NTNU – Trondheim
Norwegian University of
Science and Technology

Development of a Dynamic Positioning System for Merlin WR 200 ROV

Development and implementation of a model based control law for station keeping and path following

Sveinung Johan Ohrem

Master of Science in Cybernetics and Robotics
Submission date: June 2015
Supervisor: Kristin Ytterstad Pettersen, ITK
Co-supervisor: Ments Tore Møller, IKM Subsea AS
Sverre Wendt Slettebø, IKM Subsea AS

Norwegian University of Science and Technology
Department of Engineering Cybernetics

NTNU
Norges teknisk-naturvitenskapelige
universitet

Fakultet for informasjonsteknologi,
matematikk og elektroteknikk
Institutt for teknisk kybernetikk

HOVEDOPPGAVE

Kandidatens navn: Sveinung Ohrem

Fag: Teknisk kybernetikk

Oppgavens tittel (norsk): Utvikling av et dynamisk posisjoneringssystem for Merlin WR200 ROV

Oppgavens tittel (engelsk): Development of a Dynamic Positioning System for Merlin WR200 ROV

Oppgavens tekst:

Control of unmanned vehicle-manipulator systems (UVMS) including both ROVs and Autonomous Underwater Vehicles (AUV) with manipulator arms, require full-DOF control of the vehicle. In particular, it is necessary to be able to keep the vehicle at a constant position and orientation, in order for the manipulator to efficiently perform its task. The vehicles considered in this project are therefore fully actuated ROV and/or AUV. Survey AUV with torpedo-shape and no sway actuation are thus not considered.

In particular, the project will consider the Merlin WR200, which is a Class 3 work class-ROV with fully electric propulsion. Today it only has auto functions for altitude, depth and direction. In order to increase the precision and make marine operations close to the sea floor more efficient (depth < 100 meters), it is desirable to equip the vehicle with local dynamic positioning capabilities (station keeping). (Local in the sence that there is no available global reference system, and the station keeping will thus be relative to the sea floor).

This project is part of addressing the main challenge of increasing the level of autonomy and robustness for automatic mapping, monitoring and intervention, high-level planning/re-planning and reconfiguration of single and multiple vehicles subject to the particular mission, environmental condition, available energy, communication constraints, and any failure conditions.

The Centre of Excellence Autonomous Marine Operations and Systems (AMOS) is dedicated to address these control challenges. The center has high expertise and cutting edge experimental facilities to solve the corresponding control problems. This MSc project will be integrated in the AMOS research, with the corresponding access to cutting edge expertise and experimental facilities.

The main goals of the thesis is the development and implementation of a new model based controller and the implementation of a path following algorithm for underwater vehicles.
The following subtasks are proposed for this project:

1. Develop a model-based controller for path following/tracking control of ROVs. Dynamic positioning should also be considered as a special case. Case study: Merlin WR 200.
2. Develop a motion planner based on path followed by the ROV when controlled by the operator.
3. Validate the derived controller and motion planner through simulations.
4. Compare the performance of the derived controller with existing controllers studied in the pre-project.
5. Test and verify the proposed controller through simulator experiments in cooperation with IKM Subsea AS.

Oppgaven gitt: 05.01.2015

Besvarelsen leveres: 01.06.2015

Medveileder: Ments Tore Møller/Sverre Wendt Slettebø, IKM Subsea AS.

Utført ved Institutt for teknisk kybernetikk

Trondheim, den 05.01.2015

Kristin Y. Pettersen

Faglærer

Preface

This thesis is submitted as the final requirement for the degree of Master of Science at the Department of Engineering Cybernetics at the Norwegian University of Science and Technology (NTNU). The work was carried out from January to June of 2015.

The work has been done in collaboration with IKM Subsea which provided 2 supervisors and data for the Merlin WR200 ROV. They also provided the use of their state-of-the-art simulator for testing.

Trondheim, June 1, 2015

Sveinung Johan Ohrem

Acknowledgements

I am very happy that I was given the opportunity to work on this project for my master thesis. Throughout my education I have always been fond of control theory and controller design, so this thesis was a prefect fit in that sense. The last 5 months have been a mixture of fun, hard work, happiness and frustration, but all in all they have been very educational. I have learned a lot and had a really good time!

I would like to thank my supervisor Prof. Kristin Y. Pettersen for giving me good advice and guidance throughout the semester and Ments Tore Møller and all the other very nice people at IKM Subsea for taking me on as a summer intern and for good advice on the thesis. Huge thanks to my classmates and to my office mates Thomas, Stine and Bjørn-Olav for good advice, fruitful discussions and for all the good times shared.

This thesis would not exist if it weren't for coffee and music, so I would like to thank whoever invented coffee and music. This thesis marks the end of my education and with that the end of my period working at Café Ni Muser. I would like to thank everyone I have worked with at the café over the past 7 years for your friendship and for all the good times!

Lastly, I would like to thank my family, my friends and especially my fiancé Synnøve for always believing in me and supporting me.

- S. J. O

Abstract

Control of Remotely Operated Vehicles (ROVs) with manipulator arms require control of all degrees of freedom. In particular, it is necessary to be able to keep the vehicle at a constant position and orientation in order for the manipulator arms to efficiently perform its task. To increase the level of autonomy of a ROV, station keeping, trajectory tracking and path following can be implemented. These functions all require a controller and the development of such a controller is the key task in this thesis. The design of a path following algorithm for backtracking the path driven by the ROV when controlled by the operator is also one of the goals.

When designing a control system for a marine craft, the choice of controller is not obvious. Several controllers exist with different properties and demands. A desired feature of a controller is that it is robust to parameter uncertainty, since the parameters of a marine craft may change with different operating conditions. For an underwater vehicle, the ocean current is present as a disturbance, so the controller needs to be able to compensate this disturbance. In this thesis a position and velocity controller that adapts the system parameters and current disturbance is developed. Convergence of the position error and virtual velocity error to zero is proven through Lyapunov theory and by utilizing Barbălat's lemma.

The developed controller has been implemented and tested on a Simulink simulation model of the Merlin WR200 ROV from IKM Subsea. The tests considered station keeping, trajectory tracking, path following of paths generated by the ROV as it drives through the ocean space and of predefined paths. For all cases, the ROV is able to converge to the desired position, velocity or path. The performance of the developed controller and 2 other controllers is compared for station keeping and trajectory tracking and the developed controller shows very good results. The controller was also tested for station keeping and trajectory tracking in the advanced simulator from CM Labs at the IKM headquarters in Bryne. The performance was very good under the conditions of the simulation.

Future work includes among other things developing a more extensive dynamic model for Merlin WR200 that includes the effects of cable drag, developing a steering law for 6 DOF underwater vehicles and conducting a sea trial.

The goal of this thesis was to develop a new model based control law for ROVs and verify its performance through simulations. This has been successfully accomplished.

Sammendrag

Kontroll av fjernstyrte undervannsfartøy (ROV) med manipulatorarmer krever kontroll i alle frihetsgrader. Det er spesielt viktig at fartøyet klarer å holde en konstant posisjon og orientasjon slik at arbeidet som utføres med manipulatorarmene blir utført så effektivt som mulig. For å utvide automasjonsgraden til en ROV kan man implementere posisjonsregulering og både tidsavhengig og tidsuavhengig banefølging. Disse funksjonene krever en regulator og utviklingen av en slik regulator er hovedmålet for oppgaven. Utviklingen av en banefølgingsalgoritme som gjør det mulig å følge banen ROVen har kjørt når den ble styrt av en operatør, er også et av målene.

Valget av regulator til et marint styresystem er ikke opplagt. Flere regulatorer eksisterer og alle har forskjellige egenskaper og krav. En ønsket egenskap med en regulator er at den skal være robust mot parameterusikkerhet siden parameterne til et marint fartøy kan forandre seg med ulike operasjonstilstander. Et undervannsfartøy vil være påvirket av havstrøm, så det er viktig at regulatoren klarer å kompansere forstyrrelsen havstrømmen forårsaker. I denne oppgaven utvikles det en posisjons,- og hastighetsregulator som adapterer systemparametrene og strømningsforstyrrelsen. Konvergens av posisjonsfeilen og hastighetsfeilen til null bevises ved hjelp av Lyapunovteori og ved anvendelse av Barbălats lemma.

Den utviklede regulatoren ble implementert og testet i en Simulink simuleringsmodell av Merlin WR200 ROV fra IKM Subsea. Posisjonsregulering og banefølging av baner generert av ROVen når den kjører gjennom havrommet og forhåndsdefinerte baner ble testet. ROVen konvergerer til den ønskede posisjonen, hastigheten eller banen i alle tilfeller. Prestasjonen til den utviklede regulatoren og to andre regulatorer sammenlignes for posisjonsreguleringen og den utviklede regulatoren viser meget gode resultater. Regulatoren ble også testet for posisjonsregulering i den avanserte simulatoren fra CM Labs på IKMs hovedkvarter i Bryne. Prestasjonen var under forholdene meget bra.

Som fremtidig arbeid foreslåes blant annet utviklingen av en mer omfattende dynamisk modell av Merlin WR200 som inkluderer påvirkningen fra kabelen, utvikling av en styrelov for undervannsfartøy i 6 frihetsgrader og utførelse av sjøtest.

Hovedmålet med oppgaven var å utvikle en ny modellbasert regulator for ROVer og verifisere prestasjonen gjennom simuleringer. Dette ble utført.

”A mind needs books like a sword needs a whetstone, if it is to keep its edge.”

- George R. R. Martin

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Control of unmanned vehicle-manipulator systems (UVMS) including remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) with manipulator arms requires control in all 6 degrees of freedom (DOF). When performing operations with the manipulators it is necessary to keep the vehicle at a constant position and attitude. An underwater vehicle is only exposed to environmental disturbances caused by current since it is operating at a depth where the wave and especially the wind effects can be ignored. The control system needs to be able to handle this current disturbance.

To achieve the control objective and handle the disturbances, the ROV is equipped with a dynamic positioning (DP) system. The design of a DP system is not straightforward, and one of the challenges is the choice of controller. Several controllers with different properties and requirements exist for this purpose and the choice is not obvious. The controller needs to be robust to parameter uncertainty, be able to handle disturbances and at the same time respond to changes in the reference position or velocity in a satisfying way.

The Merlin WR200 is currently only fitted with automatic control functions of altitude, depth and heading, so it is of interest to increase the level of autonomy to include position control in the horizontal plane as well, enabling the ROV to keep its position (station keeping). This increased autonomy would also lead to the ROV being able to perform path following missions.

The controller must make the ROV able to both hold the position and follow a path. It should also be easily implementable, meaning it should not rely on perfect knowledge of system parameters. The derivation of the controller should be as simple as possible.

The goal of this thesis is therefore the development of a new model based control law based on the mentioned criteria. The control law should be tested in a dynamic positioning setting and in a path following mission through simulations.

1.2 Contributions

In this thesis a new model based control law for marine vehicles in 6 DOFs has been developed. It is attempted to write the derivation of the new controller in a more readable and simplified way by using a diffeomorphism. Emphasis was placed on deriving a control law that is robust to parameter uncertainty and is able to handle current disturbances.

A control system for the Merlin WR200 has been developed utilizing the new controller. For the DP part of the control system, reference models and thruster allocation algorithms have been implemented and the DP system has been tested in the simulator at IKM Subsea for verification. A path following system based on the already driven path of the ROV has also been developed. The path following system uses a modified version of the Line of Sight steering law to ensure correct heading and velocity.

1.3 Problem description

The overall problem in this thesis is increasing the level of autonomy and robustness for underwater vehicles in general and the Merlin WR200 in particular. The following subtasks are proposed for this thesis:

1 Develop a model-based controller for path following/tracking control of ROVs. Dynamic positioning should also be considered as a special case. Case study: Merlin WR200.

2 Develop a motion planner based on path followed by the ROV when controlled by the operator.

3 Validate the derived controller and motion planner through simulations.

4 Compare the performance of the derived controller with existing controllers studied in the pre-project.

5 Test and verify the proposed controller in simulator experiments in cooperation with IKM Subsea AS.

1.4 Thesis outline

Chapter 2 defines the notation used in the thesis. Background information on modeling of marine vehicles and ocean currents is presented along with an introduction to marine vehicle guidance, navigation and control systems. Previous work done on the DP system of Merlin WR200 is presented along with limitations and available measurements. A brief overview of the mathematical definitions used in this thesis is given.

Chapter 3 is the modeling chapter. The mass, damping and Coriolis matrices for the Merlin WR200 are described and the process plant and control plant models are presented.

Chapter 4 presents the DP and path following control system. The control law is derived and stability is proven. A Line of Sight path following control system is derived for the return path mode. As an additional feature, a path following system for predefined paths is presented.

Chapter 5 presents the Simulink simulation system that has been developed.

Chapter 6 contains the results from the Simulink simulations and from the simulation done in the advanced simulator at IKM Subsea. The results are discussed consecutively.

Chapter 7 contains suggestions for future work and concluding remarks.

1.5 Abbreviations

Abbreviation	Description
AUV	Autonomous Underwater Vehicle
ABSwB	Adaptive Backstepping with Bound estimation
CAD	Computer Aided Design
CB	Center of Buoyancy
CG	Center of Gravity
CO	Center of Origin
DOF	Degree of Freedom
DP	Dynamic Positioning
DVL	Doppler Velocity Log
EOM	Equations of Motion
GES	Global Exponential Stability
GNC	Guidance Navigation and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HMI	Human Machine Interface
LOS	Line of Sight
NED	North-East-Down
P-CABS	Parameter and Current Adaptive Backstepping
PE	Persistently Exciting
PID	Proportional, Integral, Derivative
PLC	Programmable Logic Controller
RMS	Root Mean Square
ROV	Remotely Operated Vehicle
SF	Serret-Frenet
TMS	Tether Management System
UVMS	Unmanned Vehicle Manipulator System
VS-MRAC	Variable Structure - Model Reference Adaptive Control
WP	Waypoint

1.6 Greek letters

Letter	**Description**
α	Angle between N and LOS vector
β	Sideslip angle
γ	Arbitrary constant between 0 and 1
$\mathbf{\Gamma}$	Gain matrix
δ	Bound on disturbance
Δ	Lookahead distance, tuning parameter, ”change in”
$\boldsymbol{\epsilon}$	Coordinates of vessel in the path fixed frame
ζ	Damping ratio
$\boldsymbol{\eta}$	Position and attitude vector
θ	Pitch angle
$\boldsymbol{\theta}$	Parameter vector
Θ	Attitude vector
κ	Curvature of path
λ	Eigenvalue
$\boldsymbol{\nu}$	Velocity vector
ξ	Parametrized path
π	Pi
ρ	Density of seawater
τ	Time delay
$\boldsymbol{\tau}$	Force vector
ϕ	Roll angle
$\boldsymbol{\phi}$	Regressor matrix
χ	Course angle
ψ	Heading angle
ω	Path variable or natural frequency

The following accents are used in the thesis:

Accent	**Example**	**Description**
Tilde	$\tilde{\boldsymbol{\eta}}$	Error between measured/actual value and the desired value
Bar	$\bar{\boldsymbol{e}}_2$	A modified version of the unbarred variable
Hat	$\hat{\boldsymbol{\theta}}$	An estimated variable
Dot	$\dot{\boldsymbol{\nu}}$	The time derivative of the variable

Chapter 2

Theory

2.1 Reference frames

The positions, attitudes and motions of a marine craft are expressed in different reference frames to best describe the behaviour. The reference frames used in this thesis are as explained in [Fossen, 2011]:

- NED: The *North-East-Down* frame, $\{n\} = \{N,\ E,\ D\}$, with origin o_n, is considered inertial in this thesis. This is the coordinate system we refer to in our everyday life. The x-axis points towards true North, the y-axis towards East and the z-axis points down, nominal to the Earth's surface. The NED frame can be seen as a tangent plane to the earth's surface (figure 2.1). This assumption holds for local operations which is the case for the ROV operations in this thesis. The positions and attitudes of a vehicle are given in the NED-frame. From here on out, the NED frame is denoted $\{n\}$
- BODY: The *BODY* frame, $\{b\} = \{x_b, y_b, z_b\}$, with origin o_b, is fixed on the vehicle and thus it is a moving coordinate frame. The axes of the BODY frame are chosen to coincide with the principal axes of inertia (figure 2.2). The position and orientation of the vehicle is given as the position and orientation of the BODY reference frame relative to the $\{n\}$ frame. The linear and angular velocities however are given in the BODY reference frame. The origin o_b is usually chosen as a point in the middle of the vessel. This point is referred to as CO. From here on out, the BODY frame is denoted $\{b\}$

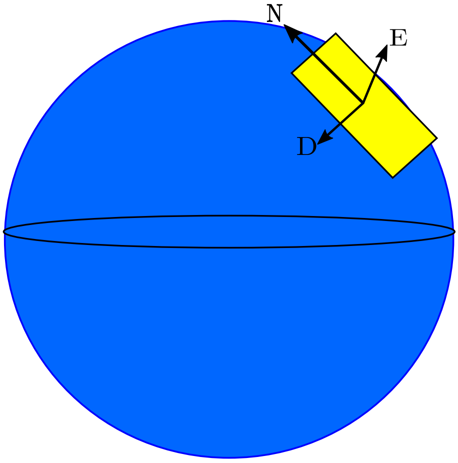


Figure 2.1: The $\{n\}$ frame.

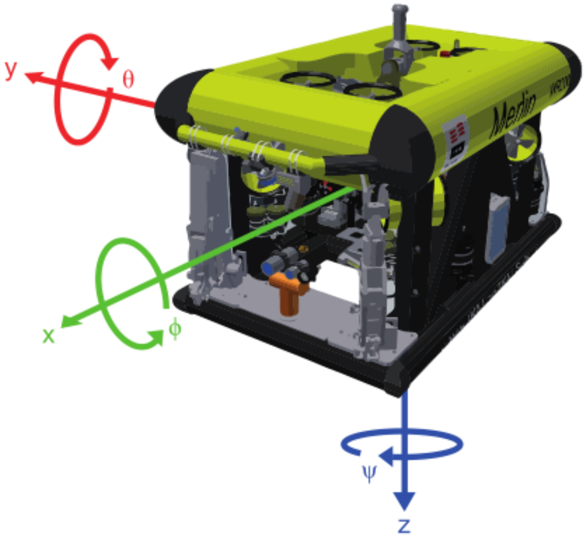


Figure 2.2: The 6 DOFs of Merlin WR200 in the $\{b\}$ frame. Image courtesy of [Knausgård, 2013].

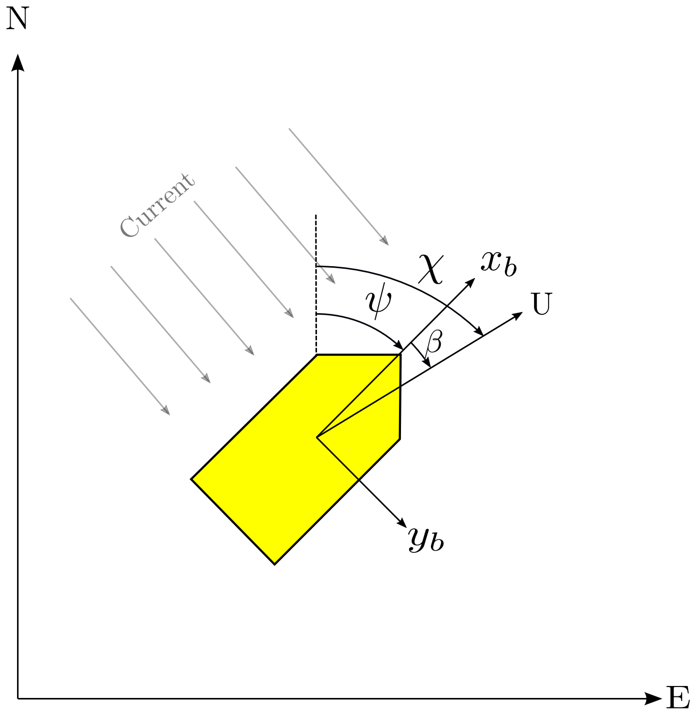


Figure 2.3: The $\{b\}$ frame of the ROV as it is seen in $\{n\}$. $\{n\}$ is constant and $\{b\}$ moves through it. ψ is the heading angle of the ROV, χ is the course angle and β is the sideslip angle caused by ocean currents, U is the speed of the ROV

2.1.1 Notation

For a marine craft operating on 6 DOF, 6 different coordinates are necessary to describe the position and motion. These 6 coordinates are defined by the Society of Naval Architects and Marine Engineers(SNAME) and can be seen in table 2.1 [SNAME, 1950].

		Forces and		Positions and
DOF	Description	moments	Velocities	Euler angles
1	motions in the x direction(surge)	X	u	x
2	motions in the y direction(sway)	Y	v	y
3	motions in the z direction(heave)	Z	w	z
4	rotation about the x axis(roll)	K	p	ϕ
5	rotation about the y axis(pitch)	M	q	θ
6	rotation about the z axis(yaw)	N	r	ψ

Table 2.1: Formulations for the different degrees of freedom of a marine vessel as defined by SNAME in 1950

In this thesis, vectors and matrices are expressed in **bold**. Sub,- and superscripts are used as follows: $\boldsymbol{v}^b_{b/n}$ and is read as *the linear velocity of the BODY frame with respect to the NED frame, expressed in the BODY frame.* Angular representations are $\boldsymbol{\Theta}_{nb}$ and is read as *the Euler angles between NED and BODY.* Since the position and velocity of the marine craft is described in two different coordinate frames, it is appropriate to separate them into two vectors. The vectors contains the states of the marine craft in the appropriate reference frame. See table 2.2.

Vector	Vector	State	Description	Force/Moment
$\boldsymbol{\eta}^n_{b/n}$	$\boldsymbol{p}^n_{b/n}$	N	North position	X
		E	East position	Y
		D	Down position	Z
	$\boldsymbol{\Theta}_{nb}$	ϕ	Attitude about x axis	K
		θ	Attitude about y axis	M
		ψ	Attitude about z axis	N
$\boldsymbol{\nu}^b_{b/n}$	$\boldsymbol{v}^b_{b/n}$	u	Surge velocity	X
		v	Sway velocity	Y
		w	Heave velocity	Z
	$\boldsymbol{w}^b_{b/n}$	p	Roll rate	K
		q	Pitch rate	M
		r	Yaw rate	N

Table 2.2: The position and attitude vector $\boldsymbol{\eta}$, the velocity vector $\boldsymbol{\nu}$ and the forces and moments that influence motion in the state

2.1.2 Definitions of course, heading and sideslip angles

The relationship between the angular variables *course*, *heading* and *sideslip* is important for manoeuvring of a marine craft in $\{n\}$. First, the speed of a marine craft is defined as [Fossen, 2011]:

$$U := \sqrt{u^2 + v^2} \tag{2.1.1}$$

The course, heading and sideslip angle can then be defined as [Fossen, 2011]:

Definition 2.1: *The **course angle** χ is the angle from the x_n axis of $\{n\}$ to the velocity vector U of the craft. Positive rotation about the z_n axis of $\{n\}$ by the right-hand screw convention.*

Definition 2.2: *The **heading angle** ψ is the angle from the x_n axis of $\{n\}$ to the x_b axis of $\{b\}$. Positive rotatoin about the z_n axis of $\{n\}$ by the right-hand screw convention.*

Definition 2.3: *The **sideslip angle** β is the angle from the x_b axis of $\{b\}$ to the velocity vector of the vehicle. Positive rotation about the z_b axis of $\{b\}$ by the right-hand screw convention.*

From this it is clear that

$$\chi = \psi + \beta \tag{2.1.2}$$

2.2 Transformations

The $\{n\}$ and $\{b\}$ frame are related through a transformation matrix $\boldsymbol{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})$. This matrix consists of the linear velocity rotation matrix $\boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})$ and the angular velocity transformation matrix $\boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb})$

$$\boldsymbol{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb}) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) \end{bmatrix} \tag{2.2.1}$$

where the linear velocity transformation matrix is

$$\boldsymbol{R}_b^n(\boldsymbol{\Theta}) = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \tag{2.2.2}$$

$\mathrm{s}(\cdot) = \sin(\cdot)$, $\mathrm{c}(\cdot) = \cos(\cdot)$. The *Rotation Matrix* $\boldsymbol{R}_b^n(\boldsymbol{\Theta}) \in \mathbb{R}^{3\times3}$ is an element in $SO(3)$ which is the *special orthogonal group of order 3.* A matrix $\boldsymbol{R}$ in $SO(3)$

has the properties $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}$, $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$, $\det \boldsymbol{R} = 1$ [Egeland and Gravdahl, 2002]

The angular velocity transformation matrix is

$$\boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix} \tag{2.2.3}$$

$\mathrm{t}(\cdot) = \tan(\cdot)$. Notice that (2.2.3) is undefined for $\theta = 90°$. One way around this singularity is to use quaternion representation of the angles. However, this is not done in this thesis because the roll and pitch angle of the Merlin WR200 is limited to $\theta = \pm 20°$. If the roll and pitch angles can be assumed ($\phi = \theta = 0$), the angular velocity transformation matrix reduces to

$$\boldsymbol{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) = \boldsymbol{I}_{3\times3} \tag{2.2.4}$$

and a transformation between $\{n\}$ and $\{b\}$ can be defined as:

$$\boldsymbol{P}(\psi) := \begin{bmatrix} \boldsymbol{R}_b^n(\psi) & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{2.2.5a}$$

$$\boldsymbol{P}^T(\psi) = \boldsymbol{P}^{-1}(\psi) \tag{2.2.5b}$$

2.3 Mathematical modelling of underwater vehicles

The position and attitude of the marine craft is expressed in $\{n\}$. The derivative of the position, or the velocity in $\{n\}$ can be expressed as:

$$\dot{\boldsymbol{p}}_{b/n}^n = \boldsymbol{v}_{b/n}^n = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})\boldsymbol{v}_{b/n}^b \tag{2.3.1a}$$

$$\dot{\boldsymbol{\Theta}}_{nb} = \boldsymbol{w}_{b/n}^n = \boldsymbol{T}_{\Theta}(\boldsymbol{\Theta}_{nb})\boldsymbol{w}_{b/n}^b \tag{2.3.1b}$$

Collecting these terms gives rise to the 6 DOF kinematic model:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.3.2}$$

The kinetics, or the forces that are causing the motions, are modelled after the vectorial notation of [Fossen, 1991]:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{2.3.3}$$

where

- $\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A > 0, \in \mathbb{R}^{6\times6}$ is the system inertia matrix. It is the sum of the rigid body system inertia matrix $\boldsymbol{M}_{RB} = \boldsymbol{M}_{RB}^T > 0$ and the hydrodynamic system inertia matrix or added mass matrix $\boldsymbol{M}_A = \boldsymbol{M}_A^T \geq 0$.

- $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}) \in \mathbb{R}^{6\times6}$ is the Coriolis matrix. It is the sum of the rigid body Coriolis matrix $\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = -\boldsymbol{C}_{RB}^T(\boldsymbol{\nu}),\ \forall\, \boldsymbol{\nu} \in \mathbb{R}^6$ and the hydrodynamic Coriolis matrix $\boldsymbol{C}_A(\boldsymbol{\nu}) = -\boldsymbol{C}_A^T(\boldsymbol{\nu}),\ \forall\, \boldsymbol{\nu} \in \mathbb{R}^6$. This matrix can always be parametrized such that it is skew-symmetric. This is useful since the quadratic form $\boldsymbol{\nu}^T\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} \equiv 0$.
- $\boldsymbol{D}(\boldsymbol{\nu}) = \boldsymbol{D}_l + \boldsymbol{D}_{nl}(\boldsymbol{\nu}) > 0 \in \mathbb{R}^{6\times6}$ is the damping matrix. It is the sum of the linear damping matrix $\boldsymbol{D}_l$ and the nonlinear damping matrix $\boldsymbol{D}_{nl}(\boldsymbol{\nu})$. Linear damping is caused by potential damping that arises when a body is forced to oscillate and by skin friction. The nonlinear damping is caused by quadratic damping and higher order terms. The damping matrix has the properties of being real, asymmetric and strictly positive, that is $\boldsymbol{D}(\boldsymbol{\nu}) > 0\,\forall\, \boldsymbol{\nu} \in \mathbb{R}^6$.
- $\boldsymbol{g}(\boldsymbol{\eta}) \in \mathbb{R}^6$ is the vector of generalized gravitational and buoyancy forces.
- $\boldsymbol{\tau} \in \mathbb{R}^6$ is the vector of generalized control forces acting on the different DOFs.
- $\boldsymbol{\tau}_e \in \mathbb{R}^6$ is the vector of environmental forces. For an underwater vehicle, this represents the ocean current forces.

The kinematics and kinetics gives us the equations of motion(EOM) for a marine craft in 6DOFs expressed in $\{b\}$:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{2.3.4a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_e \tag{2.3.4b}$$

2.3.1 Ocean current modeling

Ocean currents are described in [Fossen, 2011] as:

*"Horizontal and vertical circulation systems of ocean waters produced by gravity, wind friction and water density variations in different parts of the ocean."*4

The ocean current itself has a velocity and a direction in $\{n\}$ and to model the effects of the current on a marine craft, [Fossen, 2011] proposes to use the *relative velocity vector*:

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c \tag{2.3.5}$$

where $\boldsymbol{\nu}_c$ is the generalized ocean current velocity of an irrotational fluid:

$$\boldsymbol{\nu}_c = \begin{bmatrix} u_c & v_c & w_c & 0 & 0 & 0 \end{bmatrix}^T \tag{2.3.6}$$

with $\boldsymbol{v}_c^b = \begin{bmatrix} u_c & v_c & w_c \end{bmatrix}^T$ as the linear current velocity expressed in $\{b\}$. The ocean current can, according to [Fossen, 2011], be assumed as slowly varying or constant such that its derivative in $\{n\}$, $\dot{\boldsymbol{v}}_c^n \approx 0$ and consequently $\dot{\boldsymbol{v}}_c^b = -\boldsymbol{S}(\boldsymbol{w}_{b/n}^b)\boldsymbol{v}_c^b$.

If it is assumed that the current is the only environmental force acting on the marine craft, the kinetics can be expressed as equations of the relative velocity:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{2.3.7}$$

2-D irrotational current model

The current is modelled in 2-dimensions with a speed V_c and a sideslip angle β_c. The current in $\{n\}$ is expressed as:

$$\boldsymbol{v}_c^n = \begin{bmatrix} V_c\cos(\beta_c) \\ V_c\sin(\beta_c) \\ 0 \end{bmatrix} \tag{2.3.8}$$

where V_c and β_c are considered constant in $\{n\}$. The current velocity in $\{b\}$ is found as:

$$\boldsymbol{v}_c^b = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{nb})^T \boldsymbol{v}_c^n \tag{2.3.9}$$

or

$$u_c = V_c\cos(\beta_c - \psi), \quad v_c = V_c\sin(\beta_c - \psi) \tag{2.3.10}$$

Current modeled as a force disturbance

The model (2.3.7) is a suitable model when building a process plant model for simulation purposes. When designing a model based control law, however, it is now clear that the parameters are dependent on the relative velocity $\boldsymbol{\nu}_r$. To get the current velocity from this measurement, one needs a measure of the vehicle velocity as well. To overcome this need, one can consider the current as a constant force disturbance $\boldsymbol{\tau}_C$ in $\{n\}$[Antonelli et al., 2003] and build proper compensation of this disturbance with the control law. The kinetics can then be expressed as:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_C \tag{2.3.11}$$

2.4 ROV guidance, navigation and control

Guidance, navigation and control(GNC) deals with the design of the entire control system for a marine craft, both on the surface and under water. The GNC system consists of 3 parts. The guidance system is what decides where the marine craft should go, the navigation system determines where the craft actually is, and the control system calculates the forces necessary to bring the craft from where it is, to where it should be.

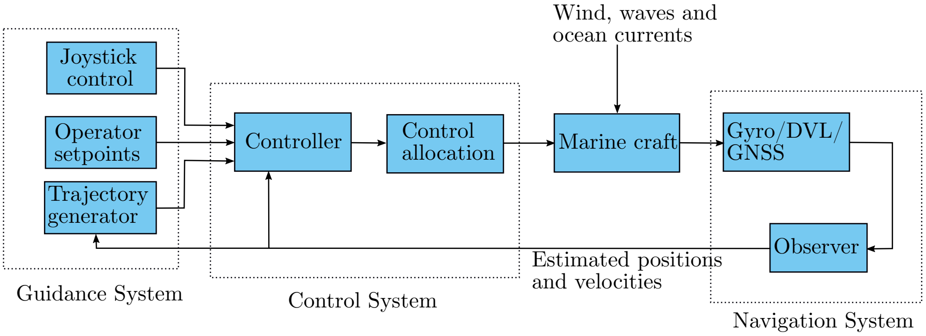


Figure 2.4: A typical Guidance, Navigation and Control system for a marine craft.

2.4.1 Guidance system

A guidance system is a system for automatically guiding the position or path of a marine craft, usually without direct or continuous human control [Fossen, 2011]. There are several methods for guiding the marine craft to the desired position or to specify for instance the desired velocity. Among the most common are:

- **Dynamic Positioning(DP):** This is the most basic guidance system and consists of one or more constant setpoints usually provided by an operator. DP is used in station keeping or when it is necessary for the marine craft to stand still. It may also be used to control a state to zero, like for instance the roll and pitch of an ROV.

- **Trajectory-tracking control:** The position and velocity of the marine craft should track the desired time-varying reference signal. This is for instance used for course changing, setpoint changing or speed changing. The reference change is usually passed through a reference model to generate a smooth, time-varying reference signal.

- **Path-following:** This mode commands the marine craft to follow a predefined path regardless of time. The path may be generated by an operator, or, as in this thesis, by the marine craft itself as it moves through the ocean space.

2.4.2 Navigation system

Navigation involves determining for instance the position, attitude, velocity, course or distance travelled for a marine craft. The most commonly used navigation system for marine vehicles is the US satellite navigation system GPS. [Sørensen, 2013]. The GPS signal is unavailable for a vehicle operating under water, like a ROV. The position must therefore be decided in another way. One possibility is to design a position observer based on the velocity measurements. The velocity measurements can be obtained by using a Doppler Velocity Log(DVL) as is done with the Merlin WR200. For more information about the observer developed for Merlin WR200, see the master thesis [Knausgård, 2013].

2.4.3 Control systems

The use of DP systems began in the 1960s when PID control was implemented for control in surge, sway and yaw. This was improved by Balchen to include multivariable optimal control, [Balchen et al., 1980]. Balchen also implemented the Kalman filter in the control loop which is considered a breakthrough in marine control systems,[Sørensen, 2013, p. 228]. The introduction of Fossen's vectorial representation in 1991, [Fossen, 1991], simplified the representation of the generalized positions and velocities and became the new standard for marine control systems design. A variety of non-linear control methods was developed throughout the 1990s, like a vectorial backstepping controller [Fossen and Berge, 1997], adaptive Variable Structure control [Da Cunha et al., 1995], sliding mode with adaptive gains [Cristi et al., 1990] and adaptive backstepping for tracking control [Godhavn et al., 1998] to mention some.

2.4.4 Controller types

P, PI, PD and PID controllers

The well known and widely used P, PI, PD and PID controllers utilizes the error $e = x - x_d$ to calculate the input τ. x is the current value of the state and x_d is the desired value. K_p, K_i and K_d are controller gains chosen by the control designer to scale the input to an appropriate value. The gain K_p affects the immediate error. Increasing it causes a faster response, but may cause overshoot if too high. K_d affects the term regarding the derivative, or the rate of change, of the error. Increasing it will therefore cause lower overshoot and a faster settling time. K_i affects the sum of the error over time and is used to remove a constant disturbance acting on the system. Increasing K_i leads to a faster removal of the disturbance error and a faster settling time, but it also leads to a larger overshoot.

P	$\tau = -K_p e$
PI	$\tau = -K_p e - K_i \int_0^t e\, dt$
PD	$\tau = -K_p e - K_d \dot{e}$
PID	$\tau = -K_p e - K_d \dot{e} - K_i \int_0^t e dt$

Table 2.3: Input for the different control strategies P, PI, PD and PID control.

These controllers only considers the error. The gains can be chosen regardless of the system parameters, so the controllers are not model-based. The choice of the

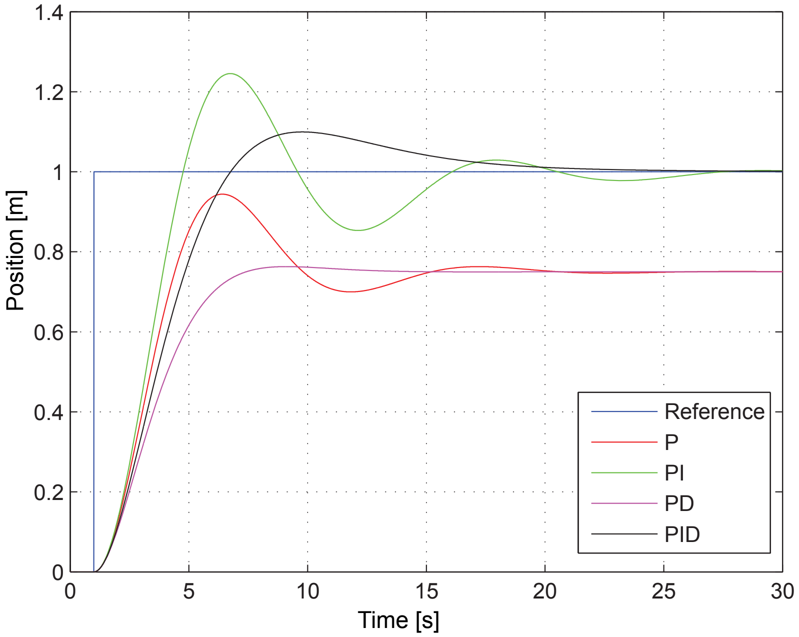


Figure 2.5: The behaviour of a mass spring damper system with different controllers applied. It is seen that using a derivative part causes less oscillations and using an integral part removes the stationary deviation.

control gains can however be based on the system parameters in order to achieve a desired system behaviour. This method is called *pole-placement* and is very useful when controlling the characteristics of a system.

Feedback linearization controllers

The concept of feedback linearization is to transform the non-linear system into an equivalent linear system [Khalil, 2002]. When dealing with non-linear systems, like a marine craft, on the form

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau} \tag{2.4.1}$$

where $\boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta})$ is the non-linear part, it would be beneficiary if there existed a feedback control :

$$\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) \tag{2.4.2}$$

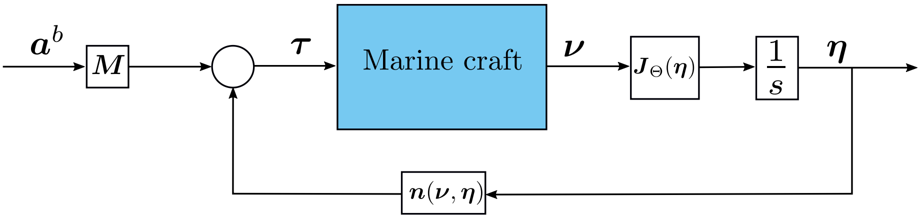


Figure 2.6: A non-linear marine craft system is linearized through the feedback linearizing controller $\boldsymbol{\tau} = \boldsymbol{M}\boldsymbol{a}^b + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})$

with commanded acceleration $\boldsymbol{a}^b$, that transformed the non-linear system into an equivalent linear system:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{a}^b \tag{2.4.3}$$

such that linear design methods like pole placement and linear quadratic control theory can be applied. See figure 2.6. Not all systems are feedback linearizable, but as it turns out, ships and underwater vehicles, which are basically non-linear mass-damper-spring systems, can be linearized through a non-linear mapping as long as all DOFs are controllable. If not, partial feedback linearization is still possible. The commanded acceleration a^b can be chosen by for instance pole placement [Fossen, 2011].

In [Fossen and Pettersen, 2014], [Holden and Pettersen, 2007] and [Moe et al., 2014] feedback linearizing controllers are used. This is fine if one knows the exact parameters of the marine craft.

Backstepping controllers

Backstepping and feedback linearization are closely linked, but where feedback linearization attempts to cancel all non-linearities, backstepping allows us to be more selective in what we wish to remove. A marine craft may for instance have a stabilizing damping term which is in fact doing exactly what we want to do with our controller, only to a different degree. If we cancel it with feedback linearization we are removing all damping, even the one that is ”helping” us, and then adding our own through the controller. With backstepping we look at the Lyapunov function of the system to decide which damping terms we need to be cancel through feedback linearization and which terms we can keep. It is also difficult to obtain the exact model of the non-linearities, so if we can avoid cancelling them, this would make the controller more robust.

When developing backstepping controllers, a recursive method is applied by ”backstepping” through the integrators of the system until the input is reached. For every step a virtual control is chosen to control a new state variable to zero.

Backstepping controllers are used for control of robot manipulators in [Slotine and Weiping, 1988]. The concept of vectorial backstepping was first introduced in [Fossen and Berge, 1997] where it was applied to a marine craft showing simultaneous global exponential stability (GES) of the surge and sway position and the yaw angle whilst incorporating the effects of actuator dynamics. Vectorial backstepping is used to show robust GES for a fully actuated 6 DOF AUV in [Holden and Pettersen, 2007]. A 6 DOF vectorial backstepping controller utilizing a diffeomorphism between the positions and velocities of a marine craft is presented in [Holden and Pettersen, 2007].

Adaptive controllers

Of the aforementioned controllers, both the feedback linearization controller and the backstepping controller are model based. Since they try to cancel the nonlinearities in for instance the damping and Coriolis matrices and multiply the commanded acceleration with the mass matrix, these matrices needs to be known. The problem with this is that values of these matrices changes with different sea states [Sørensen, 2013]. The requirement that the matrices are known can be relaxed by introducing *parameter adaptation.* Consider again the system in equation (2.4.1), but now the controller is chosen as:

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}\boldsymbol{a}^b + \hat{\boldsymbol{n}}(\boldsymbol{\nu}, \boldsymbol{\eta}) \tag{2.4.4}$$

where the *hat* denotes an adapted parameter. The error dynamics of the system is:

$$\boldsymbol{M}(\dot{\boldsymbol{\nu}} - \boldsymbol{a}^b) = (\hat{\boldsymbol{M}} - \boldsymbol{M})\boldsymbol{a}^b + (\hat{\boldsymbol{n}}(\boldsymbol{\nu}, \boldsymbol{\eta}) - \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})) \tag{2.4.5}$$

If the system is linear in the parameters, the right hand side can be written as:

$$(\hat{\boldsymbol{M}} - \boldsymbol{M})\boldsymbol{a}^b + (\hat{\boldsymbol{n}}(\boldsymbol{\nu}, \boldsymbol{\eta}) - \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})) = \boldsymbol{\phi}(\boldsymbol{a}^b, \boldsymbol{\nu}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \tag{2.4.6}$$

where $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is the *unknown* parameter error and $\boldsymbol{\phi}(\boldsymbol{a}^b, \boldsymbol{\nu}, \boldsymbol{\eta})$ is a vector of measured and known signals referred to as the *regressor* matrix.

Adaptation on the form in equation (2.4.6) can also be used to cancel unknown constant disturbances. Consider a marine craft under the effect of ocean currents $\boldsymbol{\tau}_c$:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_c \tag{2.4.7}$$

by choosing

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}\boldsymbol{a}^b + \hat{\boldsymbol{n}}(\boldsymbol{\nu}, \boldsymbol{\eta}) - \hat{\boldsymbol{\tau}}_c \tag{2.4.8}$$

equation (2.4.6) can be rewritten as:

$$(\hat{\boldsymbol{M}} - \boldsymbol{M})\boldsymbol{a}^b + (\hat{\boldsymbol{n}}(\boldsymbol{\nu}, \boldsymbol{\eta}) - \boldsymbol{n}(\boldsymbol{\nu}, \boldsymbol{\eta})) - (\hat{\boldsymbol{\tau}}_c - \boldsymbol{\tau}_c) = \boldsymbol{\phi}(\boldsymbol{a}^b, \boldsymbol{\nu}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \tag{2.4.9}$$

and hence, the current disturbance is also adapted. It is shown in section 4.2 that by choosing an update law for the parameter adaptation vector $\hat{\boldsymbol{\theta}}$ on the form:

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{\phi}\bar{\boldsymbol{e}}_2 \tag{2.4.10}$$

the overall system is stabilized. $\bar{\boldsymbol{e}}_2$ is an error variable and $\boldsymbol{\Gamma}$ is matrix of gains that are defined in section 4.2.

The convergence of $\hat{\boldsymbol{\theta}}$ to $\boldsymbol{\theta}$ can only be guaranteed if the input that drives $\dot{\hat{\boldsymbol{\theta}}}$ is *persistently exciting*. This is defined in Appendix C.2.

In [Antonelli et al., 2001] and [Antonelli et al., 2003] an adaptive control law is developed for an AUV and a ROV respectively. The controller adapts the environmental disturbances. This controller was tested in [Ohrem, 2014] showing good results. Adaptive control is also used in [Zhu and Gu, 2011] where it is combined with backstepping. All disturbances are lumped into a single disturbance term, so it does not separate for instance the current disturbance. In [Patompak and Nilkhamhang, 2012], the controller from [Zhu and Gu, 2011] is modified to include a bound estimation on the uncertainties.

Sliding mode and variable structure control

Sliding mode control, or variable structure control, is based on switching terms that constraints the motion of the system to a sliding surface [Utkin, 1977]. Once on this sliding surface, the performance of the system is insensitive to parameter variations or disturbances. This technique has been used in combination with Model Reference Adaptive Control in [Hsu, 1990] where a novel controller named the VS-MRAC is derived. The VS-MRAC method is applied to an ROV and used for position control in [Da Cunha et al., 1995]. In [Cristi et al., 1990], sliding mode control is used in the dive plane for controlling an AUV. [Lyshevski, 2001] describes the motions in 6 DOF and derives a sliding mode controller that utilizes the continuous function tanh(·) instead of the discrete sign(·) function to ensure feasible control inputs. Sliding mode control may cause *chattering* in the control input [Young et al., 1999].

2.5 Previous work on the DP system of Merlin WR200

Merlin WR200 is equipped with automatic depth, altitude and heading control. It has velocity control by joystick in surge, sway, heave and yaw and the possibility of slightly changing the roll and pitch angles. The master project of Knausgård [2013] aimed to design and implement a DP control system with observers for Merlin WR200. The system showed successful results in the sea-trials and the observers worked particularly well, but due to lack of time the system was not tested in all DOFs. The controller developed in Knausgård [2013] is a feedback linearizing controller. A controller on this form requires a very accurate model of the system and Knausgård concludes that a different controller should be developed and tested. In the project thesis [Ohrem, 2014], several controllers was tested in a comparative study. It was concluded that a controller with current adaptation was a good choice for DP control of Merlin WR200. IKM Subsea has had some troubles with the continuity of the control system and as far as the author is aware, the station keeping mode is not implemented in the Merlin WR200 today.

2.6 Path following for marine crafts

2.6.1 LOS steering

The following is collected from [Fossen, 2011, Ch. 10.3.2]. The LOS guidance system considers the vessels position relative to two waypoints $\boldsymbol{wp}_k = \begin{bmatrix} x_k & y_k \end{bmatrix}^T$ and $\boldsymbol{wp}_{k+1} = \begin{bmatrix} x_{k+1} & y_{k+1} \end{bmatrix}^T$. The path fixed reference frame with origin in $\boldsymbol{wp}_k$ is rotated a positive angle:

$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \in \mathbb{S} \tag{2.6.1}$$

relative to the x axis. atan2 is the four-quadrant version of $\arctan(y/x) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. The coordinates of the vessel in the path-fixed reference frame is:

$$\boldsymbol{\epsilon}(t) := \boldsymbol{R}_p^T(\alpha_k)(\boldsymbol{p}(t) - \boldsymbol{wp}_k) \tag{2.6.2}$$

where

$$\boldsymbol{R}_p(\alpha_k) = \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \in SO(2) \tag{2.6.3}$$

and $\boldsymbol{\epsilon}(t) = \begin{bmatrix} s(t) & e(t) \end{bmatrix}^T \in \mathbb{R}^2$. The variables $s(t)$ and $e(t)$ are described as:

$$\begin{aligned} s(t) &= \text{along track distance(tangential to path)} \\ e(t) &= \text{cross track error(normal to path)} \end{aligned}$$

Expanding the equation (2.6.2) gives:

$$\begin{aligned} s(t) &= \left[x(t) - x_k\right] \cos(\alpha_k) + \left[y(t) - y_k\right] \sin(\alpha_k) \\ e(t) &= -\left[x(t) - x_k\right] \sin(\alpha_k) + \left[y(t) - y_k\right] \cos(\alpha_k) \end{aligned} \tag{2.6.4}$$

For path following of a straight line path, the control objective becomes:

$$\lim_{t \to \infty} e(t) = 0 \tag{2.6.5}$$

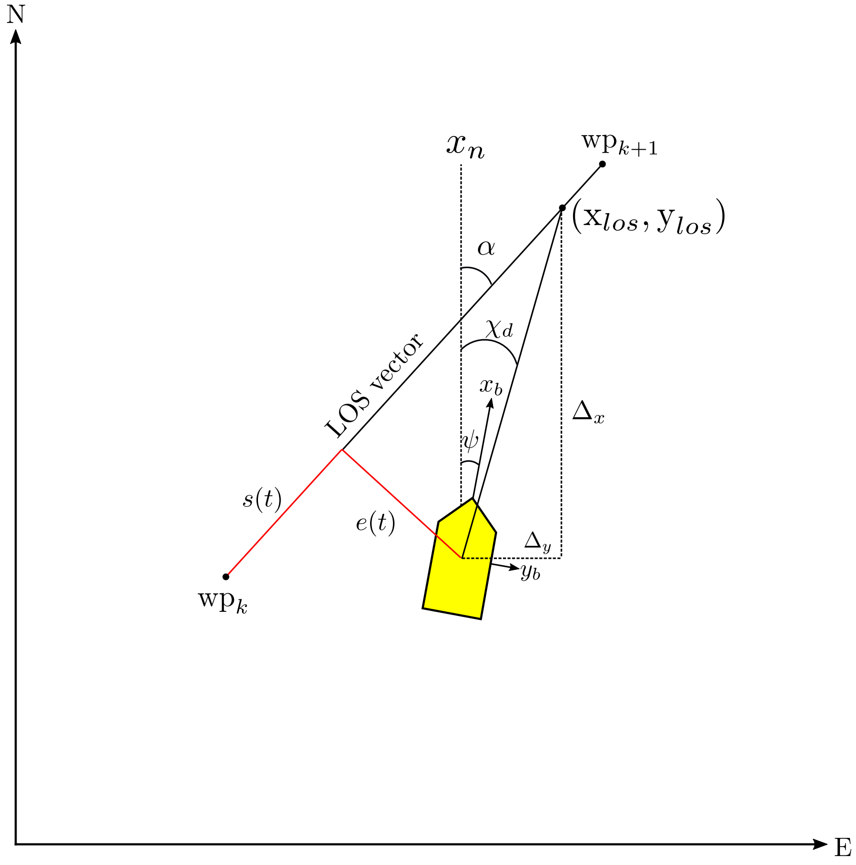


Figure 2.7: The Line of Sight guidance law. The desired course angle χ_d is chosen to point toward the intersection point (x_{los}, y_{los}).

which implies that the vessel is on the straight line between the waypoints. See figure 2.7 for illustration.

As the vessel approaches the waypoint $\boldsymbol{wp}_{k+1}$ a new waypoint should be chosen. This is done by defining a circle of acceptance with radius R_{k+1} around the waypoint $\boldsymbol{wp}_{k+1}$. If the vessel position at time t satisfies:

$$\left[x_{k+1} - x(t)\right]^2 + \left[y_{k+1} - y(t)\right]^2 \leq R_{k+1}^2 \tag{2.6.6}$$

the next waypoint is chosen. The velocity of the marine craft is either chosen as constant or it can be chosen as:

$$\begin{bmatrix} u_d \\ v_d \end{bmatrix} = \begin{bmatrix} -u_{max}\frac{s}{\sqrt{s^2+\Delta_s^2}} \\ -v_{max}\frac{e}{\sqrt{e^2+\Delta_e^2}} \end{bmatrix} \tag{2.6.7}$$

where $\Delta_s > 0$ and $\Delta_e > 0$ are speed tuning parameters that ensures a smooth ramping down of the velocity as the vessel approaches the waypoint.

Two different guidance principles can be used for steering along the LOS vector, *enclosure-based* and *lookahead-based* steering. This thesis only considers the missile guidance motivated lookahead-based approach since it has several advantages over enclosure-based steering. For lookahead based steering the desired course angle χ_d is chosen so it points towards the the intersection point (x_{los}, y_{los}):

$$\chi_d(e) = \alpha_k + \arctan\left(\frac{-e}{\Delta}\right) \tag{2.6.8}$$

where $\Delta > 0$ is the lookahead distance. The last term ensures that the velocity is directed towards a point on the path that is located a lookahead distance away from the direct projection of the vessels position on to the path.

LOS steering is chosen because this method popular and effective, and it is motivated by how the operator is most likely to steer the ROV [Fossen and Lekkas, 2015].

2.7 Limitations

No dynamic model for the thrusters has been implemented due to lack of data regarding the thruster characteristics. A model of the thruster dynamics is important to assure feasible changes in thruster rates. A simple rate limiter that limits the rate of change in the thrusters to 8000 N/s is used. This is based on information given by IKM Subsea. The controllers are tuned with this in mind. All measurements used in the simulations are noiseless and no observer is implemented. The system parameter matrices are based on the the results presented in the master thesis [Knausgård, 2013] and may therefore be outdated due to design changes in the Merlin WR200. The forces due to cable drag are not modeled.

2.8 Available measurements

Merlin WR200 is equipped with a Doppler Velocity Log (DVL) sensor called TOGSNAV. This provides the controller with measurements of the depth, velocities and attitudes when not operating too close to the seabed. Estimates are used when the DVL measurements are unavailable. The positions are obtained through an observer that was developed and verified in Knausgård [2013] showing very good results. For the simulations in this thesis, noiseless measurements of the positions, attitudes and velocities are assumed.

2.9 Mathematical review

Some important lemmas and definitions that are used in the thesis are presented here

2.9.1 Norms

From [Khalil, 2002, Ch.5.1], the following definitions are presented regarding the norm function $||f||$:

Definition 2.4: *The norm of a signal is zero if and only if the signal is identically zero and is strictly positive otherwise*

Definition 2.5: *Scaling a signal results in a corresponding scaling of the norm. That is $||af|| = a||f||$ for any positive a and every signal f.*

Definition 2.6: *The norm satisfies the triangle inequality $||f_1 + f_2|| \leq ||f_1|| + ||f_2||$ for any signals f_1 and f_2.*

For the space of piecewise continuous, bounded functions, the norm is defined as:

$$||f||_{\mathcal{L}_\infty} := \sup_{t\geq 0} ||f(t)|| \ < \ \infty \tag{2.9.1}$$

For the space of piecewise continuous, square-integrable functions, the norm is defined as:

$$||f||_{\mathcal{L}_2} := \sqrt{\int_0^\infty f^T(t)f(t)\,dt} \ < \infty \tag{2.9.2}$$

The *Euclidian norm* is:

$$||x||_2 = \left(|x_1|^2 + \cdots + |x_n|^2\right)^{1/2} = \left(x^T x\right)^{1/2} \tag{2.9.3}$$

2.9.2 Lyapunov function

Definition 2.7: *A **Lyapunov Function** is a continuously differentiable function $V(x)$ that satisfies [Khalil, 2002]*

$$V(0) = 0 \text{ and } V(x) > 0 \ \forall \ x \neq 0 \tag{2.9.4a}$$

$$\dot{V}(x) \leq 0 \tag{2.9.4b}$$

2.9.3 Barbălat's lemma

Lemma 2.1: *If f, $\dot{f} \in \mathcal{L}_\infty$ and $f \in \mathcal{L}_p$ for some $p \in [1, \infty)$, then $f(t) \to 0$ as $t \to \infty$. [Ioannou and Sun, 2012, Lemma 3.2.5]*

Lemma 2.1 is a special case of the more general Barbălat's Lemma which is stated below

Lemma 2.2: *If $\lim_{t\to\infty} \int_0^t f(\tau)d\tau$ exists and is finite, and $f(t)$ is a uniformly continuous function, then $\lim_{t\to\infty} f(t) \to 0$. [Khalil, 2002, Lemma 8.2]*

Chapter 3

Modeling of Merlin WR200

The Merlin WR200 is a fully electric work class ROV that has an open frame solution with rounded edges. The open frame not only simplifies repairs, but it also allows water to flow more freely around and through the ROV. To maximize reliability and operation time, as many components as possible are located topside. The ROV can be fitted with two manipulator arms, cameras and skids. Merlin WR200 is equipped with a Tether Management System to reduce the cable drag and increase manoeuvrability. The main data is listed in table 3.1. The model for the Merlin WR200 is based on the vectorial notation derived in [Fossen, 1991] and the following assumptions are made:

Assumption 3.1: *The $\{n\}$ frame is considered inertial.*

Assumption 3.2: *The ROV will operate deeply submerged*

Assumption 3.3: *The ocean current is considered non-rotational and constant in $\{n\}$ with $\dot{\boldsymbol{\nu}}_c^n = 0$. It is also assumed that the current has no vertical component.*

Assumption 3.4: *The density of seawater is considered constant for all depths.*

Assumption 3.5: *The ROV is modelled as a rigid body and the mass is constant.*

Length	2.8m
Width	1.8m
Height	1.7m
Weight	3184kg

Table 3.1: Merlin data

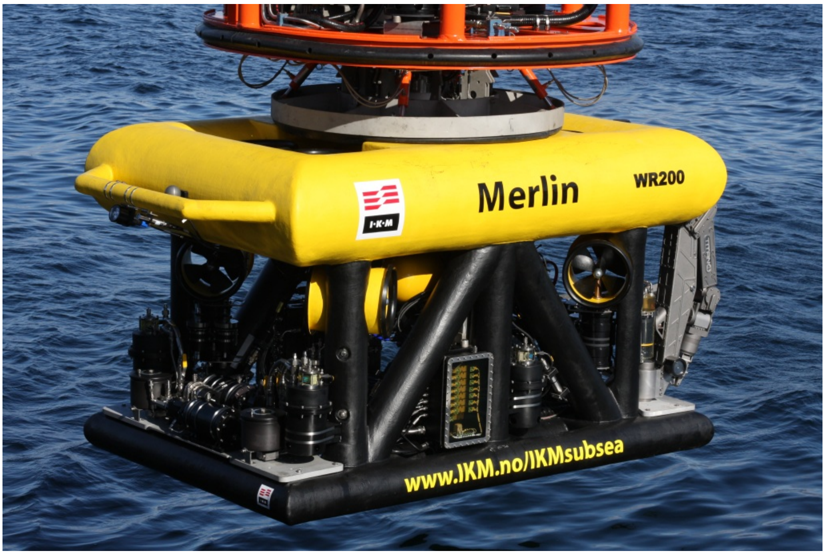


Figure 3.1: The Merlin WR200 ROV

The equations of motion are presented as both a process plant model and a control plant model. The process plant model is a comprehensive high fidelity model of the ROV. It should be as detailed as possible to replicate the real system. The control plant model is a simplified version that only considers the main physical properties of the plant to simplify the mathematical description [Sørensen, 2005]

3.1 Process plant model

A process plant model that is as accurate as possible given the available data about Merlin WR200 is developed for use in computer simulations. By utilizing the fact that the ocean currents are irrotational and constant in $\{n\}$, the following property can be applied [Fossen, 2011, Property 8.1]

$$\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{3.1.1}$$

And hence, the equations of motions are:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \tag{3.1.2}$$

Expanding the terms gives:

$$\underbrace{\boldsymbol{M}_{RB}\dot{\boldsymbol{\nu}} + \boldsymbol{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu}}_{\text{rigid-body forces}} + \underbrace{\boldsymbol{M}_{A}\dot{\boldsymbol{\nu}}_r + \boldsymbol{C}_{A}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r}_{\text{hydrodynamic forces}} + \underbrace{\boldsymbol{g}(\boldsymbol{\eta})}_{\text{hydrostatic forces}} = \boldsymbol{\tau} \tag{3.1.3}$$

3.1.1 Rigid body mass matrix

The rigid body mass matrix is given in [Fossen, 2011] as:

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

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_g & -my_g \\ 0 & m & 0 & -mz_g & 0 & mx_g \\ 0 & 0 & m & my_g & -mx_g & 0 \\ 0 & -mz_g & my_g & I_{xx} & -I_{xy} & -I_{xz} \\ mz_g & 0 & -mx_g & -I_{yx} & I_{yy} & -I_{yz} \\ -my_g & mx_g & 0 & -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{3.1.4}$$

where m is the mass of Merlin WR200, $\boldsymbol{I}_b \in \mathbb{R}^{3\times3}$ is the inertia matrix, $\boldsymbol{S}(\cdot) \in \mathbb{R}^{3\times3}$ is the cross product operator (see appendix C.1) and x_g, y_g and z_g are the distances from the center of origin, CO, to the center of gravity, CG. According to data from the CAD model of Merlin WR200 in appendix A, the vector $\boldsymbol{r}_g$ is:

$$\boldsymbol{r}_g = \begin{bmatrix} -0.002341 & 0.003014 & -0.021193 \end{bmatrix}^T \text{ [m]} \tag{3.1.5}$$

The mass of Merlin WR200 is $m = 3184$ [kg] and the inertia is as presented in appendix A

$$\boldsymbol{I} = \begin{bmatrix} 1819 & 0 & -120 \\ 0 & 3064 & 7 \\ -120 & 7 & 2887 \end{bmatrix} \tag{3.1.6}$$

This gives a rigid body mass matrix:

$$\boldsymbol{M}_{RB} = \begin{bmatrix} 3184 & 0 & 0 & 0 & -67.5 & -9.6 \\ 0 & 3184 & 0 & 67.5 & 0 & -7.5 \\ 0 & 0 & 3184 & 9.6 & 7.5 & 0 \\ 0 & 67.5 & 9.6 & 1819 & 0 & -120 \\ -67.5 & 0 & 7.5 & 0 & 3064 & 7 \\ -9.6 & -7.5 & 0 & -120 & 7 & 2887 \end{bmatrix} \tag{3.1.7}$$

3.1.2 Added mass matrix

The added mass matrix is given in [Fossen, 2011] and consists of the hydrodynamic derivatives. It is shown in (3.1.8).

$$\boldsymbol{M}_A = - \begin{bmatrix} X_{\dot{u}} & X_{\dot{v}} & X_{\dot{w}} & X_{\dot{p}} & X_{\dot{q}} & X_{\dot{r}} \\ Y_{\dot{u}} & Y_{\dot{v}} & Y_{\dot{w}} & Y_{\dot{p}} & Y_{\dot{q}} & Y_{\dot{r}} \\ Z_{\dot{u}} & Z_{\dot{v}} & Z_{\dot{w}} & Z_{\dot{p}} & Z_{\dot{q}} & Z_{\dot{r}} \\ K_{\dot{u}} & K_{\dot{v}} & K_{\dot{w}} & K_{\dot{p}} & K_{\dot{q}} & K_{\dot{r}} \\ M_{\dot{u}} & M_{\dot{v}} & M_{\dot{w}} & M_{\dot{p}} & M_{\dot{q}} & M_{\dot{r}} \\ N_{\dot{u}} & N_{\dot{v}} & N_{\dot{w}} & N_{\dot{p}} & N_{\dot{q}} & N_{\dot{r}} \end{bmatrix} \tag{3.1.8}$$

This modeling of Merlin WR200 is based on [Knausgård, 2013]. The added mass is assumed to be 10 % of the mass in surge, sway and heave and 5% of the mass in roll, pitch and yaw. All other added mass terms are assumed equal to 0. Though these assumptions are insecure, the results from the sea trial in [Knausgård, 2013] show that they are usable. The added mass matrix takes the form:

$$\boldsymbol{M}_A = -\begin{bmatrix} -318.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & -318.4 & 0 & 0 & 0 & 0 \\ 0 & 0 & -318.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -159.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -159.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -159.2 \end{bmatrix} \tag{3.1.9}$$

A diagonal structure of this matrix is a good approximation due to the fact that the off diagonal elements will be much smaller than their diagonal counterparts [Fossen, 2011].

3.1.3 Total mass matrix

The total mass matrix used in the simulation model for Merlin WR200 can now be presented as:

$$\begin{aligned} \boldsymbol{M} &= \boldsymbol{M}_{RB} + \boldsymbol{M}_A \\ &= \begin{bmatrix} 3502.4 & 0 & 0 & 0 & -67.5 & -9.6 \\ 0 & 3502.4 & 0 & 67.5 & 0 & -7.5 \\ 0 & 0 & 3502.4 & 9.6 & 7.5 & 0 \\ 0 & 67.5 & 9.6 & 1978.2 & 0 & -120 \\ -67.5 & 0 & 7.5 & 0 & 3223.2 & 7 \\ -9.6 & -7.5 & 0 & -120 & 7 & 3046.2 \end{bmatrix} \end{aligned} \tag{3.1.10}$$

3.1.4 Rigid body Coriolis and centripetal matrix

The rigid body Coriolis and centripetal matrix is given in [Fossen, 2011] as:

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{0}_{3\times3} & -m\boldsymbol{S}(\boldsymbol{v}) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{w})\boldsymbol{r}_g^b) \\ -m\boldsymbol{S}(\boldsymbol{v}) - m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{w})\boldsymbol{r}_g^b) & m\boldsymbol{S}(\boldsymbol{S}(\boldsymbol{v})\boldsymbol{r}_g^b - \boldsymbol{S}(\boldsymbol{I}_b\boldsymbol{w})) \end{bmatrix} \tag{3.1.11}$$

where $\boldsymbol{v}$ is the linear velocity vector and $\boldsymbol{w}$ is the angular velocity vector as presented in table 2.2. $\boldsymbol{S}(\cdot)$ is the cross product operator and $\boldsymbol{r}_g^b \in \mathbb{R}^3$ is the vector of distances from CO to the center of gravity, CG. When expanding the terms, this matrix takes the form:

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \left[\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -m(y_g q + z_g r) & m(y_g p + w) & m(z_g p - v) \\ m(x_g q - w) & -m(z_g r + x_g p) & m(z_g q + u) \\ m(x_g r + v) & m(y_g r - u) & -m(x_g p + y_g q) \end{matrix}\right.$$

$$\left.\begin{matrix} m(y_g q + z_g r) & -m(x_g q - w) & -m(x_g r + v) \\ -m(y_g + w) & m(z_g r + x_g p) & -m(y_g r - u) \\ -m(z_g p - v) & -m(z_g q + u) & m(x_g p + y_g q) \\ 0 & I_{yz} q - I_{xz} p + I_z r & I_{yz} r + I_{xy} p - I_y q \\ I_{yz} q + I_{xz} p - I_z r & 0 & -I_{xz} r - I_{xy} q + I_x p \\ -I_{yz} r - I_{xy} p + I_y q & I_{xz} r + I_{xy} q - I_x p & 0 \end{matrix}\right] \tag{3.1.12}$$

3.1.5 Added mass Coriolis and centripetal matrix

The hydrodynamic Coriolis and centripetal matrix can be parameterized such that it is skew-symmetric, according to [Fossen, 2011]. This leads to the matrix in equation (3.1.13).

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & -\boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{v} + \boldsymbol{A}_{12}\boldsymbol{w}) \\ \boldsymbol{S}(\boldsymbol{A}_{11}\boldsymbol{v} + \boldsymbol{A}_{12}\boldsymbol{w}) & -\boldsymbol{S}(\boldsymbol{A}_{21}\boldsymbol{v} + \boldsymbol{A}_{22}\boldsymbol{w}) \end{bmatrix} \tag{3.1.13}$$

where

$$\boldsymbol{M}_A = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{bmatrix} \tag{3.1.14}$$

Equation (3.1.13) can be written in component form as:

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -a_3 & a_2 \\ 0 & 0 & 0 & a_3 & 0 & -a_1 \\ 0 & 0 & 0 & -a_2 & a_1 & 0 \\ 0 & -a_3 & a_2 & 0 & -b_3 & b_2 \\ a_3 & 0 & -a_1 & b_3 & 0 & -b_1 \\ -a_2 & a_1 & 0 & -b_2 & b_1 & 0 \end{bmatrix} \tag{3.1.15}$$

where

$$\begin{aligned} a_1 &= X_{\dot{u}} u + X_{\dot{v}} v + X_{\dot{w}} w + X_{\dot{p}} p + X_{\dot{q}} q + X_{\dot{r}} r \\ a_2 &= Y_{\dot{u}} u + Y_{\dot{v}} v + Y_{\dot{w}} w + Y_{\dot{p}} p + Y_{\dot{q}} q + Y_{\dot{r}} r \\ a_3 &= Z_{\dot{u}} u + Z_{\dot{v}} v + Z_{\dot{w}} w + Z_{\dot{p}} p + Z_{\dot{q}} q + Z_{\dot{r}} r \\ b_1 &= K_{\dot{u}} u + K_{\dot{v}} v + K_{\dot{w}} w + K_{\dot{p}} p + K_{\dot{q}} q + K_{\dot{r}} r \\ b_2 &= M_{\dot{u}} u + M_{\dot{v}} v + M_{\dot{w}} w + M_{\dot{p}} p + M_{\dot{q}} q + M_{\dot{r}} r \\ b_3 &= N_{\dot{u}} u + N_{\dot{v}} v + N_{\dot{w}} w + N_{\dot{p}} p + N_{\dot{q}} q + N_{\dot{r}} r \end{aligned} \tag{3.1.16}$$

3.1.6 Damping matrix

An ROV that is deeply submerged is mostly affected by damping caused by skin friction and vortex shredding [Fossen, 2011]. The damping contributions cause both linear and non-linear damping and are separated into a linear damping term caused by potential damping and skin friction and a non-linear damping term caused by higher order terms. This results in the following damping matrix:

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \boldsymbol{D}_l + \boldsymbol{D}_{nl}(\boldsymbol{\nu}_r) \tag{3.1.17}$$

where $\boldsymbol{D}_l$ contains the linear damping coefficients

$$\boldsymbol{D}_l = -\begin{bmatrix} X_u & X_v & X_w & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_p & Z_q & Z_r \\ K_u & K_v & K_w & K_p & K_q & K_r \\ M_u & M_v & M_w & M_p & M_q & M_r \\ N_u & N_v & N_w & N_p & N_q & N_r \end{bmatrix} \tag{3.1.18}$$

and $\boldsymbol{D}_{nl}(\boldsymbol{\nu}_r)$ contains the higher order damping coefficients

$$\boldsymbol{D}_{nl}(\boldsymbol{\nu}_r) = -\begin{bmatrix} X_{|u|u}|u_r| & X_{|v|v}|v_r| & X_{|w|w}|w_r| & X_{|p|p}|p_r| & X_{|q|q}|q_r| & X_{|r|r}|r_r| \\ Y_{|u|u}|u_r| & Y_{|v|v}|v_r| & Y_{|w|w}|w_r| & Y_{|p|p}|p_r| & Y_{|q|q}|q_r| & Y_{|r|r}|r_r| \\ Z_{|u|u}|u_r| & Z_{|v|v}|v_r| & Z_{|w|w}|w_r| & Z_{|p|p}|p_r| & Z_{|q|q}|q_r| & Z_{|r|r}|r_r| \\ K_{|u|u}|u_r| & K_{|v|v}|v_r| & K_{|w|w}|w_r| & K_{|p|p}|p_r| & K_{|q|q}|q_r| & K_{|r|r}|r_r| \\ M_{|u|u}|u_r| & M_{|v|v}|v_r| & M_{|w|w}|w_r| & M_{|p|p}|p_r| & M_{|q|q}|q_r| & M_{|r|r}|r_r| \\ N_{|u|u}|u_r| & N_{|v|v}|v_r| & N_{|w|w}|w_r| & N_{|p|p}|p_r| & N_{|q|q}|q_r| & N_{|r|r}|r_r| \end{bmatrix} \tag{3.1.19}$$

A pull test in surge and sway was performed in the project thesis [Knausgård, 2012]. In this test it was found that the damping of Merlin WR200 had a very clear quadratic form in surge and sway. The linear damping matrix is therefore omitted in the simulation model since no data on this exist. Due to the symmetric shape of Merlin WR200, the damping could be assumed equal in heave and sway. As for damping in roll, pitch and yaw, these values are assumed very small. In [Knausgård, 2012] they are assigned a value of 10% of the average damping in surge and sway. The resulting damping matrix is:

$$\boldsymbol{D}(\boldsymbol{\nu}_r) = \begin{bmatrix} 1321|u_r| & 0 & 0 & 0 & 0 & 0 \\ 0 & 2525|v_r| & 0 & 0 & 0 & 0 \\ 0 & 0 & 2525|w_r| & 0 & 0 & 0 \\ 0 & 0 & 0 & 192|p_r| & 0 & 0 \\ 0 & 0 & 0 & 0 & 192|q_r| & 0 \\ 0 & 0 & 0 & 0 & 0 & 192|r_r| \end{bmatrix} \tag{3.1.20}$$

3.1.7 Hydrostatic forces

The vector of hydrostatic forces is [Fossen, 2011]:

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} (W-B)\text{sin}(\theta) \\ -(W-B)\text{cos}(\theta)\text{sin}(\phi) \\ -(W-B)\text{cos}(\theta)\text{cos}(\phi) \\ -(y_gW - y_bB)\text{cos}(\theta)\text{cos}(\phi) + (z_gW - z_bB)\text{cos}(\theta)\text{sin}(\phi) \\ (z_gW - z_bB)\text{sin}(\theta) + (x_gW - x_bB)\text{cos}(\theta)\text{cos}(\phi) \\ -(x_gW - x_bB)\text{cos}(\theta)\text{sin}(\phi) - (y_gW - y_bB)\text{sin}(\theta) \end{bmatrix} \tag{3.1.21}$$

where

$$\begin{aligned} W &= mg \\ B &= \rho g V \end{aligned} \tag{3.1.22}$$

is the weight and buoyancy of Merlin WR200 respectively. The mass of Merlin WR 200 is $m = 3184kg$, the gravitational acceleration is $g = 9.81m/s^2$, the density of sea water is $\rho = 1024kg/m^3$ and the volume of Merlin WR200 is $V = 3.22m^3$.

3.2 Control plant model

Some assumptions are made when deriving the control plant model for Merlin WR200 for this thesis:

Assumption 3.6: *The relative velocity is measured, but the current velocity is unknown*

Assumption 3.7: *The center of origin(CO) coincides with the center of gravity(CG) and the center of buoyancy(CB) is located directly over the center of gravity. That is: $x_g = y_g = z_g = x_b = y_b = 0$. This leads to a simplified rigid* *body mass and Coriolis matrix, $\boldsymbol{M}_{RB}$, $\boldsymbol{C}_{RB}$[Fossen, 2011, p.56-57]:*

$$\boldsymbol{M}_{RB} = \begin{bmatrix} m\boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_g \end{bmatrix} \tag{3.2.1}$$

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & mw & -mv \\ 0 & 0 & 0 & -mw & 0 & mu \\ 0 & 0 & 0 & mv & -mu & 0 \\ 0 & mw & -mv & 0 & I_z r & -I_y q \\ -mw & 0 & mu & -I_z r & 0 & I_x p \\ mv & -mu & 0 & I_y q & -I_x p & 0 \end{bmatrix} \tag{3.2.2}$$

Assumption 3.8: *The ROV moves at low speeds and has three planes of symmetry. This suggests that the off-diagonal elements of the added mass matrix $\boldsymbol{M}_A$ can be neglected and the added mass Coriolis matrix $\boldsymbol{C}_A(\boldsymbol{\nu}_r)$ is simplified [Fossen, 2011, p.121]:*

$$\boldsymbol{M}_A = \boldsymbol{M}_A^T = -diag\{X_{\dot{u}},\, Y_{\dot{v}},\, Z_{\dot{w}},\, K_{\dot{p}},\, M_{\dot{q}},\, N_{\dot{r}}\} \tag{3.2.3}$$

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = -\boldsymbol{C}_A^T(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{3.2.4}$$

Assumption 3.9: *The roll and pitch motions are very small, that is $\phi = \theta \approx 0$. This leads to a reduced transformation matrix $\boldsymbol{J}_\Theta(\boldsymbol{\eta})$:*

$$\boldsymbol{P}(\psi) = \begin{bmatrix} \boldsymbol{R}(\psi) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{3.2.5}$$

where $\boldsymbol{P}^{-1}(\psi) = \boldsymbol{P}^T(\psi)$

Assumption 3.10: *The current is modeled as a constant force disturbance in $\{n\}$.*

These assumptions gives the following control plant model:

$$\begin{gathered} \dot{\boldsymbol{\eta}} = \boldsymbol{P}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_C \end{gathered} \tag{3.2.6}$$

where the matrices $\boldsymbol{M}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ and the vector $\boldsymbol{g}$ takes the form:

$$\begin{aligned}\boldsymbol{M} &= \boldsymbol{M}_{RB} + \boldsymbol{M}_A \\ &= \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 & 0 & 0 & 0 \\ 0 & 0 & m - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_x - K_{\dot{p}} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix}\end{aligned} \tag{3.2.7}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB} + \boldsymbol{C}_A \tag{3.2.8}$$

$$= \left[\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & (m - Z_{\dot{w}})w & -(m - Y_{\dot{v}})v \\ -(m - Z_{\dot{w}})w & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \end{matrix}\right. \tag{3.2.9}$$

$$\left.\begin{matrix} 0 & (m - Z_{\dot{w}})w & -(m - Y_{\dot{v}})v \\ -(m - Z_{\dot{w}})w & 0 & (m - X_{\dot{u}})u \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 \\ 0 & (I_z - N_{\dot{r}})r & -(I_y - M_{\dot{q}})q \\ -(I_z - N_{\dot{r}})r & 0 & (I_x - K_{\dot{p}})p \\ (I_y - M_{\dot{q}})q & -(I_x - K_{\dot{p}})p & 0 \end{matrix}\right]$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} 1321|u| & 0 & 0 & 0 & 0 & 0 \\ 0 & 2525|v| & 0 & 0 & 0 & 0 \\ 0 & 0 & 2525|w| & 0 & 0 & 0 \\ 0 & 0 & 0 & 192|p| & 0 & 0 \\ 0 & 0 & 0 & 0 & 192|q| & 0 \\ 0 & 0 & 0 & 0 & 0 & 192|r| \end{bmatrix} \tag{3.2.10}$$

$$\boldsymbol{g}(\boldsymbol{\eta}) = \begin{bmatrix} 0 \\ 0 \\ -(W - B) \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.2.11}$$

3.3 Thrusters

Merlin WR200 is equipped with 8 thrusters. 4 vertical and 4 horizontal. The thrusters are controlled by frequency converters and there is no feedback from the thrusters. Various thrust losses will be present, but the modelling of these are not included in this thesis. In [Knausgård, 2013] a pull test performed by IKM Subsea is presented. The pull test gave rise to a mapping between desired thrust and frequency. The mapping function for a single thruster is:

$$f_{Hz_i} = 83\sin(0.0007909u_i), \;\; i = 1, 2 \ldots 8 \tag{3.3.1}$$

where f_{Hz_i} is the frequency in Hertz and u_i is the desired force for one thruster calculated by the controller and mapped to the thruster through the thruster allocation.

3.3.1 Thruster allocation

The control forces and moments and the thruster forces and moments are related through the thruster allocation:

$$\boldsymbol{\tau} = \boldsymbol{TKu} \tag{3.3.2}$$

Where $\boldsymbol{T}$ is a thruster configuration matrix, $\boldsymbol{K}$ is a diagonal force coefficient matrix set to the identity matrix in this thesis because we want the thrust-to-thrust relationship between $\boldsymbol{\tau}$ and $\boldsymbol{u}$. $\boldsymbol{u}$ contains the thruster inputs for each individual thruster, [Fossen, 2011, p. 398]:

$$\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 & u_8 \end{bmatrix}^T \tag{3.3.3}$$

The thruster configuration matrix contains the locations and geometry of the different thrusters and it relates the control force to the thruster force by calculating how much thrust each thruster can produce in each DOF. The thruster configuration matrix is based on the simulator at IKM Subsea and must not be confused with the matrix in [Knausgård, 2013] which is not the same. The numbering of the thrusters can be seen in figure 3.2.

The thruster configuration matrix for Merlin WR200 as it is in the simulator at IKM Subsea is:

$$\boldsymbol{T} = \begin{bmatrix} \sin(\alpha) & \sin(\alpha) & -\sin(\alpha) & -\sin(\alpha) & 0 & 0 & 0 & 0 \\ \cos(\alpha) & -\cos(\alpha) & -\cos(\alpha) & \cos(\alpha) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -l_5 & l_5 & -l_5 & l_5 & l_2 & -l_2 & l_2 & -l_2 \\ 0 & 0 & 0 & 0 & l_1 & l_1 & -l_3 & -l_3 \\ -l_4 & l_4 & -l_6 & l_6 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.3.4}$$

where $\alpha = 45°$ is the angle of the thrusters in the xy-plane and l_i is the arm that creates moment in roll, pitch and yaw. The moment arms are given by the location of the thruster relative to the CO. The moment arms are as given in

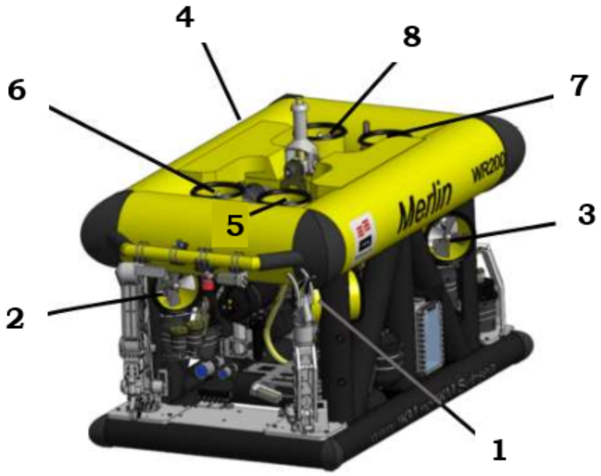


Figure 3.2: The location of the thrusters on Merlin WR200 as they are in the simulator at IKM Subsea. Image courtesy of [Knausgård, 2013]

[Knausgård, 2013]:

$$
\begin{aligned}
l_1 &= 0.73 \text{ m} \\
l_2 &= 0.24 \text{ m} \\
l_3 &= 0.73 \text{ m} \\
l_4 &= 0.84 \text{ m} \\
l_5 &= 0.10 \text{ m} \\
l_6 &= 0.84 \text{ m}
\end{aligned}
$$

See figure 3.3 and 3.4. The thrust allocation can be solved by using [Fossen, 2011, p. 405]:

$$\boldsymbol{u} = \boldsymbol{K}^{-1}\boldsymbol{T}^{\dagger}\boldsymbol{\tau} \tag{3.3.5}$$

where $\boldsymbol{T}^{\dagger}$ is the *Moore-Penrose pseudo inverse* of $\boldsymbol{T}$. All thrusters are weighted equally.

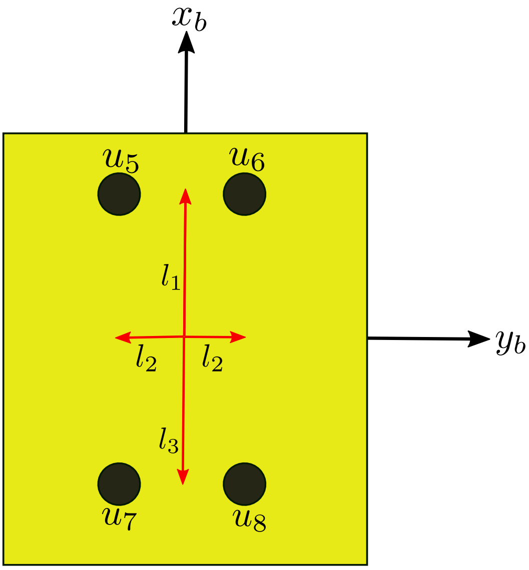


Figure 3.3: Moment arms l_1, l_2 and l_3.

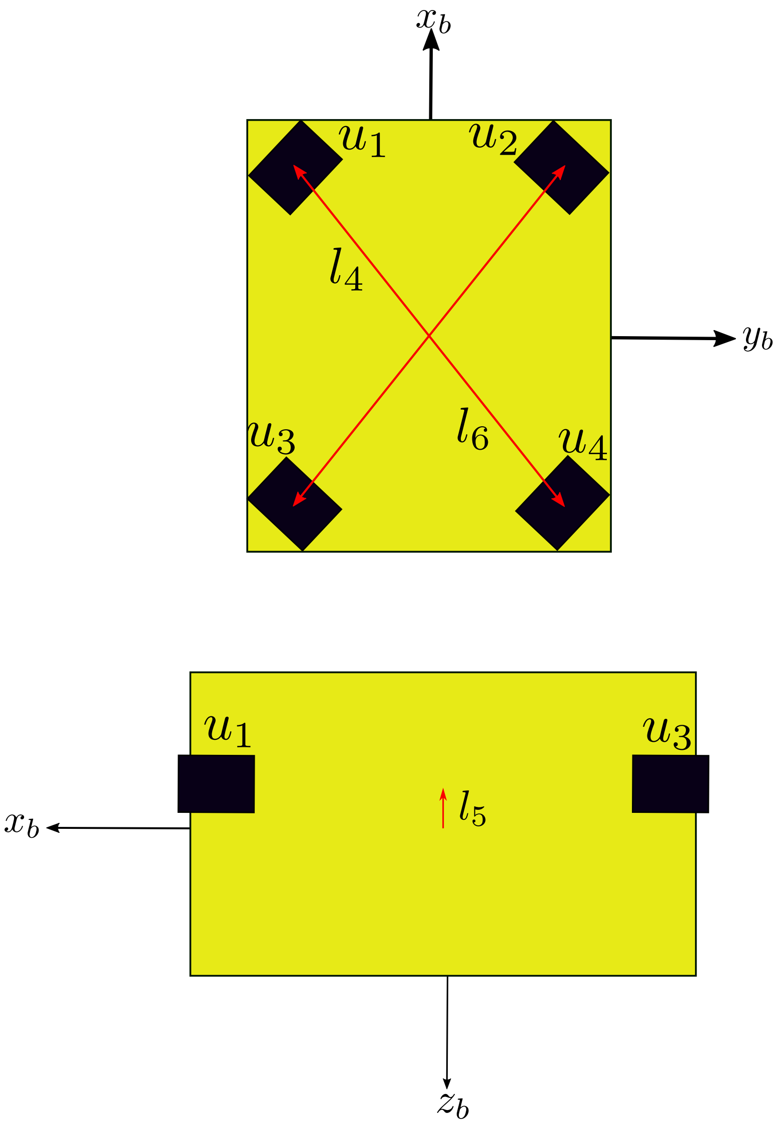


Figure 3.4: Moment arms l_4, l_5 and l_6.

Chapter 4

DP and path following control system

A novel control law has been developed for position and velocity control of the Merlin WR200. The controller is a Parameter and Current Adapting Backstepping controller (P-CABS). Since the parameters of Merlin WR200 are not known exactly, emphasis was put on not using them in the control law directly, but on adapting them and by doing so securing that the controller is robust to parameter uncertainty.

4.1 Assumptions and simplifications

4.1.1 Assumptions

The following assumptions are made in the development of the controller:

Assumption 4.1: *The vessel parameters $\boldsymbol{M}$ and $\boldsymbol{D}$ are constant such that $\dot{\boldsymbol{M}} = \dot{\boldsymbol{D}} = \boldsymbol{0}$. The gravitational and buoyancy forces are constant such that $\dot{\boldsymbol{g}}(\boldsymbol{\eta}) = 0$ The current is considered constant in $\{n\}$ with $\dot{\boldsymbol{\nu}}_c = 0$.*

Assumption 4.2: *The desired velocities and the reference trajectories are bounded. The desired velocities are passed through reference models of high enough order to ensure continuous differentiability.*

Assumption 4.3: *The velocities and accelerations are bounded*

4.1.2 Global diffeomorphism

In order to simplify notation and increase readability throughout the development of the controller, a global diffeomorphism is applied as in [Holden and Pettersen, 2007]. The vessels position and attitudes in $\{n\}$ is rotated to the *Vessel Parallel* coordinate system as described in [Fossen, 2011]. The diffeomorphism is defined as $\boldsymbol{\eta}_p \in \mathbb{R}^6$ and Assumption 3.9 is applied:

$$\boldsymbol{\eta}_p := \boldsymbol{P}^T(\psi)\begin{bmatrix}\boldsymbol{p}_p\\ \boldsymbol{\Theta}\end{bmatrix} := \begin{bmatrix}\boldsymbol{R}^T(\psi) & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}\\ \boldsymbol{\Theta}\end{bmatrix} = \boldsymbol{T}(\boldsymbol{\eta}) \tag{4.1.1}$$

The inverse of $\boldsymbol{T}(\boldsymbol{\eta})$ is:

$$\boldsymbol{\eta} = \boldsymbol{P}(\psi)\begin{bmatrix}\boldsymbol{p}\\ \boldsymbol{\Theta}\end{bmatrix} = \begin{bmatrix}\boldsymbol{R}(\psi) & \boldsymbol{0}_{3\times3}\\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3}\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_p\\ \boldsymbol{\Theta}\end{bmatrix} \tag{4.1.2}$$

Since both $\boldsymbol{T}$ and $\boldsymbol{T}^{-1}$ exists and the derivatives are continuous for all $\boldsymbol{\eta}$ and $\boldsymbol{\eta}_p$ the diffeomorphism is global.

The time derivative of $\boldsymbol{\eta}_p$ is (dependencies are skipped):

$$\dot{\boldsymbol{\eta}}_p = \begin{bmatrix}\boldsymbol{R}^T\dot{\boldsymbol{p}} + \dot{\boldsymbol{R}}^T\boldsymbol{p}\\ \dot{\boldsymbol{\Theta}}\end{bmatrix} \tag{4.1.3}$$

using $\dot{\boldsymbol{R}}^T = -\boldsymbol{S}(\boldsymbol{w})\boldsymbol{R}^T$[Egeland and Gravdahl, 2002, p.240] and $\boldsymbol{p} = \boldsymbol{R}\boldsymbol{p}_p$we get:

$$\begin{aligned}\dot{\boldsymbol{\eta}}_p &= \begin{bmatrix}\boldsymbol{R}^T\dot{\boldsymbol{p}} - \boldsymbol{S}(\boldsymbol{w})\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{p}_p\\ \dot{\boldsymbol{\Theta}}\end{bmatrix}\\ \dot{\boldsymbol{\eta}}_p &= \begin{bmatrix}\boldsymbol{v} - \boldsymbol{S}(\boldsymbol{w})\boldsymbol{p}_p\\ \dot{\boldsymbol{\Theta}}\end{bmatrix}\\ \dot{\boldsymbol{\eta}}_p &= \boldsymbol{\nu} - \begin{bmatrix}\boldsymbol{S}(\boldsymbol{w})\boldsymbol{p}_p\\ \boldsymbol{0}_{3\times1}\end{bmatrix}\end{aligned} \tag{4.1.4}$$

4.2 Control law

The position and velocity errors are defined as:

$$\begin{bmatrix}\boldsymbol{e}_1(t)\\ \boldsymbol{e}_2(t)\end{bmatrix} := \begin{bmatrix}\boldsymbol{P}^T(\psi)(\boldsymbol{\eta}(t) - \boldsymbol{\eta}_d(t))\\ \boldsymbol{\nu}(t) - \boldsymbol{\nu}_d(t)\end{bmatrix} = \begin{bmatrix}\tilde{\boldsymbol{\eta}}_p(t)\\ \tilde{\boldsymbol{\nu}}(t)\end{bmatrix} \tag{4.2.1}$$

Where $\boldsymbol{\eta}_d(t)$ is the desired position and $\boldsymbol{\nu}_d(t)$ is the desired velocity.

4.2.1 Stabilizing position and attitude

First we look at the position and attitude error(dependencies are skipped):

$$\boldsymbol{e}_1 = \begin{bmatrix} \boldsymbol{R}^T(\boldsymbol{p} - \boldsymbol{p}_d) \\ \boldsymbol{\Theta} - \boldsymbol{\Theta}_d \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{p}}_p \\ \tilde{\boldsymbol{\Theta}} \end{bmatrix} \tag{4.2.2}$$

The derivative of this error is:

$$\dot{\boldsymbol{e}}_1 = \begin{bmatrix} \tilde{\boldsymbol{v}} - \boldsymbol{S}(\tilde{\boldsymbol{w}})\tilde{\boldsymbol{p}}_p \\ \tilde{\boldsymbol{w}} \end{bmatrix} \tag{4.2.3}$$

Using a Lyapunov function candidate V_1 defined as:

$$V_1(\boldsymbol{e}_1, t) := \frac{1}{2}\boldsymbol{e}_1^T\boldsymbol{e}_1 \geq 0 \ \forall \ \boldsymbol{e}_1 \tag{4.2.4}$$

The time derivative is:

$$\begin{aligned} \dot{V}_1(\boldsymbol{e}_1, t) &= \boldsymbol{e}_1^T\dot{\boldsymbol{e}}_1 \\ \dot{V}_1(\boldsymbol{e}_1, t) &= \begin{bmatrix} \tilde{\boldsymbol{p}}_p & \tilde{\boldsymbol{\theta}} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{v}} - \boldsymbol{S}(\tilde{\boldsymbol{w}})\tilde{\boldsymbol{p}}_p \\ \tilde{\boldsymbol{w}} \end{bmatrix} \\ \dot{V}_1(\boldsymbol{e}_1, t) &= \begin{bmatrix} \tilde{\boldsymbol{p}}_p\tilde{\boldsymbol{v}} - \tilde{\boldsymbol{p}}_p\boldsymbol{S}(\tilde{\boldsymbol{w}})\tilde{\boldsymbol{p}}_p \\ \tilde{\boldsymbol{\theta}}\tilde{\boldsymbol{w}} \end{bmatrix} \end{aligned} \tag{4.2.5}$$

Since $\tilde{\boldsymbol{p}}_p\boldsymbol{S}(\tilde{\boldsymbol{w}})\tilde{\boldsymbol{p}}_p = 0$ for all $\tilde{\boldsymbol{p}}_p$ and $\tilde{\boldsymbol{w}}$, equation (4.2.5) reduces to:

$$\begin{aligned} \dot{V}_1(\boldsymbol{e}_1, t) &= \begin{bmatrix} \tilde{\boldsymbol{p}}_p & \tilde{\boldsymbol{\theta}} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{w}} \end{bmatrix} \\ \dot{V}_1(\boldsymbol{e}_1, t) &= \boldsymbol{e}_1^T\boldsymbol{e}_2 \end{aligned} \tag{4.2.6}$$

Where $\boldsymbol{e}_2 := \begin{bmatrix} \tilde{\boldsymbol{v}} & \tilde{\boldsymbol{\theta}} \end{bmatrix}^T$. It is clear that by choosing $\boldsymbol{e}_2 = -\boldsymbol{K}_1\boldsymbol{e}_1$ where $\boldsymbol{K}_1 = \boldsymbol{K}_1^T > 0 \in \mathbb{R}^{6\times 6}$ as virtual input, (4.2.6) becomes:

$$\dot{V}_1(\boldsymbol{e}_1, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 \leq -\lambda_{min}(\boldsymbol{K}_1)\boldsymbol{e}_1^T\boldsymbol{e}_1 \tag{4.2.7}$$

Where $\lambda_{min}(\boldsymbol{K}_1)$ is the smallest eigenvalue of $\boldsymbol{K}_1$. According to [Khalil, 2002, Theorem 4.10] the equilibrium point $\boldsymbol{e}_1 = \boldsymbol{0}$ is globally exponentially stable in the state space of $\boldsymbol{e}_1$ under Assumption 3.9 and if the input $\boldsymbol{e}_2$ can be made available instantaneously. The latter is not possible in reality due to the dynamics of $\boldsymbol{e}_2$ so global exponential stability exists only in theory.

4.2.2 Stabilizing velocities and choosing control input

We now look at $\boldsymbol{e}_2$(dependencies are skipped):

$$\boldsymbol{e}_2 = \begin{bmatrix} \tilde{\boldsymbol{v}} \\ \tilde{\boldsymbol{w}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v} - \boldsymbol{v}_d \\ \boldsymbol{w} - \boldsymbol{w}_d \end{bmatrix} = \boldsymbol{\nu} - \boldsymbol{\nu}_d = \tilde{\boldsymbol{\nu}} \tag{4.2.8}$$

From equation (4.2.6), we know that $\boldsymbol{e}_2$ has to be equal to $-\boldsymbol{K}_1\boldsymbol{e}_1$, so $\boldsymbol{\nu}_d$ is chosen as:

$$\boldsymbol{\nu}_d = \boldsymbol{\nu}_v + \boldsymbol{K}_1\boldsymbol{e}_1 \tag{4.2.9}$$

with $\boldsymbol{\nu}_v$ as a virtual velocity. If a control can be chosen such that $\boldsymbol{\nu} \to \boldsymbol{\nu}_v$ we will have $\boldsymbol{e}_2 = -\boldsymbol{K}_1\boldsymbol{e}_1$, which is what we want. A new variable $\bar{\boldsymbol{e}}_2 := \boldsymbol{\nu} - \boldsymbol{\nu}_v$ is defined along with a new Lyapunov function candidate:

$$V_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) := V_1(\boldsymbol{e}_1) + \frac{1}{2}\bar{\boldsymbol{e}}_2^T\boldsymbol{M}\bar{\boldsymbol{e}}_2 \geq 0 \ \forall \ \bar{\boldsymbol{e}}_2 \tag{4.2.10}$$

The time derivative of V_2 is:

$$\begin{aligned}\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) &= \dot{V}_1(\boldsymbol{e}_1) + \bar{\boldsymbol{e}}_2^T\boldsymbol{M}\dot{\bar{\boldsymbol{e}}}_2 \\ \dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) &= -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 + \bar{\boldsymbol{e}}_2^T\boldsymbol{M}(\dot{\boldsymbol{\nu}} - \dot{\boldsymbol{\nu}}_v)\end{aligned} \tag{4.2.11}$$

Perfectly known parameters and no current

If the system is modelled without considering the current disturbance forces $\boldsymbol{\tau}_c$, the equation for $\dot{\boldsymbol{\nu}}$ from (3.2.6) can be written as:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}(-\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\tau}) \tag{4.2.12}$$

Inserting equation (4.2.12) into (4.2.11) gives:

$$\begin{aligned}\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 + &\\ \bar{\boldsymbol{e}}_2^T(-\boldsymbol{M}\dot{\boldsymbol{\nu}}_v - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\tau})&\end{aligned} \tag{4.2.13}$$

If perfect parameter knowledge is assumed, $\boldsymbol{\tau}$ can be chosen as a feedback linearizing controller:

$$\boldsymbol{\tau} = -\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \boldsymbol{M}\dot{\boldsymbol{\nu}}_v + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) \tag{4.2.14}$$

with $\boldsymbol{K}_2 = \boldsymbol{K}_2^T > 0 \in \mathbb{R}^{6\times 6}$ giving:

$$\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 \tag{4.2.15}$$

This can be rewritten as:

$$\dot{V}_2(\boldsymbol{e}, t) = -\begin{bmatrix}\boldsymbol{e}_1^T & \bar{\boldsymbol{e}}_2^T\end{bmatrix}\begin{bmatrix}\boldsymbol{K}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_1 \\ \bar{\boldsymbol{e}}_2\end{bmatrix} = -\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} \tag{4.2.16}$$

with $\boldsymbol{e} = \begin{bmatrix}\boldsymbol{e}_1^T & \bar{\boldsymbol{e}}_2^T\end{bmatrix}^T$. $\boldsymbol{Q}$ is positive definite due to the choices of $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$. According to [Khalil, 2002, Theorem 4.10], $\dot{V}_2(\boldsymbol{e}, t)$ is negative definite and the equilibrium point $\boldsymbol{e} = 0$ is globally exponentially stable in the state space of $\boldsymbol{e}_1$, $\boldsymbol{e}_2$. This is only valid under Assumption 3.9.

Input-to-State stability

The overall controlled system is Input-to-State stable from unmodelled disturbances $\boldsymbol{\tau}_d$ with the controller from (4.2.14). Consider the system with an unmodeled disturbance $\boldsymbol{\tau}_d$:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}(-\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - g(\boldsymbol{\eta}) + \boldsymbol{\tau} + \boldsymbol{\tau}_d) \tag{4.2.17}$$

inserting the controller from (4.2.14) gives a system on the form:

$$\boldsymbol{M}\dot{\bar{\boldsymbol{e}}}_2 = -\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \boldsymbol{\tau}_d \tag{4.2.18}$$

The equation (4.2.16) now becomes:

$$\dot{V}_2(\boldsymbol{e}, t) = -\boldsymbol{e}^T\boldsymbol{Q}\boldsymbol{e} + \boldsymbol{e}^T\boldsymbol{\tau}_d \tag{4.2.19}$$

$$\dot{V}_2(\boldsymbol{e}, t) \leq -\lambda_Q||\boldsymbol{e}||^2 + ||\boldsymbol{e}||||\boldsymbol{\tau}_d|| \tag{4.2.20}$$

$$\dot{V}_2(\boldsymbol{e}, t) \leq -(1-\gamma)\lambda_Q||\boldsymbol{e}||^2 \;\; \forall \;\; ||\boldsymbol{e}|| \geq \frac{||\boldsymbol{\tau}_d||}{\gamma\lambda_Q} \;, 0 < \gamma < 1 \tag{4.2.21}$$

where λ_Q is the smallest eigenvalue of $\boldsymbol{Q}$ and $||\cdot||$ is the Euclidian norm. According to [Khalil, 2002, Theorem 4.19] the system (4.2.18) is Input-to-State stable.

Unknown parameters and current

It is unlikely that the system parameters are known exactly since they may vary with different sea states [Sørensen, 2013]. The current disturbance may also be modeled. Consider a control plant model with current:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}(-\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\tau} + \boldsymbol{\tau}_c) \tag{4.2.22}$$

Equation (4.2.11) becomes:

$$\begin{aligned}\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 + &\\ \bar{\boldsymbol{e}}_2^T(-\boldsymbol{M}\dot{\boldsymbol{\nu}}_v - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \boldsymbol{\tau} + \boldsymbol{\tau}_c)&\end{aligned} \tag{4.2.23}$$

The expression in the last parenthesis containing the parameters $\boldsymbol{M}$, $\boldsymbol{D}(\boldsymbol{\nu})$, $\boldsymbol{C}(\boldsymbol{\nu})$ the gravitational and buoyancy forces $\boldsymbol{g}(\boldsymbol{\eta})$ and the current disturbance $\boldsymbol{\tau}_c$ can be expressed in a linear parametrized form as:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}_v + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) - \boldsymbol{\tau}_c = \boldsymbol{\phi}^T(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})\boldsymbol{\theta} \tag{4.2.24}$$

Where $\boldsymbol{\phi} \in \mathbb{R}^{6\times 11}$ is a matrix of known, bounded signals under Assumptions 4.2 and 4.3. n is the number of DOFs in the system. $\boldsymbol{\theta} \in \mathbb{R}^{11}$ is a vector of unknown system parameters, gravitational and buoyancy forces and current disturbances. Equation (4.2.23) reduces to:

$$\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 + \bar{\boldsymbol{e}}_2^T(\boldsymbol{\tau} - \boldsymbol{\phi}^T\boldsymbol{\theta}) \tag{4.2.25}$$

The control input $\boldsymbol{\tau}$ is chosen as:

$$\boldsymbol{\tau} = -\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \boldsymbol{\phi}^T\hat{\boldsymbol{\theta}} \tag{4.2.26}$$

with $\boldsymbol{K}_2 = \boldsymbol{K}_2^T > 0 \in \mathbb{R}^{6\times 6}$ giving:

$$\dot{V}_2(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \bar{\boldsymbol{e}}_2^T\boldsymbol{\phi}^T\tilde{\boldsymbol{\theta}} \tag{4.2.27}$$

A third Lyapunov function candidate is needed in order to prove stability. This function is chosen as:

$$V_3(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, \tilde{\boldsymbol{\theta}}, t) = V_2 + \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} \ \geq 0\,\forall\,\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, \tilde{\boldsymbol{\theta}} \tag{4.2.28}$$

With $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^T > 0 \in \mathbb{R}^{6\times 6}$. The time derivative of V_3 is:

$$\begin{aligned}\dot{V}_3(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, \tilde{\boldsymbol{\theta}}, t) &= -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \bar{\boldsymbol{e}}_2^T\boldsymbol{\phi}^T\tilde{\boldsymbol{\theta}} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}} \\ \dot{V}_3(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, \tilde{\boldsymbol{\theta}}, t) &= -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 + \tilde{\boldsymbol{\theta}}^T(\boldsymbol{\phi}\bar{\boldsymbol{e}}_2 + \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}})\end{aligned} \tag{4.2.29}$$

As mentioned in Assumption 4.1, the vessel parameters, gravitational forces and the current disturbance is considered constant such that the derivatives are equal to zero. This means that $\dot{\tilde{\boldsymbol{\theta}}} = \dot{\hat{\boldsymbol{\theta}}} - \dot{\boldsymbol{\theta}} = \dot{\hat{\boldsymbol{\theta}}}$. By choosing

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{\phi}\bar{\boldsymbol{e}}_2 \tag{4.2.30}$$

the function in (4.2.29) reduces to:

$$\dot{V}_3(\boldsymbol{e}_1, \bar{\boldsymbol{e}}_2, \tilde{\boldsymbol{\theta}}, t) = -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2 \tag{4.2.31}$$

Which is negative semi-definite in the state space $\{\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2,\ \tilde{\boldsymbol{\theta}}\}$. Stability of the state space $\{\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2\}$ is proven by using Lemma 2.1 and Lemma 2.2.

The solution to the Lyapunov function candidate V_3 in the state space $\{\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2\}$ is:

$$V_3(\boldsymbol{e}_1(t),\ \bar{\boldsymbol{e}}_2(t),\ t) \leq V_3(\boldsymbol{e}_1(0),\ \bar{\boldsymbol{e}}_2(0),\ 0)\ \forall\ t \geq 0 \tag{4.2.32}$$

due to the negative semi-definiteness of $\dot{V}_3$ in this state space. Since the initial error is bounded, the states $\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2 \in \mathcal{L}_\infty$. Looking at $\dot{V}_3$ it is clear that it is a function of two signals in $\mathcal{L}_\infty$, each multiplied by its own transpose and a respective constant. This implies that $\dot{V}_3 \in \mathcal{L}_\infty$.

By taking the integral of $\dot{V}_3$ we get:

$$\int_0^\infty \dot{V}_3\,\mathrm{d}t = V_3(\infty) - V_3(0)\ <\ \infty \tag{4.2.33}$$

We know that $V_3(\infty) < \infty$ because $V_3 \geq 0$ and $\dot{V}_3 \leq 0$. V_3 converges to a finite value in the state space $\{\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2\}$ as $t \to \infty$. Inserting for V_3 in (4.2.33) gives

$$\int_0^\infty -\boldsymbol{e}_1^T\boldsymbol{K}_1\boldsymbol{e}_1 - \bar{\boldsymbol{e}}_2^T\boldsymbol{K}_2\bar{\boldsymbol{e}}_2\,\mathrm{d}t < \infty \tag{4.2.34}$$

which can be rewritten as:

$$-\boldsymbol{K}_1 \int_0^{\infty} \boldsymbol{e}_1^T \boldsymbol{e}_1 \, \mathrm{d}t - \boldsymbol{K}_2 \int_0^{\infty} \bar{\boldsymbol{e}}_2^T \bar{\boldsymbol{e}}_2 \, \mathrm{d}t \ < \infty \tag{4.2.35}$$

The integrands are ≥ 0, therefore

$$\begin{aligned} \sqrt{\int_0^{\infty} \boldsymbol{e}_1^T \boldsymbol{e}_1} < \infty \\ \sqrt{\int_0^{\infty} \bar{\boldsymbol{e}}_2^T \bar{\boldsymbol{e}}_2} < \infty \end{aligned} \tag{4.2.36}$$

and by utilizing (2.9.2), $\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2 \in \mathcal{L}_2$. Thus, $\boldsymbol{e}_1,\ \bar{\boldsymbol{e}}_2 \to 0$ as $t \to \infty$.

Convergence to 0 and by that asymptotic stability, cannot be proven for the whole state space $\{\boldsymbol{e}_1,\ \bar{\boldsymbol{e_2}},\ \tilde{\boldsymbol{\theta}}\}$ because $\tilde{\boldsymbol{\theta}}$ is only guaranteed to be bounded. This is seen from the choice of $\dot{\tilde{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\boldsymbol{\phi}\bar{\boldsymbol{e}}_2$. $\boldsymbol{\Gamma}$ is bounded, $\boldsymbol{\phi}$ is bounded and $\bar{\boldsymbol{e}}_2 \to 0$ as $t \to \infty$. This implies that $\dot{\tilde{\boldsymbol{\theta}}} \to 0$ as $t \to \infty$ and hence $\tilde{\boldsymbol{\theta}} \in \mathcal{L}_\infty$. The same properties for $\tilde{\boldsymbol{\theta}}$ and $\dot{\tilde{\boldsymbol{\theta}}}$ are presented in [Ioannou and Sun, 2012, Table 4.3 C] when the estimation model is in linear parametrized form as in (4.2.24) and the update law is chosen as in (4.2.30).

4.3 Path following control system

4.3.1 Path following problem

The path following problem is described in [Skjetne et al., 2002] as two control objectives:

1. **The geometric task:** force the state x to converge to a desired path $\xi(\omega(t))$,

$$\lim_{t \to \infty} [x(t) - \xi(\omega(t))] = 0 \tag{4.3.1}$$

2. **The Speed Assignment Task:** force the speed $\dot{\omega}$ to converge to a desired speed v_d,

$$\lim_{t \to \infty} [\dot{\omega}(t) - v_d(\omega, t)] = 0 \tag{4.3.2}$$

In [Fossen et al., 2003] these objectives are defined for an underactuated marine craft in 3DOFs using LOS guidance as:

1. **LOS Geometric task:** Force the vessel position $\boldsymbol{p} = [x, y]^T$ to converge to a desired path by forcing the yaw angle ψ to converge to the LOS angle:

$$\psi_{los} = \text{atan2}\left(y_{los} - y, x_{los} - x\right) \tag{4.3.3}$$

   where the LOS position $\boldsymbol{p}_{los} = [x_{los}, y_{los}]^T$ is the point along the path which the vessel should point to. See figure 2.7.

2. **Dynamic Task:** Force the speed u to converge to a desired speed assignment u_d, that is:

$$\lim_{t \to \infty} [u(t) - u_d(t)] = 0 \tag{4.3.4}$$

   where u_d is the desired speed of the craft along the body-fixed x-axis.

In this thesis, the vessel is a fully actuated underwater vehicle operating in 6 DOFs, but because of Assumption 3.9 the control problem is reduced to 4 DOFs. The vessels depth must be considered as well as the x and y positions. For path following tasks in this thesis it is assumed that the vessel operates at a constant depth at all times, so the path following problem is only considered in 3 DOFs.

4.3.2 Reverse path

IKM Subsea desired a path following system that was based on the already driven path of the ROV, a kind of reverse path system. The path is not generated by the operator by choosing waypoints but by the ROV based on its position and the distance to the previous waypoint. When the operator enables the reverse path mode, the ROV should return to the origin by manoeuvring through the stored waypoints. This mode may be used if the operator has manoeuvred the ROV through a narrow area and is unable to repeat the same path backwards by him or herself or if the operator has conducted a pipe survey over a longer distance and simply wants the ROV to return to the origin on its own. It is important that the heading and velocity is chosen so the ROV does not crash. If the ROV is in a narrow area as in figure 4.1 it may be impossible to turn it around and hence the ROV should reverse along the path with the heading the same way it was when it was moving into the narrow area. Since the Merlin WR200 is fully actuated, it is able to move in all DOFs. This is what makes the reverse path mode possible.

Storing waypoints

The ROV needs to store waypoints as it is moving through the ocean space. These waypoints are chosen by defining a circle of acceptance for adding waypoints:

$$\sqrt{(x_k - x(t))^2 + (y_k - y(t))^2} \geq R_{add}^2 \tag{4.3.5}$$

Where $R_{add} > 0$ is the circle of acceptance radius.

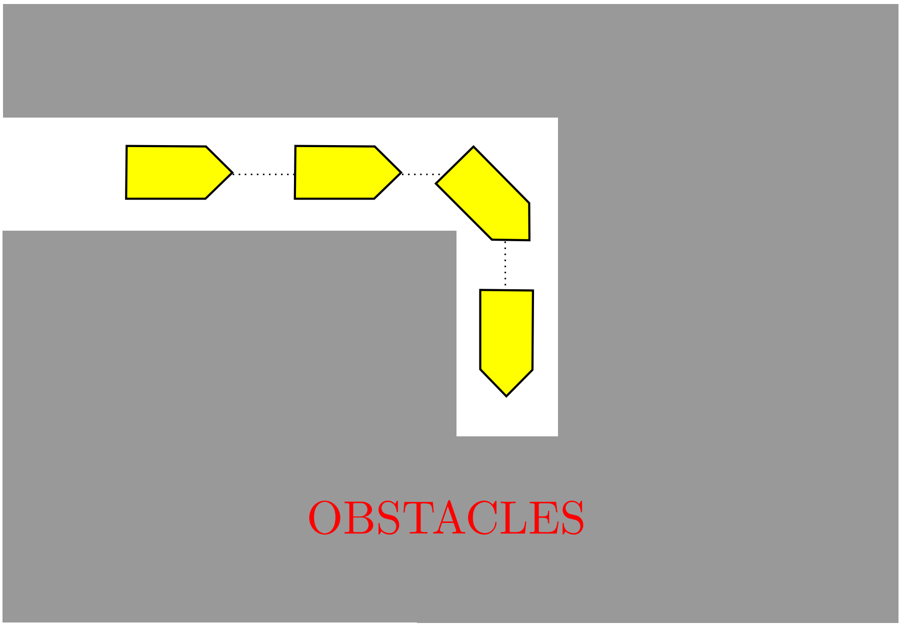


Figure 4.1: The ROV has manoeuvred into a narrow area and cannot turn around.

Ensuring correct heading

The regular LOS guidance law is designed for underactuated crafts that are not able to reverse. This means that if the waypoint is behind the craft it must circle around in order to reach it. If the craft is in a narrow area like in figure 4.1 it may not be able to perform this circular motion and it is therefore crucial that the heading is chosen correctly and thus, the desired course cannot be defined as in equation (2.6.8). A new way to define the course angle is to look at the locations of the waypoints relative to the craft and the heading of the craft. The angle α_k can be defined as described in Algorithm 1. The algorithm checks the last 2 waypoints to find out if the ROV has moved forward or backwards. Only the waypoints along the x-axis are checked because the ROV is assumed to be moving forward when adding waypoints. This is how the operator is most likely operating the ROV since the cameras are pointing forwards. This can be extended to also include sideways movement with small modifications to the algorithm. As seen in Algorithm 1, the heading angle is also checked to ensure that the ROV as moved forward or backwards.

Algorithm 1 Calculate α_k

```
if WP(x, i)-WP(x, i-1) $< 0$ AND abs($\psi$) $> \frac{pi}{2}$ then
    $\alpha_k \leftarrow$ atan2(WP(y, i)-WP(y, i-1),WP(x, i)-WP(x, i-1)
else if WP(x, i)-WP(x, i-1) $< 0$ AND abs($\psi$) $\leq \frac{pi}{2}$ then
    $\alpha_k \leftarrow$ atan2(WP(y, i-1)-WP(y, i),WP(x, i-1)-WP(x, i)
else
    $\alpha_k \leftarrow$ atan2(WP(y, i)-WP(y, i-1),WP(x, i)-WP(x, i-1)
```

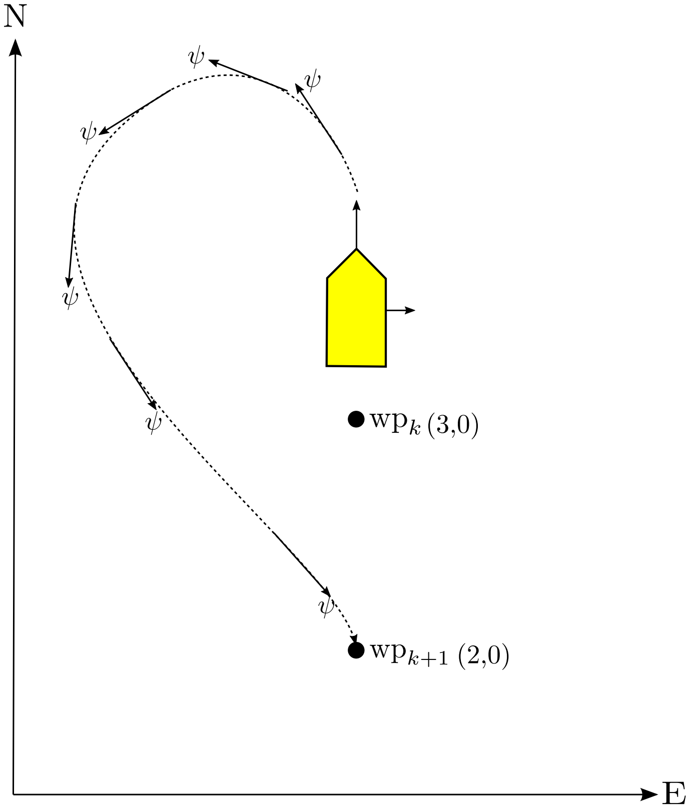


Figure 4.2: The regular LOS steering law is designed for underactuated crafts that are not able to reverse. If the waypoint is behind the craft it has to circle around to reach it. This may not be possible if the craft is in a narrow area like in figure 4.1.

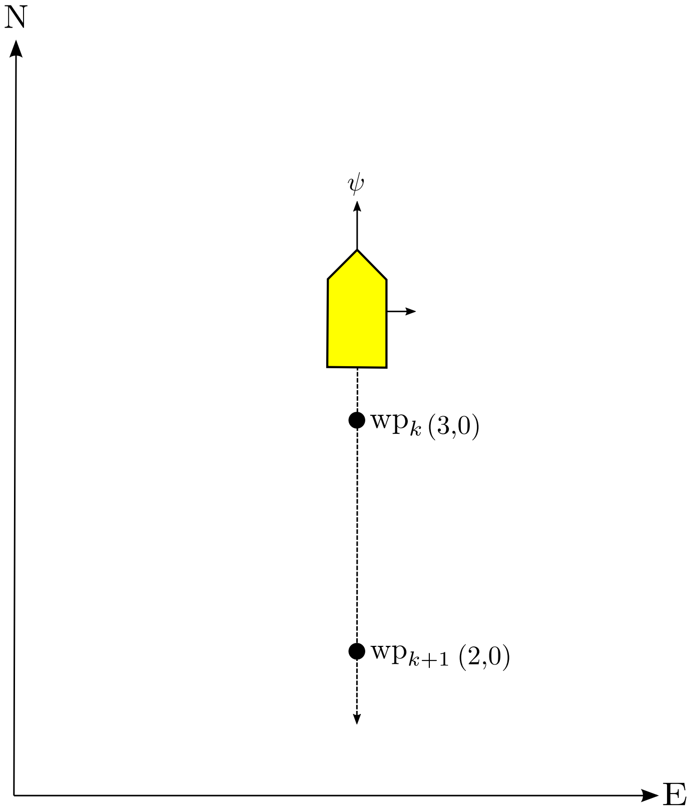


Figure 4.3: The modified LOS steering law considers the locations of the waypoints relative to the position of the craft and then calculates the heading angle and desired velocity in such a way that the craft has the same orientation it had when it generated the waypoint.

Ensuring correct velocity

By defining α_k as in Algorithm 1, the velocity will be defined correctly in equation (2.6.7) because the along track distance $s(t)$ in (2.6.4) will be positive if the next waypoint is behind the ROV and negative if it is in front of the ROV.

Example 1: Consider the stored waypoints $\mathrm{WP}_x = [1,\ 2,\ 3]$ and $\mathrm{WP}_y = [0,\ 0,\ 0]$. The ROV is in the position $[x(t), y(t)]^T = [3.5, 0]^T$ and has a heading of $\psi = 0°$. The last stored waypoint is in $[x_k, y_k] = [3, 0]$. u_{max} is chosen as 4.3m/s and $\Delta_s = 5$. The ROV should reverse back to the origin.

$$\begin{aligned}
x_k - x_{k-1} &= 3 - 2 = 1 \quad > 0 \\
y_k - y_{k-1} &= 0 \\
\alpha &= \text{atan2}(0, 1) = 0 \\
s &= (3.5 - 3)\cos(0) + 0\sin(0) = 0.5 \\
e &= 0 \\
u_d &= -4.3\frac{0.5}{\sqrt{0.5^2 + 5^2}} = -0.42 \text{ m/s} \\
\chi_d &= \alpha + \arctan\left(\frac{0}{5}\right) = 0
\end{aligned}$$

The velocity is negative and hence the ROV will reverse and keep the correct heading angle.

4.4 Using the derived controller in a path following setting

The LOS guidance system provides setpoints for the controller. Consider the system in 4 DOFs:

$$\begin{aligned}
\dot{\boldsymbol{\eta}}_{pl} &= \boldsymbol{\nu}_l \\
\boldsymbol{M}\dot{\boldsymbol{\nu}}_l + \boldsymbol{C}(\boldsymbol{\nu}_l)\boldsymbol{\nu}_l + \boldsymbol{D}(\boldsymbol{\nu}_l)\boldsymbol{\nu}_l\boldsymbol{g}(\boldsymbol{\eta}_l) &= \boldsymbol{\tau}_l - \boldsymbol{\tau}_C
\end{aligned} \tag{4.4.1}$$

Where $\boldsymbol{\eta}_{pl}$ is the vessel parallel representation of the position and attitude in 4 DOFs and $\boldsymbol{\nu}_l = \begin{bmatrix} u & v & w & r \end{bmatrix}^T$. The goal of the LOS path following control system is not to control the position in N and E, but the respective velocities. The position and attitude control is that of the depth Z and the yaw angle ψ respectively. The control law is therefore split in two. One controller for the depth and heading angle and one for the velocities.

4.4.1 Depth and heading controller

New error variables for the depth and heading are defined with the subscript l

$$\boldsymbol{e}_{1l} := \begin{bmatrix} Z(t) - Z_d(t) \\ \psi(t) - \psi_d(t) \end{bmatrix} \tag{4.4.2}$$

$$\boldsymbol{e}_{2l} := \begin{bmatrix} w(t) - w_d(t) \\ r(t) - r_d(t) \end{bmatrix} \tag{4.4.3}$$

Consider a new Lyapunov function candidate V_{1l} (dependencies are skipped):

$$V_1(\boldsymbol{e}_{1l}, t) := \frac{1}{2}\boldsymbol{e}_{1l}^T \boldsymbol{K}_{1l} \boldsymbol{e}_{1l} \tag{4.4.4}$$

where $\boldsymbol{K}_{1l} = \boldsymbol{K}_{1l}^T > \boldsymbol{0} \in \mathbb{R}^{2\times 2}$. The time derivative is:

$$\dot{V}_1(\boldsymbol{e}_{1l}, t) = \boldsymbol{e}_{1l}\dot{\boldsymbol{e}}_{1l} = \boldsymbol{e}_{1l}\boldsymbol{e}_{2l} \tag{4.4.5}$$

$\boldsymbol{e}_{2l}$ is then chosen the same way the variable e_2 was chosen in section 4.2 and the same procedure is followed giving a control input for the depth and heading controller as:

$$\boldsymbol{\tau}_{Z,\psi} = -\boldsymbol{K}_{2l}\bar{\boldsymbol{e}}_{2l} + \boldsymbol{\phi}_{1l}^T \hat{\boldsymbol{\theta}}_{1l} \tag{4.4.6}$$

with $\boldsymbol{K}_{2l} = \boldsymbol{K}_{2l}^T > 0 \in \mathbb{R}^{2\times 2}$, $\boldsymbol{\phi}_{1l}^T \in \mathbb{R}^{2\times 5}$ and $\boldsymbol{\theta}_{1l} \in \mathbb{R}^5$.

4.4.2 Velocity controller

A new error variable for the velocity is defined as:

$$\boldsymbol{e}_{3l} := \begin{bmatrix} u(t) - u_d(t) \\ v(t) - v_d(t) \end{bmatrix} = \boldsymbol{\nu}_l(t) - \boldsymbol{\nu}_{ld}(t) \tag{4.4.7}$$

where $\boldsymbol{\nu}_l(t) = \begin{bmatrix} u(t) & v(t) \end{bmatrix}^T$ and $\boldsymbol{\nu}_{ld}(t) = \begin{bmatrix} u_d(t) & v_d(t) \end{bmatrix}^T$. The derivative is:

$$\dot{\boldsymbol{e}}_{3l} = \begin{bmatrix} \dot{u} - \dot{u}_d \\ \dot{v} - \dot{v}_d \end{bmatrix} = \dot{\boldsymbol{\nu}}_l - \dot{\boldsymbol{\nu}}_{ld} \tag{4.4.8}$$

where u_d and v_d are the desired velocities in the body fixed x and y directions respectively. The desired velocity is calculated using (2.6.7). The desired acceleration $\dot{u}_d$ and $\dot{v}_d$ is obtained by filtering the desired velocities through reference models represented by low pass filters as described in [Fossen, 2011, Ch.10.2.1]. A Lyapunov function candidate $V_2(\boldsymbol{e}_{3l}, t)$ is defined as:

$$V_2(\boldsymbol{e}_{3l}, t) := \frac{1}{2}\boldsymbol{e}_{3l}^T \bar{\boldsymbol{M}} \boldsymbol{e}_{3l} \tag{4.4.9}$$

where $\bar{\boldsymbol{M}} = \bar{\boldsymbol{M}}^T > 0 = \text{diag}\{m - X_{\dot{u}},\ \ m - Y_{\dot{v}}\}$ and represents the x and y component of the mass matrix. The time derivative of V_3 is:

$$\begin{aligned}\dot{V}_2(\boldsymbol{e}_{3l}, t) &= \boldsymbol{e}_{3l}^T \bar{\boldsymbol{M}} \dot{\boldsymbol{e}}_{3l} \\ \dot{V}_2(\boldsymbol{e}_{3l}, t) &= \boldsymbol{e}_{3l}^T \bar{\boldsymbol{M}} (\dot{\boldsymbol{\nu}}_l - \dot{\boldsymbol{\nu}}_{ld})\end{aligned} \tag{4.4.10}$$

Inserting for $\dot{\boldsymbol{\nu}}_l$ and utilizing the linear parametrized form described in (4.2.24) but in 2 DOFs with $\dot{\boldsymbol{\nu}}_v = \dot{\boldsymbol{\nu}}_{ld}$, $\boldsymbol{\nu} = \boldsymbol{\nu}_l$ and $\boldsymbol{\eta} = \boldsymbol{\eta}_l = \begin{bmatrix} N & E \end{bmatrix}^T$ we get:

$$\dot{V}_2(\boldsymbol{e}_{3l}, t) = \boldsymbol{e}_{3l}^T \left(\boldsymbol{\tau}_{u,v} - \boldsymbol{\phi}_{2l}^T (\dot{\boldsymbol{\nu}}_l, \boldsymbol{\nu}_l, \boldsymbol{\eta}_l) \boldsymbol{\theta}_{2l} \right) \tag{4.4.11}$$

Where $\boldsymbol{\phi}_{2l} \in \mathbb{R}^{2\times 6}$ and $\boldsymbol{\theta}_{2l} \in \mathbb{R}^6$. The input is then chosen as:

$$\boldsymbol{\tau}_{u,v} = -\boldsymbol{K}_{3l} \boldsymbol{e}_{3l} + \boldsymbol{\phi}_{2l}^T \hat{\boldsymbol{\theta}}_{2l} \tag{4.4.12}$$

where $\boldsymbol{K}_{3l} = \boldsymbol{K}_{3l}^T > 0 \in \mathbb{R}^{2\times 2}$ giving

$$\dot{V}_2(\boldsymbol{e}_{3l}, t) = -\boldsymbol{e}_{3l}^T \boldsymbol{K}_{3l} \boldsymbol{e}_{3l} + \boldsymbol{e}_{3l}^T \boldsymbol{\phi}_{2l}^T \tilde{\boldsymbol{\theta}}_{2l} \tag{4.4.13}$$

Another Lyapunov function candidate is needed. This is defined as:

$$V_3(\boldsymbol{e}_{3l}, \tilde{\boldsymbol{\theta}}_{2l}, t) := V_2 + \frac{1}{2} \tilde{\boldsymbol{\theta}}_{2l}^T \boldsymbol{\Gamma}_l^{-1} \tilde{\boldsymbol{\theta}}_{2l} \tag{4.4.14}$$

By following the same procedure as in chapter 4.2 for the unknown parameter case, we end up with a parameter update law $\dot{\boldsymbol{\theta}}_{2l} = -\boldsymbol{\Gamma}_l \boldsymbol{\phi}_{2l} \boldsymbol{e}_{3l}$ and the derivative of the Lyapunov function candidate V_3 is:

$$\dot{V}_3(\boldsymbol{e}_{3l}, \tilde{\boldsymbol{\theta}}_{2l}, t) = -\boldsymbol{e}_{3l}^T \boldsymbol{K}_{3l} \boldsymbol{e}_{3l} \tag{4.4.15}$$

Utilizing the same arguments as in chapter 4.2, $\boldsymbol{e}_3 \to 0$ as $t \to \infty$.

4.5 Path-following of predefined paths

As an additional feature, a path following system that is not dependent on the driven path of the ROV has been implemented. This is useful if IKM Subsea wants to plan a mission for the ROV. This could be following a pipe or perform a survey of an area. A technique for guiding the craft to a predefined path is by using a path fixed reference frame. This reference frame can be chosen as the *Serret-Frenet* frame and it is defined in [Fossen, 2011, Ch. 10.4.2] as:

Definition 4.1: *The Serret-Frenet frame is the virtual target defined by the projection of an actual craft on to a path-tangential reference frame(Serret-Frenet frame{SF})[...].*

The Serret-Frenet frame has axes along the tangent T, the normal N and the binormal B of the path. It is a continuously differentiable space curve in the inertial frame and for path following of curved paths it has advantages over LOS which only considers points. As can be seen in figure 4.4, there are 3 reference frames. The inertial $\{n\}$ frame, the $\{b\}$ frame and the {SF} frame. The relationship between the frames is given in [Fossen, 2011, Ch.10.4.2] as:

$$r_{b/n} = r_{SF/n} + r_{b/SF} \tag{4.5.1}$$

The time differentiation of $r_{b/SF}$ with $\{b\}$ as the moving reference frame gives:

$$\frac{{}^n d}{dt} r_{b/SF} = \frac{{}^b d}{dt} r_{b/SF} + w_{b/n} \times r_{b/SF} \tag{4.5.2}$$

such that

$$v_{b/n} = v_{SF/n} + \left(\frac{{}^b d}{dt} r_{b/SF} + w_{SF/n} \times r_{b/SF} \right) \tag{4.5.3}$$

Expressing this in {SF} gives:

$$\begin{aligned} \boldsymbol{v}_{b/n}^{SF} &= \boldsymbol{v}_{SF/n}^{SF} + \left(\frac{{}^{SF} d}{dt} \boldsymbol{r}_{b/SF}^{SF} + \boldsymbol{w}_{SF/n}^{SF} \times r_{b/SF}^{SF} \right) \\ &= \begin{bmatrix} \dot{s} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{x}_{b/SF} \\ \dot{y}_{b/SF} \\ \dot{z}_{b/SF} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \kappa \dot{s} \end{bmatrix} \times \begin{bmatrix} x_{b/SF} \\ y_{b/SF} \\ z_{b/SF} \end{bmatrix} \end{aligned} \tag{4.5.4}$$

where κ is the curvature of the path. $\boldsymbol{v}_{b/n}^{SF} = \boldsymbol{R}_b^{SF}(\boldsymbol{\Theta}_{SFb}) \begin{bmatrix} u & v & w \end{bmatrix}^T$ and $\boldsymbol{R}_b^{SF}(\boldsymbol{\Theta}_{SFb}) = \boldsymbol{R}_b^n(\boldsymbol{\Theta}_{SFn})^T \boldsymbol{R}_b^n(\boldsymbol{\Theta})$ where $\boldsymbol{\Theta}$ is the orientation of $\{b\}$ with respect to $\{n\}$ and $\boldsymbol{\Theta}_{SFn}$ is the orientation of the {SF} frame with respect to $\{n\}$. $\dot{s}$ is the speed of a virtual particle that moves along the path.

In this thesis it is assumed that the depth is kept constant using the depth controller from (4.4.6). With this in mind, rearranging the terms in (4.5.4), the kinematic relationship between the {SF} frame and body frame can be expressed as:

$$\begin{bmatrix} \dot{x}_{b/SF} \\ \dot{y}_{b/SF} \end{bmatrix} = \begin{bmatrix} \cos(\psi_{SFb}) & -\sin(\psi_{SFb}) \\ \sin(\psi_{SFb}) & \cos(\psi_{SFb}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} \dot{s} \\ 0 \end{bmatrix} - \dot{s} \begin{bmatrix} 0 & -\kappa \\ \kappa & 0 \end{bmatrix} \begin{bmatrix} x_{b/SF} \\ y_{b/SF} \end{bmatrix} \tag{4.5.5}$$

where $\psi_{SFb} = \psi - \psi_f$ is the heading of $\{b\}$ relative to the heading of the {SF} frame. The defined path is in 2DOFs only, so the reference position for the craft is the origin of the {SF} frame. The control objective becomes:

$$\begin{aligned} \lim_{t \to \infty} x_{b/SF} &= 0 \\ \lim_{t \to \infty} y_{b/SF} &= 0 \\ \lim_{t \to \infty} u - u_d &= 0 \end{aligned} \tag{4.5.6}$$

Where $x_{b/SF}$ and $y_{b/SF}$ can be seen as the difference between $\{b\}$ and the {SF} frame as shown in figure 4.4. The desired velocity of the craft as it moves along the curve is specified by u_d. How the origin of the {SF} frame propagates along the path can be chosen freely and in [Børhaug, 2008] it is suggested to choose an update law as:

$$\dot{s} = \sqrt{u_d^2 + v^2}\frac{\sqrt{\Delta^2 + x_{b/SF}^2} + x_{b/SF}}{\sqrt{\Delta^2 + x_{b/SF}^2 + y_{b/SF}^2}} \tag{4.5.7}$$

and the desired heading is proposed as:

$$\psi_d = \psi_f - \operatorname{atan}\left(\frac{v}{u_d}\right) - \operatorname{atan}\left(\frac{y_{b/SF}}{\sqrt{\Delta^2 + x_{b/SF}^2}}\right) \tag{4.5.8}$$

Δ is a constant tuning parameter.

It should be noted that this method was designed for underactuated marine crafts with no direct control in sway. To compensate the ocean current, the craft has to sideslip. Merlin WR200 is fully actuated, so it could compensate the current without sideslipping by controlling the sway velocity to $v = v_d = 0$.

In the doctoral thesis of [Børhaug, 2008], it is shown that the update law (4.5.7) and the guidance law (4.5.8) ensures exponential achievement of the control objective when using feedback linearizing surge and yaw controllers. The same behaviour cannot be expected when using the P-CABS controller, but it is assumed that the control objective will be achieved.

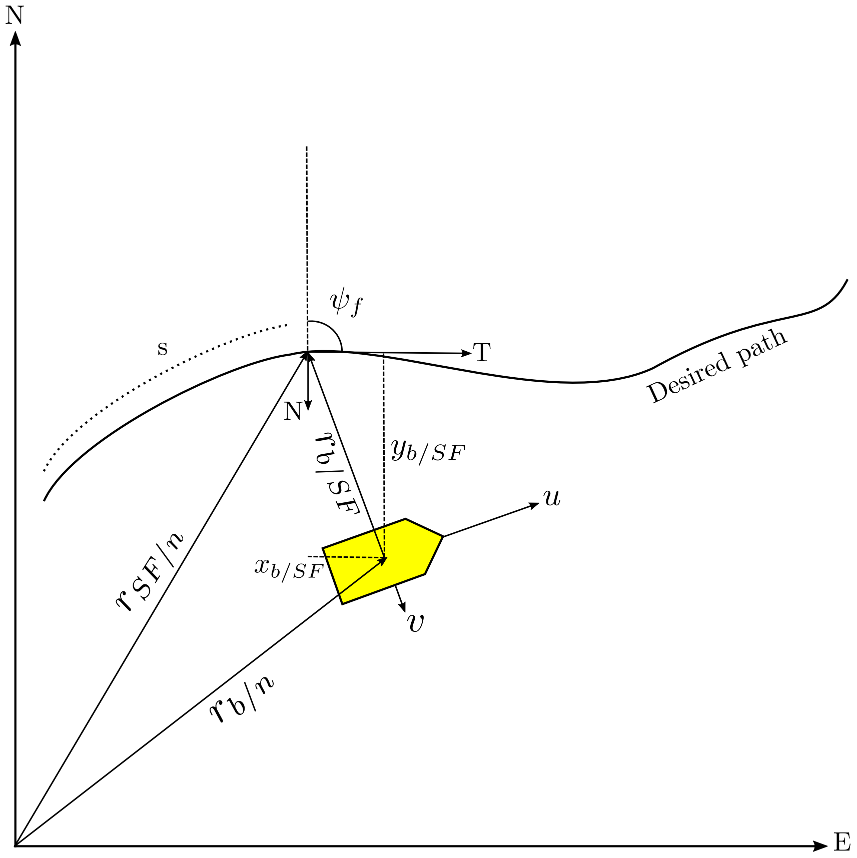


Figure 4.4: Illustration of the Serret-Frenet frame in 2D.

4.6 Joystick control

It is trivial to implement a joystick as means of deciding the reference velocity u_d and v_d. Joystick control may also implemented for the yaw rate r. This has been done in the simulations for this thesis to test the waypoint storing and reverse path functionality. The velocity controller when using the joystick takes the same form as (4.4.12) but is augmented to include the yaw rate r as well. The same stability properties apply to the controller when using joystick control.

The raw velocity signal from the joystick is filtered through a reference model to ensure smooth desired velocity $\boldsymbol{\nu}_d$ and acceleration $\dot{\boldsymbol{\nu}}_d$. See figure 4.5 and figure 4.6.

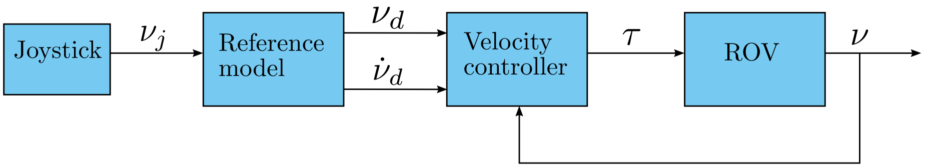


Figure 4.5: The joystick input is filtered through a reference model to ensure smooth desired velocities and accelerations.

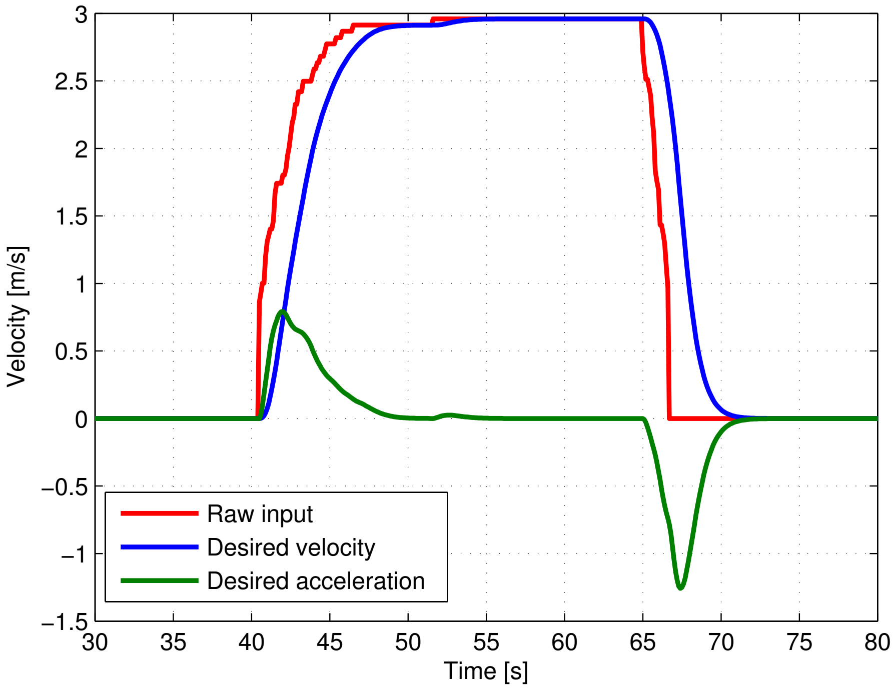


Figure 4.6: The raw input signal from the joystick is fed through the reference model which generates a smooth desired velocity and a desired acceleration.

4.7 Comparison of the derived controller to other controllers

The derived controller is similar to the controllers derived in for instance [Slotine and Weiping, 1988], [Fossen and Berge, 1997], [Holden and Pettersen, 2007], [Antonelli et al., 2003] and [Patompak and Nilkhamhang, 2012](ABSwB). The difference is that it combines the adaptive abilities of the controller from [Slotine and Weiping, 1988] with the vectorial backstepping from [Fossen and Berge, 1997]. The adaptive part handles the system parameter uncertainty, the gravitational and buoyancy forces and the current disturbance. It is combining the adaptation done in [Patompak and Nilkhamhang, 2012](ABSwB) and [Antonelli et al., 2003] and it is presented in a simplified way by using the diffeomorphism from [Holden and Pettersen, 2007].

Let us take a closer look at the ABSwB controller from [Patompak and Nilkhamhang, 2012] and the controller from [Antonelli et al., 2003] to easier present the difference.

4.7.1 ABSwB controller

The control plant model is presented as:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \Delta\boldsymbol{f} \tag{4.7.1}$$

Where $\Delta\boldsymbol{f}$ is a disturbance that satisfies $||\Delta\boldsymbol{f}(t)|| \leq \delta, \forall\, t$. The position and virtual velocity error is presented as:

$$\boldsymbol{e} := \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta} - \boldsymbol{\eta}_{df} \\ \boldsymbol{\nu} - \boldsymbol{\nu}_v \end{bmatrix} \tag{4.7.2}$$

Where $\boldsymbol{\eta}_{df}$ is the filtered desired position. The virtual velocity control is chosen as:

$$\boldsymbol{\nu}_v = -\boldsymbol{J}^{-1}(\boldsymbol{\eta})\boldsymbol{K}_1\boldsymbol{e}_1 + \boldsymbol{J}^{-1}(\boldsymbol{\eta})\dot{\boldsymbol{\eta}}_{df} \tag{4.7.3}$$

The derivative of the errors are:

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_{df} \\ \boldsymbol{M}^{-1}(\boldsymbol{\tau} + \Delta f - \boldsymbol{\phi}^T(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})\boldsymbol{\theta}) \end{bmatrix} \tag{4.7.4}$$

where the term $\boldsymbol{\phi}^T(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})\boldsymbol{\theta}$ is the linear parameterized form of the system as in (4.2.24). The control law is then proposed as:

$$\boldsymbol{\tau} = -\boldsymbol{J}^T(\boldsymbol{\eta})\boldsymbol{e}_1 - \boldsymbol{K}_2\boldsymbol{e}_2 + \boldsymbol{\phi}^T(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})\hat{\boldsymbol{\theta}} - \boldsymbol{B}_e \tag{4.7.5}$$

Where $\boldsymbol{K}_2$ is a diagonal matrix with positive constant gains, $\hat{\boldsymbol{\theta}}$ is the parameter estimates and $\boldsymbol{B}_e$ is the robustifying term that handles disturbances and is defined as:

$$\boldsymbol{B}_e := \begin{cases} \frac{\hat{\boldsymbol{\delta}}\boldsymbol{e}_2}{||\boldsymbol{e}_2||} & \text{if } \boldsymbol{e}_2 \neq 0 \\ 0 & \text{if } \boldsymbol{e}_2 = 0 \end{cases} \tag{4.7.6}$$

The update laws are:

$$\begin{aligned} \dot{\hat{\boldsymbol{\theta}}} &= -\boldsymbol{\Gamma}_\theta \boldsymbol{\phi}^T(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})\boldsymbol{e}_2 \\ \dot{\hat{\boldsymbol{\delta}}} &= \boldsymbol{\Gamma}_\delta ||\boldsymbol{e}_2|| \end{aligned} \tag{4.7.7}$$

Where $\boldsymbol{\Gamma}_\theta$ and $\boldsymbol{\Gamma}_\delta$ are positive constant matrices. The regressor matrix takes the form:

$$\boldsymbol{\phi}(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta}) = \begin{bmatrix} \dot{u}_v & ur & 0 & -uv \\ -vr & \dot{v}_v & 0 & uv \\ 0 & 0 & \dot{w}_v & 0 \\ 0 & 0 & 0 & \dot{r}_v \\ |u|u & 0 & 0 & 0 \\ 0 & |v|v & 0 & 0 \\ 0 & 0 & |w|w & 0 \\ 0 & 0 & 0 & |r|r \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.7.8}$$

And the parameter vector is:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} m - X_{\dot{u}} \\ m - Y_{\dot{v}} \\ m - Z_{\dot{w}} \\ m - N_{\dot{r}} \\ X_{|u|u} \\ Y_{|v|v} \\ Z_{|w|w} \\ N_{|r|r} \\ g_z \end{bmatrix} \tag{4.7.9}$$

The controller does not specifically adapt the current disturbance and in the article of [Patompak and Nilkhamhang, 2012] the disturbance is described as being produced by collisions and water current. The disturbance is modelled as $\Delta f = 50 + 100 \sin(0.785t)$kgf in the body x, y and z directions. The controller is able to suppress the disturbance, but only due to the discontinuous control input generated by the robustifying term $\boldsymbol{B}_e$.

The main difference between this and the developed P-CABS controller is the modeling and adaptation of the current and the discontinuous robustifying term. Also, throughout the derivation of the control law, the term $\boldsymbol{J}^{-1}(\boldsymbol{\eta})$ has to be included because the method does not utilize the diffeomorphism. This causes more variables to appear in the derivation, which may complicate the derivation.

4.7.2 Antonelli controller

In [Antonelli et al., 2003] the control plant model is presented as:

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + C(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} - \boldsymbol{\tau}_C \tag{4.7.10}$$

The error variables are presented in a vehicle fixed frame, not unlike what is done in this thesis:

$$\begin{aligned}\tilde{\boldsymbol{y}} &:= \begin{bmatrix}\boldsymbol{R}^{-1}(\psi)\tilde{\boldsymbol{\eta}} \\ \tilde{\boldsymbol{\epsilon}}\end{bmatrix} \\ \tilde{\boldsymbol{\nu}} &:= \boldsymbol{\nu} - \boldsymbol{\nu}_d\end{aligned} \tag{4.7.11}$$

where $\tilde{\boldsymbol{\eta}}$ is the position error, $\tilde{\boldsymbol{\epsilon}}$ is the quaternion based attitude error and $\tilde{\boldsymbol{\nu}}$ is the velocity error. A means of tracking the error is defined as:

$$\boldsymbol{s} := \tilde{\boldsymbol{\nu}} + \boldsymbol{K}_1\tilde{\boldsymbol{y}} \tag{4.7.12}$$

where $\boldsymbol{K}_1 \in \mathbb{R}^{6\times 6}$ is a positive definite diagonal matrix of gains. The control law is then presented as:

$$\begin{aligned}\boldsymbol{\tau} &= -\boldsymbol{K}_2\boldsymbol{s} - \boldsymbol{K}\tilde{\boldsymbol{y}} + \boldsymbol{\phi}(\boldsymbol{R}^{-1}(\psi), \boldsymbol{\eta})^T\hat{\boldsymbol{\theta}} \\ \dot{\hat{\boldsymbol{\theta}}} &= \boldsymbol{\Gamma}\boldsymbol{s}\boldsymbol{\phi}(\boldsymbol{R}^{-1}(\psi), \boldsymbol{\eta})\end{aligned} \tag{4.7.13}$$

where $\boldsymbol{K}_2, \boldsymbol{K}, \boldsymbol{\Gamma} \in \mathbb{R}^{6\times 6}$ are all positive definite and diagonal matrices of gains. $\boldsymbol{\phi}$ is the regressor matrix and $\hat{\boldsymbol{\theta}}$ is the vector of unknown disturbances caused by the ocean current and the hydrostatics. This represents a PD controller with adaptive compensation of environmental and hydrostatic disturbances. Note that the system parameters are not included in the control law, which separates it from the controller in this thesis and from [Patompak and Nilkhamhang, 2012]. The adaptive part takes the following form:

$$\boldsymbol{\phi}(\boldsymbol{R}^{-1}(\psi), \boldsymbol{\eta}) = \begin{bmatrix} 0 & 0 & 1 \\ \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \end{bmatrix} \tag{4.7.14}$$

and the unknown disturbances are:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} g_z \\ \tau_{Cx} \\ \tau_{Cy} \end{bmatrix} \tag{4.7.15}$$

This control plant model contains the same model of the current and gravitational disturbances as in this thesis, namely that they are constant in $\{n\}$. There is no adaptation of the system parameters as they are completely left out of the control law.

The main difference between this controller and the P-CABS controller is that P-CABS adapts the system parameters as well as the current disturbance.

Chapter 5

Simulation

5.1 Simulink Model

In order to test the controller and the different functionalities including the dynamic positioning and path following, a Simulink model of the whole system has been developed. The Simulink model contains the dynamic model of the Merlin WR200 based on the modeling done in chapter 3, a DP controller, velocity and heading controller for the LOS system, velocity controller for joystick control and a guidance system for storing waypoints. A 3D-world was created to get an overview of what happened to the ROV as it moved through the ocean space. The Simulink system is shown in figure 5.1.

5.1.1 Joystick block

To replicate the joystick input, an Xbox 360 controller was connected to the Simulink model and the joysticks on it was used to control the velocities and to activate the return path mode, see figure 5.2. To reduce the sensitivity of the joysticks an adjustment is added in the Simulink diagram.

Since several different controllers are used, the system needs to switch between them. The system switches to DP mode when all the joystick signal is 0. It switches to velocity control when any of the joysticks are pushed in any direction and lastly, the return path mode is activated when the blue X button is pushed. Every time a switch occurs, the integrators in the controller it is switched to is reset. This is because the integrators build compensation even though the controller is not in use.

5.1.2 Guidance system

This block contains the guidance system for the return path mode. The system adds waypoints as described in equation (4.3.5). When the return path button is pushed, the system calculates the desired heading and velocity and sends it to the velocity and heading controller. The guidance system block also contains a setpoint storing function that, when the joystick is released, stores the position and attitude and sends it as a setpoint to the DP controller.

5.1.3 Input selection block

This block chooses which of the calculated inputs that are to be used, depending on whether or not the joysticks are active or the return path is active.

5.1.4 Controller blocks

The DP controller block, the velocity controller block and the LOS controller block simply contains the controllers for the respective control system. Notice however, that there is an input *tau* to the LOS controller. This is the depth, roll and pitch control force calculated by the DP controller to keep these values as desired.

5.2 Regressor matrix and parameter vector

The control plant model used in the simulations is in 6DOF, but it is assumed that the roll and pitch motions are very small so the gravitational and buoyancy forces are as in (3.2.11). Since the current is considered irrotational and in 2 dimensions, no current disturbance is considered in heave, roll, pitch or yaw. Because the center of gravity is below the center of buoyancy, the Merlin WR200 is naturally stable in roll and pitch, but to reduce the influence of roll and pitch on other variables the roll and pitch is controlled by the same controller as the other DOFs, only without the adaptive part. This, along with assumption 3.7 reduces the mass, damping and Coriolis matrices to the following:

$$\boldsymbol{M} = \mathrm{diag}\{m - X_{\dot{u}},\ m - Y_{\dot{v}},\ m - Z_{\dot{w}},\ m - N_{\dot{r}}\} \tag{5.2.1}$$

$$\boldsymbol{D}(\boldsymbol{\nu}_l) = \mathrm{diag}\{X_{u|u|}|u|,\ Y_{v|v|}|v|,\ Z_{w|w|}|w|,\ N_{r|r|}|r|\} \tag{5.2.2}$$

$$\boldsymbol{C}(\boldsymbol{\nu}_l) = \begin{bmatrix} 0 & 0 & 0 & -(m - Y_{\dot{v}})v \\ 0 & 0 & 0 & -(m - X_{\dot{u}})u \\ 0 & 0 & 0 & 0 \\ (m - Y_{\dot{v}})v & -(m - X_{\dot{u}})u & 0 & 0 \end{bmatrix} \tag{5.2.3}$$

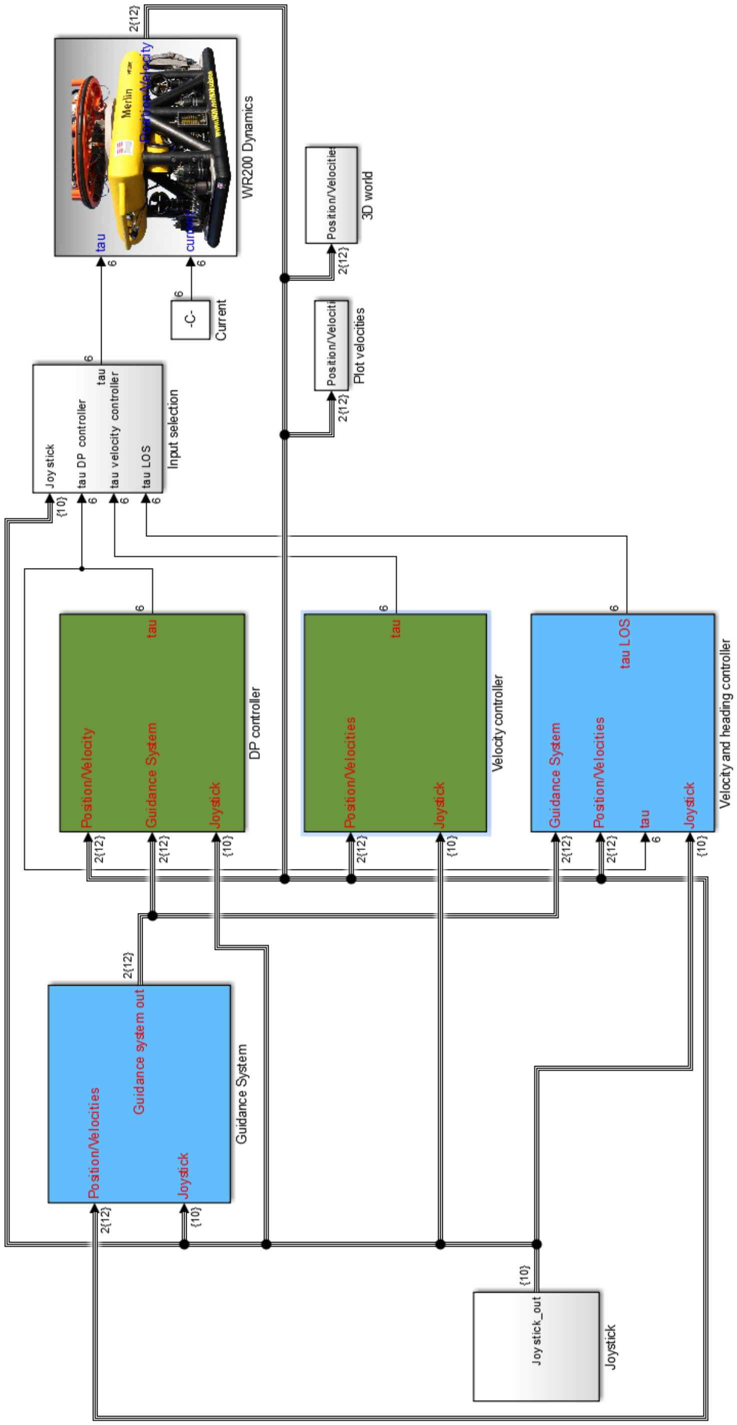


Figure 5.1: The Simulink diagram of the control system.

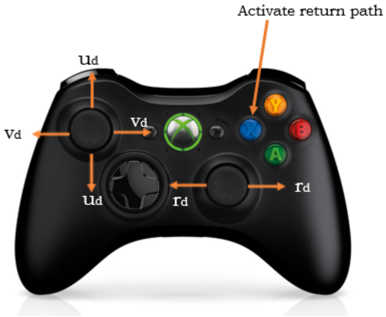


Figure 5.2: The Xbox 360 controller buttons that are used to control the Merlin WR200 in the simulation setup.

This leads to the regressor matrix $\boldsymbol{\phi}(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta})$:

$$\boldsymbol{\phi}(\dot{\boldsymbol{\nu}}_v, \boldsymbol{\nu}, \boldsymbol{\eta}) = \begin{bmatrix} \dot{u}_v & ur & 0 & -uv \\ -vr & \dot{v}_v & 0 & uv \\ 0 & 0 & \dot{w}_v & 0 \\ 0 & 0 & 0 & \dot{r}_v \\ |u|u & 0 & 0 & 0 \\ 0 & |v|v & 0 & 0 \\ 0 & 0 & |w|w & 0 \\ 0 & 0 & 0 & |r|r \\ 0 & 0 & 1 & 0 \\ -\cos(\psi) & -\sin(\psi) & 0 & 0 \\ \sin(\psi) & -\cos(\psi) & 0 & 0 \end{bmatrix} \tag{5.2.4}$$

and the parameter vector:

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} m - X_{\dot{u}} \\ m - Y_{\dot{v}} \\ m - Z_{\dot{w}} \\ m - N_{\dot{r}} \\ X_{|u|u} \\ Y_{|v|v} \\ Z_{|w|w} \\ N_{|r|r} \\ g_z \\ \tau_{cx} \\ \tau_{cy} \end{bmatrix} \tag{5.2.5}$$

5.3 Reference model

The variables $\dot{u}_v$, $\dot{v}_v$, $\dot{w}_v$ and $\dot{r}_v$ are produced by using reference models for the change in position and velocity. These reference models are motivated by the dynamics of mass-damper-spring systems as described in [Fossen, 2011, Ch. 10.2.1]. A position and attitude reference model is chosen as a third order mass-damper-spring system on the form:

$$\frac{\eta_{d_i}}{r_i^n}(s) = \frac{\omega_{n_i}^3}{s^3 + (2\zeta_i + 1)\omega_{n_i} s^2 + (2\zeta_i + 1)\omega_{ni}^2 s + \omega_{n_i}^3} \quad (i = 1, \dots, n) \tag{5.3.1}$$

where r_i^n is the specified reference signal. The different DOFs require different reference models and the values used are shown in table 5.1.

DOF	ζ	ω
X	1	1
Y	1	0.8
Z	1	1
ϕ	N/A	N/A
θ	N/A	N/A
ψ	1	0.9

Table 5.1: Values for the natural frequency and damping ratio of the reference models used in the DP test

5.4 Choosing Γ

In the parameter update law $\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\bar{\boldsymbol{e}}_2\boldsymbol{\phi}$ the gain matrix $\boldsymbol{\Gamma}$ can be chosen as a constant gain matrix by tuning or it may be time varying. By choosing it as a time varying matrix the update law becomes a *Modified Least-Squares with Forgetting Factor*, as described in [Ioannou and Sun, 2012, p.198]:

$$\begin{aligned} \dot{\hat{\boldsymbol{\theta}}} &= -\boldsymbol{\Gamma}(t)\bar{\boldsymbol{e}}_2\boldsymbol{\phi} \\ \dot{\boldsymbol{\Gamma}} &= \begin{cases} \beta\boldsymbol{\Gamma} - \boldsymbol{\Gamma}\frac{\boldsymbol{\phi}\boldsymbol{\phi}^T}{m^2}\boldsymbol{\Gamma} & \text{if } ||\boldsymbol{\Gamma}(t)|| \leq \boldsymbol{R}_0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.4.1}$$

where $\boldsymbol{\Gamma}(0) = \boldsymbol{\Gamma}_0 = \boldsymbol{\Gamma}_0^T > 0$, $||\boldsymbol{\Gamma}_0|| \leq \boldsymbol{R}_0$ and $\boldsymbol{R}_0$ is a constant that serves as an upper bound for $||\boldsymbol{\Gamma}||$. $m^2 = 1 + n_s^2$ and n_s is chosen such that $\frac{\boldsymbol{\phi}}{m} \in \mathcal{L}_\infty$. With $\beta > 0$, $\boldsymbol{\Gamma}(t)$ will not become arbitrarily small and because of the upper bound $\boldsymbol{R}_0$ it will not become infinitely large. This modification guarantees that $\boldsymbol{\Gamma} \in \mathcal{L}_\infty$. The proof for the properties of the update law can be found in Appendix D

Chapter 6

Results

The derived controller is compared to two other controllers in a dynamic positioning setting to evaluate its performance. The other controllers are those from [Patompak and Nilkhamhang, 2012] and [Antonelli et al., 2003]. It was found in [Ohrem, 2014] that the controller from [Antonelli et al., 2003] is preferred to for instance PID and LQ controllers so a comparison with these controllers is not performed. The controller from [Patompak and Nilkhamhang, 2012] is an Adaptive Backstepping controller with Bound estimation and is referred to as "ABSwB" in the figures. The controller from [Antonelli et al., 2003] is a PD controller with current and gravity adaptation. It is referred to as "Antonelli" in the figures.

When testing the reverse path control system the ROV is first manoeuvred in an S-shaped path and then commanded to return to the origin. This is done both forwards and backwards. The path following of the predefined path considers a straight path and a circular path under the influence of current.

The comparison is done in the Simulink model of Merlin WR200 and for all simulations, the current angle is 45° with a velocity of 0.5m/s.

The results from the simulator at IKM Subsea does not include a comparison, since this is very hard to achieve when the conditions are unequal in each simulation. Only the derived controller is tested.

6.1 Simulink Dynamic Positioning results

The DP controllers are compared for the following cases:

- Stand still for 50 seconds.
- Move 2 meters North.

Parameter	Value
$\boldsymbol{K}_1$	diag{1.0542, 1.2307, 0.6557, 2.6345, 2.0862, 2.1444}
$\boldsymbol{K}_2$	diag{6323.4, 5416.3, 6100, 1087.3, 1373.1, 1335.9}
$\boldsymbol{\Gamma}(0)$	diag{1, 1, 1, 1}

Table 6.1: Controller parameters used in the simulations

- Rotate 90°.

The error in North, East and ψ and the control input X is presented for comparison basis. The position in Down is also presented. The controller parameters in table 6.1 are used in the simulations. To measure the performance, the mean of the 1-norm and the RMS, or 2-norm, of the error during the step change in North and ψ are used:

$$\begin{aligned} \text{Mean of 1-norm: } \bar{e} &= \frac{|e_1| + |e_2| + \cdots + |e_n|}{n} \\ \text{RMS or 2-norm: } e_{rms} &= \sqrt{\frac{1}{n}(e_1^2 + e_2^2 + \cdots + e_n^2)} \end{aligned} \tag{6.1.1}$$

The error values are presented in tables 6.2 - 6.3 and the plots can be seen in figures 6.1 - 6.14.

6.1.1 Error in North

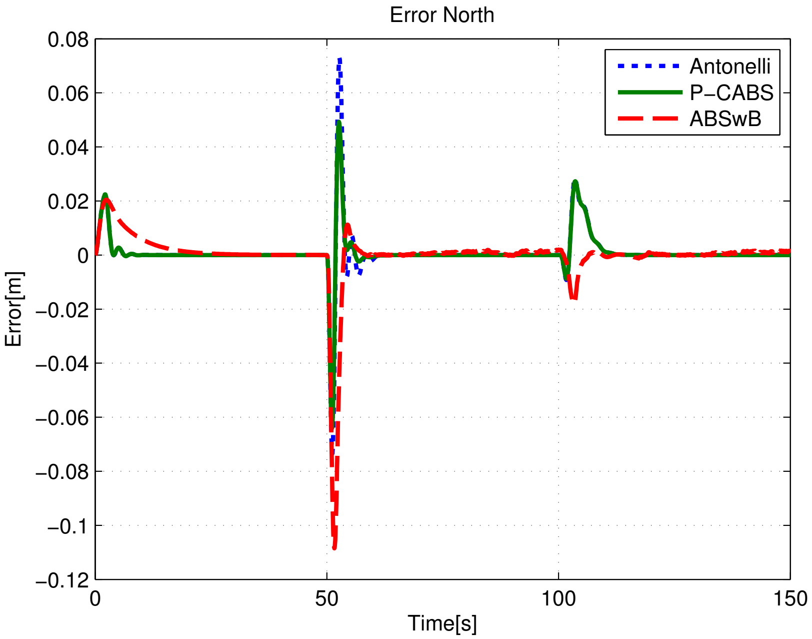


Figure 6.1: Error in North. For the first 50 seconds, the ROV should hold its position in 0, then move 2 meters and finally rotate 90 degrees after 100 seconds.

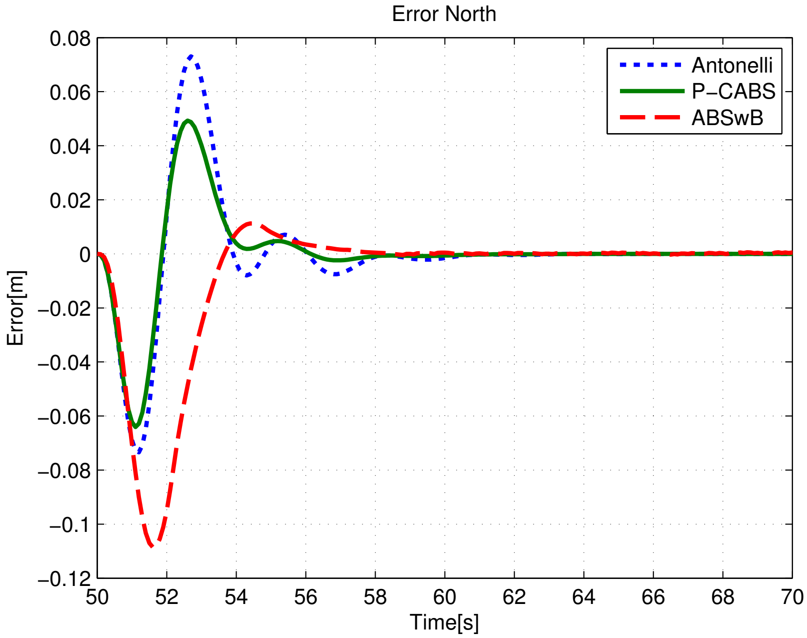


Figure 6.2: Close up on the error in North when the ROV is moving 2 meters.

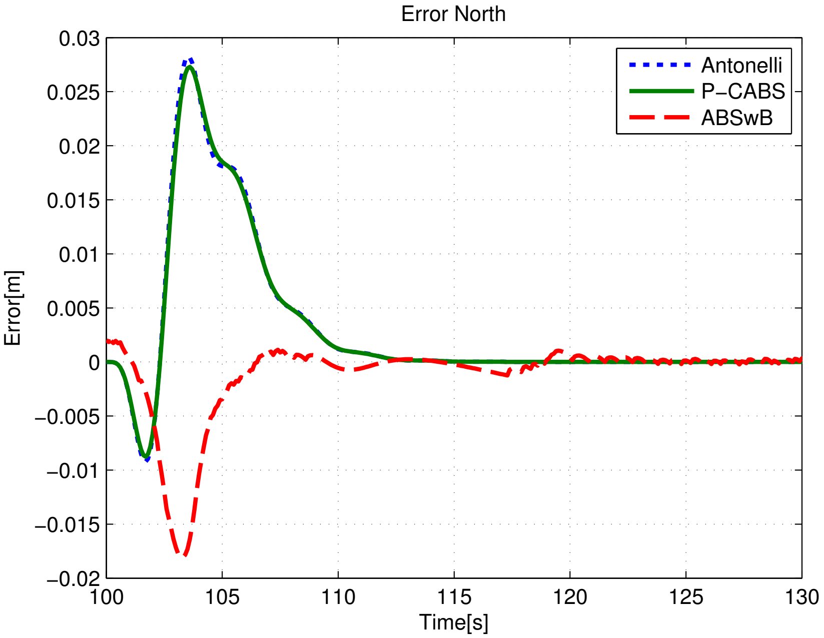


Figure 6.3: Close up on the error in North when the ROV turns 90°

Discussion of results

There are slight variations in the performance of the controllers. The ABSwB is slower to converge in the beginning when the ROV should keep its position. This is because it does not possess the same current adaptation part as the other controllers. The P-CABS controller and the controller from [Antonelli et al., 2003] show very similar behaviour in the first 50 seconds. It is when the setpoint change in North occurs that the difference between the controllers becomes clear. From table 6.2 it can be seen that the mean of the 1-norm and RMS error of the P-CABS controller is smaller than the other two controllers. This is also seen in figure 6.2, where the P-CABS controller shows a lower peak value and faster convergence. During the step in ψ however, the error is very similar for the P-CABS controller and the controller from [Antonelli et al., 2003], but ABSwB controller gives approximately half the error. Looking at the plots in figure 6.3 the error of the ABSwB controller is not a smooth curve like the controller of Antonelli and the P-CABS controller. This is because the control input generated by the ABSwB is very choppy.

	Step		Rotation	
Controller	Mean of 1-norm[m]	RMS[m]	Mean of 1-norm[m]	RMS[m]
P-CABS	0.009	0.2320	0.0052	0.1321
Antonelli	0.0122	0.3018	0.0053	0.1344
ABSwB	0.0138	0.3854	0.0023	0.0684

Table 6.2: Mean of the 1-norm and RMS of the error in North for the 3 controllers during the 2 meter step in North and 90° rotation.

6.1.2 Error in East

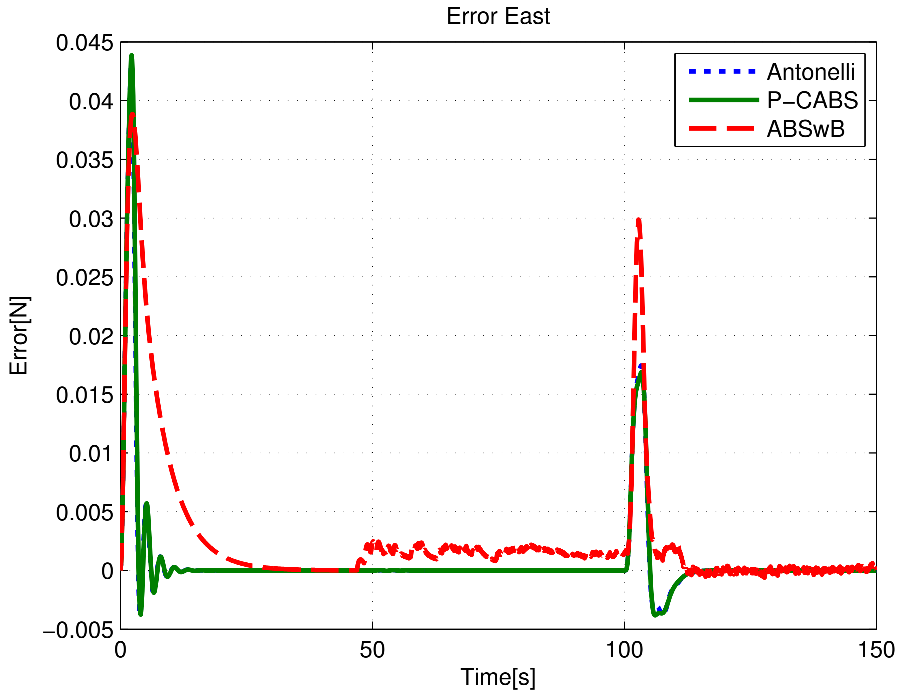


Figure 6.4: Error in East when the ROV moves 2 meter in North and turns 90°

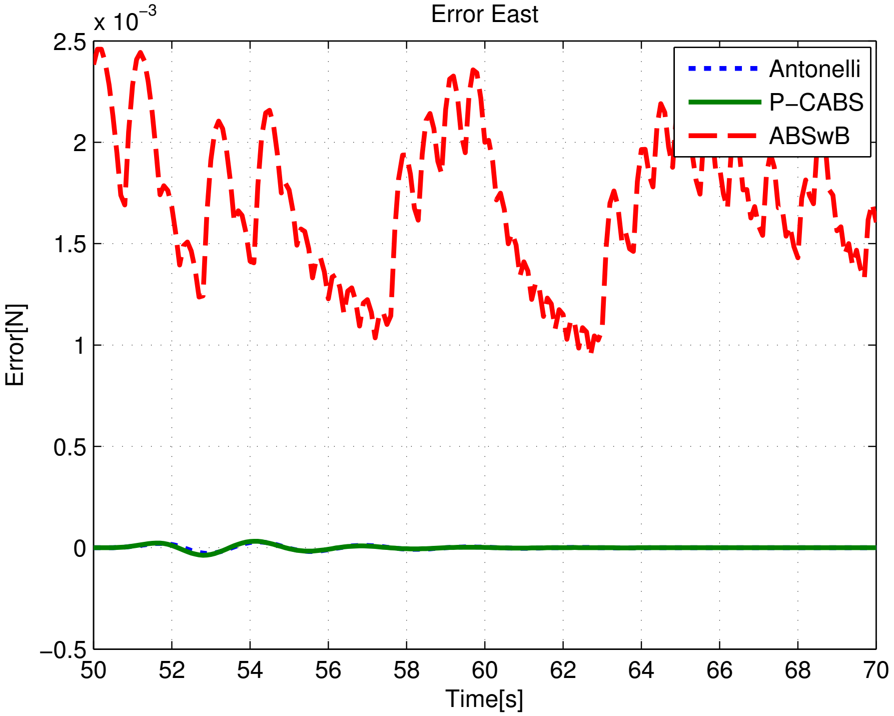


Figure 6.5: Close up of the error in East when the ROV moves 2 meter in North

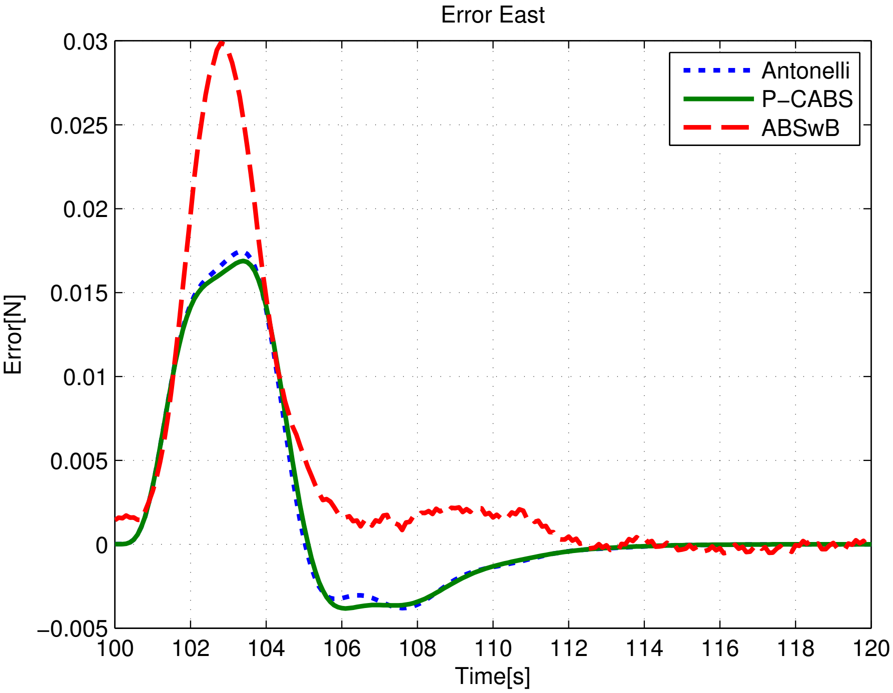


Figure 6.6: Close up of the error in East when the ROV rotates 90°

Discussion of results

There is no setpoint change in the East direction, so the error here is strictly due to current and couplings between the DOFs of Merlin WR200. Looking at the mass, damping and Coriolis matrices used in the process plant model, it can be seen that there is a coupling between North and ψ and between ψ and East. This leads to an indirect coupling between North and East, so one can expect some change in East with movement in North. This coupling is very weak, so not much influence occurs.

Like in North, the ABSwB controller is slower to converge to 0 in the first 50 seconds as seen in figure 6.4. The P-CABS and Antonelli controller show very similar behaviour again. When the step in North happens, one can observe a very small change in East with the Antonelli and P-CABS controller, but the ABSwB controller show a deviation. This deviation is thought to be caused by the discrete robustifying term. In practice, the deviation is so small it can be ignored. This deviation leads to the ABSwB controller showing a much larger mean of 1-norm and RMS error in this DOF for the step in North. The ABSwB is able to remove the error in East after the step in ψ, so the mean and RMS error for that step is smaller, but still larger than the P-CABS and Antonelli controller who show very similar behaviour.

	Step		Rotation	
Controller	Mean of 1-norm[m]	RMS[m]	Mean of 1-norm[m]	RMS[m]
P-CABS	7.31e-06	1.48e-04	0.0033	0.0855
Antonelli	7.11e-06	1.32e-04	0.0033	0.0862
ASBwB	0.0017	0.0213	0.0042	0.1236

Table 6.3: Mean of the 1-norm and RMS of the error in East for the 3 controllers during the 2 meter step in North and 90° rotation.

6.1.3 Error in ψ

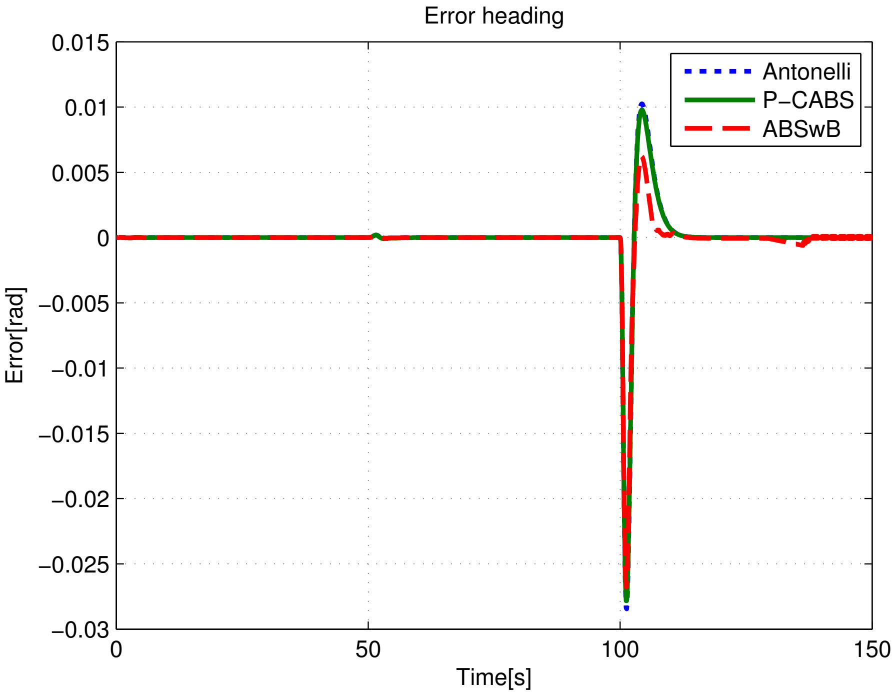


Figure 6.7: Error in ψ. For the first 50 seconds, the ROV should hold its position in 0, then move 2 meters and finally rotate 90 degrees after 100 seconds.

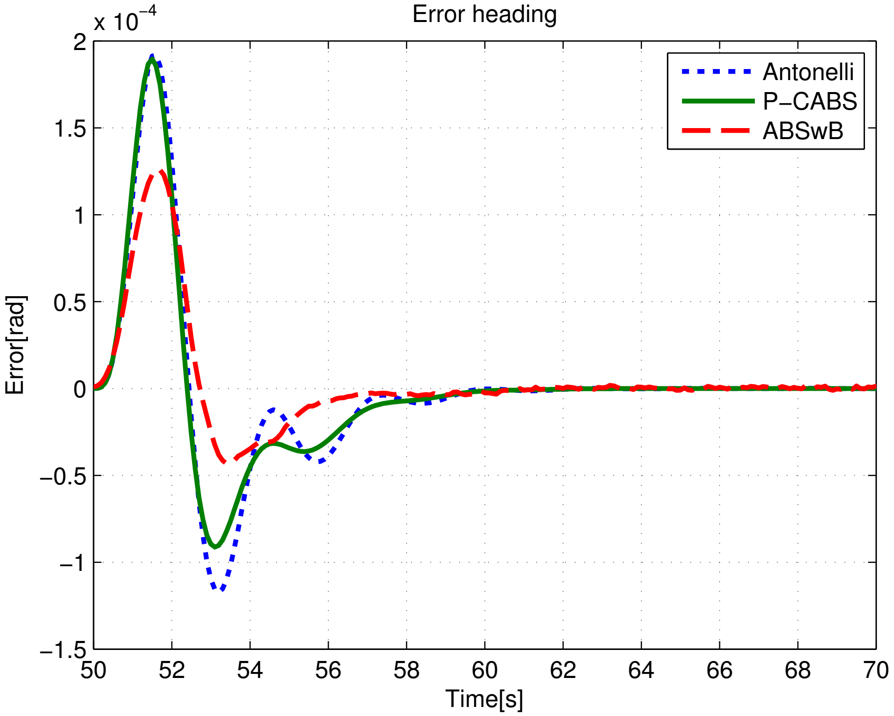


Figure 6.8: Close up on the error in ψ during the step in North.

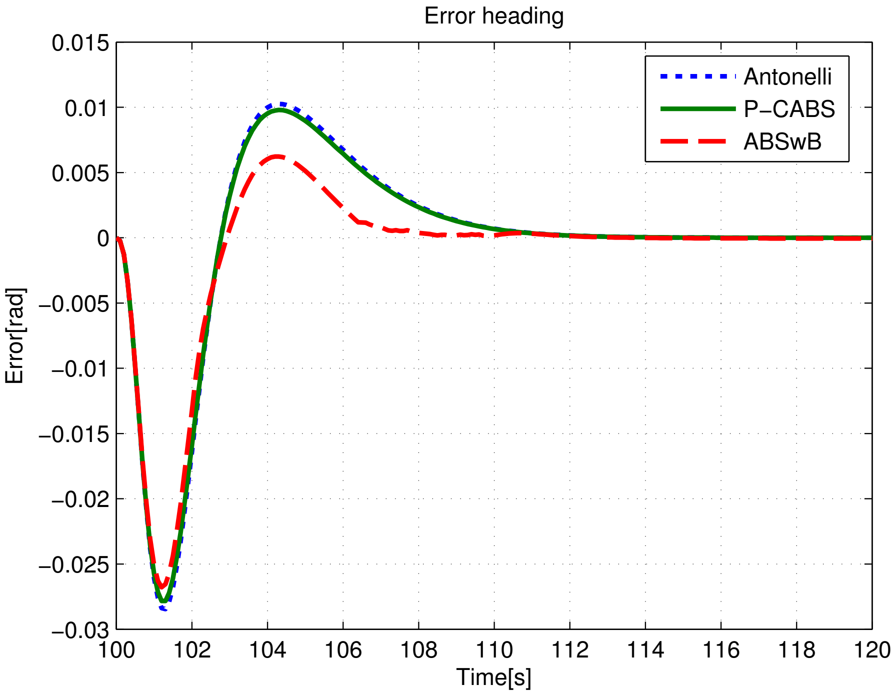


Figure 6.9: Close up on the error in ψ when the ROV is told to rotate 90°.

Discussion of results

The influence of the ocean current on ψ through couplings is minimal, so for the first 50 seconds the ROV is able to keep the desired heading. When the step in North happens, there is hardly any influence in ψ, but in figure 6.8 it can be seen that the ABSwB controller has lower peak value. This leads to the ABSwB controller having a smaller mean and RMS error. It is also seen from figure 6.9 that the ABSwB has a lower peak and fast convergenve when the setpoint in ψ changes. It seems like the ABSwB gives less error in the DOF that is not influenced by current disturbances.

	Step		Rotation	
Controller	Mean of 1-norm[rad]	RMS[rad]	Mean of 1-norm[rad]	RMS[rad]
P-CABS	2.87e-05	6.48e-04	0.0039	0.1053
Antonelli	3.00e-05	6.9e-04	0.0040	0.1081
ASBwB	1.84e-05	4.42e-04	0.0027	0.0909

Table 6.4: Mean of 1-norm and RMS of the error in ψ for the 3 controllers during the 2 meter step in North and 90° rotation.

6.1.4 Down position

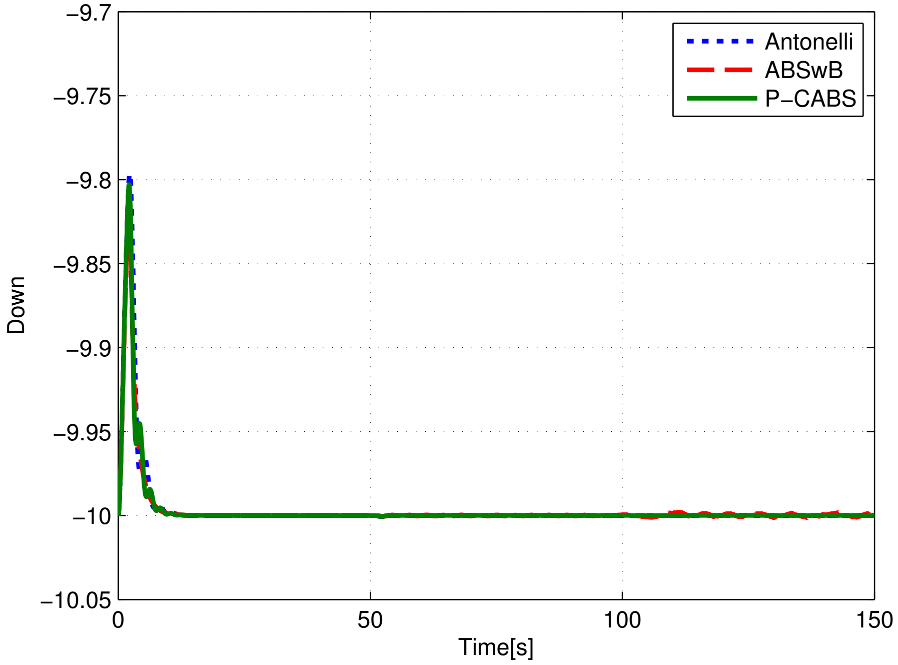


Figure 6.10: The Down position during station keeping and setpoint changes.

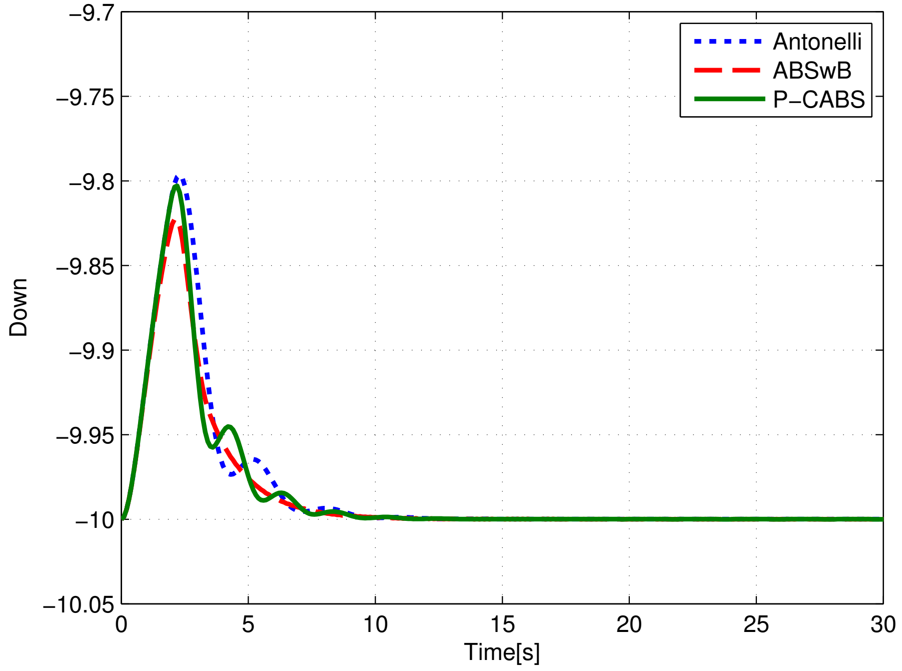


Figure 6.11: Close up of the Down position when station keeping is initiated.

Discussion of results

All the controllers are able to keep the Down position at the desired -10m. In the beginning there are some differences in the transient behaviour as can be seen in figure 6.11. The P-CABS controller generates more oscillations than the other controllers. From $t \approx 100$ some oscillations occur for the ABSwB controller caused by the fact that $\boldsymbol{e}_1$ does not reach 0 before the compensation stops.

6.1.5 Control force

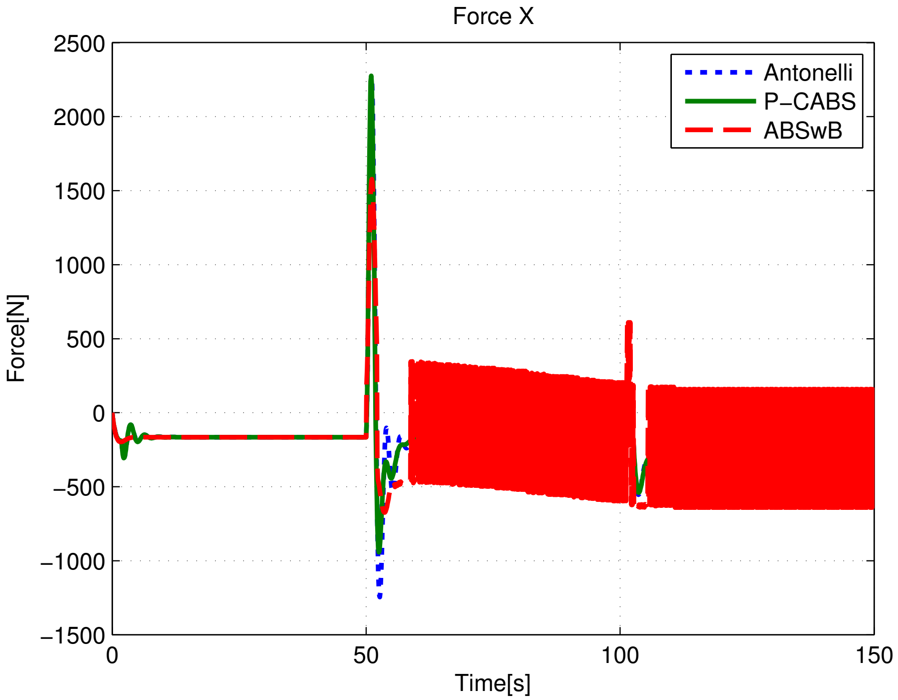


Figure 6.12: The calculated force in X for the 3 controllers during the 2 meter step in North and 90° rotation.

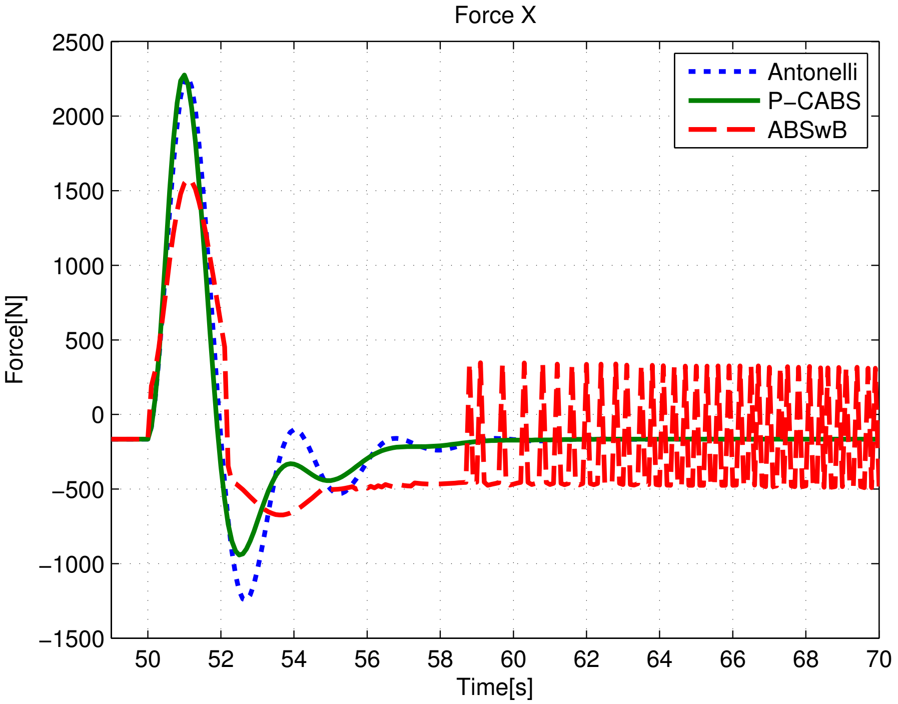


Figure 6.13: Close up of the calculated force in X for the 3 controllers during the 2 meter step in North.

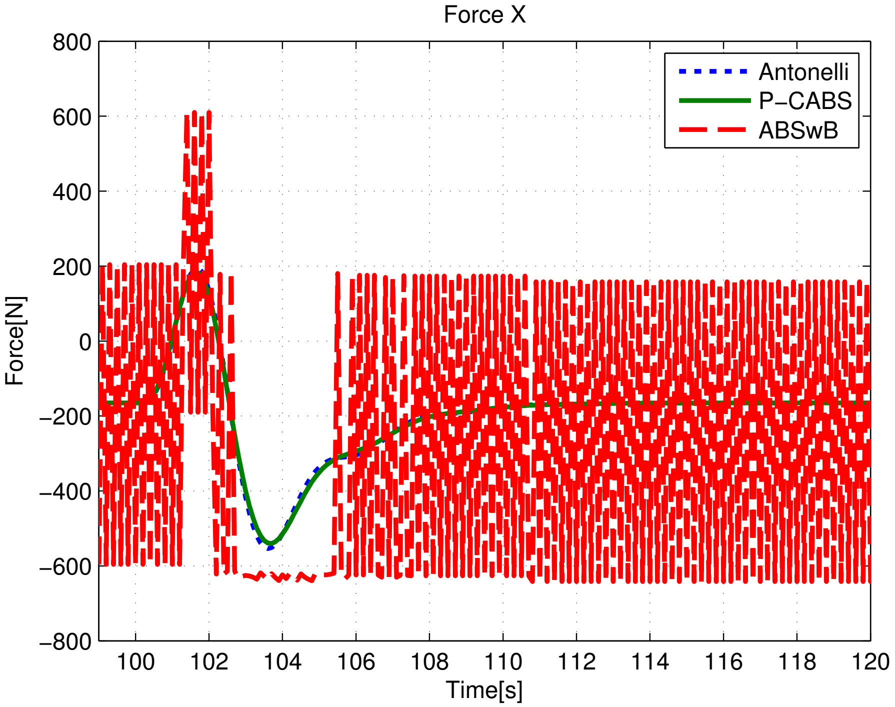


Figure 6.14: Close up of the calculated force in X for the 3 controllers during the rotation

Discussion of results

From the figure 6.12-6.14 it is clear that the force applied by the ABSwB is very different from the force applied by the P-CABS and Antonelli controllers. This is caused by the discontinuous update of the bound estimate. Because of this discontinuity the controller is able to force the system onto a sliding surface, but the cost is a very choppy input that may cause wear and tear on the actuators. The input generated by the P-CABS and Antonelli controllers are smooth. The P-CABS controller has slightly lower oscillations than the Antonelli controller during the step in North.

6.2 Simulink return path mode results

The ROV is first manoeuvred with the joystick in an S-shaped pattern and then given the command to return to the origin by using the stored waypoints as references for the LOS steering law. The parameters used are listed in table 6.5.

Parameter	Description	Value
R_{add}	Circle of acceptance for adding waypoint	3
R	Circle of acceptance for switching waypoint	0.5
Δ	Lookahead distance	20
Δ_s	Speed tuning parameter	15
Δ_e	Speed tuning parameter	15

Table 6.5: Parameters used in the return path control system

The desired velocity in the body x-direction is chosen as constant $u_d = 0.5$[m/s] for as long as the ROV is returning along the path. When the second to last waypoint is reached, the desired velocity changes to $u_d = -u_{max} \frac{s}{\sqrt{s^2+\Delta_s^2}}$ so the ROV ramps down its velocity while approaching the final waypoint and then stops there. The desired velocity in body y-direction is chosen as $v_d = -v_{max} \frac{e}{\sqrt{e^2+\Delta_e^2}}$.

The driven path that generates the waypoints is shown in figure 6.15 for when the ROV is steered forward in an S-shaped pattern and in figure 6.19 for when the ROV is reversed in an S-shaped pattern. The return path of the ROV compared to the waypoints is presented in figure 6.16 and in figure 6.20 for when the ROV is reversed. A traceplot of the ROV driving and returning along the path is shown in figures 6.18 and 6.22 for forward and reverse respectively. The crosstrack error is shown in figure 6.17 and in figure 6.21

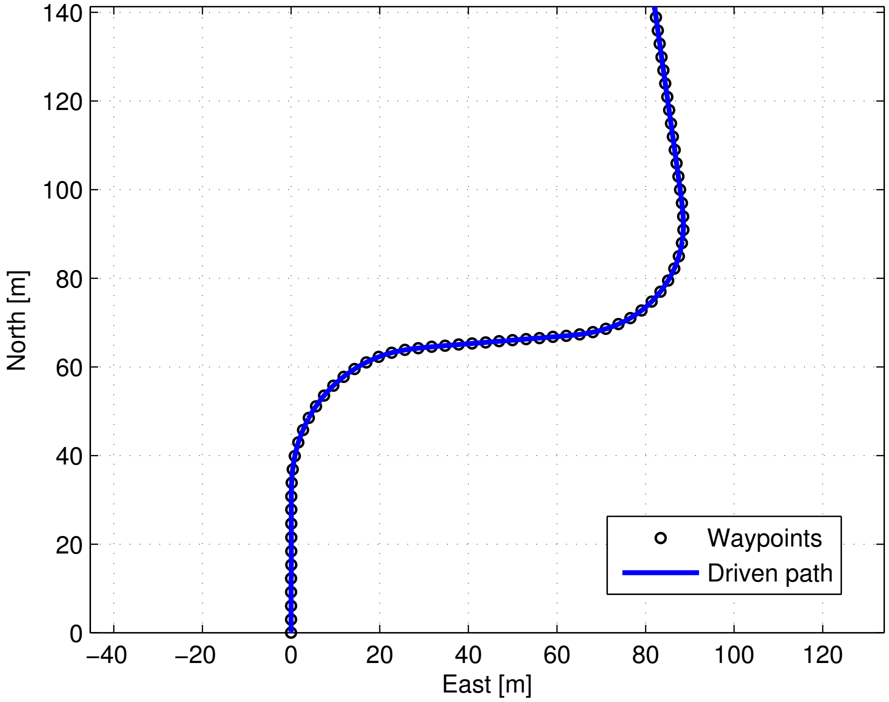


Figure 6.15: The driven path of the ROV when it is steered in an S shaped pattern. The waypoints are added every 3 meters.

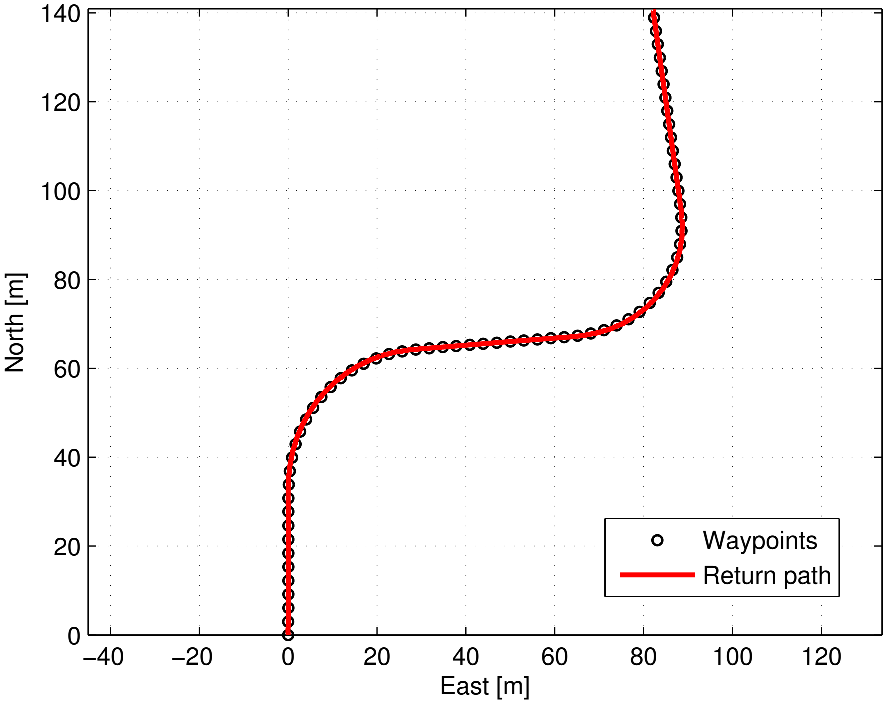


Figure 6.16: The ROV returns with constant velocity and uses the waypoints as guidance for the desired heading.

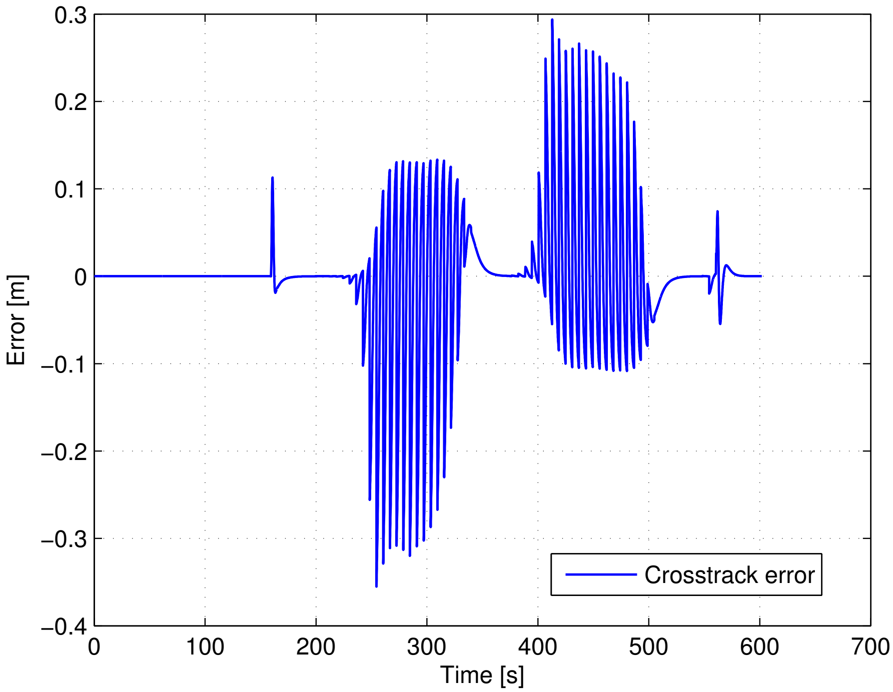


Figure 6.17: The crosstrack error during return. The return starts at about $t = 156$.

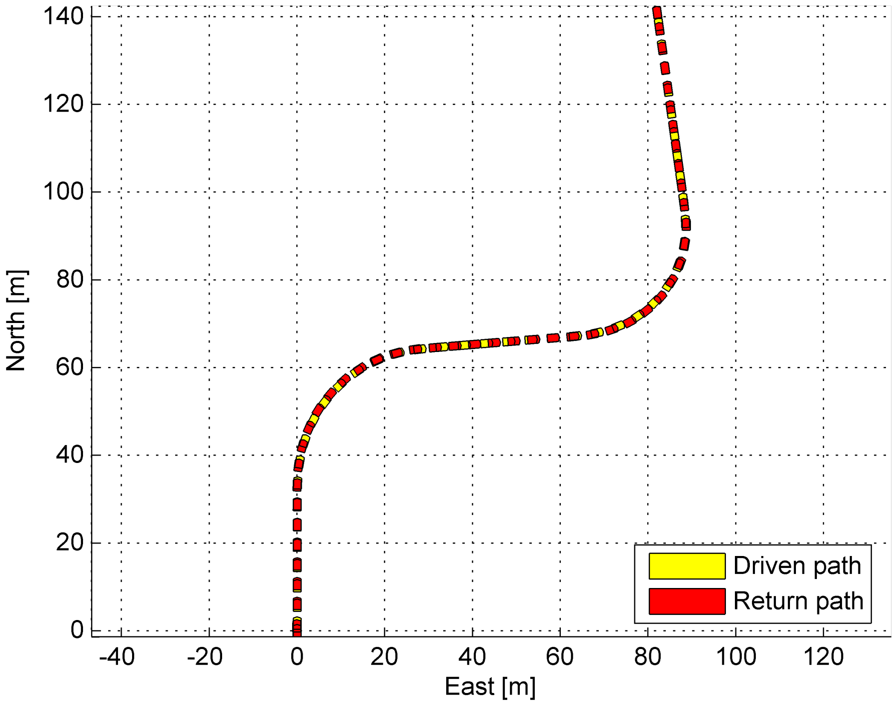


Figure 6.18: Traceplot of the ROV driving the path and then returning along it.

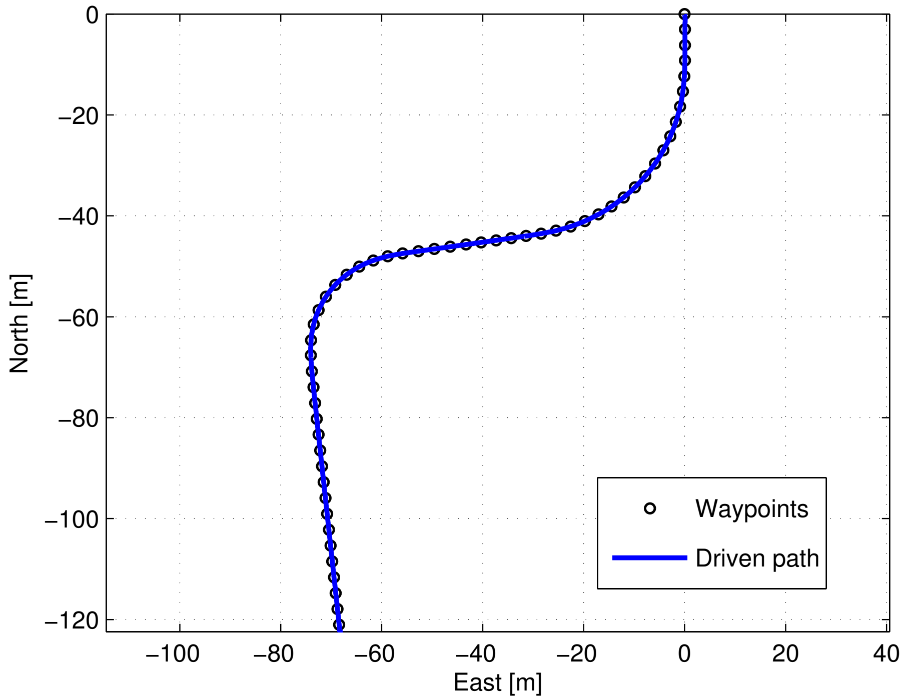


Figure 6.19: The driven path of the ROV when it is reversing in an S shaped pattern. Waypoints are added every 3 meters.

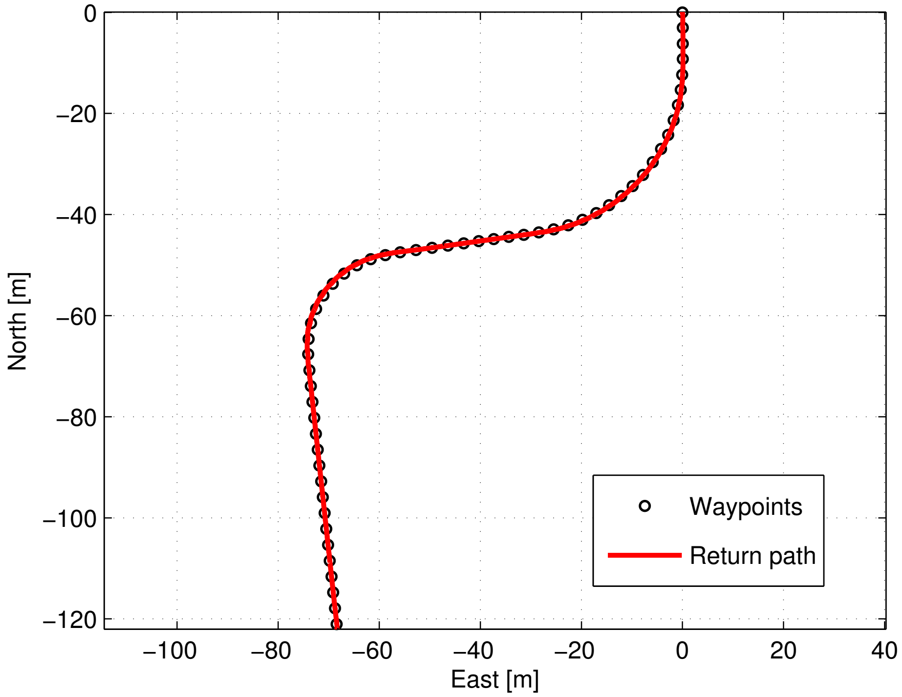


Figure 6.20: The ROV is returning along the driven path using the waypoints as guidance.

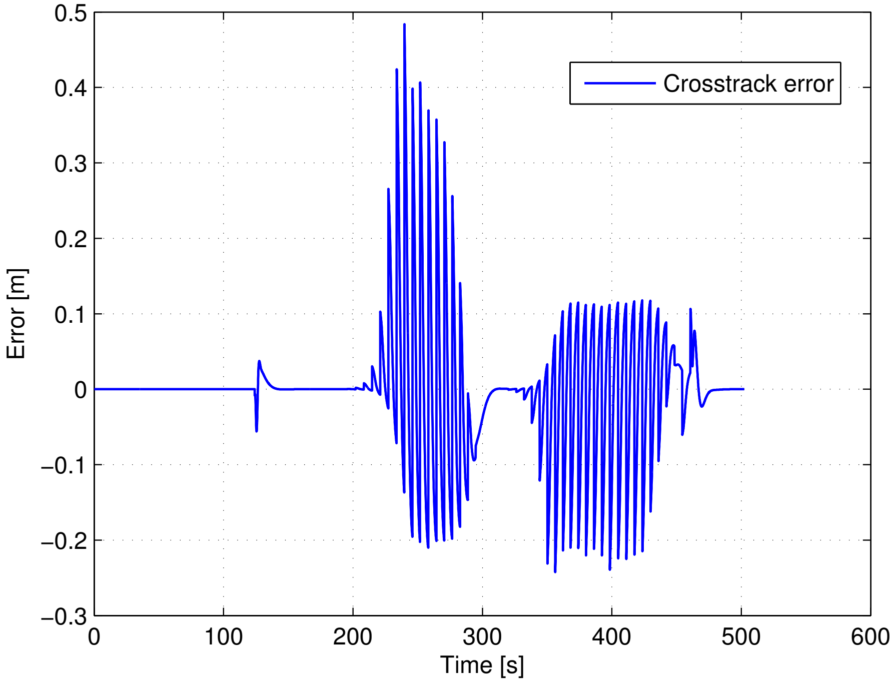


Figure 6.21: The crosstrack error when the ROV is returning along the driven path. The return starts at $t = 125$ approximately.

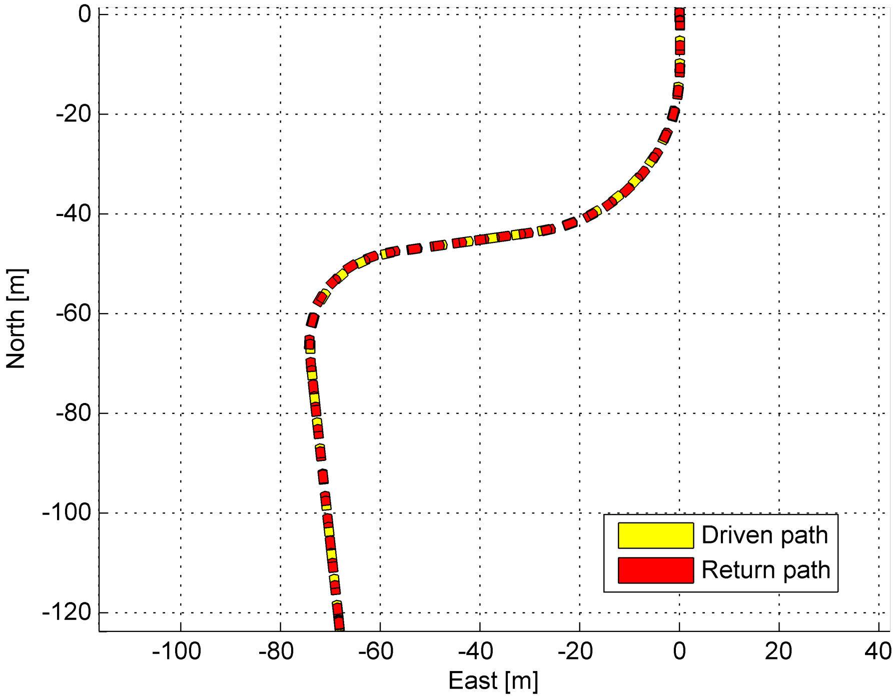


Figure 6.22: Traceplot of the ROV driving the path and then returning along it. The ROV keeps the same heading when it is returning as it did when it was driving.

Discussion of results

The traceplot in figures 6.18 and 6.22 are good indicators for the performance of the return path system. There is hardly any difference between the driven path and the return path. The crosstrack error needs some explanation as it seems like the ROV is off track a lot. Figure 6.17 shows that at the start of the return path, when the waypoints are more or less in a straight line, the error converges to 0. During turning however, the error is jumping between negative and positive values. This is because the ROV does not reach the current desired waypoint before the next waypoint is chosen as desired waypoint. Thus the crosstrack error jumps because of the LOS vector that is created when the waypoint changes. Figures 6.23 and 6.24 tries to illustrate this.

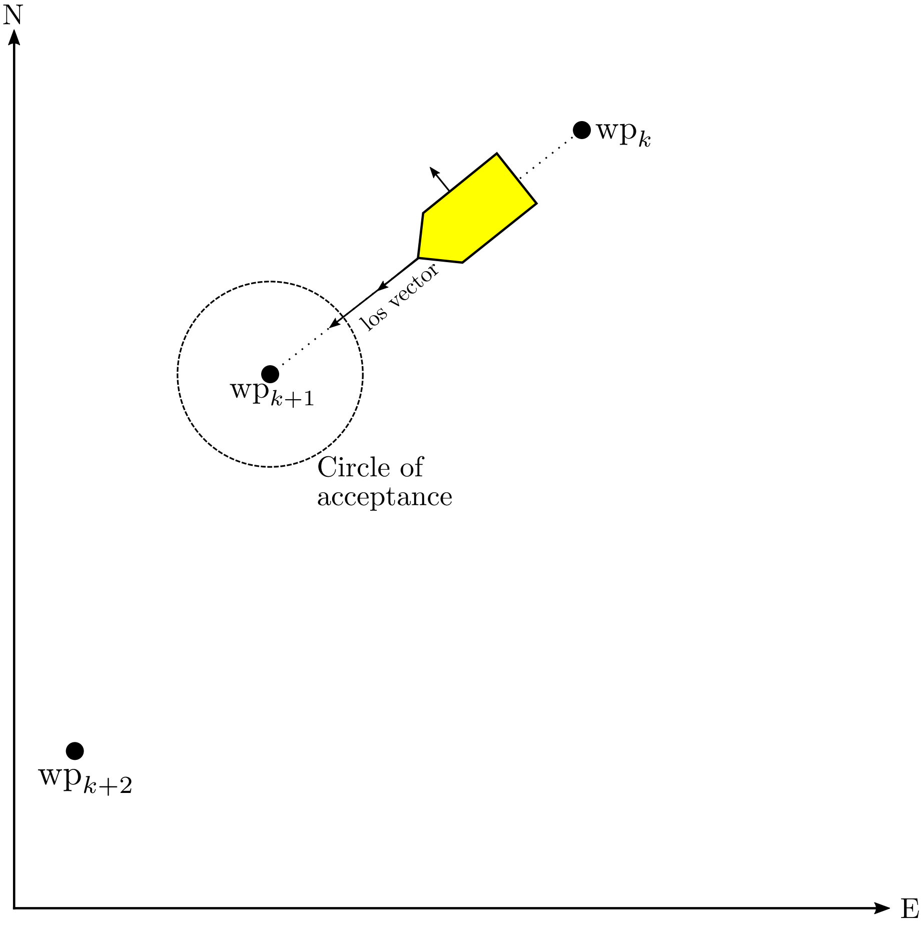


Figure 6.23: The ROV is approaching a waypoint. The crosstrack error is 0. The waypoint wp_{k+2} is not aligned with waypoint wp_k and wp_{k+1}, so a crosstrack error will occur because the waypoint order switches as the ROV reaches the circle of acceptance.

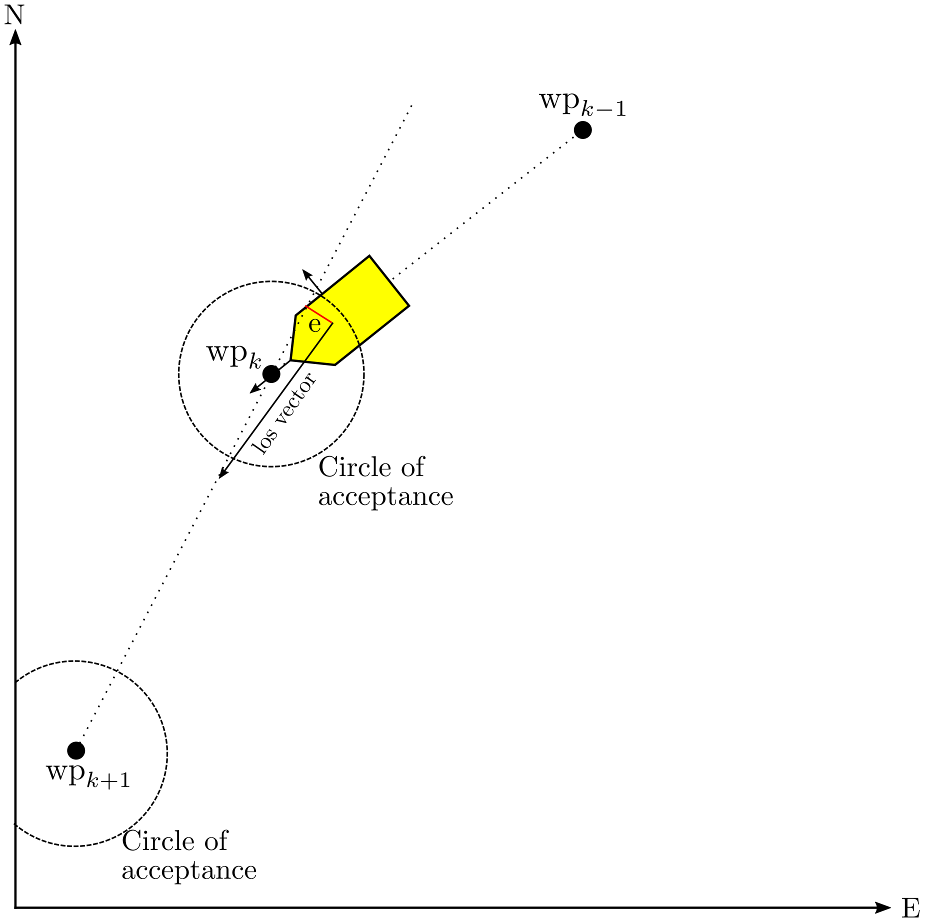


Figure 6.24: As the ROV reaches the circle of acceptance, the waypoint switches and a new LOS vector is calculated. Since the ROV is not at the waypoint when the switch occurs, there will be a crosstrack error.

6.3 Simulink predefined paths results

6.3.1 Straight line

A straight line path is defined with the following parametrization:

$$\begin{aligned} x_{SF}(s) &= s\,\cos(\psi_{SF}) \\ y_{SF}(s) &= s\,\sin(\psi_{SF}) \\ \psi_{SF} &= \frac{\pi}{8} \end{aligned} \tag{6.3.1}$$

The initial position and attitude of the ROV is chosen as

$$\boldsymbol{\eta}_0 = \begin{bmatrix} -50 & -50 & -10 & 0 & 0 & 0 \end{bmatrix}^T$$

The desired velocity $u_d = 1$m/s and the tuning parameter $\Delta = 50$. Since Merlin WR200 is fully actuated the desired velocity in sway can be set to $v_d = 0$ such that the current is properly compensated. This means that the ROV does not have to sideslip to stay at the desired path.

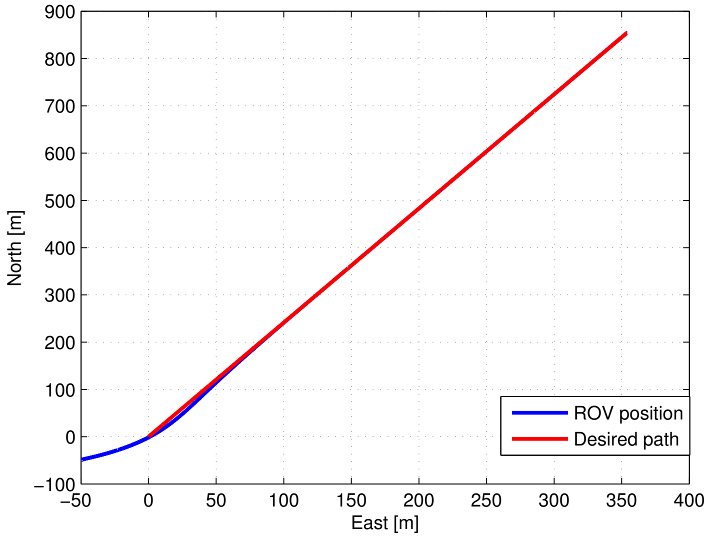


Figure 6.25: The ROV converges to the path. Due to the choice of tuning parameter Δ the transition is smooth, though with some overshoot.

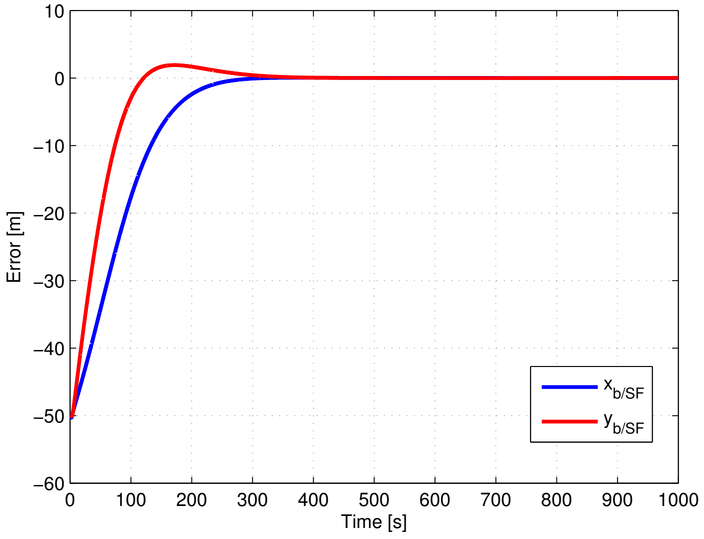


Figure 6.26: The error converges with some overshoot due to the tuning parameter Δ.

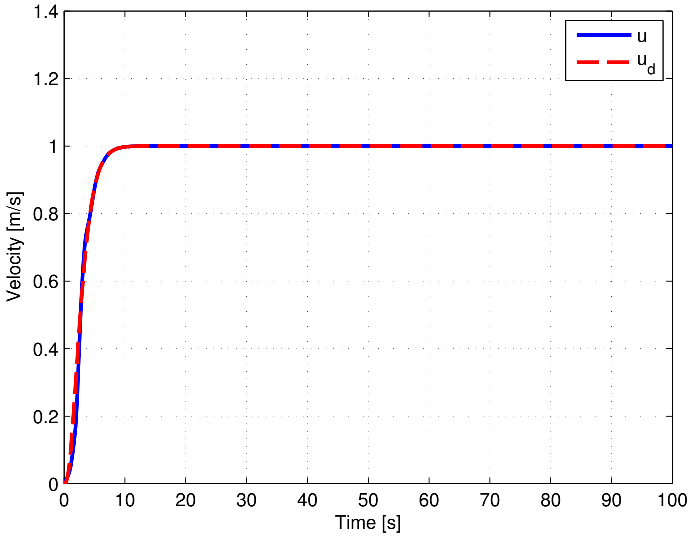


Figure 6.27: The desired velocity is reached in about 10 seconds and stays the same throughout the simulation. The plot is only for the first 100 seconds to show the transient behaviour.

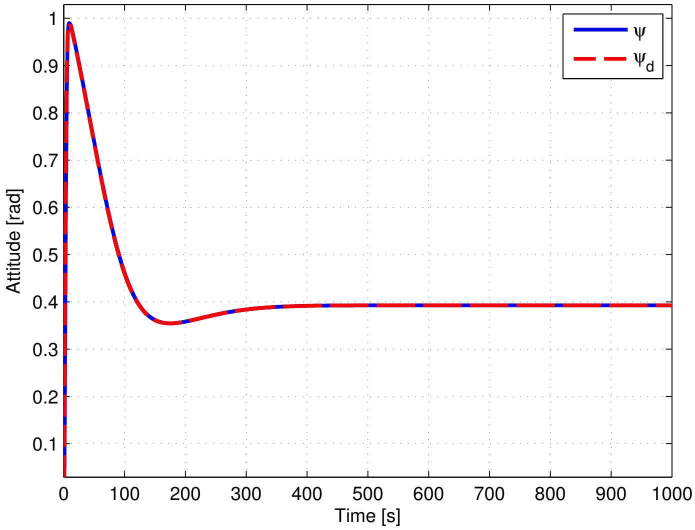


Figure 6.28: The heading angle follows the desired heading angle very nicely. The heading angle converges to a constant value as the ROV approaches the path.

6.3.2 Circular path

A circular path with the following parametrization is defined

$$\begin{aligned} x_{SF}(s) &= R\cos\left(\frac{s}{R}\right) \\ y_{SF}(s) &= R\sin\left(\frac{s}{R}\right) \\ \psi_{SF} &= \frac{\pi}{2} + \frac{s}{R} \end{aligned} \tag{6.3.2}$$

The ROV starts in the origin and $R = 300$. The desired velocity is $u_d = 1$ m/s and the tuning parameter Δ is set to the more aggressive value of 10.

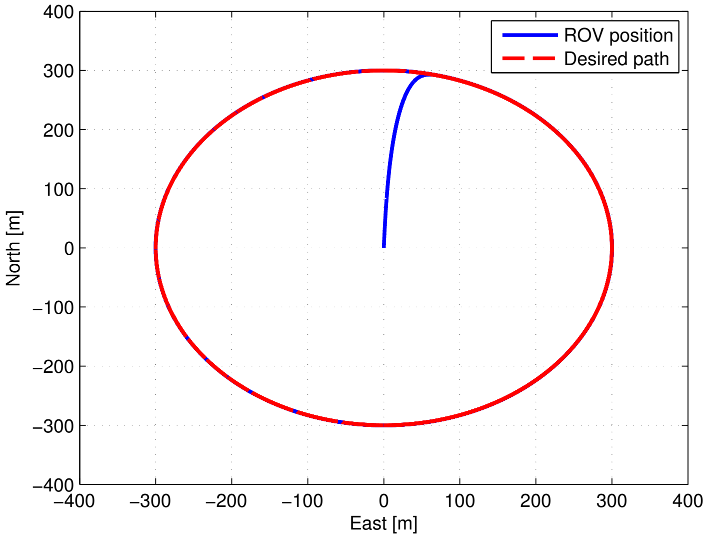


Figure 6.29: The ROV manages to follow the circular path. There is a slight deviation because the controller used is not a perfect feedback linearizing controller.

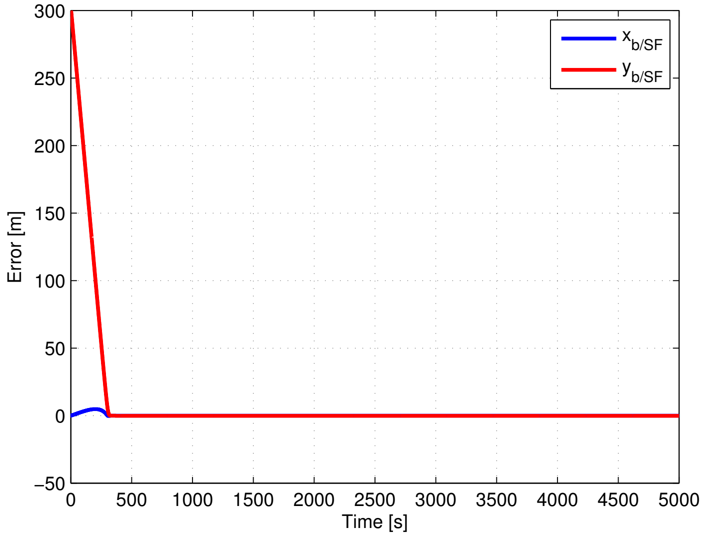


Figure 6.30: The error seems to converge to 0, but there is a very small deviation of about 0.1 m

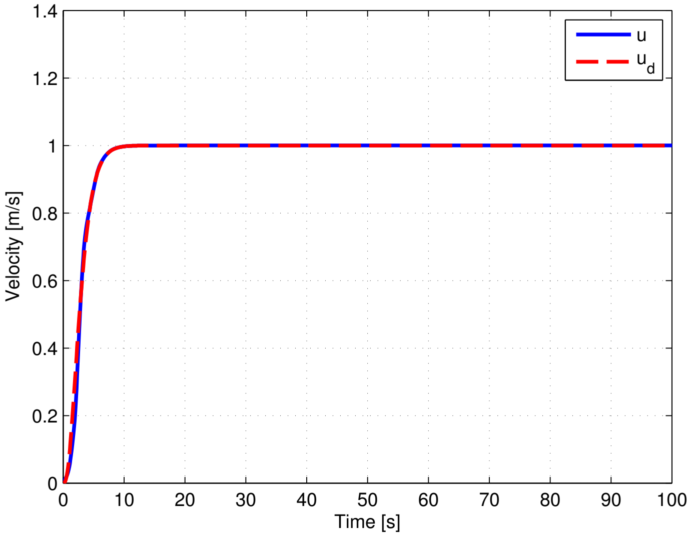


Figure 6.31: The desired velocity is reached in about 10 seconds and stays the same throughout the simulation. The plot is only for the first 100 seconds to show the transient behaviour.

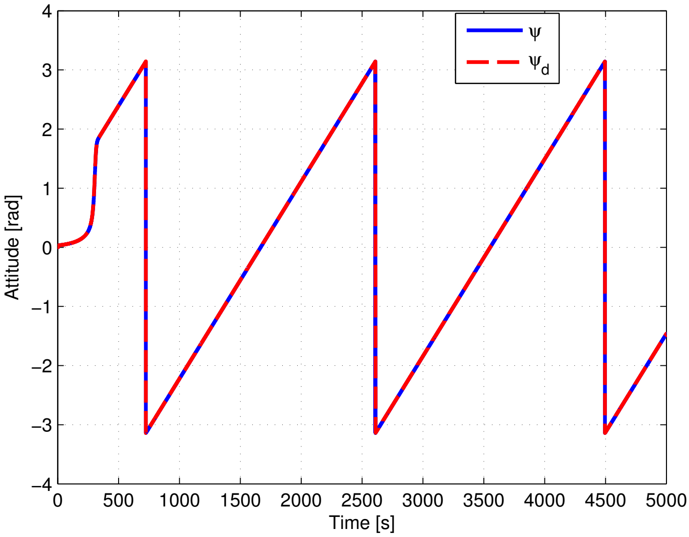


Figure 6.32: The heading angle follows the desired heading angle very nicely. The angle is limited to be between $-\pi$ and π.

Discussion of results

When the path is predefined by using the *Serret-Frenet* reference frame, the P-CABS controller is able to drive the ROV to the desired heading and velocity as seen in figures 6.28, 6.32, 6.27 and 6.31. A surge velocity controller is able to remove the necessity for sideslipping. Since the controller is not a perfect feedback linearizing controller, a very small error is present for the circular path case. The convergence is smooth, though with some overshoot for the straight path case because of the choice of the tuning parameter Δ. The velocity and heading converges very nicely to the desired values.

6.4 Results from simulator at IKM Subsea

For training and mission simulation purposes, IKM Subsea has invested in an advanced simulator from CM Labs. This is how it is described on the IKM Subsea homepage (freely translated)[IKM]:

"The simulator gives realistic and lifelike impressions of real situations. By using the same control system as in the real ROVs and lifelike dynamics and sensor simulations our pilots are trained in a very realistic environment [...] The simulator is running on Vortex Dynamics, the leading dynamics simulations engine in the industry. The Vortex engine gives the best rendition in simulation of ROV hydrodynamics, propulsion systems, tether, manipulators and sensors."

The developed DP control system was implemented in the simulator during a testing period between April 8th and April 12th of 2015. Unfortunately, there was not enough time to implement the reverse path control system since much of the time was spent on troubleshooting. The DP system had to be reprogrammed to run in Matlab instead of Simulink due to lack of Simulink software packages needed for real time control. Since the author had no knowledge of the dynamic model in the simulator, the reference frames used and the thruster definitions, a lot of time was spent on troubleshooting these areas and fixing the code to fit. One example is that the thruster numbering done in [Knausgård, 2013], which this thesis is based on, did not match the thruster numbering in the simulator. Also, the thruster positive directions was not consistent with what the author had expected. This thesis and the thesis of [Knausgård, 2013] considers a $\{n\}$ frame as mentioned in chapter 2.1. The reference frame used in the simulator was a North-West-Up frame. This was the source of a lot of confusion, as it was not expected and hence overlooked for a while. But gladly it was fixed on the final day of testing. This meant that there was only time to test the P-CABS control law in the simulator for station keeping and set point changes with current. A comparison in the simulator would be difficult to implement anyhow, because the inconsistency of such a life like system would not provide the same conditions in all tests.

The controller gains used in the simulations, $\boldsymbol{K}_1$, $\boldsymbol{K}_2$ and $\boldsymbol{\Gamma}$ had to be retuned. It was found that the dynamic model used in the simulations in Simulink did not match the dynamic model in the simulator. This is thought to be because of the thruster dynamics model and because of the disturbance forces from the tether. In the Simulink model the thrusters are limited by a *rate limiter* block. This block constraints the change in control forces to the informed value of $|\Delta\boldsymbol{\tau}| \leq 8000$[N/s]. The controller was tuned to avoid breaching this constraint, but when the same gains were applied to the control system in the simulator, the behaviour did not match. The adaptation in yaw had to be turned off to achieve proper control of this DOF, but the results are still very good in this DOF.

The code used in the simulations is attached in Appendix E

6.4.1 Available measurements & coordinate system

The simulator provides measurements of the ROV positions and attitude. The available measurements are presented in table 6.6. The coordinate system used in the simulator is not the same as the coordinate system used when deriving the model for Merlin WR200 in chapter 3. The simulator coordinate system is a North-West-Up system, rather than the North-East-Down system described in chapter 2.1. This was the source of much confusion since the control forces were calculated based on a $\{n\}$ frame. The forces where applied wrong due to the structure of the rotation matrix since ψ was defined in two different directions. The solution was to change the rotation matrix from:

Measurement	Description
N	North position
E	East position, positive right
D	Down position, positive up
ϕ	Roll angle, positive about N following the right hand rule
θ	Pitch angle, positive about E following the right hand rule
ψ	Yaw angle, positive about D following the right hand rule

Table 6.6: Available measurements from the simulator

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

to

$$\boldsymbol{R}^*(\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

utilizing $\cos(-x) = \cos(x)$ and $-\sin(x) = \sin(-x)$.

Parameter	Value
$\boldsymbol{K}_1$	$\mathrm{diag}\{0.1, 0.4, 0.2, 20, 1000, 0.4\}$
$\boldsymbol{K}_2$	$\mathrm{diag}\{3930.7, 3151.2, 5416.9, 1, 1, 2469.4\}$
$\boldsymbol{\Gamma}$	$\mathrm{diag}\{400, 400, 400, 0\}$

Table 6.7: Controller parameters used in the simulator

6.4.2 Results from DP test

After tuning of the controller parameters the DP system worked. Time delays was discovered in the transfer functions from input to velocity in surge, sway and yaw. These transfer functions are presented in section 6.4.3. The adaptation gain had to be tuned very conservative because of this, to not cause large overshoots. This could have been improved by using for instance a Smith-predictor [Smith, 1957], but unfortunately there was not enough time to implement this properly. During the work on the Smith-predictor a rudimentary velocity estimator was developed. This provided smooth estimates of the relative velocity of the ROV rather than the choppy derivative of the position and could also be used as a current estimator.

Since the focus was on controlling the Merlin in 4 DOFs, the roll and pitch was only controlled with simple P controllers. This caused problems because if the current got too high the effects of the tether cable caused large movement in the roll and pitch DOFs that the P controllers where unable to handle. The simulations had to be done with a relatively conservative current of $v_c = 0.3\text{knots} \approx 0.15[\text{m/s}]$ towards 0° to avoid this.

The following tests where carried out:

Test 1: Hold position, then move 2 meters in North.

Test 2: Hold position, then move 2 meters in East.

Test 3: Hold position, then move 2 meters in North and rotate 90°.

Test 4: Hold position, then move 5 meters in North and 1 meter up.

Table 6.7 contains the values for the controller parameters used in the simulator. The controllers had to be tuned differently from the Simulink model due to the difference in mass and damping and because of the time delay. $\boldsymbol{\Gamma}$ had to be chosen as a constant value in N, E and D to achieve proper behaviour.

Since the simulator is very advanced, no simulation is exactly the same. The drag from the tether cable will be different in each simulation and the initial position for the step change may be different. The latter part should not affect the overall performance.

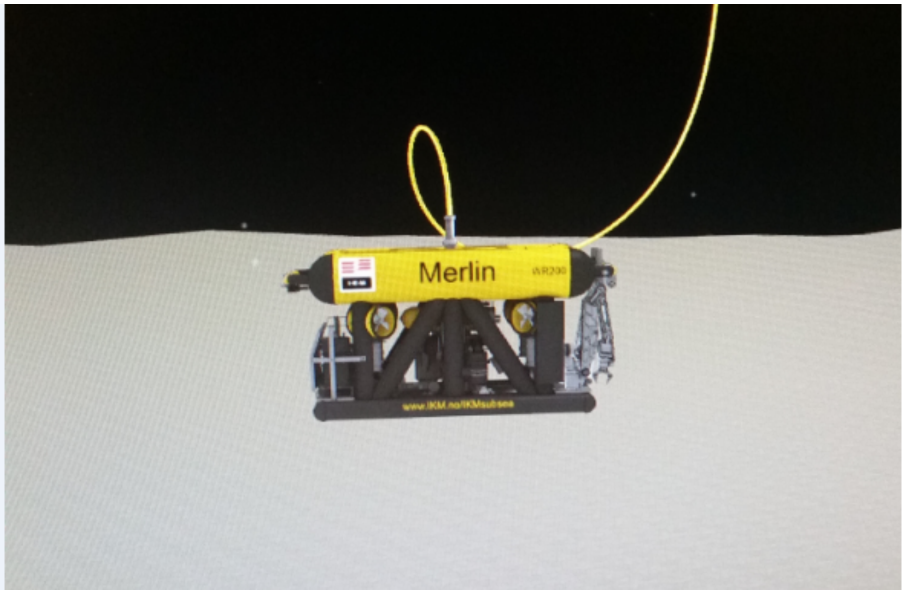


Figure 6.33: Screenshot of the Merlin WR200 during station keeping. The tether cable can be seen on top of the ROV, coiling down on the port side and causing a small roll angle.

Test 1

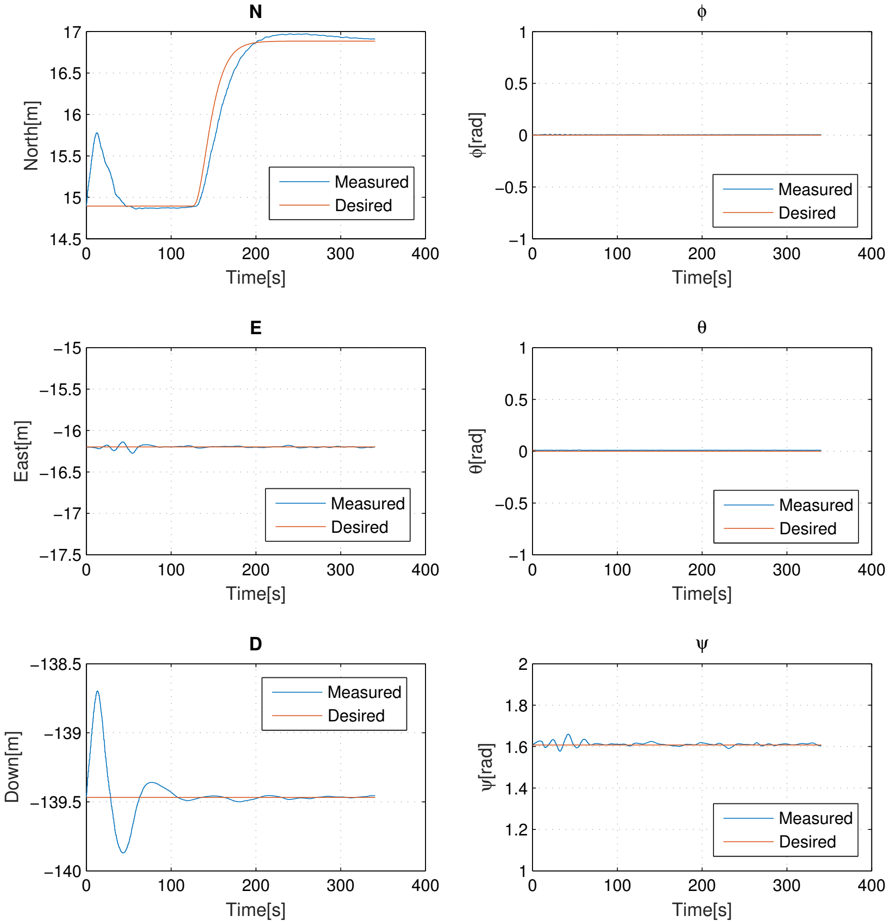


Figure 6.34: Test 1: Hold position, then move 2 meters in North. Down position and heading angle should be the same as initial and roll and pitch should be 0.

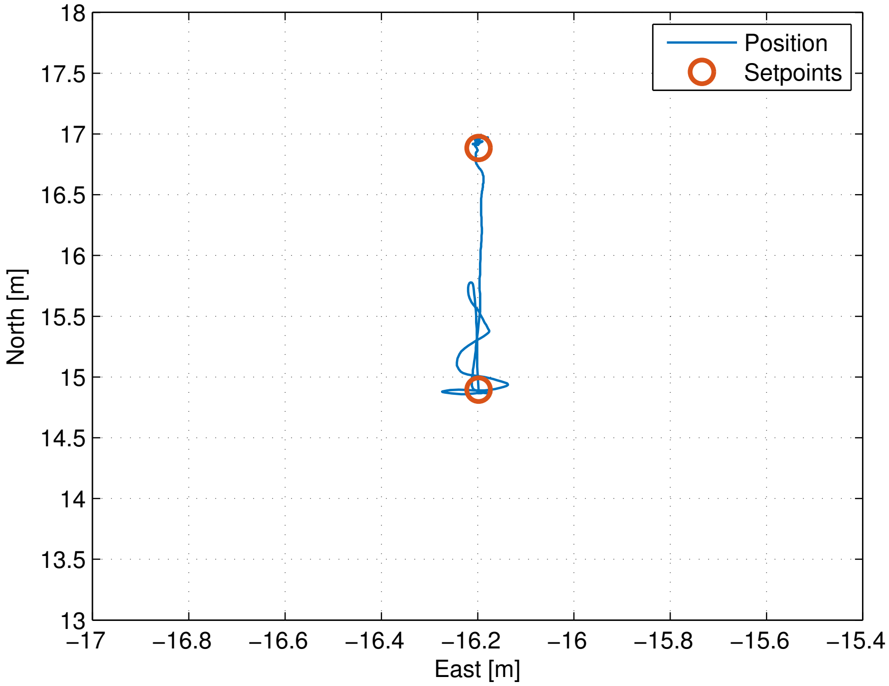


Figure 6.35: Test 1: x-y plot of the position and the setpoints when moving 2 meters North

Discussion of results

The results from figure 6.34 show that the system is very slow. It takes approximately 50 seconds to compensate the current disturbance in North. The step is also done very slowly because of the conservative gains. If the sluggishness is ignored, the step response is quite good. Low overshoot in North, almost no influence in East, roll, pitch and yaw. The ROV has a small pitch angle caused by the drag from the tether cable. The Down position is stabilized at the desired value, though with some oscillations. The influence from the step is small. The simulation was stopped a little early, but convergence in North can be seen in figure 6.38.

Test 2

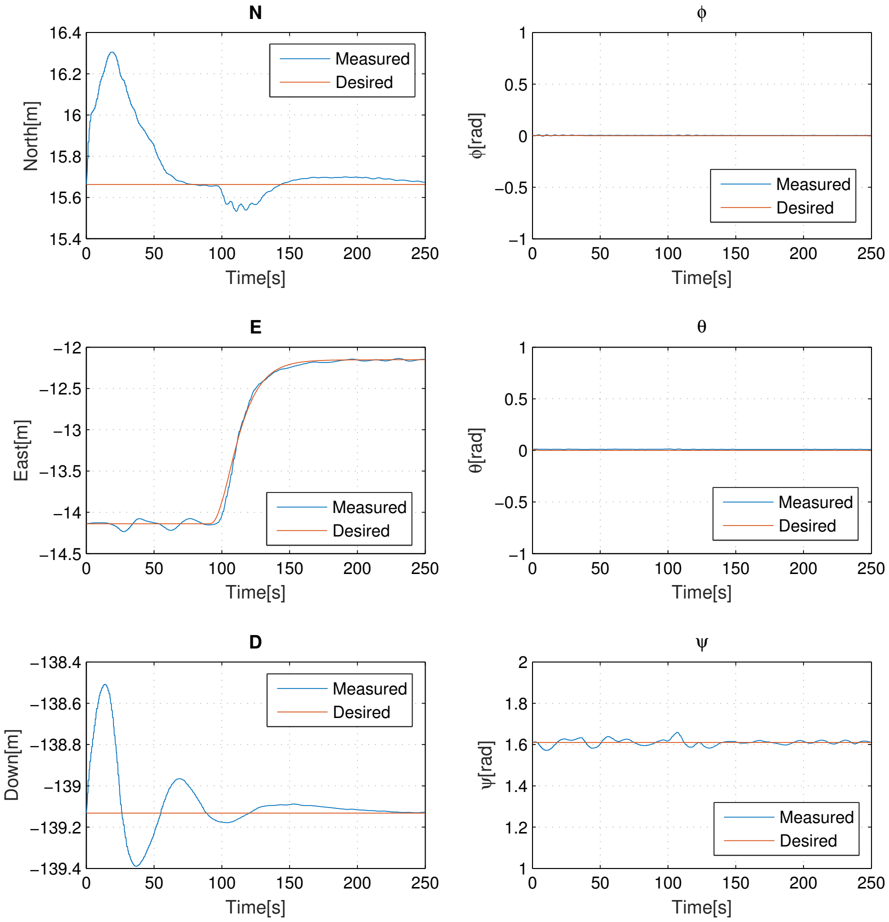


Figure 6.36: Test 2: Hold position, then move 2 meters in East. Depth and heading should be the same as initial and roll and pitch should be 0.

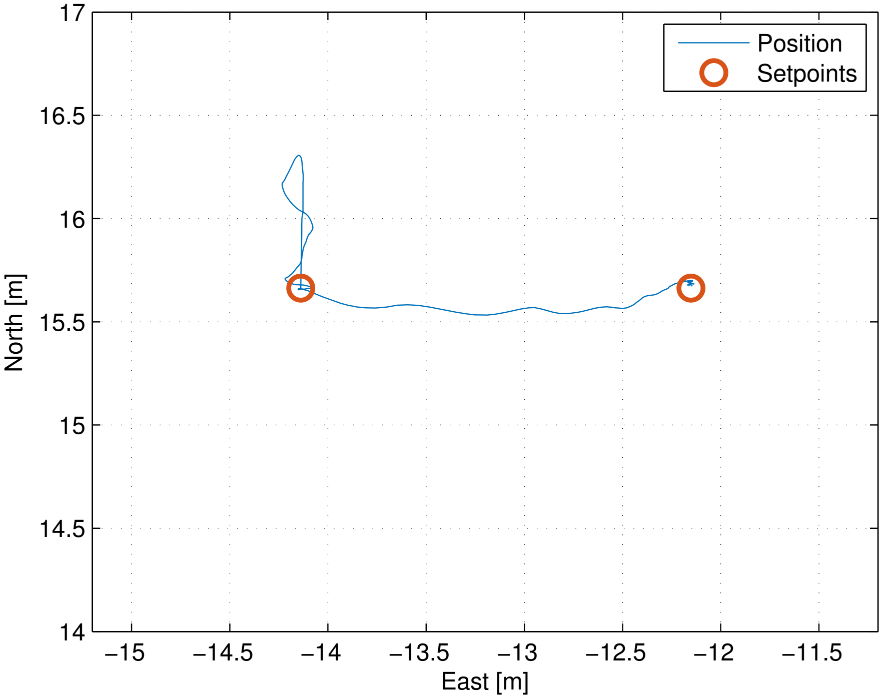


Figure 6.37: Test 2: x-y plot of the position and the setpoints when moving 2 meters East

Discussion of results

Like in test 1, the system is slow to compensate for the current in North. Some fluctuations are present in East as well. This may be caused by the tether cable. The step in East, however, is carried out very nicely. The ROV follows the reference almost perfectly. The influence on North is stronger than the equivalent influence on East was in test 1. Roll and pitch are kept very close to 0, but yaw is slightly more influenced due to the coupling between East and yaw. Down position is almost exactly like in test 1. It is stabilized and not influenced much by the step change.

Test 3

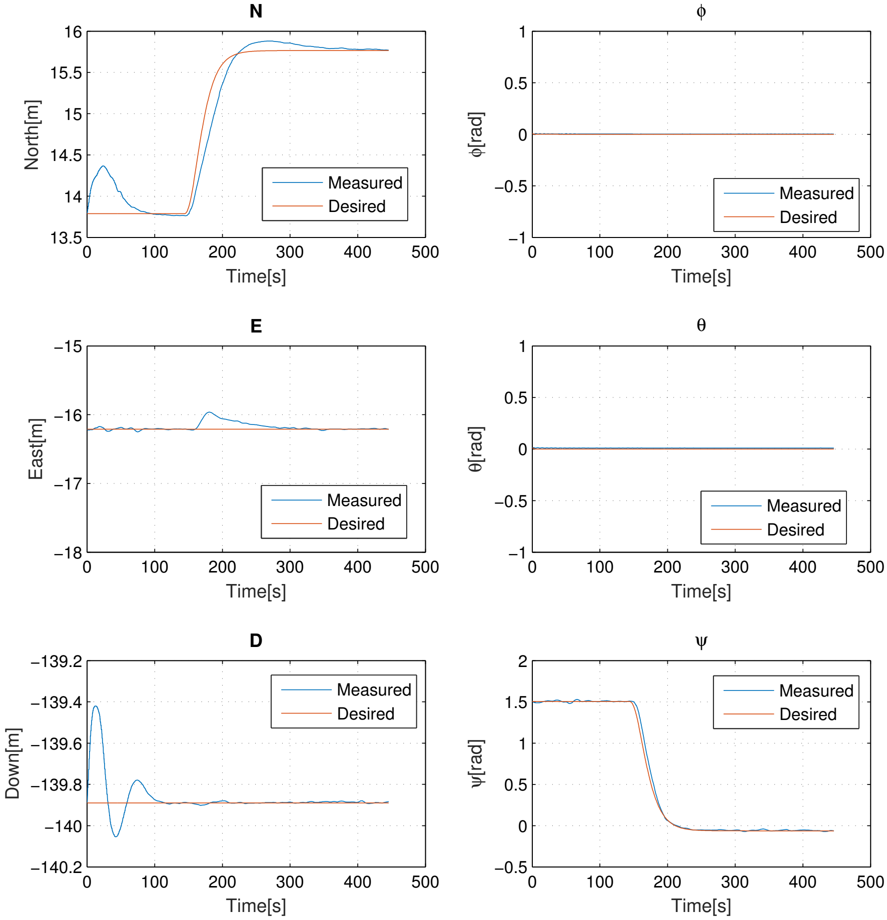


Figure 6.38: Test 3: Hold position, then move 2 meters in North and rotate 90° at the same time. Keep Depth at same level and roll and pitch in 0.

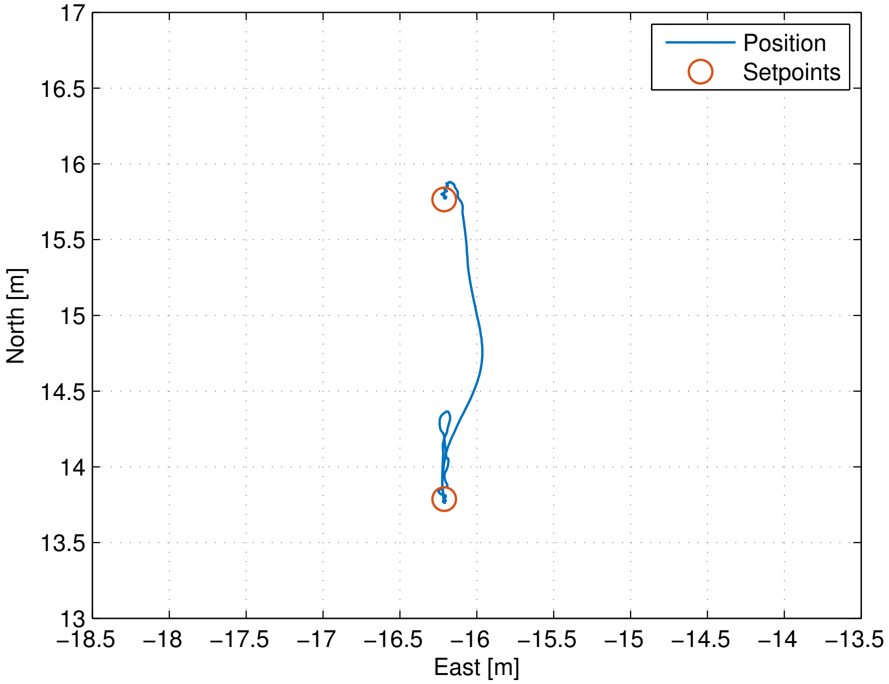


Figure 6.39: Test 3: x-y plot of the position and setpoints when moving 2 meters in North and rotating 90°.

Discussion of results

In this test, the controller is put to the test when the ROV is commanded to move 2 meters in North while rotating 90°. As can be seen from figure 6.38 the same sluggish behaviour to compensate the current is apparent. The step change is carried out very nicely. The reference is followed almost perfectly in yaw and with some deviations and overshoot in North, but it settles at the desired value. The coupling with East is stronger due to the rotation, but the error is less than 30cm. Also here, the Down position is kept at the desired value after settling and roll and pitch is kept at 0. The simulation was long enough to show proper convergence of the North position. It is clear that not using adaptation in yaw still causes a very good result, since this DOF is not under the influence of any constant disturbances.

Test 4

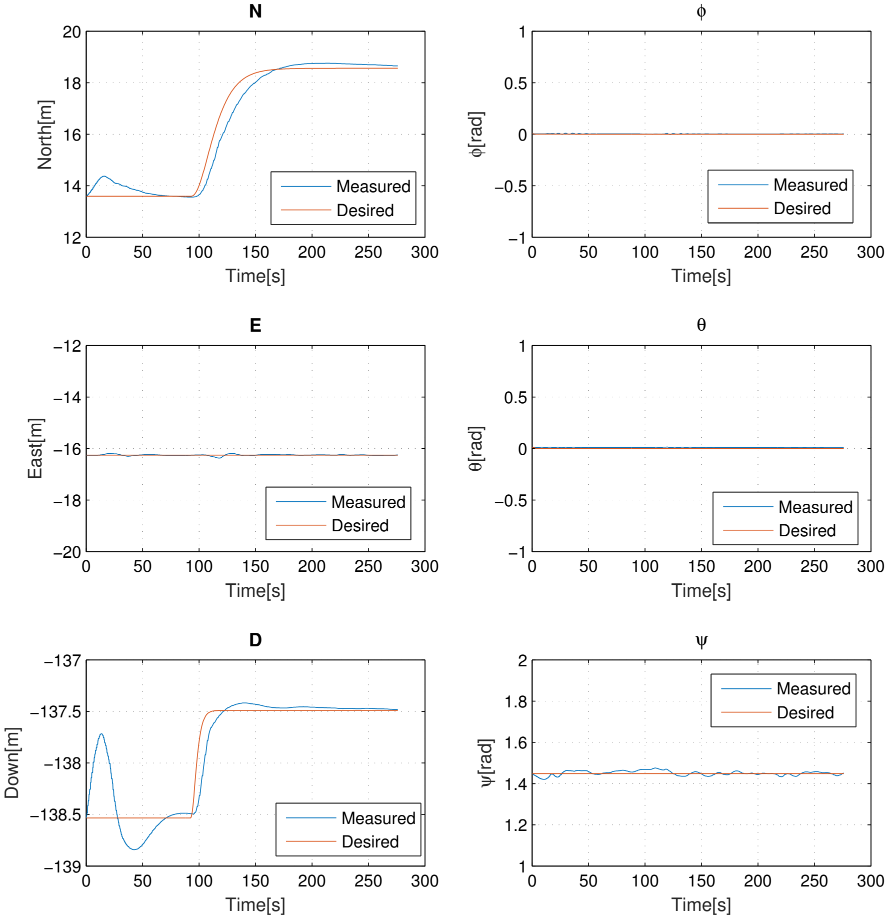


Figure 6.40: Test 4: Hold position, then move 5 meters in North and 1 meter up. Keep East, roll, pitch and yaw at initial values.

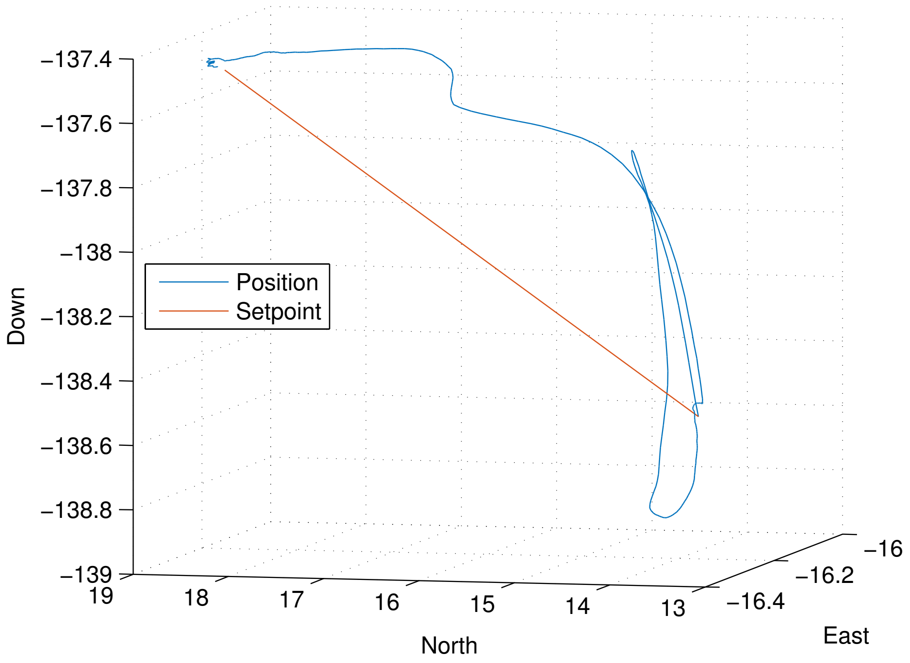


Figure 6.41: Test 4: x-y-z plot of the position and setpoints when moving 2 meters in North and 1 meter up.

Discussion of results

This test considers a larger step in North while also moving 1 meter up. The Down position does not settle before the step occurs, which may be why there is some overshoot and slow convergence to the desired value. The step in North is similar to test 1 and 3, but like in test 1, the simulation was ended a little early so the final convergence is not showed. The 3D plot in figure 6.41 gives an overview of the setpoint change and errors.

6.4.3 Identification of dynamic model

Since the behaviour was so different from the one in the Simulink model and to develop a rudimentary estimator, it was of interest to identify the system parameters used in the dynamic model in the simulator. A step response was investigated in surge, sway and yaw. A step from 0[N] to 8000[N] was applied at $t = 46$[s] and the velocity measured. That is, the derivative of the position was calculated, since the only available measurement from the simulator was the position. Based on the step response, the dynamic models was approximated by 1st order transfer functions on the form:

$$H(s) = \frac{K}{1+Ts}e^{-\tau s} \tag{6.4.1}$$

with $K = \frac{\Delta y}{\Delta u}$. T is the time constant which is the time it takes the process to reach 63% of the finite value and τ is the time delay of the step response.

Surge speed model

The surge speed model was found to be:

$$\begin{aligned} &\Delta u_u = 8000 \text{ [N]},\ \Delta y_u = 4 \text{ [m/s]},\ \tau_u = 1.2 \text{ [s]},\ T_u = 3.5 \text{ [s]} \\ &H_u(s) = \frac{0.0005}{3.5s+1}e^{-1.2s} \end{aligned} \tag{6.4.2}$$

As seen from figure 6.42, the model is quite a good fit, but it must be assumed that some uncertainty exists.

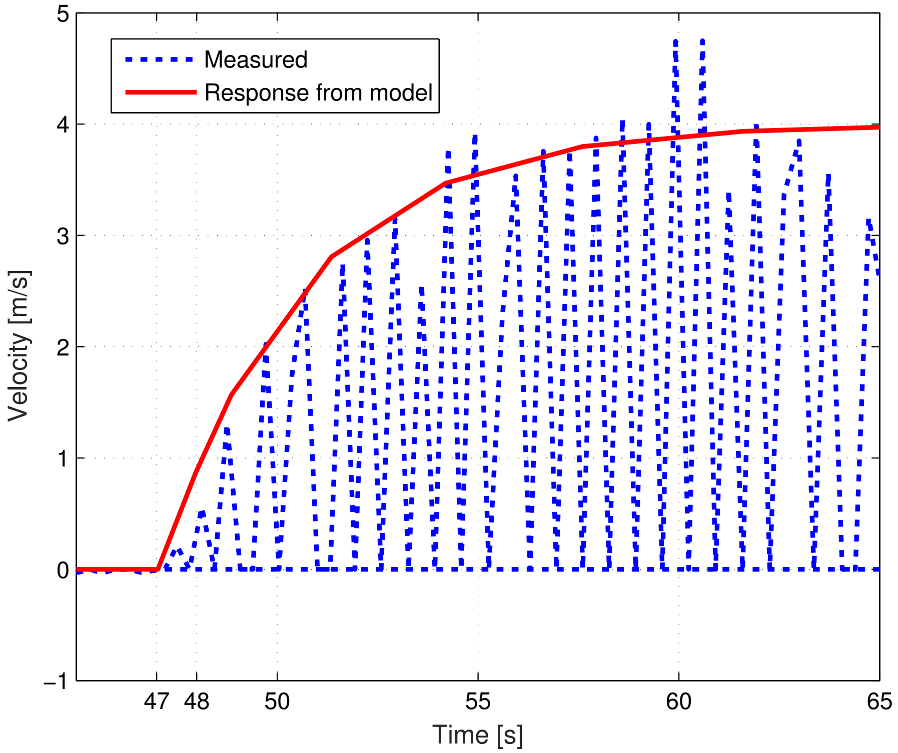


Figure 6.42: The measured surge speed response compared to the derived surge speed model.

Sway speed model

The sway speed model was found to be:

$$\begin{aligned}\Delta u_v &= 8000 \text{ [N]},\ \ \Delta y_v = 2.8 \text{ [m/s]},\ \ \tau_v = 2.5\text{[s]},\ \ T_v = 2.1\text{[s]} \\ H_v(s) &= \frac{0.00035}{2.1s+1}e^{-2.5s}\end{aligned} \tag{6.4.3}$$

In figure 6.43 the step response of the model is compared to the actual response and the model fits.

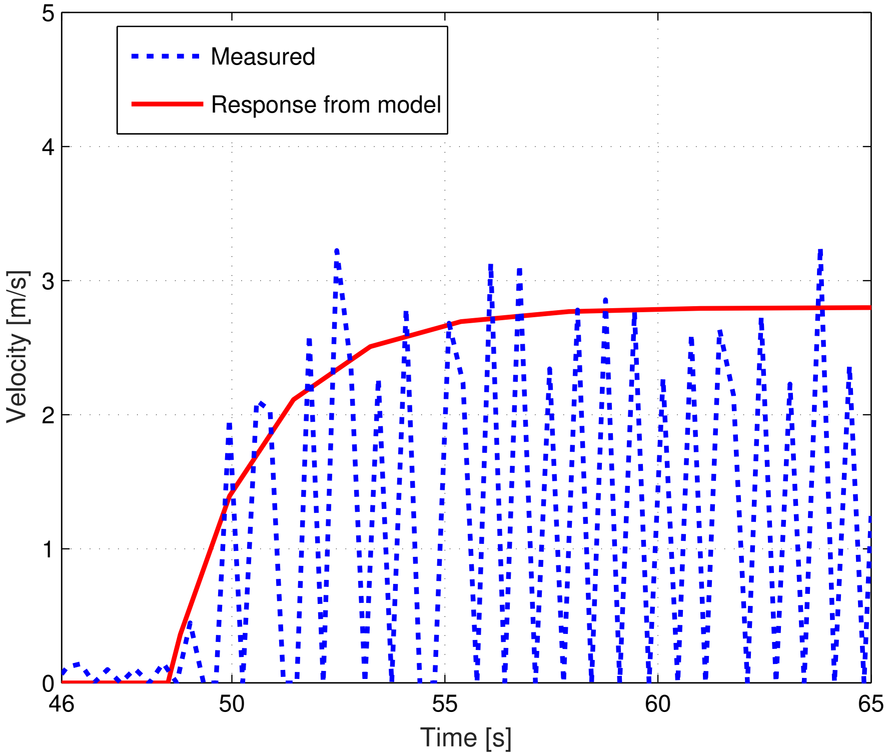


Figure 6.43: The measured sway speed response compared to the derived sway speed model.

Yaw rate model

The yaw rate model was found to be:

$$\begin{aligned}
&\Delta u_r = 8000 \text{ [N]}, \;\; \Delta y_r = 3 \text{ [m/s]}, \;\; \tau_r = 3\text{[s]}, \;\; T_r = 2.2\text{[s]} \\
&H_r(s) = \frac{0.000375}{2.2s + 1} e^{-3s}
\end{aligned} \tag{6.4.4}$$

The comparison is presented in figure 6.44 and it shows a good fit.

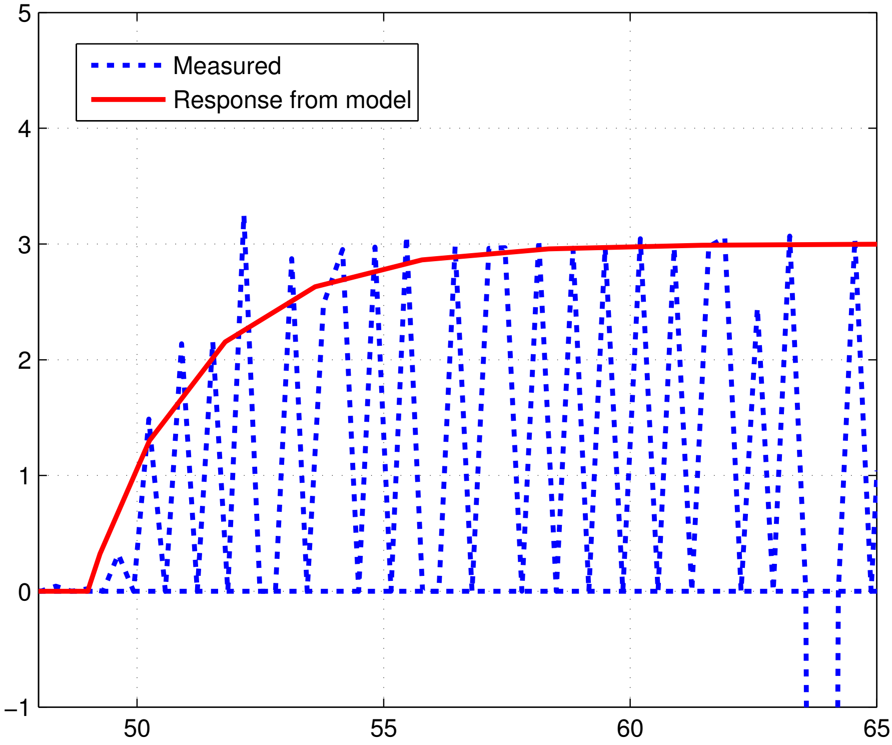


Figure 6.44: The measured yaw rate response compared to the derived yaw rate model

Parameter	Model	Simulator
$m - X_{\dot{u}}$	3502	7000
$m - Y_{\dot{v}}$	3502	6000
$m - N_{\dot{r}}$	3031	5867
$-(X_u + X_{\|u\|u})$	1321	2000
$-(Y_v + Y_{\|v\|v})$	2525	2857
$-(N_r + N_{\|r\|r})$	192	2667

Table 6.8: Table presenting the parameters used in the Simulink model and the parameters identified in the simulator when assuming no time delay.

Compared to Simulink model

Writing the transfer functions $H_u(s)$, $H_v(s)$ and $H_r(s)$ from (6.4.2)-(6.4.4) as linear mass-damper systems as presented in [Fossen, 2011, Ch.12.2.1] with Δy_u, Δy_v and Δy_r as the velocities in the body fixed x and y directions and as the angular velocity about the z-axis and assuming no time-delay we get:

$$\begin{aligned}
\frac{T_u}{K_u}\ddot{x} + \frac{1}{K_u}\dot{x} &= (m + X_{\dot{u}})\ddot{x} + (X_{|u|u})\dot{x} &&= X \\
\frac{T_v}{K_v}\ddot{y} + \frac{1}{K_v}\dot{y} &= (m + Y_{\dot{v}})\ddot{y} + (Y_{|v|v})\dot{y} &&= Y \\
\frac{T_r}{K_r}\ddot{r} + \frac{1}{K_r}\dot{r} &= (m + N_{\dot{r}})\ddot{r} + (N_{|r|r})\dot{r} &&= N
\end{aligned} \tag{6.4.5}$$

A comparison of this to the model found using the mass and damping matrices from chapter 3 is presented in table 6.8. The difference in mass, and damping in yaw is very big. It is unknown what causes this difference, but it might be the tether cable or that the model used in the simulator simply is different. However, the controller performs very well after some tuning and since it is not dependent on the model parameters, the robustness is proven.

Velocity estimator

A rudimentary relative velocity estimator was developed based on the derived dynamic model from the step responses. The model for the estimator is based on the relative velocity representation of the kinetics as in (2.3.7):

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{6.4.6}$$

$$\boldsymbol{M}\dot{\hat{\boldsymbol{\nu}}}_r + \boldsymbol{D}\hat{\boldsymbol{\nu}}_r = \boldsymbol{\tau} \tag{6.4.7}$$

The estimator in 3 DOFs takes the form:

$$\begin{bmatrix} \dot{\hat{u}}_r \\ \dot{\hat{v}}_r \\ \dot{\hat{r}}_r \end{bmatrix} = \begin{bmatrix} -\frac{1}{3.5} & 0 & 0 \\ 0 & -\frac{1}{2.1} & 0 \\ 0 & 0 & -\frac{1}{2.2} \end{bmatrix} \begin{bmatrix} \hat{u}_r \\ \hat{v}_r \\ \hat{r}_r \end{bmatrix} + \begin{bmatrix} \frac{0.0005}{3.5} & 0 & 0 \\ 0 & \frac{0.00035}{2.1} & 0 \\ 0 & 0 & \frac{0.000375}{2.2} \end{bmatrix} \begin{bmatrix} X \\ Y \\ N \end{bmatrix} \tag{6.4.8}$$

The input X, Y and N is the same input as applied to the acutal ROV. To imitate the time delay found in the models (6.4.2) - (6.4.4) the input is held back an approximate amount of simulation steps. In this test, the desired position is constant, but the ROV is commanded to rotate 90° at $t \approx 150$. The result is shown in figure 6.45. Unfortunately, there was not enough time to implement the estimated velocity in the control loop, but for future work it could be useful.

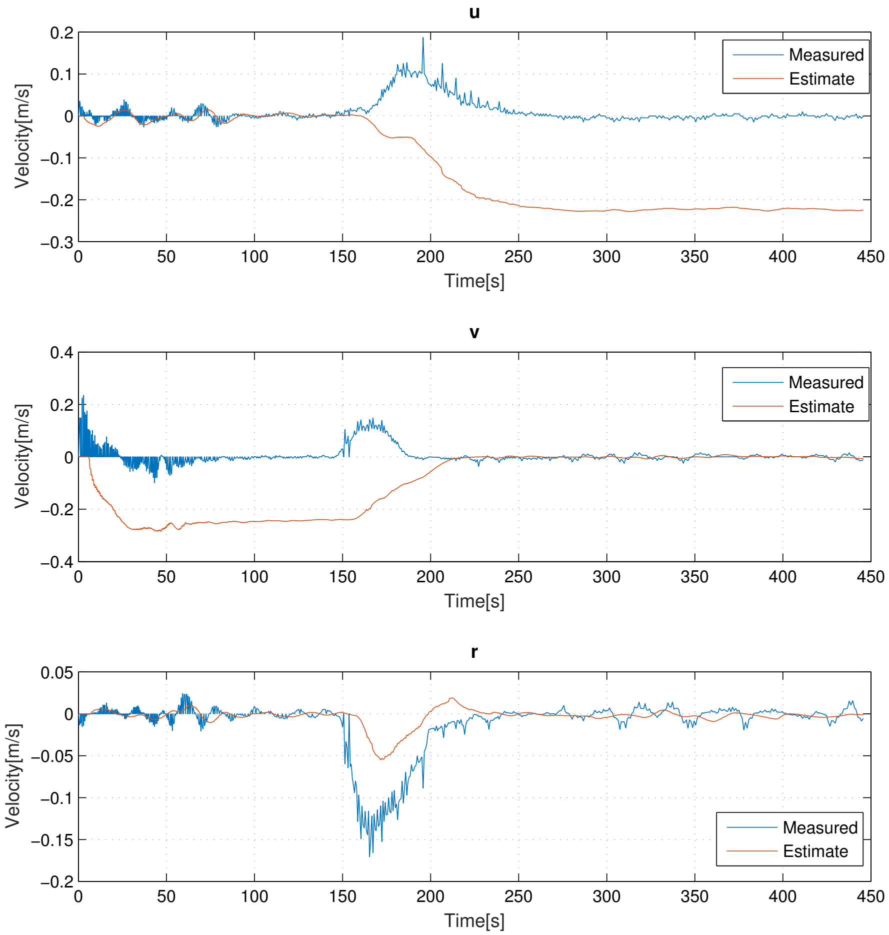


Figure 6.45: Estimated velocity plotted against relative velocity. The difference is due to current and tether disturbances. At $t \approx 150$ the ROV turns from $-90°$ to $0°$.

Discussion of results

Figure 6.45 shows the derivative of the position which can be interpreted as the velocity $\boldsymbol{\nu} = \boldsymbol{\nu}_r - \boldsymbol{\nu}_c$. The estimated relative velocity $\hat{\boldsymbol{\nu}}_r$ is also shown. For the first 150 seconds the ROV is heading towards −90° and the current, which is constant in $\{n\}$, is causing the estimated relative velocity of ≈ -0.21 m/s. The derivative of the position however, is ≈ 0. This means that the current in $\{n\}$ can be calculated as:

$$\hat{\boldsymbol{\nu}}_c^n = \boldsymbol{R}^*(\psi)\,(\boldsymbol{\nu} - \hat{\boldsymbol{\nu}}_r))$$

$$\begin{bmatrix} \hat{u}_c \\ \hat{v}_c \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} u - \hat{u}_r \\ v - \hat{v}_r \end{bmatrix}$$

$$\begin{bmatrix} \hat{u}_c \\ \hat{v}_c \end{bmatrix} = \begin{bmatrix} -\hat{v}_r \\ 0 \end{bmatrix} = \begin{bmatrix} 0.21 \\ 0 \end{bmatrix}$$

The same method may be used after the ROV has turned 90° yielding the same results. The current is known to be $\boldsymbol{\nu}_c^n = \begin{bmatrix} 0.15 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ [m/s], so the estimation error may be caused by tether cable drag, model errors or the simplicity of the estimator. The estimate of the yaw rate r follows the measured state, but during the rotation it shows a lower value. This is probably caused by model errors or the simplicity of the estimator.

Chapter 7

Future work and conclusion

7.1 Future work

Sea trial

There was no time for a sea trial to test the theories in this master thesis. It is highly recommended that this is carried out to verify the results in a full scale test. The tests done in the simulator are a very good alternative, but nothing beats the real thing.

Dynamic model

It was found during the testing in the simulator that the dynamic model used in the computer simulations did not match the dynamic model in the simulator. It is suggested that a new and more extensive dynamic model is developed based on the simulator at IKM Subsea. The parameters of the mass, damping and Coriolis matrices should be identified using basic control theory or more extensive system identification methods, like those mentioned in [Ioannou and Sun, 2012]. The model should include thruster dynamics and tether cable dynamics.

Controller

Based on the more extensive dynamic model, the P-CABS controller should be re-verified. The P-CABS controller could be improved to include the thruster and tether dynamics in the control loop and more emphasis should be put on also controlling the roll and pitch angles.

Path following

Further improve the path following control system developed in this thesis by utilizing the new dynamic model and improved controller. Investigate the possibilities of using a different steering law than LOS or develop a new steering law for 6 DOF underwater vehicles. The path following system developed in this thesis should be tested in the simulator at IKM Subsea for verification.

Observer/Estimator

An observer could be developed to improve the overall control system. The observer may be based on the Kalman filter from [Knausgård, 2013] or other methods like the nonlinear passive observer [Fossen and Strand, 1999] could be put to use. This observer could include current estimates as well as position and attitude estimates. The observer can be tested in the Simulink model by introducing noisy measurements and in both the simulator at IKM Subsea and in sea trials.

HMI

A more intuitive Human Machine Interface (HMI) should be developed for the DP and path following system. Now, the system only exists as a Matlab script in pure code. There are several advantages to a good HMI system, for instance easier path planning and set point selection, simple logging and easy access to data and simpler tuning of controller gains.

7.2 Conclusion

The goal of this thesis was to develop a new controller for path following and tracking control of ROVs in general and the Merlin WR200 in particular. The Merlin WR200 is only fitted with automatic control of depth, altitude and heading, so to increase the level of autonomy it is of interest to include position control in the horizontal plane as well. Some controllers demand knowledge of the system parameters. In reality, these system parameters are not exactly known, so working around this demand was one of the fundamentals in the design of the new control law. The derivation of the control law was attempted as simple and readable as possible.

A new control law was derived by using parameter and current adaptation and integrator backstepping. The convergence of the error states to zero was proven and the controller was verified through simulations in Simulink and in the simulator at IKM Subsea.

The developed controller was compared to 2 existing controllers in a simulation study. The controller from [Antonelli et al., 2003] which has adaptation of current and gravitational terms and the Adaptive Backstepping with Bound Estimation (ABSwB) controller from [Patompak and Nilkhamhang, 2012]. The ABSwB controller has the same parameter adaptation part as was desired for the controller in this thesis and hence it was included in the comparison. The developed P-CABS controller showed equal or lower error than the controller from [Antonelli et al., 2003] in all cases. The controller from [Patompak and Nilkhamhang, 2012] had smaller mean and RMS value of the error in some cases, but the control input was very choppy. The control input from the P-CABS controller was smooth at all times.

Path following of already driven paths was also one of the goals of this thesis. The ROV had to add waypoints while driving through the ocean space and to do so, a circle of acceptance for adding waypoints was defined. A modified version of the popular LOS steering law was developed. This method considers the ROVs position and heading relative to the stored waypoints in order to calculate the desired heading and velocity. This is important because the ROV may be in a narrow area where turning around is impossible. The ROV should therefore follow the exact same path back to the origin with the same heading angle as it had while driving the path in the first place. This was tested and accomplished in simulations in Simulink for S-shaped paths showing some crosstrack error during turning caused by the waypoint switching. Unfortunately there was not enough time to implement this in the simulator at IKM Subsea.

During testing in the simulator it was found that the dynamic model used in the simulator did not match the dynamic model used in Simulink, so further work should be done on identifying this model. The roll and pitch was only controlled with simple P controllers, so the disturbances from the tether cable where not compensated properly. This should be looked at in the future by extending the parameter adaptation and disturbance rejection to all 6 DOFs. The path following system should be implemented in the simulator at IKM Subsea for verification and the controller should be tested in a sea trial to verify the performance in a real setting. The DP system, consisting of the station keeping and trajectory tracking did work in the simulator and the ROV was able to hold its position under the influence of ocean currents and change the position to the desired position in North, East, Down and ψ. The controller had to be tuned quite conservative because of time delays in the dynamic model causing a slow but otherwise very good response.

Bibliography

Ikm subsea, merlin academy. `http://www.ikm.no/ikm-subsea-norge/merlin-academy`. Acessed 30.04.2015.

Gianluca Antonelli, F Caccavle, Stefano Chiaverini, and Giuseppe Fusco. A novel adaptive control law for autonomous underwater vehicles. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 447–452. IEEE, 2001.

Gianluca Antonelli, Fabrizio Caccavale, Stefano Chiaverini, and Giuseppe Fusco. A novel adaptive control law for underwater vehicles. *Control Systems Technology, IEEE Transactions on*, 11(2):221–232, 2003.

Jens G Balchen, Nils A Jenssen, Eldar Mathisen, and Steinar Sælid. A dynamic positioning system based on kalman filtering and optimal control. 1980.

Even Børhaug. *Nonlinear control and Synchronization of Mechanical Systems*. PhD thesis, Norwegian University of Science and Technology, 2008.

Roberto Cristi, Fotis A Papoulias, and Anthony J Healey. Adaptive sliding mode control of autonomous underwater vehicles in the dive plane. *Oceanic Engineering, IEEE Journal of*, 15(3):152–160, 1990.

JPVS Da Cunha, Ramon R Costa, and Liu Hsu. Design of a high performance variable structure position control of rovs. *Oceanic Engineering, IEEE Journal of*, 20(1):42–55, 1995.

Olav Egeland and Jan Tommy Gravdahl. *Modeling and simulation for automatic control*, volume 76. 2002.

Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

Thor I Fossen and Svein P Berge. Nonlinear vectorial backstepping design for global exponential tracking of marine vessels in the presence of actuator dynamics. In *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, volume 5, pages 4237–4242. IEEE, 1997.

Thor I Fossen and Anastasios M Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, 2015.

Thor I Fossen and Kristin Y Pettersen. On uniform semiglobal exponential stability (usges) of proportional line-of-sight guidance laws. *Automatica*, 50(11): 2912–2917, 2014.

Thor I Fossen and Jann Peter Strand. Passive nonlinear observer design for ships using lyapunov methods: full-scale experiments with a supply vessel. *Automatica*, 35(1):3–16, 1999.

Thor I Fossen, Morten Breivik, and Roger Skjetne. Line-of-sight path following of underactuated marine craft. *Proceedings of the 6th IFAC MCMC, Girona, Spain*, pages 244–249, 2003.

Thor Inge Fossen. *Nonlinear modelling and control of underwater vehicles.* PhD thesis, Norwegian University of Science and Technology, 1991.

John-Morten Godhavn, Thor I Fossen, and Svein P Berge. Non-linear and adaptive backstepping designs for tracking control of ships. *International Journal of Adaptive Control and Signal Processing*, 12(8):649–670, 1998.

Christian Holden and Kristin Y Pettersen. Robust globally exponentially stabilizing control law for fully actuated 6-dof auvs. In *Control Applications in Marine Systems*, volume 7, pages 343–348, 2007.

Liu Hsu. Variable structure model-reference adaptive control (vs-mrac) using only input and output measurements: the general case. *Automatic Control, IEEE Transactions on*, 35(11):1238–1243, 1990.

Petros A Ioannou and Jing Sun. *Robust adaptive control.* Courier Dover Publications, 2012.

Hassan K Khalil. *Nonlinear systems*, volume 3. Prentice hall New Jersey, 2002.

Lasse Muri Knausgård. Development of dp system for merlin wr200 rov, project thesis, 2012.

Lasse Muri Knausgård. Development of dp system for merlin wr200 rov, master thesis, 2013.

Sergey Edward Lyshevski. Autopilot design for highly maneuverable multipurpose underwater vehicles. In *American Control Conference, 2001. Proceedings of the 2001*, volume 1, pages 131–136. IEEE, 2001.

Signe Moe, Walter Caharija, Kristin Y Pettersen, and Ingrid Schjolberg. Path following of underactuated marine surface vessels in the presence of unknown ocean currents. In *American Control Conference (ACC), 2014*, pages 3856–3861. IEEE, 2014.

Sveinung Johan Ohrem. Development of dp system for merlin wr200 rov, project thesis, 2014.

Pakpoom Patompak and Itthisek Nilkhamhang. Adaptive backstepping sliding-mode controller with bound estimation for underwater robotics vehicles. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2012 9th International Conference on*, pages 1–4. IEEE, 2012.

Roger Skjetne, Thor I Fossen, and Petar Kokotovic. Output maneuvering for a class of nonlinear systems. Citeseer, 2002.

J-JE Slotine and Li Weiping. Adaptive manipulator control: A case study. *Automatic Control, IEEE Transactions on*, 33(11):995–1003, 1988.

Otto Smith. Closer control of loops with dead time. 1957.

Society of Naval Architects and Marine Engineers (U.S.). Technical and Research Committee. Hydrodynamics Subcommittee SNAME. Nomenclature for treating the motion of a submerged body through a fluid: Report of the american towing tank conference. 1950. URL `http://books.google.no/books?id=VqNFGwAACAAJ`.

Asgeir J. Sørensen. Structural issues in the design and operation of marine control systems. *Annual Reviews in Control*, 29(1):125 – 149, 2005. ISSN 1367-5788. doi: http://dx.doi.org/10.1016/j.arcontrol.2004.12.001. URL `http://www.sciencedirect.com/science/article/pii/S136757880500012X`.

Asgeir J. Sørensen. *Marine Control Systems, lecture notes.* Department of Marine Technology, NTNU, 2013.

Vadim I Utkin. Survey paper variable structure systems with sliding modes. *IEEE Transactions on Automatic control*, 22(2), 1977.

K David Young, Vadim I Utkin, and Umit Ozguner. A control engineer's guide to sliding mode control. *IEEE transactions on control systems technology*, 7 (3):328–342, 1999.

Kangwu Zhu and Linyi Gu. A mimo nonlinear robust controller for work-class rovs positioning and trajectory tracking control. In *Control and Decision Conference (CCDC), 2011 Chinese*, pages 2565–2570. IEEE, 2011.

Appendices

Appendix A

Merlin Data

Merlin WR200 - complete (new)

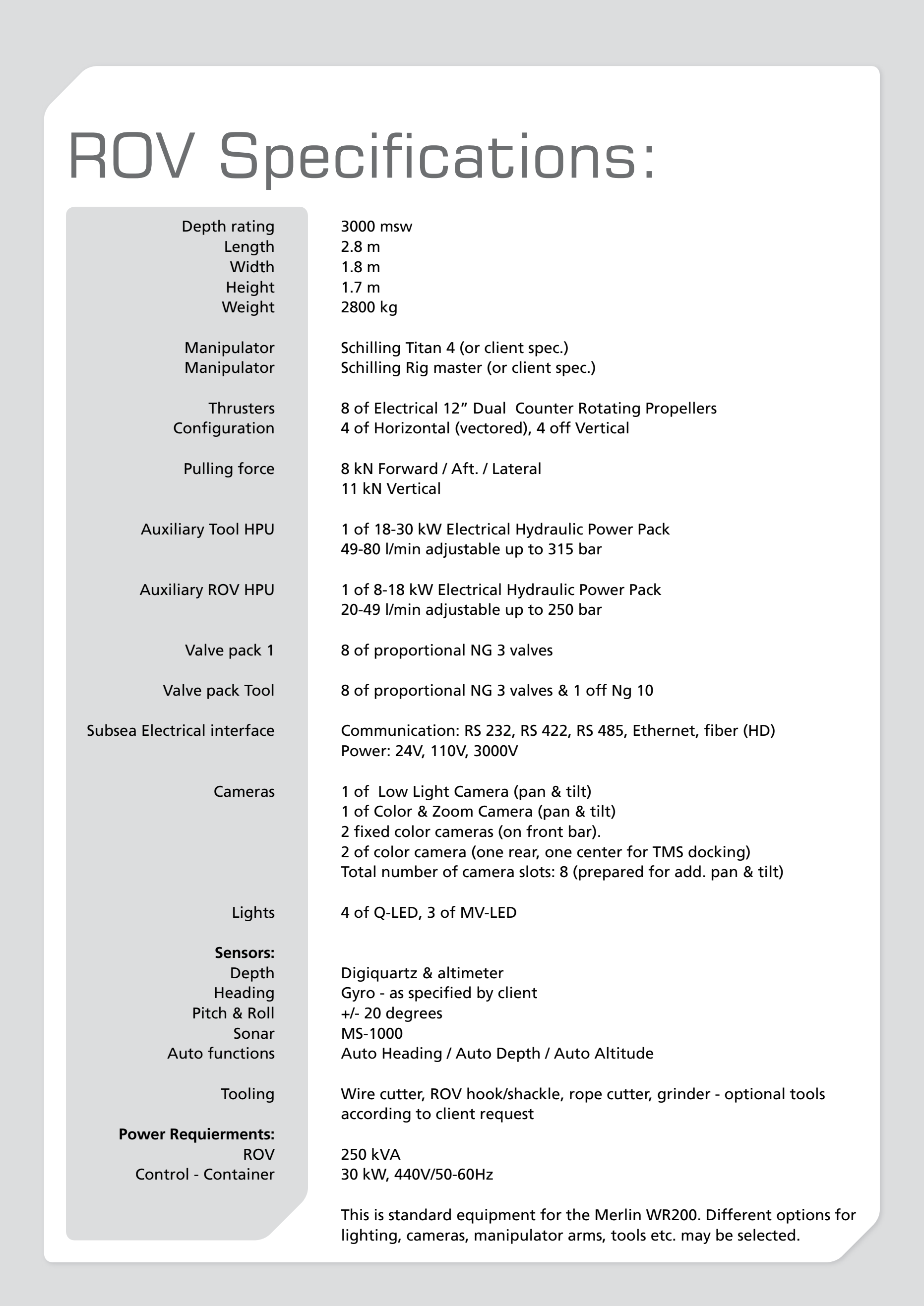
Physical Properties for IKM-1013251_Ver2				
General Properties:				
	Material:	{}		
	Density:	1,039 g/cm^3		
	***Mass:**	3019,249 kg (Relative Error = 0,000245%)		
	Area:	186542278,399 mm^2 (Relative Error = 0,005466%)		
	Volume:	2,905829086726E+009 mm^3 (Relative Error = 0,000245%)		
****Center of Gravity:**				
	X:	-2,341 mm (Relative Error = 0,000245%)		
	Y:	3,014 mm (Relative Error = 0,000245%)		
	Z:	-21,193 mm (Relative Error = 0,000245%)		
****Mass Moments of Inertia with respect to Center of Gravity(Calculated using negative integral)**				
	Ixx	1819601608,273 kg mm^2 (Relative Error = 0,000245%)		
	Iyx Iyy	-2696884,126 kg mm^2 (Relative Error = 0,000245%)	3,064990196615E+009 kg mm^2 (Relative Error = 0,000245%)	
	Izx Izy Izz	120369718,084 kg mm^2 (Relative Error = 0,000245%)	-7207982,764 kg mm^2 (Relative Error = 0,000245%)	2,887168166629E+009 kg mm^2 (Relative Error = 0,000245%)
****Mass Moments of Inertia with respect to Global(Calculated using negative integral)**				
	Ixx	1820985065,487 kg mm^2 (Relative Error = 0,000245%)		
	Iyx Iyy	-2675581,594 kg mm^2 (Relative Error = 0,000245%)	3,066362768103E+009 kg mm^2 (Relative Error = 0,000245%)	
	Izx Izy Izz	120219937,803 kg mm^2 (Relative Error = 0,000245%)	-7015121,476 kg mm^2 (Relative Error = 0,000245%)	2,887212140375E+009 kg mm^2 (Relative Error = 0,000245%)
****Principal Moments of Inertia with respect to Center of Gravity**				
	I1:	1806195213,215 kg mm^2 (Relative Error = 0,000245%)		
	I2:	3,065330983149E+009 kg mm^2 (Relative Error = 0,000245%)		
	I3:	2,900233775152E+009 kg mm^2 (Relative Error = 0,000245%)		
****Rotation from Global to Principal**				
	Rx:	2,61 deg (Relative Error = 0,000245%)		
	Ry:	-6,34 deg (Relative Error = 0,000245%)		
	Rz:	-0,37 deg (Relative Error = 0,000245%)		

*Calculations are based on user overwritten values

**Values do not reflect user-overridden mass or volume

Appendix B

Merlin Specifications



ROV Specifications:

Depth rating	3000 msw
Length	2.8 m
Width	1.8 m
Height	1.7 m
Weight	2800 kg
Manipulator	Schilling Titan 4 (or client spec.)
Manipulator	Schilling Rig master (or client spec.)
Thrusters	8 of Electrical 12” Dual Counter Rotating Propellers
Configuration	4 of Horizontal (vectored), 4 off Vertical
Pulling force	8 kN Forward / Aft. / Lateral 11 kN Vertical
Auxiliary Tool HPU	1 of 18-30 kW Electrical Hydraulic Power Pack 49-80 l/min adjustable up to 315 bar
Auxiliary ROV HPU	1 of 8-18 kW Electrical Hydraulic Power Pack 20-49 l/min adjustable up to 250 bar
Valve pack 1	8 of proportional NG 3 valves
Valve pack Tool	8 of proportional NG 3 valves & 1 off Ng 10
Subsea Electrical interface	Communication: RS 232, RS 422, RS 485, Ethernet, fiber (HD) Power: 24V, 110V, 3000V
Cameras	1 of Low Light Camera (pan & tilt) 1 of Color & Zoom Camera (pan & tilt) 2 fixed color cameras (on front bar). 2 of color camera (one rear, one center for TMS docking) Total number of camera slots: 8 (prepared for add. pan & tilt)
Lights	4 of Q-LED, 3 of MV-LED
Sensors:	
Depth	Digiquartz & altimeter
Heading	Gyro - as specified by client
Pitch & Roll	+/- 20 degrees
Sonar	MS-1000
Auto functions	Auto Heading / Auto Depth / Auto Altitude
Tooling	Wire cutter, ROV hook/shackle, rope cutter, grinder - optional tools according to client request
Power Requierments:	
ROV	250 kVA
Control - Container	30 kW, 440V/50-60Hz

This is standard equipment for the Merlin WR200. Different options for lighting, cameras, manipulator arms, tools etc. may be selected.

Appendix C

Mathematics

C.1 Cross-product operator

The vector cross product × is

$$\boldsymbol{\lambda} \times \boldsymbol{a} = \boldsymbol{S}(\lambda)a \tag{C.1.1}$$

where $\boldsymbol{S} \in \ SS(3)$ is

$$\boldsymbol{S}(\boldsymbol{\lambda}) = -\boldsymbol{S}^T(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -\lambda_3 & \lambda_2 \\ \lambda_3 & 0 & -\lambda_1 \\ -\lambda_2 & \lambda_1 & 0 \end{bmatrix} \tag{C.1.2}$$

C.2 Persistence of Excitation(PE)

Persistence of Excitation is defined in [Ioannou and Sun, 2012] as:

Definition C.1: *A piecewise continuous signal vector $u : \mathcal{R}^+ \rightarrow \mathcal{R}^n$ is PE in mathcalR^n with a level of excitation $\alpha_0 > 0$ if there exists constantsα_1, $T_0 > 0$ such that:*

$$\alpha_1 \boldsymbol{I} \geq \frac{1}{T_0} \int_t^{t+T_0} u(\tau)u^T(\tau)d\tau \geq \alpha_0 \boldsymbol{I}, \ \forall \, t \geq 0 \tag{C.2.1}$$

In general, this means that the signal u has to be a rich signal. A rich signal could for instance be a sum of sin and cos signals. It is referred to [Ioannou and Sun, 2012] for a more thorough explanation on PE signals and examples and proofs.

C.3 Stability

From [Khalil, 2002], the following definitions of stability are collected

Consider the nonautonomous system

$$\dot{x} = f(x, t) \tag{C.3.1}$$

Definition C.2: *The equilibrium point $x = 0$ is*

- *stable, if for each $\epsilon > 0$ there is $\delta = \delta(\epsilon, t_0) > 0$ such that*

$$||x(t_0)|| < \delta \implies ||x(t)|| < \epsilon, \ \forall \ t \geq t_0 \geq 0$$

- *unstable if it is not stable*
- *asymptotically stable if it is stable and there is a constant $c = c(t_0)$ such that $x(t) \to 0$ as $t \to \infty$, for all $||x(t_0)|| < c$*

Stability can also be described using Lyapunov functions

Definition C.3: *The equilibrium point $x = 0$ is*

- *stable, if there exists a Lyapunov function $V(x)$ with the properties*

$$\begin{aligned} &V(0) = 0 \textit{ and } V(x) > 0 \ \forall \ x \neq 0 \\ &\dot{V}(x) \leq 0 \end{aligned}$$

- *asymptotically stable, if there exists a Lyapunov function $V(x)$ with the properties*

$$\begin{aligned} V(0) = 0 \textit{ and } V(x) > 0 \ \forall \ x \neq 0 \\ \dot{V}(x, t) < 0 \ \forall \ x \neq 0 \end{aligned}$$

- *exponentially stable, if there exists a Lyapunov function $V(x)$ with the properties*

$$\begin{aligned} k_1 ||x||^a \leq V(t, x) \leq k_2 ||x||^a \\ \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(x, t) \leq -k_3 ||x||^a \end{aligned}$$

  $\forall \ t \geq 0$, where k_1, k_2, k_3 and a are positive constants.

C.4 Differentiation of vectors

Differentiation of a vector $\vec(u)$ in a moving reference $\{b\}$ frame satisfies [Fossen, 2011]

$$\frac{{}^{i}d}{dt}\vec{u} = \frac{{}^{b}d}{dt}\vec{u} + \omega_{b/i} \times \vec{u} \tag{C.4.1}$$

Appendix D

Modified Least-Squares with Forgetting Factor

The following is gathered from [Ioannou and Sun, 2012, Ch. 4.3.6]:

Consider a a case where the unknown parameters appear in a linear form as:

$$z = \theta^T \phi \tag{D.0.1}$$

where θ is the vector of unknown parameters and ϕ is the vector of known signals. Consider a simple scalar plant:

$$y = \theta u + d_n \tag{D.0.2}$$

where d_n is a noise disturbance; y, $u \in \mathcal{R}^+$ and $u \in \mathcal{L}_\infty$. An approach to generate $\hat{\theta}$ is to minimize the cost function:

$$J(\hat{\theta}) = \frac{1}{2} \int_0^t (y(\tau) - \hat{\theta}(t)u(\tau))^2 d\tau \tag{D.0.3}$$

with respect to $\hat{\theta}$ at any given time t. The cost $J(\hat{\theta})$ penalizes the past errors from $\tau = 0$ to t that are due to $\hat{\theta}(t) \neq \theta$. Because $J(\hat{\theta})$ is a convex function over $\mathcal{R}^1$ at each time t, its minimum satisfies:

$$\nabla J(\hat{\theta}) = -\int_0^t y(\tau)u(\tau)d\tau + \hat{\theta}(t)\int_0^t u^2(\tau)d\tau = 0 \tag{D.0.4}$$

for any given time t, which gives:

$$\hat{\theta}(t) = \left(\int_0^t u^2(\tau)d\tau\right)^{-1} \int_0^t y(\tau)u(\tau)d\tau \tag{D.0.5}$$

provided the inverse exists. This is the *least-squares estimate*. Extending the problem to the linear model (D.0.1). The estimate of z and the normalized estimation error are generated as:

$$\begin{aligned}\hat{z} &= \hat{\theta}^T\phi \\ \epsilon &= \frac{z-\hat{z}}{m^2} = \frac{z-\hat{\theta}^T\phi}{m^2}\end{aligned} \tag{D.0.6}$$

where $m^2 = 1 + n_s^2$. m satisfies $\frac{\phi}{m} \in \mathcal{L}_\infty$. Consider the following cost function:

$$J(\hat{\theta}) = \frac{1}{2}\int_0^t e^{-\beta(t-\tau)}\frac{[z(\tau)-\hat{\theta}^T(t)\phi(\tau)]^2}{m^2(\tau)}d\tau + \frac{1}{2}e^{-\beta t}(\hat{\theta}-\hat{\theta}_0)^T Q_0(\hat{\theta}-\hat{\theta}_0) \tag{D.0.7}$$

where $Q_0 = Q_0^T > 0$, $\beta \geq 0$, $\hat{\theta}_0 = \hat{\theta}(0)$, which is a generalization of (D.0.3) to include discounting of past data and a penalty on the initial parameter estimate $\hat{\theta}_0$. Because $\frac{z}{m}$, $\frac{\phi}{m} \in \mathcal{L}_\infty$, $J(\hat{\theta})$ is a convex function of $\hat{\theta}$ over $\mathcal{R}^n$ at each time t. Any local minimum is also a global and satisfies:

$$\nabla J(\hat{\theta}(t)) = 0, \;\; \forall\, t \geq 0 \tag{D.0.8}$$

i.e.,

$$\nabla J(\hat{\theta}) = e^{-\beta t}Q_0(\hat{\theta}(t)-\hat{\theta}_0) - \int_0^t e^{-\beta(t-\tau)}\frac{z(\tau)-\hat{\theta}^T(t)\phi(\tau)}{m^2(\tau)}\phi(\tau)d\tau = 0$$

This yields the so-called *nonrecursive least-squares* algorithm:

$$\hat{\theta}(t) = \Gamma(t)\left[e^{-\beta t}Q_0\hat{\theta}_0 + \int_0^t e^{-\beta(t-\tau)}\frac{z(\tau)\phi(\tau)}{m^2(\tau)}d\tau\right] \tag{D.0.9}$$

where:

$$\Gamma(t) = \left[e^{-\beta t}Q_0 + \int_0^t e^{-\beta(t-\tau)}\frac{\phi(\tau)\phi(\tau)^T}{m^2(\tau)}d\tau\right]^{-1} \tag{D.0.10}$$

The parameter update law is found by taking the time derivative of $\hat{\theta}$:

$$\begin{aligned}\dot{\hat{\theta}}(t) = \Gamma(t)&\left[-\beta e^{-\beta t}Q_0\hat{\theta}_0 + \int_0^t -\beta e^{-\beta(t-\tau)}\frac{z(\tau)\phi(\tau)}{m^2(\tau)}d\tau + \frac{z(t)\phi(t)}{m(t)}\right] \\ &+ \dot{\Gamma}(t)\left[e^{-\beta t}Q_0\hat{\theta}_0 + \int_0^t e^{-\beta(t-\tau)}\frac{z(\tau)\phi(\tau)}{m^2(\tau)}d\tau\right]\end{aligned} \tag{D.0.11}$$

this is equal to:

$$\begin{aligned}\dot{\hat{\theta}} &= \Gamma(t)\left[-\beta\Gamma^{-1}(t)\hat{\theta}(t) + \frac{z(t)\phi(t)}{m^2(t)}\right] + \dot{\Gamma}(t)\Gamma^{-1}(t)\hat{\theta}(t) \\ &= -\beta\hat{\theta}(t) + \Gamma(t)\frac{z(t)\phi(t)}{m^2(t)} + \dot{\Gamma}(t)\Gamma^{-1}(t)\hat{\theta}(t)\end{aligned} \tag{D.0.12}$$

Because $Q_0 = Q_0^T > 0$ and $\phi\phi^T$ is positive semidefinite, $\Gamma(t)$ exists at each time t. Using the identity:

$$\frac{d}{dt}\Gamma\Gamma^{-1} = \dot{\Gamma}\Gamma^{-1} + \Gamma\frac{d}{dt}\Gamma^{-1} = 0$$

we achieve

$$\dot{\Gamma} = \beta\Gamma - \Gamma\frac{\phi(t)\phi^T(t)}{m^2}\Gamma\,, \;\; \Gamma(0) = Q_0^{-1} \tag{D.0.13}$$

Inserting this into (D.0.12) and using $\epsilon m^2 = z - \hat{\theta}^T\phi$ we get:

$$\begin{aligned} \dot{\hat{\theta}} &= -\Gamma(t)\epsilon\phi \\ \dot{\Gamma} &= \begin{cases} \beta\Gamma - \Gamma\frac{\phi(t)\phi^T(t)}{m^2}\Gamma & \text{if } ||\Gamma(t)|| \leq R_0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{D.0.14}$$

where $\Gamma(0) = \Gamma_0 = \Gamma_0^T > 0$, $||\Gamma_0|| \leq R_0$ and R_0 is a constant that serves as an upper bound for $||\Gamma||$. $m^2 = 1 + n_s^2$ and n_s is chosen such that $\frac{\phi}{m} \in \mathcal{L}_\infty$. With $\beta > 0$, $\Gamma(t)$ will not become arbitrarily small and because of the upper bound R_0 it will not become infinitely large. This modification guarantees that $\Gamma \in \mathcal{L}_\infty$. The stability properties of the least-squares with forgetting factor depends on the value of β. The properties are presented for $\beta > 0$:

Theorem D.1: *The modified least-squares with forgetting factor* (D.0.14) *has the following properties:*

1. ϵ, ϵn_s, $\hat{\theta}$, $\dot{\hat{\theta}} \in \mathcal{L}_\infty$
2. ϵ, ϵn_s, $\dot{\hat{\theta}} \in \mathcal{L}_2$

Proof. Consider the function

$$V(\tilde{\theta}) = \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \tag{D.0.15}$$

where Γ is given by (D.0.13). Because Γ^{-1} is a bounded positive definite symmetric matrix, it follows that V is decrescent and radially unbounded in the space of $\tilde{\theta}$. Along the solution of (D.0.13) we have

$$\dot{V} = \frac{1}{2}\tilde{\theta}^T\frac{d(\Gamma^{-1})}{dt}\tilde{\theta} + \tilde{\theta}^T\Gamma^{-1}\dot{\tilde{\theta}} = -\epsilon^2 m^2 + \frac{1}{2}\tilde{\theta}^T\frac{d(\Gamma^{-1})}{dt}\tilde{\theta}$$

By using the identity $\frac{d(\Gamma^{-1})}{dt} = -\Gamma^{-1}\dot{\Gamma}\Gamma^{-1}$ we can establish

$$\frac{d(\Gamma^{-1})}{dt} = \begin{cases} -\beta\Gamma^{-1} + \frac{\phi(t)\phi^T(t)}{m^2}\Gamma & \text{if } ||\Gamma(t)|| \leq R_0 \\ 0 & \text{otherwise} \end{cases}$$

where $\Gamma^{-1}(0) = \Gamma_0^{-1}$, which leads to:

$$\dot{V} = \begin{cases} -\frac{\epsilon^2 m^2}{2} - \frac{\beta}{2}\tilde{\theta}^T \Gamma^{-1} \tilde{\theta} & \text{if } ||\Gamma(t)|| \le R_0 \\ -\frac{\epsilon^2 m^2}{2} & \text{otherwise} \end{cases} \tag{D.0.16}$$

Because $\dot{V} \le -\frac{\epsilon^2 m^2}{2} \le 0$ and $\Gamma(t)$ is bounded and positive definite $\forall\, t \ge 0$, $V \in \mathcal{L}_\infty$ and ϵ, $\epsilon m \in \mathcal{L}_2$. From this we have $\tilde{\theta} \in \mathcal{L}_\infty$, which implies that ϵ, $\epsilon m \in \mathcal{L}_\infty$. Using $\epsilon m \in \mathcal{L}_\infty \wedge \mathcal{L}_2$ and that $\Gamma(t)$ is bounded we have $\dot{\hat{\theta}} \in \mathcal{L}_\infty \wedge \mathcal{L}_2$ □

Appendix E

Matlab code

```
%%%Simulation model for Merlin WR200%%%
clear all
clc
close all
%%
%Integration values
sim_time=600;                          %Simulation end time
dt=0.2;                                %Steplength
L=sim_time/dt;                         %Simulation steps
%%
%%%%%%%%%%%%%%%%      Dimensions and constants   %%%%%%%%%%%%%%%%%%%%%%%%%%
tau_thruster_max=1382;                  %Max forward thrust [N]
i=1;
m=3184; %3184                           %Mass of the ROV %3300
g=9.81;                                 %Gravitational acceleration
rho=1024;                               %Density of salt water
V=3.22;  %3.22                          %Volume of WR200
alpha=pi/4;                             %Angle of thrusters
l1=0.73;                                %
l2=0.24;                                %
l3=0.73;                                %
l4=0.84;                                %Length of moment arms
l5=0.1;                                 %
l6=0.84;                                %
r_g=[-0.002341 0.003014 -0.021193];     %Location of CG wrt CO
r_b=[0.12 0 0.197];                     %Location of CB wrt CO
I=[1805+0.05*m 0 -120;0 3050+0.05*m 7;-120 7 2872+0.05*m];   %Inertia matrix
B=rho*g*V;                              %Buoyancy
W=m*g;                                  %Weight
M_RB=1.1*m*eye(3);                      %Mass matrix including added mass
M=[M_RB*eye(3) -m*Smtrx(r_g);m*Smtrx(r_g) I];%System inertia matrix
M_inv=inv(M);
D=diag([1321 2525 2525 192 192 192]);   %Damping matrix
%%%%%%%%%%%%%%%%%%%%   Thruster allocation    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%
T=[zeros(1,4) sin(alpha) sin(alpha) -sin(alpha) -sin(alpha);zeros(1,4)...
    cos(alpha) -cos(alpha) -cos(alpha)...
    cos(alpha);ones(1,4) zeros(1,4);l2 -l2 -l2 l2...
    -l5 l5 -l5 l5;l1 l1 -l3 -l3...
    -l5*0*ones(1,4);zeros(1,4) -l4 l4 -l6 l6];
T_inv=pinv(T);
%%
%%%%%%%%%%%%%%%%%%% LQR settings   %%%%%%%%%%%%%%%%%%%%%%%%%%%
A6=[zeros(6,6) eye(6);zeros(6,6) -M_inv*D];
B6=[zeros(6,6);M_inv];

x_max=0.15; %Max deviation in X
y_max=0.15; %Max deviation in Y
z_max=0.15; %Max deviation in Z
psi_max=5*pi/180; %Max deviation in phi
theta_max=5*pi/180; %Max deviation in theta
phi_max=2*pi/180; %Max deviation in psi

Q6=diag([1/(x_max^2) 1/(y_max^2) 1/(z_max^2) 1/(phi_max^2)...
    1/(theta_max^2) 1/(psi_max^2) 10 10 10 10 10 10]);% Generate Q matrix

% Specify max allowable force in the different DOFs. Chosen rather
% conservative to not break the saturation:
tau_max_x=500; % Max allowable force in X
tau_max_y=500; % Max allowable force in Y
tau_max_z=1000; % Max allowable force in Z
tau_max_psi=100; % Max allowable force in K
tau_max_theta=100; % Max allowable force in M
tau_max_phi=500; % Max allowable force in N

r6=zeros(6,1);

r6(1)=1/(tau_max_x^2);
r6(2)=1/(tau_max_y^2);
r6(3)=1/(tau_max_z^2);
r6(4)=1/(tau_max_phi^2);
r6(5)=1/(tau_max_theta^2);
r6(6)=1/(tau_max_psi^2);

R6=diag(r6); %Make R matrix
[K6,P6,e6]=lqr(A6,B6,Q6,R6); %Generate K, R and eigenvalues
Kd_LQ6=diag(diag(K6(:,7:12))); %Pick out Kd

%%%%%%%%%%%%%%%%% Controller Gains %%%%%%%%%%%%%%%%%%%%%%%%%%%
Kd=diag([Kd_LQ6(1,1) Kd_LQ6(2,2) Kd_LQ6(3,3) 1 1 Kd_LQ6(6,6)]);
LAM=diag([0.1 0.389 0.2 20 1000 0.4]);
Gamma = diag([400 400 400 0]);

Cl=[zeros(6,6) eye(6);-(Kd*LAM)/M -((D/M)+(Kd/M))]; % Closed loop A matrix

%%
%Creating tables for data storage
nu=zeros(6,L);           %Velocity vector
e_2=zeros(6,L);          %Error in velocity
tau=zeros(6,L);          %Control forces and moments
eta=zeros(6,L);          %Position and attitude vector
eta_dot=zeros(6,L);      %Derivative of position(same as velocities)
e_1=zeros(6,L);          %Error in position and attitude
eta_corr=zeros(1,L);     %Vector to store corrections when passing 360 degrees
```

```
eta_corr2=zeros(1,L);
nu_dot=zeros(6,L);       %Derivative of velocities = acceleration
joy_x=zeros(1,L);        %Joystick input in X
joy_y=zeros(1,L);        %Joystick input in Y
joy_z=zeros(1,L);        %Joystick input in Z
joy_n=zeros(1,L);        %Joystick input in N
nu_ref=zeros(12,L);      %Reference velocity (filtered desired velocity)
eta_ref=zeros(18,L);     %Reference pos. and att.(filtered desired pos and att.)
g_eta=zeros(6,L);        %Restoring forces
tau_thruster=zeros(8,L); %Force to each thruster
u=zeros(8,L);            %Frequency to each thruster
u_pros=zeros(8,L);       %Percentage to each thruster
telapsed=zeros(1,L);     %Elapsed time
u_corrected=zeros(8,L);  %Percentage to each thruster, corrected to fit simulator
nu_v = zeros(6,L);       %Virtual velocity vector
eta_dot_p_d = zeros(6,L);%Eta dot paralell desired
eta_d_dot_p_d = zeros (6,L); %Eta double dot paralell desired
e_1_dot = zeros(6,L);    % Error variable
nu_v_dot = zeros(6,L);   % Virtual velocity vector derivative
theta = zeros(14,L);     % Parameter vector
theta_dot = zeros(14,L); % Derivative of parameter vector
tau_c_temp = zeros(4,L); % Temporary storage of current estimate
tau_c = zeros(6,L);      % Current force estimate
eta_d = zeros(6,L);      % Desired position
nu_d = zeros(6,L);       % Desired velocity
tau_pd = zeros(6,L);     % PD part of force vector
pausetid = zeros(1,L);   % Pause time to ensure equal steplengths
X_hat = zeros(2,L);      % Estimated X position
tau_ned = zeros(6,L);    % Control force in NED frame
Y_hat = zeros(2,L);      % Estimated Y position
yaw_hat = zeros(2,L);    % Estimate heading
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%      UDP connection   %%%%%%%%%%%%%%%%%%%%%%%%%

%Define UDP object
obj1 = instrfind('Type', 'udp', 'RemoteHost', '10.3.0.44',...
    'RemotePort', 11001, 'Tag', '');

if isempty(obj1)
    obj1 = udp('10.3.0.44', 11001, 'LocalPort',11000);
end

if strcmp(obj1.status,'closed') == true % open connection if closed
    fopen(obj1);
end

flushinput(obj1);           %Empty buffer input
flushoutput(obj1);          %Empty buffer output

u_send='0 0 0 0 0 0 0 0';   %Send zeros to all thrusters to initiate contact
tic                         %Start timer
sendt=0;mottatt=0;          %Set sent and recieved counter to zero
%%
%%%%%%%%%%%%%%%%%%%%%%%%  LOOP   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 while telapsed < 1000                  %Enter desired time to run in seconds
```

```
fwrite(obj1,u_send);             %Write to PLC to be able to recieve
sendt=sendt+1;                    %Count how many packages sent
data1 = fscanf(obj1);             %Reading values from PLC

if isempty(data1)==false
    mottatt=mottatt+1;            %Count how many packages recieved.
end
data2 = strsplit(data1);  %Split into separate words
x_c=cell2mat(data2(1,1)); %Convert from cell type to string(char)
eta(1,i)=str2double(x_c); %Convert from char to num
y_c=cell2mat(data2(1,2)); %Convert from cell type to string(char)
eta(2,i)=str2double(y_c); %Convert from char to num
z_c=cell2mat(data2(1,3)); %Convert from cell type to string(char)
eta(3,i)=str2double(z_c); %Convert from char to num
roll_c=cell2mat(data2(1,5)); %Convert fro cell type to string(char)
eta(4,i)=str2double(roll_c)*pi/180;  %Convert from char to num
pitch_c=cell2mat(data2(1,6));  %Convert from cell type to string(char)
eta(5,i)=str2double(pitch_c)*pi/180; %Convert from char to num
yaw_c=cell2mat(data2(1,4));    %Convert from cell type to string(char)
eta(6,i)=str2double(yaw_c)*pi/180;   %Convert from char to num

if i>1
    if (eta(6,i)-eta(6,i-1))>pi
        eta_corr(1,i)=eta(6,i)-2*pi;
        eta(6,i)=eta_corr(1,i);  %Discontinuity fix for heading
    elseif (eta(6,i)-eta(6,i-1))<-pi
        eta_corr2(1,i)=eta(6,i)+2*pi;
        eta(6,i)=eta_corr2(1,i);
    end
end
 joy_x_c=cell2mat(data2(1,7)); %Convert from cell type to string(char)
 joy_x(i)=str2double(joy_x_c);     %Joystick input in X in -1 to 1
 joy_y_c=cell2mat(data2(1,8)); %Convert from cell type to string(char)
 joy_y(i)=str2double(joy_y_c);     %Joystick input in Y in -1 to 1
 joy_n_c=cell2mat(data2(1,9));  %Convert from cell type to string(char)
 joy_n(i)=str2double(joy_n_c);     %Joystick input in N in -1 to 1
 joy_z_c=cell2mat(data2(1,10));  %Convert from cell type to string(char)
 joy_z(i)=str2double(joy_z_c);     %Joystick input in Z in -1 to 1

 flushinput(obj1);                     %Empty input buffer memory
 flushoutput(obj1);                    %Empty output buffer memory

 %%
 R_mat = [cos(eta(6,i)) sin(eta(6,i)) 0;
          -sin(eta(6,i)) cos(eta(6,i)) 0;  % Rotation matrix fix
          0 0 1];
 P_mat = [R_mat zeros(3,3);
          zeros(3,3) eye(3)];
 if i == 1
    nu(:,1)=0;
    nu_dot(:,1)=0;
    eta_dot(:,1) = 0;
 else
    eta_dot(:,i)=(eta(:,i)-eta(:,i-1))/dt;  % Update velocity
    nu(:,i)=P_mat\eta_dot(:,i);
 end
 eta_d(1:3,i) = eta(1:3,1);
```

```
% eta_d(1,300:end) = eta(1,299)+5;
eta_d(2,300:end) = eta(2,299)+2;
eta_d(4:6,i,:) = [0;0;eta(6,1)];%45*(pi/180)
% eta_d(6,400:end)=eta(6,399)-90*pi/180;
nu_d(:,i) = [joy_x(i)/100*4.3;joy_y(i)/100*3.2...
    ;joy_z(i)/100;0;0;joy_n(i)/100*0.5];%zeros(6,1);%nu(:,i);
%%
%%%%%%%%%%%%%%%%%%%%   REFERENCE MODEL   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
omega_ref_nu=diag([2 2 2
    1 1 2]); %Velocity reference model natural frequency
zeta_ref_nu=1;  %Velocity reference model damping. 1=no overshoot
omega_ref_eta=diag([0.3 0.3 1
    1 1 0.3]);%Pos and att reference model natural frequency
zeta_ref_eta=1;  %Pos and att reference model damping 1=no overshoot

%Velocity reference model
A_ref_nu=[zeros(6,6) eye(6);-omega_ref_nu^2*eye(6)
    -2*zeta_ref_nu*omega_ref_nu...
    *eye(6)];    %Reference model system matrix, continuous
B_ref_nu=[zeros(6,6);
    omega_ref_nu^2*eye(6)]; %Reference model input matrix, continuous

%Position and attitude reference model   Same steps as above
A_ref_eta=[zeros(6,6) eye(6) zeros(6,6);zeros(6,6) zeros(6,6) eye(6);...
    -omega_ref_eta^3*eye(6) -(2*zeta_ref_eta*eye(6)+eye(6))...
    *omega_ref_eta^2 -(2*zeta_ref_eta*eye(6)+eye(6))*omega_ref_eta];
B_ref_eta=[zeros(6,6);zeros(6,6);omega_ref_eta^3*eye(6)];

eta_ref_0=[eta(:,1);zeros(12,1)];
nu_ref_0=[nu(:,1);zeros(6,1)];

%Estimator
A_X_hat =[0 1;0 -1/3.5];
B_X_hat =[0;0.0005/3.5];
X_hat_0 = [0 0]';

A_Y_hat = [0 1;0 -1/2.1];
B_Y_hat = [0;0.00035/2.1];
Y_hat_0 = [0 0]';

A_yaw_hat = [0 1;0 -1/2.2];
B_yaw_hat = [0;0.000375/2.2];
Z_yaw_0 = [0 0]';

if i <= 16
    X_hat(:,i) = (eye(2)+dt.*A_X_hat)*X_hat(:,i) + dt*B_X_hat*tau(1,i);
else
    X_hat(:,i) = (eye(2)+dt.*A_X_hat)*X_hat(:,i-1)...
        + dt*B_X_hat*tau(1,i-16);
end

if i<=30
    Y_hat(:,i) = (eye(2)+dt.*A_Y_hat)*Y_hat(:,i) + dt*B_Y_hat*tau(2,i);
else
    Y_hat(:,i) = (eye(2)+dt.*A_Y_hat)*Y_hat(:,i-1) +...
        dt*B_Y_hat*tau(2,i-30);
end
```

```
    if i<=40
        yaw_hat(:,i) = (eye(2)+dt.*A_yaw_hat)*yaw_hat(:,i) +...
            dt*B_yaw_hat*tau(6,i);
    else
        yaw_hat(:,i) = (eye(2)+dt.*A_yaw_hat)*yaw_hat(:,i-1) +...
            dt*B_yaw_hat*tau(6,i-40);
    end
    if i==1 % First iteration, use initial state as nu_ref_0 and eta_ref_0
        nu_ref(:,i)=(eye(12)+dt.*A_ref_nu)*nu_ref_0+dt*B_ref_nu*nu_d(1:6,i);
%Velocity
        eta_ref(:,i)=(eye(18)+dt.*A_ref_eta)*eta_ref_0+dt*B_ref_eta*eta_d(1:6,i); %Pos

    else    % Then feed the reference signal through the filter
        nu_ref(:,i)=(eye(12)+dt.*A_ref_nu)*nu_ref(:,i-1)+dt*B_ref_nu*nu_d(1:6,i);%Velo
        eta_ref(:,i)=(eye(18)+dt.*A_ref_eta)*eta_ref(:,i-1)+dt*B_ref_eta*eta_d(1:6,i);
    end
    %%
    %%%%%%%%%%%%%%%%%      CONTROL LOOP   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    g_eta(:,i)=[0 0 0 -4820*cos(eta(5,i))*sin(eta(4,i))...          %
            -4820*sin(eta(5,i))*cos(eta(5,i))*cos(eta(4,i)) 0];%Restoring
    e_1(:,i) = (eta(:,i)-eta_ref(1:6,i));
    if e_1(6,i)<=-pi             %Discontinuity fix for heading
        e_1(6,i)=2*pi-e_1(6,i);
    elseif e_1(6,i)>pi
        e_1(6,i)=-2*pi+e_1(6,i);
    end

    eta_dot_p_d(:,i) = 0.8*eta_ref(7:12,i);
    eta_d_dot_p_d(:,i) = 0.1*eta_ref(13:18,i);
    e_1_dot(:,i) = nu(:,i) - eta_dot_p_d(:,i);
    nu_v(:,i) = -LAM*e_1(:,i)+eta_dot_p_d(:,i);
    nu_v_dot(:,i) = -LAM*e_1_dot(:,i)+eta_d_dot_p_d(:,i);
    e_2(:,i)=eta_dot(:,i)-nu_v(1:6,i);

    %%
    %%%%%%%%%%%%%%%%%%%%%%% Generate regressor %%%%%%%%%%%%%%%%%%%%%%%%%%

    phi = [nu_v_dot(1,i) 0 0 0 -abs(nu(1,i))*nu(1,i) 0 ...
        0 0 0 0 0 cos(eta(6,i)) -sin(eta(6,i)) 0;
           0 nu_v_dot(2,i) 0 0 0 -abs(nu(2,i))*nu(2,i) 0
           0 0 0 0 sin(eta(6,i)) cos(eta(6,i)) 0;
           0 0 nu_v_dot(3,i) 0 0 0 -abs(nu(3,i))*nu(3,i)
           0 0 0 0 0 0 1;
           0 0 0 nu_v_dot(6,i) 0 0 0 -abs(nu(6,i))*nu(6,i)...
           0 0 0 0 0 0];
    %%

    %%%%%%%%%%%%%%%%%%%%%%%%   Adaptation    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    tau_c_temp(:,i) = phi*theta(:,i);
    tau_c(:,i) = [tau_c_temp(1,i) tau_c_temp(2,i)
        theta(14,i) 0 0 tau_c_temp(4,i)]';

    %     tau_c(:,i) = [0 0 theta(14,i) 0 0 0]';
    theta_dot(:,i) = phi'*Gamma*e_2([1 2 3 6],i);
    theta(:,i+1) = theta(:,i)+dt*theta_dot(:,i);
    %%%%%%%%%%%%%%%%%%%   Set output   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    tau_pd(:,i) = -e_1(:,i) - Kd*e_2(:,i);
    tau_ned(:,i) = tau_pd(:,i)-tau_c(:,i);
    tau(:,i) = P_mat\tau_ned(:,i);%+g_eta(:,i);

    tau_thruster(:,i)=pinv(T)...
        *tau(:,i); %Desired force from each thruster in [N] and [Nm]
                   %Desired force is unlimited, but the actual force
                   %must be limited
    if i>1
        for s=1:8              %Check to see if thruster is at limit
            if abs(tau_thruster(s,i))>=tau_thruster_max
                tau_thruster(s,i)=(tau_thruster(s,i)/...
                    abs(tau_thruster(s,i)))*tau_thruster_max;%Limit thrust
            end
        end
    end

    u(:,i)=83*sin(0.0007909*tau_thruster(:,i));

    u_corrected(1,i)=u(6,i);
    u_corrected(2,i)=u(5,i);
    u_corrected(3,i)=u(7,i);
    u_corrected(4,i)=u(8,i);%Changes numbering of thrusters to fit simltr
    u_corrected(5,i)=u(2,i);
    u_corrected(6,i)=u(1,i);
    u_corrected(7,i)=u(3,i);
    u_corrected(8,i)=u(4,i);

    u_pros(:,i)=(u_corrected(:,i)/100)*100;  %Convert to percent

    u_send=sprintf('%f ',u_pros(:,i));%Stores the % in u_send as a string.

    telapsed(i)=toc;      %Updates elapsed time
%%
%%%%%%%%%%%%%%%%%%%%%%%%%  REAL TIME PLOTS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%    plot(telapsed(1:i),joy_x(1,1:i));hold on;
%    plot(telapsed(1:i),nu(1,1:i),'r');
%    plot(telapsed(1,1:i),joy_x(1:i),'b');hold on;
% plot(telapsed(1,1:i),eta_ref(1,1:i),'r');hold on;

figure(1)
subplot(3,2,1)
plot(telapsed(1,1:i),eta(1,1:i)),title('N');hold on;
plot(telapsed(1,1:i),eta_ref(1,1:i));
subplot(3,2,3)
plot(telapsed(1,1:i),eta(2,1:i)),title('E');hold on;
plot(telapsed(1,1:i),eta_ref(2,1:i));
subplot(3,2,5)
plot(telapsed(1,1:i),eta(3,1:i)),title('D');hold on;
plot(telapsed(1,1:i),eta_ref(3,1:i));
subplot(3,2,2)
plot(telapsed(1,1:i),eta(4,1:i)),title('\phi');hold on;
plot(telapsed(1,1:i),eta_ref(4,1:i));
subplot(3,2,4)
plot(telapsed(1,1:i),eta(5,1:i)),title('\theta');hold on;
plot(telapsed(1,1:i),eta_ref(5,1:i));
subplot(3,2,6)
plot(telapsed(1,1:i),eta(6,1:i)*180/pi),title('\psi');hold on;
plot(telapsed(1,1:i),eta_ref(6,1:i)*180/pi);

figure(2)
subplot(3,2,1)
plot(telapsed(1,1:i),nu(1,1:i)),title('u');hold on
plot(telapsed(1,1:i),X_hat(2,1:i));
subplot(3,2,3)
plot(telapsed(1,1:i),nu(2,1:i)),title('v');hold on
plot(telapsed(1,1:i),Y_hat(2,1:i));
subplot(3,2,5)
plot(telapsed(1,1:i),nu(3,1:i)),title('w');
subplot(3,2,2)
plot(telapsed(1,1:i),nu(4,1:i)),title('p');
subplot(3,2,4)
plot(telapsed(1,1:i),nu(5,1:i)),title('q');
subplot(3,2,6)
plot(telapsed(1,1:i),nu(6,1:i)),title('r');hold on
plot(telapsed(1,1:i),yaw_hat(2,1:i));

   drawnow
   pausetid(i)=dt-(toc-telapsed(i));
   pause(pausetid)                          %Pauses for sampling time

   i=i+1;                      %Update count

 end
%%
```