

Master of Science Thesis in Automatic Control  
Department of Electrical Engineering, Linköping University, 2016

# Model-based Design, Development and Control of an Underwater Vehicle

**Adam Aili and Erik Ekelund**



Master of Science Thesis in Automatic Control

**Model-based Design, Development and Control of an Underwater Vehicle**

Adam Aili and Erik Ekelund

LiTH-ISY-EX--16/4979--SE

Supervisor:     **Jonas Linder**  
                         ISY, Linköping University  
                         **Rikard Hagman**  
                         Combine Control Systems

Examiner:       **Daniel Axehill**  
                         ISY, Linköping University

*Division of Automatic Control  
Department of Electrical Engineering  
Linköping University  
SE-581 83 Linköping, Sweden*

Copyright © 2016 Adam Aili and Erik Ekelund

## Abstract

With the rising popularity of ROVs and other UV solutions, more robust and high performance controllers have become a necessity. A model of the ROV or UV can be a valuable tool during control synthesis. The main objective of this thesis was to use a model in design and development of controllers for an ROV.

In this thesis, an ROV from Blue Robotics was used. The ROV was equipped with 6 thrusters placed such that the ROV was capable of moving in 6-DOFs. The ROV was further equipped with an IMU, two pressure sensors and a magnetometer. The ROV platform was further developed with EKF-based sensor fusion, a control system and manual control capabilities.

To model the ROV, the framework of Fossen (2011) was used. The model was estimated using two different methods, the prediction-error method and an EKF-based method. Using the prediction-error method, it was found that the initial states of the quaternions had a large impact on the estimated parameters and the overall fit to validation data. A Kalman smoother was used to estimate the initial states. To circumvent the problems with the initial quaternions, an EKF was implemented to estimate the model parameters. The EKF estimator was less sensitive to deviations in the initial states and produced a better result than the prediction-error method. The resulting model was compared to validation data and described the angular velocities well with around 70 % fit.

The estimated model was used to implement feedback linearisation which was used in conjunction with an attitude controller and an angular velocity controller. Furthermore, a depth controller was developed and tuned without the use of the model. Performance of the controllers was tested both in real tests and simulations. The angular velocity controller using feedback linearisation achieved good reference tracking. However, the attitude controller could not stabilise the system while using feedback linearisation. Both controllers' performance could be improved further by tuning the controllers' parameters during tests.

The fact that the feedback linearisation made the ROV unstable, indicates that the attitude model is not good enough for use in feedback linearisation. To achieve stability, the magnitude of the parameters in the feedback linearisation were scaled down. The assumption that the ROV's center of rotation coincides with the placement of the ROV's center of gravity was presented as a possible source of error.

In conclusion, good performance was achieved using the angular velocity controller. The ROV was easier to control with the angular velocity controller engaged compared to controlling it in open loop. More work is needed with the model to get acceptable performance from the attitude controller. Experiments to estimate the center of rotation and the center of gravity of the ROV may be helpful when further improving the model.



## Acknowledgments

First of all, would we like to thank our supervisor Jonas Linder for his support, inspirational ideas and for not being afraid to use his red ballpoint pen.

We would also like to thank Rikard Hagman and Combine for the opportunity to write this thesis.

We thank Albin and Cornelis for their company and the never ending supply of "quality" coffee.

Lastly, we would like to thank Ljungsbro fritidscenter and Linköpings simhall for letting us test the ROV in their pools.

*Linköping, June 2016  
A A and E E*



---

# Contents

<b>Notation</b>	<b>xi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Purpose . . . . .	2
1.3 Objective . . . . .	2
1.4 Methodology . . . . .	3
1.5 Exclusions . . . . .	3
1.6 Thesis Outline . . . . .	3
<b>2 Hardware and Software</b>	<b>5</b>
2.1 BlueROV Package . . . . .	5
2.2 ROV I/O . . . . .	7
2.3 Power . . . . .	7
2.4 The Onboard Computer . . . . .	7
2.5 The Workstation . . . . .	9
<b>3 Modelling the ROV</b>	<b>11</b>
3.1 Coordinate Systems and Kinematics . . . . .	13
3.2 Rigid-Body Kinetics . . . . .	17
3.3 Hydrodynamics . . . . .	18
3.4 Hydrostatics . . . . .	19
3.5 Actuators . . . . .	20
3.6 Model in Component Form . . . . .	21
3.7 Simplified Model Structures . . . . .	22
<b>4 Sensor Fusion</b>	<b>25</b>
4.1 The Extended Kalman Filter . . . . .	25
4.2 Motion Model . . . . .	26
4.3 Measurement Equations . . . . .	28
<b>5 Parameter Estimation</b>	<b>31</b>
5.1 Data Collection and Processing . . . . .	32

5.2	Prediction-Error Method . . . . .	32
5.3	Estimation Using Prediction-Error Method . . . . .	36
5.4	Extended Kalman Filter Estimation . . . . .	41
5.5	Estimation Using an Extended Kalman Filter . . . . .	42
5.6	Investigating Low Fit in p Dynamics . . . . .	42
5.7	Estimated Parameters . . . . .	46
<b>6</b>	<b>Controlling the ROV</b>	<b>49</b>
6.1	Open-Loop Control . . . . .	50
6.2	Feedback Linearisation . . . . .	51
6.3	Attitude Controller . . . . .	51
6.4	Angular Velocity Controller . . . . .	52
6.5	Depth Controller . . . . .	53
6.6	Benchmarking . . . . .	55
6.7	Discussion . . . . .	61
<b>7</b>	<b>Conclusions and Future Work</b>	<b>65</b>
7.1	Conclusions . . . . .	65
7.2	Future Work . . . . .	66
<b>A</b>	<b>Dependencies and Installation</b>	<b>69</b>
A.1	Raspberry Pi Installation . . . . .	69
A.1.1	ROS Installation . . . . .	69
A.1.2	Tether Setup . . . . .	70
A.1.3	Installation of the ROV Package . . . . .	70
A.2	Workstation Installation . . . . .	71
A.2.1	ROS Installation . . . . .	71
A.2.2	Tether Setup . . . . .	71
A.2.3	Installation of the ROV Package . . . . .	72
<b>B</b>	<b>Operation of the ROV</b>	<b>73</b>
B.1	Wiring . . . . .	73
B.2	Start up of the ROV . . . . .	73
B.3	Shutdown of the ROV . . . . .	74
B.4	Operating the ROV . . . . .	74
B.4.1	Xbox Mode . . . . .	75
B.4.2	GUI Mode . . . . .	75
B.4.3	Logging Data . . . . .	75
B.4.4	Displaying the Continuous Plots . . . . .	76
B.4.5	LED Lights on the HKPilot . . . . .	76
B.5	New Parameter Estimation and New Controllers . . . . .	76
B.5.1	New Parameter Estimation . . . . .	77
B.5.2	New Controllers . . . . .	77
B.6	Known Issues and Troubleshooting . . . . .	77
B.6.1	Calibration of Sensors . . . . .	77
B.6.2	ROS Debugging . . . . .	78
B.6.3	Check Ethernet Connection . . . . .	78

---

B.6.4 One or Several Thrusters Are Unresponsive . . . . .	78
B.6.5 Checking and Changing Rotation of Thrusters . . . . .	79
<b>C Thruster Mapping</b>	<b>83</b>
<b>D Controller Test Results</b>	<b>85</b>
<b>Bibliography</b>	<b>99</b>
<b>Index</b>	<b>101</b>



---

# Notation

## ABBREVIATIONS

Abbreviation	Description
CB	Center of buoyancy.
CG	Center of gravity.
CO	Center of origin.
DOF	Degrees of freedom.
EKF	Extended Kalman filter.
ESC	Electronic speed controller.
IMU	Inertial measurement unit.
I/O	Input/Output.
KF	Kalman filter.
MPC	Model predictive control.
PID	Proportional, integral, differential (regulator).
PI	Proportional, integral (regulator).
ROS	Robot Operating System.
ROV	Remotely operated vehicle.
RPM	Rotations per minute.
SLAM	Simultaneous localisation and mapping.
SNR	Signal to noise ratio.
TDOA	Time difference of arrival.
UV	Unmanned vehicle.

---



# 1

---

## Introduction

This is the master's thesis *Model-based Design Development and Control of an Underwater Vehicle*. This master's thesis was performed at Combine in Linköping, Sweden.

### 1.1 Background

During recent years there have been an explosive growth in popularity and public availability of drones and unmanned vehicles (UVs) (Cutler, 2013). With this increased popularity, some new remotely operated underwater vehicles (ROVs) have been made available for public purchase. There have been new releases like the BlueROV from Blue Robotics (BlueRobotics, 2016) and the Trident from Open ROV (OpenROV, 2016). With open source products like the aforementioned ROVs being readily available, the subject of underwater navigation and control has become more and more relevant to hobbyists and enthusiasts.

ROVs have a large area of application and commercial ROVs are at the present time used for inspection of naval structures, seabed examination, underwater welding, ship cleaning, object location and recovery (SAAB, 2016). The open source products are more oriented towards exploration. It was of special interest for us to investigate how the control systems of an open source ROV solution, in this thesis the BlueROV from Blue Robotics, could be developed via model-based design and control. The possibility of autonomous operation and underwater positioning was also of interest.

Since a typical ROV solution has 6 degrees of freedom (DOF) and most often is not decoupled, it is advantageous to use a control system when executing advanced manoeuvres during exploration and missions. The controller structure originally

implemented in the BlueROV platform was an open-loop controller with ad hoc decoupling. This type of control is somewhat capable during manual operation with low requirements on accuracy but might be too inexact in autonomous and more delicate operations.

Autonomous operation places special requirements on a control system. This is due to safety and precision requirements during operation (Podofillini et al., 2015, p. 416). To meet these needs, a model-based control strategy might be used which, however, needs a good model of the system. A model can be created via some base knowledge of the system and the underlying physics, via system identification or a combination of both (Ljung and Glad, 2004).

A typical UV uses a GPS unit to estimate its position and to improve the velocity estimates. Unfortunately, GPS signals quickly lose strength in underwater environments, which in turn places extra importance in how system identification of ROV platforms is performed.

## 1.2 Purpose

The purpose of this master's thesis were to show how model-based design development could be used to implement a robust control system for a ROV. The result of this thesis will also be an input for future work regarding control of nautical vehicles.

## 1.3 Objective

The objectives of this thesis were to develop a model of an ROV and to use the model for developing a robust control system for the ROV.

### Sub-Objectives

To get a better overview, the objective have been divided into the following sub-objectives:

- Assemble the ROV.
- Develop a framework for changing controllers in the ROV.
- Estimate a model of the ROV.
- Create a plant model of the ROV in MATLAB/Simulink.
- Develop a robust model-based controller and evaluate its performance with simulations and tests.

## 1.4 Methodology

At first, a theoretical study of the ROV's model was performed. A literature study was also performed to gain experience from earlier studies in this field. Then a plan for estimating the model parameters was formulated. Different approaches of system identification were tested and well-thought-out before experiments were conducted. The parameter estimation was iterated several times using several methods to get well-estimated parameters.

The ROV's computer system was built on top of the Robot Operating System (ROS) using several different packages. A list of dependencies is available in Appendix A. The software was implemented with the divide and conquer method, *i.e.* the software nodes were implemented stepwise with increasing complexity. The different nodes in the system had only basal communication in the beginning and were developed to contain more complex functions, such as sensor fusion and controllers.

Different predetermined tests were conducted to evaluate the different controllers against each other. The controllers were finely tuned before the tests and thus the most suitable controller/controllers were found.

## 1.5 Exclusions

Since no absolute position measurements are available on the ROV platform, this thesis only concerns parameter estimation and control in attitude and angular velocities.

## 1.6 Thesis Outline

In Chapter 2, the ROV and its components are described. The ROV's capabilities, physical limitations and its performance are explained and the operating system on which the ROV platform is built on is briefly explained.

The necessary prerequisites for parameter estimation, namely sensor fusion and modelling, are explained in Chapter 3 and Chapter 4. In Chapter 3, the coordinate systems used to model the ROV are displayed and kinematic relations are presented. Furthermore, the complete ROV model is presented step by step. The effects of rigid-body kinetics, hydrostatics, hydrodynamics and the ROV's actuators are modelled separately and then combined to produce the complete 6-DOF model. Lastly in Chapter 3, ways of simplifying the model to improve identifiability and to reduce the number of parameters to be estimated are presented together with assumptions that have been made in order for the changes in the model to hold.

In Chapter 4, an implementation of the chosen method of sensor fusion is explained. Two different motion models are presented together with measurements

equations for each ROV sensor. Moreover, methods of outlier rejection are presented for the sensors.

In Chapter 5, the general idea and purpose of parameter estimation is explained. Two different ways of estimating parameters are presented, estimation using the prediction-error method and estimation using an EKF with an expanded state vector. Furthermore, the process of collecting and preprocessing data for parameter estimation is presented. Results using the different methods and different model structures are presented with parameter values. Comparisons between simulations of the model and validation data are presented along with the fit-values. Lastly, the advantages and drawbacks of the different approaches and models are discussed. Also a set of parameters are chosen for controller design.

In Chapter 6, the control problem is briefly described and different ways of state control and exact linearisation are presented. The controllers implemented on the ROV are further explained and results from tests and simulations are presented. Additionally, problems encountered during tests and ideas for solving these issues are brought up and discussed.

The thesis is concluded in Chapter 7 and areas of improvement of the ROV platform are suggested.

# 2

---

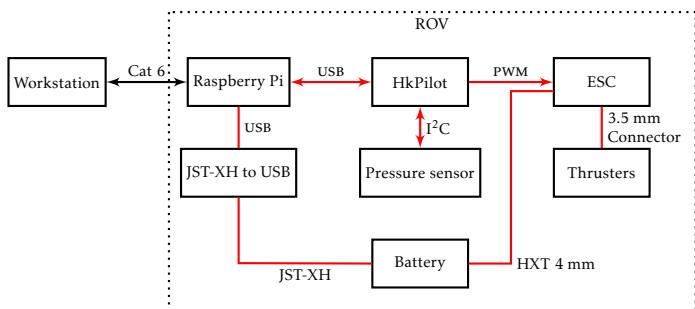
## Hardware and Software

The goal of this chapter is to present the capabilities and limitations of the hardware used in this thesis. The hardware is divided into different sections and are described in moderate detail, for example, how things are connected and controlled.

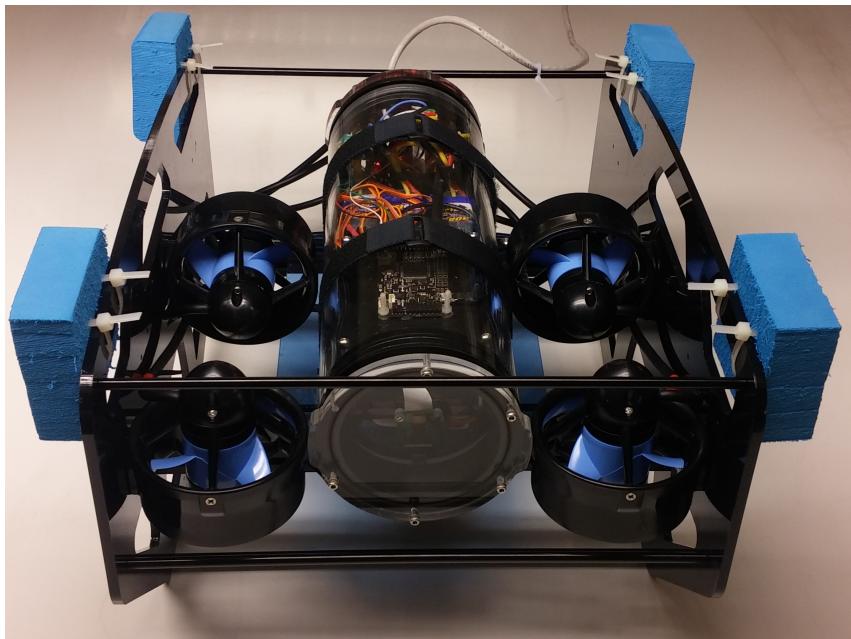
The ROV frame and thrusters were included in a package from Blue Robotics. In addition to the ROV frame, a Raspberry Pi was used as an onboard computer and a HKPilot Mega 2.7 was used as an input/output (I/O) unit, see Figure 2.4. The software in the ROV was built on top of ROS. Instructions for installation of software and operation of the ROV can be found in Appendices A and B. ROS is an open source operating system for robot applications. The operating system provides message passing and hardware abstraction, thus simplifies communication between different computers (ROS, 2016). The message passing in ROS consists of two parties, subscribers and publishers. When a publisher sends a message on a specific topic any subscribers that listen to that topic receives the message. Figure 2.1 shows a schematic over the ROV components and their connections.

### 2.1 BlueROV Package

Figure 2.2 shows the BlueROV from Blue Robotics used in this thesis. The BlueROV package includes an acrylic chassis, an acrylic tube, six electronic speed controllers (ESCs), six Blue Robotics T200 thrusters, cable penetrators and a cradle for mounting of electronics.



**Figure 2.1:** Schematic of how ROV components communicate (arrows) and how they are powered (red).



**Figure 2.2:** A frontal view of the BlueROV from Blue Robotics that was used in the thesis. Note the four blue squares made of EVA foam mounted in the corners for extra buoyancy.

## 2.2 ROV I/O

The ROV's I/O consists of an HKPilot Mega 2.7 which is based on Ardupilot Mega. The HKPilot Mega 2.7 has the following on chip sensors

- Magnetometer - HMC5883L.
- Barometer - MS5611-01BA.
- Inertial measurement unit (IMU) - MPU6000.

An external pressure sensor MS5837-30BA which was encased in a watertight case by Blue Robotics was connected to the HKPilot Mega 2.7 by I<sup>2</sup>C. The HKPilot Mega 2.7 also controls the six ESCs. The ESCs are 30A AfroESCs flashed with Blue Robotics linearising firmware. The HKPilot Mega 2.7 is connected to the onboard computer by USB cable. The HKPilot Mega 2.7 runs a rosserial-arduino node which is a simpler ROS node that communicates with a master node by serial communication. Scaling and calibration of the sensors are done automatically. However, the offset calibration of the magnetometer and accelerometer has to be performed manually by following the instructions that are produced in the workstation terminal window when the calibration script is run. The external pressure sensor uses the internal barometer to remove the atmospheric pressure offset. The atmospheric pressure offset is measured once, at the start up of the ROV.

## 2.3 Power

To power the ROV a Turnigy 5000mAh 4S 25C Lipo Pack was used. This is a high discharge battery which ensures that all thrusters can be run at the same time without disruptive voltage drops. To power the Raspberry Pi 2, a HobbyKing LiPo to USB Charging Adapter was used. This adapter connects to the JST-XH connector on the LiPo battery and then outputs regular USB voltages and currents. A USB to micro-USB adapter was used to route the power to the Raspberry Pi. The ESCs are powered via the main lead of the LiPo battery. Lastly, the HKPilot Mega 2.7 is powered via USB by the Raspberry Pi **and** by the ESCs.

## 2.4 The Onboard Computer

The onboard computer was a Raspberry Pi 2 Model B which can be seen in Figure 2.3. A Raspicam was connected to the Raspberry Pi 2 and was used in conjunction with a ROS node to create a video feed. The ROS nodes running on the onboard computer can be seen in Table 2.1.

**Table 2.1:** The different nodes that run on the onboard computer.

Node	Description
roscore	Node that handles the ROS backend.
raspicam_node	Camera node for streaming video from the ROV.
controller	Node that can run different controllers.
rosserial	Serial node for communication with the HKPilot Mega 2.7.



**Figure 2.3:** The Raspberry Pi 2 Model B, the onboard computer, is shown to the left and the raspicam is shown on the right.



Figure 2.4: The HKPilot Mega 2.7 used for I/O.

## 2.5 The Workstation

The workstation used in the thesis is a Lenovo T430 with an Intel®i5-3210M processor and Intel®HD Graphics 4000. The workstation was connected via a Cat 6 tether to the Raspberry Pi 2. The different ROS nodes that are run on the workstation can be seen in Table 2.2.

Table 2.2: The different nodes that run on the workstation.

Node	Description
heartbeat	Node for checking the connection with the HKPilot Mega 2.7.
teleop_xbox	Xbox node for handling inputs from the Xbox controller.
joy	A joystick node for interacting with the OSS USB inputs.
rqt	A GUI for the ROV.
sensorfusion	The sensor fusion node.



# 3

---

## Modelling the ROV

This chapter describes how the ROV is modelled using the framework of Fossen (2011). A model describes how an object is affected by forces such as gravity and friction. Models can be derived by using different physical laws. Assumptions can often be made how the physical properties of the object affect the model and a simpler model can be derived. Assumptions can be symmetry of the object, no coupling inertia, low speed and several others.

An underwater vehicle with 6 DOF can be described by

$$\dot{\eta} = J(\eta)\nu \tag{3.1}$$

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \tag{3.2}$$

where  $\eta$  are generalised positions and  $\nu$  are generalised velocities that are used to describe motion in 6 DOF (Fossen, 2011, p. 15). The matrices  $M$ ,  $C(\nu)$ ,  $D(\eta)$  and the vector  $g(\nu)$  respectively describes how inertia, Coriolis forces, damping forces, gravity and buoyancy affect the ROV. The vector  $\tau$  are the generalised forces caused by the ROV's actuators and unmodelled disturbances.

In this chapter, the vector cross-product  $S(\cdot)$  is defined as  $S(aA)B = aA \times B$ . The notation used for the parameters in this thesis can be seen in Table 3.1. The notation for forces, moments, linear and angular velocities, positions and Euler angles used in the model is summarised in Table 3.2.

**Table 3.1:** The notation and description of the parameters used in the ROV model.

Notation	Description
$I_b$	Inertia matrix for rotation around CO.
$I_g$	Inertia matrix for rotation around CG.
$K_p, M_q, N_r$	Linear damping coefficients for rotation in water.
$K_{p p}, M_{q q}, N_{r r}$	Quadratic damping coefficients for rotation in water.
$K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}$	Increased inertia about $x, y, z$ -axis due to rotation in water.
$X_u, Y_v, Z_w$	Linear damping coefficients for translation in water.
$X_{u u}, Y_{v v}, Z_{w w}$	Quadratic damping coefficients for translation in water.
$X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}$	Added mass in $x, y, z$ -direction due to translation in water.
$l_{x_i}, l_{y_i}, l_{z_i}$ .	Moment arms from CG to each thruster $i$ .
$m$	The ROV's mass
$z_B$	Distance between CB and CG along the $z$ -axis.
$V$	Displaced volume.
$\rho$	Water density.
$g$	Gravitational constant.
$r_b^g$	The distance between CO and CG.

**Table 3.2:** The notation of SNAME (1950) for marine vessels.

DOF	Description	Forces and moments	Linear and angular velocities	Positions and Euler angles
1	Motions in the $x$ direction (surge).	$X$	$u$	$x$
2	Motions in the $y$ direction (sway).	$Y$	$v$	$y$
3	Motions in the $z$ direction (heave).	$Z$	$w$	$z$
4	Rotation about the $x$ axis (roll, heel).	$K$	$p$	$\phi$
5	Rotation about the $y$ axis (pitch, trim).	$M$	$q$	$\theta$
6	Rotation about the $z$ axis (yaw).	$N$	$r$	$\psi$

To get a more compact way of writing, the following vector notation has been used

- $\mathbf{p}$ : Positions.
- $\mathbf{v}$ : Linear velocities.
- $\Theta$ : Euler Angles.
- $\mathbf{q}$ : Quaternions.
- $\omega$ : Angular velocities.
- $\eta_1$  and  $\nu_1$ : positions and linear velocities.
- $\eta_2$  and  $\nu_2$ : attitude and angular velocities.

### 3.1 Coordinate Systems and Kinematics

When modelling it is important to choose proper coordinate systems in which to describe the systems behaviour. In this thesis, two coordinate systems are used in the ROV model.

The first system, the body-fixed coordinate system, is fixed to the ROV and rotates with the ROV. The body-fixed coordinate system is a right-hand system, the  $x$ -axis is placed along the length of the ROV pointing towards its bow. The  $y$ -axis points starboard, and the  $z$ -axis points downwards towards the vehicles keel. The coordinate system is assumed to be centred in the ROV's center of gravity (CG), i.e.  $r_b^g = 0$ . The body-fixed coordinate system makes it easier to describe sensor readings, since the sensors rotate with the ROV. It is also easier to express the effect of each thruster with forces and moments expressed in the body-fixed coordinate system. How the body-fixed coordinate system is placed in the ROV can be seen in Figure 3.1 and Figure 3.2.

The second global coordinate system is Earth fixed, with axes  $N$ ,  $E$  and  $D$ . The  $N$  axis points in the direction of the calibrated North, the  $E$  axis points in the direction of calibrated East and the  $D$  axis points down towards the CG of the Earth. This coordinate system is used to express buoyancy and gravitational forces acting on the ROV, their effects are transformed to the body-fixed coordinate system by a rotation matrix. The transformation for linear velocities from the body-fixed to the global coordinate system is

$$\mathbf{v}_{b/n}^n = \mathbf{R}_b^n(\Theta) \mathbf{v}_{b/n}^b \quad (3.3)$$

where the transformation matrix  $\mathbf{R}_b^n(\Theta)$  in  $zyx$  convention is

$$\mathbf{R}_b^n(\Theta) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (3.4)$$

where  $s \cdot$  stands for  $\sin(\cdot)$  and  $c \cdot$  stands for  $\cos(\cdot)$  (Fossen, 2011, p. 22). The

inverse of the transformation matrix  $\mathbf{R}_b^n(\Theta)$  is

$$\mathbf{R}_b^n(\Theta)^{-1} = \mathbf{R}_b^n(\Theta)^T \quad (3.5)$$

The super- and subscript notation is defined as

$$v^{\text{to}} = \mathbf{R}_{\text{from}}^{\text{to}}(\Theta)v^{\text{from}} \quad (3.6)$$

where  $n$  denotes the global coordinate system, while  $b$  denotes the body-fixed coordinate system. The notation  $v_{b/n}$  means that  $v$  is measured in  $b$  relative  $n$ .

Transformation of angular velocities is similarly given by

$$\dot{\Theta} = \mathbf{T}_\theta(\Theta)\omega_{b/n}^b \quad (3.7)$$

where

$$\mathbf{T}_\theta(\Theta) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (3.8)$$

and  $t \cdot$  stands for  $\tan(\cdot)$  (Fossen, 2011, p. 24). Similar to the linear velocities transformation matrix, the inverse of the transformation matrix is

$$\mathbf{T}_\theta(\Theta)^{-1} = \mathbf{T}_\theta(\Theta)^T \quad (3.9)$$

The kinematic equations can then be expressed in vector form as

$$\dot{\eta} = \mathbf{J}_\theta(\eta)\nu \iff \begin{bmatrix} \dot{p}_{b/n}^n \\ \dot{\Theta} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\Theta) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{T}_\theta(\Theta) \end{bmatrix} \begin{bmatrix} v_{b/n}^b \\ \omega_{b/n}^b \end{bmatrix} \quad (3.10)$$

where

$$\eta = [N, E, D, \phi, \theta, \psi]^T \quad (3.11)$$

and

$$\nu = [u, v, w, p, q, r]^T \quad (3.12)$$

Euler angles are singular for  $\pm\frac{\pi}{2}$  in  $\theta$  which might occur in an ROV. Thus it is useful to express angles with unit quaternions since these are always well defined (Gustafsson, 2012). To use quaternions,  $\eta$  is instead defined as

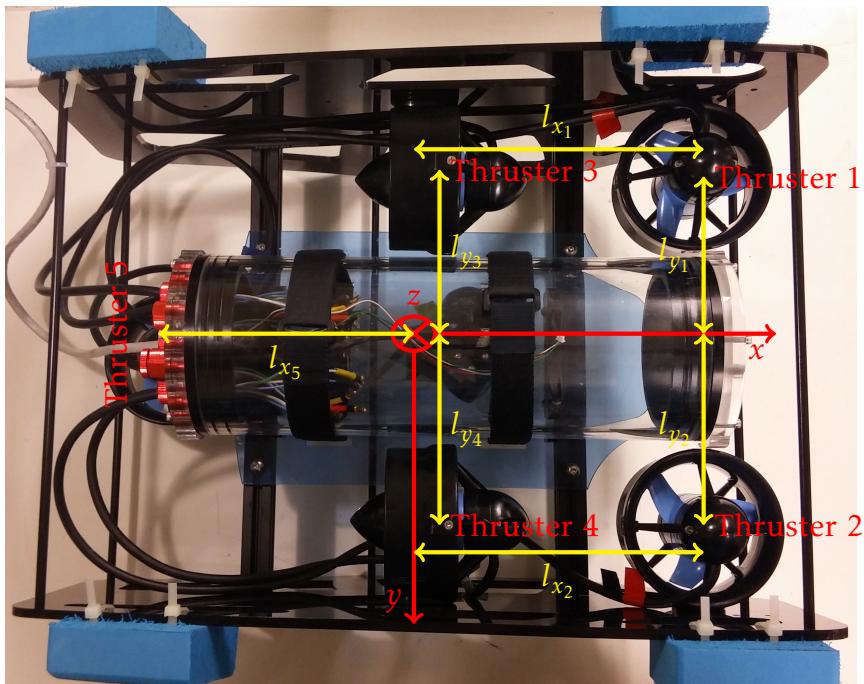
$$\eta = [N, E, D, \mathbf{q}]^T \quad (3.13)$$

where  $\mathbf{q} = [\eta, \epsilon_1, \epsilon_2, \epsilon_3]^T$

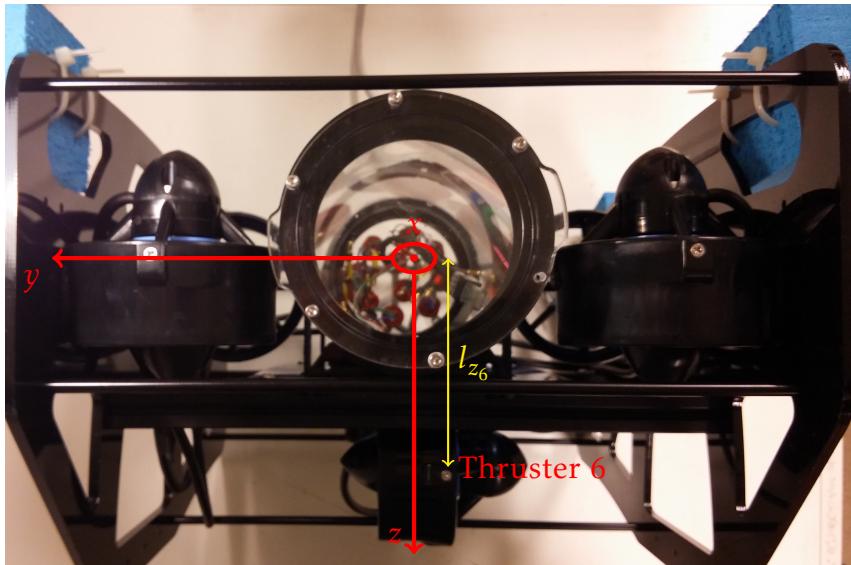
The quaternions need to satisfy  $\mathbf{q}^T \mathbf{q} = 1$  in order to represent angles and thus needs to be normalised. Approximate quaternion normalisation in continuous time can be achieved by adding a normalising term

$$\frac{\gamma}{2}(1 - \mathbf{q}^T \mathbf{q})\mathbf{q} \quad (3.14)$$

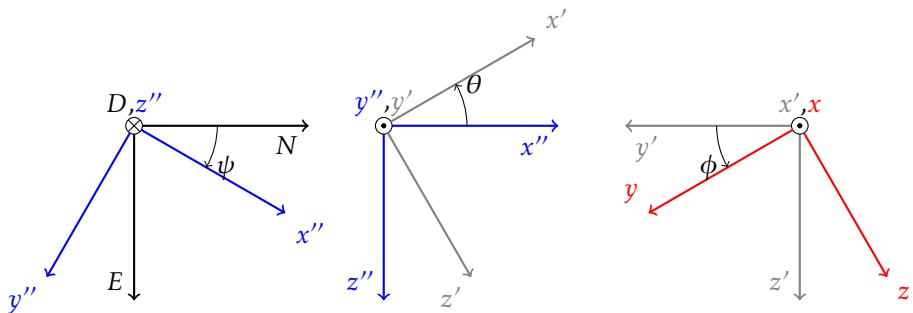
to the dynamics of  $\dot{\eta}$  (Fossen, 2011, p. 31). The parameter  $\gamma$  is a design parameter,  $\gamma \geq 0$ , usually  $\gamma = 100$ , indicating the convergence rate of the normalisation. In



**Figure 3.1:** Top view of the ROV. The body-fixed coordinate system (red) is fixed in the ROV. Yellow lines show the different moment arms to the thrusters. Each thruster is numbered in red.



**Figure 3.2:** Front view of the ROV. The body-fixed coordinate system (red) is fixed in the ROV. The yellow line shows the moment arm to thruster 6 which is numbered in red.



**Figure 3.3:** The local and global coordinate systems relate to each other by the rotations  $\psi$ ,  $\theta$  and  $\phi$ . The rotations are defined from the global coordinate system (black) to the body-fixed coordinate system (red).

discrete time, approximate quaternion normalisation is given by

$$\mathbf{q}(k+1) = \frac{\mathbf{q}(k+1)}{\sqrt{\mathbf{q}^T(k+1)\mathbf{q}(k+1)}} \quad (3.15)$$

If quaternions are used to represent the attitude, the transformation matrix from body-fixed coordinates to global coordinates is defined as

$$\mathbf{R}_b^n(\mathbf{q}) = \begin{bmatrix} 1 - 2(\epsilon_2^2 + \epsilon_3^2) & 2(\epsilon_1\epsilon_2 - \epsilon_3\eta) & 2(\epsilon_1\epsilon_3 + \epsilon_2\eta) \\ 2(\epsilon_1\epsilon_2 + \epsilon_3\eta) & 1 - 2(\epsilon_1^2 + \epsilon_3^2) & 2(\epsilon_2\epsilon_3 - \epsilon_1\eta) \\ 2(\epsilon_1\epsilon_3 - \epsilon_2\eta) & 2(\epsilon_2\epsilon_3 + \epsilon_1\eta) & 1 - 2(\epsilon_1^2 + \epsilon_2^2) \end{bmatrix}, \quad (3.16)$$

the angular velocities transformation matrix is defined as

$$\mathbf{T}_q(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{bmatrix} \quad (3.17)$$

and the kinematic equation (3.10) is changed to

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_q(\boldsymbol{\eta})\boldsymbol{\nu} \iff \begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\mathbf{q}) & \mathbf{0}_{3x3} \\ \mathbf{0}_{4x3} & \mathbf{T}_q(\mathbf{q}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix}, \quad (3.18)$$

see Fossen (2011, Ch. 2) for more information.

## 3.2 Rigid-Body Kinetics

The rigid-body kinetic relations of the ROV can be derived using the Newton-Euler formulation and can be expressed as

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}_{RB} \quad (3.19)$$

where  $\mathbf{M}_{RB}$  is the rigid-body inertia matrix,  $\mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}$  is a vector describing the centripetal and Coriolis effects and  $\boldsymbol{\tau}_{RB}$  are the forces and moments acting on the rigid body (Fossen, 2011, p. 45). The rigid-body inertia matrix

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3x3} & -m\mathbf{S}(\mathbf{r}_g^b) \\ m\mathbf{S}(\mathbf{r}_g^b) & \mathbf{I}_b \end{bmatrix} \quad (3.20)$$

describes the resistance to change in linear- and angular velocity in the ROV's 6 DOF. Here,  $\mathbf{r}_g^b$  is the vector from the ROV's center of origin (CO) to its CG,  $m$  is the mass of the ROV and  $\mathbf{I}_b$  is the inertia matrix for rotation about the ROV's CO (Fossen, 2011, p. 50). Assuming that the ROV's CO and CG coincide, simplifies (3.20) to

$$\mathbf{M}_{RB} = \begin{bmatrix} m\mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_g \end{bmatrix} \quad (3.21)$$

where  $\mathbf{I}_g$  is the inertia matrix about the ROV's CG. Since the ROV travels in a rotating reference frame, the Earth, the ROV is subjected to inertial forces called

Coriolis forces. These forces are modelled by the vector  $\mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}$  which describes the Coriolis and centripetal forces caused by the rigid body's mass. The vector  $\mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}$  is defined as

$$\mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \begin{bmatrix} mS(\boldsymbol{\nu}_2) & -mS(\boldsymbol{\nu}_2)S(r_g^b) \\ mS(r_g^b)S(\boldsymbol{\nu}_2) & -S(I_b\boldsymbol{\nu}_2) \end{bmatrix} \boldsymbol{\nu} = \begin{bmatrix} m(qw - rv) \\ m(ru - pw) \\ m(pv - qu) \\ qr(I_y - I_z) \\ rp(I_z - I_x) \\ qp(I_x - I_y) \end{bmatrix} \quad (3.22)$$

where it is assumed that the ROV is symmetric about the  $xyz$ -plane to eliminate cross-terms (Fossen, 2011, p. 55).

### 3.3 Hydrodynamics

The ROV experiences forces and effects caused by interaction with water. These hydrodynamic effects can be modelled as

$$\boldsymbol{\tau}_{\text{Dyn}} = -\mathbf{M}_A \dot{\boldsymbol{\nu}} - \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} \quad (3.23)$$

where  $\mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu}$  and  $\mathbf{M}_A$  models the Coriolis forces and inertia from the added mass and moment of inertia and the vector  $\mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu}$  models linear and quadratic damping effects. The added mass and moment of inertia is defined as

$$\mathbf{M}_A = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & N_{\dot{r}} \end{bmatrix}, \quad (3.24)$$

under the assumption that the ROV moves at low speeds relative to the water (Fossen, 2011, p. 121). The Coriolis-centripetal effects from the added mass and the added moment of inertia are described as

$$\begin{aligned} \mathbf{C}_A(\boldsymbol{\nu})\boldsymbol{\nu} &= \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \boldsymbol{\nu} \\ &= \begin{bmatrix} Y_{\dot{v}}vr - Z_{\dot{w}}wq \\ Z_{\dot{w}}wp - X_{\dot{u}}ur \\ X_{\dot{u}}uq - Y_{\dot{v}}vp \\ (Y_{\dot{v}} - Z_{\dot{w}})vw + (M_{\dot{q}} - N_{\dot{r}})qr \\ (Z_{\dot{w}} - X_{\dot{u}})uw + (N_{\dot{r}} - K_{\dot{p}})pr \\ (X_{\dot{u}} - Y_{\dot{v}})uv + (K_{\dot{p}} - M_{\dot{q}})pq \end{bmatrix}, \end{aligned} \quad (3.25)$$

under the assumption that the ROV is moving slowly and has three planes of symmetry (Fossen, 2011, p. 121).

There are three main sources of hydrodynamic damping acting upon a submerged vehicle: potential damping, skin friction and damping from vortex shedding (Fossen, 2011, p. 122). The effects are split in to two parts, linear damping and quadratic damping. The matrix  $D$  contains the linear damping terms, while the matrix  $D_n(\nu)$  contains the quadratic, or non-linear, damping terms (Fossen, 2011). The sum of these two matrices form the Viscous damping matrix  $D(\nu)$  which in turn can be simplified to

$$D(\nu) = D + D_n(\nu) = - \begin{bmatrix} X_u + X_{u|u}|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & Y_v + Y_{v|v}|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & Z_w + Z_{w|w}|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & K_p + K_{p|p}|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & M_q + M_{q|q}|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r + N_{r|r||r|} \end{bmatrix} \quad (3.26)$$

where it is assumed that the ROV is symmetric about the  $xz$ -plane and that the damping is decoupled, thus a diagonal matrix  $D(\nu)$  is obtained (Fossen, 2011, p. 129). When the matrix  $D(\nu)$  is multiplied with  $\nu$  the following vector is obtained

$$D(\nu)\nu = - \begin{bmatrix} (X_u + X_{u|u}|u|)u \\ (Y_v + Y_{v|v}|v|)v \\ (Z_w + Z_{w|w}|w|)w \\ (K_p + K_{p|p}|p|)p \\ (M_q + M_{q|q}|q|)q \\ (N_r + N_{r|r||r|})r \end{bmatrix} \quad (3.27)$$

## 3.4 Hydrostatics

The ROV will experience forces and moments caused by the Earth's gravitational pull and the buoyancy force, these are called restoring forces (Fossen, 2011). These forces and moments can be modelled on the form

$$\tau_{\text{Stat}} = -g(\eta) \quad (3.28)$$

where  $\tau_{\text{Stat}}$  are generalised forces. The restoring forces and moments are calculated using four main parameters; the mass  $m$  of the vehicle, its buoyancy  $B$  and lastly the coordinates of the ROV's CG and center of buoyancy (CB). Their effects are computed as follows

$$g(\eta) = - \begin{bmatrix} f_g + f_b \\ r_b \times f_b + r_g \times f_g \end{bmatrix} = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ -z_B B \cos \theta \sin \phi \\ -z_B B \sin \theta \\ 0 \end{bmatrix} \quad (3.29)$$

or alternatively using quaternions

$$\mathbf{g}(\eta) = -\begin{bmatrix} \mathbf{f}_g + \mathbf{f}_b \\ \mathbf{r}_b \times \mathbf{f}_b + \mathbf{r}_g \times \mathbf{f}_g \end{bmatrix} = \begin{bmatrix} (B-W)(2\epsilon_1\epsilon_3 - 2\epsilon_2\eta) \\ (B-W)(2\epsilon_2\epsilon_3 + 2\epsilon_1\eta) \\ (W-B)(2\epsilon_1^2 + 2\epsilon_2^2 - 1) \\ -Bz_B(2\epsilon_2\epsilon_3 + 2\epsilon_1\eta) \\ Bz_B(2\epsilon_1\epsilon_3 - 2\epsilon_2\eta) \\ 0 \end{bmatrix} \quad (3.30)$$

where  $\mathbf{r}_b = [0, 0, z_B]^T$  and  $\mathbf{r}_g = [0, 0, 0]^T$  are moment arms,  $\mathbf{f}_g = \mathbf{R}_b^n(\Theta)^T[0, 0, W]^T$  and  $\mathbf{f}_b = -\mathbf{R}_b^n(\Theta)^T[0, 0, B]^T$  are restoring forces (Fossen, 2011, p. 60). Here,  $B = \rho g V$  and  $W = mg$ . In other words, the magnitude of the buoyancy forces is equal to the weight of the displaced water. For a fully submerged vehicle,  $V$  will naturally be equal to the volume of the vehicle. It is henceforth assumed that the ROV has neutral buoyancy i.e.  $B = W$ . Note that the positions of the three centers are described using the coordinate system described in Section 3.1, a roll and pitch stable ROV should thus have a  $z_B < 0$ .

### 3.5 Actuators

The ROV is equipped with six identical, three-bladed thrusters. These can be modelled as

$$\boldsymbol{\tau}_{\text{Act}} = Tf(\mathbf{u}) \quad (3.31)$$

where  $T$  is a matrix describing the geometry of the actuators (Fossen, 2011, p. 401). Figures 3.1 and 3.2 illustrate the moment arms to each thruster. The positive rotation of the thrusters and the resulting positive forces can be seen in Figures B.3 to B.6. This gives

$$\begin{aligned} \boldsymbol{\tau}_{\text{Act}} = Tf(\mathbf{u}) &= \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & -1 & 0 \\ l_{y_1} & -l_{y_2} & 0 & 0 & 0 & l_{z_6} \\ l_{x_1} & l_{x_2} & 0 & 0 & -l_{x_5} & 0 \\ 0 & 0 & l_{y_3} & -l_{y_4} & 0 & 0 \end{bmatrix} \begin{bmatrix} f(u_1) \\ f(u_2) \\ f(u_3) \\ f(u_4) \\ f(u_5) \\ f(u_6) \end{bmatrix} = \\ &= \begin{bmatrix} f(u_3) + f(u_5)g \\ -f(u_6) \\ -f(u_1) - f(u_2) - f(u_4) \\ f(u_2)l_{y_2} - f(u_1)l_{y_1} + f(u_6)l_{z_6} \\ f(u_2)l_{x_2} - f(u_1)l_{x_1} - f(u_4)l_{x_4} \\ f(u_3)l_{y_3} - f(u_5)l_{y_5} \end{bmatrix} \end{aligned} \quad (3.32)$$

where  $l_{x_i}$ ,  $l_{y_i}$  and  $l_{z_i}$  are the offsets in the  $x$ ,  $y$  or  $z$  directions of the  $i^{\text{th}}$  thruster, respectively, and  $f(u_i)$  is a lookup table from control signal  $u_i$  to thrust in Newton. See Appendix C for details regarding the look-up table.

## 3.6 Model in Component Form

Using the relation

$$\tau_{RB} = \tau_{Dy} + \tau_{Stat} + \tau_{Act} + \tau_{Dist} \quad (3.33)$$

(3.19), (3.23) and (3.28) can be combined to form

$$M_{RB}\dot{v} + C_{RB}(v)v + M_A\dot{v} + C_A(v)v + D(v)v + g(\eta) = \tau_{Act} + \tau_{Dist} \quad (3.34)$$

This equation can be rearranged to

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau_{Act} + \tau_{Dist} \approx \tau_{Act} \quad (3.35)$$

where  $M = M_{RB} + M_A$  and  $C(v) = C_{RB}(v) + C_A(v)$ . The expression can then be solved for  $\dot{v}$  and split into components

$$\begin{aligned} \dot{u} &= \frac{f(u_3) + f(u_4)}{m - X_{\dot{u}}} + \frac{(X_u + X_{u|u}|u|)u}{m - X_{\dot{u}}} + \frac{(B - W)\sin(\theta)}{m - X_{\dot{u}}} + \\ &\quad \frac{m(rv - qw)}{m - X_{\dot{u}}} + \frac{-Y_{\dot{v}}rv}{m - X_{\dot{u}}} + \frac{Z_{\dot{w}}qw}{m - X_{\dot{u}}}, \end{aligned} \quad (3.36a)$$

$$\begin{aligned} \dot{v} &= \frac{-f(u_6)}{m - Y_{\dot{v}}} + \frac{(Y_v + Y_{v|v}|v|)v}{m - Y_{\dot{v}}} + \frac{-(B - W)\cos\theta\sin\phi}{m - Y_{\dot{v}}} + \\ &\quad \frac{m(pw - ru)}{m - Y_{\dot{v}}} + \frac{X_{\dot{u}}ru}{m - Y_{\dot{v}}} + \frac{-Z_{\dot{w}}pw}{m - Y_{\dot{v}}}, \end{aligned} \quad (3.36b)$$

$$\begin{aligned} \dot{w} &= \frac{-f(u_1) - f(u_2) - f(u_5)}{m - Z_{\dot{w}}} + \frac{(Z_w + Z_{w|w}|w|)w}{m - Z_{\dot{w}}} + \frac{-(B - W)\cos\phi\cos\theta}{m - Z_{\dot{w}}} + \\ &\quad \frac{m(qu - pv)}{m - Z_{\dot{w}}} + \frac{-X_{\dot{u}}qu}{m - Z_{\dot{w}}} + \frac{Y_{\dot{v}}pv}{m - Z_{\dot{w}}}, \end{aligned} \quad (3.36c)$$

$$\begin{aligned} \dot{p} &= \frac{f(u_1)l_{y_1} - f(u_2)l_{y_2} + f(u_6)l_{z_6}}{I_x - K_{\dot{p}}} + \frac{(K_p + K_{p|p}|p|)p}{I_x - K_{\dot{p}}} + \frac{-M_{\dot{q}}qr}{I_x - K_{\dot{p}}} + \frac{N_rqr}{I_x - K_{\dot{p}}} + \\ &\quad \frac{qr(I_y - I_z)}{I_x - K_{\dot{p}}} + \frac{-Y_{\dot{v}}vw}{I_x - K_{\dot{p}}} + \frac{Z_{\dot{w}}vw}{I_x - K_{\dot{p}}} + \frac{Bz_B\cos\theta\sin\phi}{I_x - K_{\dot{p}}}, \end{aligned} \quad (3.36d)$$

$$\begin{aligned} \dot{q} &= \frac{f(u_1)l_{x_1} + f(u_2)l_{x_2} - f(u_5)l_{x_5}}{I_y - M_{\dot{q}}} + \frac{(M_q + M_{q|q}|q|)q}{I_y - M_{\dot{q}}} + \frac{K_{\dot{p}}pr}{I_y - M_{\dot{q}}} + \frac{-N_rpr}{I_y - M_{\dot{q}}} + \\ &\quad \frac{pr(I_z - I_x)}{I_y - M_{\dot{q}}} + \frac{-Z_{\dot{w}}uw}{I_y - M_{\dot{q}}} + \frac{X_{\dot{u}}uw}{I_y - M_{\dot{q}}} + \frac{Bz_B\sin\theta}{I_y - M_{\dot{q}}} \end{aligned} \quad (3.36e)$$

and

$$\dot{r} = \frac{f(u_3)l_{y_3} - f(u_4)l_{y_4}}{I_z - N_r} + \frac{(N_r + N_{r|r}|r|)r}{I_z - N_r} + \frac{-K_p pq}{I_z - N_r} + \frac{M_{\dot{q}} pq}{I_z - N_r} + \frac{pq(I_x - I_y)}{I_z - N_r} + \frac{-X_{\dot{u}} uv}{I_z - N_r} + \frac{Y_{\dot{v}} uv}{I_z - N_r} \quad (3.36f)$$

### 3.7 Simplified Model Structures

Since the ROV can not measure its position in  $N$  and  $E$  nor its velocity in  $u$ ,  $v$  and  $w$  estimation of parameters linked to these terms are difficult to identify. To make estimation more convenient, data-collection experiments can be designed such that linear velocities will be small. The effects of the linear velocities in (3.36) can then be neglected. For identifiability reasons it is necessary to reparametrise the model by congregating some of the original parameters before conducting parameter estimation. To gain identifiability, the reparametrisation  $A_p = I_x - K_p$ ,  $B_q = I_y - M_{\dot{q}}$  and  $C_r = I_z - N_r$  was introduced. These changes resulted in

$$\dot{p} = \frac{f(u_1)l_{y_1} - f(u_2)l_{y_2} + f(u_6)l_{z_6}}{A_p} + \frac{Bz_B \cos \theta \sin \phi}{A_p} + \frac{(K_p + K_{p|p}|p|)p}{A_p} + \frac{qr(B_q - C_r)}{A_p}, \quad (3.37a)$$

$$\dot{q} = \frac{f(u_1)l_{x_1} + f(u_2)l_{x_2} - f(u_5)l_{x_5}}{B_q} - \frac{Bz_B \sin \theta}{B_q} + \frac{(M_q + M_{q|q}|q|)q}{B_q} - \frac{pr(A_p - C_r)}{B_q} \quad (3.37b)$$

and

$$\dot{r} = \frac{f(u_3)l_{y_3} - f(u_4)l_{y_4}}{C_r} + \frac{(N_r + N_{r|r}|r|)r}{C_r} + \frac{pq(A_p - B_q)}{C_r} \quad (3.37c)$$

From (3.37) it can be seen that  $r$  is not affected by the same thrusters as  $p$  and  $q$ . It can therefore be convenient to excite the ROV mainly in  $p$  and  $q$  or in  $r$ . If the ROV is excited in this way, while still keeping the linear velocities low, the effects of  $r$  can then be assumed to be small in (3.37a) and (3.37b). Similarly the effects of  $p$  and  $q$  can be assumed to be small in (3.37c). Under these assumptions, the

model structure (3.37) can be reduced to

$$\dot{p} = \frac{f(u_1)l_{y_1} - f(u_2)l_{y_2} + f(u_6)l_{z_6}}{A_p} + \frac{(Kp + K_{p|p}|p|)p}{A_p} + \frac{Bz_B \cos \theta \sin \phi}{A_p}, \quad (3.38a)$$

$$\dot{q} = \frac{f(u_1)l_{x_1} + f(u_2)l_{x_2} - f(u_5)l_{x_5}}{B_q} + \frac{(M_q + M_{q|q}|q|)q}{B_q} + \frac{Bz_B \sin \theta}{B_q} \quad (3.38b)$$

and

$$\dot{r} = \frac{f(u_3)l_{y_3} - f(u_4)l_{y_4}}{B_q} + \frac{(N_r + N_{r|r}|r|)r}{B_q} \quad (3.39)$$



# 4

---

## Sensor Fusion

In order to properly estimate the ROV's attitude in the global coordinate system, the ROV needs sensors to measure external effects from its environment. Unfortunately signals from sensors do not necessarily give direct information about attitude and their measurements are to some extent noisy. Algorithms can nevertheless be used to extract and combine the information from the different sensors into an attitude estimate. The process of combining, or fusing, the information from several measurements with or without a motion model to produce an estimate of a state is called sensor fusion. Since sensor fusion is only a prerequisite for control of the ROV's attitude, no results will be presented in this section.

To be able to understand how a sensor fusion algorithm works, it is important to understand the notation. In Table 4.1, the notation used in this chapter is listed.

### 4.1 The Extended Kalman Filter

The Kalman filter (KF) is a linear state-space observer, it estimates both measurable and unmeasurable states in a linear system (Gustafsson, 2012). It utilises a linear motion model, a model of the systems dynamics, in conjunction with measurements and linear measurement equations to provide the best possible estimate of the system's states. The KF is the best possible linear filter for the given measurement  $y_k$  (Gustafsson, 2012). One filter that can accomplish the task of fusing different measurements and estimating states in a non-linear dynamic system is the extended Kalman filter (EKF). The EKF can, unlike the regular KF, handle non-linear motion models and measurement equations. The EKF accomplishes this by using a linearised model of the non-linear system and the measurement equations. If an EKF can provide satisfactory results depends on

**Table 4.1:** The notation used for describing a sensor fusion algorithm.

Notation	Description
$x$	State vector.
$\hat{x}$	State vector estimate.
$y$	Measurement vector.
$u$	Control signal vector.
$v, e$	Noise vectors.
$k$	At time $k$ .
$k m$	At time $k$ given information up to time $m$ .
$f_x$	Jacobian of $f$ with respect to states $x$ .
$f_v$	Jacobian of $f$ with respects to noise $v$ .
$E(x)$	The expected value of $x$ .
$\text{Cov}(x)$	The covariance of $x$ .

the rest terms from the linearisation. It is therefore dependent on the degree of non-linearity of the system and the measurement equations (Gustafsson, 2012). As rule of thumb the rest term will be small enough if the system model is close to linear and if measurements are of good quality, meaning that the signal to noise ratio (SNR) is high (Gustafsson, 2012).

The EKF algorithm is comprised of two key steps called updates. The time update uses the current state estimates and the user specified motion model to predict the values of the states at the next time instant. The second update is called the measurement update and it uses sampled sensor data, previous state estimates and user specified measurement equations to fuse measurements into a state estimate (Gustafsson, 2012). If the measurements are independent, a measurement update can be performed at the arrival of each measurement without the need of a time update in between (Gustafsson, 2012, p. 170). This is useful in the ROV since the sensors are sampled in different rates. The complete EKF algorithm is summarised in Algorithm 1.

## 4.2 Motion Model

A KF uses a model of the system dynamics to improve the estimates of the model's states. It is therefore important to choose a model that describes the system's dynamics well. In this thesis, two different motion models have been used, a model using the measured angular velocities as inputs and a more advanced model using the angular velocities as states. All models in this chapter use quaternions and thus quaternion normalisation is required as described in Chapter 3.

The simple model using the measured angular velocities as inputs was based on the quaternion kinematics model in Törnqvist (2006, p. 47). The model was expanded with depth as an extra state which was modelled as constant position and was discretised using Euler forward. See Ljung and Glad (2004, p. 378) for

---

**Algorithm 1** The extended Kalman filter algorithm (Gustafsson, 2012).

The extended Kalman filter applied on a model

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{\epsilon}_k)$$

is given by the following algorithm:

**Initialisation:**

$$\hat{\mathbf{x}}_{1|0} = E(\mathbf{x}_0)$$

$$\mathbf{P}_{1|0} = \text{Cov}(\mathbf{x}_0)$$

**Measurement update:**

$$\mathbf{S}_k = h_x(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k) \mathbf{P}_{k|k-1} h_x(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k)^T + \mathbf{R}_k$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} h_x(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k)^T \mathbf{S}_k^{-1}$$

$$\boldsymbol{\epsilon} = \mathbf{y}_k - h(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k)$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \boldsymbol{\epsilon}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{P}_{k|k-1} h_x(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k)^T \mathbf{S}_k^{-1} h_x(\hat{\mathbf{x}}_{k|k-1}, \mathbf{u}_k) \mathbf{P}_{k|k-1}$$

**Time update:**

$$\hat{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)$$

$$\mathbf{P}_{k+1|k} = f_x(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \mathbf{P}_{k|k} f_x(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)^T + f_v(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k) \mathbf{Q}_k f_v(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)^T$$


---

details regarding Euler discretisation. The complete simple model is

$$\begin{bmatrix} \boldsymbol{\eta}_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} I_{4 \times 4} + T_s \bar{T}(\boldsymbol{\nu}_k) & \mathbf{0}_{4 \times 1} \\ \mathbf{0}_{1 \times 4} & I_{1 \times 1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_k \\ d_k \end{bmatrix} + \begin{bmatrix} T_s T(\boldsymbol{\eta}_k) & \mathbf{0}_{4 \times 1} \\ \mathbf{0}_{1 \times 3} & T_s \end{bmatrix} \begin{bmatrix} \mathbf{v}_\eta \\ v_d \end{bmatrix} \quad (4.1)$$

Here  $\bar{T}(\boldsymbol{\nu})$  is defined as

$$\bar{T}(\boldsymbol{\nu}) = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \quad (4.2)$$

and  $T(\boldsymbol{\eta})$  is defined in (3.17). Note that quaternion normalisation (3.15) is done after each time- and measurement update. It is also important to note that this is an attitude model and thus  $\boldsymbol{\eta}$  and  $\boldsymbol{\nu}$  only contains quaternions and angular velocities respectively. Note that  $\boldsymbol{\nu}$  is not modelled as a state but is used as an input to reduce the dimension of the model as in Törnqvist (2006).

The second model was implemented to improve sensor fusion performance. This model was based on the continuous-time rigid-body kinematics model in Gustafsson (2012, p. 351) and incorporated gyroscope bias estimates and  $\boldsymbol{\nu}$  as a constant position states. The continuous-time model in Gustafsson (2012, p. 351) also modelled positions, linear- accelerations and velocities. Since no position measurements except for depth-measurements were available on the ROV platform,

linear- accelerations and velocities were neglected and the position state vector was reduced to only contain depth. The noise model was also slightly modified by giving each state its own noise source. This is not true from a physical stand-point since for example, a disturbance in angular velocity will effect the angles, but it made the filter easier to trim. The entire model was discretised using Euler forward which yielded the following discrete model

$$\begin{bmatrix} \eta_{k+1} \\ v_{k+1} \\ b_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} I_{4 \times 4} + T_s \bar{T}(\nu_k) & 0_{4 \times 7} \\ 0_{7 \times 4} & I_{7 \times 7} \end{bmatrix} \begin{bmatrix} \eta_k \\ v_k \\ b_k \\ d_k \end{bmatrix} + \begin{bmatrix} v_\eta \\ v_v \\ v_b \\ v_d \end{bmatrix} \quad (4.3)$$

where  $T_s$  is the sample time. Note that the quaternion normalisation (3.15) is done after each update.

### 4.3 Measurement Equations

To fuse information from different sensors, the readings of each sensor has to be related with the states and noise sources which is done using measurement equations.

The ROV is equipped with an IMU containing a gyroscope. Since readings from a gyroscope might not be zero in all axes even if the gyroscope is at rest, it is important to incorporate bias states. Modelling for biases in the gyroscope gives the measurement equation

$$y_{\text{Gyro}} = \nu + b + v_{\text{Gyro}} \quad (4.4)$$

The vector  $y_{\text{Gyro}}$  is the reading from the IMU's gyroscope in rad/s and  $v_{\text{Gyro}}$  is the measurement noise. The vector  $b$  contain the gyroscope's biases in the  $x$ -,  $y$ - and  $z$ -axes, respectively. Since the noise level of the gyroscope was low and no significant disturbances were observed, no outlier rejection was performed on the gyroscope measurements.

The IMU measures acceleration in addition to angular velocities. The measurement equation for the accelerometer is

$$y_{\text{Acc}} = R_b^n(q)^T \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} + v_{\text{Acc}} \quad (4.5)$$

where  $R_b^n(q)$  is the rotation matrix defined in Chapter 3,  $g$  is the gravitational constant and  $v_{\text{Acc}}$  is measurement noise. Since the accelerometer is not perfectly centred in the ROV's CG and since the ROV rotates, accelerates and decelerates, the gravity is not the only thing that is being measured by the accelerometer. This leads to problems when trying to estimate the ROV's attitude since the sensor fusion algorithm tries to use the known direction and magnitude of the Earth's gravitational pull to estimate the ROV's attitude. To ensure that only the gravitational constant  $g$  is used to update the ROV's attitude, outlier rejection is performed.

Accelerometer measurements are only used if

$$|\|y_{\text{Acc}}\| - g| < \epsilon_{\text{Acc}} \quad (4.6)$$

Here,  $\epsilon_{\text{Acc}}$  is a design parameter that tweaks how much the magnitude of the measurement may deviate from  $g$  before being considered an outlier.

The I/O-unit of the ROV is also equipped with a magnetometer which enables the ROV to measure the magnetic field strength in the three local axes  $x$ ,  $y$  and  $z$ . Assuming the magnetometer only measures the Earth's magnetic field in the body-fixed coordinate system gives the measurement equation

$$y_{\text{Mag}} = R_b^n(\mathbf{q})^T \begin{bmatrix} \sqrt{m_N^2 + m_E^2} \\ 0 \\ m_D \end{bmatrix} + v_{\text{Mag}} \quad (4.7)$$

where  $y_{\text{Mag}}$  is the measured magnetic field in the body-fixed coordinate system and  $v_{\text{Mag}}$  is measurements noise. The parameters  $m_N$ ,  $m_E$  and  $m_D$  are the measured magnetic field in the local coordinate system at start up of the ROV. Since the strength and inclination of the Earth's magnetic field vary with location  $m_N$ ,  $m_E$  and  $m_D$  are set to values that represent the magnetic field at the current location. This is done via a calibration script that sets the  $m_N$ ,  $m_E$  and  $m_D$  parameters to the current reading which in turn sets the current direction as the global coordinate system's North.

The ROV is not a noise free environment from an electromagnetic standpoint. Currents in the ROV's electronics may induce magnetic fields which will distort the sensor readings of the magnetometer. If such noisy measurements are used the sensor fusion will not perform well when estimating the ROV's attitude. To ensure that only measurements in good condition are used, an outlier rejection criteria was implemented. Magnetometer measurements are only used if

$$\left| \|y_{\text{Mag}}\| - \left\| \begin{bmatrix} m_N \\ m_E \\ m_D \end{bmatrix} \right\| \right| < \epsilon_{\text{Mag}} \quad (4.8)$$

holds. Here,  $\epsilon_{\text{Mag}}$  is a design parameter that tweaks how much the magnitude of the measurements may deviate from the magnitude of the Earth's magnetic field before being rejected.

The ROV is also equipped with a pressure sensor and a barometer. The pressure sensor is placed in the rear or stern of the ROV which in turn means that the attitude of the ROV needs to be taken in to account when estimating the depth. Taking this into consideration yields the following measurement equation for the pressure sensor

$$y_{\text{Pre}} = \rho g \left( d + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} R_b^n(\mathbf{q}) \begin{bmatrix} x_{\text{offset}} \\ 0 \\ 0 \end{bmatrix} \right) + v_{\text{Pre}} \quad (4.9)$$

where  $\rho$  is the density of the water,  $d$  is the current depth in meters and  $v_{\text{Pre}}$  is

measurement noise. The parameter  $x_{\text{offset}}$  is the pressure sensor's offset from the ROV's CO in the  $x$ -direction. In this case,  $x_{\text{offset}}$  is a negative number. Here,  $y_{\text{Pre}}$  is a fictive measurement in Pascal, created by subtracting the reading from the internal barometer from the reading of the external pressure sensor, i.e.  $y_{\text{Pre}} = y_{\text{Ext}} - y_{\text{Int}}$  where  $y_{\text{Ext}}$  is the pressure measured by the external pressure sensor and  $y_{\text{Int}}$  is the pressure measured by the barometer. A basic form of outlier rejection is implemented in the measurement update. The update is performed if

$$p \geq 0 \quad (4.10)$$

since a reading lower than zero would imply that the ROV is above the water surface.

# 5

---

## Parameter Estimation

A central part of modelling is parameter estimation and its importance is often underestimated. If the model is not well estimated several problems could arise when trying to synthesise model-based controllers. If a system's dynamics are completely unknown, a black-box method can be used, *i.e.* general equations are fitted to test data and the parameters have no physical significance. When a system can be completely modelled using first principles, for example, Newtons laws, it is called white-box modelling. In this thesis, the model structure of the system is assumed to be known and given by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k | \boldsymbol{\theta}) \quad (5.1)$$

$$\mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{v}_k | \boldsymbol{\theta}) \quad (5.2)$$

where several of the parameters in  $\boldsymbol{\theta}$  are unknown. To estimate parameters in a known model structure is called gray-box modelling.

In general, parameter estimation is to fit a model structure's parameter vector  $\boldsymbol{\theta}$  such that the model best describes the estimation data. The model is then validated with a dataset which was not used during estimation. If the model describes the validation data well, the model and its parameter values are accepted. A measure of how well a model describes a dataset is the normalised root mean square error given by

$$\text{Fit} = 100 \left( 1 - \frac{\sqrt{\sum_{t=1}^N (\mathbf{y}(t) - \hat{\mathbf{y}}(t))^2}}{\sqrt{\sum_{t=1}^N (\mathbf{y}(t) - \frac{1}{N} \sum_{t=1}^N \mathbf{y}(t))^2}} \right) \quad (5.3)$$

A high fit value indicates that the model describes the validation data well.

An aspect to consider when estimating model parameters is what to model as inputs and what to model as outputs. Inputs to a system are considered to be true signals, thus if they are noisy or inaccurate they can effect the estimation negatively (Ljung and Glad, 2004). Outputs from a system are measurements of one or several states and are thus expected to be somewhat noisy and can not be used as ground truth.

To estimate the unknown parameters in the ROV model described in Chapter 3, different methods can be used. Two different methods will be explained further in Section 5.2 and Section 5.4.

## 5.1 Data Collection and Processing

To conduct parameter estimation, experimental data sets had to be collected. To be able to use the assumptions in (3.38) and (3.39) the data collection had to be performed in three specific ways. First the ROV was excited in  $p$  and  $q$ , then in  $r$  and lastly the ROV was excited in all rotations. Figure 5.1 and Figure 5.2 illustrates the conducted tests. All experiments were conducted in such a way that linear velocities were kept at a minimum. The main control signals used during data collection tests were telegraph signals. Both the scaling of the output magnitude and switch factors were changed in between tests in order to find signals that excited the desired states sufficiently. An example of a such a signal can be seen in Figure 5.3.

Data preprocessing was done by synchronising the data streams by checking the start and end of each experiment, here chosen as the time of arrival of the first control signal. The sensor data was aligned such that the data sample of each sensor whose arrival time was closest to the starting time of the test was chosen as the initial sample of each data stream. Data outside the test interval was discarded. After alignment, the data was resampled to 100 Hz. The sensor data was resampled using first-order hold while the control signals were resampled using zero-order hold since the control signal data only contained points where the control signals changed.

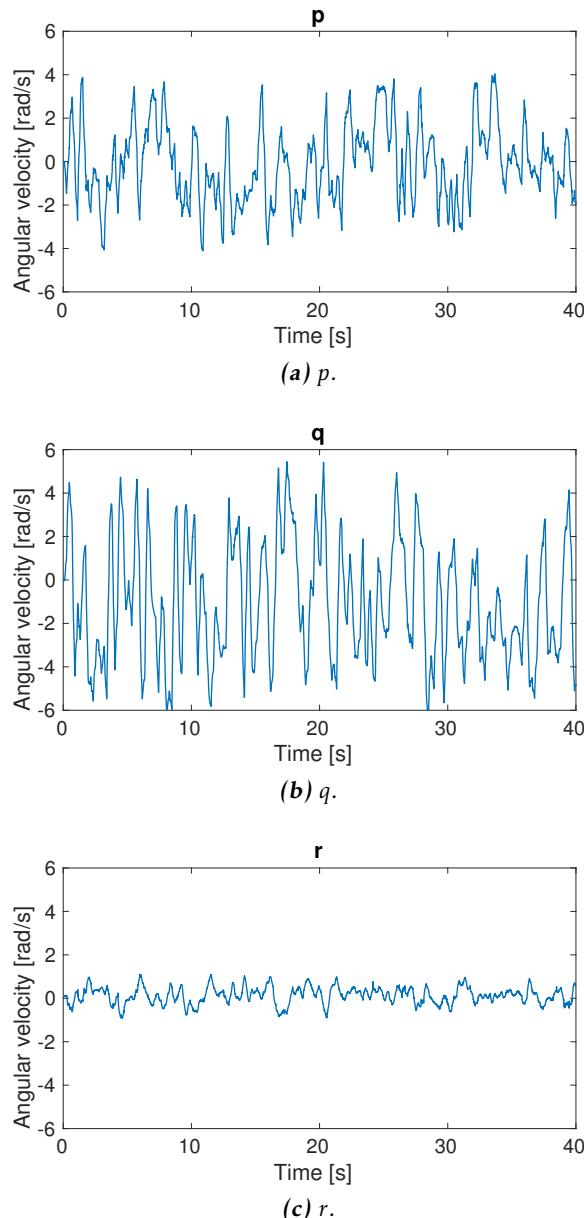
## 5.2 Prediction-Error Method

The prediction-error method uses a predictor  $\hat{y}_k(\theta)$  of a model's output to compare with the present output of the system. The discrete-time predictor can be described by

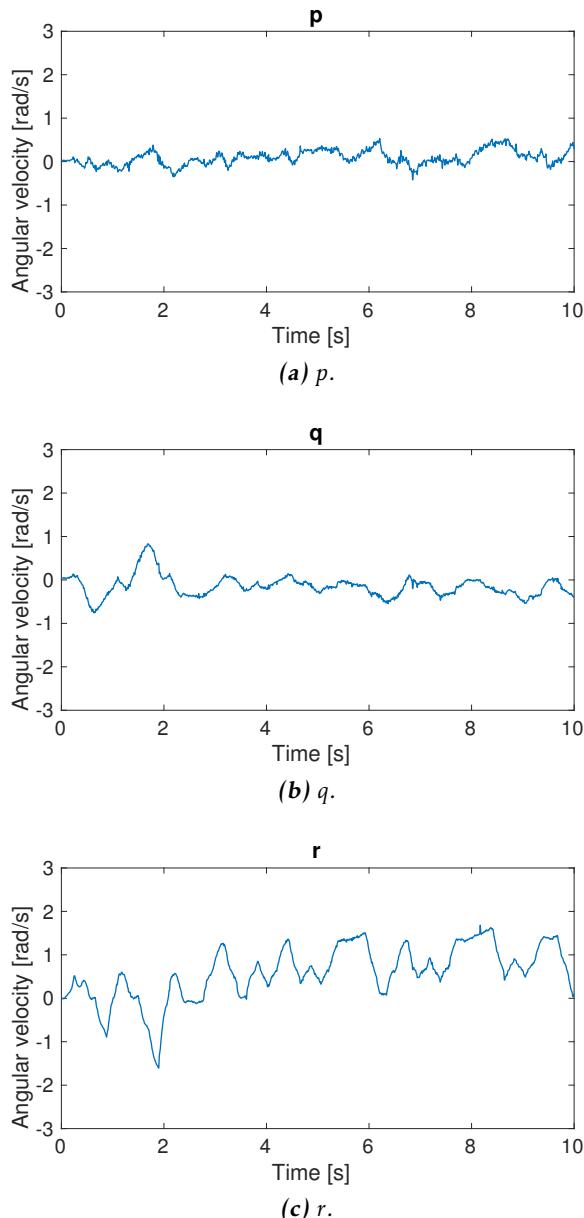
$$\hat{x}_{k+1}(\theta) = f(\hat{x}_k(\theta), u_k, y_k, \theta) \quad (5.4)$$

and

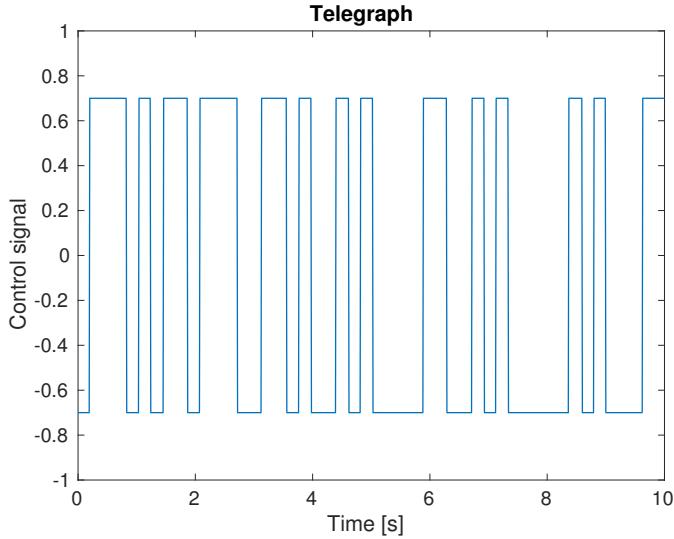
$$\hat{y}_k(\theta) = h(\hat{x}_k, u_k, \theta) \quad (5.5)$$



**Figure 5.1:** Plots of excitation of the three angular velocities  $p$ ,  $q$  and  $r$  from a test intended to mainly excite  $p$  and  $q$ . Note that the amplitude in  $r$  is four times smaller than that of  $p$  and  $q$ .



**Figure 5.2:** Plots of excitation of the three angular velocities  $p$ ,  $q$  and  $r$  from a test intended to mainly excite  $r$ . Note that the amplitude in  $r$  is approximately two times larger than that of  $p$  and  $q$ .



**Figure 5.3:** An example of a telegraph signal which was sent to a thruster during data collection experiments.

where  $\hat{x}_k$  is the estimated state vector,  $u_k$  are the inputs and  $\hat{y}_k(\theta)$  is the predicted output of the model (Larsson, 2013, p. 13).

The fitting is done by minimising a cost function  $V(\theta)$  with respect to the parameter vector  $\theta$  i.e.

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} V(\theta) \quad (5.6)$$

The cost function in this thesis has been defined as the squared error with an error threshold to handle noisy signals. An error threshold means that after a breakpoint the cost function becomes linear instead of quadratic, which makes the parameter estimation less sensitive to outliers (Ljung, 1999). To reduce the impact of the noise even further, the weight matrix was chosen as the inverse of the estimated noise covariance (Ljung, 1999). The cost function is

$$V(\theta) = \frac{1}{N} \left( \sum_{k \in \mathcal{I}} e_k^T(\theta) W(\theta) e_k(\theta) + \sum_{k \in \mathcal{J}} v_k^T(\theta) W(\theta) v_k(\theta) \right) \quad (5.7)$$

where  $N$  is the number of samples in the dataset,  $e_k(\theta)$  is the error vector with the parameter vector  $\theta$  and  $W$  is a positive definite weight matrix (Ljung, 1999). The set  $\mathcal{I}$  is the subset of indices for which  $|e_k(\theta)| < \sigma\rho$  holds. Here,  $\sigma$  is the estimated standard deviation of  $e_k(\theta)$  and  $\rho$  is the chosen error threshold. The set  $\mathcal{J}$  is the complement of  $\mathcal{I}$ . The error  $v_k(\theta)$  is defined as

$$v_k(\theta) = e_k(\theta) \sigma \frac{\rho}{\sqrt{|e_k(\theta)|}} \quad (5.8)$$

### 5.3 Estimation Using Prediction-Error Method

An initial set of model parameters were estimated using the decoupled models (3.38) and (3.39), together with appropriate data sets. These initial parameters were then used to estimate the parameters of model (3.37). Estimated angles were used as inputs and angular velocities as outputs. This gave the following model structure to be estimated

$$\dot{\hat{\eta}} = f(\hat{v}, \bar{\eta}, \tau) \quad (5.9)$$

and

$$\hat{y} = \hat{v} \quad (5.10)$$

where  $\bar{\eta}$  are the estimated Euler angles from the sensor fusion. The result of the estimation can be seen in Table 5.1. The fit of the model compared to validation data can be seen in Figure 5.4. The model has a high fit of 60 % in  $p$  and  $q$ , thus it describes the validation data well. The model describes validation data well in  $r$  which can be seen in the fit of 50 %.

The estimate of  $\eta$  did not follow the kinematic relations described in Section 3.1. The problem with  $\eta$  not following the kinematic relations was further investigated by comparing integration of (3.7) with  $\dot{\hat{\eta}}$  from the sensor fusion. As can be seen in Figure 5.5 the result was not satisfactory, with the estimated angles and the integration of (3.7) being dissimilar. It was therefore concluded that the estimated angles were unsuitable for use as outputs during parameter estimation unless a new motion model for the sensor fusion was created to solve the issue. It is therefore recommended that the validity of the observer is controlled using relations such as those in Section 3.1 before collecting data.

Due to the aforementioned problems, the model (3.37) was used with angular velocities and linear acceleration as outputs. Thus the estimation structure became

$$\dot{\hat{\eta}} = J(\hat{\eta})\hat{v}, \quad (5.11)$$

$$\dot{\hat{v}} = f(\hat{\eta}, \hat{v}, \tau) \quad (5.12)$$

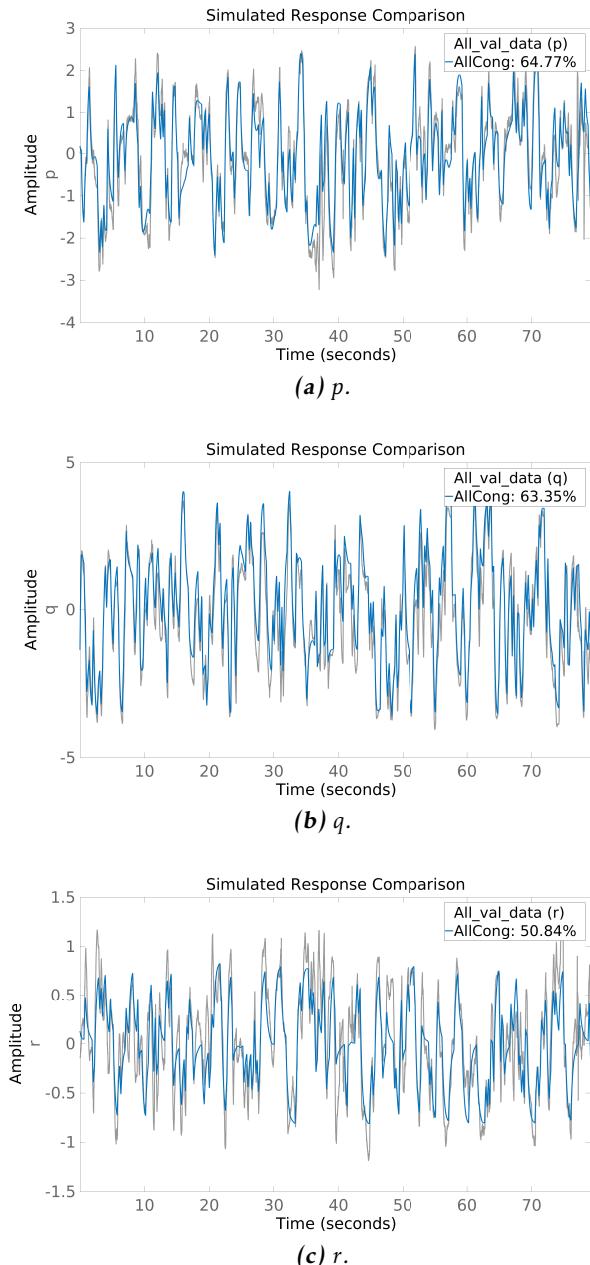
with

$$\hat{y} = \begin{bmatrix} \hat{v} \\ \hat{a} \end{bmatrix} \quad (5.13)$$

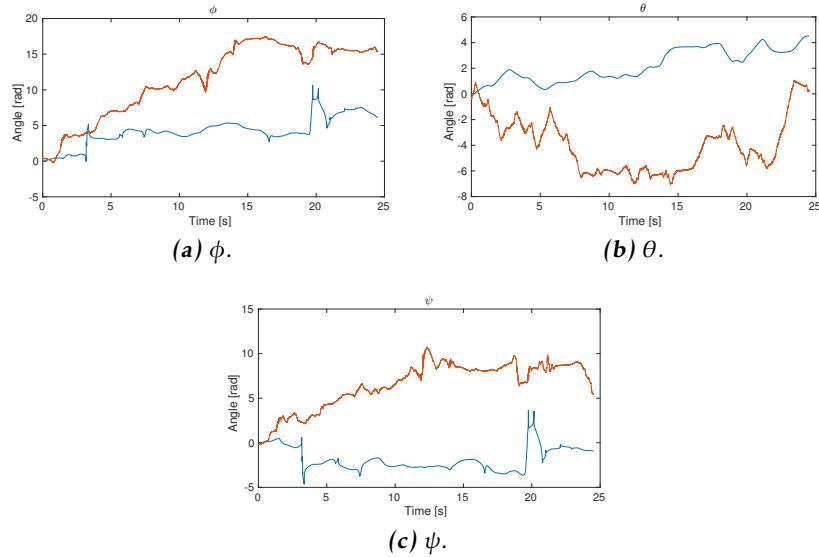
where  $\hat{a}$  is the estimated linear acceleration in the body frame and  $v$  is a state. The measurement equation for  $a$  is

$$a = \begin{bmatrix} 2g\eta\epsilon_2 - 2g\epsilon_1\epsilon_3 \\ -2g\eta\epsilon_1 - 2g\epsilon_2\epsilon_3 \\ 2g\epsilon_1^2 + 2g\epsilon_2^2 - g \end{bmatrix} \quad (5.14)$$

An issue that was encountered when estimating the parameters was that when angular velocities and linear accelerations were used as outputs, the model was sensitive to the initial value of the quaternions. To examine this problem further,



**Figure 5.4:** Comparison of simulation of the attitude model (blue) against validation data (gray) using the estimated angles as inputs and angular velocities as outputs. The simulated model has a high fit compared to the validation data.

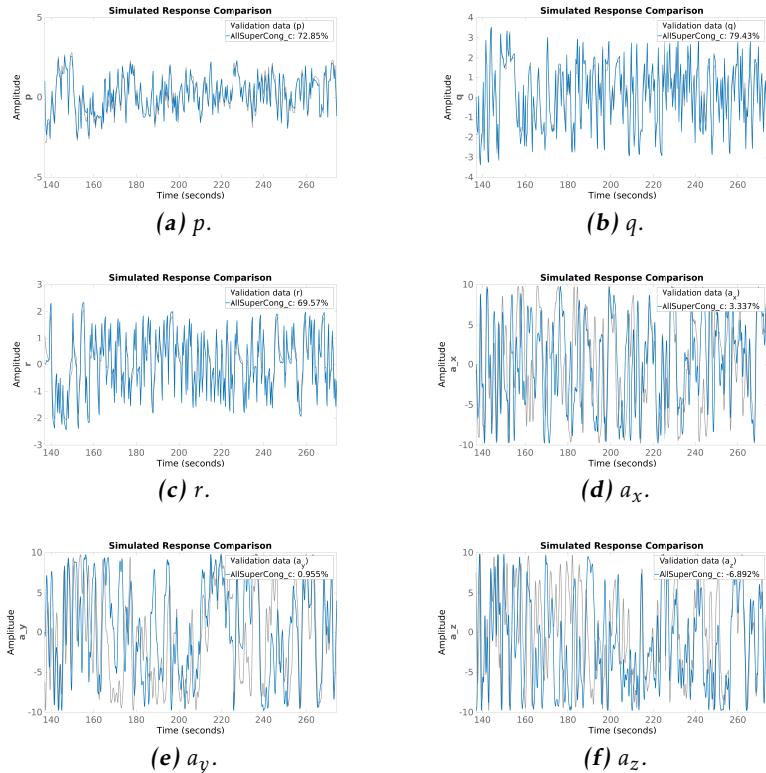


**Figure 5.5:** A comparison between integration of angular velocities (blue) using (3.7) and the estimated Euler angles from the EKF (red). The angles are poorly estimated and thus not suitable to use as outputs in parameter estimation.

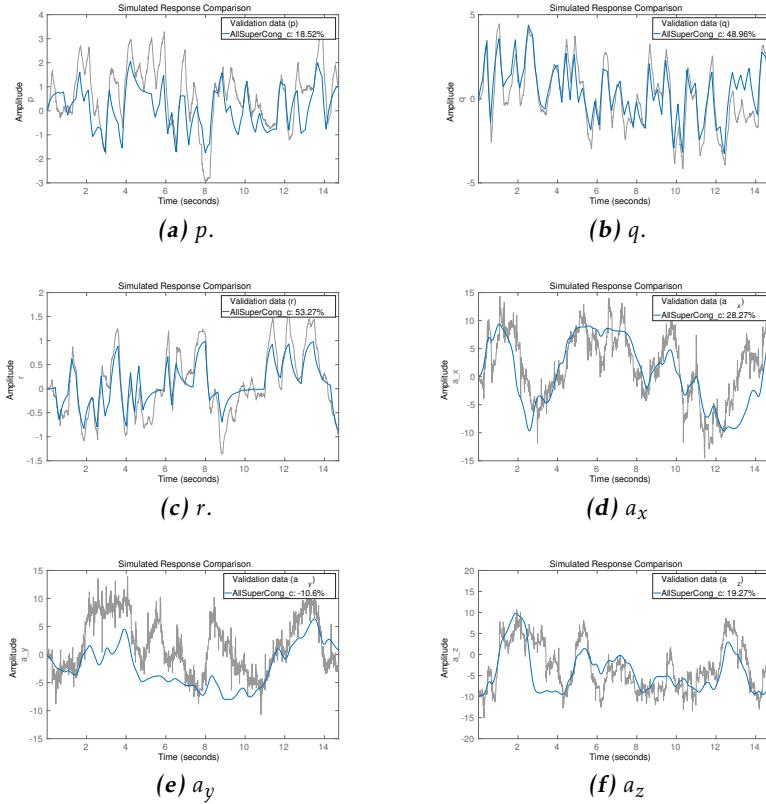
data was generated using a simulator and the simulated data was used in the estimation. The estimator was initialised with the correct initial parameter values from the simulator, but it was free to estimate the initial value of the quaternions. As expected, the estimated starting quaternions were far from the true initial values, which in turn led to the parameters diverging from their true values and a low fit was obtained even though the true parameters were used. A result from such a test can be seen in Figure 5.6.

A Kalman smoother described in Wallin and Zachrisson (2013) was used to estimate the initial state of the quaternions. The magnetometer was also added as an output in the Kalman smoother to further reduce the uncertainty of the initial quaternions. This issue could be avoided if the ROV is in a known state or the initial condition is logged at the start of the data collection.

The estimated parameter values obtained when using (3.37) with angular velocities and linear accelerations as inputs can be seen in Table 5.1. The fit of the model using the estimated parameters can be seen in Figure 5.7. The high fit of 50 % in  $q$  and  $r$  means that the model describes parts of the validation data well. The fit of the model with estimated angles as input is higher, but since that model did not follow the kinematic relations well, these results are considered more trustworthy. Unfortunately, the model did not describe the linear accelerations and  $p$  well, this will be discussed further in Section 5.6.



**Figure 5.6:** Comparison between simulated validation data (grey) and the simulated response from the estimated model (blue). The fit for the model in each state is stated in each plot. The validation data has been generated using the initial parameters used in the parameter estimation. The poor initial state estimate results in that the estimated parameters diverge from their true values.



**Figure 5.7:** Comparison of validation data (grey) against the simulated response from the model (blue). The fit for the model in each state is stated in each plot. The estimation used angular velocities and linear accelerations as outputs. The model describes  $q$  and  $r$  well which can be seen in the high fit. The model does not describe the linear accelerations and  $p$  well, this can be seen in the low fit.

## 5.4 Extended Kalman Filter Estimation

To circumvent the issues of estimating the initial state during the prediction-error method, an EKF was used to estimate the parameters. An EKF can be used to estimate parameters by extending the state vector of the EKF with the parameters that need to be estimated (Larsson, 2013). Extending the state vector gives the following augmented state

$$\bar{\mathbf{x}}_k = \begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\theta}_k \end{bmatrix} \quad (5.15)$$

where  $\boldsymbol{\theta}_k$  are the modelled parameters and time  $kT_s$ . The added parameters were modelled using constant position, which gave the following augmented state-space form

$$\bar{\mathbf{x}}_{k+1}(\boldsymbol{\theta}) = \begin{bmatrix} f(\mathbf{x}_k(\boldsymbol{\theta}), \mathbf{u}_k, \boldsymbol{\theta}) \\ \boldsymbol{\theta}_k \end{bmatrix} + \boldsymbol{\nu} \quad (5.16)$$

$$\mathbf{y}_k = \mathbf{h}(\bar{\mathbf{x}}_k(\boldsymbol{\theta}), \mathbf{u}_k) \quad (5.17)$$

Since the EKF tries to minimise the state variance, it will also try to estimate the parameters  $\boldsymbol{\theta}$  if they are included in the state vector (Larsson, 2013).

To increase the performance of the filter, a method for outlier detection and rejection was implemented. Since time updates were performed batch wise it was decided to use a different form of outlier rejection and not to reimplement the method described in Section 4.3. The outlier rejection method is based on the assumption that the normalised innovation

$$\boldsymbol{\epsilon}^T \mathbf{S}^{-1} \boldsymbol{\epsilon} \sim \chi_n^2 \quad (5.18)$$

can be used. Here,  $n$  is the number of measurements. To check the validity of a measurement from one sensor, (5.18) gives that  $\epsilon_i^2 / S_{i,i} \sim \chi_1^2$  (Gustafsson, 2012). Using this assumption, it is possible to eliminate bad measurements from individual sensors instead of rejecting all sampled data at that time instance. This is implemented in the EKF using Algorithm 2.

---

**Algorithm 2** The outlier rejection algorithm used during the measurement update step of the parameter estimation EKF.

---

1. Calculate  $\mathbf{S} = \mathbf{h}_x \mathbf{P} \mathbf{h}_x^T + \mathbf{R}$  and the innovation  $\boldsymbol{\epsilon}$ .
2. For each row  $i$  in  $\boldsymbol{\epsilon}$  do the comparison

$$\epsilon_i^2 > \sigma_i S_{i,i} \quad (5.19)$$

If the expression holds true, remove the  $i$ :th row from  $\boldsymbol{\epsilon}$  and the  $i$ :th row and column from  $\mathbf{S}$  and  $\mathbf{h}_x$ .

3. Proceed with the Kalman algorithm using the cropped  $\mathbf{h}_x$ ,  $\mathbf{S}$  and  $\boldsymbol{\epsilon}$ .

Here,  $\sigma$  is a  $n \times 1$  dimensional design variable and  $n$  is the number of measurements that are used in the EKF. A higher value of  $\sigma_i$  decreases the sensitivity of the outlier rejection for the  $i$ :th measurement.

---

## 5.5 Estimation Using an Extended Kalman Filter

The EKF estimation used the method described in Section 5.4, with additional bias states. The model (3.37) with quaternions as attitude representation was used to model  $\nu$ . The estimator used the process noise covariance matrix

$$Q = \text{diag}[\underbrace{1000}_{\eta}^{\times 4} \dots \underbrace{1000}_{\nu}^{\times 3} \dots \underbrace{0.01}_{b}^{\times 3} \dots \underbrace{0.001}_{\theta}^{\times 10}],$$

and the measurement noise covariance matrix

$$R = \text{diag}[\underbrace{0.001}_{\text{Gyro}}^{\times 3} \dots \underbrace{0.1}_{\text{Acc}}^{\times 3} \dots \underbrace{1000}_{\text{mag}}^{\times 3}]$$

Three data sets were fed to the estimator, where a fourth and fifth were used as validation data. All data sets were of the type with excitations in  $p$ ,  $q$  and  $r$  simultaneously. Parameter values from the estimation can be viewed together with the initial states in Table 5.2. The fit of the model using the estimated parameters can be viewed in Figure 5.8. The high fit in  $q$  and  $r$  indicates that the model describes the data well. However, the model does not describe the validation data well in  $p$ .

## 5.6 Investigating Low Fit in $p$ Dynamics

An issue that was present in Section 5.3 and Section 5.5 was a low fit in  $p$  during estimation. This was further investigated by manually testing each thruster individually and visual inspecting the response of the ROV. During tests it was discovered that thruster 6, see Figure 3.2, had minimal effect on the  $p$ -dynamics. A test showed that thruster 6 only had significant effect in  $p$  when it was actuated quickly. If the sixth thrusters power was incremented slowly an initial response in  $p$  was noted before it settled at zero again. The effect of thruster 6 in  $p$  can be seen in Figure 5.9. Thruster 6 also affected  $r$ , this can be due to unmodelled water interaction or that the thruster is in front or behind the center of rotation.

It was decided that thruster 6 was to be eliminated from the estimation, and a second estimation was run with the moment arm for thruster 6  $l_{z_6}$  fixed to zero. The filter was once again initialised with the following settings

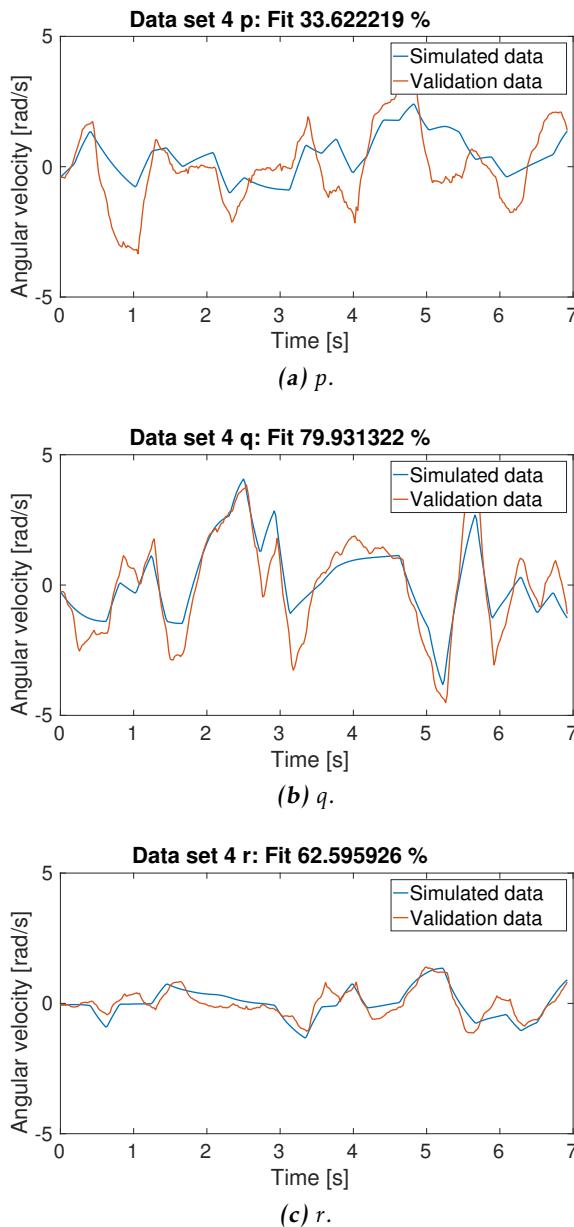
$$Q = \text{diag}[\underbrace{1000}_{\eta}^{\times 4} \dots \underbrace{1000}_{\nu}^{\times 3} \dots \underbrace{0.01}_{b}^{\times 3} \dots \underbrace{0.001}_{\theta}^{\times 10}]$$

**Table 5.1:** The estimated parameters from the prediction-error method using estimated angles as inputs and angular velocities as outputs (left) and the estimated parameters from the prediction-error method using angular velocities and linear acceleration as outputs (right).

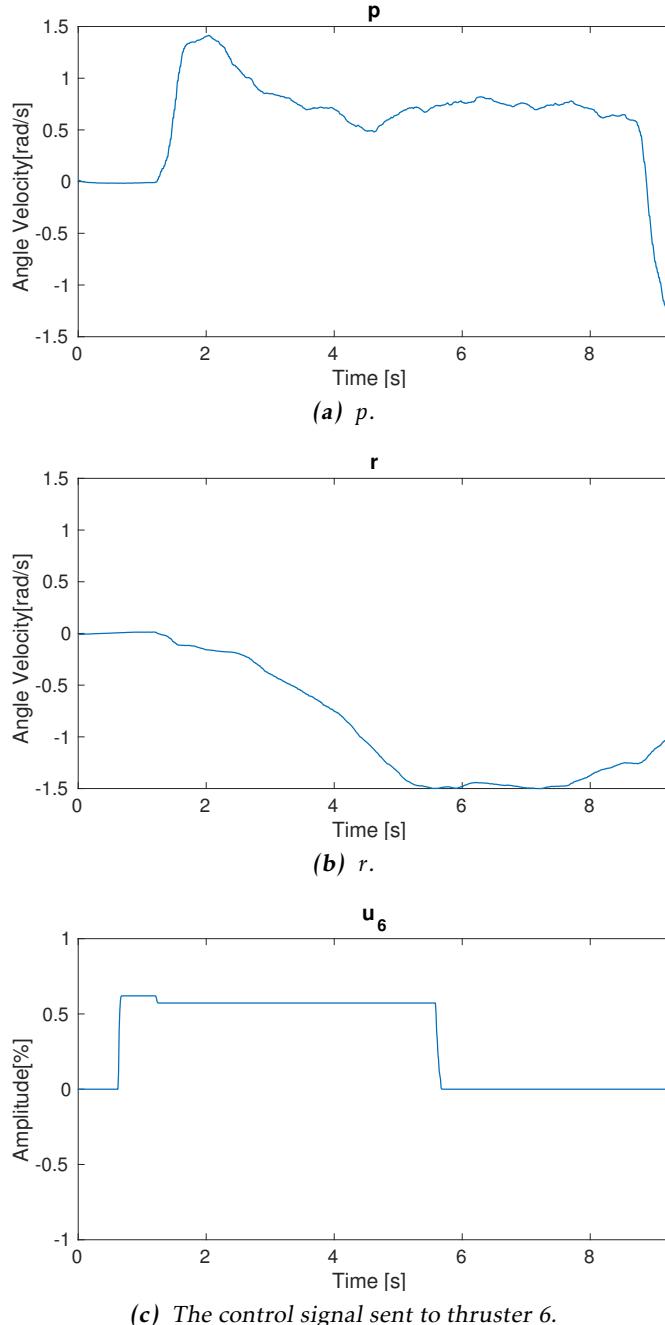
Notation	Starting Value	Estimated Value	Estimated Value
$z_B$	-0.01 m	-0.0178 m	-0.0294 m
$K_p$	-1 kg m <sup>2</sup>	-1.3275 kg m <sup>2</sup>	-2.5940 kg m <sup>2</sup>
$K_{p p }$	-1 kg m <sup>2</sup>	0 kg m <sup>2</sup>	-0.3092 kg m <sup>2</sup>
$M_q$	-1 kg m <sup>2</sup>	-1.1925 kg m <sup>2</sup>	-2.0425 kg m <sup>2</sup>
$M_{q q }$	-1 kg m <sup>2</sup>	-0.1094 kg m <sup>2</sup>	-0.0071 kg m <sup>2</sup>
$N_r$	-1 kg m <sup>2</sup>	-2.7838 kg m <sup>2</sup>	-2.9364 kg m <sup>2</sup>
$N_{r r }$	-1 kg m <sup>2</sup>	-0.6751 kg m <sup>2</sup>	-2.1843 kg m <sup>2</sup>
$A_p$	1.5 kg m <sup>2</sup>	0.3255 kg m <sup>2</sup>	0.7186 kg m <sup>2</sup>
$B_q$	1.5 kg m <sup>2</sup>	0.3753 kg m <sup>2</sup>	0.6112 kg m <sup>2</sup>
$C_r$	1.5 kg m <sup>2</sup>	0.9546 kg m <sup>2</sup>	1.0981 kg m <sup>2</sup>

**Table 5.2:** The estimated parameters from the Kalman estimator method. Moment arms are fixed to measured values.

Notation	Starting Value	Estimated Value
$\eta$	$[1 \ 0 \ 0 \ 0]^T$	
$v$	$[0 \ 0 \ 0]^T$	
$b$	$[0 \ 0 \ 0]^T$	
$z_B$	-0.05 m	-0.0463 m
$K_p$	-1 kg m <sup>2</sup>	-0.9163 kg m <sup>2</sup>
$K_{p p }$	-1 kg m <sup>2</sup>	-0.7591 kg m <sup>2</sup>
$M_q$	-1 kg m <sup>2</sup>	-0.8557 kg m <sup>2</sup>
$M_{q q }$	-1 kg m <sup>2</sup>	-0.3396 kg m <sup>2</sup>
$N_r$	-1 kg m <sup>2</sup>	-1.0266 kg m <sup>2</sup>
$N_{r r }$	-1 kg m <sup>2</sup>	-1.0236 kg m <sup>2</sup>
$A_p$	1 kg m <sup>2</sup>	1.0924 kg m <sup>2</sup>
$B_q$	1 kg m <sup>2</sup>	0.8162 kg m <sup>2</sup>
$C_r$	1 kg m <sup>2</sup>	1.1519 kg m <sup>2</sup>



**Figure 5.8:** Comparison between simulation of  $\nu$  (blue) with validation data (red). Moment-arm parameters are fixed. Quality of fit statistic is displayed at the top of each sub-figure. The model describes the validation data well in *q* and *r* which can be seen in the high fit. The low fit in *p* indicates that the model does not describe the validation data well.



**Figure 5.9:** The effect in  $p$  and  $r$  while only using thruster 6. As can be seen thruster 6 has a relatively small effect in  $p$ , thus ought the moment arm of thruster 6 be small. However, thruster 6 affected  $p$  more when thruster 6 was quickly actuated.

and

$$R = \text{diag}[\underbrace{0.001 \dots}_{\text{Gyro}}^{\times 3}, \underbrace{0.1 \dots}_{\text{Acc}}^{\times 3}, \underbrace{1000}_{\text{mag}}^{\times 3}]$$

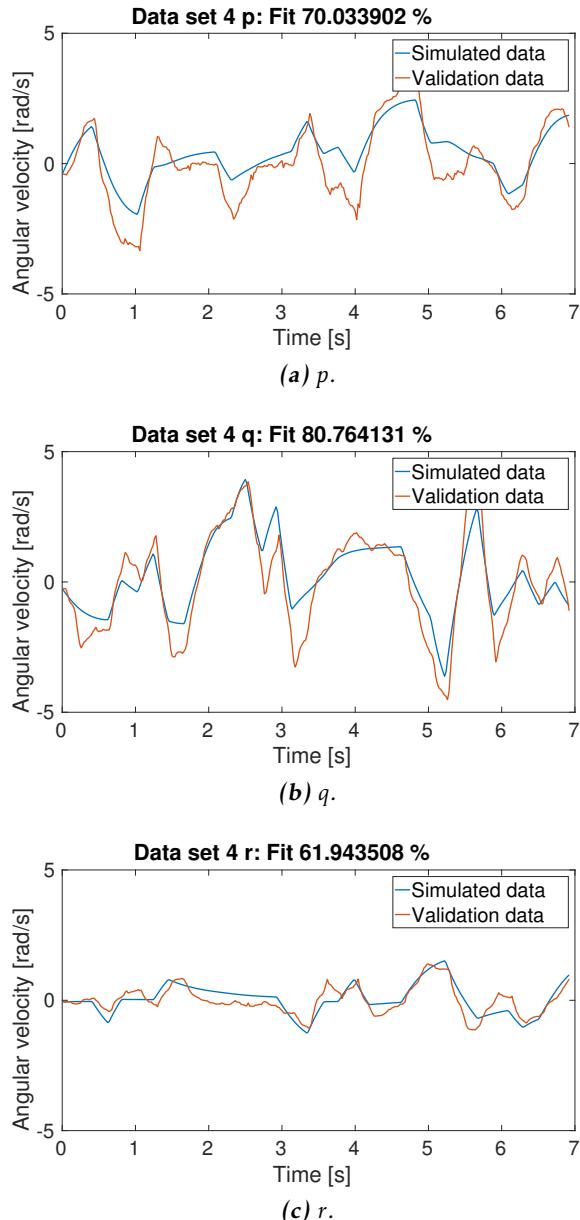
The estimation was done with five data sets, of which the first three were used for estimation while the fourth and fifth were used for validation. The estimator was iterated 2 times, using the parameter values from the previous iteration as the new initial states. The resulting parameter values and their initial values can be viewed in Table 5.3 and a comparison of a simulated run against validation data is shown in Figure 5.10. As can clearly be seen in Figure 5.10, a drastic improvement in fit for  $p$  is noted. It is unclear what was causing thruster 6 to diminish the fit of the model, but there are some possible explanations. The assumption in Section 3, that CG and the CO are placed at the same place in the ROV may be invalid. If the CG is placed lower than the CO the measured moment arm of thruster 6  $l_{z_6}$  will be to long. This will result in the effect in  $p$  of thruster 6 to be less than expected. A similar performance increase was achieved when using the prediction-error method with  $l_{z_6} = 0$ .

**Table 5.3:** The estimated parameters from the Kalman estimator method with moment arms fixed to measured values but with  $l_{z_6}$  fixed to zero.

Notation	Starting Value	Estimated Value
$\eta$	$[1 \ 0 \ 0 \ 0]^T$	
$v$	$[0 \ 0 \ 0]^T$	
$b$	$[0 \ 0 \ 0]^T$	
$z_B$	-0.05 m	-0.0420 m
$K_p$	-1 kg m <sup>2</sup>	-0.8842 kg m <sup>2</sup>
$K_{p p }$	-1 kg m <sup>2</sup>	-0.6682 kg m <sup>2</sup>
$M_q$	-1 kg m <sup>2</sup>	-0.8547 kg m <sup>2</sup>
$M_{q q }$	-1 kg m <sup>2</sup>	-0.3354 kg m <sup>2</sup>
$N_r$	-1 kg m <sup>2</sup>	-1.0280 kg m <sup>2</sup>
$N_{r r }$	-1 kg m <sup>2</sup>	-1.0249 kg m <sup>2</sup>
$A_p$	1 kg m <sup>2</sup>	0.8337 kg m <sup>2</sup>
$B_q$	1 kg m <sup>2</sup>	0.7987 kg m <sup>2</sup>
$C_r$	1 kg m <sup>2</sup>	1.1250 kg m <sup>2</sup>

## 5.7 Estimated Parameters

Table 5.4 shows the known, measured and estimated parameters used in the ROV and controller development. These were chosen solely based on fit to validation data during simulation.



**Figure 5.10:** Comparison between simulation of  $\nu$  (blue) with validation data (red). Moment-arm parameters are fixed and  $l_{z_6}$  is set to zero. Quality of fit statistic is displayed at the top of each sub-figure.

**Table 5.4:** The known, measured and estimated parameters used in the ROV model.

Notation	Value	Description
$m$	6.621 kg	Mass of the ROV.
$g$	9.82 m/s <sup>2</sup>	Gravity acceleration.
$\rho$	1000 kg/m <sup>3</sup>	Density of water.
$l_{x_1}$	0.19 m	Distance from CG to thruster 1 in $x$ -direction.
$l_{y_1}$	0.11 m	Distance from CG to thruster 1 in $y$ -direction.
$l_{y_2}$	0.11 m	Distance from CG to thruster 2 in $y$ -direction.
$l_{x_2}$	0.19 m	Distance from CG to thruster 2 in $x$ -direction.
$l_{y_3}$	0.11 m	Distance from CG to thruster 3 in $y$ -direction.
$l_{x_5}$	0.17 m	Distance from CG to thruster 5 in $x$ -direction.
$l_{y_4}$	0.11 m	Distance from CG to thruster 4 in $y$ -direction.
$l_{z_6}$	0 m	Distance from CG to thruster 6 in $z$ -direction.
$z_B$	-0.0420 m	Distance from CG to CB.
$K_p$	-0.8842 kg m <sup>2</sup>	Linear damping coefficient due to rotation in water about the $x$ -axis.
$K_{p p }$	-0.6682 kg m <sup>2</sup>	Quadratic damping coefficient due to rotation in water about the $x$ -axis.
$M_q$	-0.8547 kg m <sup>2</sup>	Linear damping coefficient due to rotation in water about the $y$ -axis.
$M_{q q }$	-0.3354 kg m <sup>2</sup>	Quadratic damping coefficient due to rotation in water about the $y$ -axis.
$N_r$	-1.0280 kg m <sup>2</sup>	Linear damping coefficient due to rotation in water about the $z$ -axis.
$N_{r r }$	-1.0249 kg m <sup>2</sup>	Quadratic damping coefficient due to rotation in water about the $z$ -axis.
$A_p$	0.8337 kg m <sup>2</sup>	Inertia around the $x$ -axis and increased inertia around the $x$ -axis.
$B_q$	0.7987 kg m <sup>2</sup>	Inertia around the $y$ -axis and increased inertia around the $y$ -axis.
$C_r$	1.1250 kg m <sup>2</sup>	Inertia around the $z$ -axis and increased inertia around the $z$ -axis.

# 6

---

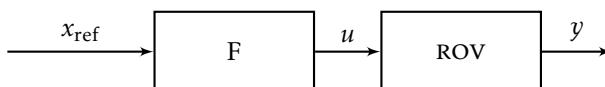
## Controlling the ROV

Automatic control is a way of regulating a process without direct human interaction. The complexity can vary from decentralised proportional, integral and derivative controllers (PIPs) to more advanced model-based control methods, such as model predictive control (MPC) (Glad and Ljung, 2003).

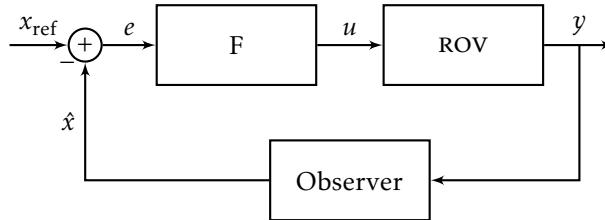
There are two main concepts of control, open-loop and feedback control (Glad and Ljung, 2012). An open-loop controller is a controller that computes its output based on a model of the system. A disadvantage of open-loop controllers is that they require exact knowledge of the controlled system (Glad and Ljung, 2012). Figure 6.1 illustrates the open-loop control scheme used in the ROV.

A feedback controller uses measurements of the outputs in a system to get a desired behaviour. One such method is error-controlled regulation, where the difference between the desired value, the setpoint, of an output and its measured or estimated value is used for control (Glad and Ljung, 2012). The feedback control scheme used in the ROV can be seen in Figure 6.2.

Similarly to an open-loop controller, a feedback controller can use a model of the system. Using a model of the controlled system in the control structure can produce better performance and compensate for unwanted effects such as non-



**Figure 6.1:** The open-loop control scheme used in the ROV. The control block  $F$  can be any type of open-loop control. Notice that this is an ideal case where no disturbances affect the system.



**Figure 6.2:** The feedback control scheme used in the ROV. The controller  $F$  can be any of the controllers discussed in this chapter. The observer in the ROV is the sensor fusion described in Chapter 4. Notice that this is an ideal case where no disturbances effect the system.

linearities. Compensating for non-linearities is desired due to the fact that a lot of control principles are based on linear systems (Glad and Ljung, 2003).

## 6.1 Open-Loop Control

The open-loop control of the ROV consists of a static thrust-allocation matrix which is

$$\mathbf{T}_G^+ = \mathbf{T}_G^T (\mathbf{T}_G \mathbf{T}_G^T)^{-1} \quad (6.1)$$

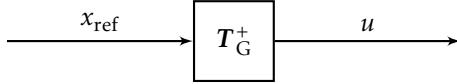
where  $\mathbf{T}_G$  describes how the actuators effect the ROV (Garus, 2004). If the thrust-geometry matrix is chosen as

$$\mathbf{T}_G = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ -1 & -1 & 0 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \quad (6.2)$$

maximum thrust can be achieved but coupling effects can occur. Combining (6.1) and (6.2) gives

$$\mathbf{T}_G^+ = \begin{bmatrix} 0 & 0 & -0.25 & 0.5 & 0.25 & 0 \\ 0 & 0 & -0.25 & -0.5 & 0.25 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0 & -0.5 \\ 0 & 0 & -0.5 & 0 & -0.5 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.3)$$

The static thrust-allocation matrix  $\mathbf{T}_G^+$  is the pseudo inverse of the thrust-geometry matrix  $\mathbf{T}_G$ . An approximately decoupled control is achieved when the static thrust allocation matrix is used and thus the ROV can be controlled better. Figure 6.3 illustrates how the control signals are allocated to the different thrusters when given a control input. The thrust-geometry matrix could be chosen as



**Figure 6.3:** The open-loop control allocate control signals using the thrust-allocation matrix  $T_G^+$ . The static thrust-allocation matrix approximately decouples the system.

$T_G = T_{\text{normed}}$ . Here,  $T_{\text{normed}}$  is the row-wise normed thrust matrix  $T$  from Section 3.5. The use of the aforementioned thruster matrix reduces coupling effects but was not implemented on the ROV since maximum thrust would be reduced.

## 6.2 Feedback Linearisation

Feedback linearisation is to compensate for the non-linearities in a system using a non-linear control law. The goal is to make the system linear from an input-output perspective (Glad and Ljung, 2003). Using the model structure from Chapter 3 and the estimated parameters from Chapter 5, a non-linear control law was created

$$\tau_{\text{Lin}} = G^{-1}(\hat{\eta}_2, \hat{v}_2, \mathbf{a}^b) = \mathbf{M}\mathbf{a}^b + \mathbf{D}(\hat{v}_2)\hat{v}_2 + \mathbf{C}(\hat{v}_2)\hat{v}_2 + \mathbf{g}(\hat{\eta}_2) \quad (6.4)$$

where  $\mathbf{a}^b$  is the desired angular acceleration in the body-fixed frame and  $\tau_{\text{Lin}}$  is the estimated generalised force needed to achieve the desired acceleration (Fossen, 2011). The desired control signal  $\mathbf{u}_{\text{Lin}}$  was then chosen as

$$\mathbf{u}_{\text{Lin}} = f^{-1}(T^{-1}\bar{\tau}_{\text{Lin}}) \quad (6.5)$$

where  $T$  is the geometry matrix defined in (3.32),  $\bar{\tau}_{\text{Lin}} = [0 \ 0 \ 0 \ \tau_{\text{Lin}}^T]^T$  and  $f(\cdot)$  is the look-up table from control signal,  $u \in [-1 \ 1]$ , to thrust defined in Appendix C. In an ideal case, where the model is exact, using the non-linear control law (6.4) would produce the system

$$\dot{\mathbf{v}}_2 = \mathbf{a}^b \quad (6.6)$$

where  $\mathbf{a}^b$  could be chosen using any desired control method (Fossen, 2011, p.451).

## 6.3 Attitude Controller

An attitude controller was also implemented on the ROV. The controller was chosen as a PID-controller utilising feedback linearisation (6.4). Since (6.4) aims to linearise the angular accelerations in the body-fixed frame, these have to be transformed into the global coordinate system. This is achieved by choosing

$$\mathbf{a}^b = L(\mathbf{a}^n, \hat{\Theta}, \hat{v}_2) = T_\theta^{-1}(\hat{\Theta})(\mathbf{a}^n - \dot{T}_\theta(\hat{\Theta})\hat{v}_2) \quad (6.7)$$

where  $\alpha^n$  is the desired angular acceleration in the global-frame (Fossen, 2011). This gives the following linear system

$$\ddot{\eta}_2 = \alpha^n \quad (6.8)$$

Like the desired body-fix acceleration  $a^b$ , the desired global frame acceleration can be chosen using an arbitrary control method. Here, the control error is chosen as

$$\tilde{\eta}_2 = \hat{\eta}_2 - \eta_{2,\text{ref}} \quad (6.9)$$

The desired acceleration  $\alpha^n$  could be chosen using the following feedback

$$\alpha^n = -K_p \tilde{\eta}_2 - K_i \int \tilde{\eta}_2 dt - K_d \dot{\tilde{\eta}}_2 \quad (6.10)$$

where  $K_p$ ,  $K_i$  and  $K_d$  are positive definite design matrices (Fossen, 2011, p. 453). Using (3.10) and the assumption that  $\eta_{2,\text{ref}}$  is piece-wise constant, the derivative of the attitude error could be defined as

$$\dot{\tilde{\eta}}_2 = T_\theta(\hat{\Theta})\hat{v}_2 \quad (6.11)$$

Combining (6.10) with (6.11) gives the feedback

$$\alpha^n = -K_p \tilde{\eta}_2 - K_i \int \tilde{\eta}_2 dt - K_d T_\theta(\hat{\Theta})\hat{v}_2 \quad (6.12)$$

The attitude controller was also combined with an open-loop control of the linear velocities

$$u = \underbrace{f^{-1}(T^{-1}G^{-1}(\hat{\eta}_2, \hat{v}_2, L(\alpha^n, \hat{\Theta}, \hat{v}_2)))}_{u_{\eta_2}} + \underbrace{T_G^+ \begin{bmatrix} v_{1,\text{ref}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}}_{u_{v_1}} \quad (6.13)$$

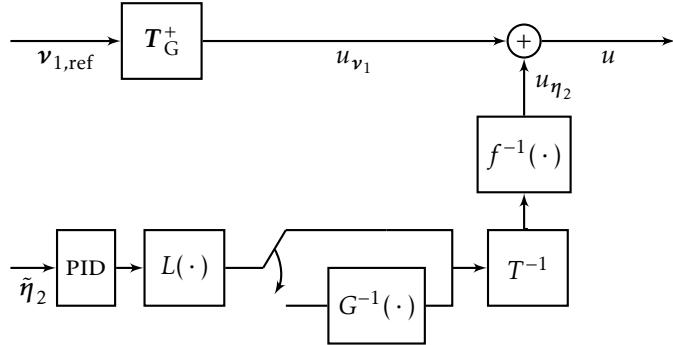
or

$$u = \underbrace{f^{-1}(T^{-1}L(\alpha^n, \hat{\Theta}, \hat{v}_2))}_{u_{\eta_2}} + \underbrace{T_G^+ \begin{bmatrix} v_{1,\text{ref}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}}_{u_{v_1}} \quad (6.14)$$

without the feedback linearisation. This was implemented to allow the user to steer the ROV while the controller ensures that the ROV holds a given attitude. An illustration of how the attitude controller with open-loop control is implemented can be seen in Figure 6.4.

## 6.4 Angular Velocity Controller

An angular velocity controller was also implemented using the feedback linearisation (6.4). Since no transformation is needed in order to control the angular velocities in the body-fixed frame,  $a^b$  can be chosen using the following PI feed-



**Figure 6.4:** The linear velocities are controlled in the same way as in Section 6.1. Furthermore, the attitude is controlled via a PID and feedback linearisation can be enabled.

back

$$\mathbf{a}^b = -K_p \tilde{\mathbf{v}}_2 - K_i \int \tilde{\mathbf{v}}_2 dt \quad (6.15)$$

where  $K_p$  and  $K_i$  are positive definite design matrices and

$$\tilde{\mathbf{v}}_2 = \hat{\mathbf{v}}_2 - \mathbf{v}_{2,\text{ref}} \quad (6.16)$$

(Fossen, 2011, p. 453).

The angular velocities PI controller was also extended with an open-loop control solution for control of linear velocities

$$u = \underbrace{f^{-1}(T^{-1}G^{-1}(\hat{\eta}_2, \hat{\mathbf{v}}_2, \mathbf{a}^b))}_{u_{\mathbf{v}_2}} + \underbrace{T_G^+ \begin{bmatrix} \mathbf{v}_{1,\text{ref}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}}_{u_{\mathbf{v}_1}} \quad (6.17)$$

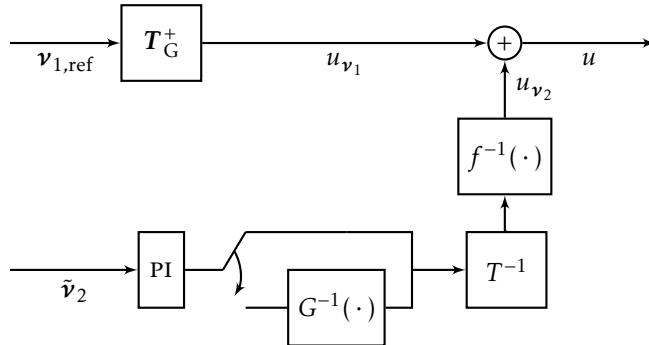
or alternatively

$$u = \underbrace{f^{-1}(T^{-1}\mathbf{a}^b)}_{u_{\mathbf{v}_2}} + \underbrace{T_G^+ \begin{bmatrix} \mathbf{v}_{1,\text{ref}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}}_{u_{\mathbf{v}_1}} \quad (6.18)$$

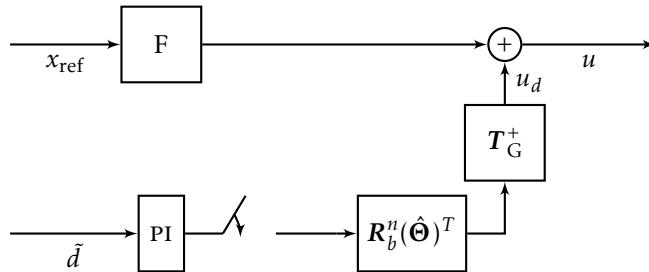
if the feedback linearisation is inactive. Figure 6.5 illustrates how the angular velocity controller is implemented in conjunction with the open-loop control of the linear velocities.

## 6.5 Depth Controller

A PI-depth controller was also implemented on the ROV. The depth controller was designed such that it could be used simultaneously with any other of the



**Figure 6.5:** The linear velocities are controlled in the same way as in Section 6.1. Furthermore, the angular velocities are controlled via a PI and feedback linearisation can be enabled.



**Figure 6.6:** The PI depth controller can be used when the open-loop control is engaged or when the other controllers are used. In this figure the  $F$  symbolises the chosen way of controlling the ROV.

aforementioned angular velocity and attitude controllers.

If the error in depth  $\tilde{d}$  is defined as

$$\tilde{d} = \hat{d} - d_{\text{ref}} \quad (6.19)$$

then the PI depth controller can be defined as

$$u_d = \mathbf{T}_G^+ \left[ \mathbf{R}_b^n(\hat{\Theta})^T \begin{bmatrix} 0 \\ 0 \\ -K_p \tilde{d} - K_i \int \tilde{d} dt \end{bmatrix} \right] \quad (6.20)$$

where  $K_p$  and  $K_i$  are design parameters. The rotation matrix  $\mathbf{R}_b^n(\hat{\Theta})^T$  is used to enable the depth controller to distribute control signals depending on the attitude of the ROV. This enables the controller to regulate the depth regardless of the ROV's attitude. The implementation of the depth controller can be seen in Figure 6.6.

**Table 6.1:** The parameters used in the PIDs.

	$K_p$	$K_i$	$K_d$
$\phi$	2	0.1	0.1
$\theta$	2.7	0.1	0.1
$\psi$	0.7	0.1	0.1
$p$	3.5	2	-
$q$	3.5	2	-
$r$	3.0	2	-
$z$	1	0.2	-

## 6.6 Benchmarking

To be able to draw any conclusions regarding the performance of the controllers the following reference signals were used during tests for each controller separately.

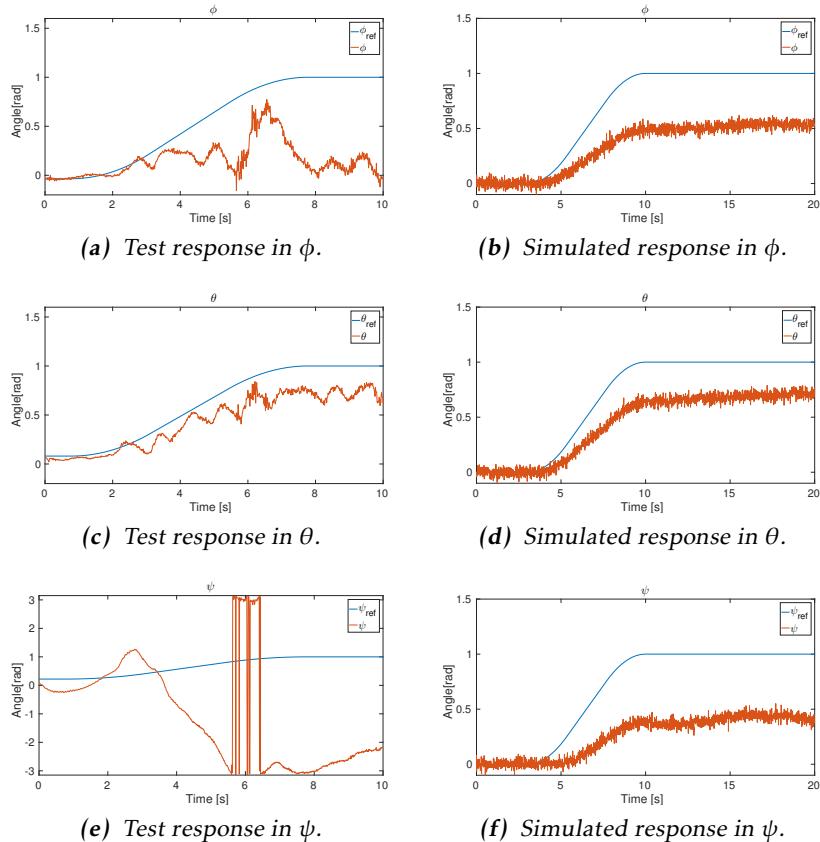
**Constant** A constant reference was applied to all DOF. This reference signal was only used for trimming the controllers initially, thus, no results are presented except for the depth controller.

**Sine** A  $\sin(\cdot)$  signal was applied to one DOF at the time and then to all DOF. Two sine signals of different amplitudes were used, amplitude 1 and 0.5. The frequency of the sines were 0.5 Hz.

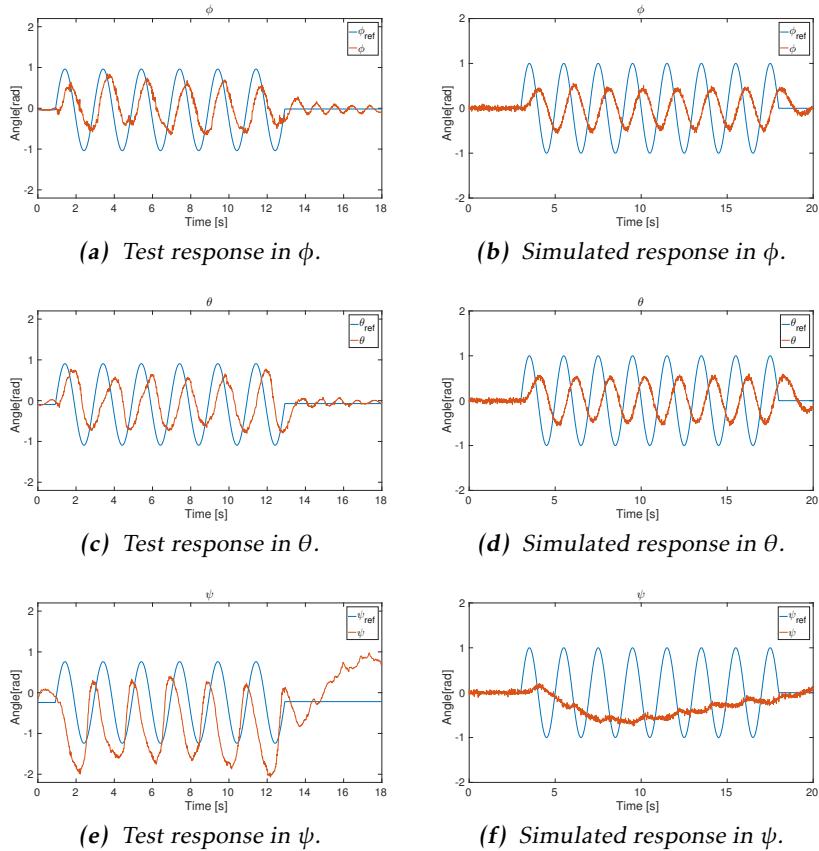
**Smooth step** A step with smooth acceleration, was applied to one DOF at the time and then to all DOFs at the same time. The used smooth step was the same as in Spong et al. (2006, p. 192). The smooth step parameters were  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$ .

For each conducted test, excluding the depth tests, a simulated test was also performed. The ROV simulator used the parameters from Section 5.7. The feedback linearisation, both in the simulator and the ROV, used the parameters from Section 5.7 with a scaling factor of 0.9 except for  $z_B$ , which was scaled by 0.5. In this section, a few representative test cases are presented. Several more tests were conducted and these can be seen in Appendix D. Tests were conducted with the PID attitude controller without feedback linearisation, the PI angular velocity controller with feedback linearisation and with the PI depth controller. The control parameters used in the PID and the PIs during simulation and field tests can be seen in Table 6.1. These were initially chosen using simulations and were further tuned during controller tests. Different performance measures, such as overshoot, undershoot and steady-state error are used to specify the performance of the controllers. These measures are defined as in Glad and Ljung (2003).

Figure 6.7 shows the smooth-step reference signals applied in all attitude angles while using the attitude controller without linearisation. Initially, the roll and pitch angle did not follow the reference signals. During field tests, the roll angle



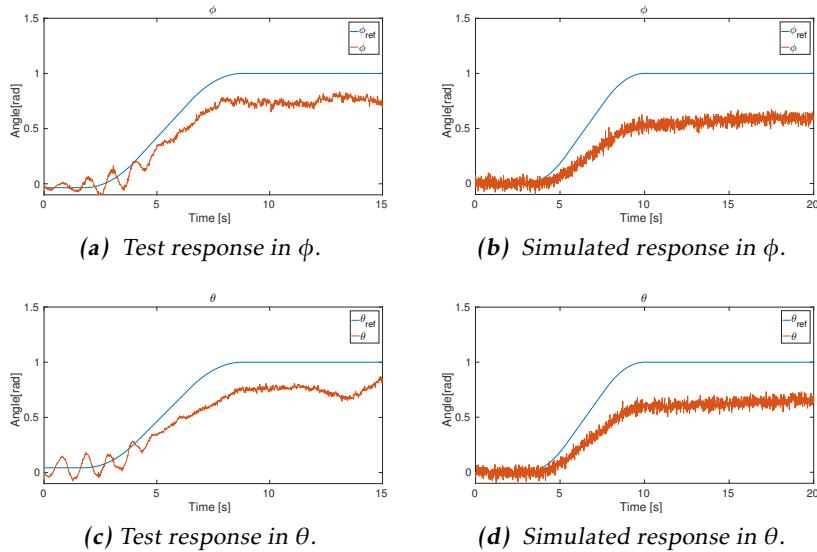
**Figure 6.7:** A smooth step with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  was applied in all attitude angles at the same time while using the attitude controller.



**Figure 6.8:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in all attitude angles at the same time while using the attitude controller.

was oscillative and had a steady-state error of 0.9. However, the control of the pitch angle was objectively better, with a steady-state error of 0.4. The yaw angle could not be stabilised during the field test and drifted throughout the tests. The attitude angles did not reach the reference signals in the simulated test case either. The pitch angle had a steady-state error of 0.3 while the other angles had a steady-state error of 0.5.

Results from an attitude test with sine reference signals of amplitude 1 and frequency 0.5 Hz can be seen in Figure 6.8. The roll angle and pitch angle did not reach the desired amplitudes in the field test. However, they followed the general form of a 0.5 Hz sine with a phase shift relative to the reference signals. The same phase shift and lack of amplitude was observed in simulations. The yaw angle followed the reference signal well during the live test but had a phase shift and a bias. The simulated result in yaw angle did not follow the reference signal at all. This may be caused by the feedback being too low for the system. This is a sign



**Figure 6.9:** Smooth steps with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$ ) were applied in  $\theta$  and  $\phi$  at the same time while using the attitude controller. The attitude angle  $\psi$  was kept free during the test.

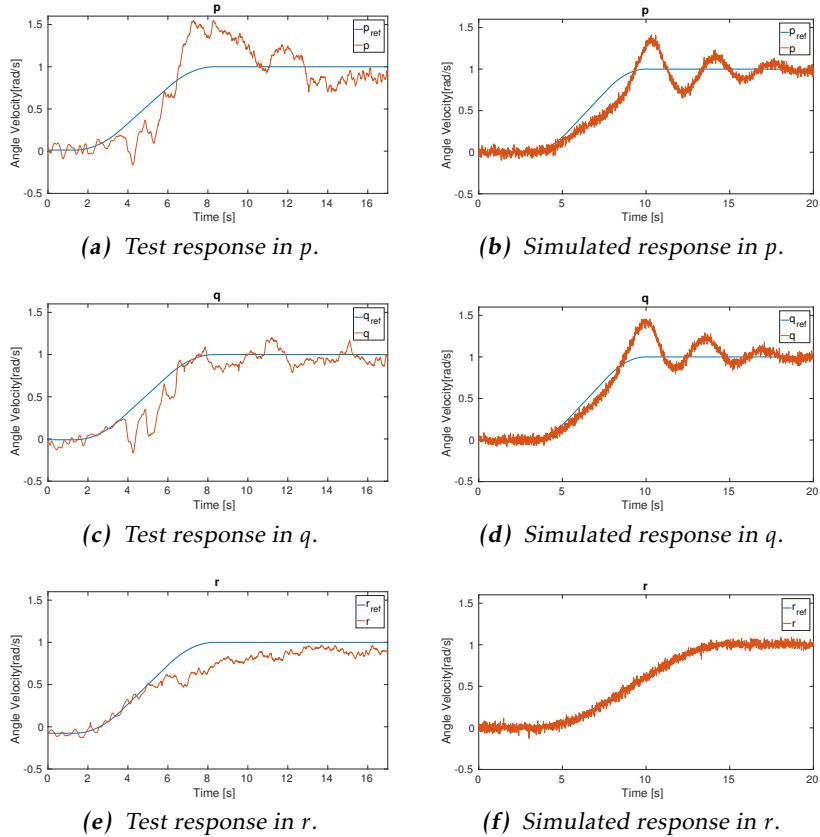
of a model error.

The result from a smooth step test in roll angle and pitch angle can be seen in Figure 6.9. During the test, the yaw angle was kept free. Both the roll angle and the pitch angle followed the shape of the reference signal but did not reach the desired magnitude. The steady-state errors for roll and pitch angle were approximately 0.35. The simulated result in roll angle and pitch angle performed worse than the live test. The simulated roll angle had a steady-state error 0.5 and the steady-state error in pitch angle was 0.4.

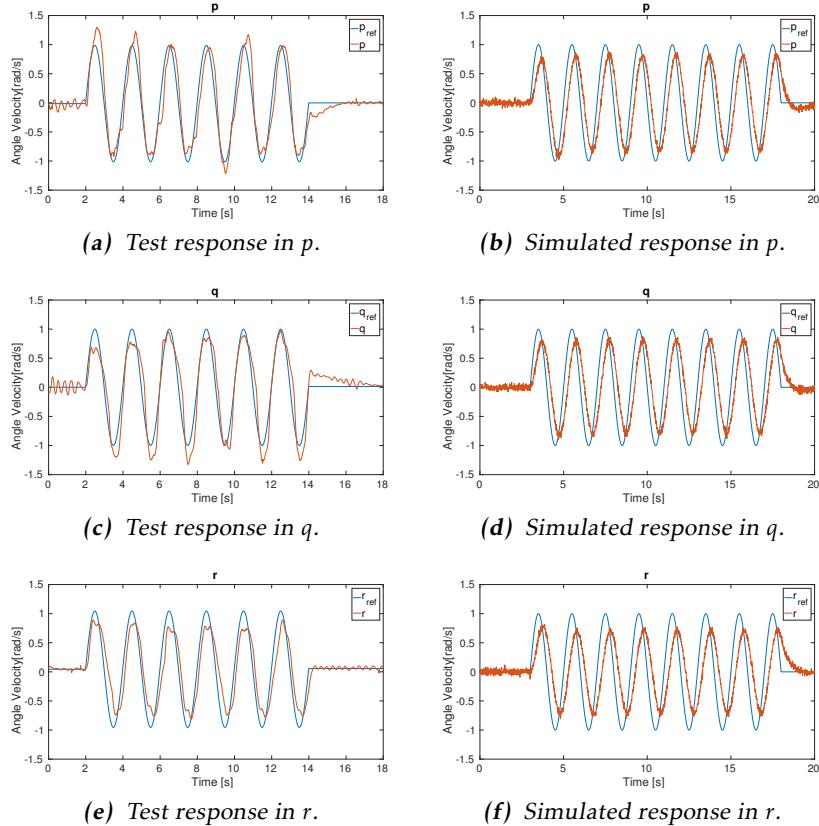
Figure 6.10 shows a smooth step applied in all angular velocities while using the angular velocity controller. An overshoot of 0.5 was obtained in  $p$  for the field test while the simulation it only had an overshoot of 0.2 but suffers from stronger oscillations. Both the field test and the simulation in  $p$  followed the reference signal and had a small steady-state error.

The field and simulated test in  $q$  followed the reference value well and had negligible steady-state errors with some oscillative behaviour. Moreover, the field test had no overshoot while the simulation in  $q$  had stronger oscillations and an overshoot of 0.4.

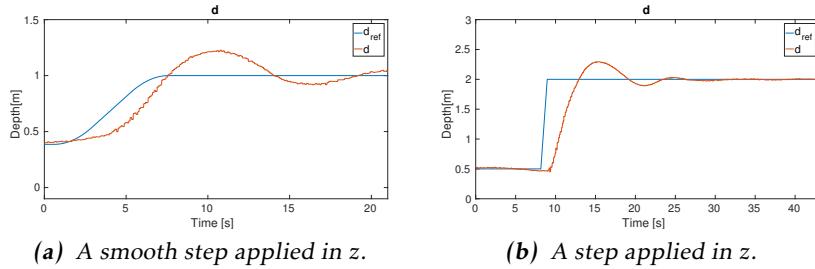
The simulated test in  $r$  performed well and followed the reference relatively well. The field test in  $r$  performed well, but it was slow to rise to the requested reference signal and had a small steady-state error of 0.1.



**Figure 6.10:** A smooth step with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$ ) was applied in all angular velocities at the same time while using the angular velocity controller.



**Figure 6.11:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in all angular velocities at the same time while using the angular velocity controller.



**Figure 6.12:** Steps applied to  $z$ . A smooth step from 0.4 m to 1 m is shown in (a) and a step from 0.5 m to 2 m is shown in (b).

A sine signal was also applied to all angular velocities while using the angular velocity controller. The results can be seen in Figure 6.11. During the field tests all the angular velocities followed the reference signals well, except for some overshoots and undershoots. The simulated test followed a general sine form but the reference amplitude was not reached and a phase shift could be noted.

Figure 6.12 shows a smooth step from 0.4 m to 1 m and a step from 0.5 m to 2 m. A delay in the response can be seen and overshoots of approximately 0.2.

## 6.7 Discussion

Due to problems during implementation of the controllers, they were not as well tuned as they could have been. This can be seen in the results since large steady-state errors were acquired during the tests. The PID-parameters were trimmed in simulation before live test but the feedback used in simulations was too strong for live tests. The parameters in the feedback and the linearising control law both had to be scaled down to achieve a stable system. Since the ROV is a non-linear and stable system, compensating for non-linearities run the risk of producing a non-linear and unstable system. This is further illustrated in Example 6.1. We believe such an over compensation was what caused most of the initial problems during controller tests. This idea is further strengthened by the fact that the controller instability diminished when the parameters in the linearising control law were scaled down.

---

### Example 6.1

---

Consider the non-linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -ax_1 - bx_2|x_2| + u \end{bmatrix} \quad (6.21)$$

Using feedback linearisation, the ideal case would be to use the control law

$$u = ax_1 + bx_2|x_2| + \bar{u} \quad (6.22)$$

which would yield the linear system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \bar{u} \end{bmatrix} \quad (6.23)$$

Here,  $\bar{u}$  can be used to achieve the desired dynamics, for example, using a PID-controller.

In practice however, both  $a$  and  $b$  have to be estimated with  $\hat{a}$  and  $\hat{b}$ . Using the estimated values  $\hat{a}$  and  $\hat{b}$  in (6.22) gives

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \tilde{a}x_1 + \tilde{b}x_2|x_2| + \bar{u} \end{bmatrix} \quad (6.24)$$

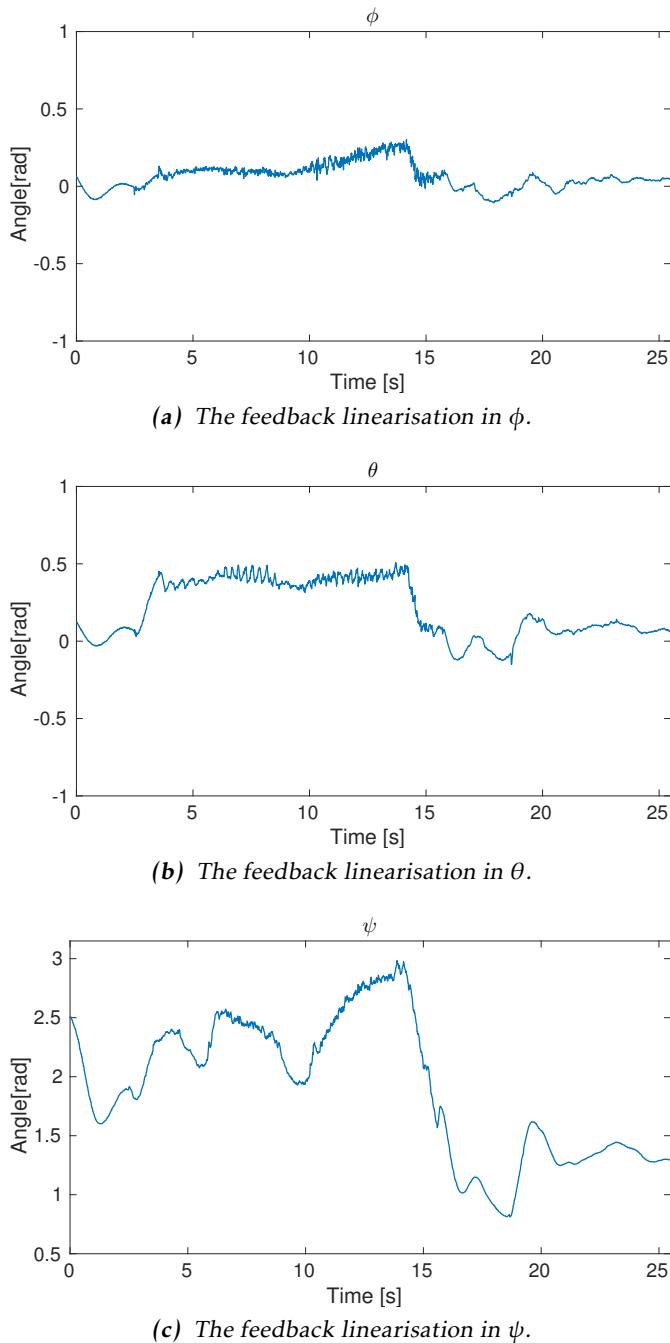
with  $\tilde{a} = \hat{a} - a$  and  $\tilde{b} = \hat{b} - b$ .

If  $\hat{a}$  and  $\hat{b}$  are estimated such that  $\tilde{a} > 0$  and  $\tilde{b} > 0$ , the control law has created an unstable non-linear system. The instability can nevertheless be compensated for by applying a strong enough feedback using  $\bar{u}$ . But a strong feedback is often undesirable since using a stronger control signal means that the system will use more energy. Furthermore, a large control signal is not always possible since  $u$  might be saturated.

Additional problems were encountered during tuning of the attitude PID using feedback linearisation. The core of the problem was that regardless of the strength of the feedback the ROV never stabilised around the desired references. The result from an attempt to stabilise the ROV using the attitude controller can be seen in Figure 6.13. This problem might originate from bad magnetometer readings, which fluctuated when the ROV engaged its thrusters. It is suspected that such fluctuations in magnetic field strength may cause rapid changes in the estimated quaternions and angular velocities. This would in turn lead to problems when computing (6.7) in the feedback linearisation.

In general, comparisons of simulated tests and field tests gave mixed results. In some test the field tests outperformed their simulated counterparts. This is probably due to the fact that damping parameters in the simulator were larger than the actual damping values of the system. The larger damping parameters in the simulation caused the simulated ROV to be more heavily damped, which decreased the magnitude of overshoots and but increased the steady-state errors. The larger damping parameters may also explain the larger phase shift in the simulations of sine-signal tests.

Overall, results were good in the view of the short amount of time available for controller tuning. The problem with the attitude controller and the linearising control law is unfortunate, but should be easily rectified with magnetometer calibration and some additional PID tweaking. Moreover, it is believed that exceptional angular velocity controller performance may be possible with further trimming.



**Figure 6.13:** The result of the feedback linearisation attitude controller with setpoints  $\phi_{ref} = 0$ ,  $\theta_{ref} = 0$  and  $\psi_{ref} = 0$ . Note the instability in  $\theta$  and  $\psi$ .



# 7

---

## Conclusions and Future Work

In this thesis, an attitude model for an ROV has been estimated. The model has then been used in controller synthesis to get satisfactory performance from the controllers. This chapter will discuss and summarise the results from Chapter 3 - Chapter 6. Lastly, ideas for future work and development will be presented.

### 7.1 Conclusions

The estimated model could be used for controller synthesis but the estimation process took longer time than expected. The problem was mainly that model fit in  $p$  was low in all early validation tests. It was not until late in the project that the source of the problem was identified as the effect of thruster 6. Thruster 6 is placed on the keel of the ROV and its actuation was thought to induce both  $p$ - and translational  $y$ -motion. This assumption was, as it would seem, unfortunately not entirely true. After an initial response in  $p$ , either the hydrostatic restoring moment from the ROV's buoyancy or damping from the translational movement caused a drop in  $p$  response. No progress was made in determining which of the aforementioned ideas was the source of the issue, but model fit was increased by ignoring thruster 6 in angular-velocity dynamics.

Out of the three controllers (attitude, angular velocity and depth), depth and angular velocity achieved the most satisfactory performance. These controllers could be used in conjunction with another controller. The rate controller, objectively, performed best but suffered from more steady state-error than desired. Control parameters that gave satisfactory results in simulations were too strong during live tests. This in conjunction with the estimated parameters being too large in the feedback linearisation led to the ROV being unstable during initial

test. After scaling down both control- and model parameters, stability and reference tracking was achieved with the rate controller. Control parameters that stabilised the ROV during live tests were tested in simulations using the estimated model with scaled-down parameter values in the feedback linearisation. This produced unsatisfactory performance, strengthening the argument that though the parameters produced good results in the validation, they were poorly estimated.

Stability could not be reached while using both feedback linearisation and attitude control. The source of the issue was not identified, but is suspected to originate from the conversion between demanded acceleration in the global and local frame or from bad magnetometer performance. Rudimentary performance was reached using the attitude controller by bypassing the feedback linearisation, further underlining the issue with model-parameter scale.

## 7.2 Future Work

To get a better model of the ROV, a way of estimating the ROV's linear velocities could be introduced. This could be done using hydrophones, a sound-emitter and software to estimate the ROV's position and velocity using time difference of arrival (TDOA). Though expensive, a Doppler velocity log could also be used to directly estimate the ROV's velocity compared to the water. With an estimate of linear velocity the complete 6-DOF model could be used which would allow for a better feedback-linearisation control law. A full 6-DOF model could possibly model the effects noted in Figure 5.9, which could lead to increased performance for attitude controllers and allow the ROV to hold its position.

An upgrade to the BlueESCs might be done in order to properly model the relation between thrust and RPM instead of using a look-up table. This would also allow for RPM to be controlled, which could lead to better performance during manoeuvres.

It was also noted that magnetometer measurements were difficult to include in sensor fusion algorithms if the readings from the magnetometer were not normed to a origin centred sphere. Offsets in magnetometer readings would lock the estimated  $\psi$ -angle and ruining any hope of controlling  $\psi$ . Improvements could be done to the sensor fusion module by creating a small program that, using magnetometer samples, returns offsets and scaling values and displays the transformed results so that it can be validated to lie on an origin centred sphere. Improvements could possibly be achieved by temporarily disabling the magnetometer and relying more on the gyroscope during fast manoeuvres and just use the magnetometer to eliminate drift.

# **Appendix**



# A

---

## Dependencies and Installation

The following dependencies are some of the needed packages for compiling and running the ROV.

- Ubuntu 14.04
- ROS Indigo Igloo
- rosserial (Indigo)
- rosserial\_arduino (Indigo)
- image\_common (Indigo)
- image\_transport (Indigo)
- joy (Indigo)

More details of the necessary packages are listed in the installation guide.

### A.1 Raspberry Pi Installation

This section summarises how to setup the Raspberry Pi. It is assumed that the Raspberry Pi is already running Ubuntu 14.04.

#### A.1.1 ROS Installation

To install ROS the ROS source needs to be in the source listing run

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/roslatest.list' && sudo apt-get install ros-indigo-ros-
```

```
base -y && sudo rosdep init && rosdep update && echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc &&
source ~/.bashrc && sudo apt-get upgrade -y && sudo ln
-s /usr /opt/vc
```

to install ROS and source the necessary files.

The packages needed and some other packages used in the ROV is installed by

```
sudo apt-get install libraspberrypi-bin libraspberrypi-dev
openssh-server build-essential avahi-daemon linux-
firmware python-rosinstall ros-indigo-rosserial ros-
indigo-rosserial-arduino ros-indigo-image-common ros-
indigo-image-transport-plugins git && sudo sh -c 'echo
"start_x=1\ngpu_mem=128" >> /boot/config.txt'
```

then restart the Raspberry pi.

### A.1.2 Tether Setup

For communication with the workstation the hostnames and Ethernet port needs to be setup. Configure the host file

```
sudo nano /etc/hosts
```

add

```
10.0.0.20 bluerov
10.0.0.10 workstation
```

to the host file. To use a static IP for the ROV configure the interfaces file

```
sudo nano /etc/network/interfaces
```

add

```
auto eth0
iface eth0 inet static
    address 10.0.0.20
    netmask 255.255.255.0
```

to the interfaces file. Reboot the computer or restart the networking services for changes to take effect.

### A.1.3 Installation of the ROV Package

To download and to compile the package run

```
git clone https://github.com/MrEkelund/ExjobbROV.git && cd  
ExjobbROV/catkin_ws && catkin_make && echo "source ~/  
path-to-package/ExjobbROV/catkin_ws/devel/setup.bash"  
>> ~/.bashrc
```

Setup the communication with the arduino.

```
sudo cp ExjobbROV/catkin_ws/src/bluerov/extra/99-bluerov.  
rules /etc/udev/rules.d/ && sudo udevadm trigger
```

Compile the arduino code and flash it to the arduino by running

```
cd ExjobbROV/catkin_ws && catkin_make  
bluerov_arduino_firmware && catkin_make  
bluerov_arduino_firmware-upload
```

## A.2 Workstation Installation

This section summarises how to setup the workstation for operation with the ROV. It's assumed that the workstation is already running Ubuntu 14.04.

### A.2.1 ROS Installation

To install ROS run the following command

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(  
lsb_release -sc) main" > /etc/apt/sources.list.d/ros-  
latest.list' && sudo apt-get install ros-indigo-ros-  
desktop-full && sudo rosdep init && rosdep update &&  
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc  
&& source ~/.bashrc
```

then run

```
sudo apt-get install python-rosinstall ros-indigo-rosserial  
ros-indigo-rosserial-arduino ros-indigo-image-common  
ros-indigo-image-transport-plugins ros-indigo-joy  
sshpass git
```

to install some necessary packages.

### A.2.2 Tether Setup

For communication with the ROV the hostnames and Ethernet port needs to be setup. Configure the host file

```
sudo nano /etc/hosts
```

add

```
10.0.0.20 bluerov  
10.0.0.10 workstation
```

to the host file. To use a static IP for the workstation configure the interfaces file

```
sudo nano /etc/network/interfaces
```

add

```
auto eth0  
iface eth0 inet static  
    address 10.0.0.10  
    netmask 255.255.255.0
```

to the interfaces file. Reboot the computer or restart the networking services for changes to take effect. The static IP can also be set in the network configuration GUI for Ubuntu.

### A.2.3 Installation of the ROV Package

To download and compile the package run

```
git clone https://github.com/MrEkelund/ExjobbROV.git && cd  
ExjobbROV/catkin_ws && catkin_make
```

# B

---

## Operation of the ROV

This Chapter describes normal operation of the ROV, how to implement new controllers and some tips for troubleshooting.

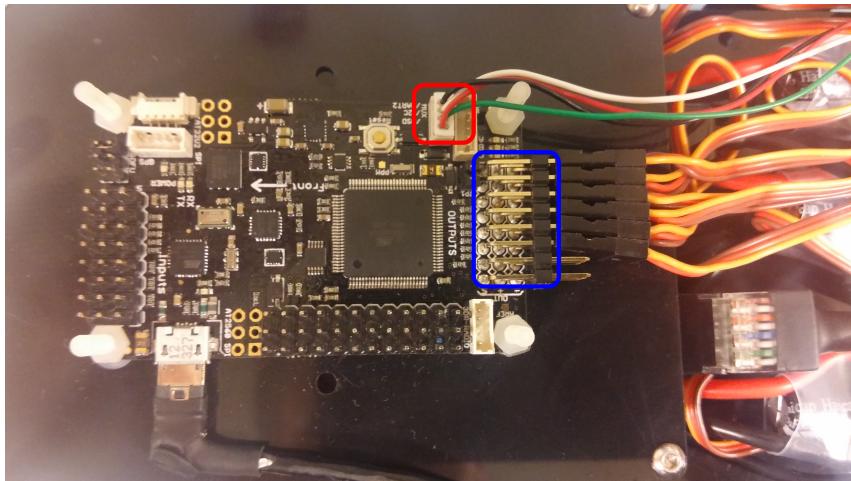
### B.1 Wiring

First connect the cat 6 Ethernet cable to the Raspberry Pi. Then connect the pressure sensor to the I<sup>2</sup>C port on the HKPilot, see Figure B.1. Make sure that ESC  $i$  is connected to the HKPilot's output port  $i$ , see Figure B.1. Connect the Raspberry Pi to the HKPilot by a USB cable. By following the color coded cords, connect the thrusters and the ESCs. Connect the Raspberry Pi with the JST-XH to USB converter. Finally, connect the battery with the JST-XH to USB converter and to the ESCs.

### B.2 Start up of the ROV

For starting the ROV connect all cables according to Section B.1.

- Make sure that the battery is tightly fastened and fully charged.
- Slide the cradle gently into the ROV tube.
- If needed apply some silicone grease to the O-rings of the end cap. Then slide the end cap into the ROV tube.
- Insert the vent bolt in to the vent nut on the end cap.
- Insert the cat 6 cable to the workstation.



**Figure B.1:** The  $I^2C$  port where the pressure sensor ought to be connected is circled in red. Note that the H KPilot has to be soldered for enabling the red circled  $I^2C$  port. The H KPilot's output ports, were the ESC ought to be connected, are circled in blue.

Navigate to the catkin\_ws folder on the workstation and run the start script in a terminal for the ROV

```
./startrov.sh
```

Then in a new terminal navigate to the catkin\_ws folder and run the workstation script

```
./startworkstation.sh
```

### B.3 Shutdown of the ROV

To shut down the ROV run the following command

```
./shutdownrasp.sh
```

otherwise can the micro SD card in the Raspberry Pi take damage.

### B.4 Operating the ROV

The ROV can be controlled via the GUI and by an Xbox controller. Three different controllers are implemented in the ROV. Choose which controller that is enabled by Dynamic Reconfigure → controller → controllers and choose wanted

controller. To chose if Xbox or GUI is enabled do Dynamic Reconfigure → controller → Xbox and chose either GUI or Xbox. The exact linearisation can be enabled and disabled by Dynamic Reconfigure → controller → lin\_active. Note that the ROV has to be armed in either mode for the thruster to work.

### B.4.1 Xbox Mode

In manual mode is the ROV controlled via the Xbox controller. Three different controllers can be used, the angular velocity controller depth controller and the open-loop controller. Table B.1 summarises the Xbox controllers default configuration.

**Table B.1:** The Xbox controllers default configuration. Note that X and Y button only has any effect when the depth controller is enabled.

Button/Stick	Description
Left trigger	Increase depth.
Right trigger	Decrease depth.
Left bumper	Roll negative.
Right bumper	Roll positive.
Left stick	Translational velocities.
Right stick	Angular velocities.
A button	Arm the ROV.
B button	Disarm the ROV.
X button	Decrease depth reference (depth controller).
Y button	Increase depth reference (depth controller).

### B.4.2 GUI Mode

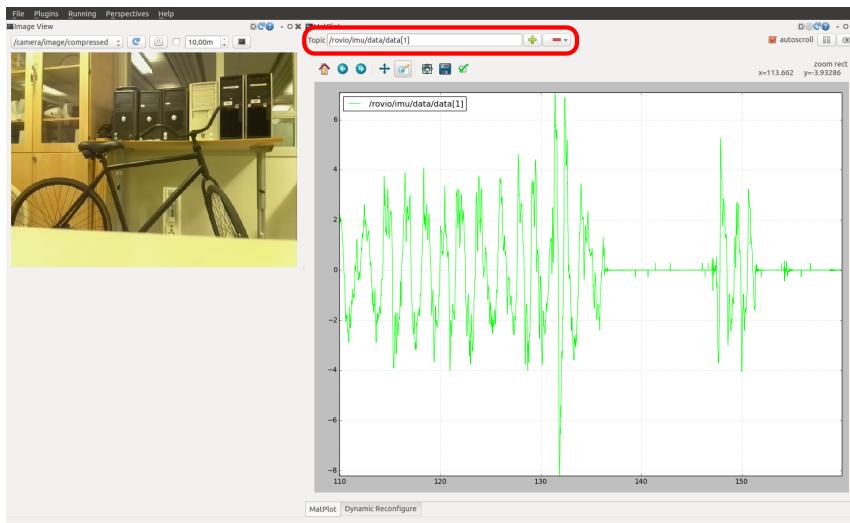
In GUI mode is the ROV's attitude, angular velocities and depth controlled by the GUI. However, the translational velocities are controlled from the Xbox controller. To send reference signals, enable the wanted attitude or angular velocities. Then choose the wanted reference signals, check and uncheck start\_reference\_signals.

### B.4.3 Logging Data

For logging data rqt-bag or the supplied script can be used. In rqt on the work-station start rqt-bag by Plugins → Logging → Bag. Start the recording of data by pressing the red circle. A menu of available topics is showed, select the topics that you want to record. Then give the logfile a name in the pop-up. To stop recording data press the red circle and close Bag by Running → Close:Bag. To use the supplied logging scripts type

```
./test.sh
```

in a terminal window and follow the instructions.



**Figure B.2:** The topic field where the specification of what data should be plotted is circled in red. The green plus adds data to be plotted and the red minus removes data from the graph.

#### B.4.4 Displaying the Continuous Plots

Displaying plots in rqt is enabled by Plugins → Visualization → Plot. In the plot plugin, data can be plotted by choosing topics in the topic field this is shown in Figure B.2. The active topics can be seen in the command line by running

```
rostopic list
```

For multi-array messages an extra \data field has to be added to the topic. Data is then accessed by indexing with hard brackets.

##### Example B.1

Plot the data with index 1 in the multi-array message on topics /rovio imu/data. Type /rovio imu/data/data[1] in the into the topic field and press the green plus.

#### B.4.5 LED Lights on the HKPilot

There are several different LED lights on the HKPilot Mega 2.7, Table B.2 summarizes some of LED statuses

### B.5 New Parameter Estimation and New Controllers

If a new parameter estimation is done or if a new controller is synthesised follow this section.

**Table B.2:** Some of the LED statuses used by the HKPilot Mega 2.7.

Light	Description
Red	The HKPilot is setting up sensors and a connection to ROS. The red light won't turn off until the connection to ROS has been established.
Pulsating blue	Pulsating blue light indicate that the HKPilot has connection with the workstation.
Yellow	Shows that the ROV is disarmed.
Red, yellow and blue	Shows that the ROV is calibrating a sensor.

### B.5.1 New Parameter Estimation

When a new parameter estimation is done the exact linearisation has to be updated. The model parameters that ought to be updated is in controller.h. If the simulator shall be used, ought the EstimatedParams.mat file be updated.

### B.5.2 New Controllers

To implement a new controller edit the controller source files controller.cpp and controller.h. For easy online trimming use dynamic reconfigure parameters in the controller. Modify the controller.config file so that all wanted parameters can be modified. The controller.config file that ought to be modified is located in ExjobbROV/catkin\_ws/src/controller/config/controller.config. To implement the new controller in the simulator modify the simulator.slx file located in ExjobbROV/simulink.

## B.6 Known Issues and Troubleshooting

This sections brings up some of the known issues and how to solve them. Some general debugging and troubleshooting is also brought up.

### B.6.1 Calibration of Sensors

If drift or bias is noted in the sensor fusion or in the raw sensor reading it could be due to bad calibration of the sensors. Calibration of the gyroscope is done by

```
rostopic pub --once /rovio/gyro/calibrate_offsets std_msgs/Bool true
```

The calibration of the accelerometer is done by running

```
rostopic pub --once /rovio/accelerometer/calibrate_offsets std_msgs/Bool true
```

and following the instructions that shows up. Calibration of the magnetometer is done by the command

```
rostopic pub --once /rovio/magnetometer/calibrate_offsets
std_msgs/Bool true
```

and following the commands that shows up. For setting the magnetic field in the sensor fusion run

```
./calibratemag.sh
```

## B.6.2 ROS Debugging

Debugging ROS nodes can be done by listening on ROS messages. This is done by

```
rostopic echo /example/topic
```

In the nodes several different log messages can be generated for example by *ROS\_INFO* and *ROS\_DEBUG*.

## B.6.3 Check Ethernet Connection

It is assumed that the setup from Appendix A is done. To check the Ethernet connection from the ROV to the workstation or the other way around do

```
#From the ROV to the workstation
ping -c 10 10.0.0.10
#From the workstation to the ROV
ping -c 10 10.0.0.20
```

If the connection is good the output ought to be something like 64 bytes received from 10.0.0.10... Otherwise the connection has to be checked. Check that the Ethernet cable is connected and that both the ROV and workstation is powered on. Check that both the workstation and the ROV has the correct IP by the command

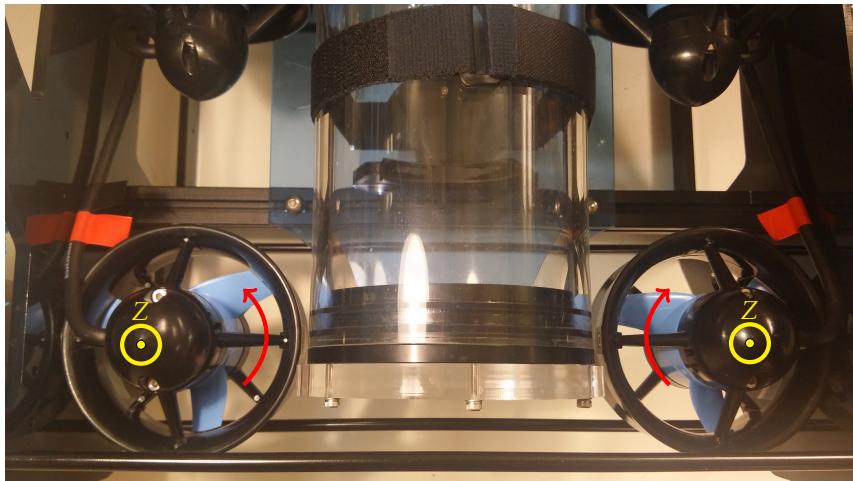
```
ifconfig
```

The output at eth0 ought to contain the correct IP. Otherwise the setup from the Appendix A has to revisited.

## B.6.4 One or Several Thrusters Are Unresponsive

To check for unresponsive thrusters run the following commands

```
rostopic pub --once /rovio/thrusters_enable std_msgs/Bool
true
./thrusterTest.sh
```

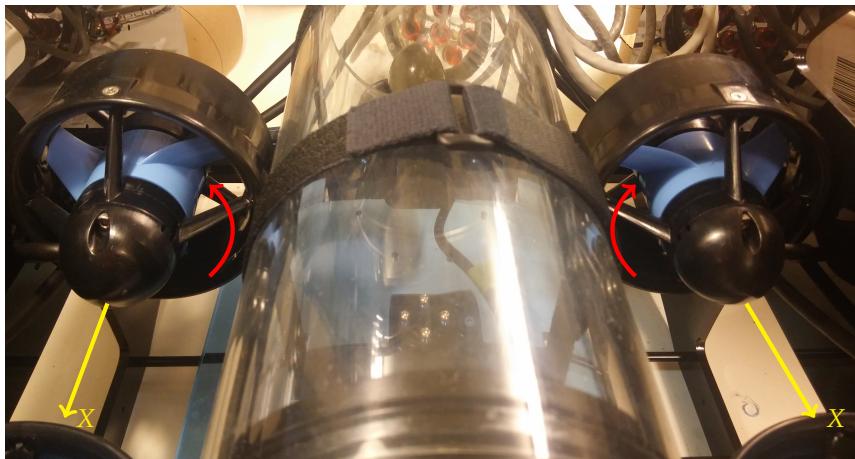


**Figure B.3:** The arrows indicate the direction of rotation of thruster 1 and thruster 2 when supplied with a positive control signal. The resulting positive forces are indicated in yellow.

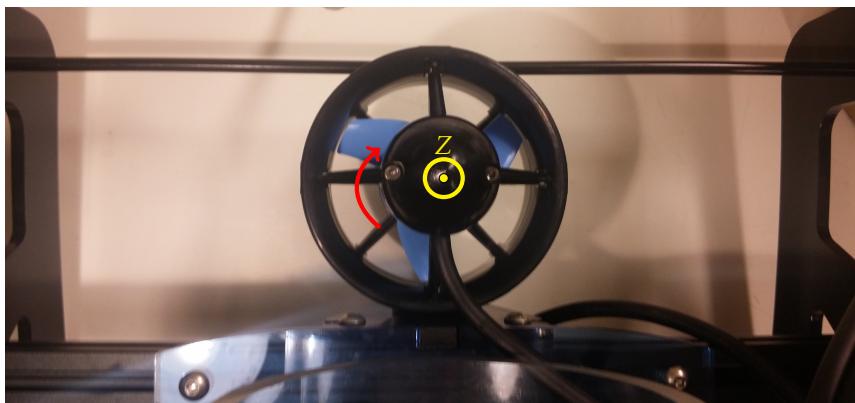
The thruster test will run the thrusters at a minimal torque in incremental order. If one or several thrusters are unresponsive check that the thrusters are connected according to Section B.1. If all thrusters are unresponsive check that the HKPilot Mega 2.7 and the Raspberry has power and connected to each other. Note that if the ROV is disarmed no control signals will be sent to the thrusters.

### B.6.5 Checking and Changing Rotation of Thrusters

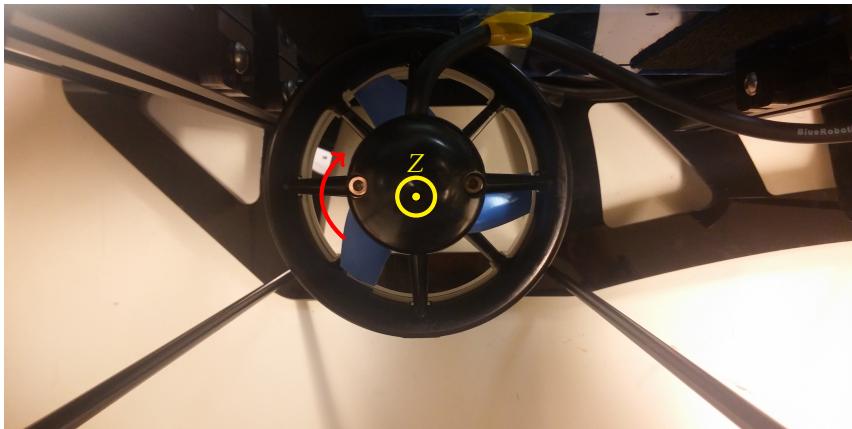
The rotation of thrusters are important due to the moments they create. Thus are two thruster pairs counter-rotating. For checking the rotation of the thrusters the script mentioned in Subsection B.6.4 can be run. Figure B.3 - Figure B.6 shows how the different thrusters should rotate when supplied with a positive control signal. If one or more thrusters are rotating in the wrong direction, change the order of the corresponding thruster cables that are connected.



**Figure B.4:** The arrows indicate the direction of rotation of thruster 3 and thruster 4 when supplied with a positive control signal. The resulting positive forces are indicated in yellow.



**Figure B.5:** The arrows indicate the direction of rotation of thruster 5 when supplied with a positive control signal. The resulting positive force is indicated in yellow.



**Figure B.6:** The arrows indicate the direction of rotation of thruster 6 when supplied with a positive control signal. The resulting positive forces are indicated in yellow.



# C

---

## Thruster Mapping

---

measuredthrust.txt

---

Thrust [N] Amount of thrust [-]

-40.0788	-1.0000
-40.0788	-0.9750
-40.0337	-0.9500
-37.5428	-0.9250
-36.6092	-0.9000
-35.4971	-0.8750
-33.7623	-0.8500
-32.3835	-0.8250
-31.2263	-0.8000
-29.9809	-0.7750
-28.5128	-0.7500
-27.6234	-0.7250
-26.5554	-0.7000
-24.7765	-0.6750
-23.1751	-0.6500
-22.1522	-0.6250
-21.3510	-0.6000
-19.5280	-0.5750
-18.4600	-0.5500
-17.3480	-0.5250
-16.1025	-0.5000
-14.6345	-0.4750
-13.0781	-0.4500
-12.1877	-0.4250
-10.9874	-0.4000
-10.0528	-0.3750
-8.8966	-0.3500
-7.7404	-0.3250
-6.8058	-0.3000
-6.0497	-0.2750

-5.1151	-0.2500
-4.2698	-0.2250
-3.4696	-0.2000
-2.7135	-0.1750
-1.8682	-0.1500
-1.1121	-0.1250
-0.5335	-0.1000
-0.2667	-0.0750
0	-0.0500
0	-0.0250
0	0
0	0.0250
0	0.0500
0.5786	0.0750
1.1121	0.1000
1.8240	0.1250
2.5801	0.1500
3.4255	0.1750
4.0923	0.2000
5.1151	0.2250
6.2272	0.2500
7.0284	0.2750
8.2739	0.3000
9.2526	0.3250
10.8089	0.3500
12.2328	0.3750
13.6558	0.4000
14.9904	0.4250
16.4585	0.4500
17.7039	0.4750
19.2161	0.5000
20.5508	0.5250
21.7521	0.5500
23.6644	0.5750
24.7324	0.6000
26.4221	0.6250
27.7126	0.6500
29.6249	0.6750
31.3156	0.7000
32.8729	0.7250
33.8065	0.7500
34.2517	0.7750
37.0534	0.8000
38.6549	0.8250
39.2776	0.8500
41.1458	0.8750
42.9257	0.9000
44.7938	0.9250
47.1062	0.9500
48.0408	0.9750
49.9531	1.0000

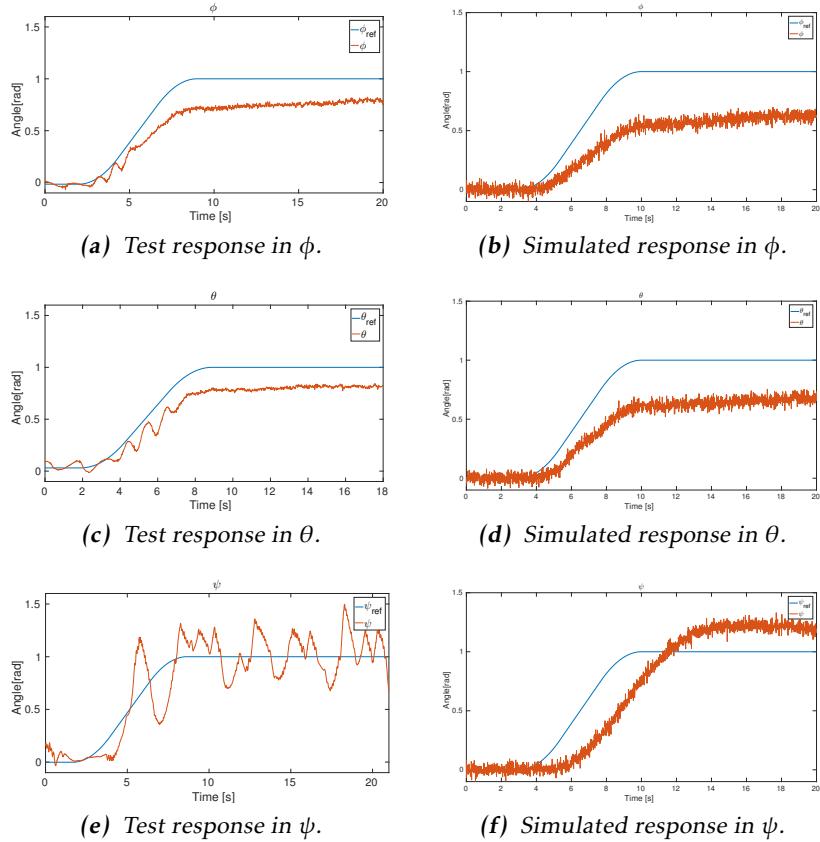
---

# D

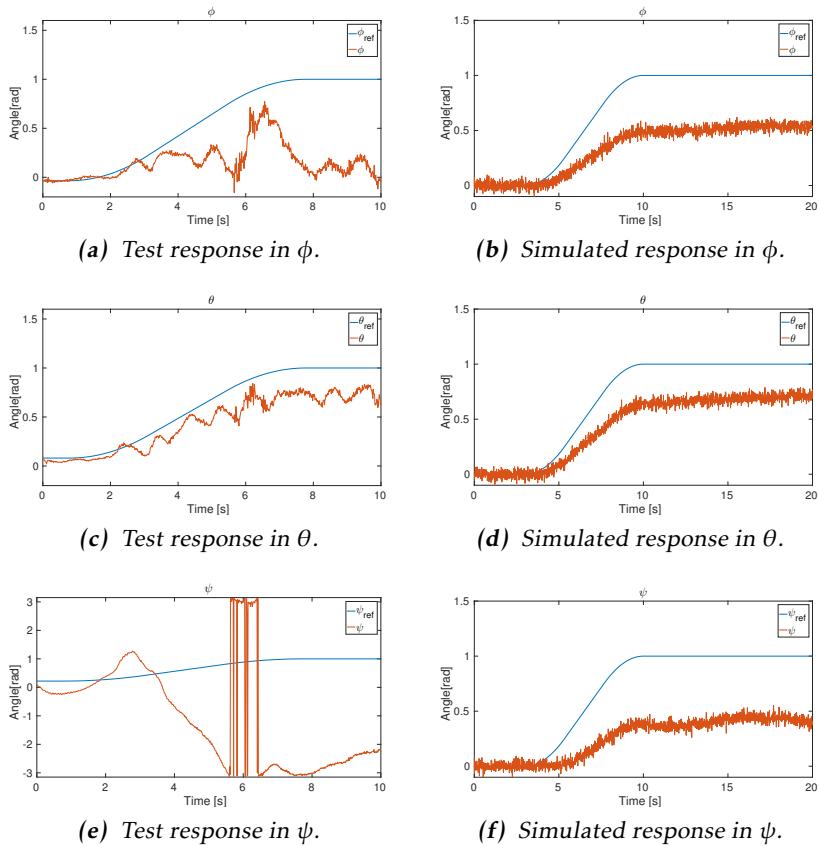
---

## Controller Test Results

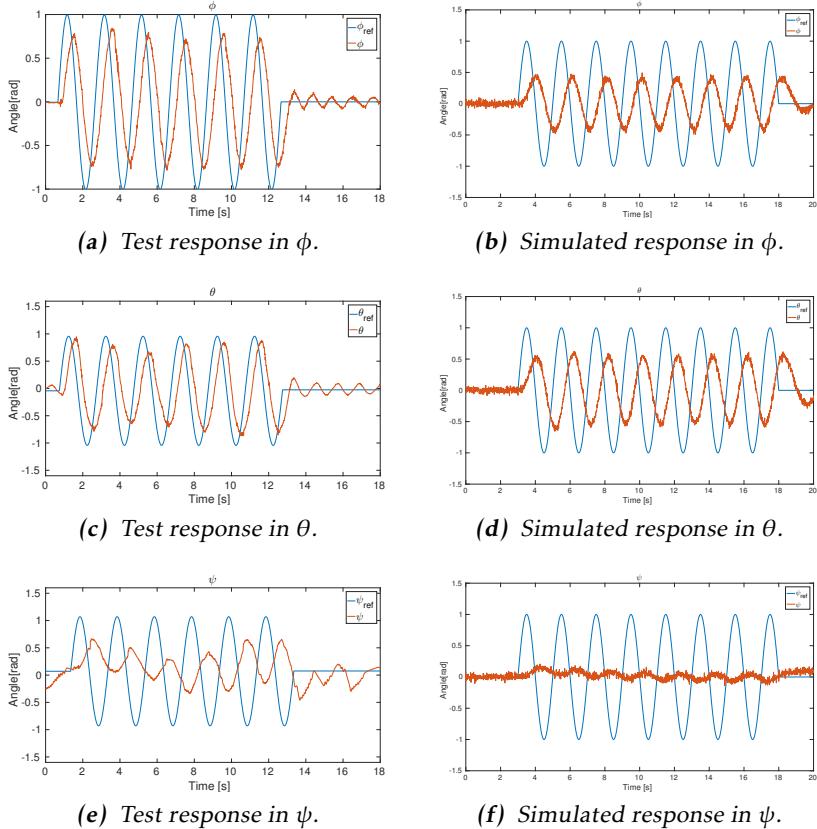
The following Figures are the results from the conducted tests.



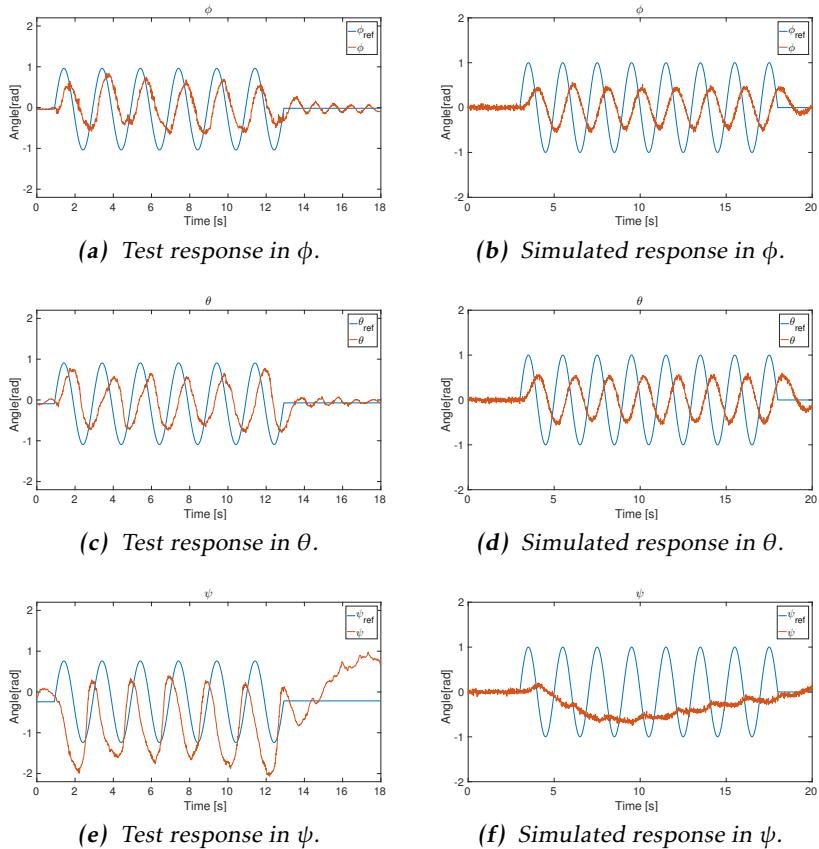
**Figure D.1:** A smooth step with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  was applied in one attitude angle at a time while using the attitude controller. While a smooth step was applied in one attitude angle the other attitude angles were not controlled.



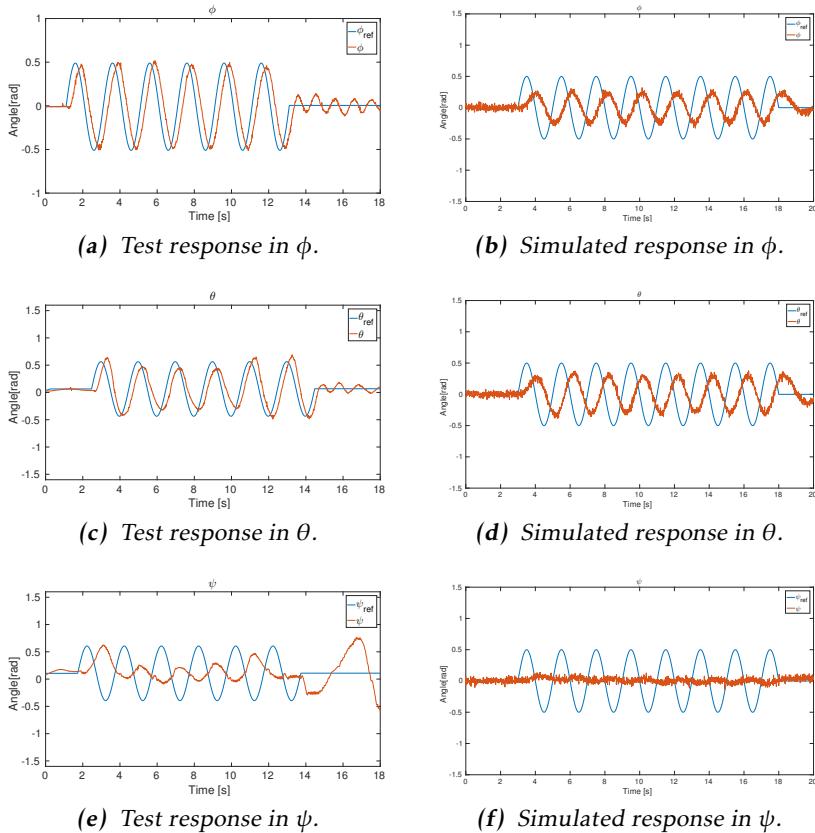
**Figure D.2:** A smooth steps with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  was applied in all attitude angles at the same time while using the attitude controller.



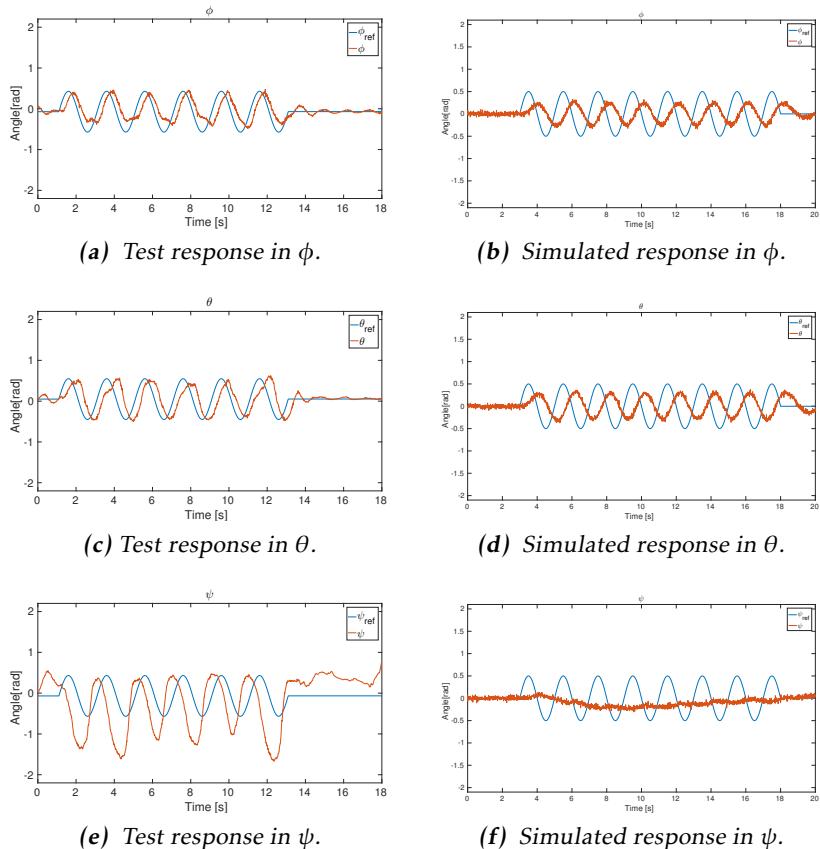
**Figure D.3:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in one attitude angle at a time while using the attitude controller. While a sine signal was applied in one attitude angle the other attitude angles were not controlled.



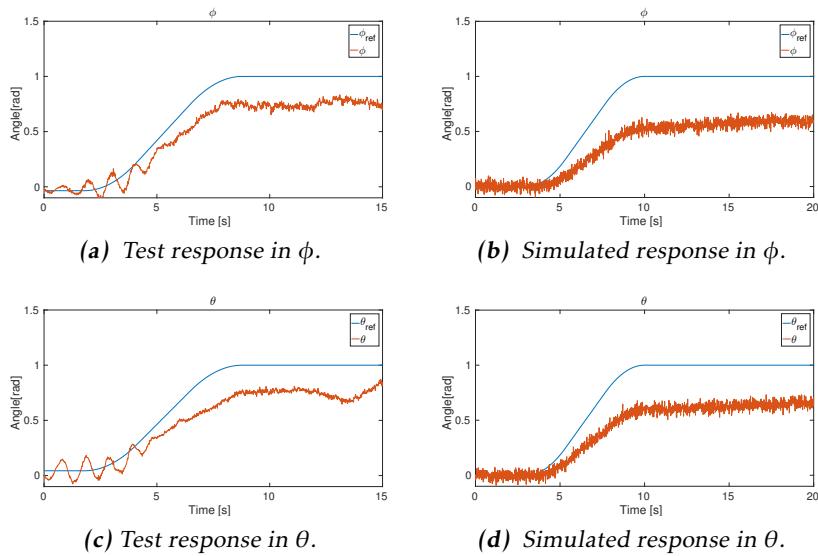
**Figure D.4:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in all attitude angles at the same time while using the attitude controller.



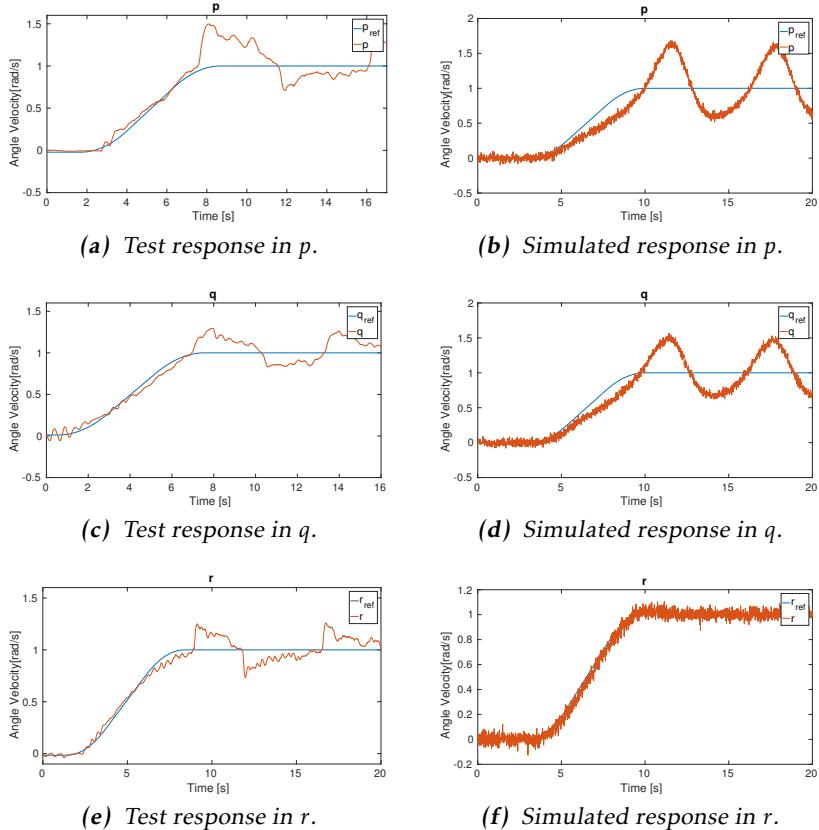
**Figure D.5:** A sine signal with amplitude 0.5 and frequency 0.5 Hz was applied in one attitude angle at a time while using the attitude controller. While a sine signal was applied in one attitude angle the other attitude angles were not controlled.



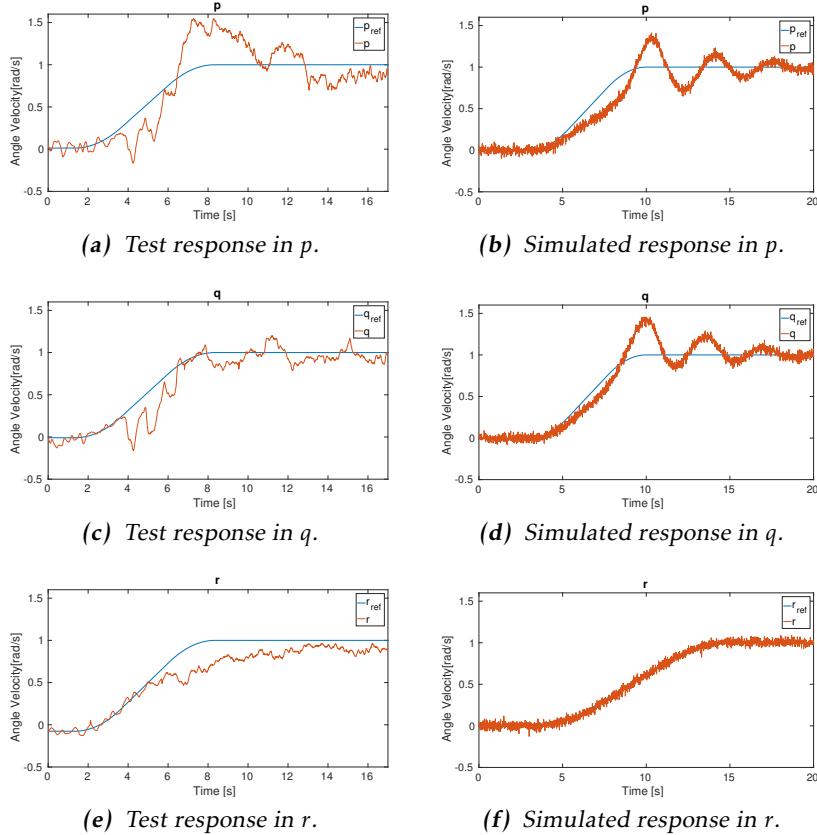
**Figure D.6:** A sine signal with amplitude 0.5 and frequency 0.5 Hz was applied in all attitude angles at the same time while using the attitude controller.



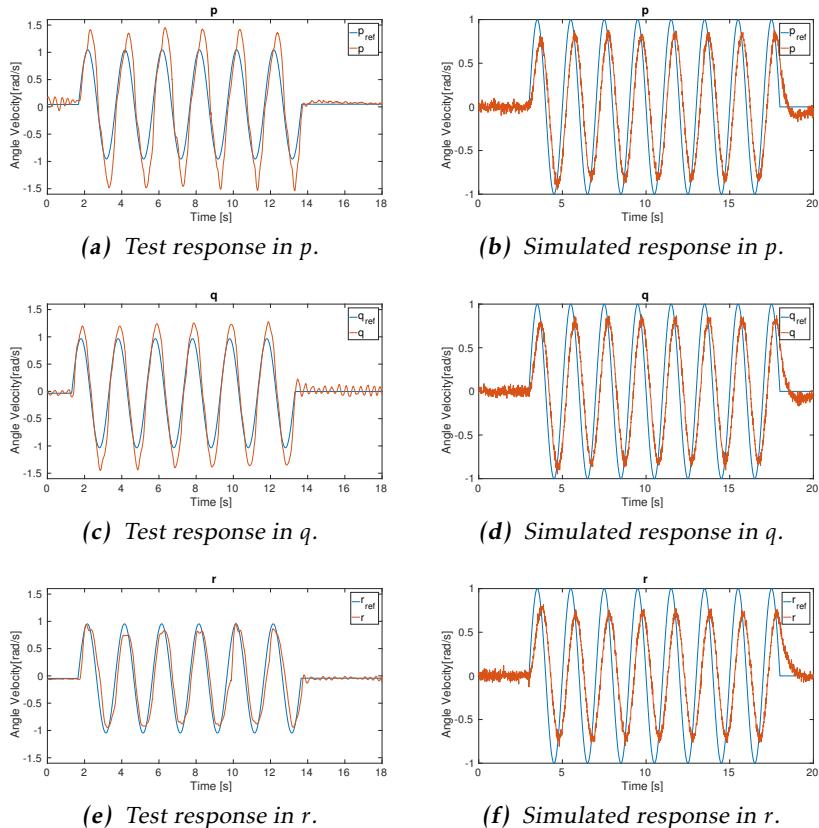
**Figure D.7:** Smooth steps with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  were applied in  $\theta$  and  $\phi$  at the same time while using the attitude controller. The attitude angle  $\psi$  was kept free during the test.



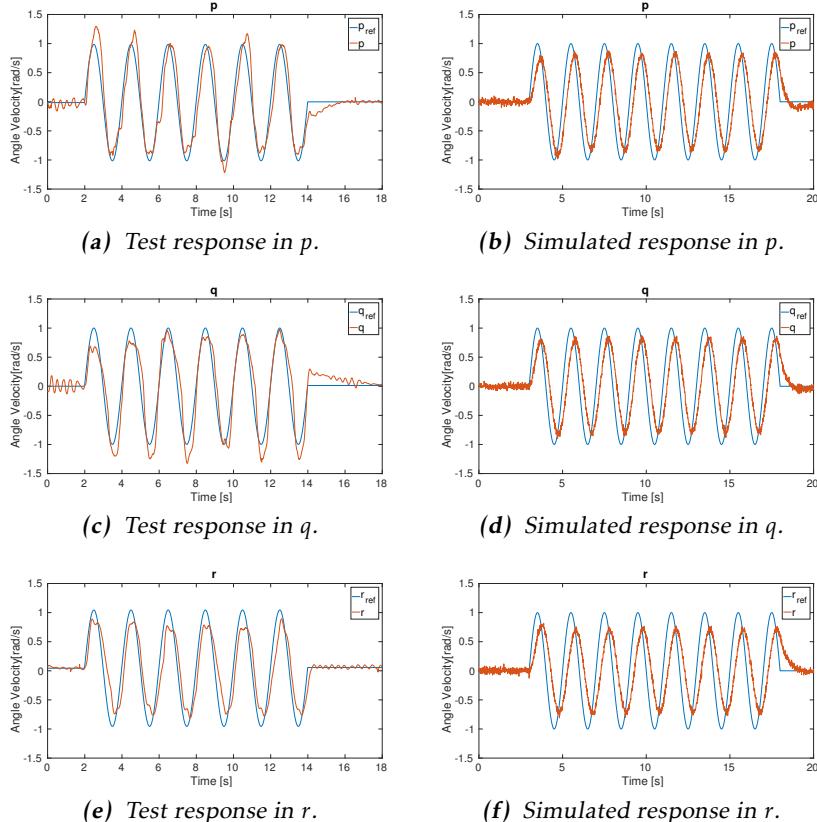
**Figure D.8:** A smooth step with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  was applied in one angular velocity at a time while using the rate controller. While a smooth step was applied in one angular velocity the other angular velocities were controlled with the reference zero.



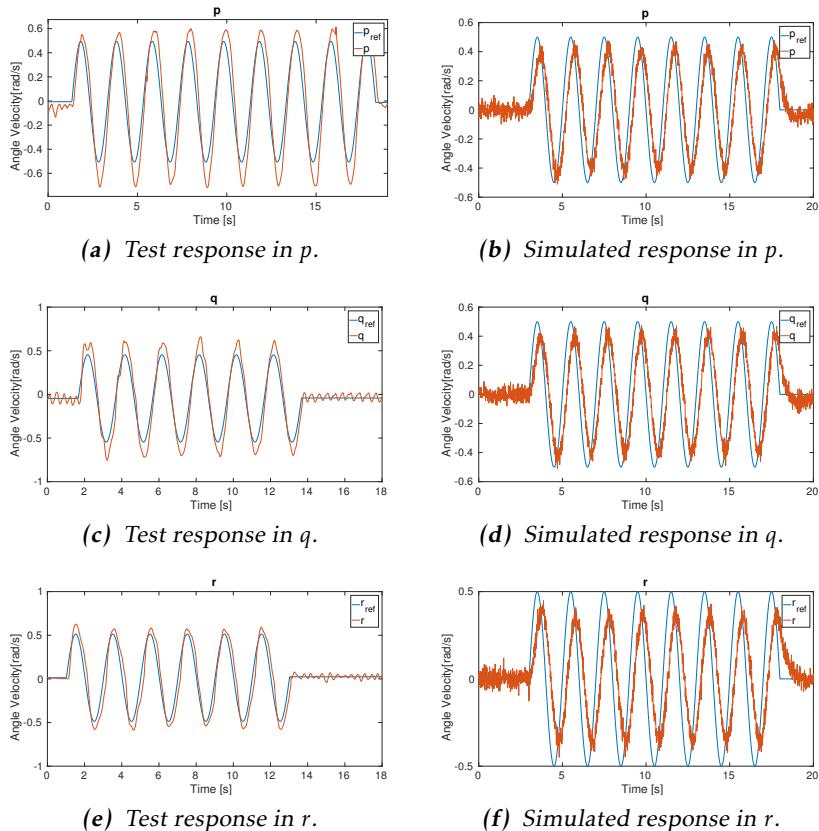
**Figure D.9:** A smooth step with  $q_0 = 0$ ,  $q_f = 1$ ,  $t_s = 3$ ,  $t_f = 15$  and  $V = 1.5(q_f - q_0)/(t_f - t_s)$  was applied in all angular velocities at the same time while using the rate controller.



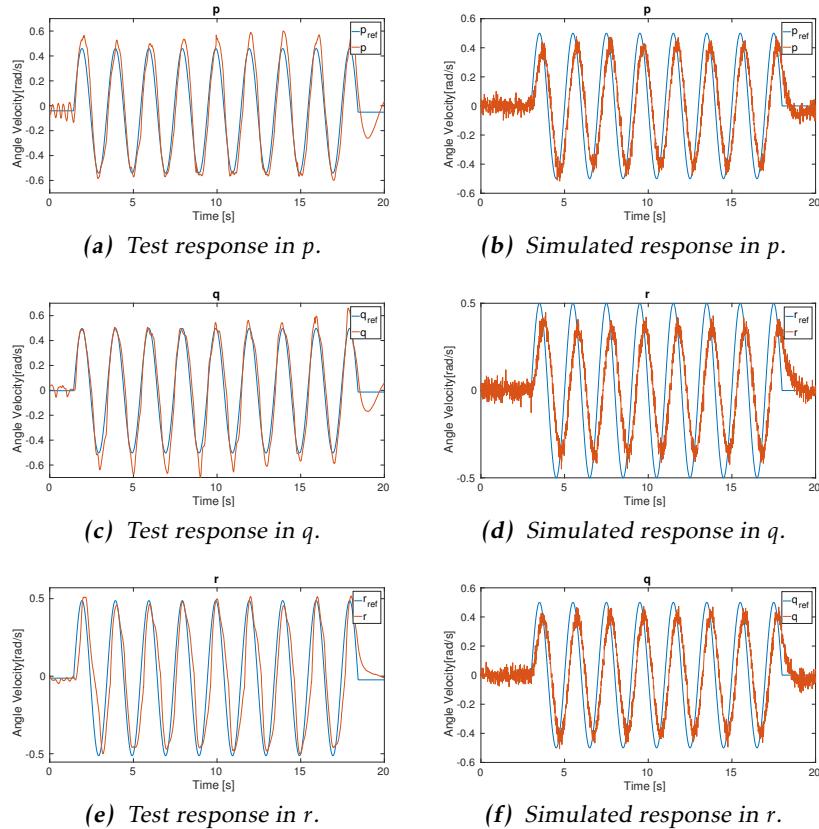
**Figure D.10:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in one angular velocity at a time while using the rate controller. While a sine signal was applied in one angular velocity the other angular velocities were controlled with the reference zero.



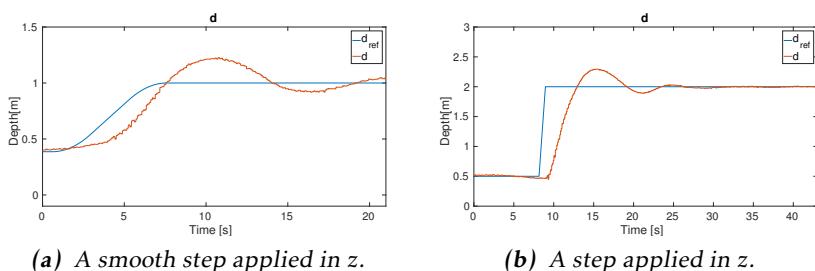
**Figure D.11:** A sine signal with amplitude 1 and frequency 0.5 Hz was applied in all angular velocities at the same time while using the rate controller.



**Figure D.12:** A sine signal with amplitude 0.5 and frequency 0.5 Hz was applied in one angular velocity at a time while using the rate controller. While a sine signal was applied in one angular velocity the other angular velocities were controlled with the reference zero.



**Figure D.13:** Sine signals with amplitude 0.5 and frequency 0.5 Hz were applied in all angular velocities at the same time while using the rate controller.



**Figure D.14:** Steps applied to  $z$ . A smooth step from 0.4 m to 1 m is shown in (a) and a step from 0.5m to 2m is shown in (b).

---

## Bibliography

- BlueRobotics. *BlueRobotics webpage*. Online, 2016. URL <http://www.bluerobotics.com/>. [Accessed: 2016-02-04]. Cited on page 1.
- Craig Cutler. *Drone Skies: The Unmanned Aircraft Revolution Is Coming*. Online, 2013. URL <http://www.popularmechanics.com/military/a9407/drone-skies-the-unmanned-aircraft-revolution-is-coming-15894155/>. [Accessed: 2016-02-02]. Cited on page 1.
- Thor I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. Wiley, 2011. ISBN 9781119994138. Cited on pages iii, 11, 13, 14, 17, 18, 19, 20, 51, 52, and 53.
- Jerzy Garus. Optimization of thrust allocation in the propulsion system of an underwater vehicle. *International Journal of Applied Mathematics and Computer Science*, 14(4):461–467, 2004. Cited on page 50.
- Torkel Glad and Lennart Ljung. *Control Theory : Multivariable and Nonlinear Methods*. Taylor Francis Ltd, 2000, 2003. ISBN 9789144022758. Cited on pages 49, 50, 51, and 55.
- Torkel Glad and Lennart Ljung. *Reglerteknik : Grundläggande teori*. Studentlitteratur AB, 2006, 2012. ISBN 9780748408788. Cited on page 49.
- Fredrik Gustafsson. *Statistical sensor fusion*. Lund : Studentlitteratur, 2012, 2012. ISBN 9789144077321. Cited on pages 14, 25, 26, 27, and 41.
- Roger Larsson. *System Identification of Flight Mechanical Characteristics*. PhD thesis, Linköping University, 2013. Cited on pages 35 and 41.
- Lennart Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999. ISBN 0138816409. Cited on page 35.
- Lennart Ljung and Torkel Glad. *Modellbygge och simulering*. Lund : Studentlitteratur, 2004, 2004. ISBN 9144024436. Cited on pages 2, 26, and 32.

- OpenROV. *OpenROV webpage*. Online, 2016. URL <http://www.openrov.com/>. [Accessed: 2016-02-04]. Cited on page 1.
- L. Podofillini, B. Sudret, B. Stojadinovic, E. Zio, and W. Kröger. *Safety and Reliability of Complex Engineered Systems: ESREL 2015*. CRC Press, 2015. ISBN 9781315648415. Cited on page 2.
- ROS. *ROS wiki page*. Online, 2016. URL <http://wiki.ros.org/>. [Accessed: 2016-05-17]. Cited on page 5.
- SAAB. *SAAB underwater systems webpage*. Online, 2016. URL <http://saab.com/naval/underwater-systems/remotely-operated-vehicles/>. [Accessed: 2016-02-04]. Cited on page 1.
- SNAME. The society of naval architects and marine engineers. nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin*, 5(1), 1950. Cited on page 12.
- Mark W. Spong, Seth Hutchinson, and Mathukumalli Vidyasagar. *Practical OpenCV. [Elektronisk resurs]*. John Wiley & Sons, 2006. ISBN 9780471649908. Cited on page 55.
- David Törnqvist. *Statistical Fault Detection with Applications to IMU Disturbances*. PhD thesis, Linköping University, 2006. Cited on pages 26 and 27.
- Jonas Wallin and Joakim Zachrisson. Sensor fusion in smartphones : with application to car racing performance analysis. Master's thesis, Linköping University, Automatic Control, The Institute of Technology, 2013. Cited on page 38.

---

# Index

- Actuators, 20  
Angular Velocities Controller, 52  
Attitude Controller, 51  
body-fixed coordinate system, 13  
buoyancy, 19  
CG abbreviation, xi  
CG abbreviation, xi  
CO abbreviation, xi  
Controller, 49  
Coriolis, 18, 19  
Coriolis matrix, 11  
Cost function, 33  
Damping, 19  
Damping matrix, 11  
Data preprocessing, 32  
Depth Controller, 53  
DOF abbreviation, xi  
EKF abbreviation, xi, 39  
ESC abbreviation, xi, 7  
Exact Linearisation, 49, 51, 52  
Feedback Control, 49  
Fit, 32, 35, 38, 42  
global coordinate system, 13  
Hydrodynamics, 18  
Hydrostatics, 19  
I/O abbreviation, xi  
IMU abbreviation, xi  
Inertia, 17  
Inertia matrix, 11  
KF abbreviation, xi  
Kinematics, 13, 14, 17  
Look-up table, 20, 51  
Model of an underwater vehicle, 11  
Model structure, 22, 35, 37–39  
Motion model, 26  
MPC abbreviation, xi  
NED, 13  
Open-Loop Control, 49, 50  
Parameter Estimation, 31  
PI abbreviation, xi, 52, 53  
PID abbreviation, xi, 51  
Prediction-Error Method, 33  
Quaternions, 14, 19, 26, 38

- Restoring forces, 19
- Restoring forces matrix, 11
- Rigid-Body Kinetics, 17
- ROS
  - abbreviation, xi, 5, 7, 9
- ROV
  - abbreviation, xi
- RPM
  - abbreviation, xi
- SLAM
  - abbreviation, xi
- SNR
  - abbreviation, xi
- TDOA
  - abbreviation, xi
- thrust allocation, 50
- thruster geometry, 20, 50
- Transformation matrices, 14
- UV
  - abbreviation, xi
- Viscous damping, 19