Automatic Tuning of Motion Control System for an Autonomous Underwater Vehicle

Markus Andersson
Master of Science Thesis in Electrical Engineering

Automatic Tuning of Motion Control System for an Autonomous Underwater Vehicle:

Markus Andersson

LiTH-ISY-EX--19/5249--SE

Supervisor: Fredrik Ljungberg

ISY, Linköpings universitet

Anders Peterson

Saab Dynamics

Examiner: Martin Enqvist

ISY, Linköpings universitet

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

Copyright © 2019 Markus Andersson
Abstract

The interest for marine research and exploration has increased rapidly during the past decades and autonomous underwater vehicles (AUV) have been found useful in an increased amount of applications. The demand for versatile platform AUVs, able to perform a wide range of tasks, has become apparent. A vital part of an AUV is its motion control system, and an emerging problem for multipurpose AUVs is that the control performance is affected when the vehicle is configured with different payloads for each mission. Instead of having to manually re-tune the control system between missions, a method for automatic tuning of the control system has been developed in this master’s thesis.

A model-based approach was implemented, where the current vehicle dynamics are identified by performing a sequence of excitation maneuvers, generating informative data. The data is used to estimate model parameters in predetermined model structures, and model-based control design is then used to determine an appropriate tuning of the control system.

The performance and potential of the suggested approach were evaluated in simulation examples which show that improved control can be obtained by using the developed auto-tuning method. The results are considered to be sufficiently promising to justify implementation and further testing on a real AUV.

The automatic tuning process is performed prior to a mission and is meant to compensate for dynamic changes introduced between separate missions. However, the AUV dynamics might also change during a mission which requires an adaptive control system. By using the developed automatic tuning process as foundation, the first steps towards an indirect adaptive control approach have been suggested.

Also, the AUV which was studied in the thesis composed another interesting control problem by being overactuated in yaw control, because yawing could be achieved by using rudders but also by differential drive of the propellers. As an additional and separate part of the thesis, an approach for using both techniques simultaneously have been proposed.
Acknowledgments

I would like to direct my deepest gratitude to my supervisors, Fredrik Ljungberg at Linköping University for his guidance, support and insightful discussions, and Anders Peterson at Saab Dynamics for his input and feedback. A special thank you is directed to Nicklas Johansson at Saab Dynamics for providing me with the simulation model and the means to comprehend and modify it.

I also want to thank the kind and helpful people involved with developing Mari-bot LoLo at KTH Royal Institute of Technology for welcoming me into their team and aiding me whenever needed.

Lastly, I want to direct my gratitude to my examiner Martin Enqvist at Linköping University, Torbjörn Crona at Saab Dynamics and Jakob Kuttenkeuler at KTH for giving me the opportunity to write this thesis.

Linköping, June 2019

MA
# Contents

## Notation ix

## 1 Introduction 1

1.1 Background ................................................. 2
1.2 Purpose .................................................... 2
1.3 Objectives .................................................. 3
1.4 Method ...................................................... 3
1.5 Related Work ............................................... 4
1.6 Limitations .................................................. 5
1.7 Thesis Outline ............................................. 5

## 2 Theory and Preliminaries 7

2.1 Vehicle Description ........................................ 7
  2.1.1 AUV Design ........................................... 7
2.2 Simulation Environment .................................... 10
  2.2.1 Model Description ................................... 10
2.3 Dynamic Model ............................................. 11
  2.3.1 Kinematics ............................................. 12
  2.3.2 Rigid-body Kinetics .................................. 14
  2.3.3 Hydrodynamics ........................................ 15
  2.3.4 Hydrostatics .......................................... 16
  2.3.5 Actuators .............................................. 18
  2.3.6 Motion Model of AUV ................................ 19
2.4 Parameter Estimation ...................................... 20
  2.4.1 Experiments & Data Collection ......................... 21
  2.4.2 Least Squares Estimation ............................... 22
  2.4.3 Estimation Using Instrumental Variables .............. 22
  2.4.4 Recursive Parameter Estimation ....................... 23
2.5 Motion Control ............................................. 24
  2.5.1 Control Problem ....................................... 25
  2.5.2 Pole Placement ........................................ 25
  2.5.3 Cascade Control ...................................... 27
  2.5.4 Feedback Linearization ............................... 28
# Contents

## 3 System Identification & Control Design

3.1 System Identification ........................................... 29  
3.1.1 Actuators .................................................. 30  
3.1.2 Nonlinear Models ........................................... 31  
3.1.3 Linear Models .............................................. 34  
3.1.4 Estimation Using Instrumental Variables .................. 39  
3.2 Control System .................................................. 40  
3.2.1 Speed Controller ............................................ 41  
3.2.2 Depth Controller ............................................ 43  
3.2.3 Heading Controller ......................................... 46  
3.2.4 Differential Drive .......................................... 48  
3.3 Automatic Tuning Process ..................................... 50  
3.4 Adaptive Control ............................................... 52  
3.4.1 Indirect Adaptive Control Approach ....................... 53

## 4 Results

4.1 Motion Control .................................................. 56  
4.1.1 Speed Control ............................................... 56  
4.1.2 Depth Control ............................................... 57  
4.1.3 Heading Control ............................................. 60  
4.2 Automatic Tuning ............................................... 61  
4.2.1 Case 1 - Change in Mass .................................. 62  
4.2.2 Case 2 - Change in Mass and No Propeller Saturation .... 67  
4.2.3 Case 3 - Change in Elevator Lift .......................... 68  
4.2.4 Case 4 - Change in Rudder Lift ............................ 69  
4.3 Differential Drive .............................................. 71  
4.4 Online Identification .......................................... 73

## 5 Discussion and Conclusion

5.1 Discussion ...................................................... 77  
5.2 Conclusions .................................................... 80  
5.3 Future Work .................................................... 81

## Bibliography

83
## Notation

### Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUV</td>
<td>Autonomous underwater vehicle</td>
</tr>
<tr>
<td>CB</td>
<td>Centre of buoyancy</td>
</tr>
<tr>
<td>CG</td>
<td>Centre of gravity</td>
</tr>
<tr>
<td>CO</td>
<td>Centre of origin</td>
</tr>
<tr>
<td>DOF</td>
<td>Degree of freedom</td>
</tr>
<tr>
<td>IV</td>
<td>Instrumental variables</td>
</tr>
<tr>
<td>LS</td>
<td>Least squares</td>
</tr>
<tr>
<td>NED</td>
<td>North, east, down (coordinate system)</td>
</tr>
<tr>
<td>PID</td>
<td>Proportional, integral, differential (regulator)</td>
</tr>
<tr>
<td>ROV</td>
<td>Remotely operated (underwater) vehicle</td>
</tr>
<tr>
<td>SISO</td>
<td>Single-input, single-output</td>
</tr>
<tr>
<td>SMaRC</td>
<td>Swedish Maritime Robotics Centre</td>
</tr>
<tr>
<td>UUV</td>
<td>Unmanned underwater vehicle</td>
</tr>
<tr>
<td>VBS</td>
<td>Variable buoyancy system</td>
</tr>
</tbody>
</table>
About two-thirds of the earth is covered by ocean, and most of it is still unexplored. It is believed that our oceans may unveil solutions for many of the issues faced by mankind today, such as sustainable production of energy, food and raw materials, and addressing climate change (SSF, 2008). Due to the hazardous and challenging conditions of sub sea operation, the need for and use of underwater robotics is apparent.

Most unmanned underwater vehicles (UUVs) commercially available are remotely operated via a tether, referred to as remotely operated vehicles (ROVs). Hence, the usage of ROVs is currently limited by their dependency of human operators and mother vessels. The demand for marine robotics able to operate with minimum human interaction has resulted in an increased development of autonomous underwater vehicles (AUVs) during the last decades (Yuh, 2000). AUVs have a wide range of applications, such as environmental monitoring, oceanography, military, ocean mining and inspection of ship hulls and naval structures (Yuh et al., 2011).

A central part of an AUV, is its motion control system, hereon referred to as the control system. The control system has the responsibility of maneuvering the vehicle in order to fulfill certain control objectives, such as keeping desired speed or course, or following a specified trajectory. Control of underwater vehicles, in general, is difficult due to highly nonlinear, coupled dynamics and environmental disturbances such as ocean currents and waves. The control problem becomes even more difficult for AUVs which are equipped with varying payloads for different missions, since these alterations will affect the vehicle dynamics and thereby the maneuverability. This master’s thesis aims to develop and evaluate a control system able to adjust to such changes in vehicle configuration.

In this first chapter, the background and purpose of the thesis will be presented,
along with project objectives and a brief description of the methodology. Also, an overview of related work will be given in order to set the thesis in context, and to justify some of the exclusions made. Lastly, the structure and content of the thesis is outlined.

This master's thesis was written at Saab Dynamics in Linköping, Sweden, in collaboration with Linköping University and KTH Royal Institute of Technology.

1.1 Background

The Swedish Maritime Robotics Centre (sMaRC) is a research centre founded in 2017 dedicated to the development of the next generation of marine robotics. sMaRC's cross-disciplinary projects require a platform vehicle which can be used by researchers from various fields for tasks such as data collection, testing and demonstration of new technologies. This requires a versatile vehicle able to perform a wide range of missions with varying configurations and payload. This resulted in the design and construction of the AUV Maribot LoLo, developed at the Centre for Naval Architecture at KTH Royal Institute of Technology (Deutsch et al., 2018).

One of many challenges when developing a platform AUV like Maribot LoLo, is the design of its control system, which must provide good maneuverability despite changes in vehicle dynamics. One way to remedy this is to readjust the control system prior to each mission, which would be a time consuming process to do manually. Manual tuning would also require that the user of the AUV has prior knowledge in control theory, which diminishes the vehicle's usability. For these reasons, developers of Maribot LoLo have deemed it worthwhile to develop functionality to automatically tune the control system.

There are different approaches to tuning a control system, which mainly depend on which control algorithm is used. In general, a mathematical model of the system dynamics is required to perform methodical control design, often referred to as model-based control design. A sensible approach for an automatic tuning process will therefore consist of firstly determining a model of the AUV dynamics, and then using the model to set the design parameters in the control system.

1.2 Purpose

The purpose of this master's thesis is to develop and evaluate a method of automatically tuning the design parameters of a control system for an autonomous underwater vehicle.

This automated tuning process will be performed in water, prior to a mission, and aims to contribute to the versatility of the AUV by ensuring proper maneuverability regardless of payload. Also, by not having to tune the control system manually, lead time between missions can be reduced significantly, allowing for a more efficient usage of the vehicle.
A secondary purpose is to evaluate whether improved performance can be obtained by further tuning the control system during the mission, and in that case if there are any reliable methods for doing so.

1.3 Objectives

In order to fulfill the purpose of the thesis, the approach presented in Section 1.1 will be implemented. Using this approach, the automatic tuning process will typically consist of three parts; the AUV performs a set of predetermined experiments while collecting data from actuators and sensors, a model of the vehicle dynamics is estimated based on the collected data and lastly the estimated model is used to set the design parameters in the control system.

These three stages can be considered as separate sub-objectives. Modelling of the AUV dynamics can be divided into two distinct parts; determining a suitable model structure and a method for estimating the unknown model parameters. Likewise, the control design will consist of choosing an appropriate control algorithm and a method for selecting the control parameters.

Furthermore, an evaluation of the developed controller is required in order to determine whether the purpose of the thesis has been achieved or not. The objectives of the thesis are summarized below.

- How should the AUV be modelled?
  - What model structure should be used?
  - What estimation method should be used for the unknown model parameters?
- What type of experiments should the AUV perform in order to obtain informative data sets?
- How should the AUV be controlled?
  - What control algorithms should be used?
  - Given a model, how should the control parameters be chosen?
- How robust is the control system generated by the automatic tuning?
- Can improved performance be achieved by updating the control parameters online, i.e. during missions?

1.4 Method

First a literature study of related work was performed in order to obtain general knowledge in the field, and to identify the current state of research related to the objectives formulated in Section 1.3.
Secondly, a simulation model of the AUV had to be established, since testing on the physical vehicle was not possible throughout the course of the project. In general, having a simulation model during the early stages of control design is advantageous due to the possibility of performing tests quickly and thereby saving time.

The development of the automated tuning process was carried out in iterations, where in the initial iterations only simple techniques and methods were employed in order to identify critical aspects in all stages of the process, at an early stage. In the following iterations, more advanced methods could progressively be implemented at the stages deemed to yield the biggest improvement to the whole process.

1.5 Related Work

During the last decade, several studies of similar applications have been published. In Eng et al. (2016) an AUV is modelled in real-time by performing predetermined experiments. Their vehicle is similar to Maribot LoLo in terms of hull shape and actuation, both having streamlined, torpedo-shaped hulls, and being controlled using a combination of thrusters and rudders. However, the differences in actuator configuration and vehicle size distinguishes this thesis from their studies.

Eng et al. (2016) used recursive least squares (RLS) for real-time system identification. RLS has also been used in other studies of real-time system identification of UUVs (Karras et al., 2019, Weiss and Du Toit, 2013). However, the two vehicles considered in these studies are actuated solely using thrusters, and do not share the streamline shape of Maribot LoLo. Another method which has been used for real-time system identification of UUVs is incremental support vector regression (SVR) (Wehbe et al., 2017). The vehicle studied there was also only actuated using thrusters.

In all of the studies mentioned above, the modelling framework of Fossen (1994) has been used. Fossen has contributed with acknowledged theory in modelling and control of marine vehicles, including underwater vehicles. He has presented a general model structure for marine vehicles along with theory of how to adapt it for underwater vehicles (Fossen, 1994, 2011). This model structure is unequivocally the most commonly occurring in the literature.

In addition to the studies where system identification of AUVs is performed online, there is also a considerable amount of literature where the identification is carried out offline, i.e. the data collection and system identification is performed at separate occasions. In a previous master’s thesis at Linköping University, system identification of a ROV was evaluated, comparing a Prediction Error Method (PEM) and Extended Kalman Filter (EKF) estimation (Aili and Ekelund, 2016). Variations of EKF estimation has been suggested in several other studies (Luque et al., 2009, Sabet et al., 2018, Sabet et al., 2014).
In a survey of design and control of AUVs, a compilation of proposed control strategies is given (Yuh, 2000). The survey mentions sliding mode control, nonlinear control, adaptive control, neural network control and fuzzy control. Another common strategy for controlling nonlinear systems is feedback linearization, which has been applied for AUV control in previous master’s theses (Aili and Ekelund, 2016, Vervoort, 2008). Theory of how feedback linearization is used for control of underwater vehicles is given by Fossen (1994, 2011).

1.6 Limitations

Modelling and control design for an AUV can be quite extensive and time consuming. This is in agreement with the amount of comprehensive related works that deal with either solely modelling or solely control. Given the project purpose, and the restricted amount of time available for the thesis, a number of exclusions had to be made.

As determined in Section 1.5, the modelling framework of Fossen (1994) is the most common approach for modelling underwater vehicles. Therefore, no alternative model structures will be considered in this thesis.

Based on the actuator configuration of the AUV, motion control in all six degrees of freedom (DOF) will not be possible. Only control of forward speed, heading and depth will be developed.

As stated in Section 1.4, testing on the physical AUV was not possible at the time the thesis was written, due to Maribot LoLo not being ready for field testing. For that reason, development and testing was performed exclusively in a simulation environment and the results are solely based on studies in simulation. Furthermore, development of the simulation model was not part of the thesis, and instead an already existing model was used for simulation.

During development it was always accounted for that the auto-tuning method should be directly applicable to the physical AUV, and to perform system identification and control on the physical vehicle, measurements of certain states are required. States are in this context referred to as physical quantities such as position and velocity. Some states cannot be measured directly, but by combining the information from other measurements, estimates of these states can sometimes be constructed. This process is called sensor fusion. Sensor fusion had already been developed for Maribot Lolo, and was therefore not be considered in this thesis.

1.7 Thesis Outline

In Chapter 2, theory which will be applied throughout the thesis will be presented along with some preliminaries regarding the studied AUV and the simulation environment used for development and evaluation. The model structures used for system identification will be derived and methods for determining the
unknown model parameters will be presented. Lastly, some relevant control principles will be presented.

In Chapter 3, it will be presented how the AUV dynamics are identified using predetermined experiments, simplified model structures and a parameter estimation method. Both linear and nonlinear models will be estimated. After that, model-based control design will be used in order to tune the control system. Both a linear and a nonlinear control system will be considered, where auto-tuning functionality will be presented for both cases.

In Chapter 4, the linear and nonlinear control systems and their auto-tuning capabilities are tested and evaluated.

Finally, in Chapter 5, the results are discussed and a final conclusion of the thesis outcome is made. The thesis is concluded with some suggestions for future improvements and development.
The following chapter will provide theory and other information needed to understand the contents of the thesis. First, a description of the AUV studied will be given, followed by a brief presentation of the simulation environment. Thereafter, theory used to derive the dynamic model of the AUV will be summarized. This is followed by a short introduction to parameter estimation. Lastly, theory of the control principles used in the thesis is provided.

2.1 Vehicle Description

The AUV studied in this thesis is Maribot LoLo, see Figure 2.1, developed at the KTH Centre for Naval Architecture. Maribot Lolo was designed to be used as a platform vehicle by the researchers at SMaRC, who will equip the AUV with various payloads and perform a wide range of underwater missions such as environmental monitoring, seafloor mapping and underwater communication.

Below, a brief description of the design of the AUV will be given, focusing on the aspects relevant for modelling and control such as hull shape, hull dimensions, actuator configuration and sensor configuration.

2.1.1 AUV Design

In Figure 2.1, hull shape, dimensions and main components of Maribot LoLo are shown. The hull design is streamlined and similar to the the shape of a torpedo except for the rectangular cross-section. The dry weight of the AUV is approximately 600 kg, but will vary depending on what payload is placed in the 0.4 m³ payload bay located in the nose. Its maximum speed is approximately 2 m/s.

Energy storage and a Variable Buoyancy System (VBS) is placed in the mid-section
of the vehicle. These two components make up a significant amount of the vehicle’s weight and therefore it was favorable to place them in the middle in order to obtain a well-balanced centre of gravity (Deutsch et al., 2018).

The VBS consists of a ballast tank system which primary purpose is to provide neutral buoyancy, meaning that the AUV will neither sink nor float when completely submerged. It can also be used to control depth while at standstill and to perform emergency surfacing. The VBS is divided into two separate tank systems, a fore and an aft system, which enables a small degree of pitch control. Further, these two tank systems can control their respective starboard and port side tanks separately, which enables a small degree of roll control. It might be possible to utilize the VBS dynamically during operation in order to gain more stable and/or precise motion, but in order to confine the extent of the thesis, control of the VBS was not considered. However, the VBS will be assumed to provide certain properties to the AUV, such as neutral buoyancy and a well placed centre of gravity.

Similar to a torpedo, Maribot LoLo is thrust by propellers and steered using a set of rudders, also referred to as control surfaces. It is equipped with two counter-rotating propellers (CRP) placed in the aft, two rudders mounted on the upper side of the hull tail section, and an elevator placed between the propellers, see Figure 2.2. The rudders are used to control heading (yaw) and the elevator
to control tilt (pitch) of the AUV. Note that only the fins on the upper side are controllable.

The control signals which are sent to the propellers and control surfaces are desired values for propeller speed and deflection angle. Each actuator then has a separate controller which ensures that the requested value is achieved. Design of these low level actuator controllers is not part of this thesis. For control design, it is important to consider the constraints of the actuators. The actuator limitations are given in Table 2.1.

**Table 2.1: Actuator constraints.**

<table>
<thead>
<tr>
<th>Actuator</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propellers</td>
<td>±1000 RPM</td>
</tr>
<tr>
<td>Rudders</td>
<td>±20°</td>
</tr>
<tr>
<td>Elevator</td>
<td>±20°</td>
</tr>
</tbody>
</table>

Maribot LoLo is also equipped with a configuration of sensors used to navigate, i.e. determining the vehicle’s position and orientation. Navigation is a vital part for automatic control, and autonomous operation overall. The main sensors are a Doppler Velocity Log, an Attitude and Heading Reference System and a pressure sensor, with which it is possible to obtain measurements of orientation and velocity in all DOFs using sensor fusion. As mentioned in Section 1.6, sensor fusion will not be treated in this thesis, and instead it will be assumed that existing sensor fusion algorithms will provide sufficiently accurate state estimates.

For more details regarding the design of Maribot LoLo, please refer to Deutsch et al. (2018).
2.2 Simulation Environment

The simulation environment was considered as a prerequisite to the studies in the thesis and therefore, as declared in Section 1.6, development of it was not part of the thesis. An overview of the simulation environment will still be given in order to strengthen the credibility of the results obtained by using it.

2.2.1 Model Description

The simulation model was provided by Saab and was implemented in Matlab. It was built mainly based on the work by Campa and Innocenti (1999). In their documentation, a simulation model for a torpedo shaped AUv is derived, including the code required to implement it in Matlab. For this thesis, the simulation model was translated to Simulink due to preference.

In short, the simulation environment is based on the equations of motion for a six DOF rigid body, which is subject to external forces and moments such as thrust from propellers, damping from the surrounding fluid, gravitation and buoyancy. The AUv is modelled to be a cylindrical body, to which a propeller and a set of fins and rudders are attached. Rudders are modelled as mass-less bodies attached to the fuselage which can rotate about a specified axis. Damping forces are computed separately for the fuselage and each fin and rudder, depending on their orientation relative to the motion of the surrounding fluid.

The simulation environment is also able to consider underwater currents, allowing for the possibility to evaluate the control system’s robustness to such disturbances.

The simulation model was deemed as promising for the thesis due to the many similarities to Maribot LoLo, in regards of both shape and actuation. It was also believed to generate credible results, based on the knowledge of AUv modelling gathered during the literature study, and on the expertise of Saab. Regardless, using an already available simulation environment was considered as highly favourable compared to developing a new one.

The provided simulation model was modified to resemble Maribot LoLo, which included redefining the number and placement of propellers, fins and rudders, and to reassign vehicle parameters. The latter part was especially difficult since most parameters required testing on the physical AUv to be determined, and since such testing was not possible the parameters were roughly assigned based on vehicles similar to Maribot LoLo. It is therefore uncertain how well the simulation model describes the behaviour of Maribot LoLo, but for the purpose of developing the auto-tuning method this was not considered as an issue since the results can still be used for proof of concept.
2.3 Dynamic Model

A dynamic model is a mathematical description of how a physical system will react/change when subject to loads, such as forces and moments caused by actuators, the environment and gravity. Mathematical expressions will never be able to exactly describe the real system, meaning that all models are simplifications, but to various extents.

In general, an AUV will be able to move in six DOFs, i.e. linear and rotational motion in three directions respectively, as depicted in Figure 2.3. According to Fossen (1994), the dynamics of an underwater vehicle in 6 DOF can be described by

\[ \dot{\eta} = J(\eta)\nu \]  \hspace{1cm} (2.1)

\[ M\ddot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \]  \hspace{1cm} (2.2)

where the vector \( \eta \) is the position and orientation of the AUV with respect to an inertial reference, and the vector \( \nu \) is the body-fixed linear and angular velocities. The matrices \( M, C(\nu) \) and \( D(\nu) \) represent inertia, Coriolis and damping, while the vector \( g(\eta) \) represent gravitational and buoyancy forces. Lastly, the vector \( \tau \) represents the forces exerted on the AUV by its actuators and environmental disturbances.

In this thesis, the notation by SNAME (1950) will be adopted for forces, moments, linear and angular velocities, positions and Euler angles, which is given in Table 2.2. Using the notation in Table 2.2, the position vector \( \eta \) and the velocity vector \( \nu \) can be expressed as

\[ \eta = \begin{bmatrix} \eta_1^T, \eta_2^T \end{bmatrix}^T, \quad \eta_1 = [x, y, z]^T, \quad \eta_2 = [\phi, \theta, \psi]^T \] \hspace{1cm} (2.3a)

\[ \nu = \begin{bmatrix} \nu_1^T, \nu_2^T \end{bmatrix}^T, \quad \nu_1 = [u, v, w]^T, \quad \nu_2 = [p, q, r]^T \] \hspace{1cm} (2.3b)

In this chapter, the model formulation in (2.1)-(2.2) will be derived for the AUV studied in this thesis. A number of assumptions and simplifications will be made.

Figure 2.3: Definition of the body-fixed and inertial coordinate systems, and illustration of the six degrees of freedom.
Table 2.2: The notation by SNAME (1950) for marine vehicles.

<table>
<thead>
<tr>
<th>DOF</th>
<th>Forces and moments</th>
<th>Body-fixed velocities</th>
<th>Positions and orientations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Motions in x-direction (surge)</td>
<td>X</td>
<td>u</td>
</tr>
<tr>
<td>2</td>
<td>Motions in y-direction (sway)</td>
<td>Y</td>
<td>v</td>
</tr>
<tr>
<td>3</td>
<td>Motions in z-direction (heave)</td>
<td>Z</td>
<td>w</td>
</tr>
<tr>
<td>4</td>
<td>Rotations about x-axis (roll)</td>
<td>K</td>
<td>p</td>
</tr>
<tr>
<td>5</td>
<td>Rotations about y-axis (pitch)</td>
<td>M</td>
<td>q</td>
</tr>
<tr>
<td>6</td>
<td>Rotations about z-axis (yaw)</td>
<td>N</td>
<td>r</td>
</tr>
</tbody>
</table>

in order to reduce the complexity and number of parameters in the model, which will alleviate the parameter estimation process. For the purpose of control design, simplified models are often sufficient since unmodelled dynamics can, to an extent, be considered as disturbances which the controller can compensate for (Glad and Ljung, 2006). In control theory this is referred to as controller robustness.

Skew-symmetric matrices will be used in order to simplify notation and to denote the vector cross-product, see Definition 2.1.

**Definition 2.1 (Vector Cross-Product).** The vector cross-product \( \times \) is defined as

\[
\lambda \times a := S(\lambda)a
\]

where \( \lambda \) and \( a \) are three-dimensional vectors, and \( S(\lambda) \) is a skew-symmetric matrix defined as

\[
S(\lambda) = -S^T(\lambda) = \begin{bmatrix}
0 & -\lambda_3 & \lambda_2 \\
\lambda_3 & 0 & -\lambda_1 \\
-\lambda_2 & \lambda_1 & 0
\end{bmatrix}, \quad \lambda = \begin{bmatrix}
\lambda_1 \\
\lambda_2 \\
\lambda_3
\end{bmatrix}
\]

### 2.3.1 Kinematics

When describing the motion of an AUV, at least two coordinate systems are needed, one local and one global. The local coordinate system is body-fixed, meaning it will translate and rotate with the AUV, while the global coordinate system is an Earth-fixed (inertial) reference frame. The body-fixed coordinate system is placed in centre of origin (CO) with the x-axis pointing towards the bow, the y-axis pointing starboard and the z-axis pointing downwards, as depicted in Figure 2.3. Maribot LoLo has two planes of symmetry, the xy- and xz-planes, and
CO is placed midships along their intersection. The Earth-fixed coordinate system is in this thesis chosen as a North-East-Down (NED) system, where the x-axis is pointing north, the y-axis is pointing east and the z-axis is pointing down towards the Earth’s centre. The origin for the NED frame is an arbitrary point on the Earth’s surface. This choice of inertial reference frame is only suitable as long as the AUV operates in a local area, since it approximates the Earth’s surface as flat. If the AUV moves away from the inertial origin, e.g. along the surface of the Earth, the true downwards direction will change due to the curvature of the Earth, which will not be considered by the North-East-Down system. This can be handled by using additional coordinate systems as presented by Fossen (2011), but will not be treated in this thesis.

The linear and angular velocities of the AUV are conveniently expressed in the body-fixed coordinate system, by the vector $\nu$, while its position and orientation are described relative to the Earth-fixed coordinate system, by the vector $\eta$. The kinematic expression in (2.1), which relates linear and angular velocities in the two coordinate systems, can be derived using Euler angles. Euler angles are used to describe the relative orientation of two coordinate systems using a sequence of three principal rotations in roll, pitch and yaw respectively. In this thesis, the $zyx$ convention will be used. In Figure 2.4, the sequence of rotations are shown.

The kinematic relation which transforms linear velocities in the body-fixed frame to the inertial frame is given by

$$\dot{\eta}_1 = J_1(\eta_2)\nu_1 \quad (2.4)$$

**Figure 2.4:** The three rotations used to relate the body-fixed coordinate system to the inertial coordinate system. The rotations are defined from the inertial system, starting with a yaw rotation $\psi$, followed by a pitch rotation $\theta$ and finally a roll rotation $\phi$. The blue and red coordinate systems are intermediate systems.
where $J_1(\eta_2)$ is the rotation matrix for the $zyx$ convention

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta \phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\phi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\phi & c\theta s\phi & c\theta c\phi \end{bmatrix}$$  \hspace{1cm} (2.5)

where $s \cdot$ and $c \cdot$ stand for $\sin(\cdot)$ and $\cos(\cdot)$.

The inverted transformation matrix is simply given by

$$J_1(\eta_2)^{-1} = J_1(\eta_2)^T$$  \hspace{1cm} (2.6)

due to the orthogonality of rotation matrices.

Transformation of angular velocities is given by

$$\dot{\eta}_2 = J_2(\eta_2)\nu_2$$  \hspace{1cm} (2.7)

where the transformation matrix $J_2(\eta_2)$ is given by

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix}$$  \hspace{1cm} (2.8)

where $s \cdot$, $c \cdot$ and $t \cdot$ stand for $\sin(\cdot)$, $\cos(\cdot)$ and $\tan(\cdot)$ respectively.

Unlike the transformation matrix for linear velocities, the transformation matrix for angular velocities is not an orthogonal matrix, meaning that $J_2(\eta_2)^{-1} \neq J_2(\eta_2)^T$. Instead it is given by

$$J_2(\eta_2)^{-1} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{bmatrix}$$  \hspace{1cm} (2.9)

Using Euler angles to describe the kinematic relations comprises a singularity for $\theta = \pm \pi/2$, as can be seen in (2.8). This can be resolved by expressing the angles using quaternions instead. Since the AUV considered in this thesis is not intended to operate close to this singularity, rather it is undesired, there is no need to implement the quaternion representation.

Combining the transformations of linear and angular velocities yield the kinematic relation in (2.1) as

$$\dot{\eta} = J(\eta)\nu = \begin{bmatrix} J_1(\eta_2) & 0_{3 \times 3} \\ 0_{3 \times 3} & J_2(\eta_2) \end{bmatrix} \nu$$  \hspace{1cm} (2.10)

### 2.3.2 Rigid-body Kinetics

Rigid-body kinetics is used to describe the motion of the AUV when subject to external loads. The equations of motion can be derived using the Newton-Euler formulation and are expressed as

$$M_{RB} \ddot{\nu} + C_{RB}(\nu)\nu = \tau_{RB}$$  \hspace{1cm} (2.11)
where $M_{RB}$ is the mass and inertia matrix, $C_{RB}(\nu)$ is the Coriolis and centripetal matrix, and $\tau_{RB}$ is the vector with external loads acting on the rigid-body (Fossen, 2011). The mass and inertia matrix is a $6 \times 6$ matrix given by

$$M_{RB} = \begin{bmatrix} mI_{3\times3} & -mS(r_g) \\ mS(r_g) & I_b \end{bmatrix}$$  \hspace{1cm} (2.12)

where $m$ is the AUV mass, $I_b$ is the inertia matrix defined in CO, and $r_g$ is the vector from CO to the centre of gravity (CG). $I_{3\times3}$ is a $3 \times 3$ identity matrix. By assuming that the CG lies close to the CO, the off-diagonal elements in (2.12) will become negligible, resulting in

$$M_{RB} = \begin{bmatrix} mI_{3\times3} & 0 \\ 0 & I_b \end{bmatrix}$$  \hspace{1cm} (2.13)

Since the rigid-body will rotate relative to the inertial frame, it will be subject to Coriolis and centripetal forces determined by the matrix $C_{RB}(\nu)$ defined in (2.14) (Fossen, 2011). By once again assuming that CG lies close to CO, and also that the AUV is symmetric in all three planes, the resulting contribution by Coriolis and centripetal forces is given by

$$C_{RB}(\nu)\nu = \begin{bmatrix} mS(\nu_2) & -mS(\nu_2)S(r_g) \\ S(r_g)S(\nu_2) & -S(I_b \nu_2) \end{bmatrix} \nu = \begin{bmatrix} m(qw - rv) \\ m(ru - pw) \\ m(kv - au) \\ qr(I_{zz} - I_{yy}) \\ pr(I_{xx} - I_{zz}) \\ pq(I_{yy} - I_{xx}) \end{bmatrix}$$  \hspace{1cm} (2.14)

### 2.3.3 Hydrodynamics

When the AUV travels through water it will be subject to forces and moments caused by the surrounding fluid, referred to as hydrodynamic forces. These forces and moments can be described by

$$\tau_{dyn} = -M_A \dot{\nu} - C_A(\nu)\nu - D(\nu)\nu$$  \hspace{1cm} (2.15)

where $M_A$ represents added mass, $C_A(\nu)$ represents Coriolis and centripetal forces and moments caused by the added mass, and $D(\nu)$ represents damping (Fossen, 2011).

When an AUV accelerates/decelerates it will also accelerate/decelerate the immediate surrounding fluid (Fossen, 1994). This phenomenon can be interpreted as an additional mass attached to the AUV. By combining the assumption that the AUV has three planes of symmetry with the fact that the AUV will move at low speed, the contributions by added mass can be simplified to the expressions

$$M_A = -\text{diag}\{X_{ii}, Y_{ii}, Z_{ii}, K_p, M_{\dot{\alpha}}, N_{ij}\}$$  \hspace{1cm} (2.16)
where $X_{\dot{u}}$, $Y_{\dot{v}}$, $Z_{\dot{w}}$ are the added mass parameters in each axis direction and $K_{\dot{p}}$, $M_{\dot{q}}$, $N_{\dot{r}}$ are the added inertia (Fossen, 2011).

Hydrodynamic damping is caused by several factors such as drag and skin friction (Fossen, 2011). The various contributions are divided into two separate terms, one for linear damping, $D_l$, and one for nonlinear damping, $D_{nl}$. In Fossen (2011) the nonlinear term consists of quadratic damping. By assuming that the AUV only performs noncoupled motion, i.e. only moves in one DOF at the time, it is reasonable to assume a diagonal damping matrix $D$ (Fossen, 2011),

$$D(\nu) = D_l + D_{nl}(\nu)$$

$$= -\text{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\} - \text{diag}\{X_{|\dot{u}|u}|u|, Y_{|\dot{v}|v}|v|, Z_{|\dot{w}|w}|w|, K_{|\dot{p}|p}|p|, M_{|\dot{q}|q}|q|, N_{|\dot{r}|r}|r|\}$$

where $X_{\dot{u}}$, $Y_{\dot{v}}$, $Z_{\dot{w}}$, $K_{\dot{p}}$, $M_{\dot{q}}$, $N_{\dot{r}}$ are the linear damping coefficients and $X_{|\dot{u}|u}$, $Y_{|\dot{v}|v}$, $Z_{|\dot{w}|w}$, $K_{|\dot{p}|p}$, $M_{|\dot{q}|q}$, $N_{|\dot{r}|r}$ are the nonlinear damping coefficients.

Finally, it should be mentioned that all the hydrodynamic terms will in reality depend on the relative velocity vector $\nu$, defined as

$$\nu_r = \nu - \nu_c$$

where $\nu_c \in \mathbb{R}^6$ is the velocity of the water current (Fossen, 2011). However, for the simplified models used in this thesis it was assumed that the ocean currents were identically equal to zero, resulting in $\nu_r \equiv \nu$.

### 2.3.4 Hydrostatics

Regardless of the speed of the AUV it will always be subject to gravitational and buoyancy forces, also referred to as restoring forces by Fossen (2011). These forces are determined by

$$W = mg, \quad B = \rho g \nabla V$$

where $m$ is the vehicle mass, $g$ is the gravitational acceleration, $\rho$ is the water density and $V$ is the volume of fluid displaced by the AUV. The gravitational force will act on CG while the buoyancy force will act on the centre of buoyancy (CB). The hydrostatic forces can be written as

$$\tau_{stat} = -g(\eta)$$
where $g(\eta)$ expresses the forces in the body-fixed coordinate system as

$$
g(\eta) = -\left[ f_g \times f_g + f_b \times f_b \right] = 
\begin{bmatrix}
(W - B) \sin \theta \\
-(W - B) \cos \theta \sin \phi \\
-(W - B) \cos \theta \cos \phi \\
z_G B \cos \theta \sin \phi \\
z_G B \sin \theta \\
0
\end{bmatrix}
$$

(2.22)

In (2.22), $r_b$ and $r_g$ are the distances from CO, and $f_g = J_1(\eta_2)^T[0, 0, W]^T$ and $f_b = J_1(\eta_2)^T[0, 0, -B]^T$ are the gravitational and buoyancy forces transformed from the inertial frame to the body-fixed frame. It has been assumed that CB coincides with CO and that CG has been reallocated, either by using the VBS or by adding balancing weights, so that it lies beneath CB. These assumptions give

$$
r_b = [0, 0, 0]^T, \quad r_g = [0, 0, z_G]^T
$$

(2.23)

If it is further assumed that the net buoyancy force can be controlled, also by either using the VBS or adding float material or weights, so that neutral buoyancy is obtained, $B = W$, then the hydrostatics forces can be simplified to

$$
g(\eta) = 
\begin{bmatrix}
0 \\
0 \\
0 \\
z_G B \cos \theta \sin \phi \\
z_G B \sin \theta \\
0
\end{bmatrix}
$$

(2.24)

The two remaining terms in (2.24) will provide passive, stabilizing moments in both pitch and roll, as long as $z_G > 0$, which is illustrated in Figure 2.5. On the contrary, if $z_G < 0$ these moments will make the vehicle unstable, making it want to turn upside down.

\textbf{Figure 2.5:} Illustrations showing the passive, stabilizing moments in pitch and roll caused by CG being located directly below CB.
2.3.5 Actuators

As presented in Section 2.1.1, Maribot LoLo is equipped with two types of actuators, propellers and control surfaces. The effect of these actuators on the AUV dynamics can be modelled as

$$\tau_{act} = T(\nu)f(u)$$  (2.25)

where $u$ is a vector of actuator states, and $T(\nu)$ is a matrix describing how the actuator states affect each DOF in the body-fixed coordinate system. The vector $u$ is defined as

$$u = [n_1, n_2, \delta_{r1}, \delta_{r2}, \delta_e]^T$$  (2.26)

where $n_1, n_2$ and $\delta_{r1}, \delta_{r2}$ are the rotational speeds and deflection angles of the left and right propeller and rudder, respectively. Lastly, $\delta_e$ is the deflection angle of the elevator.

The empirical expressions for lift and drag on control surfaces are given by

$$F_{lift} = \frac{1}{2}\rho V_f^2 A_f C_L(\alpha), \quad F_{drag} = \frac{1}{2}\rho V_f^2 A_f C_D(\alpha)$$  (2.27)

where $\rho$ is the fluid density, $A_f$ a representative area, $V_f$ is the relative flow speed and $C_L(\alpha)$ and $C_D(\alpha)$ are lift and drag coefficients dependent on the angle of attack $\alpha$ (Perez, 2006). From the literature study it was found that a common approach is to disregard the dependency of the angle of attack and thereby to assume that the lift and drag coefficients are constant, and also to consider the surge speed as the true speed of the vehicle (Luque et al., 2009, Sabet et al., 2018). Furthermore, drag forces on the control surfaces will be neglected, resulting in the following models for lift and drag forces

$$F_{lift} \propto u^2 \delta, \quad F_{drag} \approx 0$$  (2.28)

where $\delta$ is the deflection angle.

Using an energy-perspective, Yoerger et al. (1990) have derived the following model for propeller thrust

$$F_{prop} \propto |n|n$$  (2.29)

which has been verified to be reasonable at low speeds (Blanke et al., 2000), and should therefore be suitable for the AUV considered in this thesis.

In general, $T(\nu)$ would be a full matrix but based on the geometrical placement of the actuators and the previous assumptions it can be reduced. Using the nomenclature for forces and moments in Table 2.2, the expression in (2.25) can be writ-
ten as

\[
\tau_{\text{act}} = \begin{bmatrix}
X_{\text{prop}} & X_{\text{prop}} & 0 & 0 & 0 \\
0 & 0 & Y_{\delta_r} u^2 & Y_{\delta_r} u^2 & 0 \\
0 & 0 & 0 & 0 & Z_{\delta_e} u^2 \\
0 & 0 & K_{\delta_r} u^2 & K_{\delta_r} u^2 & 0 \\
0 & 0 & 0 & 0 & M_{\delta_e} u^2 \\
N_{\text{prop}} & -N_{\text{prop}} & N_{\delta_r} u^2 & N_{\delta_r} u^2 & 0
\end{bmatrix}
\begin{bmatrix}
|n_1|n_1 \\
|n_2|n_2 \\
\delta_{r1} \\
\delta_{r2} \\
\delta_{e}
\end{bmatrix}
\] (2.30)

where \(X_{\text{prop}}, N_{\text{prop}}, Y_{\delta_r}, K_{\delta_r}, Z_{\delta_e}, M_{\delta_e}\) are proportionality constants. If both propellers and both rudders are controlled identically \((n = n_1 = n_2, \delta_r = \delta_{r1} = \delta_{r2})\), the number of control signals is reduced from five to three.

\[
\tau_{\text{act}} = \begin{bmatrix}
2X_{\text{prop}} & 0 & 0 \\
0 & 2Y_{\delta_r} u^2 & 0 \\
0 & 0 & Z_{\delta_e} u^2 \\
0 & 2K_{\delta_r} u^2 & 0 \\
0 & 0 & M_{\delta_e} u^2 \\
0 & 2N_{\delta_r} u^2 & 0
\end{bmatrix}
\begin{bmatrix}
|n|n \\
\delta_{r} \\
\delta_{e}
\end{bmatrix}
\] (2.31)

It might be argued that some control potential is lost by reducing the number of control inputs, and therefore this topic will be further discussed in Section 2.5.1 and 3.2.4.

As mentioned in Section 2.1, the available control inputs are reference values for the actuator states in (2.26), which then are obtained using low level controllers. In other words, the actuator state can not be chosen statically, only dynamically. The closed-loop propeller dynamics will be modelled using a first-order system

\[
\dot{n} = \frac{-n + n_{\text{ref}}}{\tau_{\text{prop}}}
\] (2.32)

and the closed-loop rudder and elevator dynamics as second order systems

\[
\dot{\delta}_r = \frac{-\delta_r + k_r (\delta_{r,\text{ref}} - \delta_r)}{\tau_r}
\] (2.33a)

\[
\dot{\delta}_e = \frac{-\delta_e + k_e (\delta_{e,\text{ref}} - \delta_e)}{\tau_e}
\] (2.33b)

### 2.3.6 Motion Model of AUV

Combining (2.15), (2.21) and (2.25) gives the vector of forces and moments acting on the rigid body

\[
\tau = \tau_{\text{dyn}} + \tau_{\text{stat}} + \tau_{\text{act}} + \tau_{\text{dist}}
\] (2.34)

where \(\tau_{\text{dist}}\) are unmodelled effects and environmental disturbances such as underwater currents. Inserting (2.34) in (2.11) yields

\[
M_{\text{RB}} \ddot{\nu} + C_{\text{RB}}(\nu) \nu + M_A \dot{\nu} + C_A(\nu) \nu + D(\nu) \nu + g(\eta) = T(\nu)f(u) + \tau_{\text{dist}}
\] (2.35)
which can finally be rewritten as the model by Fossen (1994) presented in (2.2) where

\[ M = M_{RB} + M_A, \quad C(v) = C_{RB}(v) + C_A(v) \]  

(2.36)

and by solving the system of equations for the state derivatives in \( \dot{v} \), models

\[
\begin{align*}
\dot{u} &= \frac{X_u u + X_{|v|} v}{m - X_u} + \frac{(m - Y_v r v - (m - Z_w) q w)}{m - X_u} + \frac{2X_{prop} |n| n}{m - X_u} \\
\dot{v} &= \frac{Y_v v + Y_{|v|} v}{m - Y_v} + \frac{(m - Z_w) p w - (m - X_u) r u}{m - Y_v} + \frac{2Y_{\delta_r} u^2 \delta_r}{m - Y_v} \\
\dot{w} &= \frac{Z_w w + Z_{|w|} w}{m - Z_w} + \frac{(m - X_u) q u - (m - Y_v) p v}{m - Z_w} + \frac{Z_{\delta_e} u^2 \delta_e}{m - Z_w} \\
\dot{p} &= \frac{K_p p + K_{|p|} p}{I_x - K_p} + \frac{(I_y - M_q) q r - (I_z - N_r) q r + (Z_w - Y_v) v w}{I_x - K_p} \\
\dot{q} &= \frac{M_q q + M_{|q|} q}{I_y - M_q} + \frac{(I_z - N_q) p r - (I_x - K_p) p r + (X_u - Z_w) u w}{I_y - M_q} \\
\dot{r} &= \frac{N_r r + N_{|r|} r}{I_z - N_r} + \frac{(I_x - K_p) p q - (I_y - M_q) p q + (Y_v - X_u) u v}{I_z - N_r} + \frac{2N_{\delta_r} u^2 \delta_r}{I_z - N_r}.
\end{align*}
\]

(2.37)

for each DOF are obtained.

### 2.4 Parameter Estimation

The models derived in Section 2.3 contain a number of unknown parameters which need to be estimated. The goal of parameter estimation is to select these parameters such that the resulting model best describes the system behaviour. This selection is based on data collected from the system, which typically consist of sensor measurements and control inputs. Since the data will be sampled it is sensible to treat discrete models during estimation. A general discrete-time state space model is given by

\[
x_{k+1} = f(x_k, u_k, \theta) \\
y_k = h(x_k, u_k, \theta)
\]

(2.38)

where \( x_k, u_k \) and \( y_k \) are the state vector, input vector and output vector at time sample \( k \), and \( \theta \) is the vector with unknown model parameters. For any \( \theta \), the model (2.38) will, at sample \( k - 1 \), provide a prediction of the output at time \( k \).
By comparing the predicted output \( \hat{y}_k(\theta) \) to the measured output \( y_k \), a prediction error

\[
\epsilon_k(\theta) = y_k - \hat{y}_k(\theta)
\]

can be computed (Ljung and Glad, 2004). The goal of parameter estimation can now be interpreted as finding which \( \theta \) minimizes the prediction error for a set of data, which can be represented by the following cost function

\[
V_N(\theta) = \frac{1}{N} \sum_{k=1}^{N} \epsilon_k^2(\theta)
\]

where the estimate of \( \theta \) is given by

\[
\hat{\theta} = \arg\min_{\theta} V_N(\theta)
\]

When evaluating the quality of the estimated model, also known as model validation, it will be tested how well it describes a different set of data, i.e. independent data which was not used during parameter estimation. This is known as cross validation, and is used to detect overfitting (Ljung and Glad, 2004).

To evaluate the model quality, the normalized root mean square error will be used, as by Aili and Ekelund (2016), which is a more general measure used for model validation. More specifically, the model fit

\[
\text{Fit} = 100 \left( 1 - \frac{\sqrt{\frac{1}{N} \sum_{k=1}^{N} \left( y(k) - \hat{y}(k) \right)^2}}{\sqrt{\frac{1}{N} \sum_{k=1}^{N} \left( y(k) - \frac{1}{N} \sum_{k=1}^{N} y(k) \right)^2}} \right)
\]

will be used as a measure of the model quality.

The parameter estimation process will compose two main parts of the auto-tuning approach, collecting data and a method for estimating the parameters. These two parts will be further discussed in the following sections.

### 2.4.1 Experiments & Data Collection

In order to obtain informative data which can be used for parameter estimation and model validation, some consideration must be put into planning what experiments should be made, i.e. what control signals should be used to excite the system. Ljung and Glad (2004) suggest that a signal which randomly switches between two amplitudes (binary signal) is suitable for linear systems, see Figure 2.6a. Since it has been declared that the dynamics of an AUV are nonlinear it is not suitable to use a signal which only switches between the same two levels, and therefore a signal which also switches to random amplitudes was used for all experiments, see Figure 2.6b. The switching probability of the signal can be tuned to a desired frequency, and it can be scaled and offset in order to control the magnitude of the system output.

The collected data was split into two equally large datasets, one for parameter
estimation and one for model validation.

### 2.4.2 Least Squares Estimation

A commonly used method for solving the minimization problem in (2.41) is to use Least Squares (LS). Consider a discrete-time model which can be used to formulate a prediction $\hat{y}(t|\theta)$ of the output at time $t$ using measurements of the input and output at time $s \leq t - 1$. In general, the prediction $\hat{y}(t|\theta)$ can be a complicated function of $\theta$, but a special case is obtained if it can be formulated as a linear function of $\theta$, known as a linear regression

$$\hat{y}(t|\theta) = \theta^T \varphi(t)$$

where $\varphi$ is the regression vector containing previous in- and outputs (Ljung and Glad, 2004). The elements in $\varphi$ are referred to as regressors.

The LS estimate can then be computed by the following matrix multiplication

$$\hat{\theta} = R_N^{-1} f_N$$

where the matrices $f_N$ and $R_N$ are defined as

$$f_N = \frac{1}{N} \sum_{t=1}^{N} \varphi(t)y(t)$$

$$R_N = \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi(t)^T$$

### 2.4.3 Estimation Using Instrumental Variables

Parameter estimation using the LS method is only consistent under certain restrictive conditions (Ljung, 1999). If these conditions are not met, the estimator
\( \hat{\theta} \) will contain a bias error. In some cases this will be due to the regression vector \( \varphi(t) \) being correlated with process disturbances in the system, such as noise. The parameter estimation using the LS method can be written as

\[
\hat{\theta} = \left[ \frac{1}{N} \sum_{t=1}^{N} \varphi(t)\varphi(t)^T \right]^{-1} \frac{1}{N} \sum_{t=1}^{N} \varphi(t)y(t) \tag{2.46}
\]

A similar, more general approach is given by

\[
\hat{\theta} = \left[ \frac{1}{N} \sum_{t=1}^{N} \zeta(t)\varphi(t)^T \right]^{-1} \frac{1}{N} \sum_{t=1}^{N} \zeta(t)y(t) \tag{2.47}
\]

where \( \zeta(t) \) is some other vector instead of \( \varphi(t) \). This approach for estimating linear regression models is called an Instrumental Variable (IV) method, where the elements in \( \zeta \) are called instruments or instrumental variables (Ljung, 1999).

In order for the IV method to work properly it is required that the instrumental variables in \( \zeta(t) \) are correlated with the regression vector \( \varphi(t) \) but uncorrelated with the process disturbances (Ljung, 1999). There are different approaches to selecting appropriate instruments, but for this thesis only basic methods have been considered.

### 2.4.4 Recursive Parameter Estimation

In the estimation methods in Section 2.4.2-2.4.3, the estimates are computed using all provided data simultaneously, which requires that data collection and parameter estimation are performed at separate occasions, referred to as offline identification methods (Ljung, 1999). Parameter estimation can also be performed in real-time where the estimates are based on observations up to the current time and updated recursively as new observations are provided. These methods are most commonly referred to as recursive identification methods or online identification methods, and are central in adaptive applications and for fault detection (Ljung, 1999).

There are recursive variants to both the LS and IV methods presented, but an important detail to consider is that during normal operation the AUV will be operating in closed loop, i.e. using feedback control and therefore the input will no longer be uncorrelated with process disturbances (Ljung and Glad, 2004). This might cause inconsistencies when using recursive least squares (RLS) while recursive IV is still a viable option given an appropriate choice of instruments (Gilson and Van den Hof, 2005).

Ljung (1999) has described the recursive algorithm for IV estimation, see Algorithm 1, where \( \hat{\theta}(t) \) is the current estimate, \( L(t) \) is an intermediate matrix and \( P(t) \) is a matrix representing previous observations.

The algorithm needs to be provided initial values \( \theta_0 \) and \( P_0 \). Since the matrix \( P(t) \) is analogous to the covariance matrix of \( \hat{\theta}(t) \), \( P_0 \) can interpreted as the confidence in the initial estimate \( \theta_0 \). A small \( P_0 \) corresponds to high confidence since the
Algorithm 1: Recursive IV Estimation (Ljung, 1999).

Initialization:
\[ \hat{\theta}(0) = \theta_0 \]
\[ P(0) = P_0 \]

Estimate update:
\[ L(t) = P(t-1)\zeta(t) \]
\[ \hat{\theta}(t) = \hat{\theta}(t-1) + L(t) \left[ y(t) - \varphi^T(t)\hat{\theta}(t-1) \right] \]

Covariance update:
\[ P(t) = P(t-1) - \frac{P(t-1)\zeta(t)\varphi^T(t)P(t-1)}{1 + \varphi^T(t)P(t-1)\zeta(t)} \]

The estimate will not change much from the initial value, whereas for a large \( P_0 \) the estimate will quickly deviate from \( \theta_0 \) (Soderstrom and Stoica, 1989).

2.5 Motion Control

The goal of motion control is to make a system behave as desired automatically, without human interaction. One main principle in control theory is feedback control illustrated in Figure 2.7, where the system output is either measured or estimated and then compared to a desired output. Based on the desired and measured output, a controller will compute a control signal which aims to manipulate the system output to make it more similar to the output reference.

For the initial sea trials of Maribot LoLo, a control system using linear proportional, integral, derivative (PID) controllers will be implemented, and therefore an auto-tuning approach which is applicable to such controllers will be developed. An alternative, nonlinear control system will also be developed and compared to the linear control system as a potential future improvement.

In the following section some preliminaries regarding control of Maribot LoLo will be discussed, followed by theory of relevant areas within automatic control.

\[ \sum \quad \hat{y} \quad \text{Controller} \quad u \quad \text{System} \quad y \]

Figure 2.7: Block diagram illustrating the principle of feedback control. The notation in the figure is standard notation used in control theory and should not be confused with the notation for marine vehicles.
2.5 Motion Control

2.5.1 Control Problem

In theory, a submerged vehicle could have the ability to move arbitrarily in all six DOFs, which is common for ROVs using a configuration of multiple thrusters. Maribot LoLo on the other hand is under-actuated, meaning it cannot be controlled in certain DOFs. Using its propellers and control surfaces it is possible to control the vehicle in surge, pitch and yaw, but not in sway, heave and roll. The actuation of Maribot LoLo is similar to an airplane, having a main source of thrust propelling it forward and control surfaces/wings to steer. Likewise, the AUV needs to have a speed relative the surrounding medium in order to generate lift on its control surfaces.

Each actuator of Maribot LoLo has, by design, a distinct affect on a specific DOF, and it is therefore sensible to apply decoupled control, meaning that each control input is dedicated to control a specific output, in this case a specific DOF. Any cross couplings will be considered as disturbances which the controllers will have to compensate for. The decoupled control system will consist of the following three individual controllers:

- Surge speed controller, heron referred to as simply speed controller, using the propeller speed as control signal.
- Depth controller, using the elevator as control signal.
- Heading controller, using the rudders as control signal.

In the depth controller, the elevator will be used to control the pitch angle which in combination with surge speed will affect the depth.

An interesting observation is the asymmetry in rudder configuration. Due to only being able to control the rudders on the upper side of the vehicle, a rolling moment will be generated when trying to control yaw. Since currently Maribot LoLo has no means of actively controlling roll motion, it must rely on the passive restoring moment, mentioned in Section 2.3.4, to compensate for this effect. In order for the depth controller to work as intended it is important that the roll angle is kept close to single digit degrees. This can also be ensured by performing decoupled motion, i.e. not turning while diving.

Another interesting observation is that despite being underactuated in some DOFs, Maribot LoLo is overactuated in yaw. Yawing motion can also be achieved by having the propellers generate different thrust, causing a yawing moment. This is known as differential drive and would be valuable to implement since it would allow turning at low speeds and even at standstill. An approach for integrating differential drive will be presented in Section 3.2.4.

2.5.2 Pole Placement

Pole placement is an approach for controller synthesis which is applicable to linear systems. Given that the system dynamics can be described by a linear model, the model can be used to perform analytic control design. Consider the feedback
system in Figure 2.7, where the controller and system have the transfer functions $F(s)$ and $G(s)$. The closed-loop transfer function $G_c(s)$ will be given by

$$
G_c(s) = \frac{G(s)F(s)}{1 + G(s)F(s)}
$$

(2.48)

where the main dynamic properties are determined by the roots of the denominator, known as the poles of $G_c(s)$. The principle of pole placement is to choose the controller $F(s)$ such that the poles of the closed-loop system is placed at locations which will give the system a desired behaviour. For more theory regarding transfer functions and poles, please refer to Glad and Ljung (2006) or any other literature on basic control theory.

In Åström et al. (1993), pole placement is demonstrated for a second order system with the following structure

$$
G(s) = \frac{k}{(1 + sT_1)(1 + sT_2)}
$$

(2.49)

using a PID controller in parallel form

$$
F(s) = K\left(1 + \frac{1}{T_i s} + T_d s\right)
$$

(2.50)

The second order model (2.49) has three model parameters and Åström et al. (1993) explain that by using a PID controller which also has three parameters, it is possible to arbitrarily place the three poles of the closed-loop system in (2.48).

Åström et al. (1993) suggest that

$$
(s + \alpha \omega)(s^2 + 2\zeta \omega s + \omega^2) = 0
$$

(2.51)

is a suitable closed-loop characteristic equation, which contains two dominant poles with relative damping $\zeta$ and frequency $\omega$, and a real pole located in $-\alpha \omega$. Straightforward calculations show that the characteristic equation in (2.51) is obtained by choosing the control parameters as

$$
K = \frac{T_1 T_2 \omega^2(1 + 2\zeta \alpha) - 1}{k}
$$

(2.52a)

$$
T_i = \frac{T_1 T_2 \omega^2(1 + 2\zeta \alpha) - 1}{T_1 T_2 \alpha \omega^3}
$$

(2.52b)

$$
T_d = \frac{T_1 T_2 \omega(\alpha + 2\zeta) - T_1 - T_2}{T_1 T_2(1 + 2\zeta \alpha)\omega^2 - 1}
$$

(2.52c)

Using basic knowledge in control theory, it can be shown that $\omega$ will determine the speed of the dynamics, where a larger $\omega$ will yield faster dynamics. The parameter $\zeta$ is correlated to damping, where a larger value will yield better damping, i.e. less overshoot and oscillations. The effect of the parameter $\alpha$ is not as distinct as for $\omega$ and $\zeta$. However, an important observation is that for

$$
\alpha_c = \frac{T_1 + T_2}{T_1 T_2 \omega} - 2\zeta
$$

(2.53)
pure PI control is obtained, and for $\alpha < \alpha_c$ the derivative time becomes negative which is undesired.

In the simulation model the PID controllers are implemented in an alternative parallel form

$$F(s) = K_p + K_i \frac{1}{s} + K_d s$$  \hspace{1cm} (2.54)

which will be the PID representation used in this thesis. Translation of the control parameters in (2.50) to (2.54) is simply given by

$$K_p = K, \quad K_i = \frac{K}{T_i}, \quad K_d = KT_d$$  \hspace{1cm} (2.55)

and insertion of (2.52) gives

$$K_p = \frac{T_1 T_2 \omega^2 (1 + 2\zeta \alpha) - 1}{k}$$  \hspace{1cm} (2.56a) \\
$$K_i = \frac{T_1 T_2 \alpha \omega^3}{k}$$  \hspace{1cm} (2.56b) \\
$$K_d = \frac{T_1 T_2 \omega (\alpha + 2\zeta) - T_1 - T_2}{k}$$  \hspace{1cm} (2.56c)

### 2.5.3 Cascade Control

In some systems there are more than one measurement signal which can be used for control. If the system dynamics can be separated into two subsystem in series, and the intermediate signal can be measured, cascade control can be applied (Glad and Ljung, 2006). The structure of a cascade controller is shown in Figure 2.8 where $F_1$, $F_2$ are controllers and $G_1$, $G_2$ are subsystems. The cascade controller consists of an inner and an outer loop, where the inner loop obtains its reference value from the outer loop controller. The main advantage of using cascade controllers is that disturbances on subsystem $G_2$ will immediately be compensated for by the inner loop, reducing the impact on the output $y$ (Glad and Ljung, 2006). This will be the case as long as the inner loop dynamics are faster than the outer loop dynamics, which therefore is a prerequisite for using cascade control.

Another advantage of cascade control is that if it is desired to control the inner state $z$ as the main output, the outer loop can be disabled by setting the reference value $z_{ref}$ directly instead. It will hence not be necessary to implement an

![Figure 2.8: Block diagram illustrating the principle of cascade control.](image)
addition control loop.

Procedures for tuning cascade controllers can be found in Åström and Hägglund (1995).

### 2.5.4 Feedback Linearization

Feedback linearization is a method used to handle nonlinearities in a system. If the nonlinear dynamics can be modelled, it is sensible to try to use that information in a controller to compensate for the nonlinear behaviour, effectively making the system linear (Glad and Ljung, 2003). This would allow linear control strategies to be applied instead.

Fossen (2011) demonstrates how feedback linearization could be implemented for the nonlinear model of underwater vehicles in (2.2). By choosing the actuator forces as

\[
\tau = Ma_b + C(\nu)\nu + D(\nu)\nu + g(\eta) = Ma_b + n(\nu, \eta) \quad (2.57)
\]

the resulting vehicle dynamics become

\[
\dot{\nu} = a_b \quad (2.58)
\]

where \(a_b\) is the vector of control inputs to the linearized system, which can be computed using linear control strategies. The implementation of feedback linearization is illustrated in Figure 2.9.

In the implementation by Fossen (2011) it is assumed that the vector \(\tau\) can be selected statically, which will only be the case if actuator dynamics are neglected. It is possible to also consider the actuator dynamics, but this will require state feedback linearization to be implemented instead (Glad and Ljung, 2003).

Since feedback linearization is achieved via the control input, it is important to realize that it will only work as long as the control input is not saturated (Glad and Ljung, 2003).

---

**Figure 2.9:** Block diagram illustrating the implementation of feedback linearization. The relation between \(a_b\) and \(\nu\) will in theory be linear.
This chapter will present how the theory in Chapter 2 was applied during development of the automatic tuning functionality. The first section will present how the AUV dynamics were identified, followed by a description of the control system and the approaches used to tune the controllers. An approach for implementing differential drive will also be suggested. After that, some details regarding the automatic tuning process are given, and the chapter ends with presenting an approach for indirect adaptive control.

### 3.1 System Identification

As stated in Section 2.5.1, motion in only three DOFs will be controlled and therefore it was only needed to identify the dynamics of these DOFs. In the derived model equations in Section 2.3.6, a substantial amount of unknown model parameters are present. In order to reduce the complexity of parameter estimation, additional simplifications were made.

Both nonlinear and linear models were estimated. The linear models were used for model-based control design and auto-tuning of the linear control system currently implemented on Maribot LoLo, and the nonlinear models were used to evaluate an alternative control approach which uses feedback linearization.

As shown in (2.37), the nonlinear models are expressed as functions of the states $n$, $\delta_e$, and $\delta_r$, which values are in turn dependent of the control signals $n_{ref}$, $\delta_{e,ref}$ and $\delta_{r,ref}$. In some systems, the actuator dynamics are significantly faster than the main dynamics, and are therefore neglected. Since the actuators for Maribot LoLo had not been finalized at the time of this thesis, it was not possible to make the assessment whether to neglect actuator dynamics or not. Therefore the
### Table 3.1: Notation for models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Notation</th>
<th>Cross reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation model</td>
<td>( G_o )</td>
<td>Section 2.2</td>
</tr>
<tr>
<td>Simplified nonlinear surge model</td>
<td>( G_{u, nl} )</td>
<td>Eq. (3.6)</td>
</tr>
<tr>
<td>Simplified nonlinear pitch model</td>
<td>( G_{q, nl} )</td>
<td>Eq. (3.9)</td>
</tr>
<tr>
<td>Simplified nonlinear yaw model</td>
<td>( G_{r, nl} )</td>
<td>Eq. (3.12)</td>
</tr>
<tr>
<td>Simplified linear surge model</td>
<td>( G_{u, l} )</td>
<td>Eq. (3.17)</td>
</tr>
<tr>
<td>Simplified linear pitch model</td>
<td>( G_{q, l} )</td>
<td>Eq. (3.18)</td>
</tr>
<tr>
<td>Simplified linear yaw model</td>
<td>( G_{r, l} )</td>
<td>Eq. (3.19)</td>
</tr>
</tbody>
</table>

Actuator dynamics were also estimated.

In order to distinguish the data generated by the simulation model, described in Section 2.2, and the simplified models used to identify the AUV dynamics, the notation in Table 3.1 will be used. The datasets used for system identification were collected from experiments performed in the simulation model, \( G_o \), where each experiment lasted 120 seconds with a sample time of 10 milliseconds, yielding datasets consisting of 12001 samples respectively. As mentioned in Section 2.4.1, the data from each experiment was split into two equally large subsets, where the first half was used for parameter estimation and the second half for model validation.

### 3.1.1 Actuators

Only one of the two rudders and propellers were modelled during simulations since both will have identical dynamics. In practice, it could be beneficial to estimate the dynamics of both rudders/propellers in order the consider small variations. It also enables the possibility to detect and even compensate for wear and damage.

The actuator dynamics were modelled using the linear models in (2.32) and (2.33), and estimated using the LS method presented in Section 2.4.2. In order to obtain the predictors, the models were discretized using the Euler forward method. The discretized model for the propeller dynamics is given by

\[
\dot{n}(k) \approx \frac{n(k+1) - n(k)}{T_s} = -n(k) + \frac{n_{ref}(k)}{\tau_{prop}}
\]

from which the following predictor is obtained

\[
\hat{n}(k + 1) = (1 - \frac{T_s}{\tau_{prop}})n(k) + \frac{T_s}{\tau_{prop}}n_{ref}(k)
\]

By studying (2.43), the following parameter vector \( \theta \) and regression vector \( \varphi(k) \) can be identified

\[
\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^T = \begin{bmatrix} (1 - \frac{T_s}{\tau_{prop}}) \\ \frac{T_s}{\tau_{prop}} \end{bmatrix}^T, \quad \varphi(k) = \begin{bmatrix} n(k-1) \\ n_{ref}(k-1) \end{bmatrix}^T
\]
Using (2.45) then gives

$$f_N = \frac{1}{N} \sum_{k=2}^{N} \left[ n(k-1) \right]^T n(k)$$  \hspace{1cm} (3.4a)

$$R_N = \frac{1}{N} \sum_{k=2}^{N} \left[ n(k-1) \right]^T \left[ n(k-1) \right]$$  \hspace{1cm} (3.4b)

An estimate of $\theta$ was then obtained by (2.44), and the model parameter $\tau_{prop}$ was determined by

$$\hat{\tau}_{prop} = \frac{T_s}{\theta_2}$$  \hspace{1cm} (3.5)

Cross validation between the predicted output of the estimated model and the output generated by the simulation model $G_o$, for a new set of data, is shown in Figure 3.1. It is the case that the simulation model $G_o$ also models the dynamics using a first order system, and therefore an almost perfect fit is obtained. This will not be true in practice, and the result is only used as a validation of the estimation method.

The rudder and elevator parameters were identified in the same way. Cross validation of the elevator dynamics is shown in Figure 3.2. As for the propeller dynamics, the fit is almost perfect since the simulation model also models the elevator dynamics using a second order system. A similar result is obtained for the rudder dynamics.

### 3.1.2 Nonlinear Models

By assuming decoupled motion, i.e. that the AUV will only move in one DOF at a time, most cross terms in (2.37) can be eliminated, except for the dependency of surge velocity on the pitch and yaw dynamics. Even though the AUV will not
perform decoupled motion in practice, the assumption can still be considered as reasonable since it will move at low speeds, making the contribution of cross terms small.

The resulting model for surge dynamics becomes

\[
\dot{u} = \frac{X_u u + X_{|u|u} |u|u + 2X_{prop} |n|n}{m - X_{\dot{u}}}
\]  

which consists of five unknown model parameters. Even if the AUV’s mass was considered as known, the model would still be unidentifiable, since there are only three regressors. In order to gain identifiability, the model was redefined as

\[
\dot{u} = C_1 u + C_2 |u|u + C_3 |n|n
\]  

where

\[
C_1 = \frac{X_u}{m - X_{\dot{u}}}, \quad C_2 = \frac{X_{|u|u}}{m - X_{\dot{u}}}, \quad C_3 = \frac{2X_{prop}}{m - X_{\dot{u}}}
\]  

By discretizing (3.7) a predictor could be formulated as a linear regression and parameter estimation was performed as previously. Only the redefined parameters \( C_1, C_2 \) and \( C_3 \) can be estimated, i.e. it is not possible to uniquely determine values of the physical parameters \( X_u, X_{|u|u}, X_{prop}, m, X_{\dot{u}} \). For the purpose of control, this is not an issue, but if it for some other reason is desired to find values for these parameters, other identification methods are required. Cross validation of the estimated surge model is given in Figure 3.3, where it can be seen that the estimated model is able to accurately describe the dynamics of the simulation model \( G_o \). In practice, the dynamics for forward and reverse surge motion will differ, but since the AUV will mainly be travelling forwards during operation only positive propeller speeds were used during data collection.

**Figure 3.2:** Comparison between elevator deflection angle generated by the simulation model \( G_o \) and by the estimated model in (2.33).
3.1 System Identification

The simplified model for pitch dynamics is

\[ \dot{q} = \frac{M_q q + M_{|q|q}|q| - W z \sin \theta + M_{\delta_e} u^2 \delta_e}{I_y - M_q} \]  \hspace{1cm} (3.9)

which consists of seven unknown model parameters. As for the surge dynamics, the model was redefined in order to gain identifiability

\[ \dot{q} = C_4 q + C_5 |q| q + C_6 \sin \theta + C_7 u^2 \delta_e \]  \hspace{1cm} (3.10)

where

\[ C_4 = \frac{M_q}{I_y - M_q}, \quad C_5 = \frac{M_{|q|q}}{I_y - M_q}, \quad C_6 = \frac{-W z g}{I_y - M_q}, \quad C_7 = \frac{M_{\delta_e}}{I_y - M_q} \]  \hspace{1cm} (3.11)

As before, a predictor could be expressed as a linear regression by discretizing (3.10), and the parameters \( C_4, C_5, C_6 \) and \( C_7 \) were estimated using the LS method. Cross validation is shown in Figure 3.4, where it can be seen that the simplified model \( G_{q,nl} \) describes the simulation model \( G_o \) well.

Finally, the simplified model for yaw dynamics is given by

\[ \dot{r} = \frac{N_r r + N_{|r|} |r| r + 2N_{\delta_r} u^2 \delta_r}{I_z - N_r} \]  \hspace{1cm} (3.12)

which consist of five model parameters. The model was redefined as

\[ \dot{r} = C_8 r + C_9 |r| r + C_{10} u^2 \delta_r \]  \hspace{1cm} (3.13)

where

\[ C_8 = \frac{N_r}{I_z - N_r}, \quad C_9 = \frac{N_{|r|}}{I_z - N_r}, \quad C_{10} = \frac{2N_{\delta_r}}{I_z - N_r} \]  \hspace{1cm} (3.14)

and once again parameter estimation was performed using the LS method. Cross validation is shown in Figure 3.5, where once again a proper model fit can be

Figure 3.3: Comparison between surge velocity generated by the simulation model \( G_o \) and by the estimated nonlinear model \( G_{u,nl} \). The model fit value, computed as in (2.42), is displayed at the top of the figure.
Figure 3.4: Comparison between pitch angular velocity generated by the simulation model $G_o$ and by the estimated nonlinear model $G_{q,nl}$. The model fit value, computed as in (2.42), is displayed at the top of the figure.

Figure 3.5: Comparison between yaw angular velocity generated by the simulation model $G_o$ and by the estimated nonlinear model $G_{r,nl}$. The model fit value, computed as in (2.42), is displayed at the top of the figure.

The estimated model parameters for the nonlinear models are given in Table 3.2.

3.1.3 Linear Models

As stated in Section 3.1, linear models are needed to perform model-based control design of the linear control system. Considering that nonlinear models have been estimated, a simple way of determining linear models was to linearize the nonlinear ones. An alternative approach was to estimate general linear models of corresponding order. A brief evaluation of these two alternatives was carried out for the surge and pitch dynamics.

The linearized models were obtained by formulating the nonlinear models, in-
including actuator dynamics, as nonlinear state space models

\[
\dot{x} = f(x, u) \quad (3.15a)
\]
\[
y = h(x, u) \quad (3.15b)
\]
as in Glad and Ljung (2006), where \( x \in \mathbb{R}^m \), \( u \in \mathbb{R}^n \), and \( y \in \mathbb{R}^p \) are the states, inputs and outputs respectively. By computing a number of Jacobian matrices and selecting an operating point to linearize around, the linearized model is obtained as a linear state space model

\[
\Delta \dot{x} = A \Delta x + B \Delta u \quad (3.16a)
\]
\[
\Delta y = C \Delta x + D \Delta u \quad (3.16b)
\]

where \( \Delta x, \Delta u, \) and \( \Delta y \) are deviations from the chosen linearization point, and \( A, B, C \) and \( D \) are the Jacobian matrices

\[
A = \begin{bmatrix}
\frac{\partial f_1(x,u)}{\partial x_1} & \cdots & \frac{\partial f_1(x,u)}{\partial x_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_m(x,u)}{\partial x_1} & \cdots & \frac{\partial f_m(x,u)}{\partial x_m}
\end{bmatrix}, \\
B = \begin{bmatrix}
\frac{\partial f_1(x,u)}{\partial u_1} & \cdots & \frac{\partial f_1(x,u)}{\partial u_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial f_m(x,u)}{\partial u_1} & \cdots & \frac{\partial f_m(x,u)}{\partial u_n}
\end{bmatrix}, \\
C = \begin{bmatrix}
\frac{\partial h_1(x,u)}{\partial x_1} & \cdots & \frac{\partial h_1(x,u)}{\partial x_m} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_p(x,u)}{\partial x_1} & \cdots & \frac{\partial h_p(x,u)}{\partial x_m}
\end{bmatrix}, \\
D = \begin{bmatrix}
\frac{\partial h_1(x,u)}{\partial u_1} & \cdots & \frac{\partial h_1(x,u)}{\partial u_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial h_p(x,u)}{\partial u_1} & \cdots & \frac{\partial h_p(x,u)}{\partial u_n}
\end{bmatrix}
\]
evaluated at the linearization point.

Linearization of surge and pitch dynamics yields a second and third order system respectively. It would therefore be reasonable to try to identify the dynamics using such linear model structures. The linear model structure for surge becomes

\[
\ddot{u} + C_{11} \dot{u} + C_{12} u = C_{13} n_{ref} \quad (3.17)
\]

and by discretizing (3.17) a linear regression model can be formulated.

### Table 3.2: Estimated model parameters for the nonlinear models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1 )</td>
<td>( 17.68 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>( C_2 )</td>
<td>( -96.03 \cdot 10^{-3} )</td>
</tr>
<tr>
<td>( C_3 )</td>
<td>( 29.74 \cdot 10^{-8} )</td>
</tr>
<tr>
<td>( C_4 )</td>
<td>( -0.40 )</td>
</tr>
<tr>
<td>( C_5 )</td>
<td>( -4.10 )</td>
</tr>
<tr>
<td>( C_6 )</td>
<td>( -0.19 )</td>
</tr>
<tr>
<td>( C_7 )</td>
<td>( 0.22 )</td>
</tr>
<tr>
<td>( C_8 )</td>
<td>( -1.51 )</td>
</tr>
<tr>
<td>( C_9 )</td>
<td>( -1.30 )</td>
</tr>
<tr>
<td>( C_{10} )</td>
<td>( 0.21 )</td>
</tr>
</tbody>
</table>
For control design, it is often desirable to treat models of lower order due to reduced complexity, and therefore the elevator dynamics for pitch motion was assumed to be fast enough to be neglected. A first order linear model

\[ T_1 \dot{q} + q = K_1 \delta_{c,ref} \]  \hspace{1cm} (3.18)

was therefore used for the pitch dynamics which parameters were estimated using the LS method.

In Figure 3.6, cross validation of the two different linear models are shown for surge and pitch, and by comparing the two different approaches no distinct differences could be observed. However, there are some practical differences. To linearize the nonlinear models, suitable linearization points must be chosen while estimating a linear model directly requires no such design choice. Whether this is an advantage or not could be argued both ways. Reducing the amount of design choices increases simplicity while decreasing the possibility to affect the outcome. For example, if it is known in beforehand what speed the AUV mainly will operate at during a certain mission, the user could provide this information and the linearization could be made at the corresponding operating point. For this thesis it was chosen to continue with the approach of estimating linear models directly, instead of linearization of nonlinear models.

In Figure 3.7 cross validation of the linear surge model in (3.17) is shown. The model was estimated using the same set of data as the nonlinear surge model in Figure 3.4. A slightly worse fit value is obtained than for the nonlinear model, but it is still a satisfactory result.

In Figure 3.8, two cross validation plots of the linear pitch model in (3.18) are shown. In Figure 3.8a, the linear model was estimated using the same set of data as the nonlinear pitch model in Figure 3.4. It can be seen that the linear model describes the dynamics poorly, which is believed to be caused by it not being able to consider the restoring moment. The restoring moment will counteract the rotation of the AUV, resulting in a decreased angular velocity as the pitch

\[ \text{Velocity \ [m/s]} \]

\[ \text{Angular Velocity \ [deg/s]} \]

\( \text{Cross validation of surge motion} \)
\( \text{Cross validation of pitch motion} \)

**Figure 3.6:** Comparison of the two approaches for obtaining linear models.
Figure 3.7: Comparison between surge velocity generated by the simulation model $G_o$ and by the estimated linear model $G_{u,l}$. The model fit value is displayed at the top of the figure.

angle grows. This phenomenon can be seen in Figure 3.8a at $t \approx 75$, where a constant, negative elevator deflection is maintained for about seven seconds. If the dynamics had been linear, the angular velocity would have been constantly increasing and eventually converged at a stationary level, which is how the model $G_{q,l}$ (red dashed) acts. But as can be seen by the simulation model $G_o$ (blue), the angular velocity starts to decrease despite no change in input.

Hence, it was concluded that the linear model (3.18) describes the pitch dynamics poorly for slowly varying input signals. In Figure 3.8b, the linear model was estimated using a different set of data where the input signal changes 50% more often. In this case, the pitch angle stays closer to zero and thereby the effect of the restoring moment is reduced. As can be seen, this resulted in a better fit and the model appears to at least capture the main time constant of the dynamics. Whether this is good enough for control design or not, will be evaluated in Section 4.1.

By neglecting the rudder dynamics, a first order linear system could be used to model the yaw dynamics.

$$T_2 \ddot{r} + r = K_2 \delta r,ref$$

(3.19)

In Figure 3.9, cross validation of the linear yaw model is shown. The model was estimated using the same set of data as the nonlinear model in Figure 3.5. It can be seen that the linear model describes the dynamics well and gives only a slightly worse fit value than the nonlinear model.

The estimated model parameters for the linear models are given in Table 3.3.
Figure 3.8: Comparison between pitch angular velocity generated by the simulation model $G_0$ and by the estimated linear model $G_{q,l}$. The model fit value is displayed at the top of the figure.

Figure 3.9: Comparison between yaw angular velocity generated by the simulation model $G_0$ and by the estimated linear model $G_{r,l}$. The model fit value is displayed at the top of the figure.

Table 3.3: Estimated model parameters for the linear models.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{11}$</td>
<td>148.4</td>
</tr>
<tr>
<td>$C_{12}$</td>
<td>16.8</td>
</tr>
<tr>
<td>$C_{13}$</td>
<td>0.034</td>
</tr>
<tr>
<td>$T_1$</td>
<td>2.08</td>
</tr>
<tr>
<td>$K_1$</td>
<td>-0.75</td>
</tr>
<tr>
<td>$T_2$</td>
<td>0.83</td>
</tr>
<tr>
<td>$K_2$</td>
<td>0.27</td>
</tr>
</tbody>
</table>
3.1.4 Estimation Using Instrumental Variables

One case when parameter estimation using the LS method might become inconsistent is when the system output is a regressor and the output is subject to additive measurement noise. The regression vector \( \varphi \) will then be correlated to the noise, causing bias errors in the estimators (Ljung, 1999). All models used in this thesis have been formulated as linear regressions, and in all cases the regression vector contains lags of the output, making the estimation of them susceptible to bias errors. The extent of the error will depend on the color of the noise and on the signal-to-noise ratio. The IV method is presented as an alternative to the LS method, in case the measurement noise is found to be significant in a practical application.

The inconsistency of LS estimation, and the potential of IV estimation, will be demonstrated by considering estimation of the linear surge model \( G_{u,l} \) for different measurement noises, see Figure 3.10. In Figure 3.10a the measurement noise is the same as in Section 3.1.3, i.e. white noise with nominal noise level, and in Figure 3.10b the noise level is increased. The latter case has undoubtedly diminished the estimation outcome.

One approach for implementing IV estimation is to first estimate the model using the LS method and then using the obtained model’s simulated output as instrument. As can be seen in Figure 3.10b, the simulated output of the model, estimated using the LS method, is bad at describing the system output but is highly correlated to it while also being uncorrelated to any noise, making it a suitable choice of instrument. Figure 3.11 shows cross validation of the model estimated using the described IV approach, where an improved model fit has been obtained.

\[ \text{Cross validation of surge motion} - \text{Fit: 78.25} \]
\[ \text{Cross validation of surge motion} - \text{Fit: 40.89} \]

(a) Nominal measurement noise.  
(b) Increased measurement noise.

Figure 3.10: Cross validation of linear surge models, estimated using the least squares method for different types of measurement noise.
Figure 3.11: Cross validation of a linear surge model estimated using the instrumental variables method. The measurement noise is the same as in Figure 3.10b.

3.2 Control System

Two different control structures were evaluated, one based on linear PID controllers and one which uses feedback linearization in conjunction with PID controllers. The former is the one which will be implemented on Maribot LoLo during its initial sea trials, and the latter, which is more advanced, was studied as a potential future improvement. These two control approaches will be referred to as the linear and the nonlinear control system respectively.

Block diagrams of the speed, heading and depth controllers in the linear system are given in Figure 3.12-3.14. The speed controller consists of a simple feedback loop while both the depth and heading controllers utilize cascade control, having both an inner and an outer feedback loop. In the depth controller, the outer loop controls the depth $z$ by setting a reference value for the pitch angle $\theta$ which, in turn, is controlled by the inner loop by setting a reference angle for the elevator $\delta_{e,ref}$. In the heading controller, the outer loop controls the heading/yaw angle $\psi$ by setting a reference value for the yaw angular velocity $r$ which, in turn, is controlled by the inner loop, by setting a reference angle for the rudders $\delta_{r,ref}$.

The PID controllers in the linear control system will be tuned using pole placement based on the linear models estimated in Section 3.1.3. As described in Section 2.5.3, in a cascade controller it is possible to bypass the outer loop and instead control the inner state. It is intended for Maribot LoLo to have this func-

Figure 3.12: Block diagram of the linear surge speed controller.
tionality, which will enable control of both the pitch angle $\theta$ and the yaw angular velocity $r$. Another benefit with having a cascade structure for depth control is the possibility to constrain what pitch angles the AUV will attain during diving. In Maribot LoLo’s case, it is desirable not to dive at too steep angles for safety reasons and therefore the output of the outer PID controller is bounded to $\pm 15^\circ$.

It can be realized that the actuators will often be saturated during operation, and therefore controllers that utilize integral action, i.e. PI and PID controllers, need to address integrator windup to prevent excessive overshoots. One such anti-windup method is conditional integration, which prevents the integral term from changing whenever the control output exceeds its constraints (Glad and Ljung, 2006). This anti-windup method will be implemented for all PI and PID controllers used in the control systems.

Implementation of feedback linearization for the second, more advanced control approach will be described for each controller respectively in the following sub-sections. During initial development, actuator dynamics were neglected in order to simplify the implementation.

### 3.2.1 Speed Controller

Design and tuning of the linear and nonlinear speed controllers will be presented separately below.

**Linear Speed Controller**

The transfer function of the second order model $G_{u,l}$ in (3.17) becomes

$$G(s) = \frac{C_{13}}{s^2 + C_{11}s + C_{12}}$$  

(3.20)
which can be rewritten as the transfer function (2.49), considered by Åström et al. (1993), by choosing

\[ k = \frac{C_{13}}{C_{12}}, \quad T_1 = \frac{C_{11}}{C_{12}} - T_2, \quad T_2 = \frac{C_{11}}{2C_{12}} + \sqrt{\frac{C_{11}^2}{4C_{12}^2} - \frac{1}{C_{12}}} \]  

(3.21)

As presented in Section 2.5.2, a suitable closed-loop characteristic equation is obtained by choosing the control parameters as in (2.56),

\[
K_p = \frac{T_1 T_2 \omega^2(1 + 2\zeta \alpha) - 1}{k} \\
K_i = \frac{T_1 T_2 \alpha \omega^3}{k} \\
K_d = \frac{T_1 T_2 \omega(\alpha + 2\zeta) - T_1 - T_2}{k}
\]

where the design parameters \( \alpha, \omega \) and \( \zeta \) are selected such that satisfactory control is achieved for a nominal case. These design parameters will determine the desired pole placement, i.e. the desired closed-loop surge dynamics, and the idea is that these will only have to be tuned once. When the AUV configuration changes, the model parameters \( C_{11}, C_{12} \) and \( C_{13} \) will change, which in turn will update the control parameters in accordance with (2.56) and the predetermined values for \( \alpha, \omega \) and \( \zeta \). This will in theory result in similar dynamic behaviour regardless of AUV configuration. It should also be mentioned that due to the nonlinear AUV dynamics, the parameters in the linear model will also be affected by which operating point, i.e. which speed, the model is linearized around. In other words, different tunings will be obtained at different speeds.

**Nonlinear Speed Controller**

The nonlinear model describing the surge dynamics is given in (3.7). Since actuator dynamics are neglected (3.7) becomes

\[
\dot{u} = C_1 u + C_2 |u|u + C_3 |n_{ref}|n_{ref}
\]  

(3.22)

and by studying this expression it can be observed that by choosing the control input \( n_{ref} \) as

\[
n_{ref} = \sqrt{\frac{|a_b - C_1 u - C_2 |u||}{C_3} \text{sgn}(a_b - C_1 u - C_2 |u||u)}}
\]  

(3.23)

the surge dynamics will be reduced to the linear system

\[
\dot{u} = a_b
\]  

(3.24)

The resulting linear system is an integrating system, and in theory a P controller is sufficient in order to place the poles of the closed-loop system arbitrarily. In practice it might be necessary to use a PI controller in order to eliminate non-negligible stationary tracking errors due to model imperfections.
3.2 Control System

\[ \sum P \quad \sum a_b \quad \beta \quad \sum g(\beta) \quad AUV \quad \tilde{u} \quad u_{ref} \quad u \]

**Figure 3.15:** Block diagram of the nonlinear surge speed controller.

In Figure 3.15 a block diagram of the controller with feedback linearization is shown, where the functions \( f(u) \) and \( g(\beta) \) are given by

\[
f(u) = C_1 u + C_2 |u| u, \quad g(\beta) = \sqrt{|\beta|} C_3 \text{sgn}(\beta) \quad (3.25)
\]

The \( P \) controller is tuned manually, and the idea is that this controller will not have to be re-tuned since as long as the estimated nonlinear model is accurate enough, the feedback linearization will keep the dynamics approximately equal to (3.24). In other words, the linearized system will remain the same and therefore the \( P \) controller will not have to be updated. Auto-tuning of this controller will instead consist of updating the nonlinear model parameters.

### 3.2.2 Depth Controller

The depth controller implements cascade control as depicted in Figure 3.13. The inner subsystem, denoted 'AUV' in Figure 3.13, represents the AUV dynamics from elevator reference \( \delta_{e,ref} \) to pitch angle \( \theta \). Using the notation in Figure 2.8, these dynamics are given by the transfer function \( G_2(s) \).

The outer subsystem, denoted 'Kinematics' in Figure 3.13, represent the kinematic relation between pitch angle \( \theta \) and the depth \( z \), which can be determined from (2.4) as

\[
\dot{z} = -u \sin \theta + v \cos \theta \sin \phi + w \cos \theta \cos \phi \quad (3.26)
\]

From (3.26), and from basic intuition, it can be realized that the depth will not solely depend on the pitch angle \( \theta \), but also on the linear velocities \( u, v \) and \( w \) as well as the roll angle \( \phi \). If it is assumed that the roll angle and heave velocity will be kept small, the expression can be simplified to

\[
\dot{z} = -u \sin \theta \quad (3.27)
\]

where the depth is only dependent on the pitch angle and the surge velocity.

Design and tuning of the linear and nonlinear depth controllers will be presented separately below.
Linear Depth Controller

Tuning of a cascade controller typically starts by tuning of the inner loop (Åström and Hägglund, 1995). In order to perform model-based control design, the transfer function of the inner subsystem $G_2(s)$ needs to be determined. In Section 3.1.3, a linear model was estimated for the dynamics from elevator reference $\delta_{e,ref}$ to pitch angular velocity $q$. Using the kinematics in Section 2.3.1, the relation between $\theta$ and $q$ can be found as

$$\dot{\theta} = q \cos \phi - r \sin \phi$$  \hspace{1cm} (3.28)

By again assuming that the roll angle $\phi$ will be kept close to zero, (3.28) is simplified to

$$\dot{\theta} = q$$  \hspace{1cm} (3.29)

which gives that the transfer function from $q$ to $\theta$ is

$$G_{q\theta} = \frac{1}{s}$$  \hspace{1cm} (3.30)

By combining (3.18) and (3.30), the transfer function from elevator deflection to pitch angle is found to be

$$G_2(s) = \frac{K_1}{s(sT_1 + 1)}$$  \hspace{1cm} (3.31)

which is a second order system. Unlike the surge dynamics, this transfer function cannot be reformulated in accordance with (2.49). However, by deriving the characteristic equation of the inner loop it was noted that the characteristic equation in (2.51) could still be obtained by selecting the control parameters as

$$K_p = \frac{T_1}{K_1} (2\zeta \alpha + 1) \omega^2$$  \hspace{1cm} (3.32a)

$$K_i = \frac{T_1}{K_1} \alpha \omega^3$$  \hspace{1cm} (3.32b)

$$K_d = \frac{T_1 (2\zeta + \alpha) \omega - 1}{K_1}$$  \hspace{1cm} (3.32c)

In this case, pure PI control is obtained if

$$\alpha_c = \frac{1}{T_1 \omega} - 2\zeta$$  \hspace{1cm} (3.33)

As for the linear speed controller, the design parameters $\alpha$, $\omega$ and $\zeta$ were selected such that satisfactory control was achieved for a nominal case.

Once the inner loop controller was properly tuned, the outer controller was tuned such that the complete controller provides satisfactory control. Currently, the outer controller is a simple P controller, which has been tuned manually. The idea is that the outer controller will not have to be re-tuned since the inner loop dynamics will be tuned such that it will always have the same characteristic equa-
tion. This concept only holds in theory since factors such as nonlinear dynamics and bounded control signals are not considered.

Even in theory it is not completely true that the inner closed-loop dynamics will be the same, since only the pole placement is considered and not the placement of zeros. The zeros will also affect the system dynamics but not to the same extent as the poles. In general, it is desired for the zeros to have negative real parts, just as for the poles. The zeros can be determined by deriving the transfer function of the inner loop

\[
G_{c,2}(s) = \frac{G_2(s)F_2(s)}{1 + G_2(s)F_2(s)} = \frac{K_1 K_d s^2 + K_1 K_p s + K_1 K_i}{T_1 s^3 + (1 + K_1 K_d) s^2 + K_1 K_p s + K_1 K_i}
\] (3.34)

and then computing the roots of the numerator. It can be found that the zeros will indeed at least have negative real parts. This can be realized by studying the signs of the numerator polynomial coefficients. From (3.32) it can be found that the control parameters \(K_p, K_i\) and \(K_d\) will have the same sign as \(K_1\), since \(T_1, \alpha, \omega, \) and \(\zeta\) all are positive. Therefore the coefficients in the numerator polynomial will all be positive, and by using Routh’s stability criterion (Franklin et al., 1994), it can then be shown that the zeros will always have negative real parts. It was therefore considered reasonable to assume that the inner loop dynamics would, in theory, be roughly the same for a specific pole placement.

**Nonlinear Depth Controller**

Neglecting the elevator dynamics in (3.10) gives

\[
\dot{q} = C_4 q + C_5 |q| q + C_6 \sin \theta + C_7 u^2 \delta_{e,ref}
\] (3.35)

and choosing the control input \(\delta_{e,ref}\) as

\[
\delta_{e,ref} = \frac{a_b - C_4 q - C_5 |q| q - C_6 \sin \theta}{C_7 u^2}
\] (3.36)

results in the reduced and linearized system

\[
\dot{q} = a_b
\] (3.37)

Using the same simplified kinematics as in (3.29) gives

\[
\ddot{\theta} = a_b
\] (3.38)

which is a linear, double integrating system. In other words, the feedback linearization has made the inner subsystem \(G_2(s)\) linear.

In Figure 3.16 a block diagram of the inner control loop with feedback linearization is shown, where the functions \(f(\theta, q)\) and \(g(\beta, u)\) are given by

\[
f(\theta, q) = C_4 q + C_5 |q| q + C_6 \sin \theta, \quad g(\beta, u) = \frac{\beta}{C_7 u^2}
\] (3.39)

The inner loop controller was chosen to be a PD controller, which in theory will
allow arbitrary pole placement. The PD controller was manually tuned and the same P controller as in the linear depth controller was used for the outer loop.

### 3.2.3 Heading Controller

The heading controller also implements cascade control as shown in Figure 3.14. In this case, the inner subsystem, denoted ‘AUV’ in Figure 3.14, represents the AUV dynamics from rudder reference $\delta_{r,ref}$ to yaw angular velocity $r$. Using the same notation as for the depth controller, these dynamics are given by the transfer function $G_2(s)$.

The outer subsystem, denoted ‘Kinematics’ in Figure 3.14, represents the kinematic relation between yaw angular velocity $r$ and the heading angle $\psi$, which can be determined from (2.7) as

$$\dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}$$

(3.40)

where it can be seen that the heading angle will also depend on pitch rotational velocity $q$, roll angle $\phi$ and pitch angle $\psi$. However, if it once again is assumed that the roll angle will be close to zero and also that the pitch angle will be kept small, the kinematic relation will solely depend on the yaw angular velocity

$$\dot{\psi} = r$$

(3.41)

Design and tuning of the linear and nonlinear heading controllers will be presented separately below.

**Linear Heading Controller**

As for the depth controller, tuning of the heading controller is commenced by tuning the inner loop. The inner subsystem $G_2(s)$ is obtained by transforming the linear model estimated in Section 3.1.3.

$$G_2(s) = \frac{K_2}{sT_2 + 1}$$

(3.42)
In this case, \( G_2(s) \) is a first order system and therefore a PI controller is sufficient in order to place the closed-loop poles arbitrarily. A suitable characteristic equation is

\[
s^2 + 2\zeta\omega s + \omega^2 = 0 \tag{3.43}
\]

which is similar to (2.51) but with one pole less. (3.43) is obtained by choosing the control parameters as

\[
K_p = \frac{2\zeta\omega T_2 - 1}{K_2} \tag{3.44a}
\]

\[
K_i = \frac{T_2\omega^2}{K_2} \tag{3.44b}
\]

As for the previous controllers, the design parameters \( \omega \) and \( \zeta \) were selected such that satisfactory control was achieved for a nominal case.

The outer loop PID controller is currently a simple P controller, which has been manually tuned such that the complete heading controller provides satisfactory control. And as for the linear depth controller, the idea is that this outer loop controller will not have to be re-tuned. It can be shown that the zeros of the inner closed-loop system will all have negative real parts in this case as well. The inner loop transfer function is given by

\[
G_{c,2} = \frac{K_2K_p s + K_2K_i}{T_2s^2 + (1 + K_2K_p)s + K_2K_i} \tag{3.45}
\]

As for the depth controller, the control parameters \( K_p \) and \( K_i \) will have the same sign as \( K_2 \), and in this case it is straightforward to realize that the zero of \( G_{c,2} \) will always be negative.

### Nonlinear Heading Controller

Neglecting the rudder dynamics in (3.13) gives

\[
\dot{r} = C_8 r + C_8|r|r + C_{10}u^2 \delta_{r,ref} \tag{3.46}
\]

and choosing the control input \( \delta_{r,ref} \) as

\[
\delta_{r,ref} = \frac{a_b - C_8 r - C_9|r|r}{C_{10}u^2} \tag{3.47}
\]

results in the reduced and linearized system

\[
\dot{r} = a_b \tag{3.48}
\]

Using the same simplified kinematics as in (3.41) gives

\[
\ddot{\psi} = a_b \tag{3.49}
\]

In Figure 3.17 a block diagram of the inner control loop with feedback lineariza-
Since the inner subsystem now instead is given by (3.48), a simple P controller would in theory be sufficient. However, simulations showed that a significant stationary tracking error is obtained when using P control, see Figure 3.18a, which was most likely caused by model errors. Therefore a PI controller was used instead. As mentioned in Section 3.2, controllers that use integral action must have anti-windup implemented. In this case, the output of the PI controller is the control signal, $a_b$, to a fabricated system which has no corresponding physical constraint. A slightly modified version of the anti-windup method in Section 3.2 had to be implemented, where instead of checking if the constraints on $a_b$ are exceeded, the integral state will not be updated if the constraints on $δ_{r,\text{ref}}$ are exceeded. Inner loop control using this PI controller is shown in Figure 3.18b, where it can be seen that the stationary tracking error has been successfully reduced.

The outer loop controller in the linear control system was reused for the nonlinear controller.

### 3.2.4 Differential Drive

As mentioned in Section 2.5.1, Maribot LoLo is overactuated in yaw, where yawing motion can be achieved both by using the rudders and by differential drive. A straightforward implementation of a differential drive controller is

$$\Delta n = f(ψ, ψ_{\text{ref}})$$  \hspace{1cm} (3.51)

where $f(ψ, ψ_{\text{ref}})$ is any preferred control strategy and the control output $\Delta n$ is the desired difference between the two propeller speeds. The control signal $n_{\text{ref}}$, computed by the speed controller, will be considered as the desired mean value of the two propeller speeds, and $\Delta n/2$ will be added or subtracted to the control

---

**Figure 3.17:** Block diagram of the nonlinear yaw angular velocity controller, which is the inner loop in the heading controller.

![Block diagram of the nonlinear yaw angular velocity controller](image)
signal sent to the individual propellers, i.e.

\[ n_{1,\text{ref}} = n_{\text{ref}} \pm \Delta n/2 \]  
\[ n_{2,\text{ref}} = n_{\text{ref}} \mp \Delta n/2 \]  

The remaining problem is to figure out how the control system should decide which yawing method to use at which occasions.

The two different methods have complementary properties. At low speeds the rudders are ineffective while at high speeds the propellers should be allocated to provide proper speed control. The obvious choice is to use differential drive at low surge velocities and rudders at high velocities. A simple approach would be to use differential drive while the measured surge velocity is below a certain predetermined limit, and otherwise use the rudders, but this might cause erratic and unpredictable behaviour when the AUV operates close to the specified velocity limit. A better approach is to have a smoother transition between the two methods. Tanakitkorn et al. (2017) have studied a similar control problem but for depth control, and they used weighting functions to transition between the two control strategies based on the surge velocity, see Figure 3.19. The weighting functions are given by

\[ w_1 = 1 - \frac{1}{2} \left( \tanh \left( \frac{u - u^*_1}{\sigma^*_1} \right) + 1 \right) \]  
\[ w_2 = \frac{1}{2} \left( \tanh \left( \frac{u - u^*_2}{\sigma^*_2} \right) + 1 \right) \]  

where \( u^*_1, u^*_2 \) are the surge velocities mid-transition and \( \sigma^*_1, \sigma^*_2 \) are the widths of the transition zones. The weight \( w_1 \) is multiplied with the output of the differential drive controller and \( w_2 \) with the output of the heading controller in Section 3.2.3. A schematic figure illustrating the integration of differential drive to the control system is shown in Figure 3.20, where the function \( g(n_{\text{ref}}, \Delta n) \) is
Figure 3.19: Weighting functions used to transition between using differential drive and rudders.

Figure 3.20: Block diagram illustrating how differential drive is integrated with the control system.

given by (3.52).

The differential drive controller was selected as a P controller, which was tuned manually. Auto-tuning of this controller will not be considered and neither will tuning of the transition functions in (3.53). Parameters which were deemed as reasonable were assigned, and the evaluation in Section 4.3 will focus on the potential of the suggested approach rather than how these parameters should be selected for best performance. The assigned values for the differential drive implementation are presented in Table 3.4. The selected values for the weighting functions yield the curves in Figure 3.19. When approaching the transition zone, it was considered appropriate to transition in the inactive actuator before transitioning out the other, to ensure proper yawing capabilities throughout the transition.

3.3 Automatic Tuning Process

The different stages of the suggested automatic tuning method have been presented separately in this chapter, and some remaining details concerning imple-
3.3 Automatic Tuning Process

Table 3.4: Parameter values for the differential drive controller and weighting functions.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>100</td>
</tr>
<tr>
<td>$u^*_{1}$</td>
<td>0.7</td>
</tr>
<tr>
<td>$\sigma^*_{1}$</td>
<td>0.03</td>
</tr>
<tr>
<td>$u^*_{2}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\sigma^*_{2}$</td>
<td>0.03</td>
</tr>
</tbody>
</table>

For the linear control system, the final phase is performed using pole placement and for the nonlinear control system the final phase simply consists of updating the model parameters in the feedback linearization.

In the first stage, the experiments will excite the AUV in one DOF at a time using random telegraph signals. A typical experiment sequence is presented in Figure 3.21. After the experiments the collected data is separated into three data sets corresponding each DOF. Either all of the data is used for parameter estimation, or some data is set aside for model validation. In the latter case, the AUV can provide an operator with model fit values, who can decide whether to perform

![Figure 3.21: Example of signals sent to the actuators during an experiment.](image)
the experiments again or not. It is important to point out that during excitation of
the pitch and yaw dynamics, the surge velocity will be basically constant mean-
ing that the linear models will be adapted for this specific velocity. Also, the
AUV needs to be completely submerged, which is assumed to be accomplished by
using the VBS.

It should also be mentioned that the AUV will need an open area to perform the
experiments in. If the experiment in Figure 3.21 is performed, the AUV will travel
as shown in Figure 3.22. In this case, the experiment lasts six minutes and the
AUV travels approximately 500 meters from its initial position. In practice, the
duration of the experiments might need to be longer thus requiring more space to roam in. Safety features such as geo-fencing and collision avoidance can be used
to interrupt the experiment to prevent the AUV from damaging itself during these
open-loop maneuvers.

The model parameters are estimated using the LS method, or alternatively the IV
method, as described in Section 3.1. Lastly, the controllers in the linear control
system are tuned using the expressions in (2.56), (3.32) and (3.44) respectively.
The design parameters $\alpha$, $\zeta$, $\omega$ in each expression represent the desired closed-
loop dynamics via pole placement and these will only have to be set once. Once
selected, the expressions (2.56), (3.32) and (3.44) will provide an explicit mapping
between the model parameters and the controller parameters.

3.4 Adaptive Control

The developed automatic tuning process is executed prior to a mission in order
to compensate for dynamic changes introduced between missions. However, the
dynamics might also change during operation, for example if some payload is

\[ x, y, z \] = [0, 0, 10]^T.

\textbf{Figure 3.22:} Position of the vehicle, expressed in the inertial coordinate
system, when performing the experiment in Figure 3.21. The initial position

(a) $xy$-plane.  

(b) Depth.
either picked up or dropped by the AUV. Even if the AUV has functionality to detect changes in its control performance in real-time, it would be inappropriate to re-execute the auto-tuning during a mission since it is unknown whether the current surroundings provide enough space to perform the experiments. It would also consume energy, resulting in reduced range and operating time.

An alternative approach would be to continuously adjust the control system during operation, known as adaptive control. Adaptive control techniques are often classified as either direct or indirect adaptive control, where in a direct method the control parameters are adjusted directly based on system observations while in an indirect method the observations are used to determine models of the system which in turn are used to determine the control parameters (Åström et al., 1993). Since model-based control design already has been developed, an indirect adaptive control approach would be to continue to estimate the models in real-time, and continuously adjust the control parameters the same way as in the automatic tuning.

3.4.1 Indirect Adaptive Control Approach

The suggested indirect adaptive control approach consists of two parts; online parameter estimation and adjustment of the control system based on the current model parameters. Most of the latter part has already been developed, which leaves the former part.

As explained in Section 2.4.4, recursive IV was considered more suitable than RLS for online, closed-loop parameter estimation. Gilson and Van den Hof (2005) present and compare several IV methods for closed-loop identification, and they explain that in a basic IV method the instruments are typically chosen as delayed samples of the reference signal. This would allow for simple implementation without pre-filtering and Gilson and Van den Hof (2005) state that it will also allow the controller to be nonlinear and/or time-varying which was considered as a substantial benefit considering the adaptive application.

Only a limited amount of time was available to develop and test the suggested adaptive control approach, and therefore implementation of online identification has only been implemented for the linear and nonlinear surge models. The recursive algorithm presented in Section 2.4.4 was used where the regression vectors for the linear and nonlinear models are

\[
\varphi_l = [u(k-1), u(k-2), n_{ref}(k-2)]^T
\]

\[
\varphi_{nl} = [u(k-1), |u(k-1)|u(k-1), n(k-1)]^T
\]

and the instruments were chosen as

\[
\zeta_l = \zeta_{nl} = [u_{ref}(k-2), u_{ref}(k-3), u_{ref}(k-4)]^T
\]

From intuition, this choice of instruments should be appropriate since the reference signals are uncorrelated with noise, and given that the control system works as intended the output should be correlated to its reference. The control signal should also have some correlation with the reference since an increase in refer-
ence will yield an increase in control signal and vice versa. Depending on the sample rate, it might be possible that higher correlation can be obtained for further delayed references signals, but this was not considered. The initial estimate $\theta_0$ will be assigned as the estimate obtained by the auto-tuning method, and $P_0$ will be chosen as an identity matrix multiplied with a factor $\rho$.

There are some additional aspects to consider for the online identification which have not been studied in this thesis, mainly regarding when it is appropriate to actually update the model. Firstly, the surge models have been derived under the assumption that the propellers are controlled identically and therefore it is not suitable to update them when using differential drive. Furthermore, better estimation results should be expected if the parameters are only updated while the AUV is not pitching or yawing. And finally, as for any parameter estimation, informative data are required and it might therefore not be appropriate to update the parameters when the AUV is performing monotone maneuvers such as cruising at constant speed. These aspects could potentially be considered by implementing weighted recursive IV, where low weights are placed on data which should not be used for updating the model parameters.

The remaining problem is to determine when and how the control parameters should be adjusted. Updating the control parameters at each sample time will probably yield erratic and unpredictable behaviour unless some kind of filtering is implemented. Other alternatives might be to update the control parameters at certain time intervals or only when some sort of performance index reaches a specific value. Unfortunately, there was not enough time to neither carry out a literature study regarding this issue nor test any of the suggested approaches. Results will only be presented for the online identification, and not for the adaptive control.
This chapter presents the main outcome of the thesis. In the first section, the performance of the linear and nonlinear control systems are evaluated and compared. The controllers have been tuned based on the models estimated in Section 3.1, which are considered as models of the nominal vehicle configuration. In the following section, the vehicle configuration is altered and it is investigated if the developed automatic tuning approach yields improved control performance. Evaluation of the differential drive implementation is treated in a separate section since no auto-tuning functionality was developed for this controller.

Since the AUV dynamics are nonlinear, it is important to evaluate the control performance for a wider range of operation scenarios. Therefore the reference signals have been chosen as step signals with different amplitudes for most of the tests. No measurement noise or process disturbances were applied during testing of the controllers, in order to emphasize differences caused by altering configurations and controller tuning, rather than studying the controllers’ robustness to noise and disturbances.

Since system identification was a prerequisite for model-based control design, results regarding modelling of the AUV dynamics have already been presented in Section 3.1. However, this does not include evaluation of the online identification method which will also be presented in this chapter.
4.1 Motion Control

In the following subsections, the speed, depth and heading controllers will be evaluated separately. All three controllers are active during testing, but reference tracking will only be displayed for the DOF currently evaluated, in order to reduce the amount of signals shown and thereby make the results more clear.

The performance of the controllers will be based on properties such as stationary tracking error, oscillations and over- and undershoots, as defined by Glad and Ljung (2006).

4.1.1 Speed Control

Using the estimated model parameters $C_{11}$, $C_{12}$ and $C_{13}$ in Table 3.3, a suitable tuning for the linear speed controller was obtained by choosing the following design parameters

$$\omega = 0.35, \quad \zeta = 1, \quad \alpha = 1.1\alpha_c$$

where $\alpha_c$ is given by (2.53). The choice of alpha ensures that $\alpha > \alpha_c$. The resulting PID tuning becomes

$$K_p = 2867, \quad K_i = 588, \quad K_d = 436$$

In Figure 4.1 the performance of the linear speed controller is given. In the left subfigure it can be seen that the speed follows the reference smoothly, without overshoot or stationary tracking error. In the right subfigure the propeller speed and its constraints can be seen.

A suitable tuning of the $P$ controller in the nonlinear control system was found to

![Figure 4.1: Simulation result where step signals are sent as reference to the linear speed controller. The plots show the surge velocity and propeller speed, respectively. The reference values for the depth and heading controllers were kept constant.](image)
be $K_p = 2$. In Figure 4.2 the linear speed controller is compared to the nonlinear one. A slight improvement can be observed for the nonlinear controller when operating at high speeds. Considering that the shape of the transient is very similar for all step sizes and during both acceleration and deceleration, it appears that the feedback linearization has successfully linearized the dynamics. Also, the fact that no significant stationary error can be observed indicates that the model error is small.

### 4.1.2 Depth Control

For the linear depth controller, a suitable tuning of the inner PID controller was obtained using the model parameters $K_1$ and $T_1$ in Table 3.2 and the design parameters

$$\omega = 1.6, \quad \zeta = 0.8, \quad \alpha = \text{max}\{0.3, \alpha_c\}$$

where $\alpha_c$ is given by (3.33). The resulting tuning of the inner PID controller becomes

$$K_p = -11.3, \quad K_i = -3.66, \quad K_d = -7.49$$

The control parameters become negative since positive elevator deflection is defined to be in the same direction as positive pitch. From physical insight it can be realized that a positive elevator deflection will generate negative pitch motion, which is represented by a negative value of $K_1$. Therefore the signs of the obtained control parameters are sensible.

A suitable tuning of the outer $P$ controller was found to be $K_p = -10$. A negative gain is required since in order to dive, a negative pitch angle is necessary. The performance of the linear depth controller is shown in Figure 4.3, where it can be seen that good reference tracking is obtained, without overshoot or stationary tracking error. The right subfigure shows the control performance of the inner loop, which controls the pitch angle $\theta$. In Section 2.5 it was mentioned that $\theta$
should not exceed $\pm 15^\circ$, and therefore the reference value $\theta_{ref}$ does not exceed $\pm 15^\circ$ as seen in Figure 4.3b. The test was conducted with the speed controller being set to maintain a constant speed of 1.3 m/s.

In the nonlinear depth controller, the PD controller in the inner loop was tuned to $K_p = 0.15$, $K_d = 0.1$, and the P controller in the outer was set to $K_p = -10$ as in the linear controller. In Figure 4.4 the linear and nonlinear depth controllers are compared, but in this case no noticeable difference can be observed.

However, by comparing the control performance of the inner loop, i.e. control of the pitch angle $\theta$, some differences can be observed, see Figure 4.5. Linear control causes minor overshoots which vary slightly for different step sizes, indi-
4.1 Motion Control

cating nonlinear behaviour. With nonlinear control, the shape of the transient looks the same for all steps, which suggests that the feedback linearization works as intended. Since the pitch dynamics are significantly faster than the diving dynamics, these small differences will not be noticeable during depth control.

Another important control aspect to consider is that the pitch dynamics are dependent on the surge velocity. The linear controller does not consider the surge velocity, while the nonlinear controller does. In Figure 4.6, the same test as previously is conducted but for a surge velocity of 0.8 and 1.8 m/s instead. As expected the nonlinear controller is able to cope with changes in speed better, but the linear controller still provides acceptable control. The fact that neither controller is able to achieve 15° pitch in Figure 4.6a is due to the elevator not being able to generate enough lift to overcome the restoring moment.

Figure 4.5: Comparison of the linear and nonlinear pitch controller.

Figure 4.6: Evaluation of the linear and nonlinear pitch controllers’ robustness to changes in surge velocity.

(a) $u_{ref} = 0.8$ m/s.

(b) $u_{ref} = 1.8$ m/s.
4.1.3 Heading Control

For the linear heading controller, a suitable tuning of the inner PI controller was obtained using the model parameters $K_2$, $T_2$ in Table 3.2 and the design parameters

$$\omega = 3, \quad \zeta = 1$$

which yielded the following control parameters

$$K_p = 15.1, \quad K_i = 28.2$$

A suitable tuning of the outer P controller was found to be $K_p = 1$. The performance of the resulting linear heading controller is presented in Figure 4.7, where once again no overshoots or stationary tracking errors are observed. By studying the control signal it can be seen that the rudders are fully deflected while turning, which is a sensible behaviour. The test was conducted with the speed controller set to maintain a constant speed of 1.3 m/s.

The PI controller in the inner loop of the nonlinear controller was set to $K_p = 5$, $K_i = 8$, and the outer loop P controller was set to $K_p = 1$ as in the linear controller. The linear and nonlinear heading controllers are compared in Figure 4.8, but no distinct differences can be seen.

In Figure 4.9, the control performance of the inner loop, i.e. control of the yaw angular velocity, is shown. Even in this case, no apparent differences can be observed, which indicates that the yaw dynamics are fairly linear. This observation is in accordance with the similar model fit values obtained for the nonlinear and linear models in Figure 3.5 and 3.9. However, as for the pitch dynamics, the yaw dynamics are also dependent on the surge velocity, meaning that the dynamics

Figure 4.7: Simulation result where step signals are sent as reference to the linear heading controller. The left plot shows the heading and the right plot shows the yaw angular velocity and the rudder deflection angle.
will only appear linear as long as the surge velocity is kept constant.

As for the depth control, unlike the nonlinear controller, the linear controller does not consider the surge velocity. In Figure 4.10, the surge velocity is changed to 0.8 and 1.8 m/s respectively, and the same observations as for the depth controller are made.

### 4.2 Automatic Tuning

The automatic tuning approach was evaluated using a number of test cases, where in each case an alteration to the AUV configuration was made. For each test case, three different simulations were typically made. One for the nominal configuration where the AUV dynamics are modelled by the parameters in Table 3.2-3.3, and the controllers are tuned as presented in Section 4.1. One for the altered configuration, with the controllers still being tuned for the nominal configuration. Finally, one for the altered configuration where the control system has been

---

**Figure 4.8:** Comparison of the linear and nonlinear heading controller.

**Figure 4.9:** Comparison of the linear and nonlinear yaw angular velocity controller.
auto-tuned for the new AUV configuration.

The auto-tuning performance of both the linear and the nonlinear control systems will be tested.

### 4.2.1 Case 1 - Change in Mass

In the first case, an increase in mass of the AUV was studied, which will be the most typical scenario in practice. An additional 300 kg was added to the nominal mass of 600 kg. It was assumed that the mass was equally distributed and that neutral buoyancy was maintained. The speed, depth and heading controllers are evaluated separately below.

#### Speed Control

In Figure 4.11, the performance of the linear speed controller is given, and in Table 4.1 the PID parameters obtained by the auto-tuning are presented.

The first observation is that the difference between the first and second simulation is quite small, even though the mass has been increased by 50 %, which either means that the dynamics have not been affected that much by the change in mass or that the nominal controller is robust enough to handle the change in dynamics.

The second observation is that there is basically no difference between the second and third simulation, meaning that no further improvement was obtained by using the auto-tuning. In Table 4.1, it can be seen that the parameters actually have been adjusted, and that the adjustments made are sensible considering an increase in mass. By studying Figure 4.11b it can be seen that the propeller speed becomes saturated even for the case with low mass and low speed. This means that even if the updated controller tries to thrust harder in order to compensate
4.2 Automatic Tuning

Figure 4.11: Evaluation of linear speed control performance for an increase in mass.

Table 4.1: Speed controller parameters obtained by auto-tuning.

<table>
<thead>
<tr>
<th>Mass</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>2867</td>
<td>588</td>
<td>436</td>
</tr>
<tr>
<td>900</td>
<td>4260</td>
<td>830</td>
<td>616</td>
</tr>
</tbody>
</table>

Figure 4.12: Evaluation of nonlinear speed control performance for an increase in mass.

for the increased mass, the outcome will not change due the limited thrust available.

In Figure 4.12 the performance of the nonlinear speed controller has been tested for an increased mass. The same test scenarios as for the linear controller were conducted, and the same observations are also made.

In Figure 4.13 the nominal nonlinear and linear models are cross validated with
Figure 4.13: Cross validation of the nominal nonlinear and linear surge models, estimated in Section 3.1, against data collected when the AUV mass has been increased.

Data collected when the AUV mass was 900 kg. This test was conducted in order to determine how much the vehicle dynamics changes when the mass is increased. In the left subfigure it can be seen that the fit value has decreased noticeably compared to Figure 3.3, while in the right subfigure the fit value has actually increased compared to Figure 3.7. The increased fit of the linear model will be considered as a coincidence.

In the left subfigure it can be seen that the red dashed curve has a faster transient than the blue curve, which is reasonable since the blue curve corresponds to the heavier vehicle. Based on this analysis it is found that the change in mass does affect the surge dynamics, but not to a significant extent, which might be an explanation to why auto-tuning did not result in any improvements.

Depth Control

The performance of the linear and nonlinear depth controllers are given in Figure 4.14, and in Table 4.2 the PID parameters obtained by auto-tuning the linear controller are presented.

The same two observations as for the speed control were made, i.e. that the increase in mass does not have a significant effect on the control performance and that auto-tuning will not provide improved control. These observations were also found when evaluating pitch control, see Figure 4.15.

In Figure 4.16, the nominal linear and nonlinear models of the pitch dynamics are cross validated with data from the AUV with mass 900 kg. Figure 4.16a shows that the nominal nonlinear model still gives good fit, except at $t \approx 75s$ which was found to be caused by the restoring moment having an increased influence due to the added weight. This implies that the dynamics has not been significantly affected by the increased mass, and might explain why the control performance
### 4.2 Automatic Tuning

#### Linear depth control.

![Linear depth control graph](image)

(a) Linear depth control.

#### Nonlinear depth control.

![Nonlinear depth control graph](image)

(b) Nonlinear depth control.

**Figure 4.14:** Evaluation of linear and nonlinear depth control performance for an increase in mass.

#### Linear pitch control.

![Linear pitch control graph](image)

(a) Linear pitch control.

#### Nonlinear pitch control.

![Nonlinear pitch control graph](image)

(b) Nonlinear pitch control.

**Figure 4.15:** Evaluation of linear and nonlinear pitch control performance for an increase in mass.

### Table 4.2: Depth controller parameters obtained by auto-tuning.

<table>
<thead>
<tr>
<th>Mass</th>
<th>Inner loop</th>
<th>Outer loop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K_p$</td>
<td>$K_i$</td>
</tr>
<tr>
<td>600</td>
<td>-11.3</td>
<td>-3.66</td>
</tr>
<tr>
<td>900</td>
<td>-15.2</td>
<td>-4.94</td>
</tr>
</tbody>
</table>
is unaffected.

**Heading Control**

The same results as for the speed and depth controllers were also found for the heading control, i.e. that the increased mass does not affect the control performance. In this case it is only necessary to present the results where the nominal yaw models are cross validated against data from the heavier AUV, see Figure 4.17. The figure shows that the nominal models yield only marginally lower fit values, which indicates that the dynamics are almost unaffected, and in turn the control performance will also not be affected by the increased mass.

**Figure 4.16:** Cross validation of the nominal nonlinear and linear pitch models, estimated in Section 3.1, against data collected when the AUV mass has been increased.

**Figure 4.17:** Cross validation of the nominal nonlinear and linear yaw models, estimated in Section 3.1, against data collected when the AUV mass has been increased.
Considering that most of the additional weight put on Maribot LoLo will be placed in the payload area in the front of the vehicle, different results may be found in practice. Instead of assuming that the mass will be equally distributed, more accurate results would be obtained by considering the added weight as a point mass placed in the payload area, but no such tests were made. An equally distributed mass of 300 kg was considered plentiful in order to evaluate the impact of adding mass to the AUV.

### 4.2.2 Case 2 - Change in Mass and No Propeller Saturation

In the second test case, the aim was to find a vehicle alteration which had a more distinct impact on the speed control. From the previous test case it was observed that the limited propeller speed prevented the controller from compensating for the increased inertia. For the sake of evaluating the potential of the developed auto-tuning approach, the saturation on the propeller speed was removed and the AUV mass was set to 1200 kg. The results for the linear speed controller are given in Figure 4.18 and Table 4.3. The nominal controller causes overshoots for the altered vehicle configuration. When using the auto-tuned controller, the performance is improved for the initial step, but worsened for the final step.

The performance of the nonlinear speed controller is shown in Figure 4.19, where it can be seen that the outcome of the first and last simulation are almost identical, suggesting that auto-tuning of the nonlinear controller is working as intended.

**Figure 4.18:** Evaluation of linear speed control performance for an increase in mass and no constraints on propeller speed.

**Table 4.3:** Speed controller parameters obtained by auto-tuning.

<table>
<thead>
<tr>
<th>Mass</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>2867</td>
<td>588</td>
<td>436</td>
</tr>
<tr>
<td>1200</td>
<td>5680</td>
<td>1077</td>
<td>800</td>
</tr>
</tbody>
</table>
4.2.3 Case 3 - Change in Elevator Lift

The third test case was designed to evaluate the auto-tuning performance of the pitch controller. Impacts on the control performance are more noticeable by studying the pitch controller rather than the depth controller, due to the pitch dynamics being faster. Further, a more interesting scenario to consider is when the AUV is altered such that the dynamics become faster rather than slower since this should induce overshoots and oscillations. In this test case, the nominal vehicle was altered to generate more lift on the elevator while also having a reduced added mass parameter $M_q$. The increased lift was achieved by increasing the lift coefficient in the simulation model but could also be achieved by increasing the dimensions of the elevator. A reduced value of $M_q$ can be achieved by e.g. using a hull which is not complete flat on the upper and lower side or by reducing the area of these sides. The results for the linear pitch controller are given in Figure 4.20, where increased overshoots are observed when the nominal controller is used. By using auto-tuning, the overshoots have been reduced and the performance is similar to the nominal case, indicating that the auto-tuning approach using pole placement is functioning as intended. The control parameters obtained by auto-tuning are presented in Table 4.4.

Figure 4.21 shows the performance of the nonlinear pitch controller. In this case, the altered configuration causes persistent oscillation using the nominal controller, which are eliminated when the controller is auto-tuned.

Table 4.4: Pitch controller parameters obtained by auto-tuning.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>$K_p$</th>
<th>$K_i$</th>
<th>$K_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>-11.3</td>
<td>-3.66</td>
<td>-7.49</td>
</tr>
<tr>
<td>Altered</td>
<td>-3.57</td>
<td>-1.16</td>
<td>-1.47</td>
</tr>
</tbody>
</table>
4.2 Automatic Tuning

Figure 4.20: Evaluation of linear pitch control performance for an increased elevator lift and decreased $M_q$.

Figure 4.21: Evaluation of nonlinear pitch control performance for an increased elevator lift and decreased $M_q$.

4.2.4 Case 4 - Change in Rudder Lift

The fourth test case was designed to evaluate the auto-tuning performance of the yaw angular velocity controller. Corresponding alterations as in the previous test case where made for the yaw dynamics, i.e. increased rudder lift and decreased added mass $N_p$. The performance of the linear controller is presented in Figure 4.22 and the control parameters in Table 4.5. The simulations show both overshoots and oscillations for the altered vehicle when using the nominal tuning, and that the oscillations are eliminated when that controller is auto-tuned. The same observation is made for the nonlinear controller, see Figure 4.23.
Figure 4.22: Evaluation of linear yaw angular velocity control performance for an increased rudder lift and decreased $N_f$.

Table 4.5: Parameters for the yaw angular velocity controller obtained by auto-tuning.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>$K_p$</th>
<th>$K_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>15.1</td>
<td>28.2</td>
</tr>
<tr>
<td>Altered</td>
<td>4.59</td>
<td>10.8</td>
</tr>
</tbody>
</table>

Figure 4.23: Evaluation of nonlinear yaw angular velocity control performance for an increased rudder lift and decreased $N_f$. 
4.3 Differential Drive

The suggested approach for implementing differential drive was evaluated by performing steps in heading at different surge velocities. The following four test cases were simulated:

1. $u_{ref} = 0$ - Turning while at standstill.
2. $u_{ref} = 0.3$ - Turning while moving, but without using rudders.
3. $u_{ref} = 0.5$ - Beginning of transition zone.
4. $u_{ref} = 0.7$ - End of transition zone.

The outcome of the simulations are given in Figure 4.24-4.27, which show reference tracking for both the heading and the speed controller, but also the weights and actuator states. Figure 4.24 and 4.25 show that the AUV is able to turn solely using differential drive, both while at standstill and at low speed, which will expand the viable operating region of the AUV. The heading reference tracking in Figure 4.24 has some overshoots which can possibly be reduced by further controller tuning. On the other hand, the overshoots are noticeably different depending on the step size, indicating that the nonlinear dynamics might be difficult to deal with using linear controllers.

In Figure 4.26-4.27, differential drive and rudders are used simultaneously to control the heading. A significant improvement in heading control is obtained in the third case compared to the second case. Some slight oscillations can be seen in Figure 4.26a, which is due to the weight for the rudders being mid-transition. The differential drive will disturb the speed control, causing $u_2$ to fluctuate which in turn causes the rudders to waver. In the final case, differential drive is transitioning out, and this time no oscillations in heading are observed since using the rudder does not affect the speed control as much as using differential drive.

![Reference tracking](image1.png)  ![Weights and actuator states](image2.png)

(a) Reference tracking.  (b) Weights and actuator states.

**Figure 4.24:** Simulation results of the first differential drive test.
Figure 4.25: Simulation results of the second differential drive test.

Figure 4.26: Simulation results of the third differential drive test.

Figure 4.27: Simulation results of the fourth differential drive test.
4.4 Online Identification

As stated in Section 3.4.1, the first step towards indirect adaptive control is to estimate the model parameters online. The performance of the implemented recursive IV estimator was evaluated by trying to identify the nominal surge dynamics, i.e. the same dynamics which were identified offline in Section 3.1. Instead of having the initial estimate $\theta_0$ equal to the values obtained offline, the offline values were slightly modified to emulate the scenario where the AUV dynamics during auto-tuning and during mission are different. It was then investigated whether the online estimation could identify the nominal dynamics and how the online estimate compares to the offline estimate. To obtain a quantitative measure of the quality of the current online estimate, a model fit value was computed at each sample time. Note that computing fit values online will typically not be possible in a practical application since validation data for the unknown dynamics will not be available, but when conducting tests in simulation such data is available.

The tests were meant to investigate the potential of the suggested online identification method, and therefore the AUV was commanded to perform maneuvers which would yield proper excitation of the surge dynamics rather than maneuvers which are representative to an actual mission. The controller tunings were fixed during the tests.

The results for online estimation of the linear surge model are given in Figure 4.28-4.29. The initial values were chosen as

$$\theta_0 = [C_{11}^{LS}, C_{12}^{LS}, 0.6C_{13}^{LS}], \quad P_0 = 0.01I_{3\times3}$$

where $C_{11}^{LS}, C_{12}^{LS}, C_{13}^{LS}$ are the offline estimates given in Table 3.3. The progression of the model fit can be seen in Figure 4.28a, where the red dash-dot line shows the fit value for the initial estimate and the black dashed line shows the fit value obtained by offline estimation. It can be seen that the online estimation will approach a similar fit value as for the offline estimation, and has converged after approximately 50 seconds. The performed speed maneuvers are shown in Figure 4.28b, where it can be seen that during the first 50 seconds about 7-8 accelerations and decelerations are performed, which gives some indication of how much excitation is required for the online estimation. The progression of the estimates are given in Figure 4.29 where it can be seen that the online estimation will yield similar estimates as the offline method.

The initial values for the nonlinear model were chosen as

$$\theta_0 = [0.6C_1^{LS}, 0.5C_2^{LS}, C_3^{LS}], \quad P_0 = I_{3\times3}$$

where $C_1^{LS}, C_2^{LS}, C_3^{LS}$ are the offline estimates given in Table 3.2. Corresponding figures for the nonlinear case are given in Figure 4.30-4.31. Once again, the online method is able to significantly improve the model fit, but unlike the linear model it converges to a value slightly lower than obtained offline. In Figure 4.31, it can be seen that the online estimates will not converge to its offline counterparts. An interesting observation is that the sign of $C_1$ has changed compared to
Figure 4.28: Simulation results for online parameter estimation of the linear surge model. The left figure shows the model fit at each sample time (blue) relative to the initial (red) and offline (black) fit value. The right figure shows the measured surge speed.

Figure 4.29: Progression of the online estimates for the linear surge model. The red line represents the modified initial value and the black lines represent the offline estimate values.

the value estimated offline, where a negative value is actually more reasonable since $C_1$ corresponds to linear damping (divided by an inertia term), see (3.8). The fact that the linear damping has increased, i.e. become more negative, might explain why the quadratic damping $C_2$ has decreased. It is therefore difficult to argue which estimates are more correct, since the offline estimates yield higher model fit to validation data while the online estimates are more in line with physical insight.
Figure 4.30: Simulation results for online parameter estimation of the nonlinear surge model. The left figure shows the model fit at each sample time (blue) relative to the initial (red) and offline (black) fit value. The right figure shows the measured surge speed.

Figure 4.31: Progression of the online estimates for the nonlinear surge model. The red lines represent the modified initial values and the black lines represent the offline estimate values.
This chapter presents a discussion of the obtained results and of the choice of method. The chapter and thesis is concluded by reflecting back to the purpose and objectives of the study, and by suggesting areas for future development.

5.1 Discussion

Results

Based on the simulation results in Section 4.1 it can be argued that the linear and nonlinear control systems provide almost identical control performance. However, the nonlinear strategy has been shown to yield slightly better control in certain cases. The nonlinear approach of using feedback linearization seems to work as intended since the dynamic response remains the same regardless of step size, whereas the linear controllers yield varying performance, as expected. The nonlinear AUV dynamics are expected to be more apparent for vehicles travelling at higher velocities, and therefore such vehicles would probably benefit more from the nonlinear control approach.

Changes in vehicle mass will likely be the most common alteration to the AUV configuration, and such changes were expected to have more noticeable impact on the control performance. The simulation results in Section 4.2.1 suggests that Maribot LoLo will not require automatic tuning functionality, given that solely changes in mass are made. It is believed that this might be due to the slow AUV dynamics, caused by low operating speeds and large inertia, which allows for the nominal control system to have high robustness. However, since the simulation model $G_0$ has not been validated with data from the actual AUV, there is an unknown amount of uncertainty in the vehicle parameters assigned in the simu-
Discussion and Conclusion

Discussion and Conclusion

The results in Section 4.2.2-4.2.4 are deemed as satisfactory and are in accordance with the desired outcome. The alterations made in these sections are still representative scenarios for a multipurpose platform AUV such as Maribot LoLo, where motors, propellers and control surfaces might be replaced for specific missions.

The brief evaluation of the heading controller which uses both rudders and differential drive showed promising results. Considering that only a simple $P$ controller was tested for the differential drive and that minimal effort was spent on tuning the transition functions, the performance of the heading controller is probably below its full capability. With a more advanced differential drive controller and further tuning of the transition functions, it is believed that the performance can be improved significantly.

Based on the results of the online parameter estimation, it appears that recursive IV will be able to provide similar estimates as in the offline estimation, despite that the AUV is operating in closed loop. The results also confirm that choosing the reference signals as instruments is a viable alternative both when using linear and nonlinear control, as suggested by Gilson and Van den Hof (2005). It is however not evaluated how well this choice of instruments will perform when trying to identify the pitch and yaw dynamics, since in these control loops the reference signals are for depth and heading respectively, not for the angular velocities. In these cases the regression vectors will not be as well correlated with the reference signals as in the surge case, and equally good results can therefore not be expected.

Method

A model-based approach was chosen for the study. Naturally, the performance of the controllers will depend on the quality of the estimated models, and therefore an extensive part of the thesis was dedicated to system identification. It is believed that the presented linear and nonlinear models are suitable for the application due to their ability to describe the main dynamics using few model parameters. One of the key assumptions made when deriving the models was that the dynamics are decoupled, and based on the satisfactory estimation results in Section 3.1 it can be argued that this assumption is justified, at least for decoupled motion. It has not been evaluated how well the models will describe more complex, coupled maneuvers. The coupled dynamics are likely to have a greater significance for AUVs travelling at higher speeds.

A related issue is that if predetermined model structures are used, the autotuning will not be able to consider changes in unmodelled dynamics. For example, if external equipment is mounted on the AUV an unmodelled hydrodynamic...
moment might be generated which would likely worsen the parameter estimation and thereby result in an improper controller tuning. For such applications the set of considered model structures could potentially be extended. However, by extending the set of linear models to consider coupled dynamics, they will typically no longer be single-input single-output (SISO) models and therefore a control system consisting of PID controllers is no longer as suitable. However, PID control should still be viable, but if the models are not SISO models, pole placement might not applicable as tuning method for the PID parameters. A better option might then be to formulate the models as state space models and use e.g. LQ controllers instead. Extending the set of nonlinear models should, on the other hand, not require a different control strategy since it is believed to be fairly straightforward to extend the feedback linearization to compensate for additional dynamic effects. A non-model-based auto-tuning approach might be more suitable for applications where coupled dynamics are likely to be introduced with the alterations to the AUV.

The main idea with the pole placement approach was that the AUV should behave roughly the same regardless of configuration, which in hindsight was found to be unrealistic due to the limited input amplitude available. A better approach might be to somehow also consider the constraints on the control signals, yielding a tuning which is optimal based on the current vehicle dynamics and the available input magnitudes.
5.2 Conclusions

Reflecting back to the objectives defined in Section 1.3, it can be concluded that the modelling framework by Fossen (1994) provided proper model structures for identifying the dynamics of the 6 DOF AUV simulation model considered in the study. Both nonlinear and linear models have been presented for surge, pitch and yaw dynamics respectively, all of which are able to describe the dynamics well enough for reliable control design. Two different methods for estimating model parameters have been suggested, LS and IV, where IV has been presented as an alternative method in case unmodelled disturbances are found to be significant in a practical application.

Due to the nonlinear AUV dynamics, the experiments had to be designed to excite a wide range of operating points and therefore signals that switch to random amplitudes at random occasions were deemed as appropriate input. The obtained results indicate that such experiments will generate useful and informative estimation data.

Two different control systems were considered, one consisting of solely PID controllers and one which utilizes feedback linearization in conjunction with PID controllers. Both control systems implemented decoupled control for forward speed, depth and heading. Simulation results show that both control systems are able to provide smooth and well damped motion, at least while only controlling one DOF at a time. The former control system is auto-tuned using pole placement while the latter is auto-tuned simply by updating the model parameters in the feedback linearization.

The main purpose of the thesis was to develop a method for automatic tuning of the control system of an autonomous underwater vehicle. The simulation results show that an improved control performance is obtained by using the suggested auto-tuning method. These preliminary results demonstrate the potential of the method and they are deemed to be adequate enough to justify implementation and further testing on an actual AUV. The main purpose of the thesis is therefore considered to be achieved.

However, due to the limited amount of time available within the confines of a master thesis, it was not possible to investigate whether improved control could be achieved by continuously adjusting the control system during operation. The first steps towards an indirect adaptive control approach have been taken, but further development is required before any statements regarding its potential are made.
5.3 Future Work

Since the presented results have been obtained solely using simulations, further testing and evaluation on a real AUV is necessary to draw any further conclusions regarding the potential performance of the suggested automatic tuning. In a real application, estimates of the linear and angular velocities will be provided by a navigation system implementing sensor fusion, and it will be essential to evaluate how well the system identification will perform when using these state estimates as regressors. In addition, the AUV will also be subjected to environmental disturbances such as waves and currents, which has not been properly evaluated in this study. It would be interesting to evaluate how robust the suggested parameter estimation methods are to such disturbances, but also how the control performance is affected.

Further effort is required to evaluate whether improved control performance can be achieved by continuing to adjust the control system during operation. The suggested indirect adaptive control approach requires further development and testing. The two main components, i.e. online model estimation and model-based control design, have been demonstrated to work separately but it remains to test how they will function together. Some further aspects to consider for this approach have been mentioned in Section 3.4.1 and 5.1. Alternative approaches such as direct adaptive control and reinforcement learning might also be interesting candidates.

As mentioned in Section 2.5, the nonlinear control system was studied as a potential future improvement to the linear control system, the reason being that linear control will likely yield varying control performance in different operating regions. Another approach to handle the nonlinear dynamics is to estimate several linear models for different forward speeds, combining them to a linear parameter varying (LPV) model as in Eng et al. (2016), and then implement gain scheduling. Using this approach, the same linear control system can be used where instead of having constant control parameters, the control parameters are adjusted depending on the forward speed.


