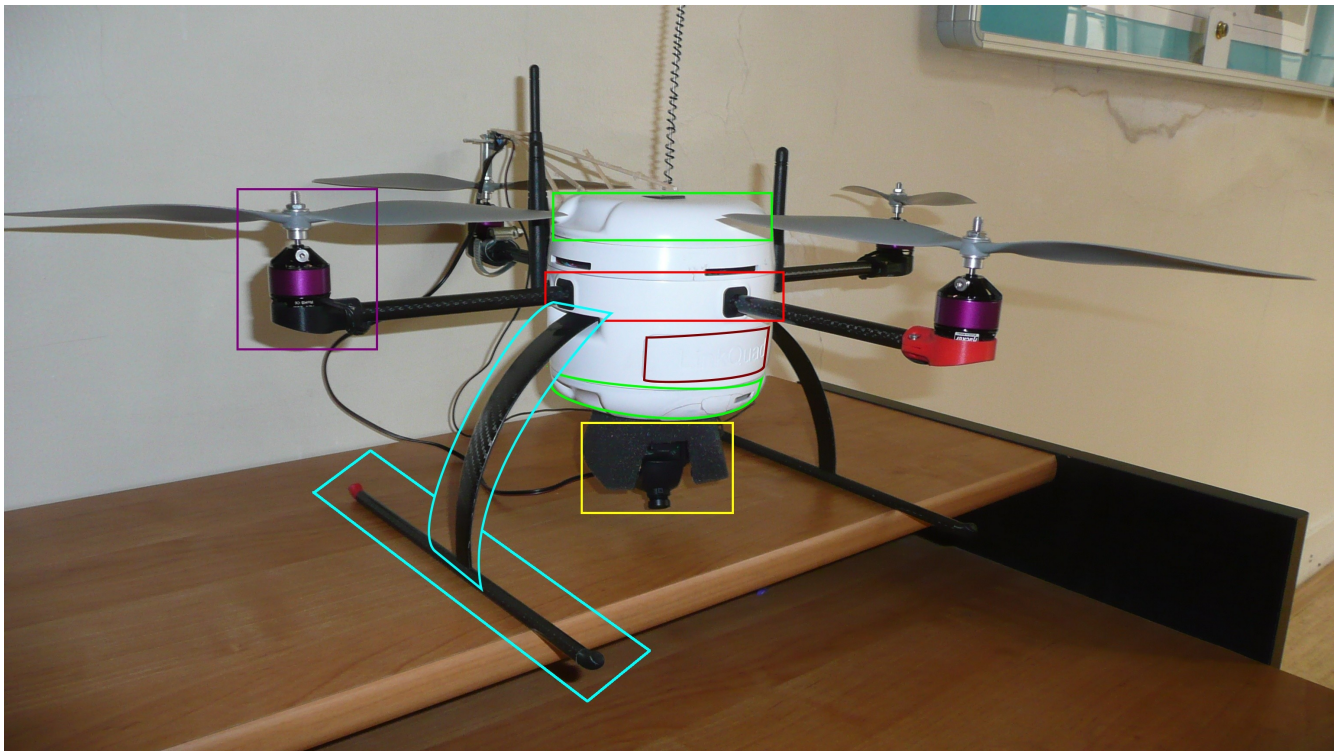
The overall transfer function of the motors from PPM to RPS in the hover trim point is

$$G_m(s) = \frac{0.7}{(0.1s + 1)} \quad (4.6)$$



## 4.2 Calculating the moment of inertia of the quadrotor

The moment of inertia of the quadrotor body is very important parameter which influences dynamics of nearly every moment. To measure it I decided to disassemble the model into “reasonable” small parts which have significant influence to the overall moment of inertia. Some parts cannot be easily dismantled, which introduces error into the calculation. Each part was then weighted and its dimensions and distance from the appropriate rotation axis was measured. Then I used the defined moments of inertia calculations for the defined body shapes (Serway & Jewett n.d., p.304) with conjunction with the parallel axis theorem to calculate their moments of inertia in each principal axes.



Picture 4.7: Dividing the quadrotor into reasonable small parts.









part	color	shape	mass [g]	dimensions [cm]*	distance from the axis [cm]**	moment of inertia [10 <sup>-3</sup> kgm <sup>2</sup> ]***
body		solid cylinder	374	$r=6.5$ $h=3$	[0 0 0]	$J_b=[0.423 \ 0.423 \ 0.789]$
top cover		solid cylinder	50	$r=6.5$	[3.5 3.5 0]	$J_{tc}=[0.114 \ 0.114 \ 0.106]$
bottom cover		cylindrical tube	53	$r_1=6.5$ $r_2=4$ $h=2$	[5.5 5.5 0]	$J_{bc}=[0.239 \ 0.239 \ 0.026]$
battery		cuboid	394	$w=9.5$ $d=7$ $h=2.5$	[3 3 0]	$J_{bat}=[0.536 \ 0.671 \ 0.457]$
autopilot board	+	cuboid	41	$w=8$ $d=4.5$ $h=1$	[1.5 1.5 0]	$J_{ap}=[0.016 \ 0.031 \ 0.029]$
leg's arm		rectangular plate	18	$w=13$ $h=2$	[1 1 1]	$J_{arm}=[0.069 \ 0.029 \ 0.029]$
leg's ski		rod	9	$l=27.5$	[19.5 14.3 13.5]	$J_{ski}=[0.336 \ 0.236 \ 0.217]$
camera		point mass	59		[10 10 0]	$J_{cam}=[0.59 \ 0.59 \ 0]$
motor		point mass	72		[21.4 21.4 21.4]	$J_m=[6.6 \ 6.6 \ 13.2]^{++}$

Table 4.2: List of the parts and their parameters.

\* -  $r$  – radius,  $r_1$  – inner radius,  $r_2$  – outer radius,  $h$  – height,  $w$  – width,  $d$  - depth,  $l$  - length

\*\* - Axes of rotation are in order  $x \ y \ z$

\*\*\* - Moment of inertia around principals axis which coincides with axes  $x \ y \ z$ .

+ - Autopilot is located inside, between the top cover and the main body element.

++ - This is the moment of inertia for all motors. Rotation around  $x$  and  $y$  axes is influenced by two motors, rotation around  $z$  axis is influenced by all four motors.

The moment of inertia for the whole body is then

$$I = J_b + J_{ic} + J_{bc} + J_{bat} + J_{ap} + 2J_{arm} + 2J_{ski} + J_{cam} + J_m = \begin{bmatrix} 0.0093 & 0 & 0 \\ 0 & 0.0092 & 0 \\ 0 & 0 & 0.0151 \end{bmatrix} kg\ m^2 \quad (4.7)$$

### 4.3 Aerodynamic coefficients identification

In the equations of motion there are two important aerodynamic coefficients. They are the  $b$ , the thrust coefficient of the propeller and  $d$ , the drag factor of the propeller.

#### 4.3.1 Thrust coefficient

To measure the thrust coefficient I used the data already used during the motor dynamics identification. The thrust is proportional to the square of the angular rate of the propeller (Hoffmann et al. 2007, p.8)

$$T = b \Omega^2 \quad (4.8)$$

Knowing the angular rate of the propeller during the hover position and the mass of the quadrotor the thrust coefficient is calculated to be,

$$b = \frac{mg}{4\Omega_0^2} = 1.5108 \cdot 10^{-5} kgm \quad (4.9)$$

### 4.3.2 Drag coefficient

The drag moment caused by the rotating propeller's is the main factor which causes the quadrotor to yaw. This drag moment is proportional to the square of the angular rate of each propeller (Bouabdallah et al. 2005, p.2).

To identify the drag coefficient of the propeller I used the already measured step response of the the motor with propeller around the hover trim point and step response of the propeller-less engine. The propeller-less motor's step response is not influenced by the aerodynamic effects. Hence comparing these two responses it is possible to estimate the drag coefficient of the propeller.

The equation (4.4) is modified. The drag function is replaced by the quadratic function and the drag coefficient  $d$ ,

$$J_r \dot{\omega}_m = -d_m \omega_m - d \omega_m^2 + \frac{k_i}{R} u \quad (4.10)$$

Then

$$\dot{\omega}_m = \frac{-d_m}{J_r} \omega_m - \frac{d}{J_r} \omega_m^2 + \frac{k_i}{R J_r} u \quad (4.11)$$

This equation is nonlinear. In order to find its transfer function, it has to be linearized using the Taylor expansion and dropping the higher order terms. The hover trim point is used. Then the linearized perturbed equation is

$$\left. \frac{\partial \dot{\omega}_m}{\partial \omega_m} \right|_{\omega_{m0}} + \left. \frac{\partial \dot{\omega}_m}{\partial u} \right|_{u_0} = \Delta \dot{\omega}_m = \frac{-d_m}{J_r} \Delta \omega_m - \frac{d}{J_r} 2 \omega_{m0} \Delta \omega_m + \frac{k_i}{R J_r} \Delta u \quad (4.12)$$

Deriving the transfer function for this equation leads to

$$\frac{\Omega_m(s)}{U(s)} = G_p(s) = \frac{k_i}{J_r R \left( s + \frac{d_m}{J_r} + \frac{2 d \omega_{m0}}{J_r} \right)} = \frac{k_i}{R (d_m + 2 d \omega_{m0}) \left( \frac{J_r}{b + 2 d \omega_{m0}} s + 1 \right)} \quad (4.13)$$

The time constant of this first order system is

$$\tau_p = \frac{J_r}{d_m + 2d\omega_{m0}} \quad J_r = J_m + I_p, \quad (4.14)$$

where  $J_m$  is the moment of inertia of the motor's rotor and  $I_p$  is the moment of inertia of the propeller.

Taking again the equation (4.4), dropping the the quadratic term and replacing  $J_r$  with  $J_{np}$  leads to

$$\dot{\omega}_m = \frac{-d_m}{J_{np}} \omega_m + \frac{k_i}{R J_{np}} u, \quad J_{np} = J_m + J_l, \quad (4.15)$$

where  $J_m$  is the moment of inertia of the motor's rotor again and  $J_l$  is the moment of inertia of an arbitrary load.

This is equation for the motor without the propeller and with some rings fitted to it to increase the moment of inertia of the rotor.

This equation is linear, so it is possible to find its transfer function immediately

$$G_{np}(s) = \frac{k_i}{R d_m \left( \frac{J_{np}}{d_m} s + 1 \right)} \quad (4.16)$$

The time constant for this system is

$$\tau_{np} = \frac{J_{np}}{d_m} \quad J_{np} = J_m + J_l \quad (4.17)$$

To drag factor can be calculated from the the equation (4.14)

$$d = \frac{1}{2\omega_{m0}} \left( \frac{J_r}{\tau_p} - d_m \right) \quad (4.18)$$

Substituting from the equation (4.17)

$$d = \frac{1}{2\omega_{m0}} \left( \frac{J_r}{\tau_p} - \frac{J_{np}}{\tau_{np}} \right) = \frac{1}{2\omega_{m0}} \left( \frac{J_m + J_p}{\tau_p} - \frac{J_m + J_l}{\tau_{np}} \right) \quad (4.19)$$

To calculate the time constant of the propeller-less motor and moments of inertia of the motor's rotor and the loading mass has to be found.



*Picture 4.8: Loading mass attached to the motor.*

The loading mass attached to the motor instead of the propeller were three metal rings. The load's moment of inertia is

$$J_l = \frac{1}{2} m_l (r_{l1}^2 + r_{l2}^2) \quad , \quad (4.20)$$

where  $m_l$  is the load's mass,  $r_{l1}$  is the inner radius of load and  $r_{l2}$  is the outer radius.

Then

$$J_l = \frac{1}{2} 32.2 \cdot 10^{-3} ((9 \cdot 10^{-3})^2 + (1.5 \cdot 10^{-2})^2) \text{ kg m}^2 = 4.927 \cdot 10^{-6} \text{ kg m}^2 \quad (4.21)$$

The rotor has the ring shape as well hence its moment of inertia is calculated using the same approach

$$J_m = \frac{1}{2} 20.3 \cdot 10^{-3} ((9.5 \cdot 10^{-3})^2 + (1.25 \cdot 10^{-2})^2) \text{ kg m}^2 = 2.506 \cdot 10^{-6} \text{ kg m}^2 \quad (4.22)$$

To find the time constant of the propeller-less motor  $\tau_{np}$  I measured the step response of this system and fitted the estimated system with time constant  $\tau_{np}$  to it.

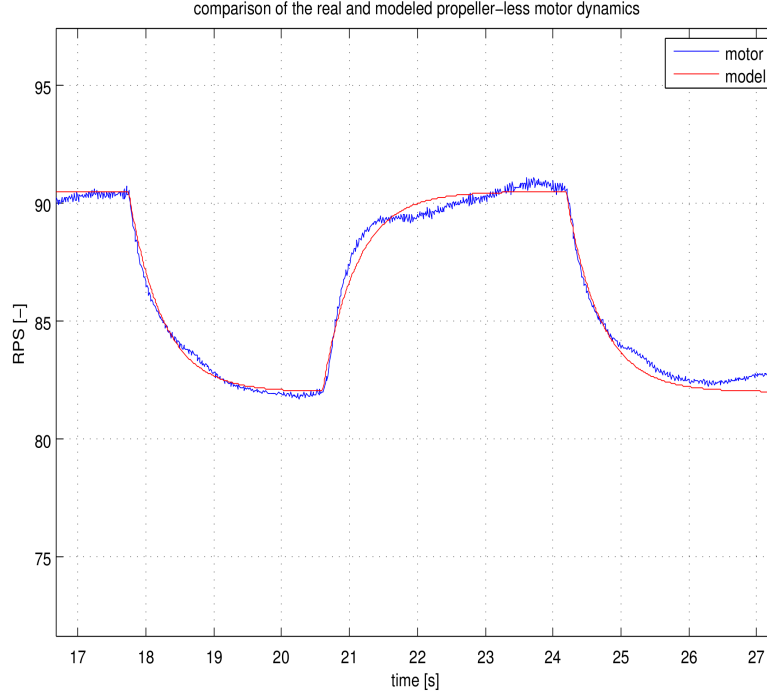


Figure 4.4: Comparison of the proper-less motor dynamics.

The estimated time constant is  $\tau_{np}=0.5$  .

Unfortunately it was not possible to measure around the trim point, because the slowest speed of the propeller-less motor is around 82 revolutions per second.

The different working point should not add too much error. The bearing damping at 82 RPS is not very different from the bearing damping at 76 RPS, the moments of inertia are not affected and the  $\omega_{m0}$  used for the calculation has the value  $\omega_{m0}=82 \cdot 2 \pi$  .

Then the drag factor is calculated using the equation (4.19) and it is  $d=4.406 \cdot 10^{-7} \text{ kg m}^2 \text{ s}^{-1}$  .

## 5 Nonlinear mathematical model

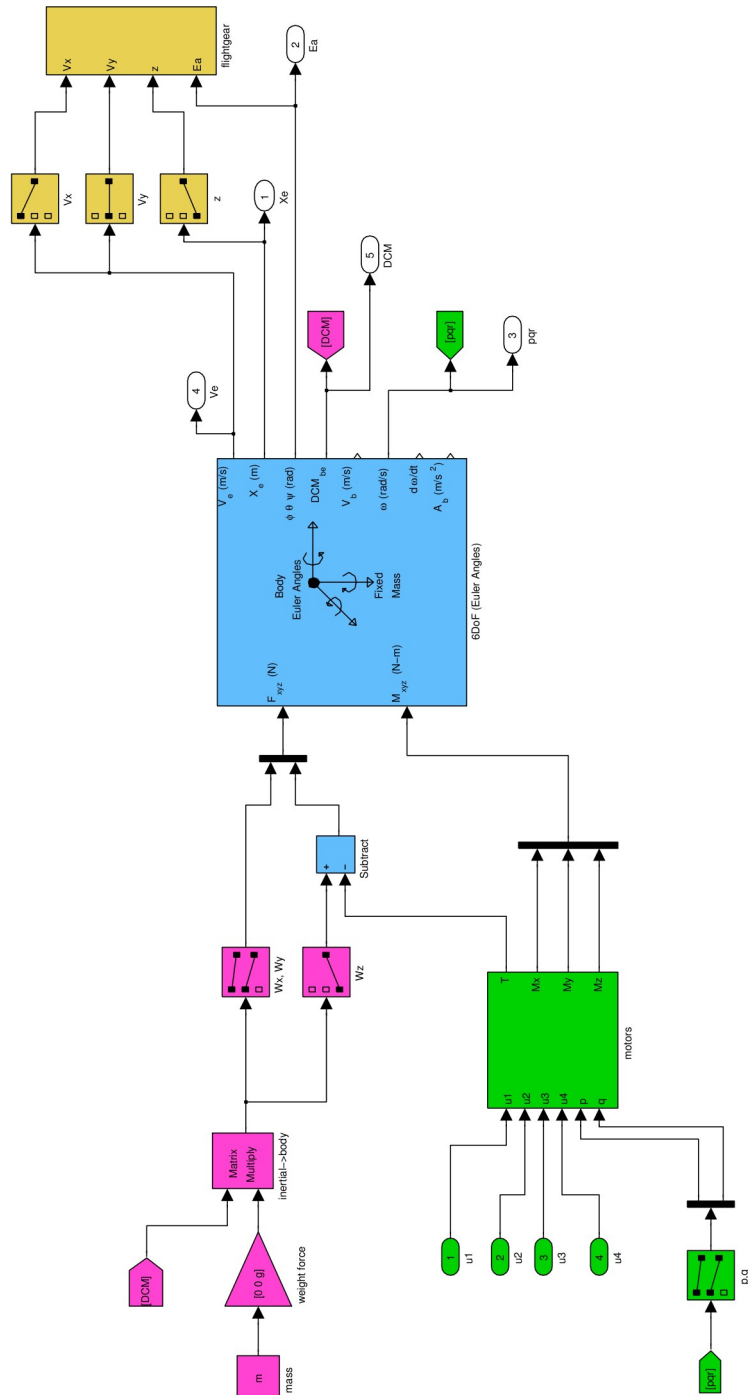
To test the linear controllers and prevent damaging the real quadrotor a nonlinear model has been built using the Simulink.

This model simulates the expanded equations of motion from the chapter 2.3 and adds saturation to the engines' speed to limit their revolutions per second in 0-150 range.

As shown in picture 5.1 the mathematical model is divided into following parts:

- 6DOF rigid body dynamics
- gravity subsystem
- engines subsystem
- FlightGear subsystem





Picture 5.1: Nonlinear model of the quadrotor.

### 5.1 6DOF rigid body subsystem

This block was taken from the Aerospace blockset in Simulink library. It contains all the basic equations of motion for rigid body in free space. The parameters for this block are mass and tensor of inertia of the body, and the initial conditions.

The inputs are forces and moments acting on the body in all three axis in body-fixed frame.

Outputs are position and velocity in earth-fixed frame, velocity and acceleration in body-fixed frame, Euler angles, angular rates and angular acceleration, and the direction cosine matrix for transformation from earth-fixed frame to body-fixed frame.

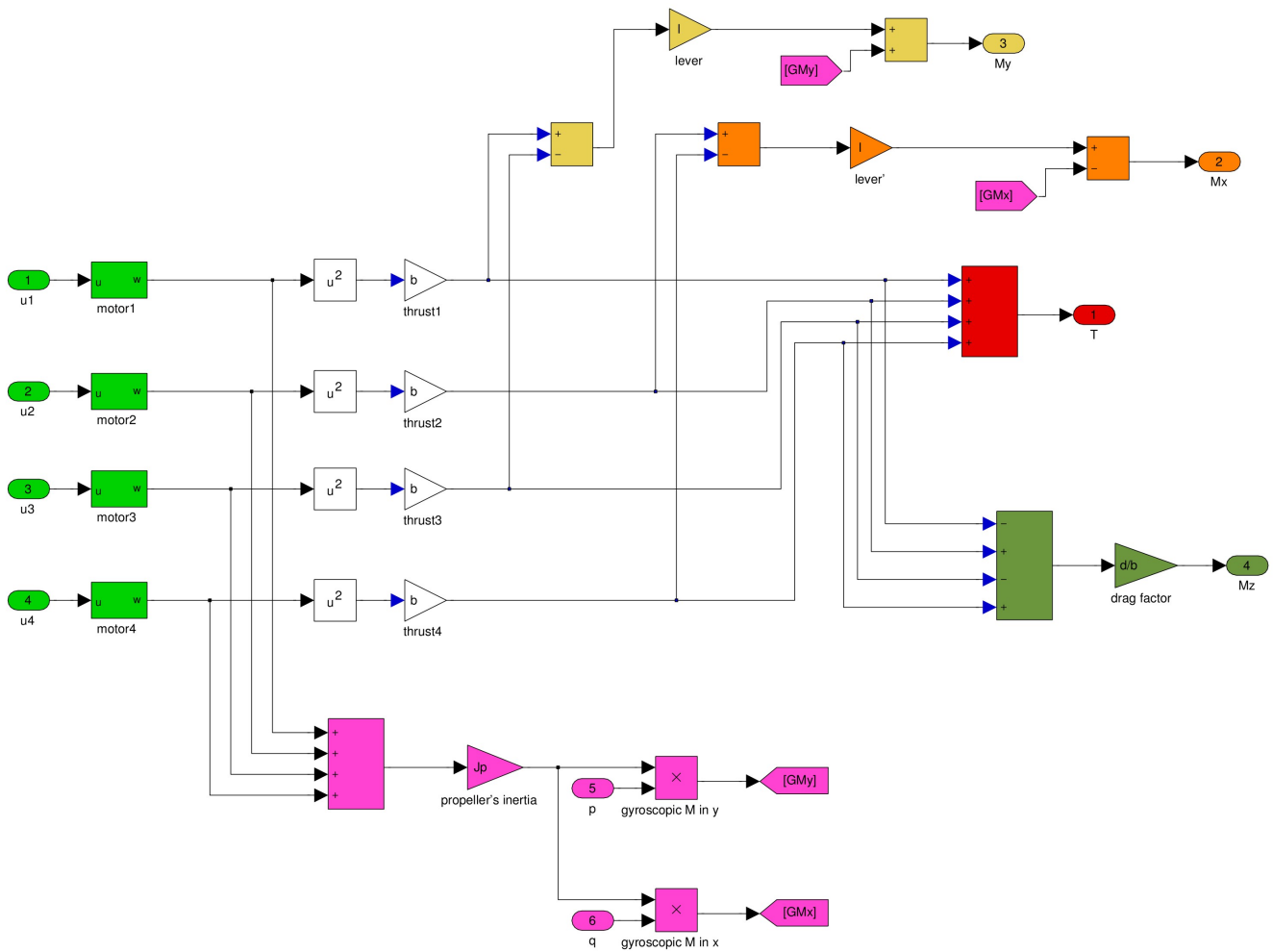
### 5.2 Gravity subsystem

This is simple transformation of weight force acting always in the  $z$  direction in the earth-fixed frame to the body-fixed frame using the direction cosine matrix. The  $z$  component of the weight force in the body-fixed frame is then subtracted by the thrust force of the motors which is always acting in the  $z$  direction in the body-fixed frame.

### 5.3 Engines subsystem

The engines subsystem produces the external thrust force and the external moments described by the equations (2.16) and (2.26).

As can be seen from the picture 5.2, there are **motors** producing angular rate which depends on the the supplied **voltage**  $u_{1-4}$  (4.4). **The square of the angular rates multiplied by the thrust constant produces the thrust** for each engine. These thrusts generate the **main thrust**  $T$  and the **rolling** and **pitching** moments  $M_x$  and  $M_y$ . The sum of squares of the angular rate is then producing the **yawing moment**  $M_z$ .

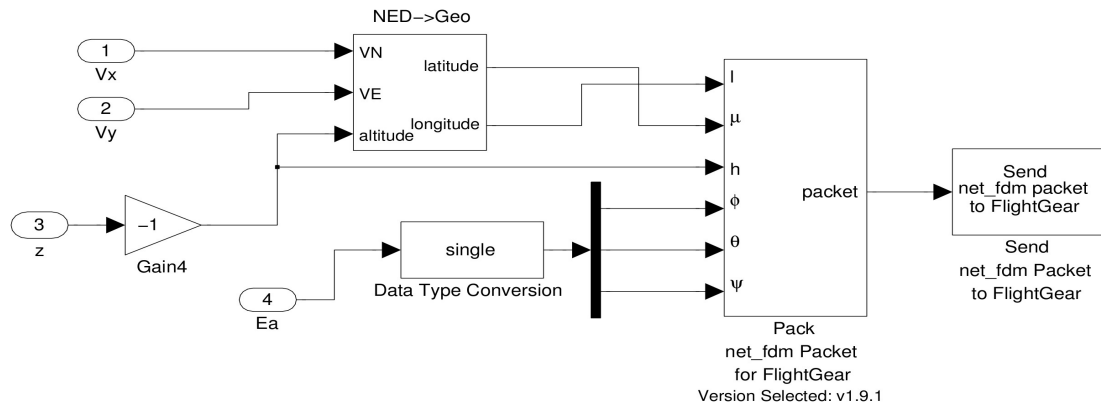


Picture 5.2: Engine subsystem from the nonlinear model.

At last the **whole angular momentum of the rotating propellers** in conjunction with the **pitch and roll rate creates the gyroscopic moments** which contributes to the rolling and pitching moments according to the equation (2.33).

## 5.4 FlightGear subsystem

The FlightGear subsystem is used for sending the position and attitude data to the FlightGear simulator which is used for visualization. The position needs to be converted from the the Cartesian NED coordinates system the the spherical Geographic system. Some other minor conversions need to be done as well. The blocks 'pack net\_fdm pakcet' and 'send net\_fsm packet' are taken from the Aerospace blockset in the Simulink library.



Picture 5.3: FlightGear subsystem from the nonlinear model.



Picture 5.4: FlightGear visualization demonstration.

## 6 Designing the control loops

In this chapter I am going to describe the following topics:

- Available controllers in the quadrotor's firmware and their features
- The overall control system strategy
- The linearized state equations and their separation
- The design of the linear controllers for each separated movement
- The verification of the designed controllers using the nonlinear model

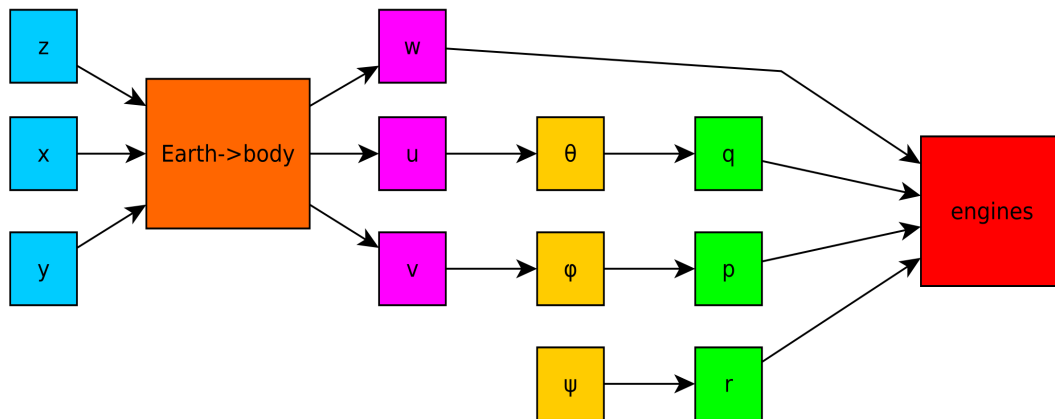
### 6.1 *Firmware limitations*

When I started working with the quadrotor the firmware was able to implement only the PI control loops (D is also present but it works only as a proportional gain, it does not differentiate the signal). I started to develop the control loops respecting these limitations. The later version of the firmware provides an interface which can be used to download the sensor data and upload control commands. Then the control algorithms can be implemented on different host. This version of firmware was released too late and I did not have sufficient time to exploit its new features allowing to use more sophisticated and modern control techniques.

The firmware provides four inner loops whose outputs are fed through a mixer to the engines. Then there are eight outer loops which can be connected to the inner loops or to each other. Loops' inputs can be other loop's output, sensor data or RC stick position. All inputs and outputs can be scaled and offsets can be added. All outputs can be limited. More information can be found in chapter 7.1.

### 6.2 *Control strategy*

The goal is to have complete position control of the quadrotor. To achieve this multiple control loops have to be implemented.



Picture 6.1: Control loops schematic

Picture showing different levels of control loops. From inner loops to outer loops: angular rates control, attitude control, body-fixed frame speeds control and earth-fixed frame position control.

In the picture 6.1 each block represents one control loop with the controlled variable written inside. The design of the nested loops was chosen because of following reasons:

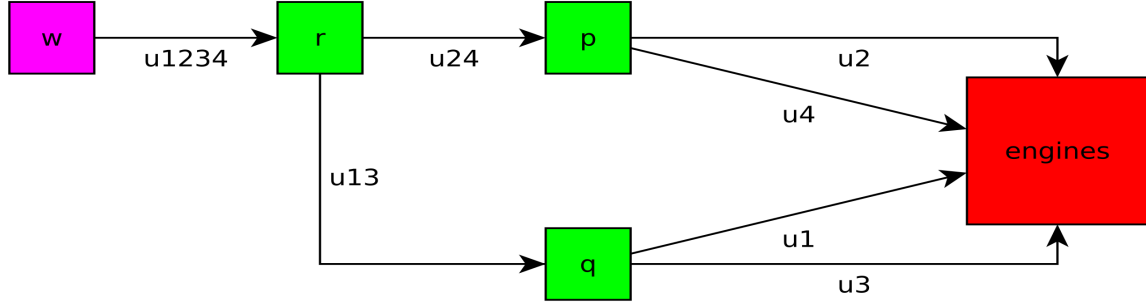
- The quadrotor system contains twelve integrators (this can be seen from the linear state equations in chapters 6.5.1, 6.5.2 and 6.5.3) which are introducing phase lag. To counter this derivative action creating phase lead needs to be introduced. As mentioned before the D parts of the PIDs are only acting as normal proportional gains thus introducing no phase lead at all. The only controllable astatic system by the simple P controller is system with only one integrator (Skogestad & Postlethwaite 1996, p.205). Hence the control loops are decomposed in the manner to have only one integrator in them.
- Nested control loops provide more precise control of the system. It is very very convenient to put limitations on some control signal (e.g. pitch and roll angle limit), which is easily achieved using the nested loops.
- Nested loops provide more flexibility. In a case when lower level control is needed (e.g. velocities, attitude, ...) it is very easy to disconnect the outer loops and leave only the appropriate inner loops.

The disadvantage of many control loops each controlling only one state is slower response time

compared to controller controlling more state simultaneously.

### 6.3 Channel Mixer

As it is shown in the picture 6.1, four control loops are accessing the engine subsystem. Since each controller has only one output a mixing strategy was developed.



Picture 6.2: Channel mixer schematic.

As can be seen in picture above, the main engine command ( $U_{1234}$ ) for all motors is created in the vertical speed controller. Then this command is split into the commands for the motor pairs consisting of engines 1 and 3 or engines 2 and 4 ( $U_{13}$  and  $U_{24}$  respectively) according to this equation

$$\begin{aligned} U_{24} &= (mm + 1) U_{1234} \\ U_{13} &= (1 - mm) U_{1234} \quad , \\ mm &\in \langle -1, 1 \rangle \end{aligned} \quad (6.1)$$

where  $mm$  is the output of the yaw rate controller.

The command for engines 1 and 3 ( $U_{13}$ ) is finally split into the separate engine commands for engines 1 and 3 ( $U_1$  and  $U_3$ ) according to the output of the pitch rate controller  $n$ . This is realized by the equation

$$\begin{aligned} U_1 &= (n + 1) U_{13} \\ U_3 &= (1 - n) U_{13} \\ n &\in \langle -1, 1 \rangle \end{aligned} \quad (6.2)$$

The commands for engines 2 and 4 are obtained using the exactly same approach as for engines 1 and 3 but the roll rate controller is supplying the mixing ratio  $o$  instead of the ratio  $n$ ,

### 6.4 Linearized state equations

The obvious trim point is when the quadrotor is hovering still at one spot and its body-fixed frame is aligned with the earth-fixed frame. This situation is described by the following state values:

$$\begin{aligned} x_0 &= 0 & \phi_0 &= 0 & u_0 &= 0 & p_0 &= 0 \\ y_0 &= 0 & \theta_0 &= 0 & v_0 &= 0 & q_0 &= 0 \\ z_0 &= 0 & \psi_0 &= 0 & w_0 &= 0 & r_0 &= 0 \\ -\Omega_{10} &= \Omega_{20} = -\Omega_{30} = \Omega_{40} = 463.1 \text{ rad s}^{-1} \end{aligned} \quad (6.3)$$

Differentiating the state equations with respect to each state in the trim point leads to perturbed state-space equations

$$\begin{aligned} \Delta \dot{x} &= \Delta u \\ \Delta \dot{y} &= \Delta v \\ \Delta \dot{z} &= \Delta w \end{aligned} \quad (6.4)$$

$$\begin{aligned} \Delta \dot{\phi} &= \Delta p \\ \Delta \dot{\theta} &= \Delta q \\ \Delta \dot{\psi} &= \Delta r \end{aligned} \quad (6.5)$$

$$\begin{aligned} \Delta \dot{u} &= r_0 v \Delta r + r v_0 \Delta v - q_0 w \Delta q - q w_0 \Delta w - g \cos \theta_0 \Delta \theta \\ \Delta \dot{v} &= p_0 w \Delta p + p w_0 \Delta w - r_0 u \Delta r - r u_0 \Delta u - g \sin \theta_0 \sin \phi_0 \Delta \theta + g \cos \theta_0 \cos \phi_0 \Delta \phi \\ \Delta \dot{w} &= q_0 r \Delta q + q r_0 \Delta r - p_0 v \Delta p - p v_0 \Delta v - g \sin \phi_0 \cos \theta_0 \Delta \phi - g \cos \phi_0 \sin \theta_0 \Delta \theta - 2 \Omega_0 \frac{b}{m} Q \\ Q &= \Delta \Omega_1 + \Delta \Omega_3 - \Delta \Omega_2 - \Delta \Omega_4 \end{aligned} \quad (6.6)$$

$$\begin{aligned} \Delta \dot{p} &= 2 \frac{lb}{I_x} \Omega_0 (\Delta \Omega_2 - \Delta \Omega_4) - (q_0 r \Delta q + q r_0 \Delta r) \frac{I_z - I_y}{I_x} \\ \Delta \dot{q} &= 2 \frac{lb}{I_y} \Omega_0 (\Delta \Omega_1 - \Delta \Omega_3) - (p_0 r \Delta p + p r_0 \Delta r) \frac{I_x - I_z}{I_y} \\ \Delta \dot{r} &= 2 \frac{d}{I_z} \Omega_0 (\Delta \Omega_1 + \Delta \Omega_2 + \Delta \Omega_3 + \Delta \Omega_4) \end{aligned} \quad (6.7)$$

I have already identified the motor system and described it by the linear transfer function. This transfer function can be realized for example by this state-space description

$$\dot{\Omega}_m = -10 \Omega_m + 7 U \quad (6.8)$$

The  $\Delta$  symbol represents the perturbation from the trim point. From now on all equations of



motions will be linear and representing the perturbed system. For easier reading the  $\Delta$  sign is dropped.

After substituting the trim point values into the linearized state equations

$$\begin{aligned}\dot{x} &= u \\ \dot{y} &= v \\ \dot{z} &= w\end{aligned}\tag{6.9}$$

$$\begin{aligned}\dot{\phi} &= p \\ \dot{\theta} &= q \\ \dot{\psi} &= r\end{aligned}\tag{6.10}$$

$$\begin{aligned}\dot{u} &= -g\theta \\ \dot{v} &= g\phi \\ \dot{w} &= -2\Omega_0 \frac{b}{m} (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4)\end{aligned}\tag{6.11}$$

$$\begin{aligned}\dot{p} &= 2 \frac{lb}{I_x} \Omega_0 (\Omega_2 - \Omega_4) \\ \dot{q} &= 2 \frac{lb}{I_y} \Omega_0 (\Omega_1 - \Omega_3) \\ \dot{r} &= 2 \frac{d}{I_z} \Omega_0 (\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4)\end{aligned}\tag{6.12}$$

## 6.5 Control system design

In this chapter all necessary controllers are designed and tested using the nonlinear model. All controllers are designed using root locus method to place poles into reasonable positions. The design performance is evaluated in time domain using the step response characteristic and in frequency domain using bode plots.

The step response should be as fast as possible but the maximum overshoot is 10%.

The bode plots are used to measure the robustness criteria, the gain and phase margin. The gain margin should be at least 3dB and the phase margin at least 30 degrees. These are the standard rule of thumb safety margins which provides safeguards against model uncertainties and delays. More in (Skogestad & Postlethwaite 1996, p.35)

The quadrotor's movement can be divided into four types:

- vertical movement (in  $z$  axis)
- longitudinal movement (in  $x$  axis)
- lateral movement (in  $y$  axis)
- yawing movement (rotation around  $z$  axis)

Each of these movements will be analyzed and appropriate control loops designed.

### 6.5.1 The vertical movement

The vertical movement represents the quadrotor keeping its attitude and only changing its vertical speed and position. This movement is described by the following equations

$$\begin{aligned}\dot{z} &= w \\ \dot{w} &= -2 \Omega_0 \frac{b}{m} (\Omega_1 + \Omega_3 - \Omega_2 - \Omega_4) \\ \dot{\Omega}_i &= -10 \Omega_i + 7 U \quad i = \{1, 2, 3, 4\}\end{aligned}\tag{6.13}$$

The vertical speed and position is controlled by all four engines, the state-space system then looks

$$\begin{bmatrix} \dot{z} \\ \dot{w} \\ \dot{\Omega}_1 \\ \dot{\Omega}_2 \\ \dot{\Omega}_3 \\ \dot{\Omega}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-b}{m} \Omega_0 & \frac{b}{m} \Omega_0 & \frac{-b}{m} \Omega_0 & \frac{b}{m} \Omega_0 \\ 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} z \\ w \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + 7 \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} U \tag{6.14}$$

With the values substituted

$$\begin{bmatrix} \dot{z} \\ \dot{w} \\ \dot{\Omega}_1 \\ \dot{\Omega}_2 \\ \dot{\Omega}_3 \\ \dot{\Omega}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0106 & 0.0106 & -0.0106 & 0.0106 \\ 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} z \\ w \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + 7 \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} U = A_v \begin{bmatrix} z \\ w \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + B_v U \quad (6.15)$$

*Note: Although the system is uncontrollable, this poses no problem. It is not possible to control angular rate of each engine separately but the angular rates for all engines, vertical speed and vertical position are controllable which is what is desired. This can be verified by transforming the system into Standard form of uncontrollable systems, more in (Antsaklis & Michel 2007, p.238). This similarly applies for the linear systems describing the other types of movement.*

### 6.5.1.1 Linear controllers' design

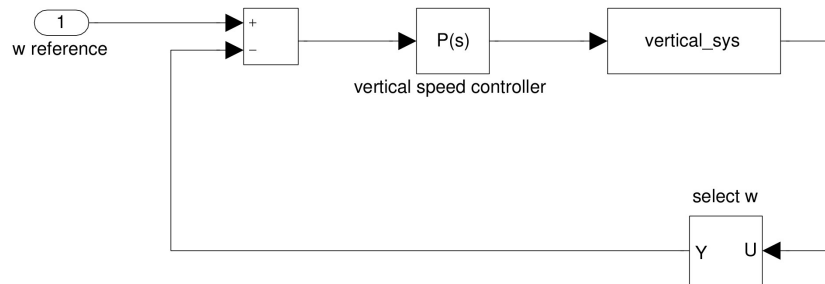
Considering the system with the vertical speed  $w$  as the output

$$y = [0 \ 1 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} z \\ w \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix}^T = C_{vw} \begin{bmatrix} z \\ w \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix}^T \quad (6.16)$$

The resulting system's transfer function is then

$$G_{vw}(s) = C_{vw} (sI - A_v)^{-1} B_v \quad (6.17)$$

Closing the loop as shown in the following picture



Picture 6.3: Vertical speed control loop.

and performing the root locus, bode and step response analysis for the  $G_{vw}$ , I tuned the P controller to gain  $P_w = -157$ .

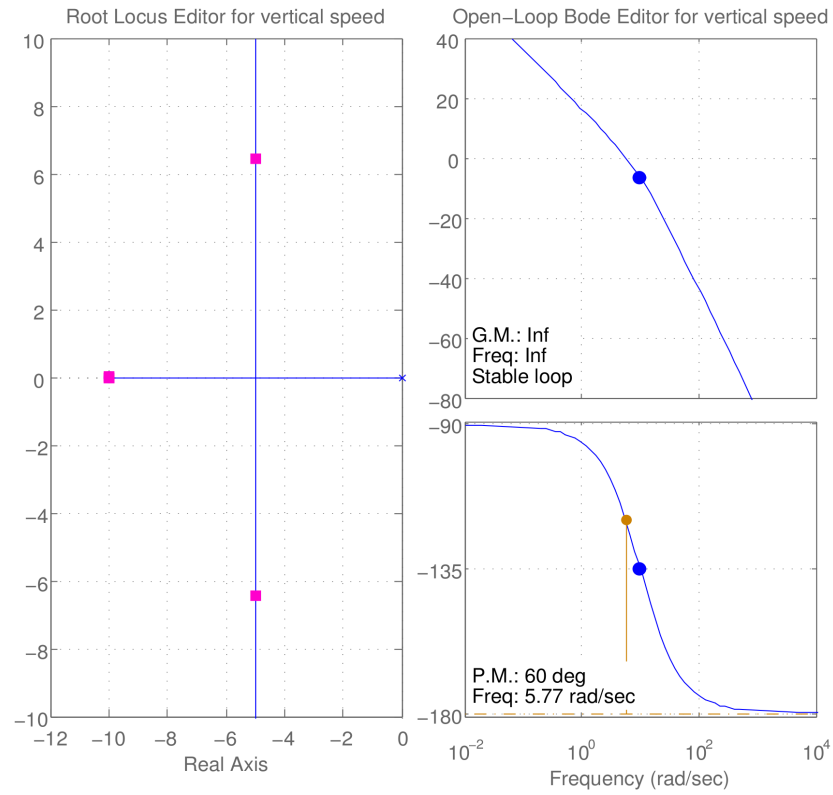


Figure 6.1: Root locus and bode plots for the vertical speed loop.

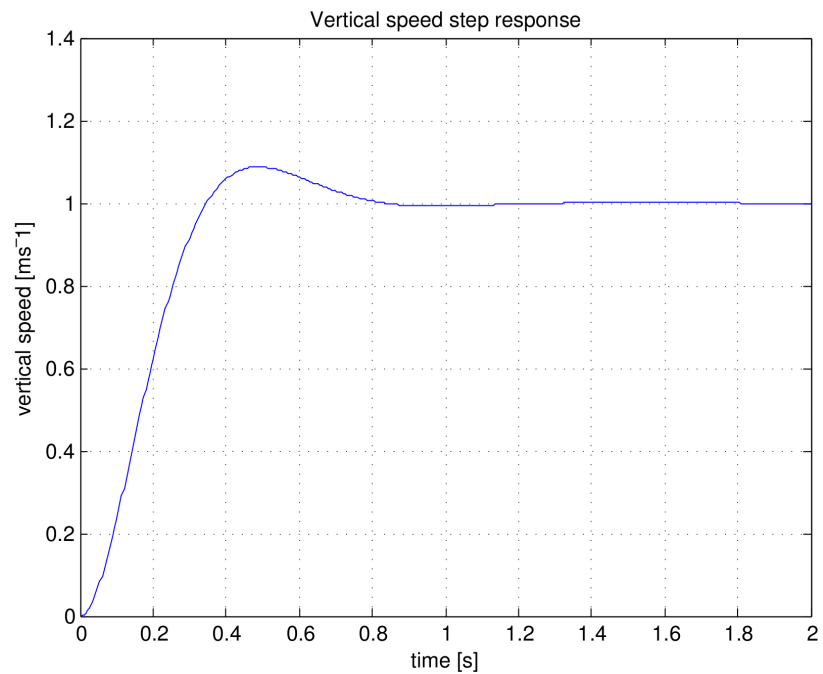
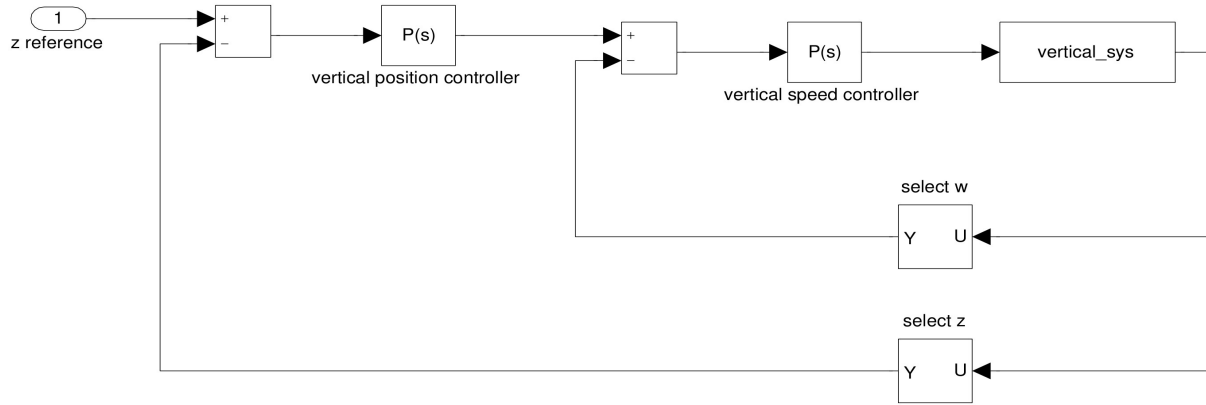


Figure 6.2: Step response of the vertical speed loop.

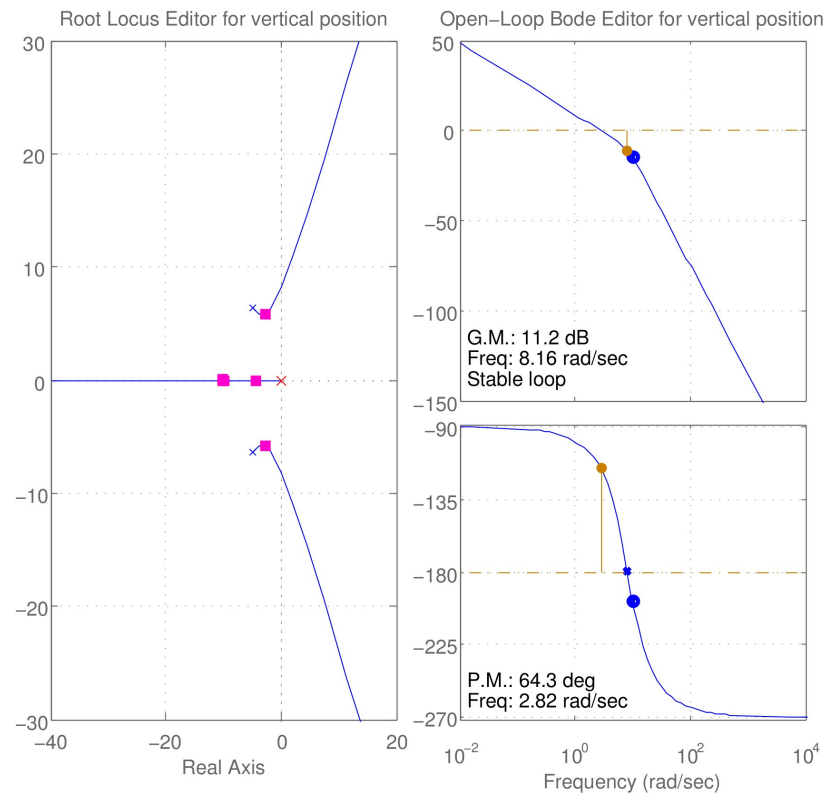
With the vertical speed loop closed I designed the control loop for vertical position  $z$  stacked on the vertical speed control loop.



Picture 6.4: Vertical position control loop.

Closing the loop of the system (6.17) using the designed controller and taking the vertical position  $z$  as an output leads to transfer function

$$G_{vz}(s) = \frac{P_w G_{vw}(s)}{1 + P_w G_{vw}(s)} \frac{1}{s} \quad (6.18)$$



*Figure 6.3: Root locus and bode plots for the vertical position loop.*

Using root locus, bode characteristics and step response for the  $G_{vz}(s)$  system, I tuned the vertical position controller's gain to  $P_x=2.76$

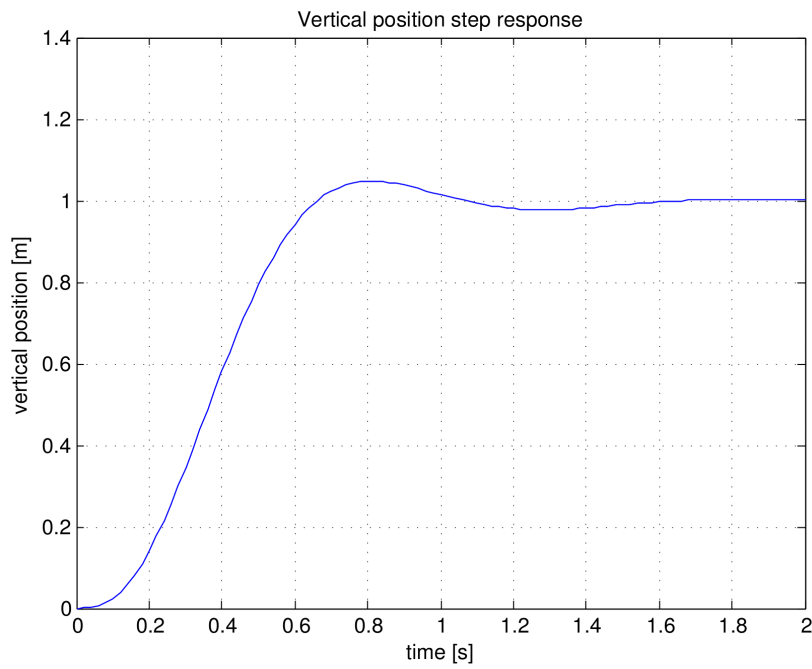
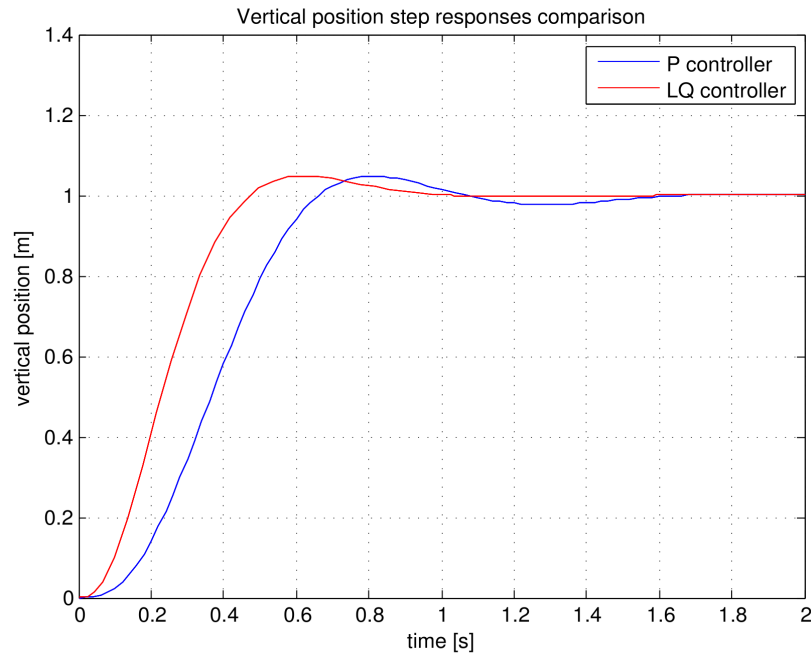


Figure 6.4: Step response of the vertical position loop.

I also tried to design an LQR controller for the vertical position. Using the the tuning parameters

$$R=1 \quad Q=\text{diag}[10^7 \ 1 \ 1 \ 1 \ 1 \ 1] \quad , \quad (6.19)$$

and the *lqr* MATLAB function I got following step response to the vertical position command.



*Figure 6.5: Comparison of the P and LQ controller using linear system.*

#### **6.5.1.2 Verification on nonlinear model**

Controllers designed in previous chapter were tested on the nonlinear model to verify their performance using more realistic model.



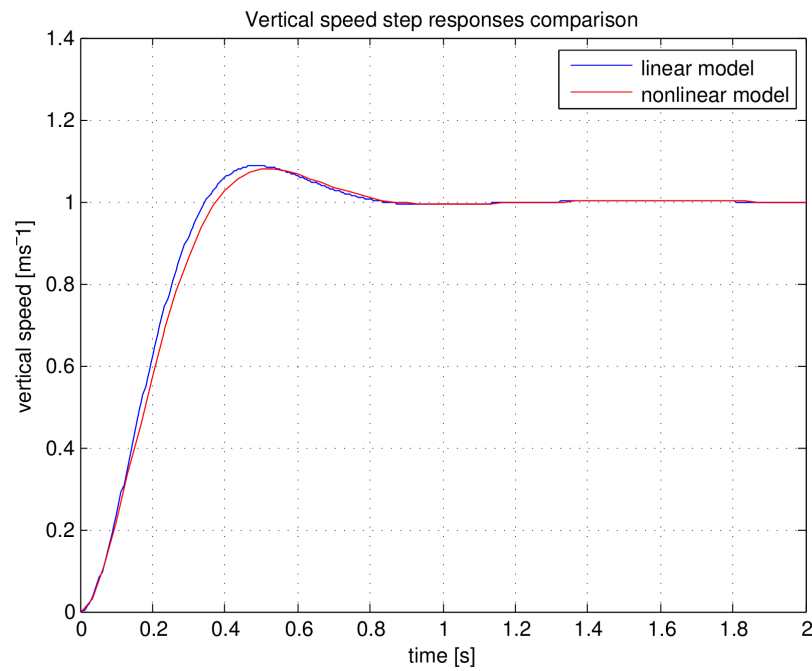


Figure 6.6: Vertical speed step responses comparison, linear and nonlinear plant control.

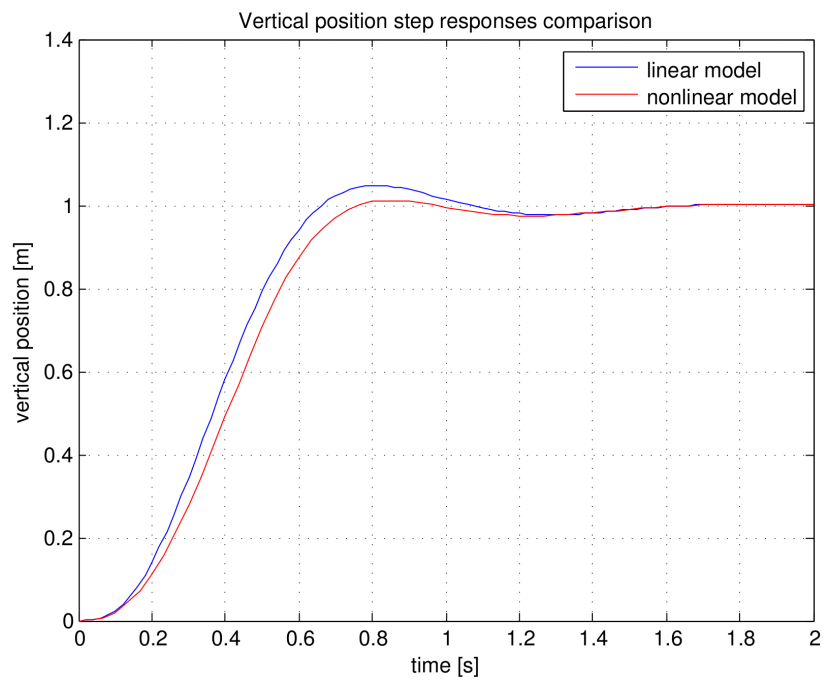


Figure 6.7: Vertical position responses comparison, linear and nonlinear plant control.

On the linear system it does not matter what is the magnitude of the reference signal, it will be linearly scaled. But the nonlinear elements in the nonlinear system behave differently when various magnitudes of the control signal are applied.

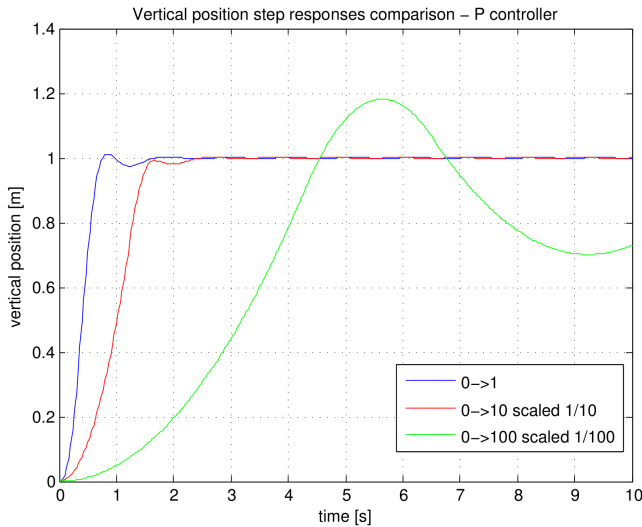


Figure 6.8: Vertical position step responses comparison, nonlinear plant, different steps, P controller.

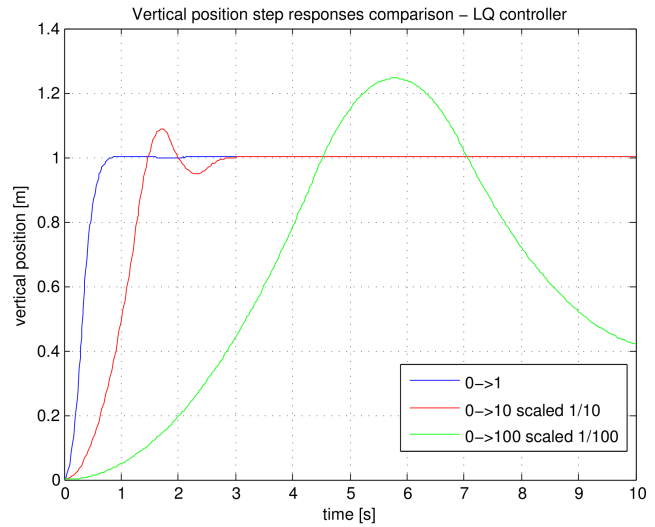


Figure 6.9: Vertical position step responses comparison, nonlinear plant, different steps, LQ controller.

As can be seen from the figures 6.8 and 6.9 when bigger steps are applied, the nonlinear system responds differently. This is caused by the nonlinear square dependency of angular rates of the motors and the generated thrust and mainly that the engines cannot turn in the opposite direction and their maximum angular rate is limited. When the controller tries to decelerate using the reverse thrust of the engines, the engines simply stop and the quadrotor overshoots the desired height. After falling through the reference the controller tries to fight the descent but the motors are already saturated and cannot provide more thrust. This leads to the big oscillations when the step from 0 to 100 is introduced.

The LQ controller is more susceptible because of its more aggressive control.

To fix this problem I introduced a limit on the vertical speed at some reasonable value of  $10 \text{ ms}^{-1}$ . For the P controller it means that the output of the vertical position controller (which is the command for the vertical speed controller.) is limited to  $\pm 10 \text{ ms}^{-1}$  range. For the LQ position controller I introduced a rate limiter with slew rate of 10 on the reference input.

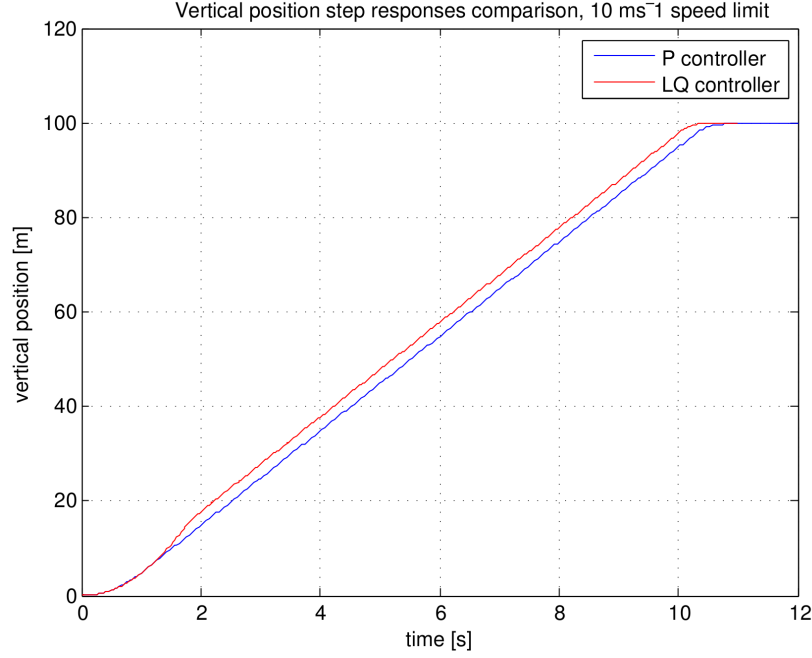


Figure 6.10: Vertical position step responses comparison with the maximum vertical speed limit.

### 6.5.2 The yawing movement

The yawing movement is the movement when the quadrotor is keeping its position and only its yaw angle changes. This movement is described by the following equations

$$\begin{aligned}\dot{\psi} &= r \\ \dot{r} &= 2 \frac{d}{I_z} \Omega_0 (\Omega_1 + \Omega_2 + \Omega_3 + \Omega_4) \\ \dot{\Omega}_i &= -10 \Omega_i + 7 U_i \quad i = \{1, 2, 3, 4\}\end{aligned}\tag{6.20}$$

As described in the chapter 6.3, the yaw rate controller is not controlling the engines directly but it is only modifying the mixing ratio  $mm$ . The input for engine 1  $U_1$  is then described by

$$\begin{aligned}U_1 &= (1+n)U_{13} \\ U_{13} &= (1-mm)U_{1234} \\ U_1 &= (1+n)(1-mm)U_{1234}\end{aligned}\tag{6.21}$$

Since there is no pitching and vertical movement involved and the linear system is perturbed system, the input equation can be simplified to

$$\Delta U_1 = (1 - mm)U_0 - U_0 = -U_0 mm \quad (6.22)$$

The same applies for the input  $U_3$  and the inputs  $U_2$  and  $U_4$  only have opposite sign. Respecting that the  $\Omega_1$  and  $\Omega_3$  create the negative yawing moment cancels out the negative sign from equation (6.22). The state-space description is then

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{\Omega}_1 \\ \dot{\Omega}_2 \\ \dot{\Omega}_3 \\ \dot{\Omega}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2\frac{d}{I_z}\Omega_0 & 2\frac{d}{I_z}\Omega_0 & 2\frac{d}{I_z}\Omega_0 & 2\frac{d}{I_z}\Omega_0 \\ 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} \psi \\ r \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + 7U_0 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} mm \quad (6.23)$$

With the values substituted

$$\begin{bmatrix} \dot{\psi} \\ \dot{r} \\ \dot{\Omega}_1 \\ \dot{\Omega}_2 \\ \dot{\Omega}_3 \\ \dot{\Omega}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.3006 & 0.3006 & 0.3006 & 0.3006 \\ 0 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} \psi \\ r \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + 3241 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} mm = A_y \begin{bmatrix} \psi \\ r \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} + B_y mm \quad (6.24)$$

### 6.5.2.1 Linear controller design

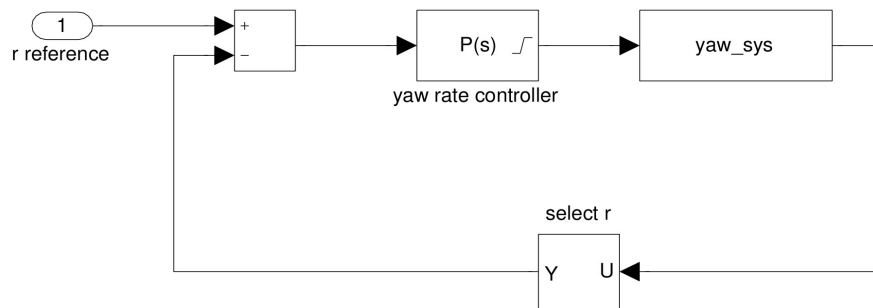
Considering the system with the yaw rate  $r$  as the output

$$y = [0 \ 1 \ 0 \ 0 \ 0 \ 0] \begin{bmatrix} \psi & r & \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^T = C_{yr} \begin{bmatrix} \psi & r & \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^T, \quad (6.25)$$

leads to the transfer function

$$G_{yr}(s) = C_{yr}(sI - A_y)^{-1} B_y \quad (6.26)$$

Closing the loop as shown in the following picture



Picture 6.5: Yaw rate control loop.

and doing the root locus, bode and step response analysis for the  $G_{yr}(s)$ .

I found a reasonable gain for the P controller to be  $P_r=0.12$ .

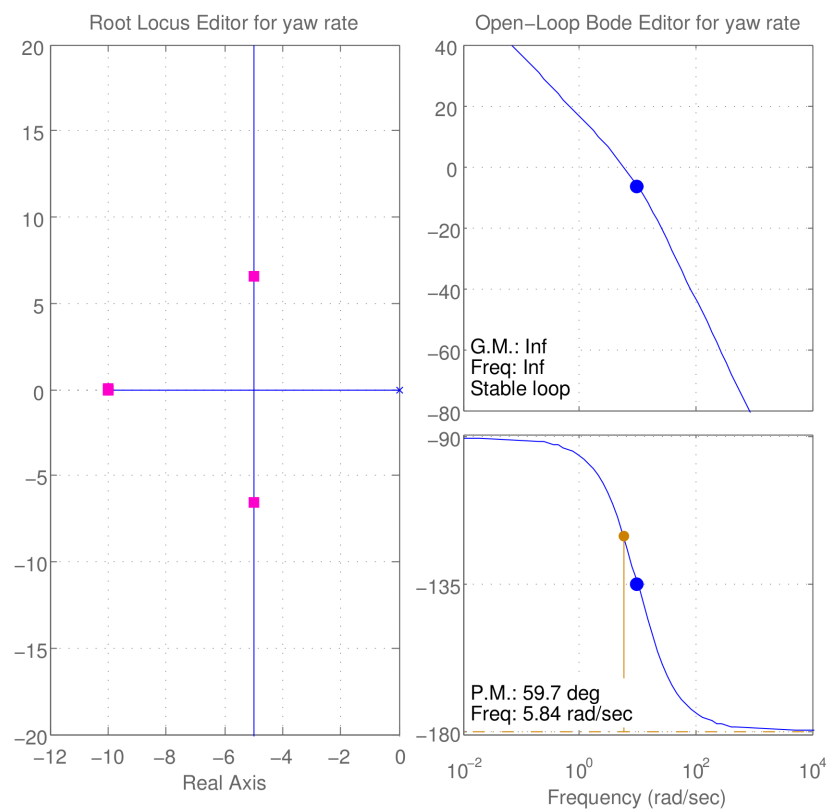


Figure 6.11: Root locus and bode plots for the yaw rate loop.

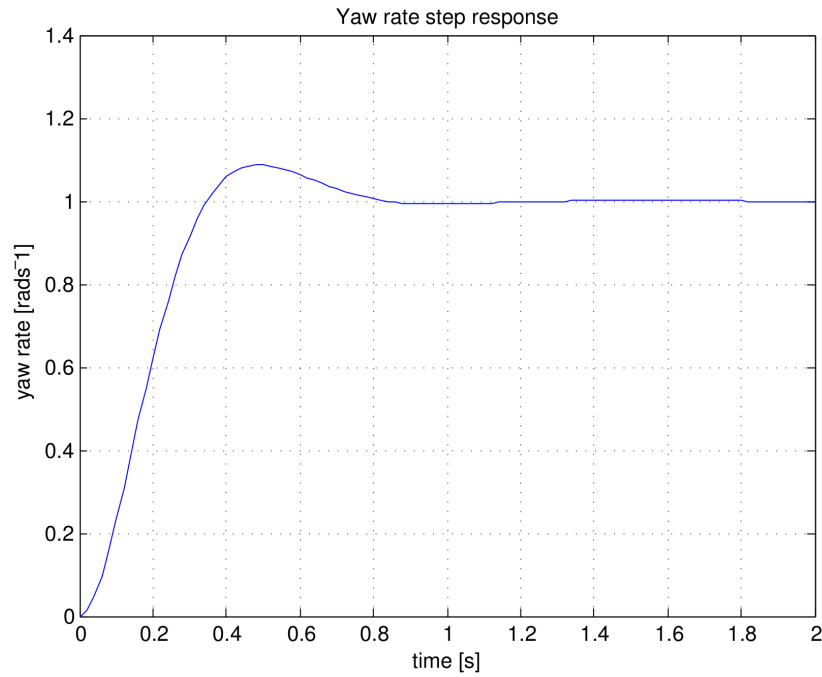
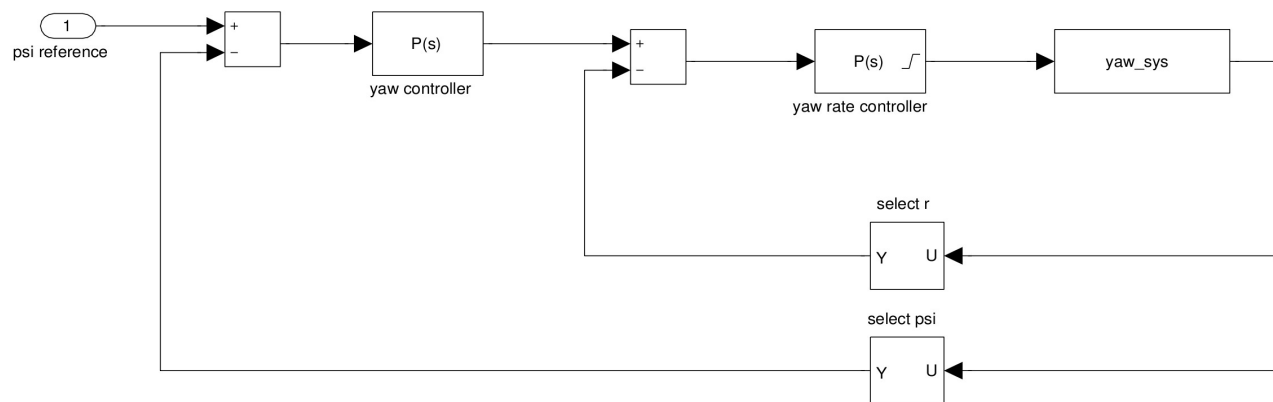


Figure 6.12: Step response of the yaw rate loop.

With the yaw rate control loop designed I stacked an yaw angle control loop on it. This means taking the closed loop transfer function with the yaw angle as an output

$$G_{y\psi}(s) = \frac{P_{yr} G_{yr}(s)}{1 + P_{yr} G_{yr}(s)} \frac{1}{s} \quad , \quad (6.27)$$

and designing the P controller for it. The controller's gain is  $P_{\psi} = 2.67$  .



Picture 6.6: Yaw angle control loop.

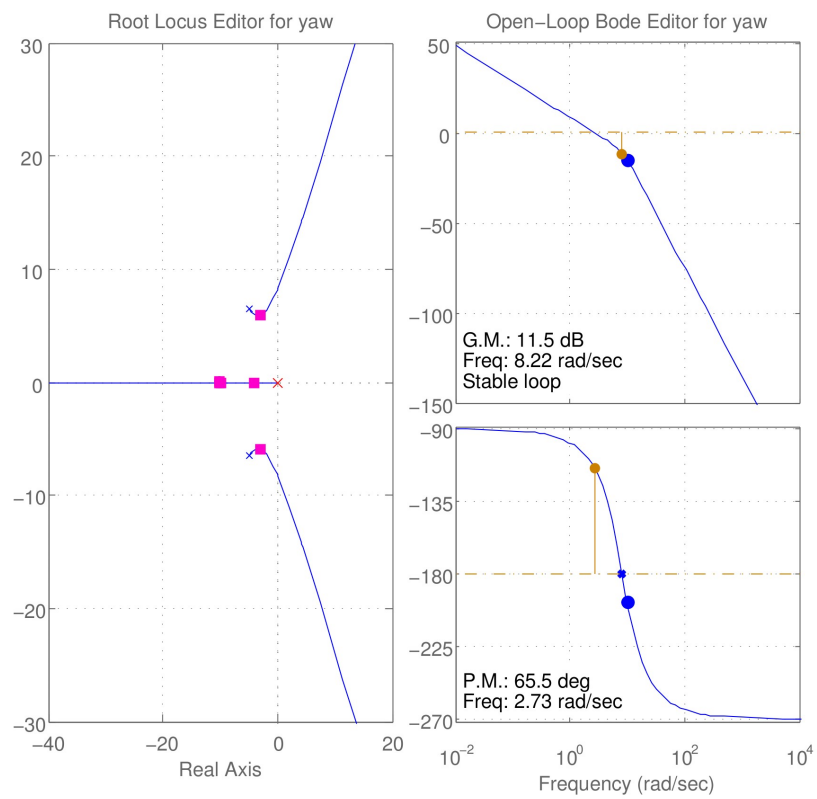


Figure 6.13: Root locus and bode plots for the yaw angle loop.

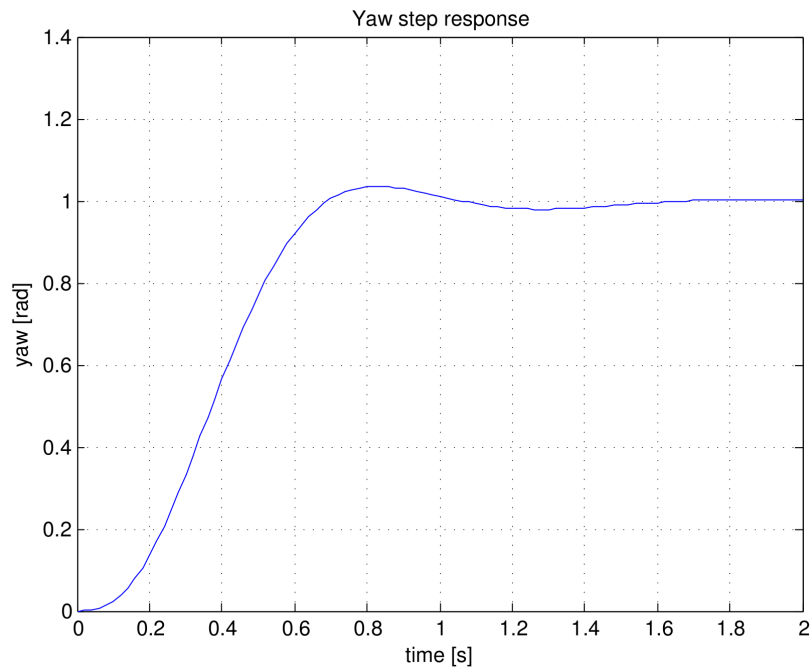


Figure 6.14: Step response of the yaw angle loop.

#### 6.5.2.2 Verification on nonlinear model

Controllers designed in previous chapter are tested on the nonlinear model to verify their performance using more realistic model.



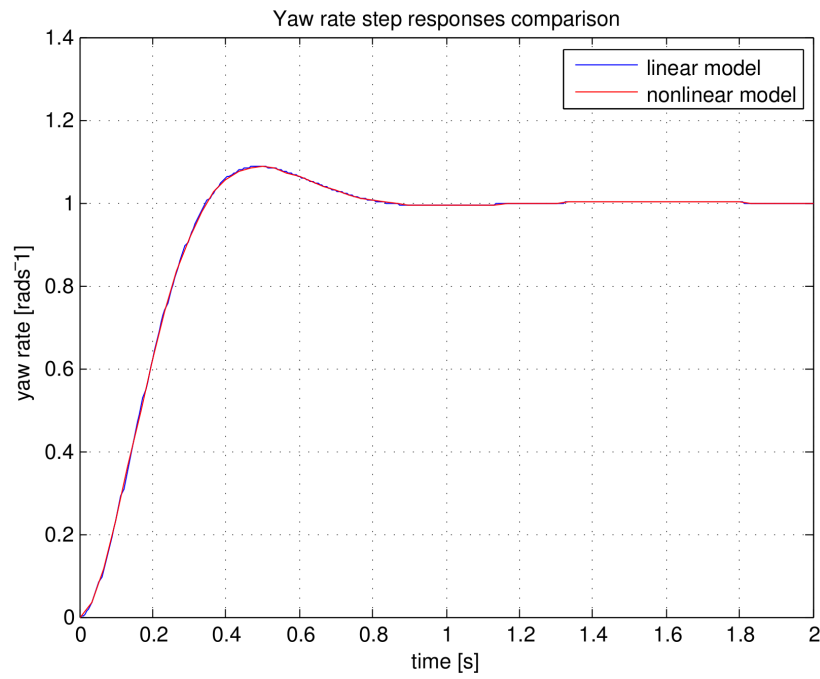


Figure 6.15: Yaw rate step responses comparison, linear and nonlinear plant control.

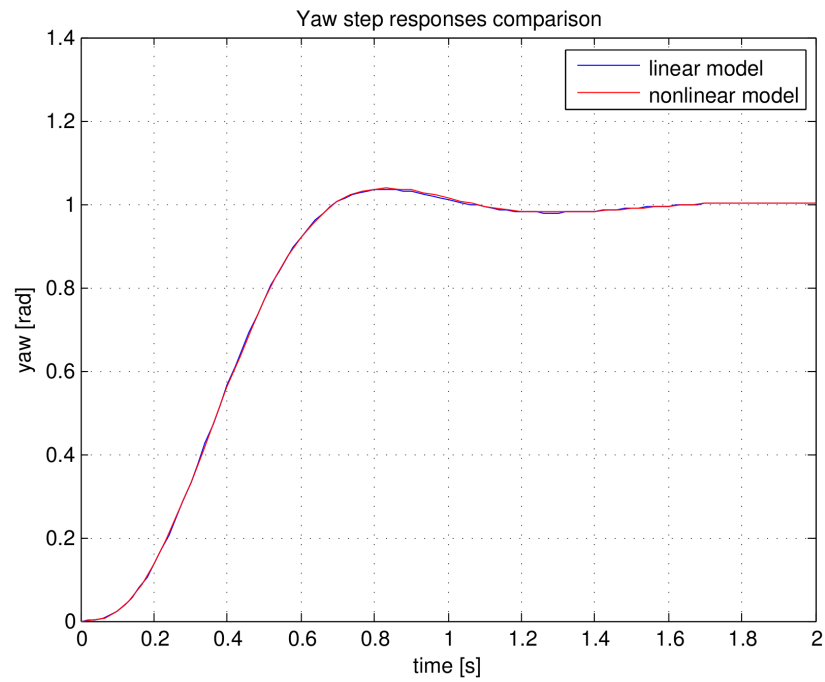


Figure 6.16: Yaw angle step responses comparison, linear and nonlinear plant control.

### 6.5.3 The longitudinal movement

The longitudinal movement is represented by the quadrotor keeping its vertical and lateral position and maintaining its yaw angle. This type of movement is described by the equations

$$\begin{aligned}\dot{x} &= u \\ \dot{\theta} &= q \\ \dot{u} &= -g \theta \\ \dot{q} &= 2 \frac{lb}{I_y} \Omega_0 (\Omega_1 - \Omega_3) \\ \dot{\Omega}_i &= -10 \Omega_i + 7 U_i \quad i = \{1, 3\}\end{aligned}\tag{6.28}$$

As it was described in chapter 6.3 the pitch rate controller is only changing the mixing ratio  $n$  for the motor 1 and 3. Similarly to chapter 6.5.2, input to longitudinal system is derived

$$U_1 = (1 + n)(1 - mm) U_{1234}\tag{6.29}$$

There is not yawing motion present hence

$$U_1 = (1 + n) U_{1234}\tag{6.30}$$

The linear system is perturbed system from the trim point which has the input  $U_0$  .

Then

$$\Delta U_1 = (1 + n) U_0 - U_0 = U_0 n\tag{6.31}$$

The input  $U_3$  has the opposite sign.

The state-space description for the longitudinal model is

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \dot{u} \\ \dot{q} \\ \dot{\Omega}_1 \\ \dot{\Omega}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2\frac{lb}{I_y}\Omega_0 & -2\frac{lb}{I_y}\Omega_0 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ u \\ q \\ \Omega_1 \\ \Omega_3 \end{bmatrix} + 7U_0 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} n \quad (6.32)$$

With the values substituted then

$$\begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \dot{u} \\ \dot{q} \\ \dot{\Omega}_1 \\ \dot{\Omega}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -9.81 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3254 & -0.3254 \\ 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ u \\ q \\ \Omega_1 \\ \Omega_3 \end{bmatrix} + 3242 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} n = A_l \begin{bmatrix} x \\ \theta \\ u \\ q \\ \Omega_1 \\ \Omega_3 \end{bmatrix} + B_l n \quad (6.33)$$

### 6.5.3.1 Linear controller design

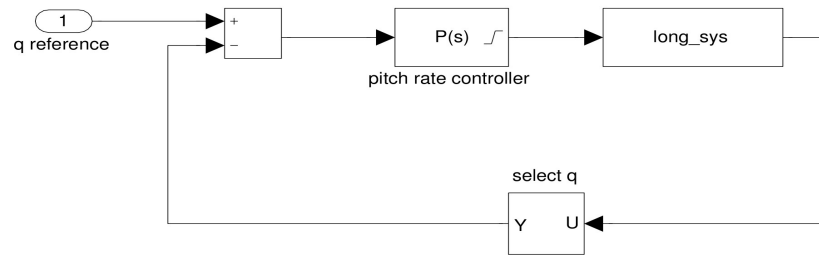
Considering system with the the pitch rate  $q$  as the output

$$y = [0 \ 0 \ 0 \ 1 \ 0 \ 0] \begin{bmatrix} x \\ \theta \\ u \\ r \\ \Omega_1 \\ \Omega_3 \end{bmatrix}^T = C_{lq} \begin{bmatrix} x \\ \theta \\ u \\ r \\ \Omega_1 \\ \Omega_3 \end{bmatrix}^T, \quad (6.34)$$

leads to the transfer function

$$G_{lq}(s) = C_{lq}(sI - A_l)^{-1} B_l \quad (6.35)$$

Closing the loop as shown in the following picture



Picture 6.7: Pitch rate control loop.

and doing the root locus, bode and step response analysis for the  $G_{lq}(s)$ . I found a reasonable gain for the P controller to be  $P_q = 0.019$ .

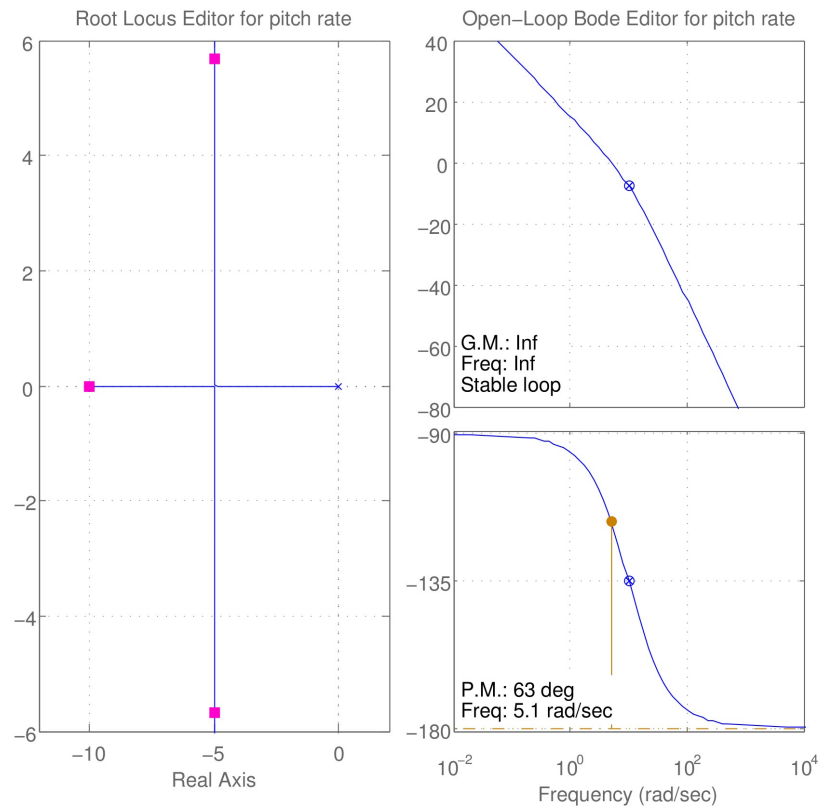


Figure 6.17: Root locus and bode plots for the pitch rate loop.

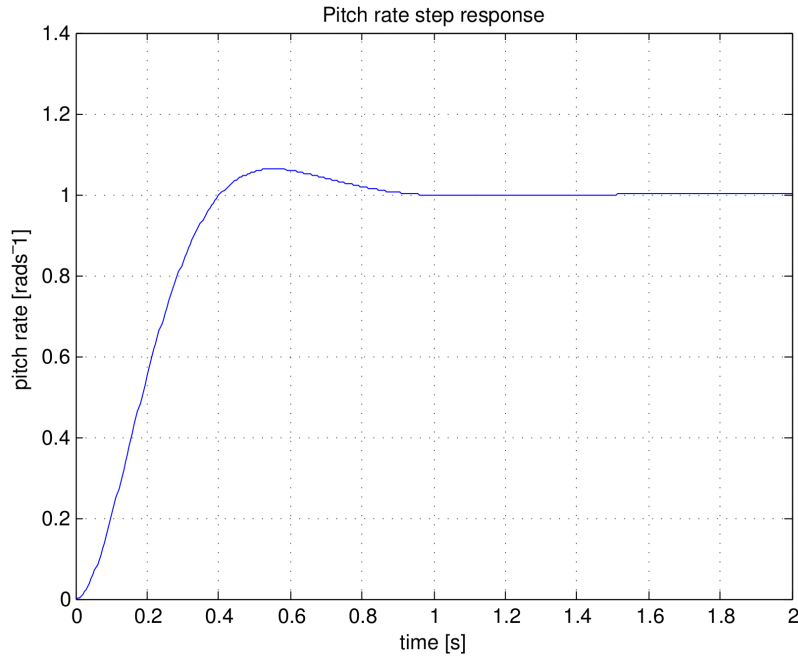


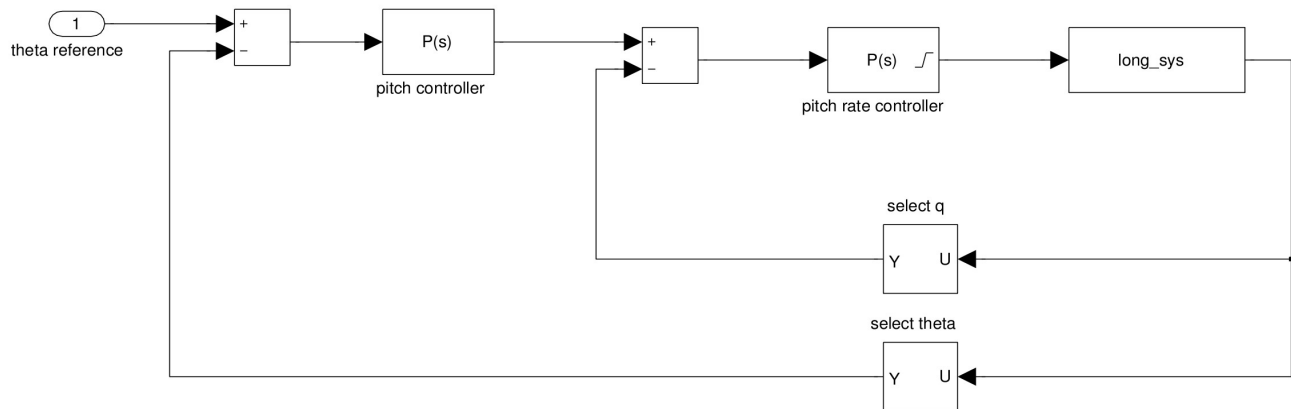
Figure 6.18: Step response of the pitch rate loop.

With the pitch rate control loop designed I stacked an pitch angle control loop on it. This means taking the closed loop transfer function with the pitch angle as the output

$$G_{l\theta}(s) = \frac{P_q G_{lq}(s)}{1 + P_q G_{lq}(s)} \frac{1}{s} \quad (6.36)$$

and designing the P controller for it. The controller's gain is  $P_\theta = 2.6$  .

Closing the loop as shown in the following picture



Picture 6.8: Pitch angle control loop.

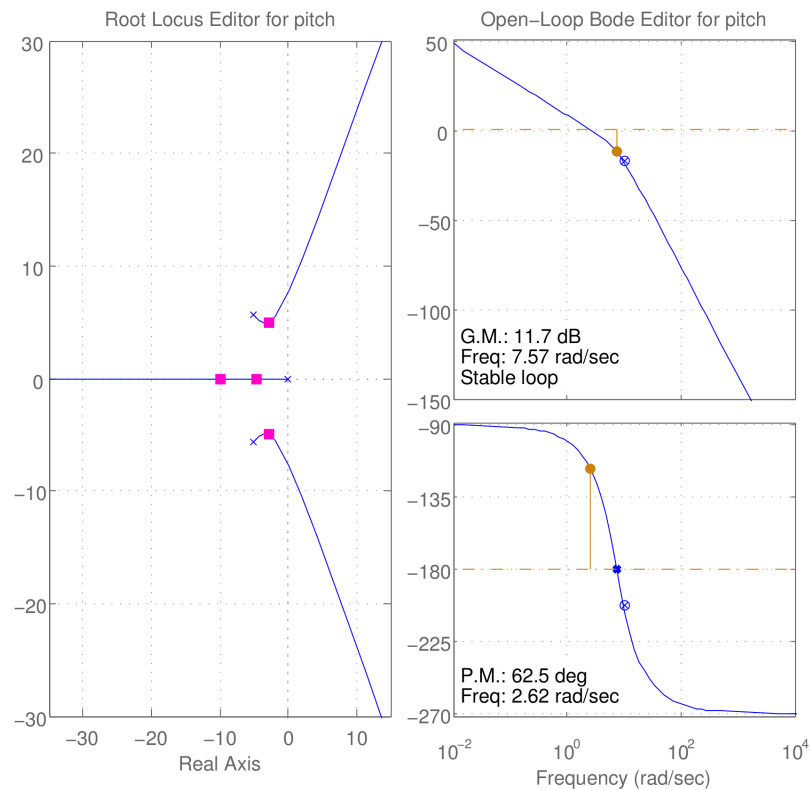


Figure 6.19: Root locus and bode plots for pitch angle loop.

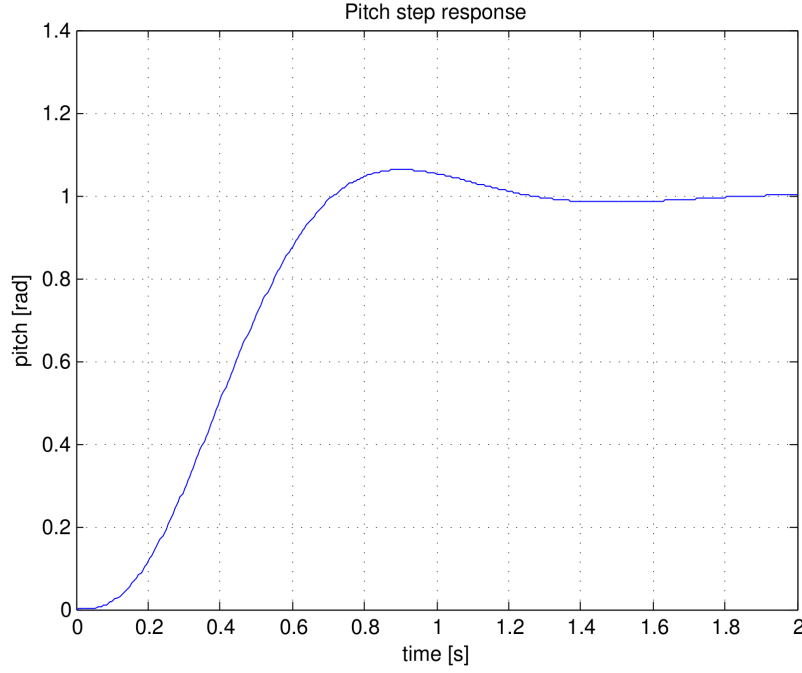


Figure 6.20: Step response of the pitch angle loop.

The pitch control loop has the transfer function

$$\frac{\theta(s)}{\theta_r(s)} = L_{l\theta}(s) = \frac{P_\theta G_{l\theta}(s)}{1 + P_\theta G_{l\theta}(s)} \quad , \quad (6.37)$$

where  $\theta_r$  is the reference for the pitch angle.

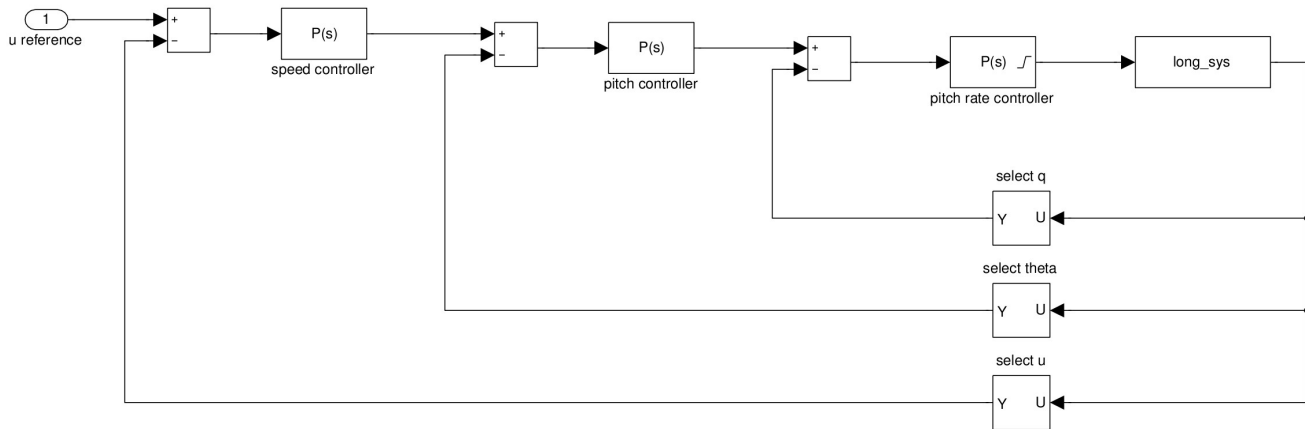
The transfer function from pitch angle to the longitudinal speed  $u$ , according to the equation (6.28) is

$$G_{lu}(s) = \frac{u(s)}{\theta(s)} = \frac{-g}{s} \quad (6.38)$$

Then

$$G_{lu}(s) = \frac{u(s)}{\theta_r(s)} = \frac{-g}{s} L_{l\theta}(s) \quad (6.39)$$

Designing the controller for this transfer function with gain  $P_u = -0.14$  , for the following control scheme



Picture 6.9: Longitudinal speed control loop.

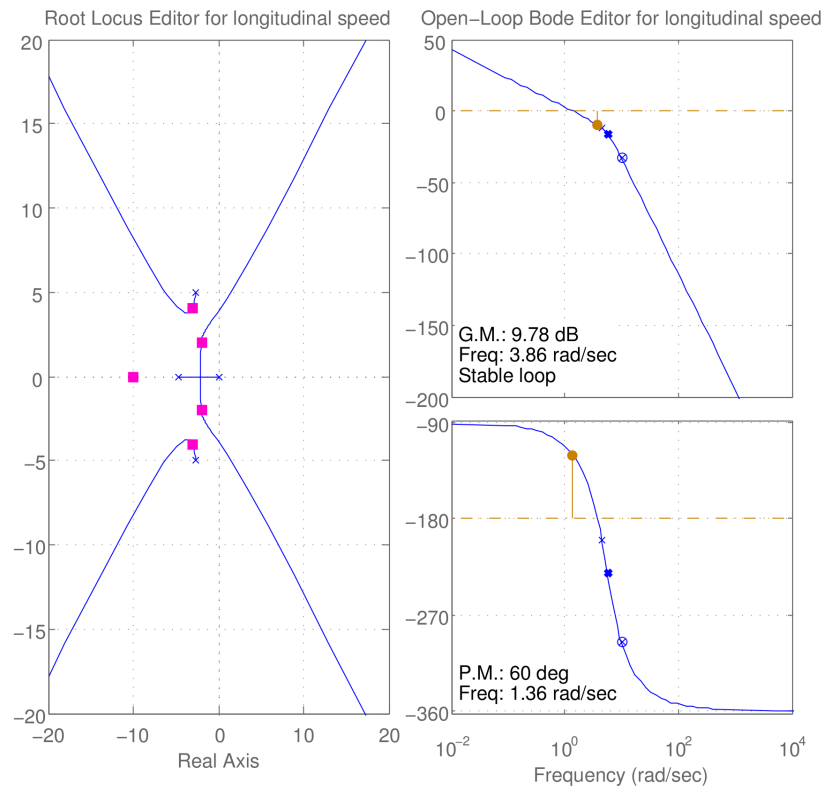


Figure 6.21: Root locus and bode plots for the longitudinal speed loop.



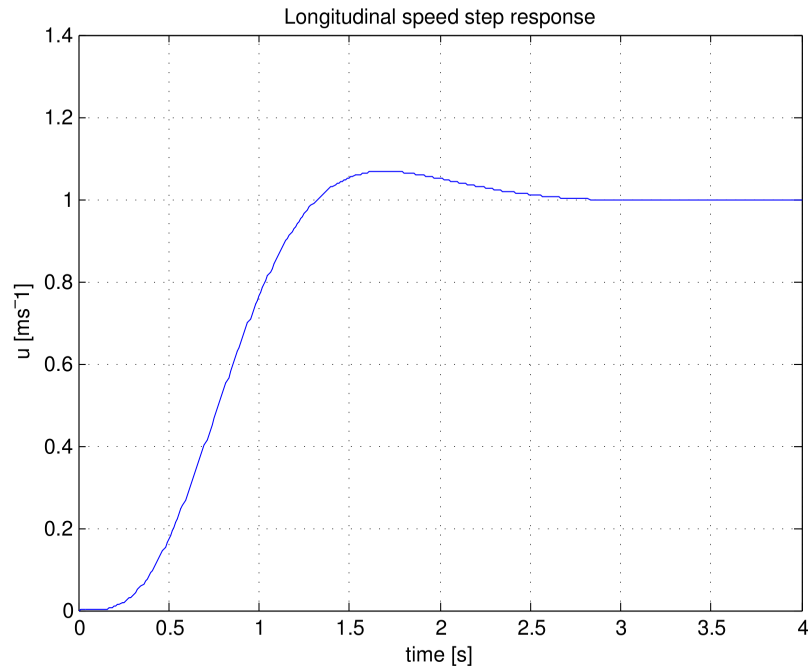
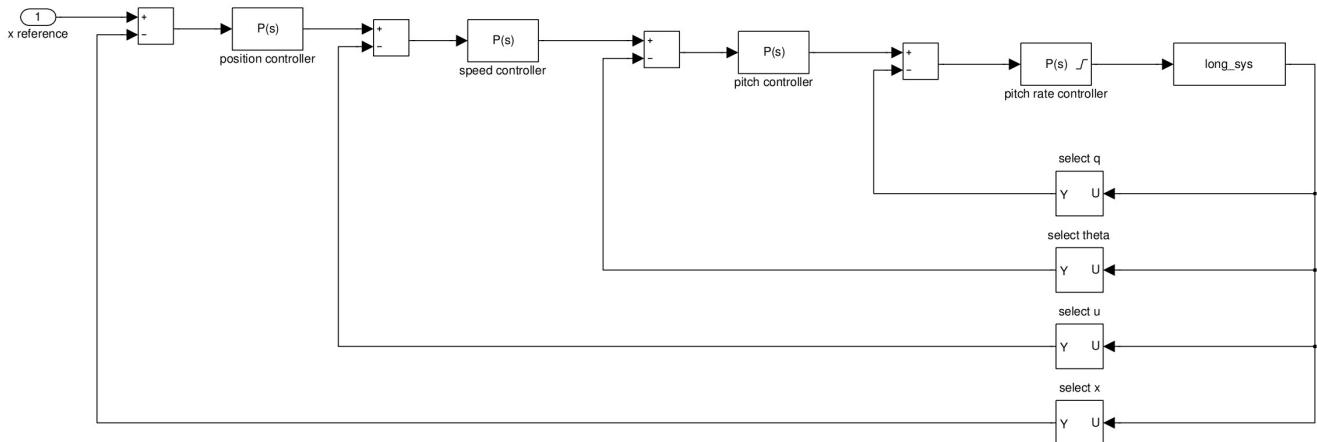


Figure 6.22: Step response of the longitudinal speed loop.

With the longitudinal speed control loop designed I stacked a longitudinal position  $x$  control loop on it. This means taking the closed loop transfer function with the position as the output

$$G_{lx}(s) = \frac{P_u G_{lu}(s)}{1 + P_u G_{lu}(s)} \frac{1}{s} \quad (6.40)$$

And designing the P controller for it. The controller's gain is  $P_x = 0.7$  for the control scheme on the picture.



Picture 6.10: Longitudinal position control loop.

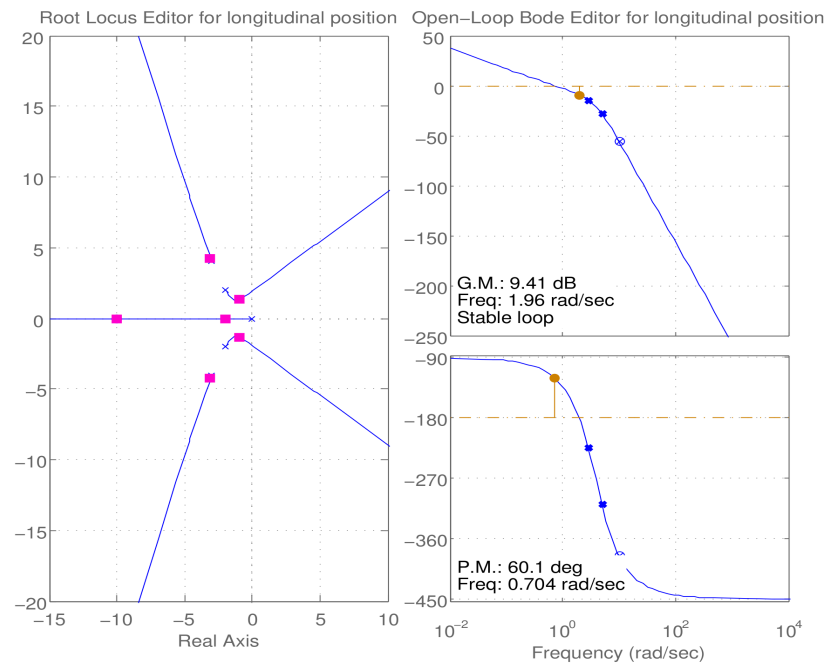


Figure 6.23: Root locus and bode plots for longitudinal position loop.

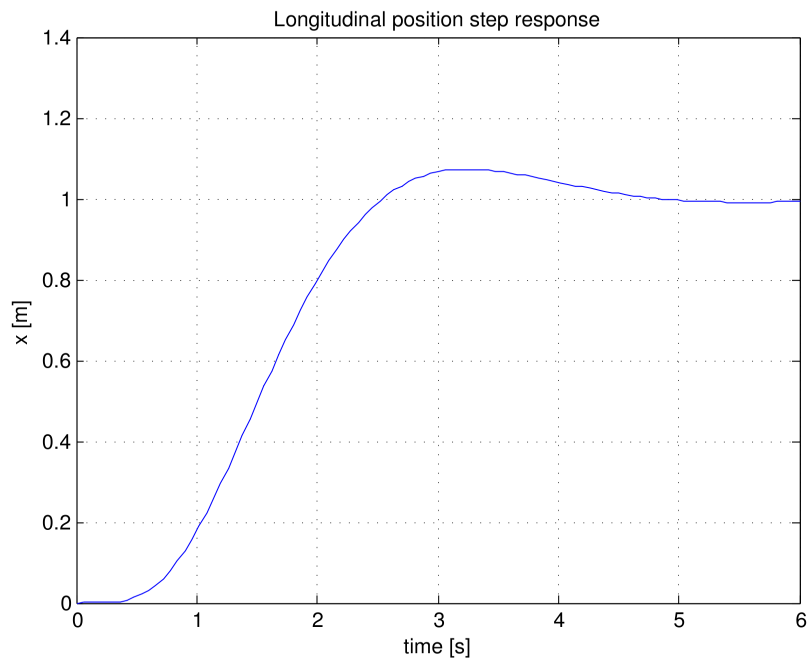


Figure 6.24: Step response of the longitudinal position loop.

### 6.5.3.2 Verification on nonlinear model

Controllers designed in previous chapter were tested on the nonlinear model to verify their performance using more realistic model.

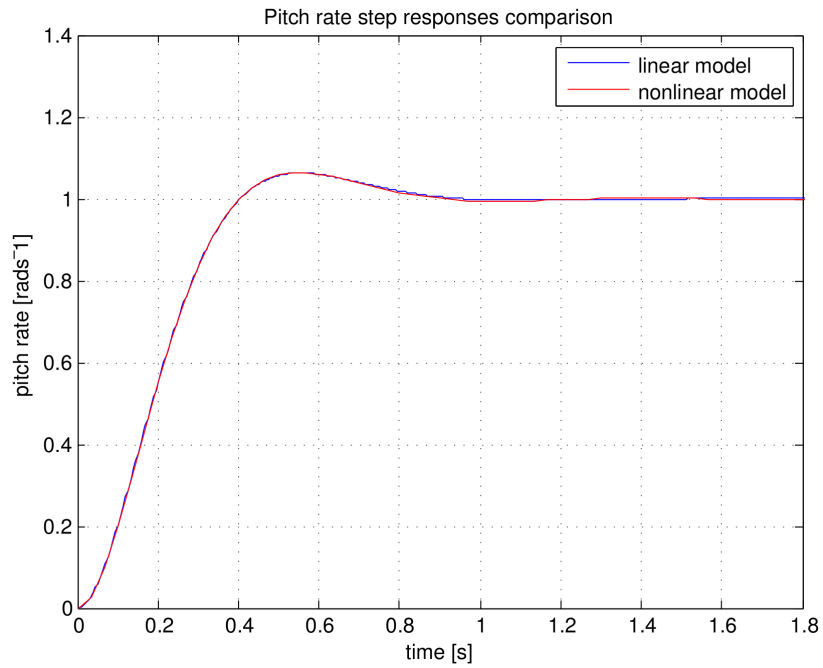


Figure 6.25: Pitch rate step responses comparison, linear and nonlinear plant control.

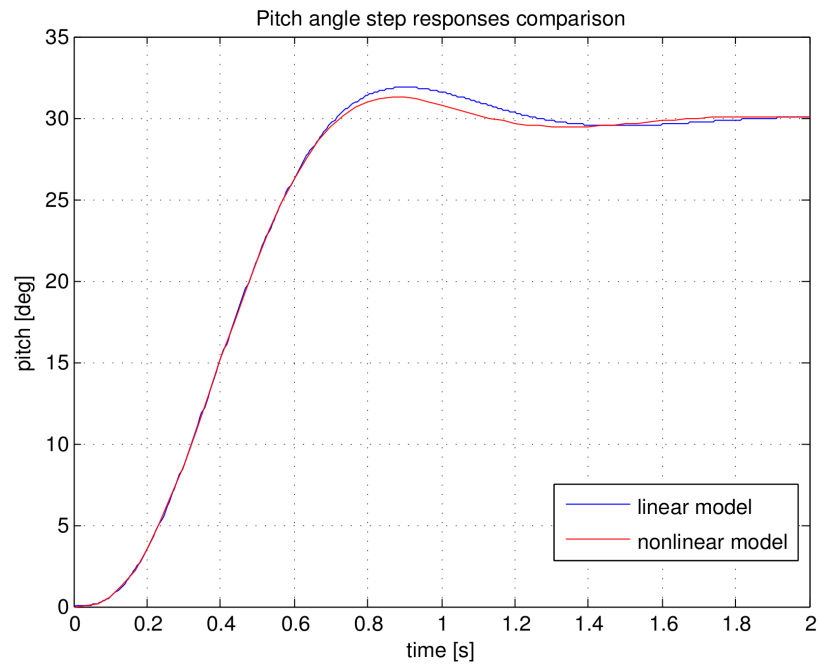


Figure 6.26: Pitch angle step responses comparison, linear and nonlinear plant control.

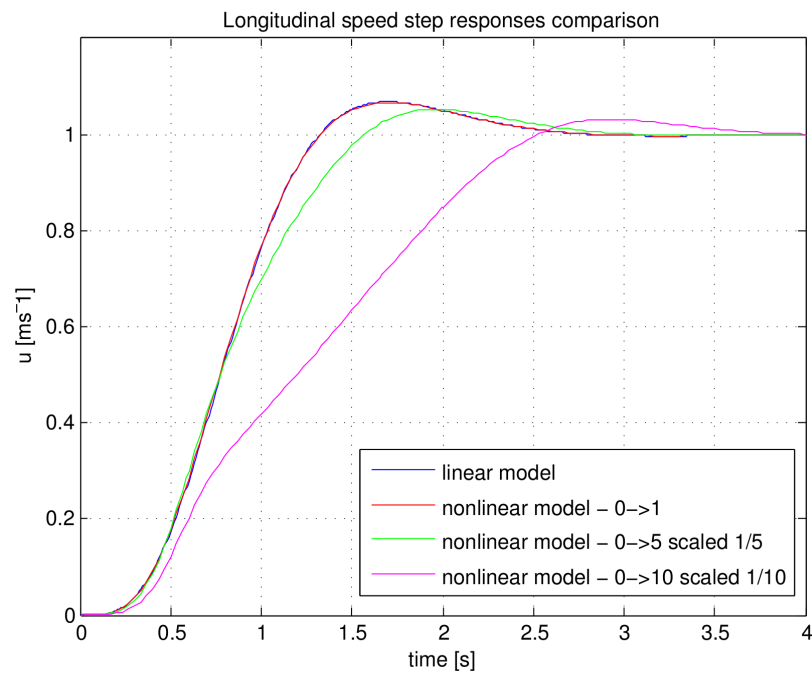


Figure 6.27: Longitudinal speed step responses comparison, linear and nonlinear plant control for different steps.

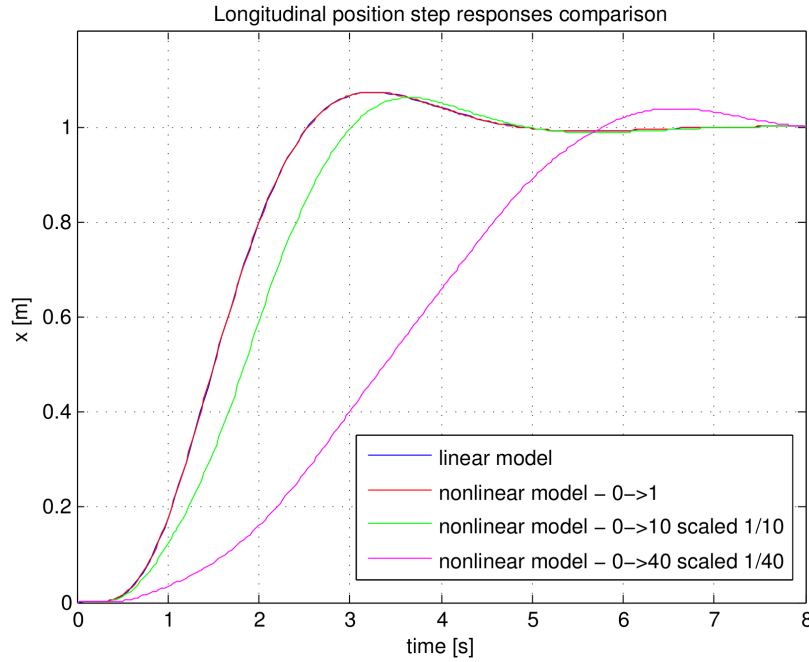


Figure 6.28: Longitudinal position step responses comparison, linear and nonlinear plant control for different steps.

As can be seen in previous figures pitch rate and pitch angle nonlinear responses are nearly the same as the responses of the linear model.

The speed response corresponds nicely for the step from 0 to 1. The other two steps are limited by the maximum pitch angle at 30 degrees. This is very obvious in the step from 0 to 10 where this limit provides only constant acceleration and the speed rises linearly.

The same applies to the position responses. There are two limits involved, the maximum pitch angle as well and a maximum speed limit. Again this is very prominent in the step from 0 to 40 where the pitch angle limit reduces the acceleration and then the maximum speed limit constrains the slope of the response.

#### 6.5.4 The lateral movement

The lateral movement is achieved when the quadrotor is keeping its vertical position  $z$ , longitudinal position  $x$  and its yaw angle. This type of movement is described by the equations

$$\begin{aligned}
\dot{y} &= v \\
\dot{\phi} &= p \\
\dot{v} &= g \phi \\
\dot{p} &= 2 \frac{lb}{I_x} \Omega_0 (\Omega_2 - \Omega_4) \\
\dot{\Omega}_i &= -10 \Omega_i + 7 U_i \quad i = \{2, 4\}
\end{aligned} \tag{6.41}$$

Because of the quadrotor's symmetry in the  $xz$  and  $yz$  planes this type of movement is completely identical to the longitudinal movement described in chapter 6.5.3. The only difference is that positive roll angle provides positive lateral acceleration which is opposite in the longitudinal situation where positive pitch angle leads to the negative longitudinal acceleration. This leads to positive gain for the lateral speed's controller.

## 6.6 Flight simulation

To fully evaluate the performance of the controllers I ran a simple flight simulation. Because of the coupling of different channels the controllers are influencing each other during the simulation. This leads to much aggressive and chaotic maneuvers so I had to tune the controllers to be more conservative. The tuning was done using the techniques described in chapter 6.5 and the performance during the more difficult maneuvers was checked using the nonlinear model.

controller	old setting	new setting
roll / pitch rate	0.019	0.009
yaw rate	0.120	0.120
roll / pitch angle	2.600	1.500
yaw angle	2.670	2.670
longitudinal speed	-0.140	-0.060
lateral speed	0.140	0.060
vertical speed	157.000	88.000
longitudinal / lateral position	0.700	0.300
vertical position	2.760	1.300

*Table 6.1: Comparison of the controllers' gains.*

Picture 6.11: Complete control scheme used in the simulation.



The initial conditions are

$$\begin{aligned} x=y=\phi=\theta=\psi=u=v=w=p=q=r=0 \\ z=-10 \end{aligned}$$

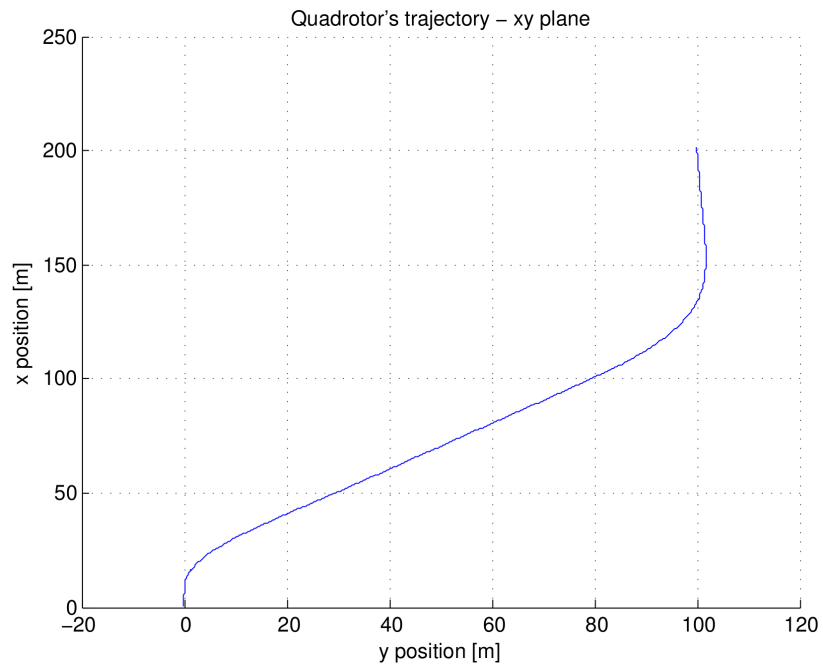
Limits for the controllers are set to  $|\phi| \leq \frac{\pi}{6}$   $|\theta| \leq \frac{\pi}{6}$   $|\dot{x}| \leq 10$   $|\dot{y}| \leq 10$   $|\dot{z}| \leq 10$

During the simulation I sent following commands to the quadrotor.

time [s]	command
1	goto [200 0 -10] m
3	goto [200 100 -10] m
9	goto [200 10 -30] m
12	change heading to 30 degrees

*Table 6.2: Mission plan.*

The simulation's results are shown in the following figures.



*Figure 6.29: Quadrotor's trajectory, projection on the xy plane.*

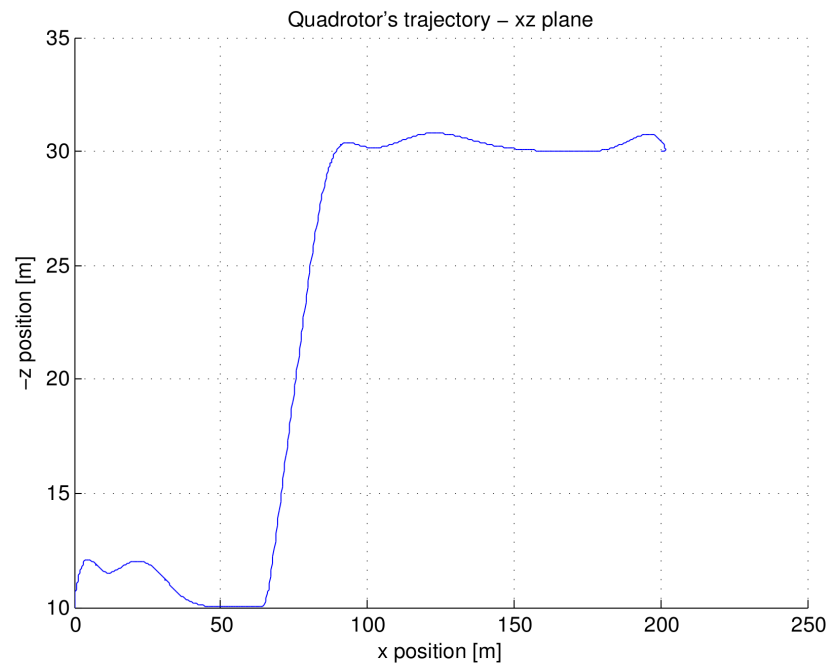


Figure 6.30: Quadrotor's trajectory, projection on the xz plane.

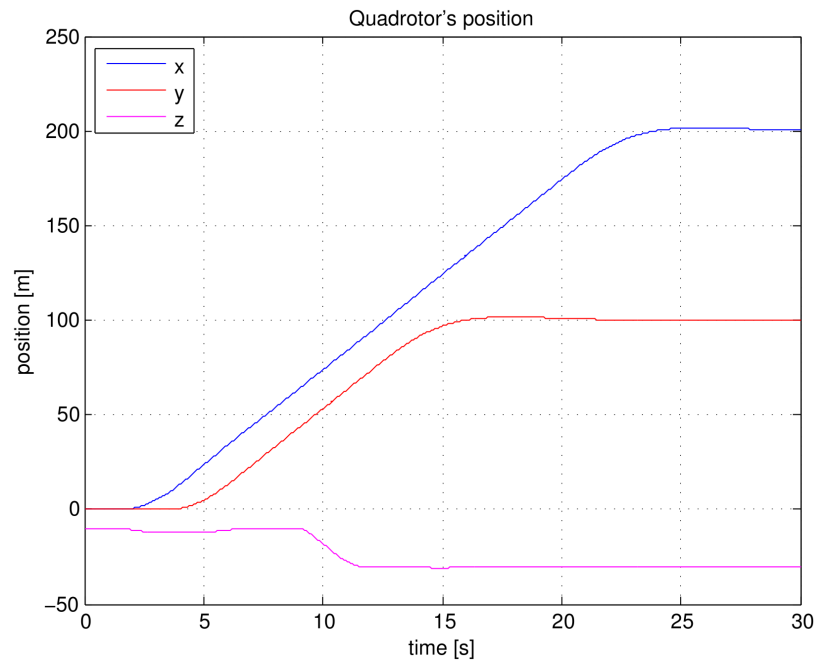


Figure 6.31: Quadrotor's position in time.

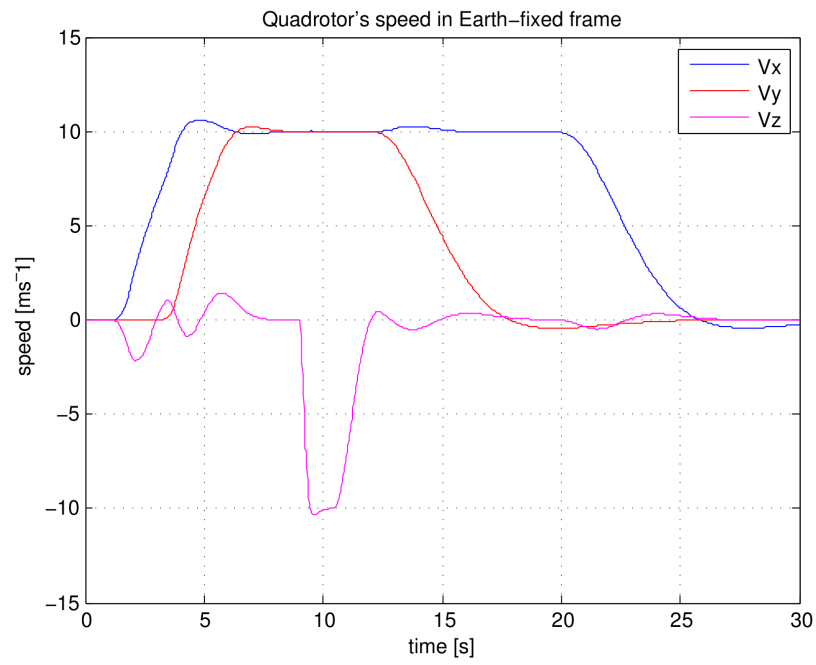


Figure 6.32: Quadrotor's speed in Earth's frame in time.

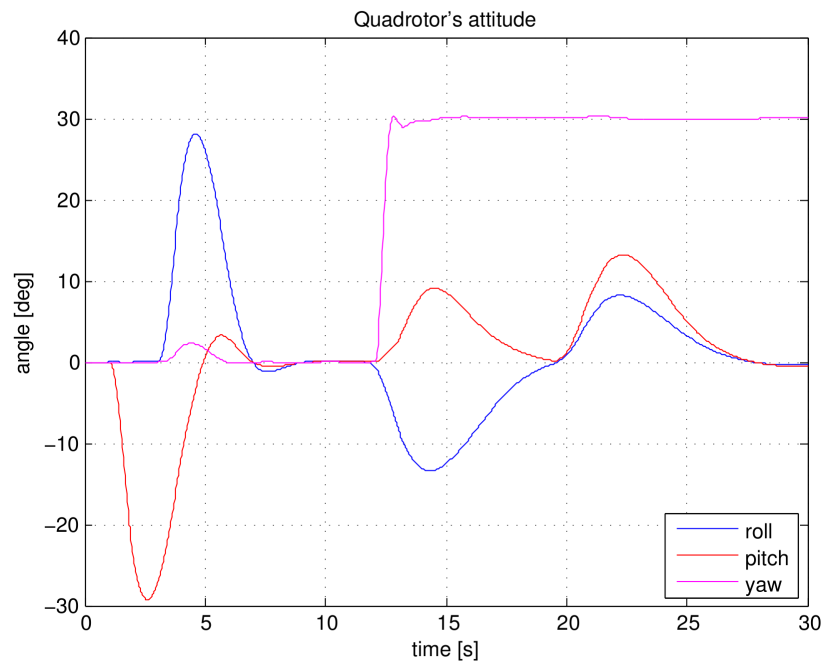


Figure 6.33: Quadrotor's attitude in time.

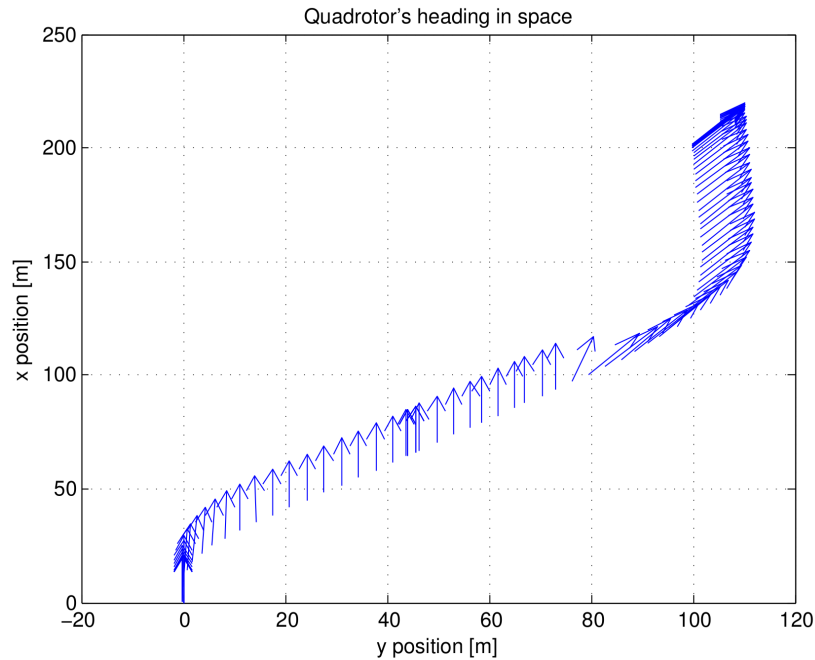


Figure 6.34: Quadrotor's heading in space, projection on  $xy$  plane.

## 7 Testing using the real model

With the controllers designed and tested on the nonlinear model I proceeded to tests using the real quadrotor. There are three ways how to implement the controllers:

- using the configuration dialogs in the LinkGS (software provided by the manufacturer for displaying the telemetry and sending configuration data to the quadrotor) which provides some simple PI loops configurations options (see chapter 7.1).
- the LinkGS is running a TCP server. It is possible to implement own control and then use the TCP connection to get the telemetry from the quadrotor and then feed back the commands.
- There is an interface between the Gumstix computers on board and the microcontrollers operating the quadrotor's HW. Then the program can be run by the Gumstix host.

At the time when this thesis was done the Gumstix option was not possible. The SW interface was not created then.

The option with the TCP host was available only short time before the submission and I did not have enough time to explore it. Surely there are issues involved such as the transportation lag and the bandwidth of the downlink channel from the quadrotor limiting the sampling rate of the sensor data.

At the time of writing this thesis the GPS data were not processed in the on board microcontrollers. There was an indoor position measurement system using camera in development at the department but it was not ready either. These issues left only the inertial navigation as an option. I discarded the idea of double integration of the acceleration to get the position which is very sensitive to the accelerometer's noise. I decided to only integrate one time to get the speed in the body-fixed frame. This left only the speed control loops and the loops beneath them to be implemented.

The first test of the inertial velocity measurement was promising. But then I ran into the problem with the subtracting the gravitational acceleration. The body-fixed frame velocity is the integral of the acceleration acting in the body frame. Because the body frame is not inertial frame of reference due to the gravitational acceleration this acceleration must be subtracted from the accelerometer's readings according the current attitude

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - D \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (7.1)$$

where  $a_x, a_y, a_z$  are the accelerations in the body frame measured by the accelerometers and  $D$  is the direction cosine matrix, see (2.4).

Unfortunately the Linkboard (the whole control circuitry in the quadrotor) firmware cannot perform this correction on its own which makes the velocity control not possible.

There is an option to control the vertical position using the pressure sensor. This is not possible either. The pressure sensor provides only the altitude value not the change of the altitude. This then leads to only one controller of the vertical movement system. As can be seen from the equation (6.13), this system contains two integrators. To control it using only one controller, the controller has to add some phase lead to compensate for the lag caused by integrators. There is no such a controller provided by the Linkboard right now.

This is the same situation with the heading control. There is a magnetometer which provides quite a nice measurement of the angular deviation from the magnetic north. But because of the lack of the  $D$  action it cannot be used. Fortunately the yaw gyro provides a good enough heading measurement thus the control is possible.

These complications left me only with the possibility to test the attitude controllers.

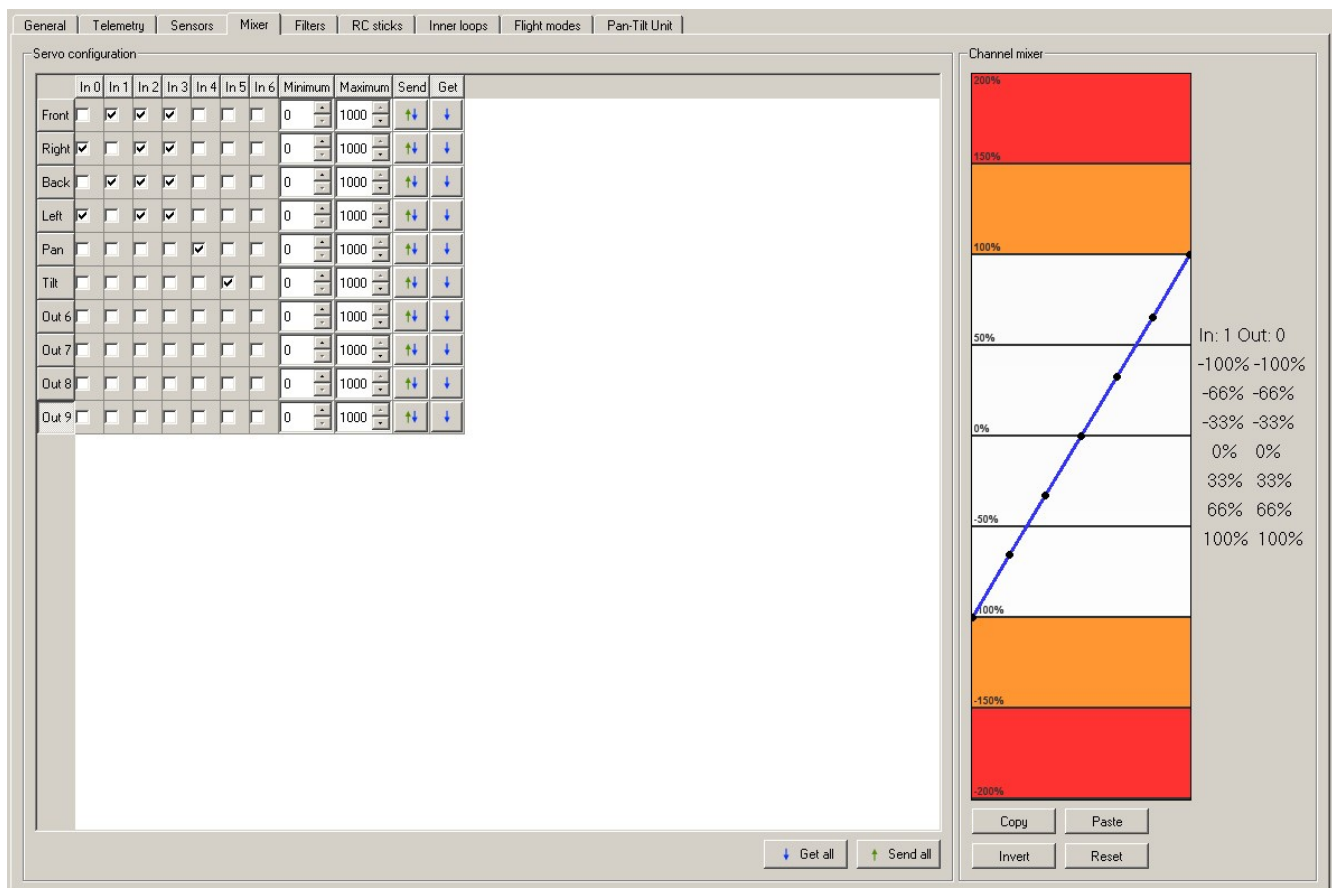
To configure the Linkboard using the LinkGS, there are three things to be done:

- configuring the mixer.
- configuring the inner loops.
- configuring the outer loops.

## 7.1 Linkboard configuration

This chapter describes how to set up the control loops using the LinkGS as a configuration tool.

### 7.1.1 Mixer configuration



Picture 7.1: Mixer configuration in LinkGS

The Linkboard mixer is mixing the control signals and creating the input signal for the engines. It works very similarly as my mixer described in chapter 6.3.

After inspecting the signals going to and leaving the mixer I found out how the mixer algorithm is working. The rows are representing the outputs of the mixer and are connected to the motors (Front-Right-Back-Left), camera pointing motors (Pan-Tilt) and some other outputs.

There are also six inputs feeding the filter. These inputs are selected from the Inner Loops, Outer Loops and Pan-Tilt Unit tabs. More on that later.

Then the inputs are simply added and the result is feed to appropriate output. It is possible to change the sign of the input by clicking on the appropriate selection box and then the Invert button. The sign of the input is represented by the slope of figure in the picture 7.1 . In the mentioned picture the Input 1 is added to the Output 0.

There is possibility to set limit on each output and send or request the settings from the quadrotor.

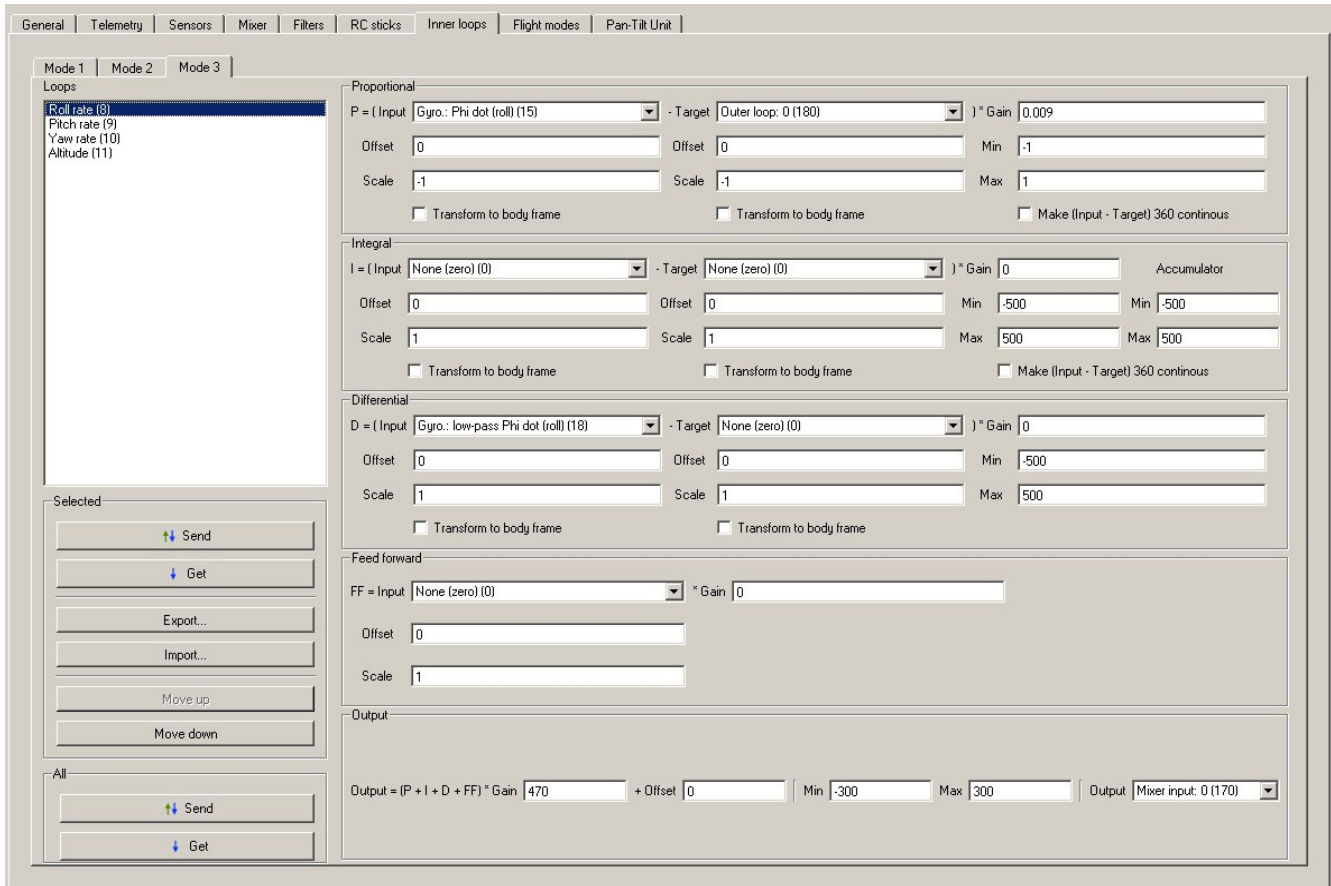
My configuration is in the following table

Motor	In0 (roll rate)	In1 (pitch rate)	In2 (yaw rate)	In3 (vertical speed)
Front		+	-	+
Right	-		+	+
Back		-	-	+
Left	+		+	+

*Table 7.1: Mixer configuration.*

There are no limits imposed on any of the motors. As can be seen from the table 7.1, the signs in table are in match with the signs present in the mixing equations (6.1) and (6.2).

## 7.1.2 Inner loops configuration



Picture 7.2: Roll rate controller configuration in LinkGS

The Inner loops are directly connected to the mixer and as their name implies they are the lowest loops in the control hierarchy. As shown in picture 6.1 these loops are used for the roll, pitch, yaw rate and vertical position control.

As can be seen in the picture 7.2 only the P part of the controller is used with the conservative gain mentioned in the chapter 6.6 and its output is limited within the  $\langle -1, 1 \rangle$  range as mentioned in chapter 6.3. The Output 0 signal is the control signal from the roll angle control loop. The overall gain located in the bottom of the screen is set to 463 which is the value of  $\Omega_0$ .

As mentioned before the Linkboard's mixer is only adding and subtracting the mixed values which differs from the mixer used in this thesis which actually performs multiplying. In the hover trim point these mixers are equivalent. The same applies for all motors only the sign of the input differs. This is sorted by the mixer settings in 7.1.



There is small limit imposed on the signal going to the filter. This limitation prevents the engines to stop completely, when it is hard to start them moving again.

As can be seen from the picture 7.2 there are three tabs named Mode 1, Mode 2 and Mode 3. These tabs are representing an option to set up different controllers for the different flight modes. The flight modes are selected using the flight mode switch, which is located at RC. There is no need to use different settings hence all tabs are set up equivalently.

The pitch rate controller is set up in the same manner.

The yaw rate controller setup is very similar. The designed controller was too aggressive for the real model so it was tuned down. That is the reason for the overall gain being just 50 not 463 as should be.

General | Telemetry | Sensors | Mixer | Filters | RC sticks | Inner loops | **Flight modes** | Pan-Tilt Unit

Mode 1 | Mode 2 | Mode 3

Loops

- Roll rate (8)
- Pitch rate (9)
- Yaw rate (10)
- Altitude (11)

Selected

Send

Get

Export...

Import...

Move up

Move down

All

Send

Get

Proportional

P = ( Input: Gyro.: Psi dot (yaw) (17) ) - Target: Outer loop: 2 (182) ) \* Gain: 0.12

Offset: 0 Offset: 0 Min: -500 Max: 500

Scale: -1 Scale: -1

☐ Transform to body frame ☐ Transform to body frame ☐ Make (Input - Target) 360 continuous

Integral

I = ( Input: None (zero) (0) ) - Target: None (zero) (0) ) \* Gain: 0 Accumulator

Offset: 0 Offset: 0 Min: -500 Max: 500

Scale: 1 Scale: 1

☐ Transform to body frame ☐ Transform to body frame ☐ Make (Input - Target) 360 continuous

Differential

D = ( Input: Gyro.: low-pass Psi dot (yaw) (20) ) - Target: None (zero) (0) ) \* Gain: 0

Offset: 0 Offset: 0 Min: -500 Max: 500

Scale: 1 Scale: 1

☐ Transform to body frame ☐ Transform to body frame

Feed forward

FF = Input: None (zero) (0) \* Gain: 0

Offset: 0

Scale: 1

Output

Output = (P + I + D + FF) \* Gain: 50 + Offset: 0 Min: -400 Max: 400 Output: Mixer input: 2 (172)

Picture 7.3: Yaw rate controller configuration in LinkGS

As has been stated in the beginning of the chapter 7 , it is not possible to control the vertical position now, hence the altitude loop is set up only as feedforward controller with gain 1. This leaves the vertical speed and position control in the hands of the pilot.

Mode 1 | Mode 2 | Mode 3

Loops

- Roll rate (8)
- Pitch rate (9)
- Yaw rate (10)
- Altitude (11)

Selected

Send

Get

Export...

Import...

Move up

Move down

All

Send

Get

Proportional

P = ( Input: None (zero) (0) - Target: None (zero) (0) ) \* Gain: 0

Offset: 0

Scale: 1

Transform to body frame

Integral

I = ( Input: None (zero) (0) - Target: None (zero) (0) ) \* Gain: 1

Offset: 0

Scale: 1

Transform to body frame

Differential

D = ( Input: None (zero) (0) - Target: None (zero) (0) ) \* Gain: 0

Offset: 0

Scale: 1

Transform to body frame

Feed forward

FF = Input: RC stick: Throttle (32) \* Gain: 1

Offset: -100

Scale: 1

Output

Output = (P + I + D + FF) \* Gain: 1 + Offset: 0

Min: 0 Max: 1000

Output: Mixer input: 3 (173)

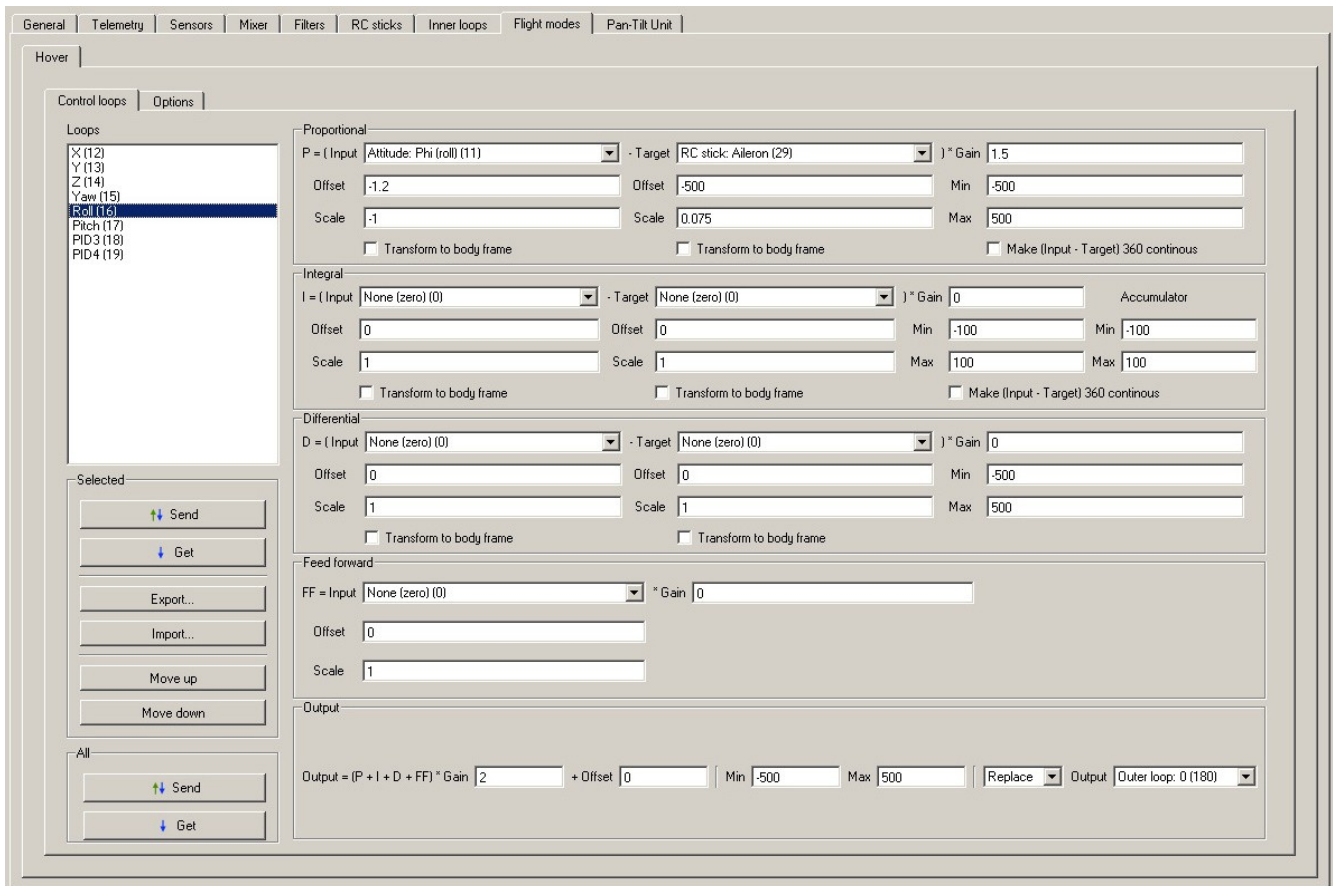
Picture 7.4: Vertical speed feedforward setting in LinkGS

### 7.1.3 Flight modes – Outer loops

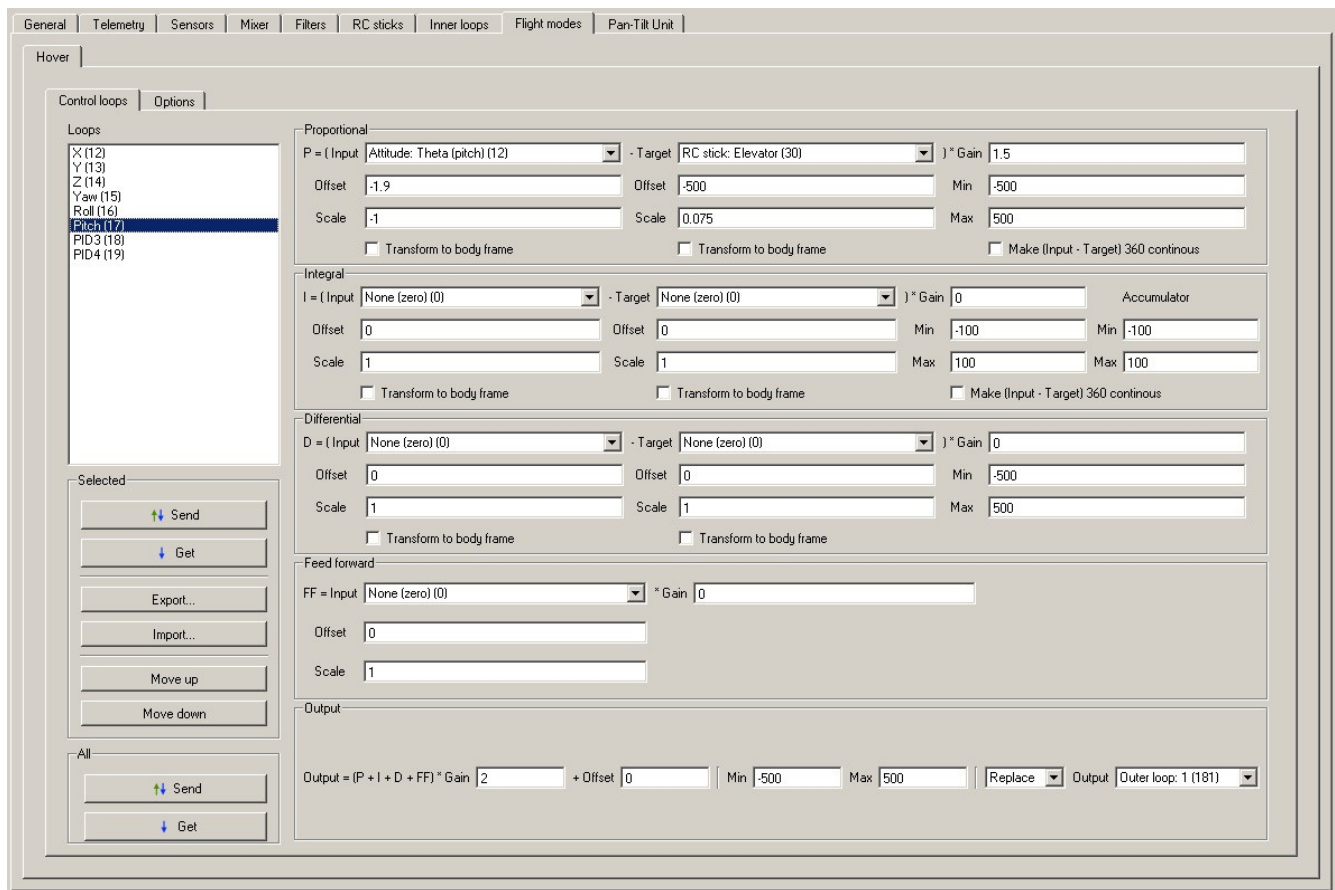
The Flight modes tab contains other eight controllers. These controllers can be connected to each other and/or to the inner loops, they cannot be connected to the mixer directly. These controllers are activated only when the flight mode switch is flipped to the Auto flight position.

This tab is used to implement the roll, pitch and yaw angle controllers.

The reference inputs are the RC sticks for aileron, elevator and rudder respectively. The range of the signals is from 100 to 900 with 500 when in neutral position. The 500 neutral bias is subtracted and aileron, elevator sticks are multiplied by 0.075 and the rudder stick by 0.45 to provide maximum 30 and 180 degrees reference respectively.

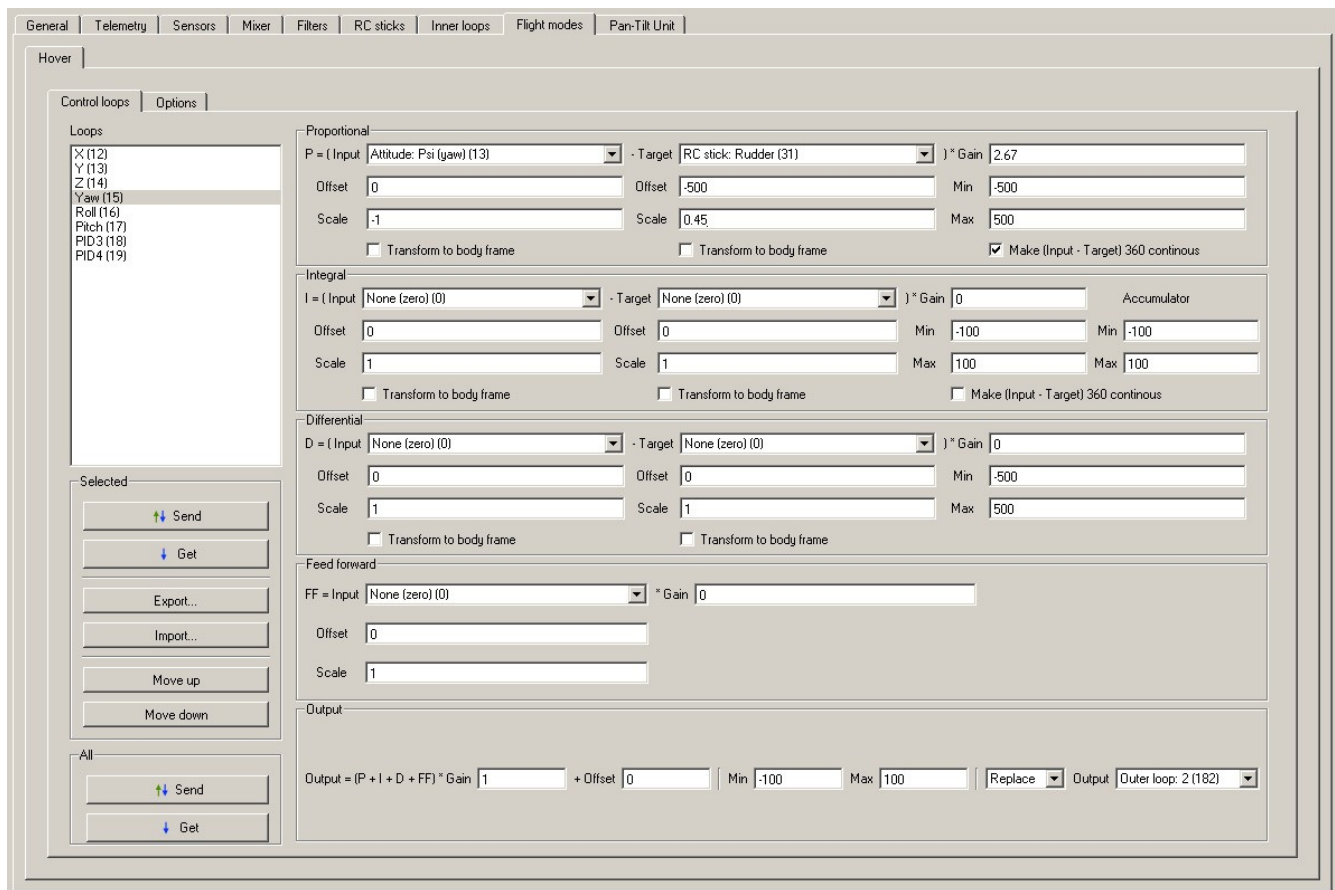


Picture 7.5: Roll angle controller configuration in LinkGS.



Picture 7.6: Pitch angle controller configuration in LinkGS.

The roll and pitch controllers were tuned to be more aggressive using the overall gain of 2.



Picture 7.7: Yaw angle controller configuration in LinkGS.

The option “Make (Input-Target) 360 continuous” transforms the yaw measurement from  $<0,360)^{\circ}$  range to  $(-180,180)^{\circ}$  range.

## 7.2 Flight test

Using the settings from the preceding chapter I ran few flight tests to evaluate the performance of the controllers. The tests were performed mostly indoors and one time in the outdoor environment.

### 7.2.1 Indoors tests

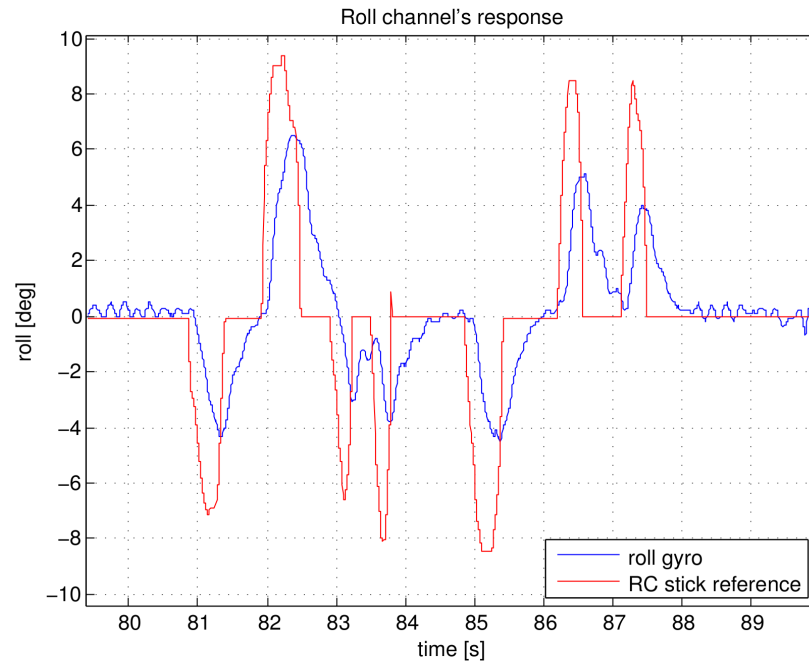


Figure 7.1: Roll channel's response in time.

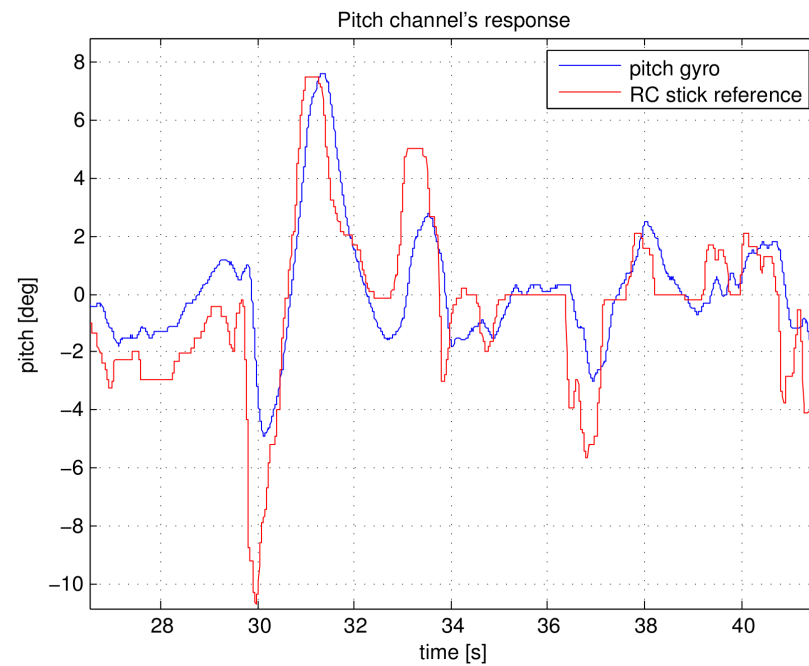


Figure 7.2: Pitch channel's response in time.

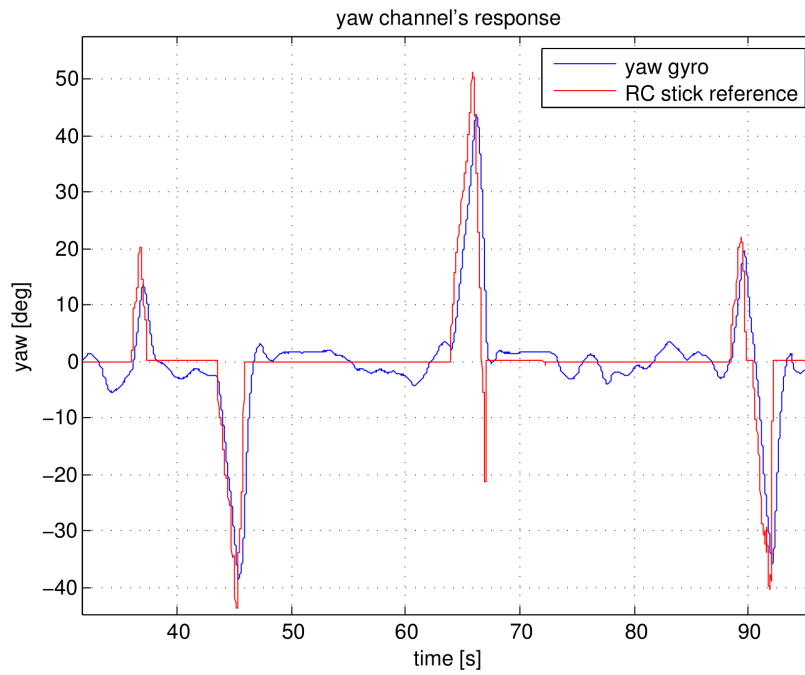


Figure 7.3: Yaw channel's response in time.

The figures are taken from different tests and different times. This was done because each test was focused on different behavior so I picked up only the interesting data.

### 7.2.2 Outdoor test

Outdoor test was performed outside of the university in park. There was wind blowing which was acting as disturbance at the output.

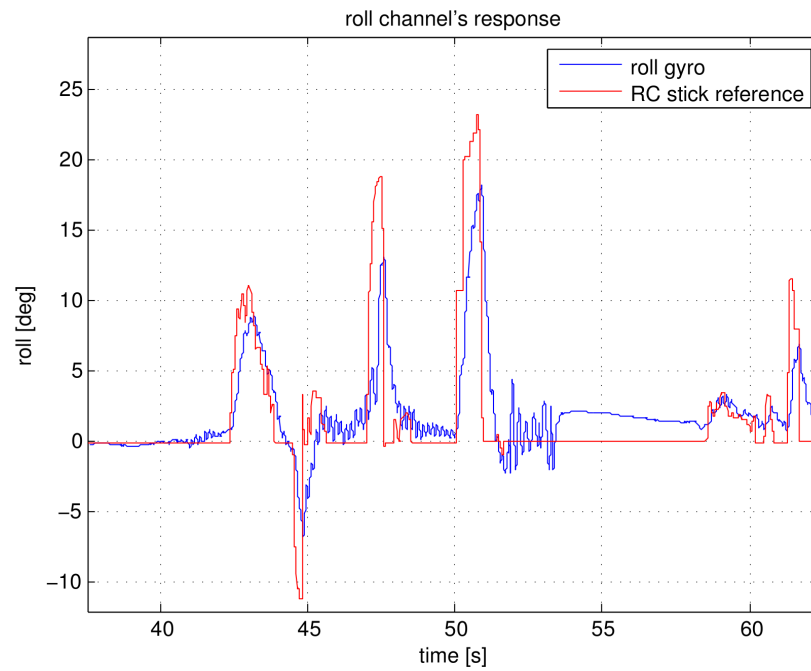


Figure 7.4: Roll channel's response in time.

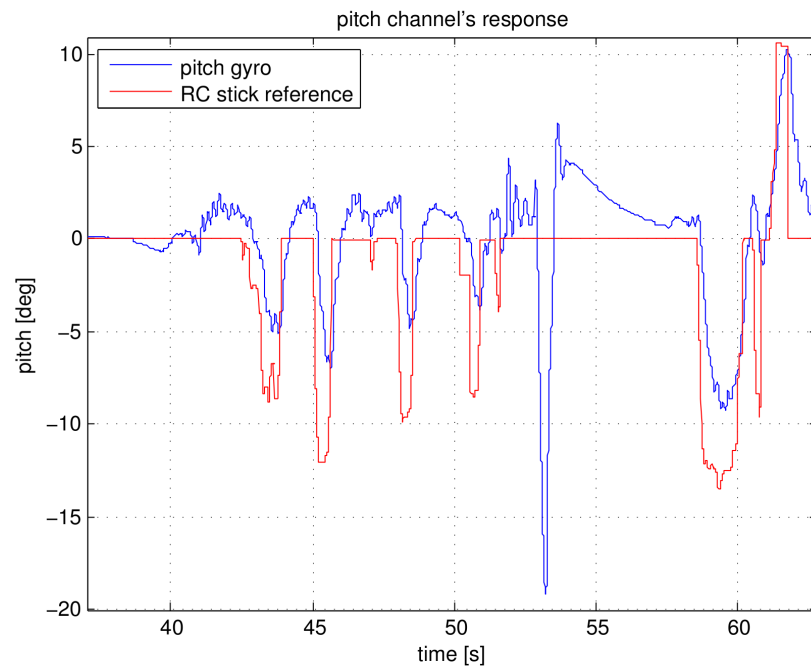


Figure 7.5: Pitch channel's response in time.



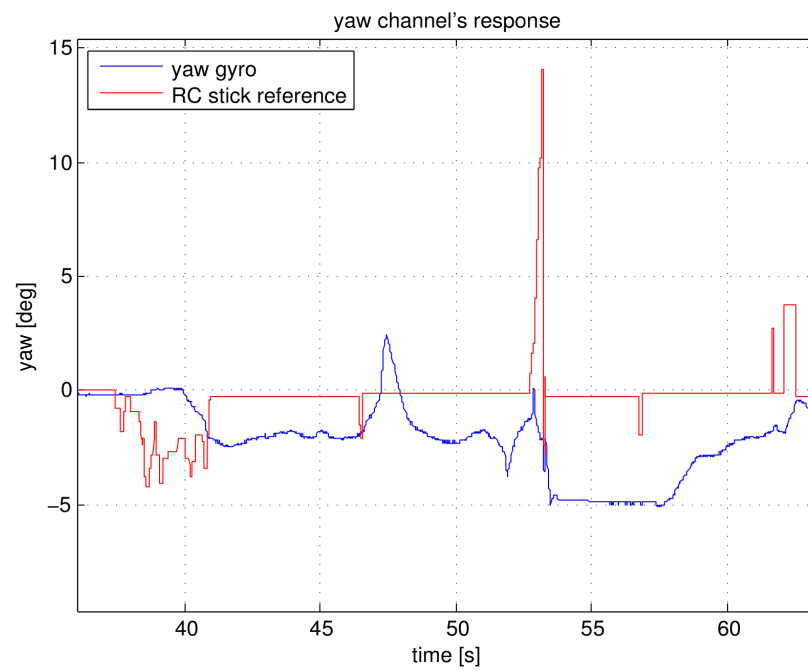


Figure 7.6: Yaw channel's response in time.

## 8 Conclusions and future work

As can be seen throughout the thesis the assigned goals were achieved to some extent. I successfully analyzed the existing hardware and derived the mathematical model. Identification and measurement of the model's parameters were challenging issues where I had to improvise. The lack of any measurement devices forced me to create my own improvised tools to get the necessary data. This was of course reflected in the quality of the measured and evaluated data. This can be seen most obviously for the drag coefficient  $d$  which has in reality much greater magnitude as can be seen from the tuning of the yaw rate controller when testing with the real model.

Even though the nonlinear mathematical model is fully functional it must be noted that it is not a true representation of the reality. The most problematic areas are the aerodynamic forces and torques which are almost fully neglected. This poses no problem when the indoor low speeds flights are considered but for faster outdoor flying one has to calculate with these simplifications.

Unfortunately it was not possible to implement more modern controllers such as the LQ optimal regulator or controllers synthesized using the  $H^\infty$  minimization. This led to pure proportional controllers' design. It was very interesting to see that even those very simple controllers are able to stabilize and even provide robust performance when a suitable architecture is chosen. The comparison between LQ and P regulator was carried out and evaluated. The LQR provides faster and smoother response but the difference is not dramatic.

Although a complete position control system was developed and simulated it was not possible to run the tests with the real system. Due to the lack of features provided by the original program and firmware and not enough time to implement my own external algorithm, it was only possible to test the attitude control. Unfortunately the inertial velocity stabilization and control was not possible to implement at all.

The indoor tests were successful and the provided attitude control gives us satisfactory results. The tests in the outdoor environments introduced the problem with the external disturbance caused by the wind. Even though the controllers were able to stabilize the attitude in the windy environment the disturbance rejection should be studied and further improved.

Even though the goals of the thesis were not fully reached the contribution of this thesis is fully functional nonlinear mathematical model and linear systems representing the quadrotor's movement

which can be used as a development environment and testbed for future work.

The last thing to be noted is that even though the position control is developed and functional it needs more testing on the real system. One has not to forget that the controllers are designed for the hover trim point. When more aggressive or more robust control is desired e.g. recovering from extreme positions, more trim points have to be introduced and some gain scheduling algorithm implemented to select or interpolate suitable control law for the given flight envelope. Once the interface for the Gumstix computers is operational this feature is possible. Then more advanced control algorithms can be implemented as well, such as already mentioned LQR and  $H_\infty$  minimization or model predictive control (MPC) algorithm as a higher level control and planning platform. This algorithm can use the already developed inner loops as a low level control interface providing optimal control therefore lowering the power consumption and improving the performance.

## 9 Enclosed CD's content

- **thesis.pdf** - This thesis.
- **def.mat** – File containing constants used in the other MATLAB files.
- **nonlin\_model.mdl** – Simulink scheme with the nonlinear mathematical model and with the complete control system.
- **lin\_modes.m** – MATLAB script with the linear systems and the designed linear controllers.
- **rpm\_fast.pde** – Arduino sketchbook file for the angular rate measurement. (C source file designated for the Arduino compiler)
- **fg.sh** – Bash script for starting the Flighgear simulator with the needed parameters.

## 10 Bibliography

- Antsaklis, P.J. & Michel, A.N., 2007. *A Linear Systems Primer* 1st ed., Birkhäuser Boston.
- Blakelock, J.H., 1991. *Automatic Control of Aircraft and Missiles* 2nd ed., Wiley-Interscience.
- Bouabdallah, S., Murrieri, P. & Siegwart, R., 2005. Towards Autonomous Indoor Micro VTOL. *Autonomous Robots*, 18(2).
- Castillo, P., Lozano, R. & Dzul, A., 2005. Stabilization of a mini rotorcraft with four rotors. *Control Systems, IEEE*, 25(6), p.45- 55. Available at: [Accessed May 25, 2011].
- Chen, M. & Huzmezan, M., 2003. A Simulation Model and H( Loop Shaping Control of a Quad Rotor Unmanned Air Vehicle. In *Modelling, Simulation, and Optimization*. pp. 320-325.
- Cook, M.V., 1997. *Flight Dynamics Principles* 1st ed., John Wiley & Sons.
- De Lellis, Marcelo, 2011. *Modeling, Identification and Control of aQuadrotor Aircraft*. Czech Technical University in Prague.
- Dorf, R.C. & Bishop, R.H., 2007. *Modern Control Systems* 11th ed., Prentice Hall.
- Franklin, G., Powell, J.D. & Emami-Naeini, A., 2005. *Feedback Control of Dynamic Systems* 5th ed., Prentice Hall.
- Hoffmann, G.M. et al., 2007. Quadrotor helicopter flight dynamics and control: Theory and experiment. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. p. 1–20.
- Kivrak, A., Ö., 2006. *Design of control systems for quadrotor flight vehicles*. Mechatronics Engineering Department: Atılım University.
- Serway, R.A. & Jewett, J.W., *Physics for Scientists and Engineers - 6th edition* 6th ed., Thomson; Brooks Cole - Custom.
- Skogestad, S. & Postlethwaite, I., 1996. *Multivariable Feedback Control: Analysis and Design* 1st ed., Wiley.
- Stepaniak, M.J., 2008. *A Quadrotor Sensor Platform*. Ohio University.