



Norwegian University of
Science and Technology

Autonomous Homing and Docking of AUV REMUS 100

Homing and Docking Guidance Algorithm and
Relative Localization

Fredrik Jonsson Ruud

Master of Science in Engineering and ICT

Submission date: June 2016

Supervisor: Martin Ludvigsen, IMT

Co-supervisor: Petter Norgren, IMT

Norwegian University of Science and Technology
Department of Marine Technology



NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER'S THESIS IN MARINE CYBERNETICS

SPRING 2016

for

STUD. TECH. FREDRIK JONSSON RUUD

Autonomous Homing and Docking of AUV REMUS 100
Homing and Docking Guidance Algorithm and Relative Localization

Work Description

To prolong Autonomous Underwater Vehicle (AUV) missions and reduce human intervention before and after missions, an autonomous docking system is favourable. This would not only reduce the operational costs but also possibly strengthen the market position of the operator company.

The foundation of an autonomous homing and docking procedure is a working homing and docking algorithm. Such an algorithm would ease the retrieval of the AUV and facilitate for future underwater docking missions. The homing phase of the algorithm steers the AUV safely close to the docking station where the docking phase takes over until the AUV is successfully docked. After docking, communication can be established and batteries charged, while the AUV is securely held in place. For this to happen, the algorithm needs to be complex and address guidance, navigation and control systems.

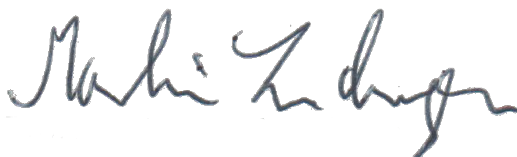
The scope of this thesis is to develop and simulate a homing and docking algorithm for NTNU's AUV, REMUS 100, developed by Hydroid. Two different underwater acoustic systems are compared; namely the Ultra-short Baseline (DUSBL) and Long Baseline (LBL) acoustic systems.

Scope of work

1. Algorithm development
 - a. Create a homing and docking algorithm that uses **DUSBL** measurements to home and dock the REMUS 100 AUV.
 - b. Create a homing and docking algorithm that uses **LBL** measurements from a single transponder to home and dock the REMUS 100 AUV.
2. Design a particle filter (PF) that can estimate the relative AUV position from LBL measurements (range-only) from a single transponder.
3. Simulation
 - a. Simulate homing and docking with the DUSBL sensor.
 - b. Use the PF-estimated position in the LBL homing and docking algorithm and simulate homing and docking.
4. Conduct a mission with the REMUS 100 to gather LBL sensor data that can be used to verify the performance of the particle filter and the homing and docking algorithm.

The report shall be written in English and edited as a research report including literature survey, description of mathematical models, description of control algorithms, simulation results, model test results, discussion and a conclusion, including a proposal for further work. Source code should be provided on a CD, memory stick or digitally on DAIM. It is supposed that the Department of Marine Technology, NTNU, can use the results freely in its research work, unless otherwise agreed upon, by referring to the student's work. The thesis should be submitted in three copies within June 10.

Advisers: Ph.D. Candidate Petter Norgren



Professor Martin Ludvigsen
Supervisor

Preface

This master's thesis is a part of the study program "Engineering and ICT" at the Faculty of Engineering Science and Technology, with specialization within Marine Cybernetics offered by the Department of Marine Technology. The work was carried out in the spring semester of 2016, between January 15 and June 10. This thesis is directed to all readers who investigate the possibilities of making the Autonomous Underwater Vehicle (AUV) Remote Environmental Measuring UnitS (REMUS) 100 dock autonomously.

Trondheim, June 10, 2016

A handwritten signature in black ink, reading "Fredrik Jonsson Ruud". The signature is written in a cursive style with a large initial 'F'.

Fredrik Jonsson Ruud

Abstract

This master's thesis presents an algorithm for homing and docking of NTNU's AUV, REMUS 100. Such an algorithm would ease the retrieval of the AUV and facilitate for future underwater docking missions. The homing phase of the algorithm steers the AUV safely close to the docking station, where the docking phase takes over until the AUV is successfully docked. After docking, communication can be established and batteries charged, while the AUV is securely held in place. For this to happen, the algorithm needs to be complex and address guidance, navigation and control systems.

The Digital Ultra-short Baseline (DUSBL) and Long Baseline (LBL) acoustic positioning systems are both considered for the homing and docking task. Only one transponder is used in the acoustic system, namely the one mounted on the docking station. This motivated to create a particle filter for position estimation from range-only LBL measurements.

The REMUS 100 AUV simulator, AUVSim, was used to make all simulations. AUVSim was extended with a homing/docking block that could simulate LBL/-DUSBL measurements, as well as the homing and docking algorithm.

Results from simulations show that the DUSBL sensor with the proposed algorithm is suitable for autonomous homing and docking. Simulations also showed that a particle filter which used range-only data from the LBL sensor was able to dock successfully. A mission conducted on Trondheimsfjorden June 2, 2016, made clear that the acoustic ping rate and LBL settings need modifications before LBL docking is possible in real life. The ping rates were too low and the LBL sensor did not get any measurements close to the transponder. LBL homing was still possible in AUVSim when adjusting the simulations to the real ping rates.

Sammendrag

Denne masteroppgaven presenterer en algoritme for homing og docking av NTNU sin AUV, REMUS 100. En slik algoritme vil gjøre det vesentlig lettere å lokalisere og hente REMUS 100 etter et oppdrag. Det vil også åpne opp for fremtidige oppdrag innenfor docking under vann. Homing-fasen i algoritmen styrer REMUS 100 nærmere dockingstasjonen, hvor docking-fasen tar over helt til REMUS 100 er plassert i selve dockingstasjonen. Etter dette kan kommunikasjon bli opprettet og batteriene ladet, mens REMUS 100 blir trygt holdt på plass. For at dette skal være mulig må algoritmen være avansert og bruke teori innenfor styring, navigasjon og kontroll.

Digital Ultra-short Baseline (DUSBL) og Long Baseline (LBL) er de to akustiske posisjonssystemene som blir brukt og sammenlignet i homing og docking situasjoner. Bare én transponder blir brukt i systemet, hvor denne er montert på dockingstasjonen. Med bakgrunn i dette ble det designet et partikkelfilter som kan estimere posisjon fra lengdemålingene fra LBL sensoren på REMUS 100.

AUV-simulatoren, AUVSim, simulerer REMUS 100 og ble brukt til alle simuleringene. AUVSim ble utvidet til å inneholde en homing/docking blokk som kunne simulere LBL og DUSBL målinger. I denne blokken ble også homing og docking algoritmen implementert.

Resultatene fra simuleringene viser at DUSBL sensoren sammen med den foreslåtte algoritmen er passende for både homing og docking. Simuleringene viser også at partikkelfilteret kombinert med lengdedata fra LBL-sensoren klarte docking. Et tokt med REMUS 100 på Trondheimsfjorden den 2. juni, 2016, gjorde det klart at den akustiske svartiden og innstillingene av LBL-sensoren må justeres for at docking skal være mulig i praksis. Raten på LBL målingene var for lav og området i nærheten av transponderen fikk ingen målinger. LBL homing ga gode resultater i AUVSim da de nye pingratene ble brukt.

Aknowledgements

I want to thank supervisor Professor Martin Ludvigsen, co-supervisor and Ph.D. candidate Petter Norgren, and Ph.D. candidate Albert Sans Muntadas for their help during my work with this thesis.

Petter Norgren has devoted much time and patience in learning me the details and inner workings of AUVSim and REMUS 100. He always had the time to answer my questions and pointed me the right direction when needed. I am immensely grateful to have had the opportunity to have Petter as a co-supervisor. Albert enthusiastically shared his thoughts about the direction my work was heading and Professor Martin Ludvigsen made sure that I always was on the right track.

F.J.R.

Nomenclature

<i>AUV</i>	Autonomous Underwater Vehicle
<i>BODY</i>	Body-fixed reference frame
<i>DOF</i>	Degrees of Freedom
<i>DUSBL</i>	Digital Ultra-Short Baseline acoustic positioning system
<i>DVL</i>	Doppler Velocity Log
<i>GPF</i>	Generic Particle Filter
<i>GPS</i>	Global Positioning System
<i>IMU</i>	Inertial Measurement Unit
<i>LBL</i>	Long Baseline acoustic positioning system
<i>LOS</i>	Line-of-Sight guidance
<i>NED</i>	North-East-Down reference frame
<i>PDF</i>	Probability Density Function
<i>PF</i>	Particle Filter
<i>PID</i>	Proportional-Integral-Derivative
<i>RECON</i>	REMUS REmote CONtrol Protocol
<i>REMUS</i>	Remote Environmental Monitoring UnitS
<i>ROV</i>	Remotely Operated Vehicle
<i>SSP</i>	Sound Speed Profile
<i>SSS</i>	Side Scan Sonar
<i>UUV</i>	Unmanned Underwater Vehicle
<i>WP</i>	Waypoint

Table of contents

List of Figures	xvi
List of Tables	xx
1 Introduction	1
1.1 Motivation	1
1.2 Scope and Limitations	1
1.3 Objectives	2
1.4 Background	3
1.4.1 Unmanned Underwater Vehicles	3
1.4.2 The REMUS 100 AUV	3
1.4.3 DUSBL and LBL Underwater Navigational Systems	4
1.4.4 AUVSIM	4
1.5 Contributions	4
1.6 Organization of the Thesis	6
2 AUVs, Modelling and Theory	7
2.1 Autonomous Underwater Vehicles	7
2.1.1 Development of AUVs	7
2.1.2 Sensor Systems and their Applications	10
2.2 Hardware and Software on the REMUS 100	17
2.2.1 Specifications	17
2.2.2 The Payload processor and the REMUS computer	18
2.2.3 HUGIN Software Development Kit	18
2.2.4 REMUS Remote Control Interface	19
2.3 Mathematical Modelling of AUVs	21
2.3.1 Kinematics	21
2.3.2 Kinetics	24
2.3.3 Simulation, Control and Observer Design Models	25
2.4 Line-of-Sight Guidance	27
2.5 Particle Filtering	29

2.5.1	The Generic Particle Filter	29
2.6	Previous Work on Homing and Docking	36
2.6.1	Homing and Docking Attempts	36
2.6.2	Underwater Navigation with Particle Filters	39
3	Methods	41
3.1	Homing and Docking Algorithm	42
3.1.1	DUSBL aided Homing and Docking	42
3.1.2	LBL aided Homing and Docking	50
3.2	Simulator Environment	54
3.2.1	Simulator Overview	54
3.2.2	Measurement Noise and Filtering	56
3.2.3	Internal Remus 100 Controllers	60
3.2.4	Tuning the Particle Filter	61
3.3	LBL Test in Trondheimsfjorden	62
4	Results	65
4.1	DUSBL Homing and Docking	65
4.2	LBL Homing and Docking	69
4.2.1	Unfiltered and Filtered Measurements	69
4.2.2	Multinomial Resampling	71
4.2.3	Residual Resampling	74
4.2.4	Systematic Resampling	76
4.3	Depth Keeping	78
4.4	Results from the LBL Test	79
4.4.1	Line	79
4.4.2	Box #1	81
4.4.3	Box #2	83
4.4.4	Side Scan Sonar Data	85
5	Discussion	87
6	Conclusions and Further Work	91
6.1	Further Work	92
	References	93
A	Reduced-order AUV Models	99
A.1	Longitudinal subsystem:	99
A.2	Lateral subsystem:	99
A.3	Systems Simplifications	100

TABLE OF CONTENTS

xv

B Particle Filter Results	101
C Additional Figures	105
D Attachments	109

List of Figures

1.1	Hydroid REMUS 100 AUV (Kongsberg Maritime, 2015).	3
2.1	The SPURV AUV on a launching crane (NavalDrones.com, 2015). . .	8
2.2	The open frame EAVE AUV (DeepOceanAndDeepSpace.com, 2015). . .	9
2.3	One of the most popular ADCPs on the market is the "Workhorse Monitor ADCP" from Teledyne RD Instruments (Teledyne RD Instruments, 2015).	11
2.4	Wet Labs ECO Triplet Puck (WetLabs, 2015).	13
2.5	LBL, SBL and DUSBL (Kongsberg Maritime, 2015).	15
2.6	Flow chart of the interaction between the Payload Processor computer and REMUS computer (Holsen, 2015).	18
2.7	The relationship between the NED and BODY reference frames (Fossen, 2011).	21
2.8	The six degrees of freedom of a moving ship (Fossen, 2011).	22
2.9	LOS guidance. Course angle χ_d points towards the LOS intersection point (x_{los}, y_{los}) (Fossen, 2011).	27
2.10	Particle resampling with ten particles (Doucet, de Freitas, & Gordon, 2001).	33
2.11	Computational time with different sized particle sets (Hol, Schon, & Gustafsson, 2006).	35
2.12	An AUV docked in a cone docking station (Stokey et al., 1997). . .	37
2.13	An AUV latched to a pole docking station (Singh et al., 2001). . . .	38
3.1	AUV homing towards docking station.	45
3.2	Waypoints are generated and LOS navigation initiated when the cross-track error is too large.	46
3.3	Initial AUV cross-track error, y_{err}	47
3.4	AUV homing control loop.	48
3.5	AUV in docking phase. The control goal is to minimize y_{err}	49
3.6	AUV docking control loop.	50
3.7	AUV homing to WP1 followed by a docking sequence.	52

3.8	Overview of main AUVSim Simulink blocks, included the homing/docking block.	55
3.9	Inside the blue homing/docking block (blue) in Figure 3.8.	55
3.10	Filtered and unfiltered DUSBL measurements.	57
3.11	Detailed view of the filtered and unfiltered DUSBL measurements.	58
3.12	Calculated dock position from DUSBL measurements.	59
3.13	Cross-track error from dock center line calculated from DUSBL measurements.	60
3.14	tl. REMUS 100 and computers with VIP software; tr. REMUS 100 in the water; bl. boat in transit; br. gear on trolley.	62
3.15	The plus symbol indicates the transponder position after box-in.	63
4.1	AUV position during DUSBL homing and docking.	66
4.2	The real cross-track error during DUSBL homing and docking.	67
4.3	AUV trajectory during repositioning and docking	68
4.4	The real cross-track error during DUSBL homing and docking when repositioning is needed.	69
4.5	Estimation of the position with the PF from unfiltered LBL measurements.	70
4.6	Estimation of the position with the PF from filtered LBL measurements.	71
4.7	The estimated position overlays the actual position well.	72
4.8	The error between the real position and the estimated position by the PF.	73
4.9	The real cross-track error during homing/docking with multinomial resampling.	73
4.10	The estimated position overlays the actual position well.	74
4.11	The error between the real position and the estimated position by the PF.	75
4.12	The real cross-track error during homing/docking with systematic resampling.	75
4.13	The estimated position overlays the actual position very well.	76
4.14	The error between the real position and the estimated position by the PF.	77
4.15	The real cross-track error during homing/docking with systematic resampling.	77
4.16	Depth error during homing and docking with DUSBL.	78
4.17	Depth error during homing and docking with LBL.	79
4.18	Results when traveling over the transponder.	80
4.19	Results when traveling in a 100x100 [m] box.	82
4.20	Results when traveling in a 200x200 [m] box.	84

4.21 The SSS show that the AUV is close to the transponder. 85

5.1 PF with LBL measurements every three seconds. 89

B.1 LBL homing/docking with 100 particles. 101

B.2 LBL homing/docking with 700 particles. 102

B.3 LBL homing/docking with 1000 particles. 102

B.4 LBL homing/docking with 1200 particles. 103

B.5 LBL homing/docking with 1500 particles. 103

C.1 Large diagram of the homing/docking Simulink block. 1/3 106

C.2 Large diagram of the homing/docking Simulink block. 2/3 107

C.3 Large diagram of the homing/docking Simulink block. 3/3 108

List of Tables

2.1	Overview of acoustic frequency bands, frequency ranges and maximum distances (Vickery, 1998).	15
2.2	AUV REMUS 100 general specifications (Hydroid, 2012).	17
2.3	AUV REMUS 100 sensor systems (Hydroid, 2012).	17
2.4	SNAME (1950) notation for marine vessels.	22
3.1	REMUS 100 heading controller PID gains.	60
3.2	REMUS 100 depth controller PI gains.	60
3.3	REMUS 100 speed controller PI gains.	61
4.1	DUSBL docking controller PID gains.	65
4.2	Gains for the LOS controller.	67

Chapter 1

Introduction

1.1 Motivation

As of today, AUV missions is highly dependent on human action in the pre-mission and post-mission stages. The AUV is programmed, launched and recovered, the batteries are charged (or replaced) and then the whole process repeats until the survey area is covered. Due to limited battery capacity, an operator with a recovery vessel needs to be on site ready to recover the AUV. If the AUV featured the possibility of autonomous docking, it could automatically recharge and get new mission objectives from an underwater docking station when needed. Thus remove the need for human intervention other than periodical service. This way missions could extend significantly in time, be more effective and the AUV could operate in harsher conditions since there is no longer a need for exposure in the wave affected zone in launching/retrieval scenarios. All of this contribute to cost savings and to strengthen the market position of the operator company.

1.2 Scope and Limitations

The aim of this thesis is to find a way to home NTNU's AUV REMUS 100 close to a docking station, followed by docking into the dock itself. This should be done with the help of either a DUSBL or an LBL sensor on the REMUS 100. The thesis revolves around the REMUS 100 and the results might not be applicable to other AUVs. That being sad, AUVs with similar dynamics as the REMUS 100 is expected to behave similarly. This can be verified by modifying the hydrodynamic parameters in the simulator, but is not within the scope of

this thesis. All results are from simulations and can therefore never completely resemble the reality. Promising simulation results mean that real life sea trials should be conducted to verify the algorithm. Sea trials of the homing and docking algorithm was not carried out during the work with the thesis. Nevertheless, a mission in Trondheimsfjorden was conducted to gather data from the LBL sensor.

Several types of AUV docking stations exist. The geometry and design of the docking station is not discussed in this thesis, but it is assumed that the docking station has only one entry point. It is the results from the homing/docking algorithm and controllers that govern which kind of docking station that should be used. For a cone dock with a "trumpet-shaped" entry, a reasonable entry diameter would be around 1 meter for REMUS 100. This estimate is based on the fact that the AUV in Bellingham, McEwen, Hobson, and McBride (2008) uses an entry diameter that is 4 times the diameter of the AUV. With a 1 meter diameter on the dock, this would be over 5 times the REMUS 100 diameter (19 cm).

1.3 Objectives

The following are the main objectives. The details of each objective are investigated closer throughout the thesis when the necessary background material and theory have been explained to the reader.

- Create a homing and docking algorithm that uses DUSBL measurements to home and dock the REMUS 100.
- Simulate homing and docking with the DUSBL sensor.
- Create a particle filter (PF) that can estimate the relative AUV position from LBL measurements (range-only) from a single transponder.
- Use the PF-estimated position in a homing and docking algorithm and simulate homing and docking.
- Conduct a mission with the REMUS 100 to gather LBL sensor data that can be used to verify the real life performance of the particle filter and the homing and docking algorithm.

1.4 Background

1.4.1 Unmanned Underwater Vehicles

Unmanned Underwater Vehicles (UUVs) are often divided into two categories: Remotely Operated Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs). An ROV has an umbilical cable which supplies the ROV with power and control signals. In most cases ROVs are controlled manually from a control room. AUVs are untethered and need an internal power supply. Also, due to the limitations of underwater communication, AUVs need to autonomously navigate and perform missions.

1.4.2 The REMUS 100 AUV

The REMUS 100 (Figure 1.1) is an AUV developed by Woods Hole Oceanographic Institution (WHOI) as a result of years of research and development. To ensure continuous product development of the REMUS family, the inventors of the REMUS 100 founded Hydroid in 2001 and was later bought by Kongsberg Maritime, in 2008.

Propulsion is done by a single propeller and steering by two stern planes (horizontal fins) and two rudder planes (vertical fins), controlling pitch and heading, respectively. The two fins in each plane move in unison and can not be subjected to individual control.



Figure 1.1: Hydroid REMUS 100 AUV (Kongsberg Maritime, 2015).

The REMUS 100 has multiple sensors for seabed mapping, biochemical measurements of surrounding water and for underwater navigation. The most important sensors used in REMUS 100 are described in Chapter 2.1.2 and the full specification list of NTNU's REMUS 100 is given in Chapter 2.2.1.

1.4.3 DUSBL and LBL Underwater Navigational Systems

DUSBL is a digital acoustic underwater navigation system that makes it possible to determine the position relative to a single underwater *transponder*. Details around this navigation system is found in Chapter 2.1.2. NTNU has not yet tried DUSBL navigation on REMUS 100, mainly due to some interfacing problems with the sensor and onboard computer. Currently, experiments are done by the Institute of Marine Technology, NTNU, to figure out the sensor characteristics and how it should integrate with the REMUS 100.

LBL is another underwater navigation system that finds the AUV position with trilateration of range measurements from transponders placed on the seabed. Preferably four transponders should be installed in a square and the AUV would operate inside the area of the square. LBL navigation works with a minimum of two transponders. The obvious problem with single transponder LBL navigation is that the possible AUV positions are represented by all points laying on the sphere surrounding the transponder. A particle filter is made to solve this problem.

1.4.4 AUVSim

Ph.D. candidate Petter Norgren created the REMUS 100 simulator, AUVSim, to test an iceberg edge following algorithm (Norgren & Skjetne, 2015). It is implemented in MATLAB and Simulink and simulates the dynamics of the REMUS 100 AUV. All simulations are done in AUVSim.

1.5 Contributions

The contributions of this thesis are the following:

- A homing and docking algorithm for the REMUS 100 has been developed, which is able to do homing and docking to a dock with a single transponder by the help of a **DUSBL** sensor. Section 3.1.1 presents the algorithm in details.
- A homing and docking algorithm for the REMUS 100 has been developed, which is able to do homing and docking to a dock with a single transponder by the help of a **LBL** sensor. Section 3.1.2 presents the algorithm in details.

- AUVSim has been extended to feature a single-transponder DUSBL/LBL simulator block. The block is explained in Chapter 3.2.
- AUVSim now has the possibility to simulate homing and docking
- A range-only PF for position estimation has been designed for the REMUS 100. The necessary theory is explained in Chapter 2.5.

1.6 Organization of the Thesis

Chapter 2 gives an introduction to AUVs, sensory systems and applications. The chapter presents a mathematical model that satisfactorily describes the kinematics and kinetics of an AUV, as well as line-of-sight guidance and particle filter theory.

Chapter 3 presents the details of the methods which were used. The main elements in this chapter are the homing and docking algorithm, simulator environment and the LBL test.

Chapter 4 presents the results from the simulations and the LBL test.

Chapter 5 discusses the most important findings from the results chapter.

Chapter 6 concludes the thesis and suggests further work.

Appendix A gives the equations of a decoupled, reduced-order, simplified, mathematical AUV model.

Appendix B presents particle filter results when using different numbers of particles.

Appendix C contains additional figures that are too large to be inserted directly into the previous chapters.

Appendix D lists the attachments that are delivered electronically in DAIM.

Chapter 2

AUVs, Modelling and Theory

This chapter provides an introduction to AUV systems and gives the necessary theoretical background for the methods chapter. Section 2.1 is intended to give an introduction to AUVs and sensory systems, while Section 2.2 describes the systems on REMUS 100. Section 2.3 presents the theory and mathematics behind mathematical modelling of an AUV. Section 2.4 presents Line-of-Sight Guidance, which play an integral part in the homing and docking algorithm. The particle filter used in the LBL homing and docking algorithm is described in Section 2.5. The chapter ends with a study of previous work on homing and docking, in Section 2.6.

2.1 Autonomous Underwater Vehicles

Some insight in the field of AUVs and mathematical modelling is necessary before moving on to the next chapters, especially Section 3.1 that discusses homing and docking.

2.1.1 Development of AUVs

In the last 40 years the development and research of AUV technology have gone through several phases, leading up the point today where we find fully commercialized AUV systems. Just like most technologies it started out with academic research and investigation.

From the late 1950s to the late 1970s, the Applied Physics Laboratory (APL) at The University of Washington developed the Special Purpose Underwater

Research Vehicle (SPURV) and the Unmanned Arctic Research Vehicle (UARS) testbeds. The SPURV (Figure 2.1) was built in 1957 and used mainly for oceanographic research, measuring temperature, conductivity, acoustic transmission and by the U.S. Navy in the study of submarine wakes (NavalDrones.com, 2015). Further work sponsored by The Advanced Research Projects Agency's (ARPA) arctic technology program resulted in the UARS, which was the first reported AUV deployed in the Arctic ocean (Francois & Nodland, 1972). Later, the UARS was used to make high resolution observations of under-ice morphology with the help of a multi-beam, upward-looking acoustic lens. An acoustic tracking system was used to determine the position to within 15 centimeters of a reference baseline (Francois, 1977).



Figure 2.1: The SPURV AUV on a launching crane (NavalDrones.com, 2015).

Concurrently, other testbeds were developed by academic communities all over the world as well. An open space-frame AUV named the Experimental Autonomous Vehicle (EAVE EAST) (Figure 2.2) was developed by the University of New Hampshire's Marine Systems Engineering Laboratory (now the Autonomous Undersea Systems Institute (AUSI)) and EAVE WEST at the Naval Oceans System Center in San Diego. They demonstrated pipe following capabilities and later the possibility of underwater docking (Brutzman, 1994). The Russian Institute of Marine Technology Problems developed the L1 and L2 deep diving AUVs, capable of diving down to 6000 meters and the SKAT and SKAT-GEO, equipped with photo and TV cameras allowing operation down to 350 meters (Funnell & Group, 1998).

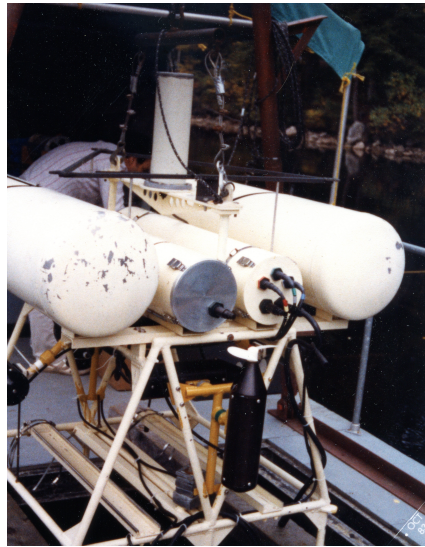


Figure 2.2: The open frame EAVE AUV (DeepOceanAndDeepSpace.com, 2015).

The technological advancements within low-powered microcomputers, memory and software engineering in the 1980s and 1990s allowed the AUV community to finally start to release the true potential of AUVs. Considerably more complex guidance and control algorithms could now be implemented to enhance the autonomy. The very first International Symposium on Unmanned Untethered Submersible Technology (UUST) was arranged in 1980 in New Hampshire, USA (AUVAC, 2013). Soon after, research programs were established giving substantial funding in order to develop working prototypes. It became clear that fully operational systems rather than testbeds would be developed in the near future.

The near future came soon enough and between 1990 and 2000 the first generation of operational systems emerged (Blidberg, 2001). These systems were able to accomplish defined objectives and potential users helped the development by stating which data they were interested to gather with an AUV system. The Autonomous Oceanographic Sampling System (AOSN) (Curtin et al., 1993) was an important step towards commercialization by showing a new paradigm for AUV usage (Blidberg, 2001). This showed that long time monitoring and hypothesis testing of the ocean environment could be economically feasible and AOSN could even revolutionize ocean sampling. For this to happen three processes needed to sustain in parallel (Curtin et al., 1993): (1) Addressing specific science questions through a series of progressively more complex experiments. (2) Integrating engineering research with basic and applied science missions. (3) Collaborating with industry to ensure economical production.

Energy capacity and total mass of the energy systems played an important role

on AUV design. Ideally, the AUVs should be fast and run long missions between each recharge. Unfortunately, average speed has the greatest impact on mass and energy requirements. Increasing the average speed 50%, from 2 to 3 m/s, demand a 100% increase in the vehicles total mass (Brighenti, 1990). Brighenti (1990) finds that an optimal trade-off between speed, range and mass is between 2 to 4 m/s. (REMUS 100 has a max speed of 2,57 m/s).

The last fifteen years AUVs have become commercial available and money has been saved (and made) by using them. WHOI, from which the REMUS vehicles emerged, has become a leader in novel research and AUV development. REMUS vehicles were used during Operation Iraqi Freedom in 2003, searching for mines in the Persian Gulf harbor of Um Qasr. It was stated that one single vehicle did a job equivalent to 12-16 human divers (WHOI, 2015).

AUSI has continued the work on AOSN and the current focus is to investigate the potential of solar powered AUVs (SAUVs). The goal is to make them operate for months by surface and recharge during the day and conduct activities during the night. Problems that arise when doing cooperative missions with many AUVs is an area with particular focus (AUSI, 2015). Highly successful experiments carried out by WHOI show that autonomous pole docking of AUVs within an AOSN, with the help of an DUSBL sensor, open the possibility for long time AUV deployment (Singh et al., 2001).

2.1.2 Sensor Systems and their Applications

An AUV can be viewed as a platform which purpose is to position an array of sensors in three dimensional space, on a specific location, underwater. The goal is to gather data that is of interest to the stakeholders, which can include academia, commercial players, military or policy sectors. This may include geological data of the seabed and/or measurements of the surrounding water.

It is common to divide the sensors into two main groups: *payload sensors* and *navigation sensors* (Sørensen & Ludvigsen, 2015). The payload sensors are the units that collect the data. Depending on the mission, an AUV can have different sensor configurations. Navigation sensors measures the state of the vehicle and is used by an internal control system to position the vehicle correctly.

Common Payload Sensors in AUVs:

Acoustic Doppler Current Profilers

Acoustic Doppler Current Profilers (ADCP) transmit an acoustic pulse and measure backscatter intensity and Doppler shift of the reflected signal. This is typically done from four transducers angled in different directions as shown in Figure 2.3. From this, the relative current velocity vector (in three spatial dimensions) can be calculated. The assumption is that the scatters in the water float with the same average speed as the current (Teledyne RD Instruments, 2011). ADCP can work as an acoustic *Doppler Velocity Log* (DVL), or *bottom-track*, by adjusting the processing of the measurements. This way the measured velocity is relative to the bottom, not the water.

An upward pointing ADCP mounted on an AUV could therefore in theory measure the velocity of a floating object. Of course the velocity of the AUV has to be known. This has applications within tracking of floating ice and was tested by Monterey Bay Aquarium Research Institute in the Arctic ocean in 2001, with promising results (McEwen et al., 2005). As of today, AUVs equipped with ADCP offer a good platform to conduct ice-monitoring, but further research is needed (Norgren & Skjetne, 2014).



Figure 2.3: One of the most popular ADCPs on the market is the "Workhorse Monitor ADCP" from Teledyne RD Instruments (Teledyne RD Instruments, 2015).

Conductivity Temperature Depth Sensors

Conductivity Temperature Depth sensors (CTD) measure the water conductivity, the temperature and the pressure (which in turn is used to calculate the depth). CTD measurements is used to find the salinity, density and speed of sound (in water). For seabed mapping and underwater navigation the speed

of sound is especially crucial to know. An example is Multibeam Echosounder Systems (MBES), where sound speed errors is one of the main causes of depth measurement errors. The measurements themselves are seldom incorrect, but the number of sound speed casts are insufficient. This way the wrong sound speed profile (SSP) is used for an area, not accounting for temporal changes in the ocean. A CTD sensor mounted on a REMUS 100 combined with the Rutgers University's Regional Ocean Modeling System (ROMS) provided a proof-of-concept of how to characterize the sound speed profile(s) in a project area (Imahori et al., 2008).

Side Scan Sonar

The Side Scan Sonar (SSS) sends fan-shaped acoustic impulse signals towards the seabed and measures the intensity and travel time of the reflected signals. By mounting the SSS on an AUV with surge speed, different cross-track slices are created which then are merged together and create a picture of the seabed. The assumption for the SSS to work is a flat seabed condition.

Compared to surface vehicles, AUVs can get close to the seabed and collect data with the SSS that has up to two orders in magnitude higher spatial resolution (Wynn et al., 2014). Other than just being used for seabed mapping, the SSS show potential in improving AUV navigation by using landmarks extracted from the SSS data together with dead-reckoning (Tena Ruiz et al., 2003).

Environmental Characterization Optics

The Environmental Characterization Optics sensor (ECO) comes in many different configurations. A typical configuration is a scattering sensor combined with a fluorimeter. This way the chlorophyll as well as the turbidity (calculated from particle scattering effects) can be measured. Field tests show that an optical sensor package like the ECO combined with CTD and ADCP creates a significant scientific data set which supports biological oceanographic research (Purcell et al., 2000). A commonly used ECO sensor which is specialized for use in AUVs is the "ECO Puck" from Wet Labs depicted in Figure 2.4 (WetLabs, 2015).



Figure 2.4: Wet Labs ECO Triplet Puck (WetLabs, 2015).

Multiparameter Sonde The vehicle also integrates other sensors for measuring Dissolved Oxygen (DO), Oxidation Reduction Potential (ORP) and pH levels of water. This data is invaluable when monitoring chemical and biological conditions in the sea.

Synthetic Aperture Sonar

Synthetic Aperture Sonar (SAS) adapts the principle of Synthetic Aperture Radar (SAR) which exploits the motion of the radar to create images with finer spatial resolution than a traditional beam-scanning radar. When the radar sends a signal and moves a distance before the signal returns, we obtain a *synthetic aperture* (the antenna seems larger). Signal processing of all recorded radar echoes outputs the final image. This processing is the "synthetic" part in SAR. SAS uses acoustic waves in water instead of radio waves in air, but the fundamentals are the same. SAS has been implemented on AUVs the last decade when problems with enabling technologies finally were figured out (Tate & Israel, 2014). The complex signal processing is done offline by powerful computers after the mission.

SAS on AUVs is particularly useful in the search of mines, since high resolution data is needed in the identification process. The two popular AUVs REMUS 600 and Bluefin 21 feature SAS capabilities (among others), where Bluefin 21 was used in the search for Malaysia Airlines Flight 370 in March 2014 after its disappearance (Tate & Israel, 2014).

Common Navigation Sensors in AUVs:

Acoustic Baseline sensors

When talking about acoustic baseline sensors, we usually talk about three classes of sensors. That is, Long Baseline (LBL) , Short Baseline (SBL) and Ultra-short Baseline (DUSBL) systems.

LBL uses two or more (preferably four) seabed mounted transponders placed around the work site to calculate the position as seen in Figure 2.5. A transceiver mounted on the UUV sends an acoustic pulse that causes the transponder to respond. From this, combined with the SSP, the distance to each transponder is found. This is done by triangulation. The deployed transponders need to be placed accurately in order to obtain a high degree of accuracy.

Compared to LBL, SBL does not require any transponder deployment on the seabed. Instead three or more transponders are mounted directly on a surface vessel. The UUV then finds its position relative to the surface vessel. The larger the transducer spacing is, the better the accuracy is. This means that SBL is best fitted for larger surface vessels.

DUSBL needs just one transponder to calculate the position. An DUSBL sensor mounted on the UUV has an array of hydrophones that is able to use the difference in phase to calculate the horizontal and vertical angles to the transponder. This combined with the range measurement gives the position relative to the transponder. Since the hydrophones typically have a spacing less than 10 centimeters, we say that the baseline is *ultra short* (compared to LBL and SBL). Angle measurement errors causes the position error to be a function of the range between the UUV and the transponder. This means that the closer the UUV is the transponder, the higher the accuracy (Kongsberg Maritime, 2015). This is later exploited when designing a docking controller.

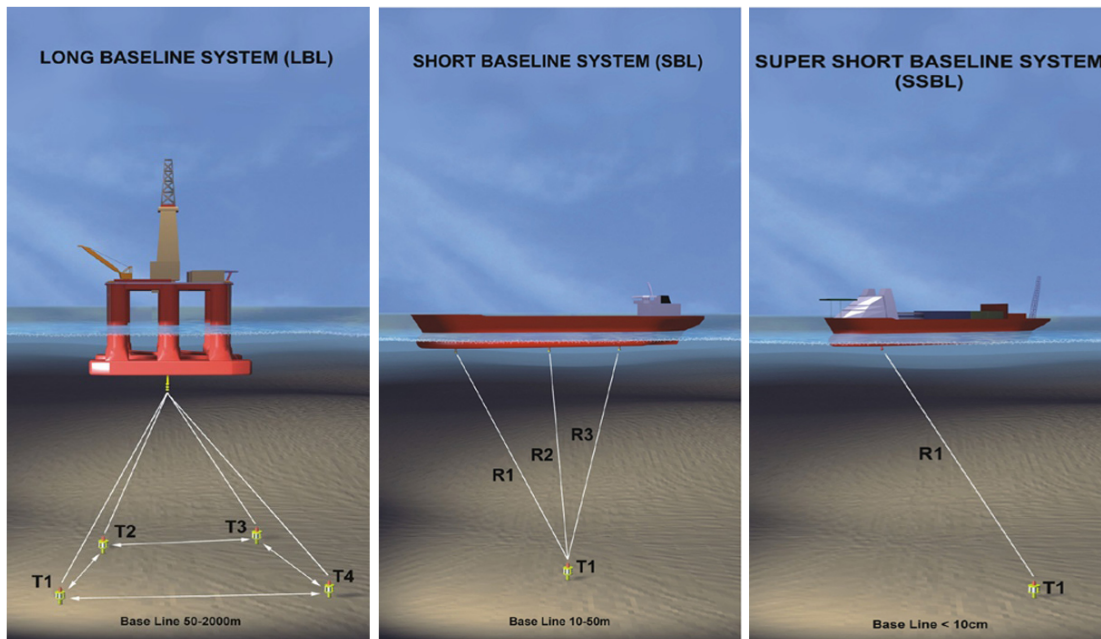


Figure 2.5: LBL, SBL and DUSBL (Kongsberg Maritime, 2015).

Table 2.1 show typical frequency ranges used by acoustic positioning systems and their maximum range. The required accuracy together with the maximum operating depth determines which frequency band the acoustic system should use.

Table 2.1: Overview of acoustic frequency bands, frequency ranges and maximum distances (Vickery, 1998).

Frequency band	Frequency Range	Maximum range*
Low Frequency (LF)	8 kHz to 16 kHz	>10 km
Medium Frequency (MF)	18 kHz to 36 kHz	2 km to 3,5 km
High Frequency (HF)	30 kHz to 60 kHz	1500 m
Extra High Frequency (EHF)	50 kHz to 110 kHz	<1000 m
Very High Frequency (VHF)	200 kHz to 300 kHz	<100 m

*This assumes in band noise on the surface vessel, at the transceiver, to be less than 95 dB and the source level of the beacon to be >195 dB re 1 μ Pa @ 1 m (Vickery, 1998).

Systems that need a range equal to the ocean depth, so called *full ocean depth systems*, usually operate in the LF band. The majority of LBL and DUSBL acoustic positioning systems operate in the MF band.

Doppler Velocity Log

By changing the signal processing of the ADCP data, it is possible to measure the Doppler shift of the signal that is reflected off the seabed. Then we say that the ADCP is working as a Doppler Velocity Log (DVL). This way the bottom relative velocity is found (called *bottom track mode*). The DVL calculated velocity can be input to a Kalman filter together with an acoustic position fix, inertial measurements, depth and a GPS signal (if at the surface) in order to determine the vehicle position, attitude, accelerations and velocities. This is particularly helpful in the case of dead-reckoning navigation of an AUV.

Heading and Inertial Sensors

There are three main concepts that are used to measure the heading of an UUV around the vertical axis: the relative position of two or more points, the magnetic field of the earth and the rotation of the earth. According to (Sørensen & Ludvigsen, 2015) the latter one is the most used for underwater navigation. A gyro compass exploits the earth's rotation and finds the vehicle heading relative to *true North* (the axis orientation with minimum potential energy), which is much more useful than the magnetic North for navigational purposes. Also, the gyro compass is not influenced by magnetic fields that the vehicle might encounter.

In the case of the REMUS 100, a HG1700AG58 Inertial Measurement Unit (IMU) from Honeywell is used to find accelerations and the rate of change of the orientation angles (Hydroid, 2012). The angular rate of change are found with three Ring Laser Gyroscopes (RLG) (Honeywell, 2015), that uses the Sagnac effect to make acceleration measurements. Accelerations are found with three quartz resonating beam accelerometers (RBA).

As mentioned in the subsection above, inertial measurements is helpful when doing dead-reckoning. Due to noise, the position obtained by integration of the accelerations and rate of angular change will drift and other measurements is needed to limit the impact of this drift.

2.2 Hardware and Software on the REMUS 100

Knowledge about the most important hardware and software components in the REMUS 100 is necessary before a controller can be implemented on the real system. This chapter provides insight in the systems that is essential for controlling the REMUS 100.

2.2.1 Specifications

The REMUS 100 can be configured in a lot of different ways from fabric to fit the customer demands and needs. Table 2.2 and Table 2.3 describe NTNU's particular configuration.

Table 2.2: AUV REMUS 100 general specifications (Hydroid, 2012).

Physical/functional characteristics	
Vehicle Diameter	19 cm
Weight in air	31 kg
Operating Depth Range	3 m to 100 m
Speed Range	0,25 m/s to 2,57 m/s
Maximum Operating Water Current	1,0 m/s
Typical Endurance	4 hours @ 4 knots 5 hours @ 3 knots

Table 2.3: AUV REMUS 100 sensor systems (Hydroid, 2012).

Sensors
Oxygen Optode Sensor (Aanderaa 4831)
Neil Brown G-CTD Sensor (NBOSI)
ECO Puck (WetLabs Triplet)
LBL High Frequency Transducer
IMU (Honeywell HG1700AG58 with NavP)
ADCP/DVL (TD Explorer R100)
Sidescan Sonar (MSTL SF 900 kHz)
GPS
Iridium modem
Wi-Fi capabilities

2.2.2 The Payload processor and the REMUS computer

The *Payload processor computer* (PP computer) is an on-board computer running the Windows 7 Embedded operating system (OS). Compared to "ordinary" Windows 7, the embedded version is a highly user customized OS that has discarded all functionality that is not needed (e.g. graphic components, drivers, applications, etc.), thus increasing performance overhead and reducing the consumption of storage space. Attempts to find documentation describing Hydroid's exact OS configuration gave no results.

Actuator control of the propeller and vertical and horizontal fins is handled by the *REMUS computer*. This is a low-level control system that gets the control inputs from the PP computer through User Datagram Protocol (UDP) (Figure 2.6) via the *RECON interface* (Section 2.2.4). When programming, the programmer must be careful not create stressful routines on the REMUS computer, since this might result in lost vehicle control.

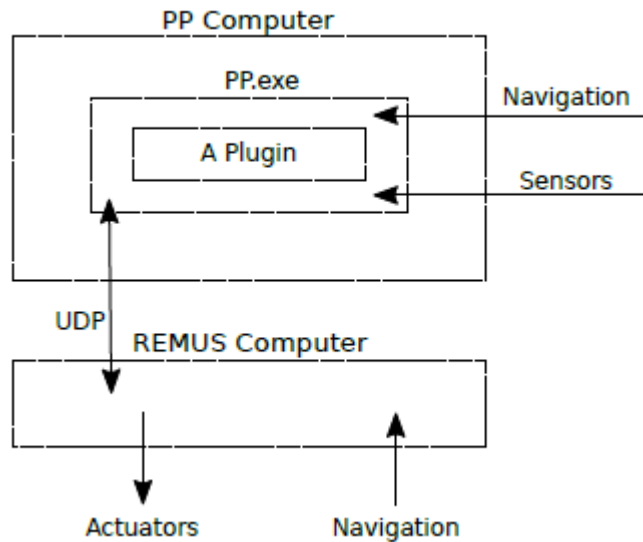


Figure 2.6: Flow chart of the interaction between the Payload Processor computer and REMUS computer (Holsen, 2015).

2.2.3 HUGIN Software Development Kit

The HUGIN Software Development Kit (SDK) is written in C++ and created to facilitate development of plugins on the HUGIN AUV, but has later been ported to support REMUS AUVs as well. It can retrieve data from the AUV (as DVL

ranges and SSS data) through an interface. Plugins written in HUGIN SDK run directly on the PP computer through the PP main executable, PP.exe program, as illustrated in Figure 2.6. The PP.exe includes a Operating System Interface (wrapper classes), device drivers (timers, IO, etc.), data storage and Common Object Request Broker Architecture (CORBA) utilities for collaboration between Oses (Kongsberg Maritime, 2012a).

Developing control and navigation algorithms is a complex task and HUGIN Software Development Kit (SDK) is not a practical developing environment for these tasks. This was the motivation behind the development of HuginDune-Bridge (Holsen, 2015), which enabled development of software in the open source framework DUNE for REMUS 100.

2.2.4 REMUS Remote Control Interface

REMUS REmote CONTROL Interface (RECON) has to be used if external programs (plugins) want to change vehicle behaviour by sending commands to the REMUS computer. Three different control modes can be selected dependent on the level of control that is needed. When a control command is accepted, the REMUS computer will send an acknowledge message back. This message needs to be equal to the original control command for the command to execute. Commands need to be sent with a frequency higher than once every 5 seconds. Else, the REMUS computer takes back control from the plugin. This is a security feature to ensure that faulty plugins or communication errors do not cause loss of vehicle control (Kongsberg Maritime, 2012b). Updates about the vehicle states (sent at the vehicle control rate of 9 Hz) is available through RECON, but not HUGIN SDK. Conversely, DVL data is only available through HUGIN SDK, but not RECON.

Control Modes

Under, the three control modes are explained. The parameters need to be set upon five seconds of activation of the control mode, and the modes can not be mixed. This means that a custom made control system can not directly regulate the propeller RPM and at the same time use the depth controller. To reduce the work of implementing the homing and docking algorithm, it should be written to utilize full-override mode. The direct-control mode would require to develop a complete new control system for the whole REMUS 100.

The control modes in RECON:

Depth-only

Only depth commands (depth or altitude) are transferred to the REMUS computer. The vehicle follows the programmed path from the mission file.

Full-override mode

Heading (angle, angular velocity or latitude/longitude goal), speed (meter per second, knots or propeller rounds per minute (RPM)) and depth (depth or altitude).

Direct-control mode

Speed (propeller RPM) and fin positions (stern and rudder positions).

2.3 Mathematical Modelling of AUVs

In order to understand the dynamics and behaviour of an AUV, it is necessary to develop a mathematical model that describes it. This is needed in model based control design, when doing simulations or creating an observer for state estimation.

The dynamics of a system is commonly divided into two parts: *kinematics* and *kinetics*. Kinematics considers the geometrical aspects of motion, while kinetics analyse the forces that creates the motion.

2.3.1 Kinematics

The use of reference frames is necessary when describing the motion of AUVs. For our purposes only the *North-East-Down* (NED) and the *body-fixed* (BODY) reference frames are considered. The NED frame is the reference frame we use everyday life to communicate position and distance and is defined relative to the Earth's reference ellipsoid. The BODY frame is fixed to the moving body, in our case an AUV. Position and orientation in BODY is relative to the inertial NED frame as illustrated in Figure 2.7.

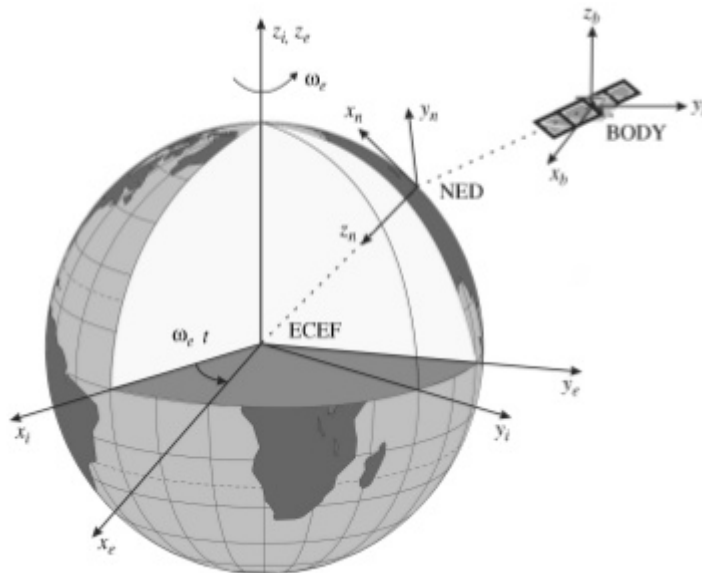


Figure 2.7: The relationship between the NED and BODY reference frames (Fossen, 2011).

Using Fossen's vectorial representation (Fossen, 2011), which is based on the SNAME standard formulation (SNAME, 1950), Table 2.4 show the notation used to describe a vessel's 6 degrees of freedom (DOF) while Figure 2.8 show the 6 DOFs.

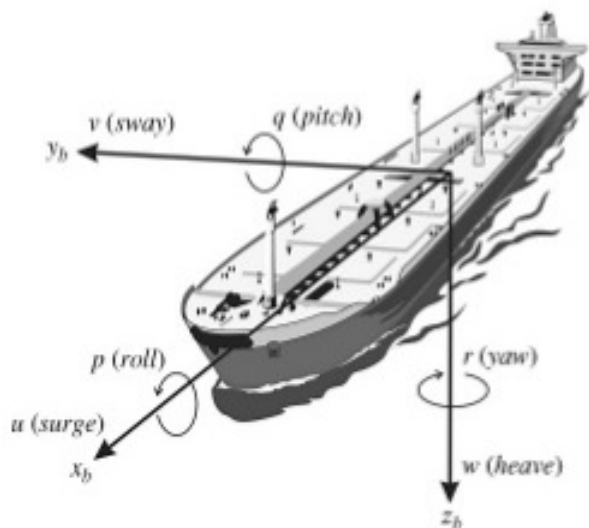


Figure 2.8: The six degrees of freedom of a moving ship (Fossen, 2011).

Table 2.4: SNAME (1950) notation for marine vessels.

DOF		Forces and moments	Linear and angular velocities	Position and Euler angles
1	Surge	X	u	x
2	Sway	Y	v	y
3	Heave	Z	w	z
4	Roll	K	p	ϕ
5	Pitch	M	q	θ
6	Yaw	N	r	ψ

$$\mathbf{P}_{\mathbf{b}/\mathbf{n}}^{\mathbf{n}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \quad (2.1)$$

$$\Theta_{\mathbf{nb}} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \in \mathbb{S}^3 \quad (2.2)$$

$$\mathbf{v}_{b/n}^b = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \in \mathbb{R}^3 \quad (2.3)$$

$$\boldsymbol{\omega}_{b/n}^b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \in \mathbb{R}^3 \quad (2.4)$$

where \mathbb{S}^3 is a sphere, \mathbb{R}^3 the real numbers, $\mathbf{p}_{b/n}^n$ the distance from NED to BODY in NED coordinates, $\boldsymbol{\Theta}$ a vector of Euler angles, $\mathbf{v}_{b/n}^b$ is the linear velocity of a point o_b with respect to n expressed in b and $\boldsymbol{\omega}_{b/n}^b$ the body fixed angular velocity of b with respect to n expressed in b . The 6 DOF general motion of a marine craft is then described by

$$\boldsymbol{\eta} = \begin{bmatrix} \mathbf{p}_{b/n}^n \\ \boldsymbol{\Theta}_{nb} \end{bmatrix} \quad (2.5)$$

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} \quad (2.6)$$

hence the 6 DOF kinematic equations becomes

$$\begin{bmatrix} \dot{\mathbf{p}}_{b/n}^n \\ \dot{\boldsymbol{\omega}}_{b/n}^b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) & 0_{3 \times 3} \\ 0_{3 \times 3} & \mathbf{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) \end{bmatrix} \begin{bmatrix} \mathbf{v}_{b/n}^b \\ \boldsymbol{\omega}_{b/n}^b \end{bmatrix} \quad (2.7)$$

or in compact form

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_{\boldsymbol{\Theta}}(\boldsymbol{\eta})\boldsymbol{\nu} \quad (2.8)$$

where the diagonal elements $\mathbf{J}_{11}(\boldsymbol{\eta}) = \mathbf{R}_b^n(\boldsymbol{\Theta}_{nb})$ and $\mathbf{J}_{22}(\boldsymbol{\eta}) = \mathbf{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb})$ are defined as the rotational matrices

$$\mathbf{R}_b^n(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\theta s\phi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (2.9)$$

$$\mathbf{T}_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}_{nb}) = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (2.10)$$

and $s \cdot = \sin(\cdot)$, $c \cdot = \cos(\cdot)$ and $t \cdot = \tan(\cdot)$ are used for compact notation.

2.3.2 Kinetics

General Kinetic Model for Ocean Vessels

It can be shown (Fossen, 2011) that a vectorial 6 DOF kinetic model can be written as

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) + g_0 = \tau + \tau_{wind} + \tau_{wave} \quad (2.11)$$

where

$$M = M_{RB} + M_A:$$

Rigid body and added mass inertial matrix.

$$C(\nu) = C_{RB}(\nu) + C_A(\nu):$$

Rigid body and added mass coriolis matrix.

$$D(\nu) = D_P + D_V + D_n(\nu) + \mu:$$

Potential, viscous and non-linear damping and fluid memory effects.

$$g(\eta):$$

Metacentric restoring forces.

$$g_0:$$

Forces and moments due to ballast tanks.

$$\tau_{wind} + \tau_{waves}:$$

Wind and wave forces.

$$\tau:$$

Propulsion forces.

6 DOF Kinetic Model for AUVs

AUVs normally operate below the wave-affected zone, hence is independent of wave excitation frequencies. Also, we can assume a starboard-port symmetrical hull geometry. According to (Fossen, 2011) 2.11 reduces to

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau \quad (2.12)$$

where

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & -X_{\dot{w}} & 0 & mz_g - X_{\dot{q}} & 0 \\ 0 & m - Y_{\dot{v}} & 0 & -mz_g - Y_{\dot{p}} & 0 & mx_g - Y_{\dot{r}} \\ X_{\dot{w}} & 0 & m - Z_{\dot{w}} & 0 & -mx_g - Z_{\dot{q}} & 0 \\ 0 & -mz_g - Y_{\dot{p}} & 0 & I_x - K_{\dot{p}} & 0 & -I_{zx} - K_{\dot{r}} \\ mz_g - X_{\dot{q}} & 0 & -mx_g - Z_{\dot{q}} & 0 & I_y - M_{\dot{q}} & 0 \\ 0 & mx_g - Y_{\dot{r}} & 0 & I_{zx} - K_{\dot{r}} & 0 & I_z - N_{\dot{r}}, \end{bmatrix} \quad (2.13)$$

$$\boldsymbol{\tau} = [X_p \ Y_r \ Z_s \ M_s \ N_r \ K_p]^T \quad (2.14)$$

and $\mathbf{C}(\boldsymbol{\nu})$ is computed from \mathbf{M} . $\mathbf{D}(\boldsymbol{\nu})$ is simplified by neglecting higher-order damping.

Y_r , Z_s , M_s and N_r are forces and moments from the rudder and stern fins. The equations needed for calculation are derived in Prestero (2001) as well as the necessary coefficients. X_p and K_p are the surge force and roll moment from the propeller which can be calculated as done in Carlton (2007) with the propeller coefficients for REMUS 100 estimated by Allen et al. (2000).

The other hydrodynamic coefficients needed for modelling the REMUS 100 have been calculated by Prestero (2001). These coefficients are used in AUVSim (Norgren & Skjetne, 2015) when implementing the mathematical model given by 2.8 and 2.12.

If subjected to irrotational and constant current 2.12 can be written as

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (2.15)$$

2.3.3 Simulation, Control and Observer Design Models

Equation 2.11 is suitable for use in a so called *simulation model*. A simulation model is a high-fidelity model that models the real world as closely as possible, with time responses similar to the real system. This model is computationally heavy to run and is used in numerical performance and robustness analysis and testing of the controller systems (Sørensen, 2013).

By simplifying the simulation model, we obtain a *control design model*. This model can be used directly in a controller (e.g. finding Proportional-Integral-

Derivative (PID) controller gains) or in stability analysis (e.g. Lyapunov stability). Also, control design models are used to generate feedforward signals in more advanced controllers. Often, coupled three dimensional motion is not needed to model in a simulation or control design model. In such cases the motion can be decomposed into a lateral and horizontal motion model. Appendix A gives the equations needed for decomposing the model of an AUV, such as REMUS 100. Equation A.3 is suited as a lateral control design model.

An *observer design model* is useful in observer design since the focus here is more shifted towards accurate noise measurements, filtering and the dynamics between sensors and the navigation system. Again, it is a simplification of the simulation model.

2.4 Line-of-Sight Guidance

Line-of-sight (LOS) is a commonly used method when a vessel needs to do path following. By creating an LOS vector between the vessel and the next waypoint, or on the line connecting two waypoints, the vessel can use its autopilot to track the path. This is done by setting the angle between the LOS vector and the path as a reference for the autopilot. LOS guidance is used in the AUV control algorithm to move the AUV to a better position if the initial cross-track error after the homing phase is too large. This section provides the theoretical background on which the implemented LOS algorithm is based upon.

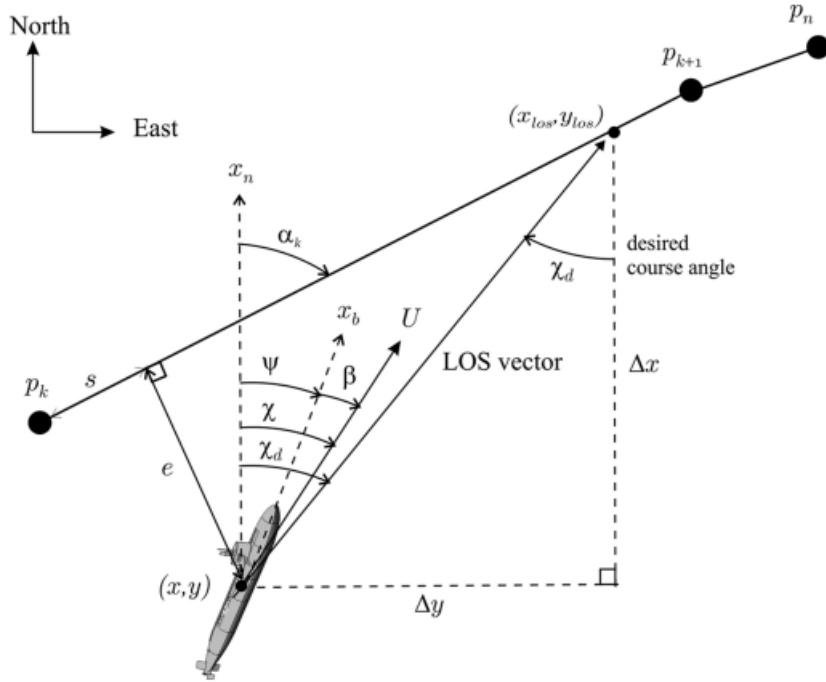


Figure 2.9: LOS guidance. Course angle χ_d points towards the LOS intersection point (x_{los}, y_{los}) (Fossen, 2011).

In figure 2.9 the points $p_k^n = [x_k, y_k]^\top \in \mathbb{R}^2$ and $p_{k+1}^n = [x_{k+1}, y_{k+1}]^\top \in \mathbb{R}^2$ are connected with a straight line. α_k is positive and defined as a rotation of the x axis of a path-fixed reference frame:

$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \in \mathbb{S} \quad (2.16)$$

Then, the along-track distance $s(t)$ (tangential to path) and cross-track error

$e(t)$ (normal to path) are expressed as:

$$s(t) = [x(t) - x_k] \cos(\alpha_k) + [y(t) - y_k] \sin(\alpha_k) \quad (2.17)$$

$$e(t) = -[x(t) - x_k] \sin(\alpha_k) + [y(t) - y_k] \cos(\alpha_k) \quad (2.18)$$

The guidance principle used for the steering is chosen as a *lookahead-based steering*, as described in Breivik and Fossen (2009). This is less computationally intensive than *enclosure-based steering* and is valid for all cross-track errors, whereas enclosure-based steering has certain requirements. Enclosure-based steering will be given no further consideration in this thesis.

The desired course angle χ_d consists of two parts, χ_p and χ_r such that

$$\chi_d(e) = \chi_p + \chi_r(e) \quad (2.19)$$

where $\chi_p = \alpha_k$ is the path-tangential angle from equation 2.16 and $\chi_r(e)$ is the velocity-path relative angle given by the expression

$$\chi_r(e) = \arctan\left(\frac{-e}{\Delta}\right) \quad (2.20)$$

Δ is called the *lookahead distance* and is the distance between (x_{los}, y_{los}) and the point where e stands orthogonal on the line between p_k and p_{k+1} , in figure 2.9. Breivik and Fossen (2009) interprets the steering law 2.20 as a saturating control law and introduce integral action, hence obtaining the desired yaw angle χ_d as

$$\chi_d(e) = \alpha_k + \arctan\left(-K_p e - K_i \int_0^t e(\tau) d\tau\right) \quad (2.21)$$

The integral action enables the control law to steer an underactuated vessel (such as AUV REMUS 100) along a path, while having a nonzero sideslip angle β . A switching mechanism for deciding when to select the next waypoint is needed. Since the main purpose of the LOS guidance is to guide the AUV in a general direction, the switch is allowed to happen even though the AUV is not within a *circle of acceptance* encircling the waypoint. Considering just the along-track distance, the switch happens when

$$k_{+1} - s(t) \leq R_{k+1} \quad (2.22)$$

2.5 Particle Filtering

”Particle filtering (PF) is a general Monte Carlo method for performing inference in state-space models where the state of a system evolves in time and information about the state is obtained via noisy measurements made at each time step” (Orhan, 2012). Because of this, PFs are often called ”sequential Monte Carlo” (SMC) methods. A Monte Carlo method is a computational algorithm that obtains numerical results by repeatedly perform random sampling. Monte Carlo methods are particularly useful in modelling systems with significant uncertainties in the input parameters. Examples of such systems are business risk models, simulation of fluids or any other system which can be interpreted as probabilistic.

The main idea behind a PF is to create a set of particles that can represent a probability density function (PDF). Every particle contains a variable with values of the system states. This way the PDF indicates the likelihood of one particular system state. Since this method is not limited to any specific distributions it is well suited for non-Gaussian PDFs and non-linear models. PFs are often used in Simultaneous Localization and Mapping (SLAM). Here both the map and vehicle/vessel location needs to be calculated at the same time. In our case, the map is not of particular interest, only the AUV position relative to the transponder. Therefore the PF described is strictly speaking a *localization* algorithm.

2.5.1 The Generic Particle Filter

From statistics the *conditional probability* is the probability of an event, given that another event has occurred. If all known (relevant) information about a random event or process is considered, then the calculated conditional probability is called the *posterior probability*. The *posterior probability distribution* is the probability distribution of a stochastic variable representing an unknown quantity. This stochastic variable is *conditional* on the observed evidence. The generic PF (GPF) is able to make an estimate of the posterior distribution of the unknown system states, given an observation measurement process. The unknown states are estimated sequentially, at each time step, from the values of the observation process.

The PF algorithm consists of three steps which are carried out at each time step - prediction, update and resample - as well as initialization at the very first time step. The following generic PF is based on the PF outlined in Bahr (2009) and is well suited for the purpose of AUV state estimation.

Initialization step

The PF is initialized with a set $\mathbf{C}(0)$, where the 0 indicates the first time step and $\mathbf{C} = \{c_1, \dots, c_i, \dots, c_n\}$ holds n samples (particles) consisting of the state vector \mathbf{x}_i and a sample weight w_i such that $\mathbf{c}_i = [\mathbf{x}_i, w_i]$. In the case of autonomous docking the state vector holds the estimated northings and eastings coordinates of the AUV, $\mathbf{x}_i = [Northings, Eastings]$. By assuming that the internal gyrocompass in the AUV provides good measurements with no significant drift, it is not necessary to include the heading in the state vector. This assumption is substantiated through previous field results with the REMUS 100 as presented in Hewage et al. (2015), where the gyro error did not have any significant effect on the total dead reckoning error.

Initially, the set $\mathbf{C}(0)$ is generated by drawing n samples from a distribution that represents the assumed state. If there is no data or measurements to make a rough estimate of the position, particles are simply scattered uniformly all over the map. If the initial position is perfectly known, all particles are instantiated with the same state vector. If the initial position are known, but is influenced by some error, this is accounted for by sampling particles around this point. For instance ± 50 meters in every direction about the initial position. All particles are initially assigned equal weights $w_i = \frac{1}{n} \quad \forall i$.

Prediction step

The prediction step applies a motion model to each particle such that the initial particle cloud moves to a new position. The motion model also includes noise parameters to simulate noise and uncertainties. These parameters can later be used to "tune" the behaviour of the PF, or more precisely, the *mutation* of the particle cloud. The prior distribution is approximated by the resulting distribution of particles.

Since the motion model is applied to every particle and the total number of particles often needs to be very large, an integral part of the prediction step is to select a motion model that computes easily. Bahr (2009) suggests the model

$$\mathbf{x}_i(k) = \begin{bmatrix} x_i = x_i(k-1) + u_i(k)\cos(\theta_i(k))dt \\ y_i = y_i(k-1) + u_i(k)\sin(\theta_i(k))dt \end{bmatrix} \quad (2.23)$$

where $u_i(k) = \text{sample}(\mathcal{U}_u)$ is a sample from the control space distribution \mathcal{U}_u (which means that the signal is influenced by noise). Physically, $u_i(k)$ is the AUV speed relative to the seabed. Recall that θ_i is taken directly from the gyro

measurements and not from its own control space \mathcal{U}_θ . Note that many different distributions could be necessary for larger state vectors.

Instead of sample from a control space, an equally good alternative is to add a random displacement of the possible particle positions afterwards (Burdinsky & Otcheskiy, 2014). This method allows for slightly easier tuning (because of fewer parameters) and is the one implemented in this thesis. The random displacement is represented by zero mean normal numbers with standard deviation, σ_{noise} , added to Equation 2.23.

Update step

The update step is performed if a measurement $m^j(k) = [\mathbf{C}^j(k), \text{id} = j_{j(k)}, r_{j(k)}]$ is available. \mathbf{C}^j contains the particles with the state values with the corresponding weights from the unit that sent the measurement. It is basically the particle cloud of another unit as well as the measured distance $r_j(k)$ between the sender and the receiver. As can be seen, this is why this algorithm is particularly useful for cooperation between several AUVs where all AUVs estimate the position from their own particle cloud. (An acoustic modem for transmitting the particle data is needed, though.) In this thesis the receiver is the REMUS 100 AUV, while the sender is the transponder mounted on the docking station. $r_{j(k)}$ is measured with the LBL sensor only between the AUV and the docking transponder, so $\text{id} = j$ is constant. By assuming the dock position is known, all particles in the set $\mathbf{C}^j(k)$ have state vectors representing the same position. This way it does not matter if the assumed position is very accurate because the AUV will find its relative position from the dock position.

The main idea in the update step is to transform the particles $\mathbf{x}_i(k)$ along r_j and find the likelihood of each particle belonging in \mathbf{C}^j . First, the weighted mean $\boldsymbol{\mu}^j(k)$ of all \mathbf{c}_i^j in \mathbf{C}^j computes as

$$\boldsymbol{\mu}^j(k) = \sum_{i=1}^n \mathbf{x}_i^j(k) w_i^j(k). \quad (2.24)$$

Then, the measured range is assumed to have an error which is normal distributed such that the likelihood can be found by

$$w_i(k) = \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left(-\frac{(\|\mathbf{x}_i - \boldsymbol{\mu}^j\|_2 - r_j)^2}{2\sigma_r}\right) \quad \forall \quad \mathbf{x}_i(\mathbf{k}), \quad (2.25)$$

where σ_r is the (assumed) standard deviation of the range measurements. In the end, the weights are normalized to make sure that $\sum_{i=1}^n w_i = 1$:

$$w'_i(k) = \frac{w_i(k)}{\sum_{i=1}^n w_i(k)}. \quad (2.26)$$

The new particle weights now represents the likelihood of getting the sensor reading $r_{j(k)}$ from the hypothesis of the particles.

Resample step

During the resample step, particles with low weights are deleted while particles with high weights are replicated. This way the particle cloud becomes more condensed and closer to the actual system state. The goal is, while keeping the total number of particles the same, to represent the PDF with accumulation of particles in high-probability areas, whereas low-probability areas are assigned fewer particles. Figure 2.10 illustrates this. First, ten particles are randomly distributed and given weights according to their probability (the blue dots, where their size represents the weight). Then the resampling step selects the fittest particles and the next prediction step introduces variety.

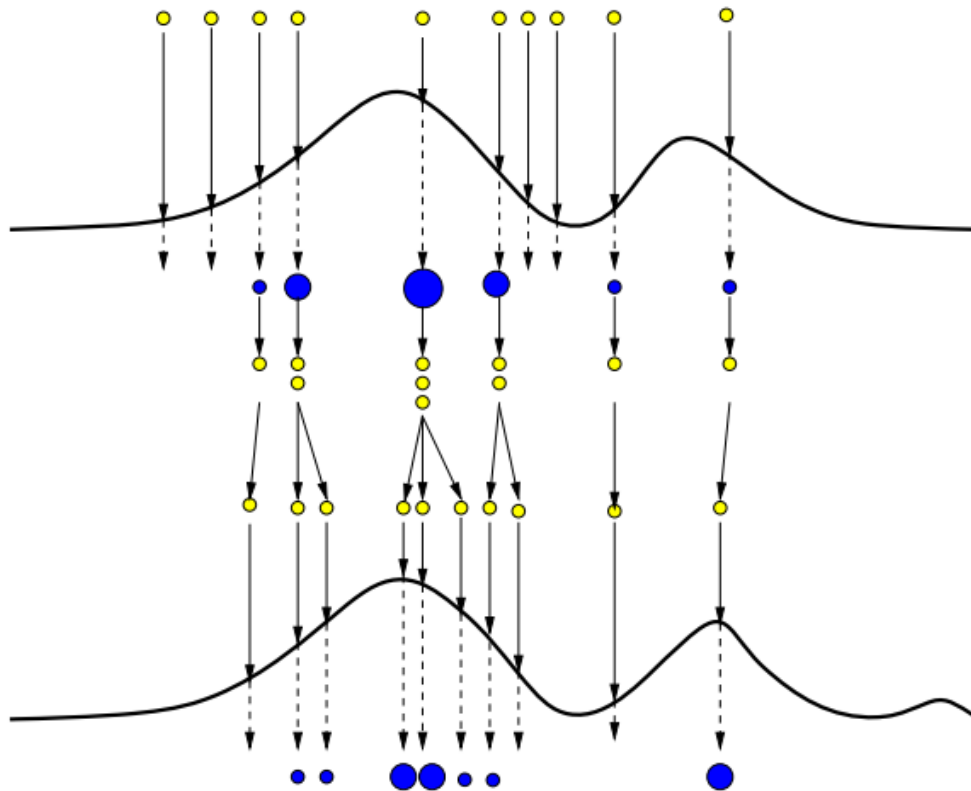


Figure 2.10: Particle resampling with ten particles (Doucet et al., 2001).

Without resampling, unlikely particle states would transition through the motion model to even more unlikely states. In the end, all particles except for one would have almost-zero weight. This phenomena is called *particle depletion* where just one "remaining" effective particle is supposed to represent the entire PDF. This is not a good solution and means that the weighting function 2.25 must be chosen with caution, despite the resampling step.

Ideally, the particles have similar weights which enables the best tracking of the distribution. For the same reasons as discussed above, particles quickly obtain either a very high or a very low weight. A commonly used estimate of the number of effective particles, N_{eff} , is

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^n w_i(k)^2}. \quad (2.27)$$

with

$$N_{\text{th}} = \frac{2N}{3}. \quad (2.28)$$

as the threshold which determines if resampling is needed (Donovan, 2012). This is called *adaptive resampling* and is often used because it has proven to work well in practice (Särkkä, 2012). The weighted mean over the particle states is an easy way to provide a state estimate that could be used as an input to a control system.

When using the term *resampling* it is in reality implied *Sequential Importance Resampling* (SIR). SIR improves *Sequential Importance Sampling* (SIS) by adding resampling, thus avoiding degenerativity (convergence to a single non-zero weight $w_i = 1$). Several resampling algorithms are tested for practical performance with simulations in this thesis. These algorithms are taken directly from Hol et al. (2006) and reproduced below.

Multinomial resampling:

Generate N ordered uniform random numbers (the variables used here are different from the above).

$$u_k = u_{k+1} \tilde{u}_k^{\frac{1}{k}}, \quad u_N = \tilde{u}_N^{\frac{1}{N}}, \quad \text{with } \tilde{u}_k \sim U[0, 1]$$

and use them to select the new set of particles, $\{x_k^*\}$, according to the multinomial distribution. That is,

$$x_k^* = x(F^{-1}(u_k)) = x_i \quad \text{with } i \text{ such that } u_k \in \left[\sum_{s=1}^{i-1} w_s, \sum_{s=1}^i w_s \right),$$

where F^{-1} denotes the generalized inverse of the cumulative probability distribution of the normal particle weights.

Systematic resampling:

Generate N ordered numbers

$$u_k = \frac{(k-1) + \tilde{u}}{N}, \quad \text{with } \tilde{u} \sim U[0, 1]$$

and use them to select x_k^* according to the multinomial distribution.

Residual resampling:

Allocate $n'_i = \lfloor Nw_i \rfloor$ copies of particle x_i to the new distribution. Additionally, resample $m = N - \sum n'_i$ particles from $\{x_i\}$ by making n''_i copies of particle x_i

where the probability for selecting x_i is proportional to $w'_i = Nw_i - n'_i$ using one of the resampling schemes above.

Extensively testing and simulations done by Hol et al. (2006) proves that multinomial and systematic resampling are similar in quality. From the three methods, systematic resampling was favourable over multinomial, and the theoretical superior one. Residual resampling was harder to place, because half of the particles were chosen deterministically and the other half by either multinomial or systematic resampling. Hence simulations are needed to determine the effectiveness. The computational times in figure 2.11 clearly shows the advantage of systematic and residual resampling over multinomial. Stratified resampling is not discussed further, but is considered to be favourable over multinomial.

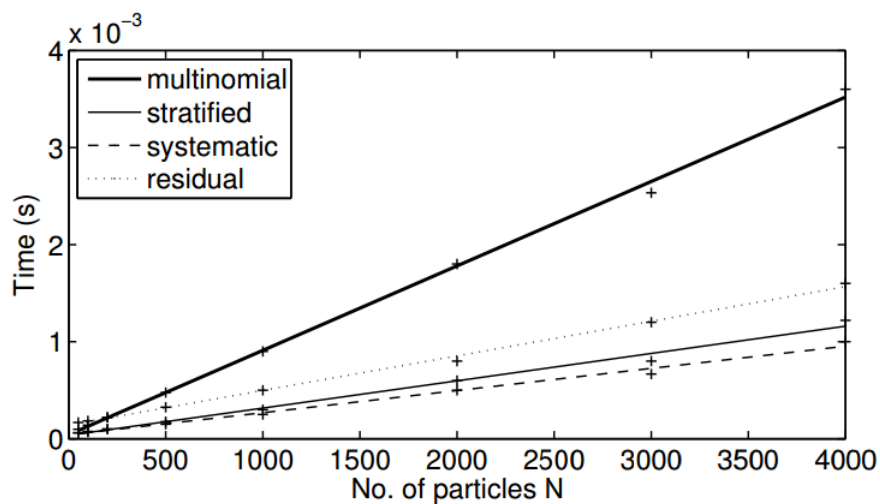


Figure 2.11: Computational time with different sized particle sets (Hol et al., 2006).

2.6 Previous Work on Homing and Docking

Since the deployment of the first AUVs, work has been done to increase their level of autonomy, thus reducing the human labour and intervention needed in AUV operations. Preparing ship and crew for *each* AUV mission takes a lot of resources and effectively limits the AUV utility. Autonomous AUV homing and docking were a natural part of this work. WHOI has proved themselves leading in this field with a lot of interesting work (Bellingham et al., 1994), (Stokey et al., 1997), (Singh et al., 2001), (Kukulya et al., 2010).

2.6.1 Homing and Docking Attempts

Odyssey II and the First Generation REMUS AUV

Bellingham et al. (1994) provided the Odyssey II AUV with homing capabilities with the help of a DUSBL system. This system was only activated in the recovery phase due to high power consumption. LBL was used in normal operation mode. This principle can be adapted to the REMUS 100 since there is room for both sensors. On the REMUS 100 the DUSBL sensor will be mounted in the front cap, while the LBL sensor is mounted underneath the front cone. Field testing of the Odyssey II in the arctic showed that the AUV returned to the homing transponder with an accuracy of just around 30 cm. A suspended net under the ice was used as a "docking station". For testing purposes a net has the advantage that it is simple to set up and has more error margin than "pole docking" systems or stations with a cone shaped inlet (a cone dock).

Stokey et al. (1997) used a cone dock (Figure 2.12) for autonomous docking of one of the early REMUS vehicles with a DUSBL system. The docking task is more complicated since the orientation of the dock has to be known (if not fastened) and a glide path needs to be calculated. This motivates to not create a cone dock in a first docking attempt with the REMUS 100. Despite this, a dock that protects the AUV from the elements should be the ultimate goal.

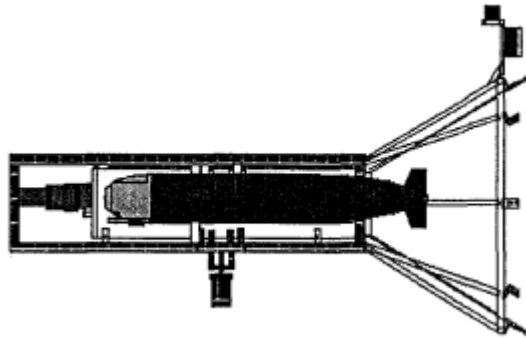


Figure 2.12: An AUV docked in a cone docking station (Stokey et al., 1997).

Short Range Positioning System on the SWIMMER AUV

A new navigation system called Short Range Positioning System (SRPS) is utilized in the SWIMMER AUV and is capable of performing autonomous docking on great depths (Evans et al., 2001). The SWIMMER AUV is a unique AUV which houses a work-class ROV that is deployed when the AUV is docked on the seabed. Power and (real-time) control signals are provided through the subsea infrastructure network to a surface vessel or a platform. Areas of usage is work on offshore structures. SRPS features Active Sonar Object Prediction (ASOP) developed by Under the Subsea Technology Research Programme at The University of Liverpool, originally as a part of a joint research programme founded by industry and the UK Government. ASOP uses a 3D map of the environment to perform a search for landmarks (significant sonar features) by analysing sonar data. In turn SRPS uses ASOP to derive the AUVs location. Field tests show that the system works very well for docking (Evans et al., 2001).

The SWIMMER AUV is fully actuated and can move in all six DOFs, compared to the REMUS 100. This makes docking of REMUS directly on the seabed a demanding task. An interesting idea would be to use the SRPS (fed with SSS data) early in the docking stage, when there still is uncertainties in the DUSBL measurements. This way a better glide path could be created towards a docking station.

Docking of AUVs within an AOSN

AOSN was explained in Chapter 2.1.1 and that autonomous docking was important to prolong oceanographic activities. Singh et al. (2001) successfully used an DUSBL system to home and dock an AUV onto a vertical pole, as can be seen

in Figure 2.13. It homes to a transponder mounted on the pole and docks when a latch in the bow of the AUV trips and captures the pole. This system allows for approach from any direction (omnidirectional), which can almost eliminate cross-track error by choosing a path that face directly towards the ocean current.

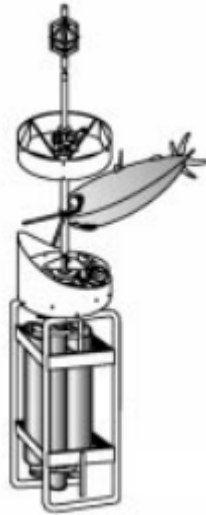


Figure 2.13: An AUV latched to a pole docking station (Singh et al., 2001).

The homing algorithm is simple and works by minimizing (nulling) the relative bearing to the dock with a PID controller. The desired heading is calculated outside the PID loop and used as an controller input. The same control strategy is used in Bellingham et al. (2008) when designing a control system for autonomous docking of the Dorado/Bluefin 21” AUV (Chapter 2.6.1).

Homing and Docking of the Dorado/Bluefin 21” AUV

An unfixed weathervaning cone dock is omnidirectional which allows for a docking phase towards the current, but the heading needs to be communicated to the AUV(s). It also needs a way of transferring power and signals over a rotating surface (e.g. a slip ring). A pole dock is also omnidirectional, but requires an external latching system mounted on the AUV. The Monterey Bay Aquarium Research Institute (MBARI) created a fixed-heading cone dock with the goal of having the dock as simple as possible (Bellingham et al., 2008). This meant that the AUV needed algorithms to cope with potential cross track error.

A homing control loop and a docking control loop were created to successfully drive the AUV into the cone dock. The homing control loop implemented pro-

portional navigation with a gain equal to unity, effectively creating a pure pursuit guidance controller. This way the AUV always pointed the nose towards the transponder. The docking controller used successive loop closure with an outer loop PID controller with the cross-track error as input. The inner loop was a proportional controller regulating the heading error.

Simulations were done using equation A.4 modified to include heading deviation and cross-track displacement. The results showed acceptable convergence to the a straight-line path leading into the dock after 200 meters. Considering the smaller size of the REMUS 100 compared to the Dorado/Bluefin 21", it is reasonable to believe that the dynamics is significantly faster and that the time to convergence can be reduced. As a result the 200 meters could most likely be reduced.

Homing and Docking in Arctic, Icy Conditions

If operating in arctic conditions with just one transponder on the docking station and a DUSBL sensor on the vehicle, a serious problem may occur. If the vessel does not establish connection with the transponder over a long period of time, the AUV can become lost due to position drift (Kukulya et al., 2010). Therefore a LBL system should be set up for backup. GPS fix is not possible under ice, so it can not contribute to the position estimate. LBL transponder deployment in several meters of ice may not be practical or possible in some cases.

Kukulya et al. (2010) successfully managed to repeatedly dock the REMUS 100 in a docking net with a transponder suspended inside and a DUSBL system. Experienced showed that if docking was not achieved on the first few attempts, more attempts were futile. The LBL system then had to be used to reestablish position estimates. These experiences should be taken seriously and accounted for when designing algorithms designed for under ice usage.

2.6.2 Underwater Navigation with Particle Filters

Particle filters have a lot of different applications in underwater navigation of AUVs. This include using a PF to estimate the position from noisy acoustic measurements or finding a position fix given a bathymetric map with a combination of payload measurements. The latter scenario is often referred to as non-traditional navigation (NTN) when used in situations where the DVL is out of bottom tracking and it's not possible/practical to get a GPS fix (e.g. the AUV is under ice or at great depths). The same principle is heavily used

in terrain navigation of cruise missiles and air crafts. Donovan (2012) discusses the performance of different altitude sensors in AUV PF terrain navigation and showed that the INS drift can be reduced, but demands a good tidal model. Even though no research about terrain navigation is done in this thesis, it is important to understand the possibility of incorporating such methods in future navigational systems together with other estimation methods. Here, single beacon PF navigation is of special interest.

Previous single beacon range-only navigation attempts have been done using traditional extended Kalman filters (EKF). Baccou and Jouvencel (2002) and Casey, Guimond, and Hu (2007) show successful results when homing to a deployed transponder and Saúde and Aguiar (2009) navigating relative to a known transponder. Since KF methods depend on Gaussian distributed noise, this motivates to use a method that does not set any restrictions on the noise distribution, thus is applicable in all scenarios. One such method is the PF.

Simulations done by the University of Porto, on the MARES AUV, show that the PF was able to successfully home the AUV to a single LBL beacon (Ferreira, Matos, & Cruz, 2010), but the number of particles needed (over 1000 particles) was too computationally heavy to actually do in practice. Further investigation on the MARES AUV shows that the onboard computational system is a PC104 stack configuration (Cruz & Matos, 2008). This is basically the same used in the 33 years old IBM Personal Computer XT. REMUS 100 runs an embedded version of Windows 7 which means that the computational power available is many orders of magnitude more powerful than that of the MARES. Hence, REMUS 100 should have no problem to run the required number of particles. Chapter 4.2 shows the results when doing simulations using 1500 particles on the REMUS 100 model.

For AUVs, literature about *single-beacon LBL localization using the GPF* is scarce. The majority of the literature consider solving the SLAM problem with either a Rao-Blackwellized Particle Filter (RBPF) or a Sum of Gaussians (SOG) filter, as implemented on the Girona500 I-AUV (Vallicrosa, Ridao, Ribas, & Palomer, 2014). Even though these methods could be used to solve the localization problem, it is hard to deduce the accuracy used in just a localization scenario from SLAM results. This motivates to research the topic further. An area of ongoing research is how to reduce the number of particles in RBPF to better the computational performance (Olson, Leonard, & Teller, 2006). This is important in smaller systems with limited processing resources. The algorithm found most suitable to solve my problem is that of Bahr (2009) described in Chapter 2.5.

Chapter 3

Methods

This chapter describes the methods used to create the final results presented in Chapter 4. First, the algorithms for homing and docking with the DUSBL and LBL sensor are described in detail in Section 3.1. Then the simulator environment, in which the algorithms are implemented, are presented in Section 3.2. Section 3.3 summarizes the LBL test conducted with REMUS 100 on Trondheimsfjorden 2. June, 2016.

3.1 Homing and Docking Algorithm

The phrase "docking of an AUV" in reality often mean "the homing and docking of an AUV". Docking is a word that is easily resorted to in daily speech and one can forget the fact that there are *two* important phases involved in a successful "docking" operation; namely, *homing* and *docking*. Homing is the determination of a signal source and the process of moving towards it. Docking is the action taken to run the AUV into the docking station and joining them together. The docking process typically starts in close proximity to the station itself and needs a high degree of accuracy. To avoid any ambiguity, this chapter has a clear distinction between the two phases and treats them separately.

First, homing and docking with the DUSBL sensor is discussed, followed by homing and docking with the LBL sensor.

3.1.1 DUSBL aided Homing and Docking

Section 3.1.1 presents my proposed algorithm for DUSBL aided autonomous homing and docking of the REMUS 100. It is written as a high level pseudo code for readability. Section 3.2 and 3.2 discusses the details and assumptions made when creating the algorithm. The work in section 3.1.1 is mainly based on the work done in my project thesis, fall 2015.

Proposed Algorithm

Algorithm 1: DUSBL Aided Homing and Docking

```

1 if No DUSBL signal received yet then
2   └ Dead reckon to assumed transponder position
3 if DUSBL signal received then
4   Start homing phase:
5     Calculate range and relative bearing to transponder;
6     Calculate relative depth to transponder;
7     Minimize relative bearing to transponder;
8     Set reference depth equal to transponder depth;
9   if No DUSBL signal then
10    └ Use last measurements of the transponder
11  if Range to transponder  $\leq 200$  [m] and cross-track error  $< 50$  [m] then
12    Start docking phase:
13      Calculate cross-track error;
14      Minimize cross-track error;
15      if Range to transponder  $\leq 15$  [m] then
16        └ Set AUV speed to 0.8 [m/s]
17      if Range to transponder  $\leq 0.2$  [m] and AUV speed  $\leq 0.2$  then
18        └ Docking successful;
19          └ Stop algorithm
20      if Range to transponder  $> 2$  [m] and Relative bearing to transponder
21         $> 90^\circ$  then
22        └ Docking unsuccessful;
23          └ Generate waypoints to new initial docking phase position;
24            └ Start LOS algorithm between the waypoints;
25              └ if At last waypoint then
26                └ Start docking phase, line 12
26      else if Range to transponder  $\leq 200$  [m] and cross-track error  $\geq 50$  [m]
27        then
28          └ Generate waypoints to new initial docking phase position;
29            └ Start LOS algorithm between the waypoints;
30              └ if At last waypoint then
31                └ Start docking phase, line 12
31      else
32        └ Continue homing

```

Homing Phase

The homing phase starts when a transponder signal is received by the DUSBL sensor, such that the position of the transponder can be determined. Before this, the AUV uses dead-reckoning (dead-reckoning phase) to the point where the dock is supposed to be located, (E_D, N_D) . This location is pre-programmed in the control algorithm and does not need to be very accurate. The main idea is to make the AUV travel the right direction, such that a transponder signal can be picked up. The last transponder measurement (range and relative bearing) is stored for use in the case of signal drop-outs when the homing phase is initiated.

In the dead-reckoning phase γ_{dock} (see Figure 3.1) is found as

$$\gamma_{dock} = \arccos \left(\frac{\vec{AB} \cdot \vec{BC}}{\|AB\| \|BC\|} \right) \quad (3.1)$$

with

$$\vec{AB} = \underbrace{(E_{AUV}, N_{AUV})}_B - \underbrace{(E_{AUV}, N_{AUV} + \Delta N)}_A \quad (3.2)$$

and

$$\vec{BC} = \underbrace{(E_D, N_D)}_C - \underbrace{(E_{AUV}, N_{AUV})}_B \quad (3.3)$$

When using the DUSBL sensor, an important task is to ensure optimal conditions for signal retrieval. Signals can be received at $\pm 45^\circ$ with decaying signal strength at increasing angles. This means that the sensor should point in the direction of the transponder on the docking station at all times for best results (Bellingham et al., 2008). For the AUV to home to the docking station, the relative bearing θ needs to be minimized. Note that θ can be both positive and negative. From Figure 3.1 we see that this is done when $\psi_{AUV} \rightarrow \gamma_{dock}$. Where $\gamma_{dock} = \psi_{AUV} + \theta$.

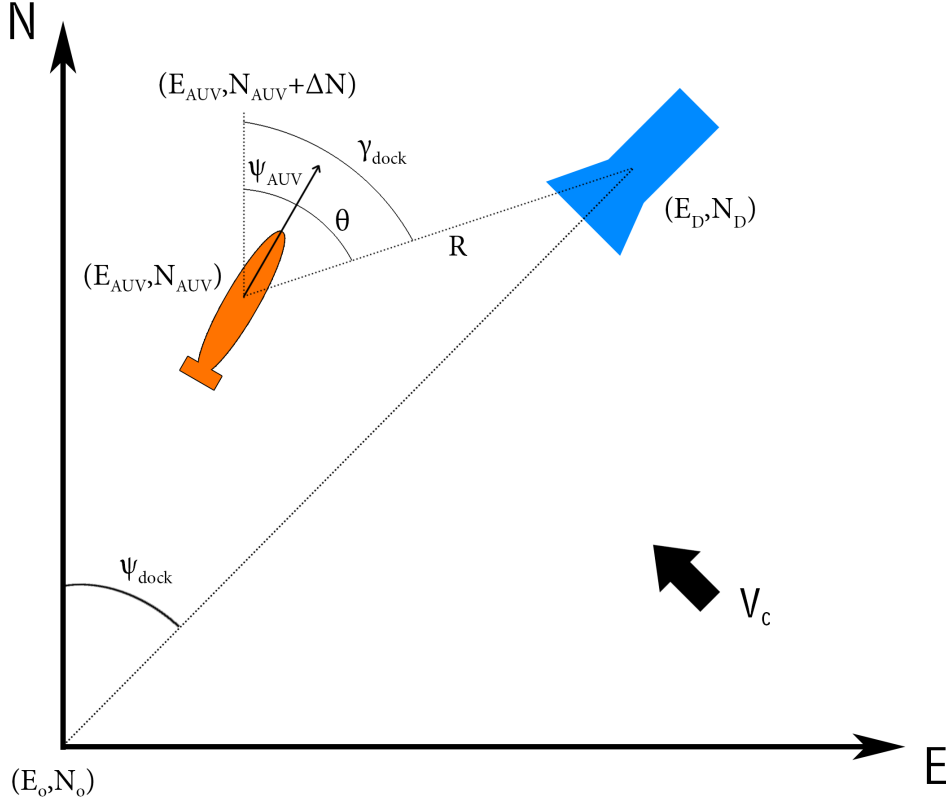


Figure 3.1: AUV homing towards docking station.

In the extreme case, the current forces the AUV to drive directly *towards* the current. Seen from above, a spiral pattern will emerge over time.

When the range, R , to the docking station is 200 meters, the algorithm decides if the docking phase should be initiated or if the AUV needs to reposition itself. If the initial cross-track error, y_{err} , is too large, the controller has not enough time to drive it to zero before reaching the docking station. Therefore this has to be checked. The allowable initial error is set to a cross-track error, y_{err} , of 50 meters. Figure 3.2 illustrates the above. If the AUV is not within the allowable limits, waypoints are generated towards a new initial start position for the homing phase. Only three waypoints (WPs) are needed to make the AUV reposition correctly with the switching criteria from Equation 2.22: One WP at the estimated dock position; one WP 300 meters down the dock center line; one WP 150 meters down the dock center line. When reaching WP3 a new homing attempt starts.

Note that $R = 200$ [m], $y_{err} = 50$ [m] and WP positions are verified through

simulations in Section 4.1 where the AUV is influenced by a conservative ocean current speed.

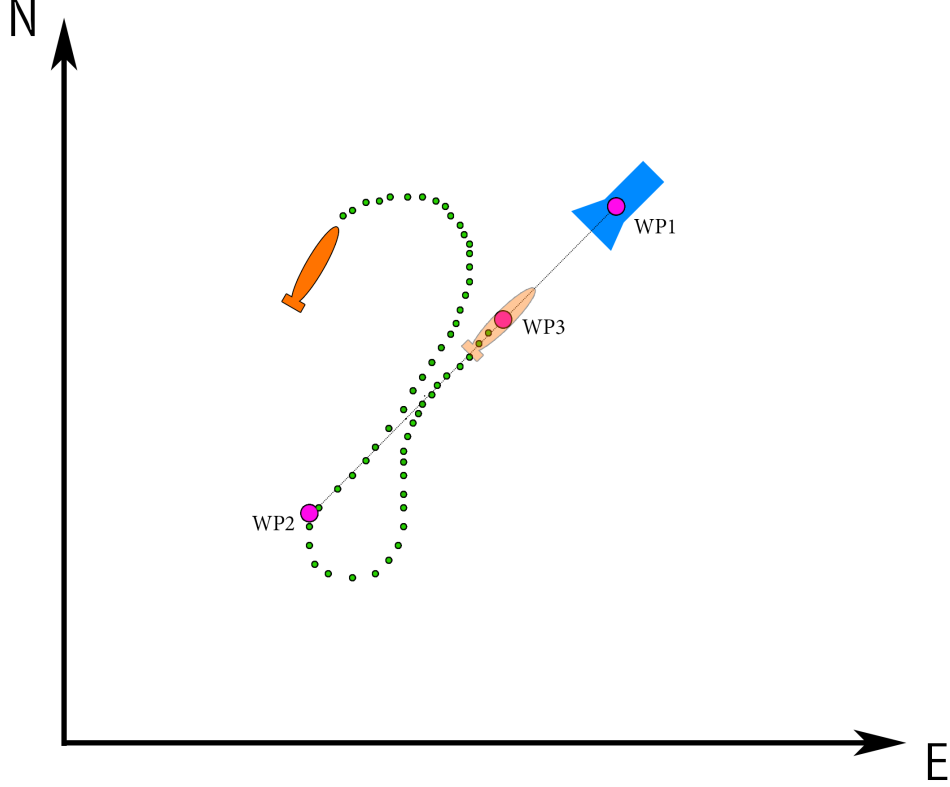


Figure 3.2: Waypoints are generated and LOS navigation initiated when the cross-track error is too large.

The position of the dock, (E_D, N_D) , is calculated as

$$(E_D, N_D) = (E_{AUV} + R \sin(\gamma_{dock}), N_{AUV} + R \cos(\gamma_{dock})) \quad (3.4)$$

and the cross track error as

$$y_{err} = R \sin(\beta) \quad (3.5)$$

with β

$$\beta = \gamma_{dock} - \psi_{dock}, \quad (3.6)$$

but can of course also be calculated by the same method used to find γ_{dock} . (E_o, N_o) may be replaced with any point along the dock center line, L (Figure 3.3).

$$\beta = \arccos \left(\frac{\vec{AB} \cdot \vec{BC}}{\|AB\| \|BC\|} \right) \quad (3.7)$$

with

$$\vec{AB} = \underbrace{(E_D, N_D)}_B - \underbrace{(E_{AUV}, N_{AUV})}_A \quad (3.8)$$

and

$$\vec{BC} = \underbrace{(E_o, N_o)}_C - \underbrace{(E_D, N_D)}_B \quad (3.9)$$

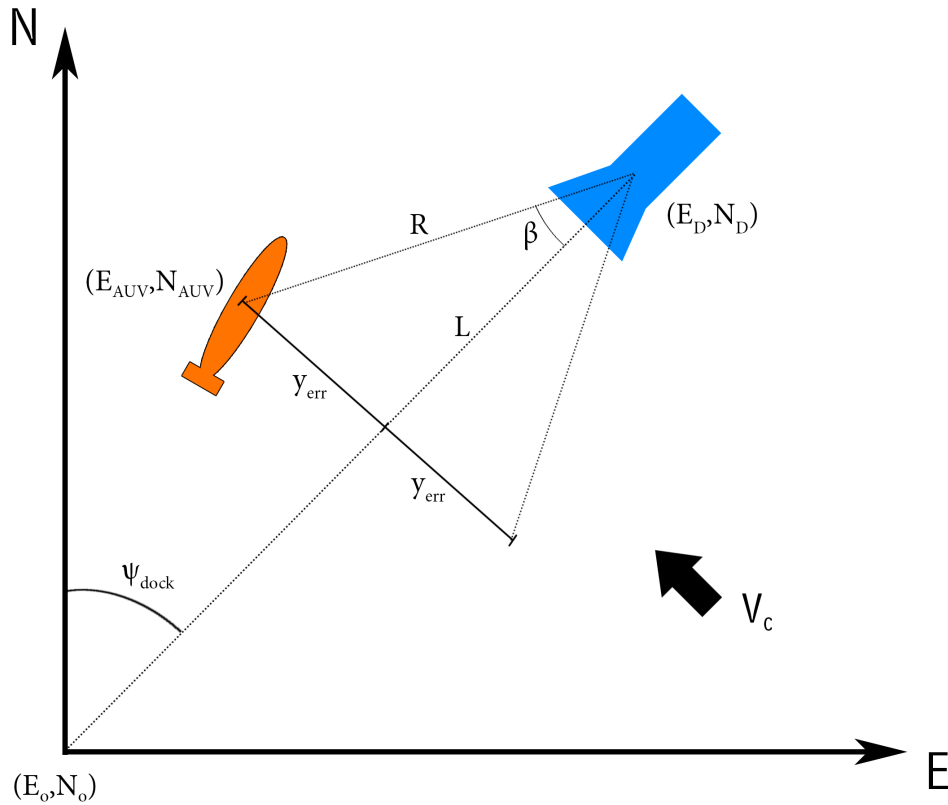


Figure 3.3: Initial AUV cross-track error, y_{err} .

Homing Controller

The setpoint γ_{dock} is used as a reference input to the REMUS 100 heading PID controller. The AUV heading, ψ_{AUV} , is used as a feedback term which gives the relative bearing, θ , as an error input to the PID controller. Figure 3.4 shows

this control approach, but is not representative to the actual layout of blocks in AUVSim. Due to the size and complexity of the block diagrams in the Matlab implementation, a choice was made to rather make a conceptual illustration.

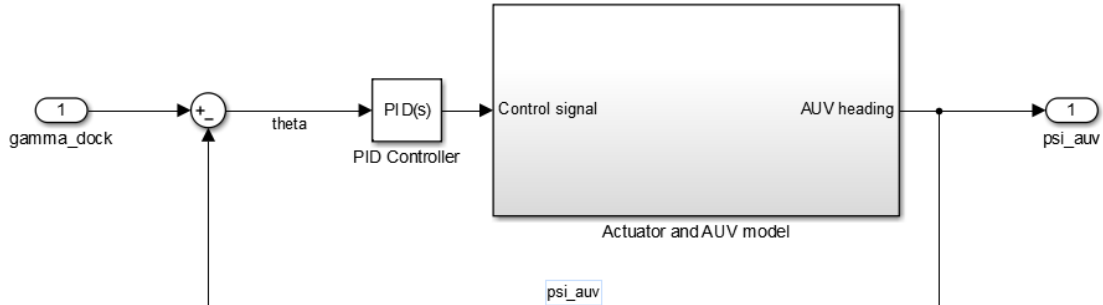


Figure 3.4: AUV homing control loop.

Docking Phase

The docking phase starts out by initiating the "docking controller". The forward speed is reduced to 0.8 m/s 15 meters from the dock in order to limit potential damages of the vehicle when docking. One has to consider the fact that the vehicle might miss the dock slightly and hit a frame or other supporting structure. The minimum speed for REMUS 100 is 0.25 m/s, but in a situation with current and movement in the water this speed most likely is too low. The DUSBL system is mounted on the front cone, without a crash cap, and an impact can potentially dislocate or damage the sensor.

The cross-track error is constantly calculated as $y_{err} = R \sin(\beta)$, regulated towards a zero reference, where β is given from Equation 3.6. When the forward speed is reduced significantly, to 0.2 m/s or lower and range measurements show 0.2 m or less, a successful docking is assumed. Preferably the AUV should start to listen for an acknowledge message sent from the dock, confirming a successful docking. On a simple "net dock", the crew can simply check the dock visually if sufficiently close to the surface. On more advanced docking systems a signal should be sent to the AUV to turn off the propulsion system.

If the relative bearing to the dock changes drastically, this means that the AUV has passed the dock without docking. It could also mean that it's actually docked and that the transponder is "behind" the AUV. If that is the case, the speed measurements would be used to confirm a successful docking. If no such confirmation is made, the algorithm would generate waypoints towards a new initial start position for the docking phase.

The algorithm assumes that the transponder is mounted in the middle of the dock. For a cone dock this is unpractical since the dock would screen the acoustic signal, but for a net dock it is very possible (Kukulya et al., 2010). Therefore, the algorithm has to adjust for transponder offset such that the signal *seems* to emerge from the center of the docking station.

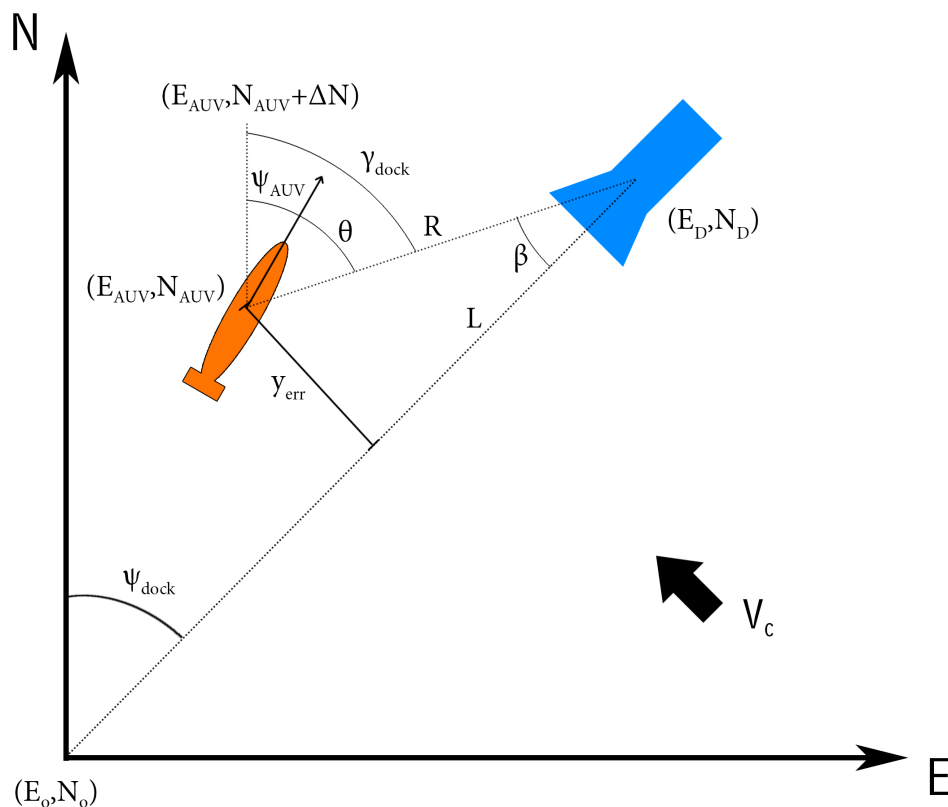


Figure 3.5: AUV in docking phase. The control goal is to minimize y_{err} .

Docking Controller

The docking controller is structured as the cascaded controller in Bellingham et al. (2008). An outer PID loop controls the cross-track error, y_{err} , and an inner loop controls the relative bearing to the dock, θ , towards a zero reference (Figure 3.6). Figure 3.5 shows the AUV in the docking phase. This is according to the principle of regulating a faster inner loop dynamics by regulating a slower outer loop dynamics (Balchen et al., 2003). Successive loop closure is also extensively used in Beard and McLain (2012) for autopilot design for small unmanned aircraft. As in Figure 3.4, the figure below is conceptual since the actual Simulink blocks is hard to depict properly.

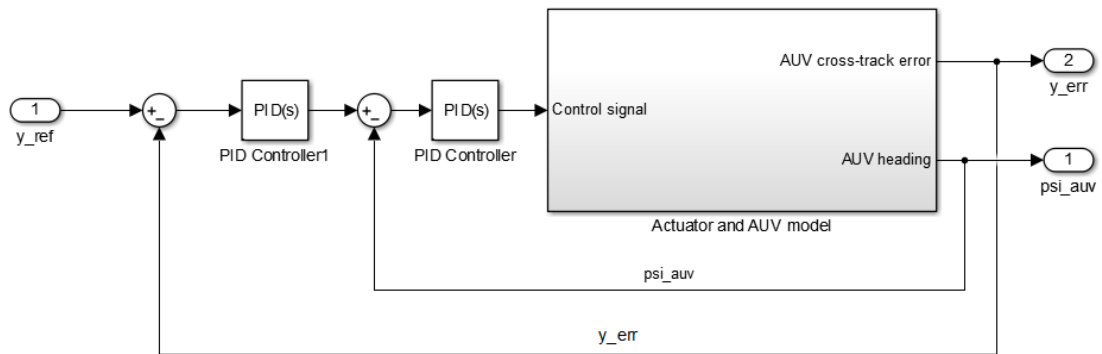


Figure 3.6: AUV docking control loop.

The relative depth to the transponder is found by the vertical angle measurement from the DUSBL sensor sent as a reference to the REMUS 100 depth controller. Since the dock altitude is known in advance, REMUS 100 could also be set to operate at this altitude (done when using the LBL sensor).

3.1.2 LBL aided Homing and Docking

This section explains the homing and docking process when using a LBL system with a single transponder. Since the AUV only knows its range from the transponder on the dock, the PF from Chapter 2.5 is used for relative positioning.

Proposed Algorithm

Algorithm 2: LBL Homing and Docking

```

1 if No LBL signal received yet then
2   └ Dead reckon to assumed transponder position
3 if LBL signal received then
4   Start homing phase:
5     Calculate position from PF;
6     Generate waypoints;
7     Start LOS algorithm to first waypoint;
8     Set reference depth equal to the dock depth;
9   if No LBL signal then
10    └ Use position from PF
11  if At first waypoint then
12    Start homing phase:
13      Start LOS algorithm between waypoints;
14      if Range to transponder  $\leq 15$  [m] then
15        └ Set AUV speed to 0.8 [m/s]
16      if Range to transponder  $\leq 0.2$  [m] and AUV speed  $\leq 0.2$  [m] then
17        └ Docking successful;
18          └ Stop algorithm
19      if Passed last waypoint and AUV speed  $\geq 0.8$  then
20        └ Docking unsuccessful;
21        └ Generate waypoints to new initial docking phase position

```

Homing Phase

In normal operation, chances are that the AUV should operate well within the range of the LBL transponder (several kilometers). If out of range, the AUV travels towards the preprogrammed dock position. The dock houses the transponder as in the case of DUSBL homing/docking. By assuming this location is the correct one, the PF gives the AUV position relative to this position. The initialization of the PF starts when the first LBL measurement is received and allows the internal AUV position estimation to have drifted within a square of ± 50 meters. These metrics are easily adjusted, but a tighter area yields accurate position estimates faster. It is the mission type that is governing. E.g. missions with DVL bottom track at all times could have even tighter initial margins. An important feature of the PF is that it is not dependent on measurements to give

a position estimate. The position estimate is just improved upon every received measurement, causing particle cloud condensation.

WPs are generated from the assumed dock position and the homing phase starts out by navigating to the first one. The homing phase is illustrated in Figure 3.7 as the green dots towards WP1. Here, an important difference between DUSBL and LBL homing emerges. The LBL sensor can send/receive from 360° , thus does not need to point directly towards the dock to receive the best quality signal. This means that the AUV can have a crab angle in the homing phase and effectively counteract any drift by ocean currents.

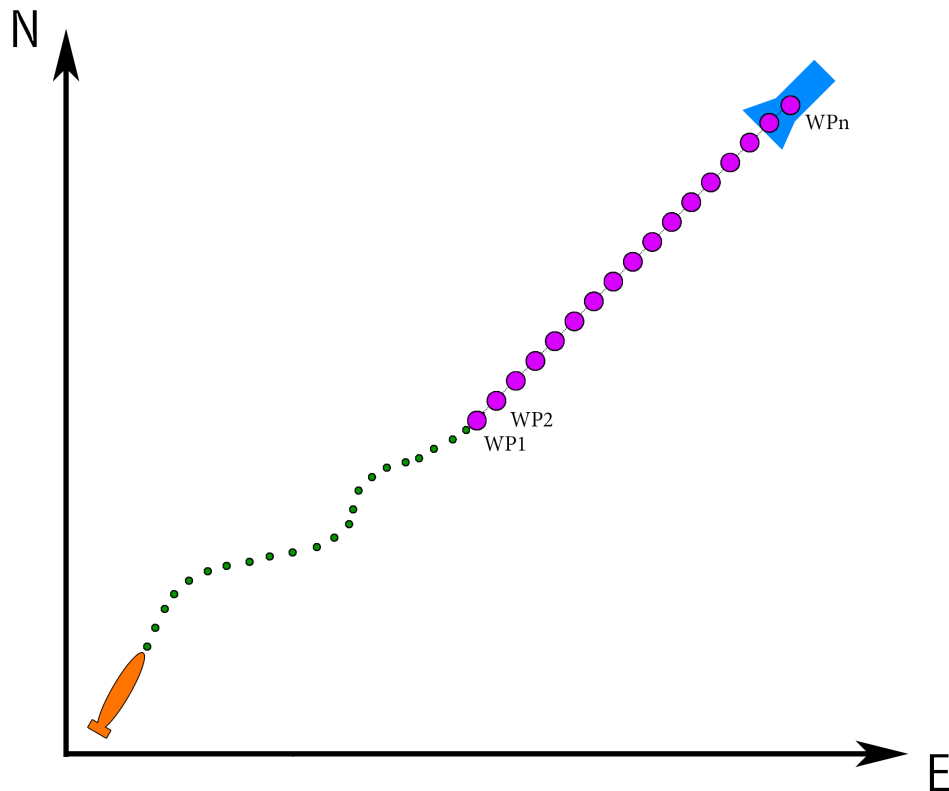


Figure 3.7: AUV homing to WP1 followed by a docking sequence.

Docking Phase

The docking phase is analogous to the homing phase, but with WPs every 10 meters the last 200 meters to the dock (the purple dots in Figure 3.7). This forces the AUV to follow the dock center line.

Homing and Docking Controller

The homing and docking controller is the saturating control law from Section 2.4, Equation 2.21, but is for convenience reproduced below:

$$\chi_d(e) = \alpha_k + \arctan \left(-K_p e - K_i \int_0^t e(\tau) d\tau \right) \quad (3.10)$$

The REMUS 100 altitude controller has previously been shown capable of regulating the AUV depth with error just over 0.5 meters (Holsen, 2015). Therefore the REMUS 100 is set to operate at the same altitude as the docking station. Compared to DUSBL docking, this altitude has to be known in advance. The big advantage with this approach is that the PF only needs to consider positions in two dimensions - in the dock plane.

3.2 Simulator Environment

AUVSim is a simulator built in MATLAB and Simulink which can be used to simulate the behaviour of the REMUS 100. Before implementing a software solution directly on the REMUS 100, a MATLAB implementation of the same solution allows for easier understanding of the vehicle behaviour and quick testing of new ideas. This is because writing mathematical expressions and designing control systems is much quicker in MATLAB and Simulink than it is in any C++ environment. Also, there is no need for cumbersome simulation setup and re-compiling between each change in the code. This chapter provides a description of the simulator environment used for all simulations. The results of the simulations are presented in Chapter 4.

3.2.1 Simulator Overview

Figure 3.8 shows the main blocks of AUVSim (yellow blocks) and the homing/docking block for simulating homing and docking with the DUSBL and LBL underwater acoustic systems. The design and development of this block is one of the main contributions in this thesis and consists of many subsystems and functions as shown in Figure 3.9. For the sake of consistency and easy comparison of results, the docking station is positioned at (400,400) [m] at a depth of 30 [m] and rotated 45° (the entry points towards (0,0) [m]). 0.5 [m/s] current is affecting the AUV from 30° North-East and creates a horizontal and vertical perturbation on the AUV. The Norwegian Coastal Current (NCC) has speeds that typically lay between 0.2-0.5 [m/s] (Havforskningsinstituttet, 2013). The "worst-case" scenario of 0.5 [m/s] is the most interesting when checking the robustness of the solution. The simulation starts with the AUV in homing mode. Time-of-flight (TOF) in modern underwater acoustic systems is typical in the range 1-3 seconds, dependent on the distance and speed of sound in water. The simulations have a constant TOF of 1 second and is realistic in the simulation scenario used. It could actually be considered conservative since it is held constant despite the fact that the AUV is moving closer to the dock, hence reducing the TOF. Also, the simulator makes sure that the AUV is only able to "receive" DUSBL measurements if the dock is within $\pm 45^\circ$ of the AUV's nose cone.

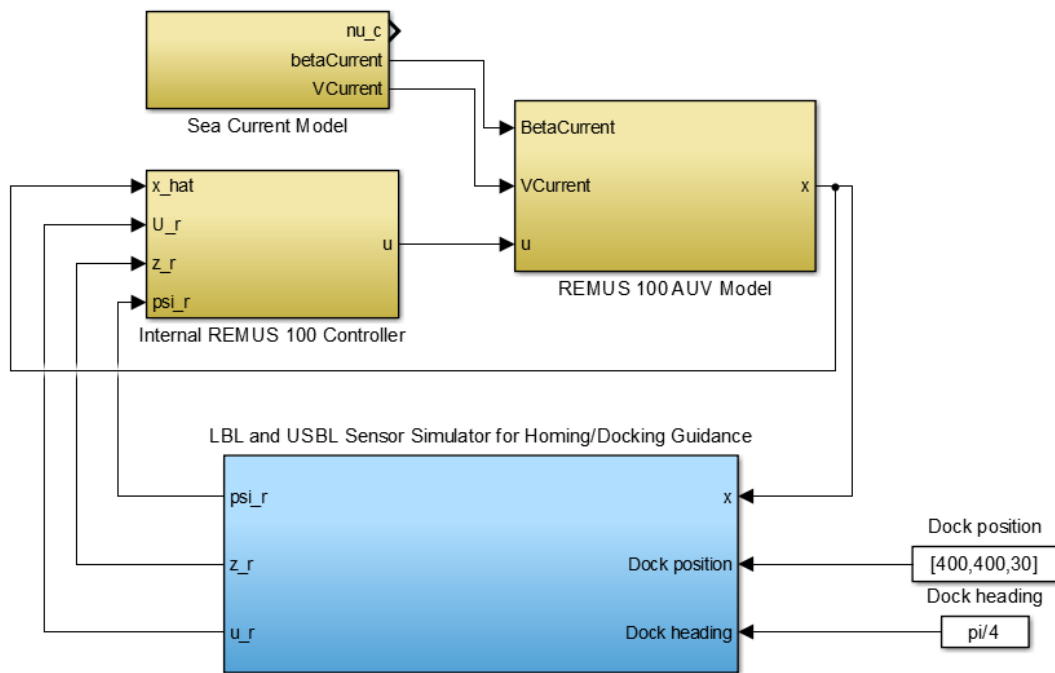


Figure 3.8: Overview of main AUVSim Simulink blocks, included the homing/docking block.

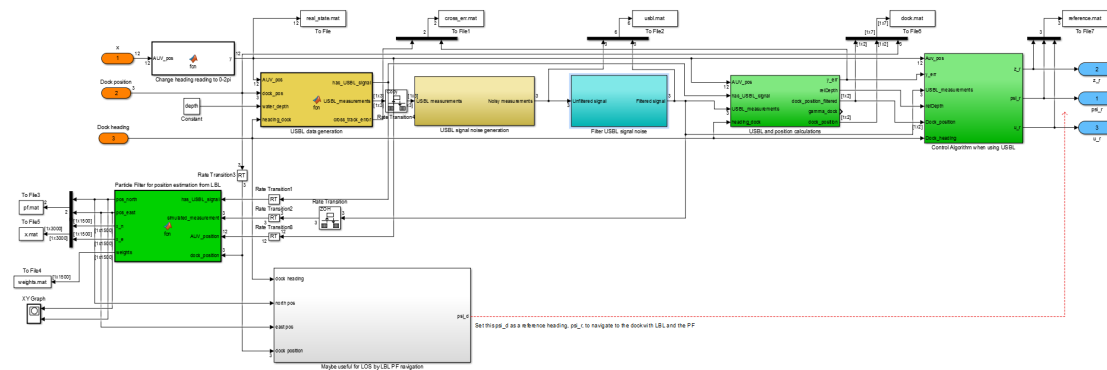


Figure 3.9: Inside the blue homing/docking block (blue) in Figure 3.8.

A larger version of Figure 3.9 is found in Appendix C, Figure C.1, C.2 and C.3. The homing/docking block simulates both a DUSBL signal and a single LBL signal (the first yellow box from the left in Figure 3.9). These measurements are then influenced by noise (the second yellow box from the left in Figure 3.9)

equal to what was found after practical DUSBL tests conducted by Ph.D. candidate Albert Sans Muntadas (IMT, NTNU) and Petter Norgren (IMT, NTNU) at Trondheim Biological Station, December 2015. The noisy signals are then filtered (turquoise block in Figure 3.9) and sent to a calculation block (second green block from the right in Figure 3.9) that calculates the inputs (cross-track error, relative depth and dock position) to the control algorithm block (green block to the right in Figure 3.9). This block calculates the necessary reference signals used by the internal REMUS 100 controller block in Figure 3.9. The green block and grey block in Figure 3.9 are the PF and LOS guidance blocks, respectively. The PF block uses the filtered DUSBL range signal as an input and should be pretty close to what a filtered LBL range signal should look like, due to the similarity of the two acoustic systems. The estimated position from the PF is then used in the LOS guidance block to generate a heading reference signal ψ_r . The change between DUSBL and LBL homing/docking is simply done by manually change the origin of ψ_r in the Simulink diagram.

3.2.2 Measurement Noise and Filtering

The DUSBL measurements are calculated as a function of the dock position and the AUV position. The real AUV position is calculated in AUVSim and the dock position is predetermined. Knowing these two things enables the calculation of a simulated DUSBL signal (range, horizontal and vertical angle). Only the simulated signal is used in the algorithm to estimate the dock and AUV position. The variance found by Muntadas in range and horizontal angle measurements were $\sigma_{range}^2 = 2.89$ [m] and $\sigma_{hor.angle}^2 = 0.015$ [rad] respectively, when measurements were assumed to be normal distributed. Zero-mean Gaussian noise with the correct variance was added in to the simulated signal in Simulink. Also, wild points were added every few seconds to the signal to simulate the behaviour of a real underwater acoustic system. The noise is held constant and not a function of range. This is conservative and one might experience better results when closer, due to a better Signal-to-Noise Ratio (SNR).

Figure 3.10 compares the filtered and unfiltered range and angular measurements. Remember that the DUSBL range measurements also works as LBL range measurements. The figure has a large scale and shows that the wild points are effectively removed. Figure 3.11 gives a more detailed view of the filtered and unfiltered signals.

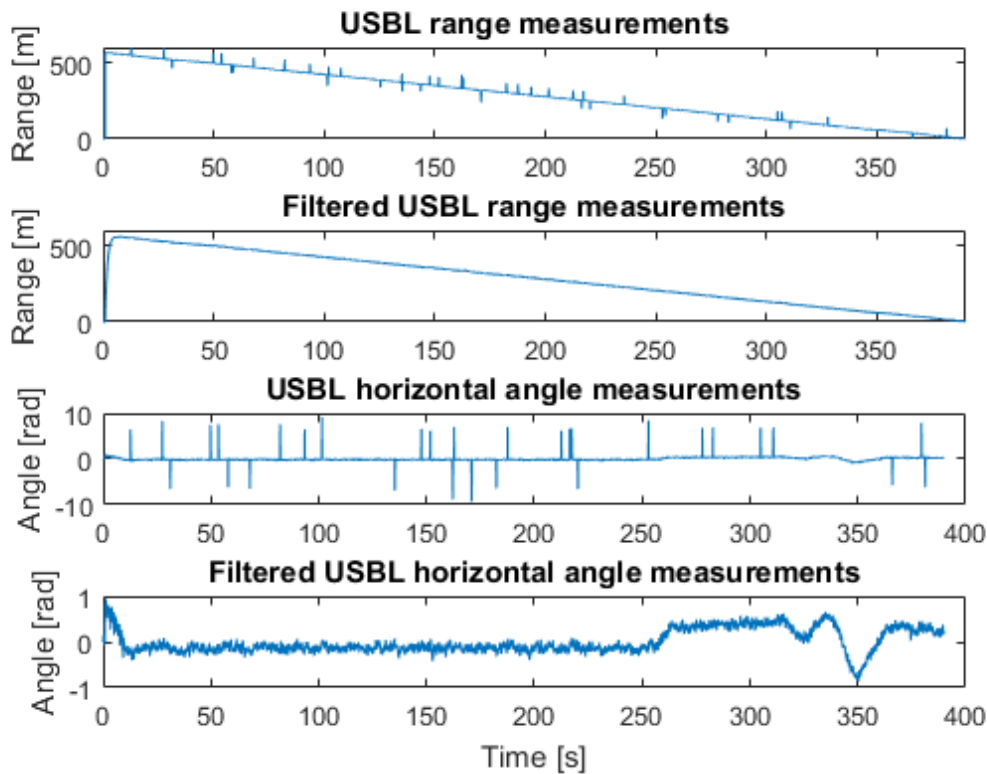


Figure 3.10: Filtered and unfiltered DUSBL measurements.

The high frequency noise in the range measurements is removed by a discrete low-pass filter with time constant $\frac{1}{\omega} = 1$ and the wild points with a rate limiter set to $[-3,3]$. Similarly the noise in the horizontal angle measurements is removed with $\frac{1}{\varepsilon} = 0.01$ and $[-0.7,0.7]$. The signal filtering adds a delay on the system, which is important to consider in systems that are supposed to run in real-time. This delay is clearly visible in Figure 3.11 in the DUSBL range measurements. Here the delay from a wild point (peak) to the top-point of the filtered signal is close to 1 second. In the angle measurements the delay is much smaller due to the smaller time constant. Too large delay would cause the system to be unstable and a trade-off between a fast system response and effective noise filtering has to be done. The values above was found by numerous simulations and gave satisfactory results.

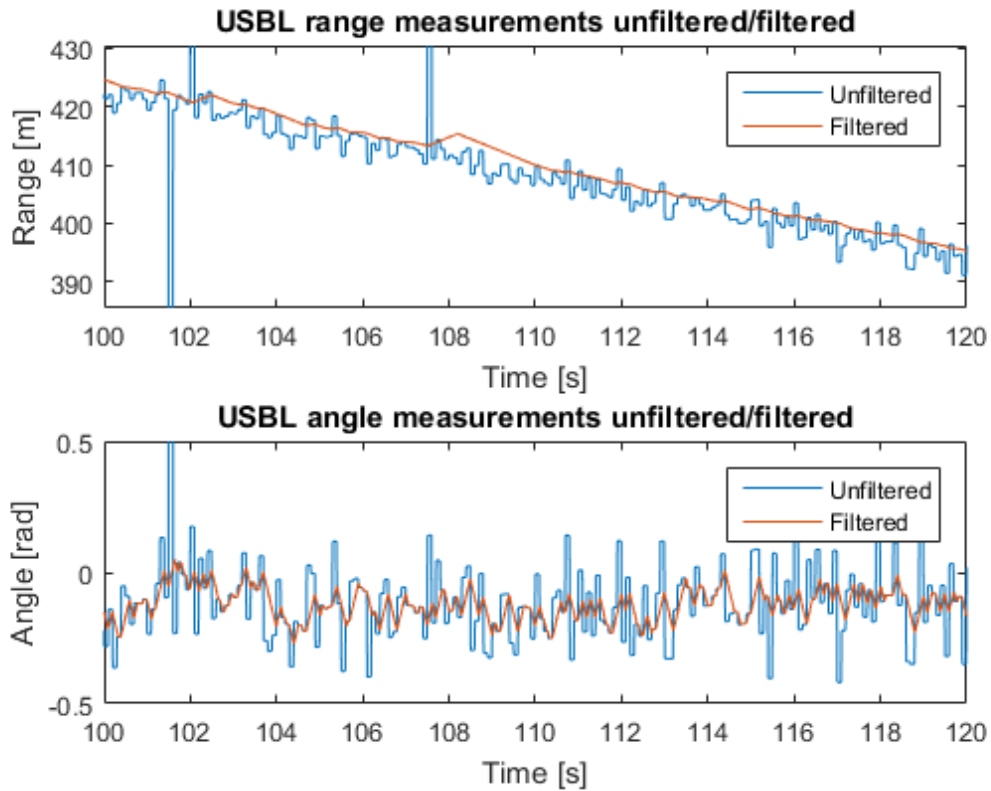


Figure 3.11: Detailed view of the filtered and unfiltered DUSBL measurements.

Even though the signal is filtered there is still a noisy component that is not possible to remove without sacrificing stability. The influence this component have on the system is important. The dock position is calculated directly from the DUSBL measurements and should not vary significantly. This is particularly important since the calculation of the cross-track error depends on the calculated dock position. Figure 3.12 shows the unfiltered and filtered and dock position as well as the error during a typical homing and docking phase. The error is within a few meters and further investigation is needed to decide if this is good enough to perform docking.

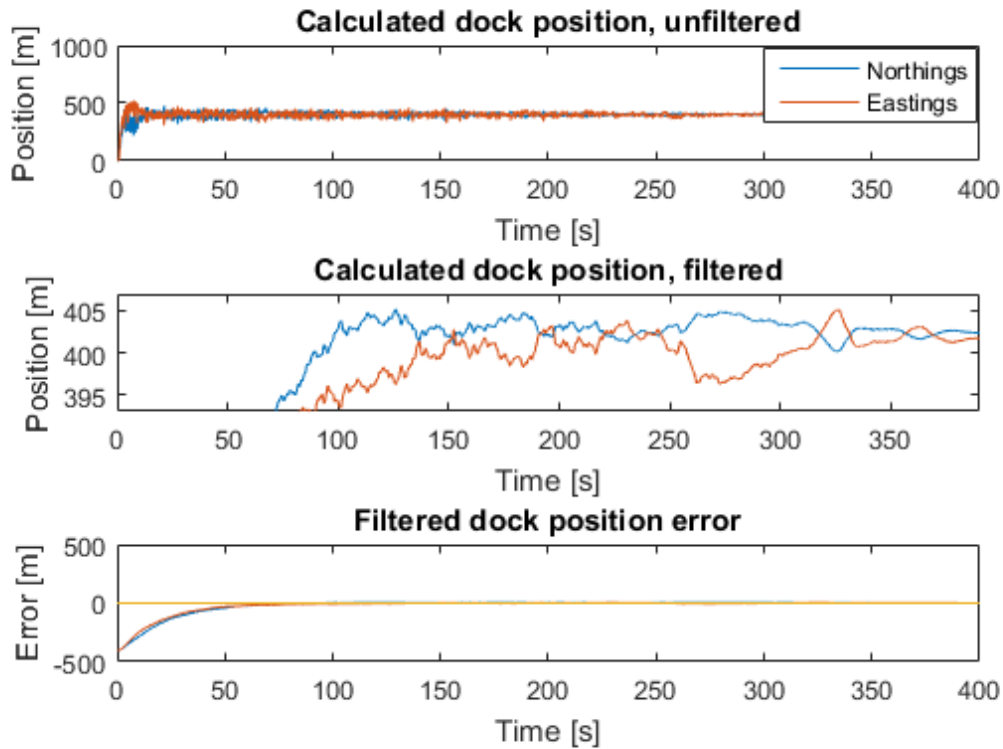


Figure 3.12: Calculated dock position from DUSBL measurements.

The estimated cross-track error is also heavily influenced by noise. As a result the input to the cascaded controller in Figure 3.5 is noisy too. The contribution of noise in the cross-track error calculation is a function of range, hence decays over time when the AUV moves closer, as shown in Figure 3.13. Intuitively this is because errors in an angle measurement spans a larger distance when the range is longer.

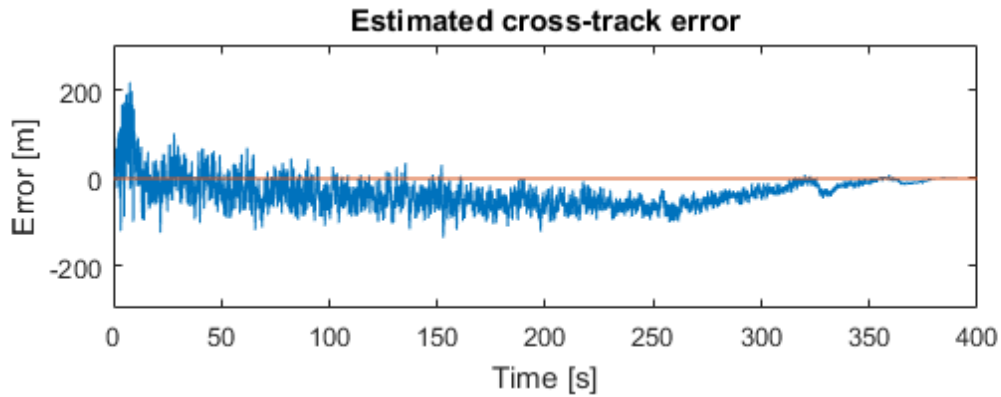


Figure 3.13: Cross-track error from dock center line calculated from DUSBL measurements.

3.2.3 Internal Remus 100 Controllers

The internal REMUS 100 controllers in AUVSim are made to behave close to the real system and is verified to do so in (Holsen, 2015). These controllers are used to steer the AUV according to the reference signals from the homing/docking block. The controllers are discrete PID and PI controllers with gains as in Table 3.1, 3.2 and 3.3.

Table 3.1: REMUS 100 heading controller PID gains.

Variable	Gain
P	1.20
I	0.20
D	1.50

Table 3.2: REMUS 100 depth controller PI gains.

Variable	Gain
P	0.100
I	0.010

Table 3.3: REMUS 100 speed controller PI gains.

Variable	Gain
P	300
I	100

3.2.4 Tuning the Particle Filter

The PF can basically be tuned with four parameters, some of which are held constant during all simulations. The ones which vary are mentioned in Chapter 4 together with the corresponding results. Below is an overview of the parameters, plus the resample methods, that decide the performance and accuracy of the PF.

- Increase/decrease the number of particles
 - 1500 was the number of particles that gave satisfactory results and which is constant for all simulations. Appendix B contains the results when experimenting with other numbers of particles in the PF.
- Change the initial conditions
 - The initial conditions (square of ± 50 [m]) are held constant in all simulations as discussed in Section 21.
- Modify the noise term in the prediction step
 - The noise added is represented by zero mean normal numbers with standard deviation, σ_{noise} , as presented in Chapter 4.
- Modify the likelihood function, Equation 2.25, by changing σ_r
 - σ_r is the (assumed) standard deviation of the range measurements and depends on whether the filtered or unfiltered measurements are used.
- Change the resample method
 - Multinomial, residual and systematic resampling are considered in this thesis.

3.3 LBL Test in Trondheimsfjorden

To check the validity of the simulations, it is important to test the performance and quality of the LBL sensor on REMUS 100. A test was conducted 2. June 2016, in Trondheimsfjorden in the area around Munkholmen, with Petter Norgren. The objective was to make REMUS 100 collect range measurements from a single transponder, at different ranges. The range data could then be used to determine LBL measurement rates, measurement noise, measurement wild points and potential signal loss.

The conditions were partially cloudy, 15-18°C air temperature, 2 [m/s] wind, current between 1-3 [cm/s], water temperature 12°C and 0 [m] waves. The whole mission with mobilizing and demobilizing took seven and a half hours, from 07:30 to 15:00. The operating vessel was a rib borrowed from the NTNUI diving group.



Figure 3.14: tl. REMUS 100 and computers with VIP software; tr. REMUS 100 in the water; bl. boat in transit; br. gear on trolley.

No unexpected events happened, but we noticed that even in the near optimal conditions it was important to launch the AUV from the side of the boat facing towards the wind. Else, the boat would tend to drift over the AUV and potentially damage the fins.

First, a transponder was placed at 63N27.305 10E23.664, as indicated by the plus symbol in Figure 3.15. The depth was roughly 27 meters and the transponder fastened approximately 12 meters above seabed on a rope between a weight and a buoy. The position was found by a box-in test, where we traveled over the transponder position several times to find the surface position with the smallest distance to the transponder. The distance was measured by a towfish that could send acoustic pings to the transponder and read out the distance.

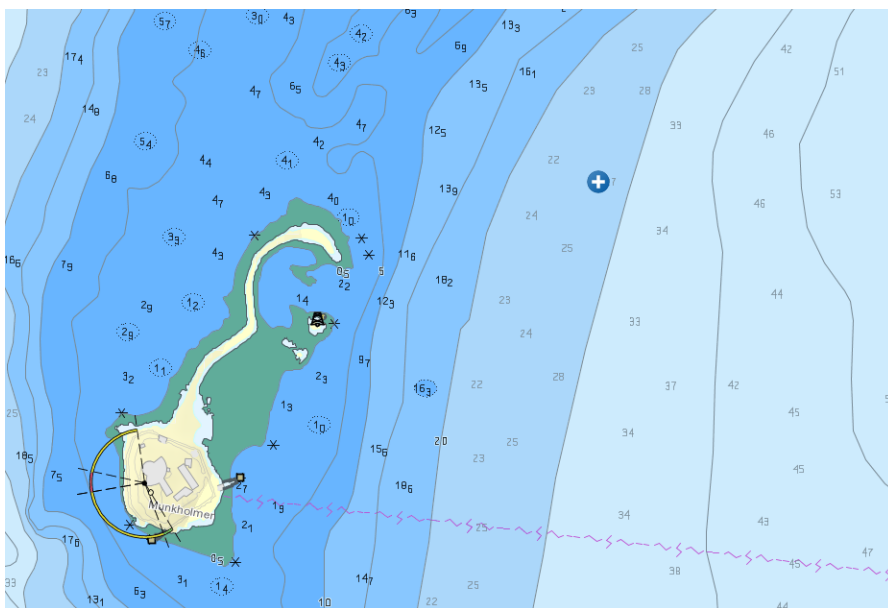


Figure 3.15: The plus symbol indicates the transponder position after box-in.

Three missions were conducted to take LBL measurements in different scenarios and ranges from the transponder. The missions were programmed in the REMUS VIP software and transferred to REMUS 100 before mission start. Normally at least two transponders are needed to make up a LBL position fix. To overcome this and run with just one transponder, two transponders were programmed in the mission file, but only one launched in the water. The three missions, *Line*, *Box #1* and *Box #2*, are explained below and the transponder position was never changed during the missions.

The *Line* mission makes REMUS 100 travel towards and past the transponder in a straight line, starting 500 meters out. It then turns and follows the line back.

The goal with the mission is to map how the LBL measurements behave when REMUS 100 travels closer and closer to the transponder.

The *Box #1* and *Box #2* missions makes REMUS 100 travel in a square around the transponder positions. The squares have sides that are 200 meters and 400 meters long, for the two mission respectively. The goal with the missions are to check the consistency of the LBL measurements when keeping a limited distance to the transponder over time.

The results from the three missions are presented in Section 4.4.

Chapter 4

Results

This chapter presents the results from the simulations and from the LBL test. Since the results are dependent on the tunable parameters in the controllers, the final results are presented together with the corresponding parameters.

4.1 DUSBL Homing and Docking

The homing controller is simple and does not need any tuning. The AUV simply points the nose in the calculated direction of the dock with the internal REMUS 100 heading controller. The internal REMUS 100 controllers are used for speed and depth keeping.

The docking controller consists of an inner- an outer-loop PID controller (Figure 3.6). The inner-loop PID controller is the internal REMUS 100 heading controller, with gains as in Table 3.1. The outer-loop PID controller gains were found by manual tuning (Table 4.1):

Table 4.1: DUSBL docking controller PID gains.

Variable	Gain
P	0.04
I	0.00054
D	0.00010

The docking controller is able to zero a cross-track error of 60 meters when 200 meters from the dock. Despite this, the algorithm uses 50 meters at the maximum cross-track error before repositioning is needed. This will add an error

margin and increase chances of success. Figure 4.2 shows the real, actual cross-track error during homing and docking phase, where the cross-track error increases during homing (because of drift) and is reduced during docking. At the dock the error is below 30 centimeters. The AUV trajectory is plotted in Figure 4.1.

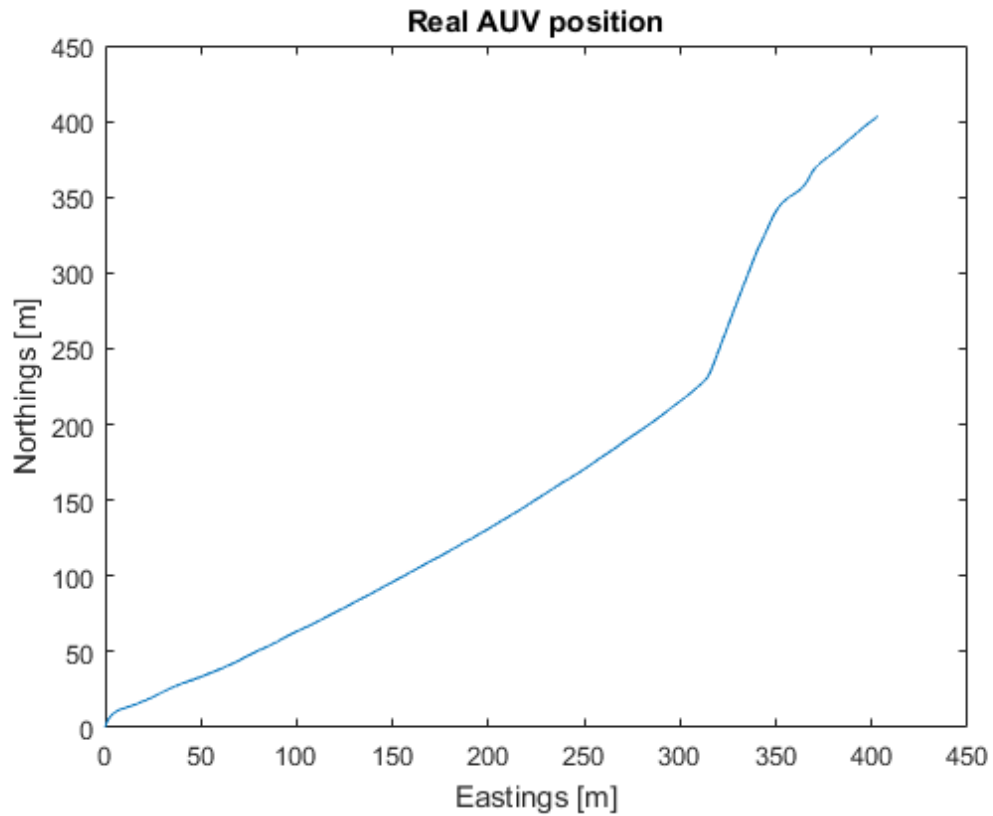


Figure 4.1: AUV position during DUSBL homing and docking.

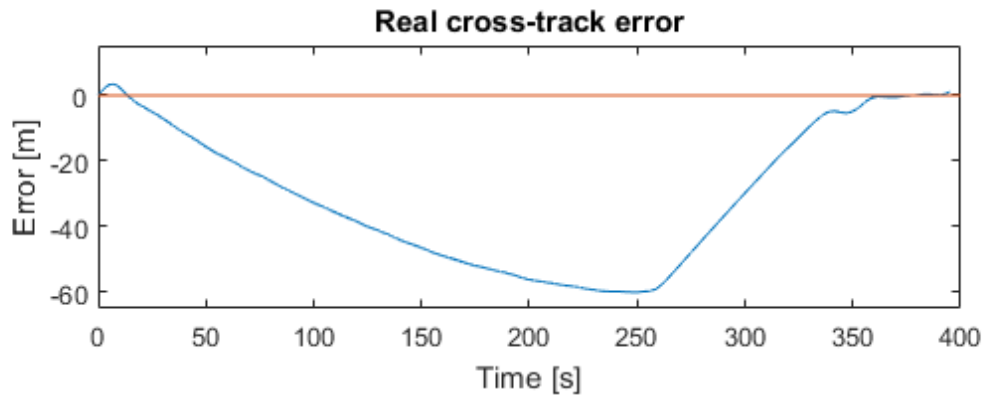


Figure 4.2: The real cross-track error during DUSBL homing and docking.

The LOS controller is found to work well with the gains:

Table 4.2: Gains for the LOS controller.

Variable	Gain
P	0.05
I	0.0001

If the cross-track error is too large after homing, the AUV needs to reposition itself and a new docking sequence starts after passing the last WP. The trajectory looks like the one in Figure 4.3. In the figure one can notice that the AUV starts to slowly oscillate when the docking PID controller is started. The oscillations are dampened long before the entry of the docking station.

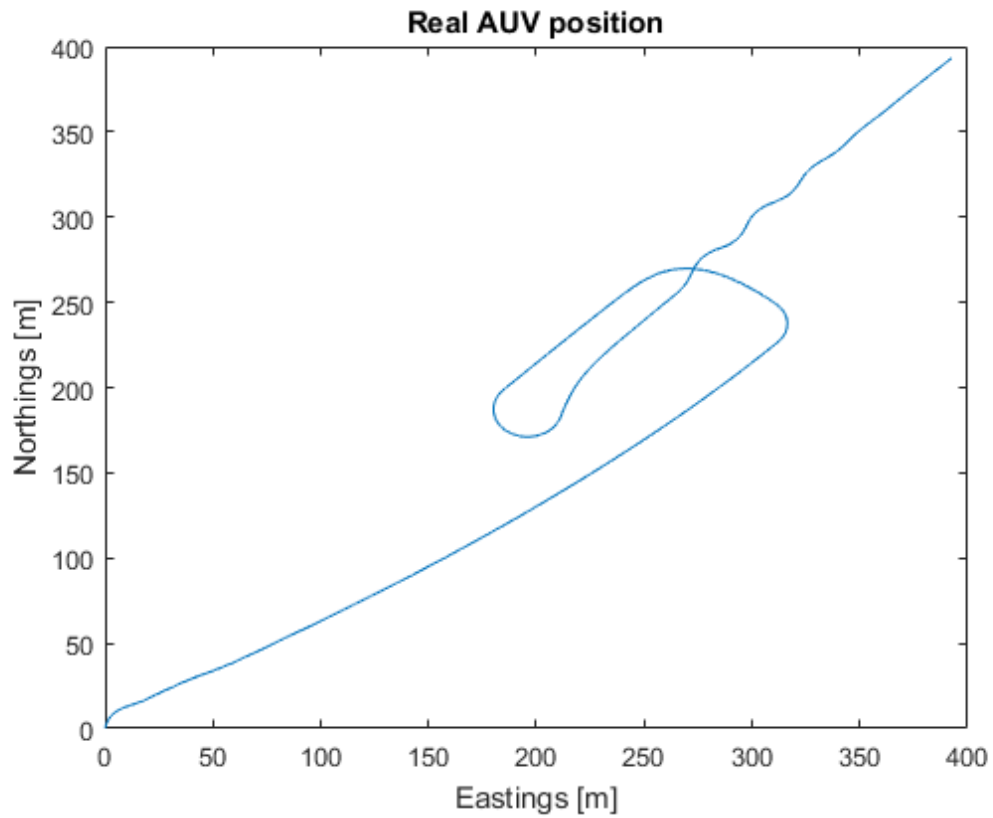


Figure 4.3: AUV trajectory during repositioning and docking

The corresponding plot of the real cross-track error is given from Figure 4.4. Once again, the error is below 30 centimeters at the dock. The docking phase, after repositioning starts 150 meters from the dock and seems to work well because of the better initial position of the docking phase.

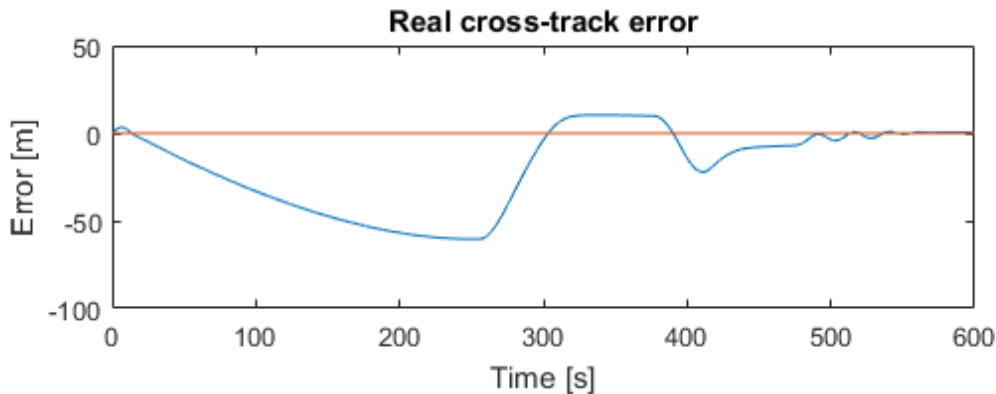


Figure 4.4: The real cross-track error during DUSBL homing and docking when repositioning is needed.

4.2 LBL Homing and Docking

This section presents the results when using different resampling strategies in the PF when performing homing and docking. The LOS controller used is the same as in Section 4.1.

4.2.1 Unfiltered and Filtered Measurements

Before letting the PF steer the AUV with the LOS algorithm, it is important to see how well it manages to estimate the position. The PF is ran in parallel with the DUSBL homing/docking algorithm and the position estimated by the PF is compared to the real position.

First, it is interesting to see how the PF deals with a noisy, unfiltered signal with wildpoints (Figure 3.11). The result is presented in Figure 4.5 and proves that the PF tracks the real position reasonably well. The position estimates are far from accurate enough to be used in a docking scenario, but are usable for homing and non-critical navigation. Because the signal was unfiltered, $\sigma_r = 1.7$. The standard deviation of the added noise was set to $\sigma_{noise} = 0.15$ and the resampling method, systematic.

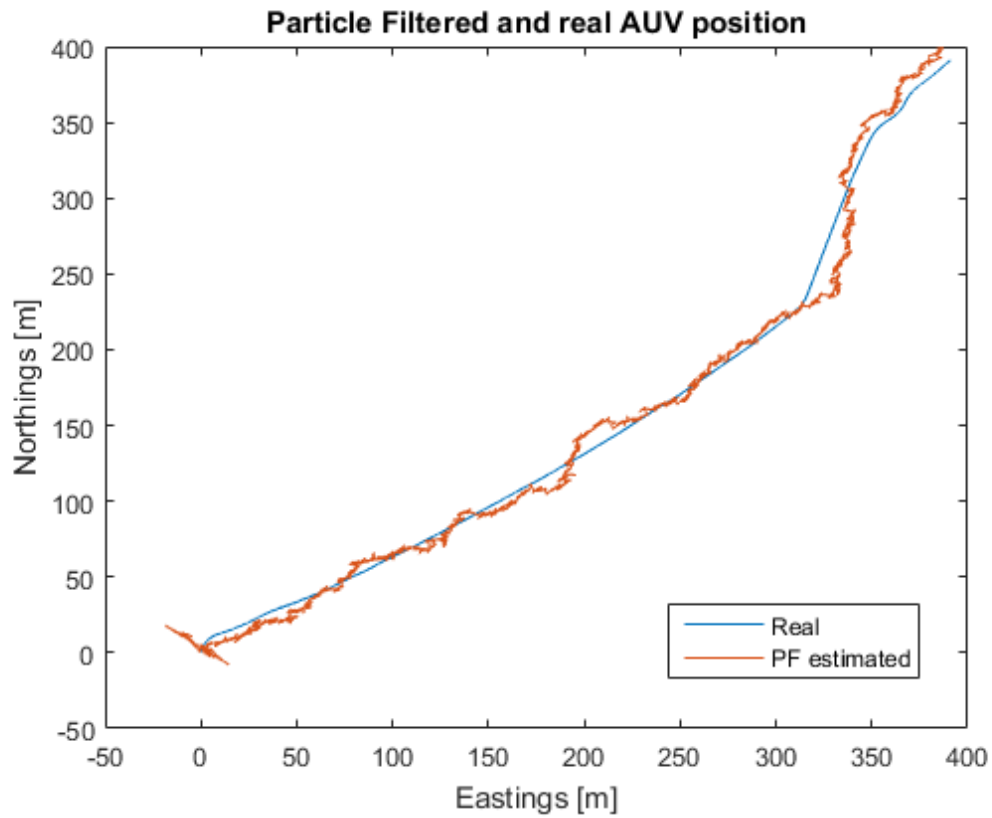


Figure 4.5: Estimation of the position with the PF from unfiltered LBL measurements.

When switching to the filtered measurements, σ_r obviously had to change. $\sigma_r = 0.5$ was found to give the best results, together with $\sigma_{noise} = 0.15$ and systematic resampling. Figure 4.6 shows significantly better results with the filtered signals.

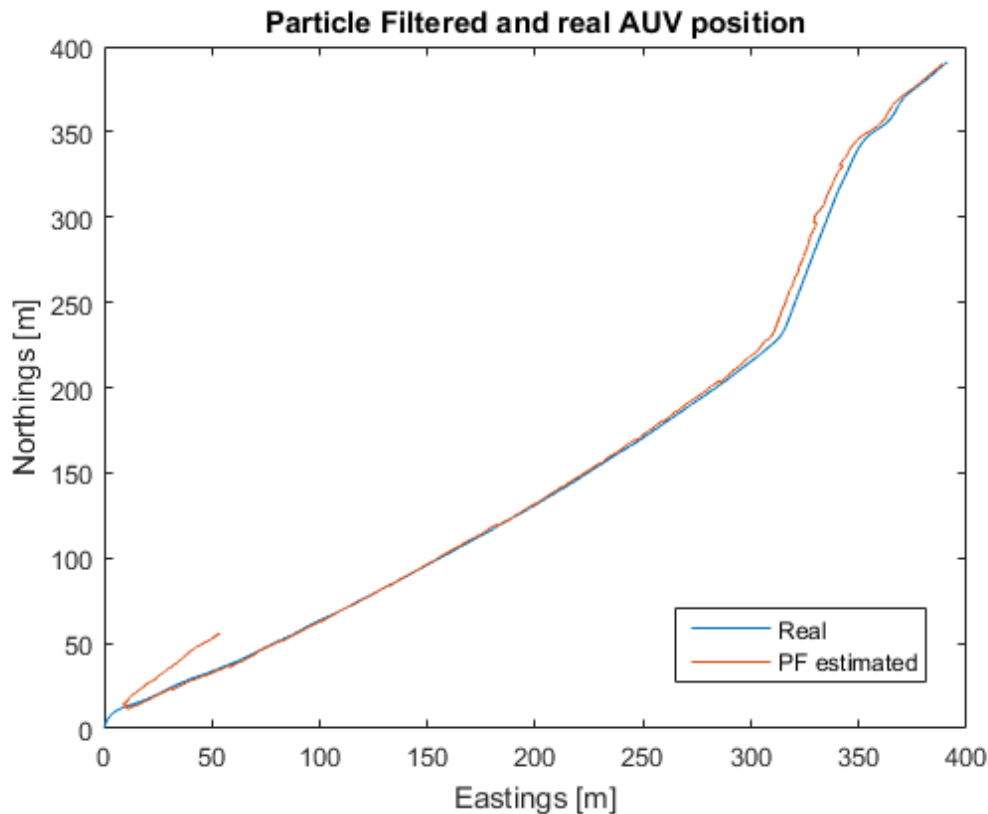


Figure 4.6: Estimation of the position with the PF from filtered LBL measurements.

The following sections capitalize on the good results obtained above, with the PF combined with the filtered measurements. Different resampling strategies are used with the homing/docking algorithm from Section 3.1.2 and the results are presented. σ_r and σ_{noise} are held at the values found above, which allows us to directly compare the resampling strategies against each other, in this particular scenario.

4.2.2 Multinomial Resampling

Figure 4.7 clearly shows the initial particle positions as a square around origo. As time goes, the particles converge to circles along the path. The PF estimated AUV position is represented by the blue line. A particle cloud is generated for each time step, so for obvious reasons only a few of them are plotted. Since each particle contains a likelihood weight, the weighted mean of one such cloud gives the estimated position.

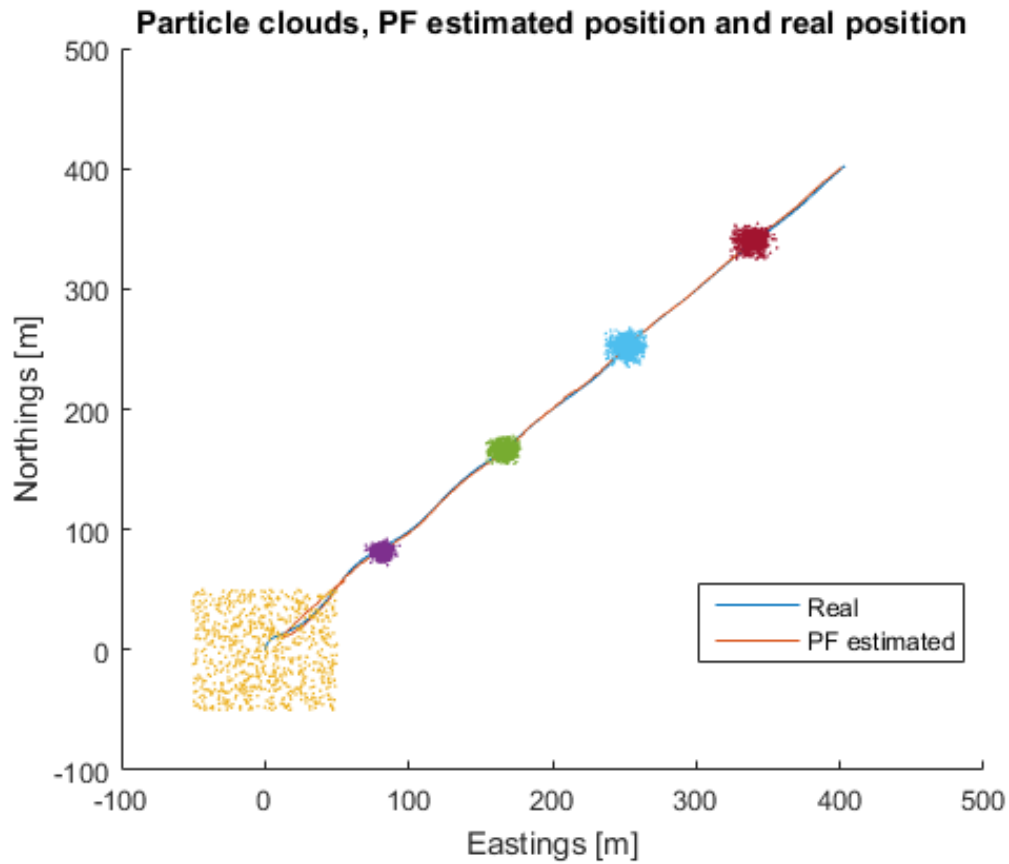


Figure 4.7: The estimated position overlays the actual position well.

The error between the real position and the estimated position used by the control system (LOS algorithm) is given in Figure 4.8. It is important that the error is as small as possible so that the control system has greater chances to steer the AUV into the dock. The simulation is stopped when the AUV reaches the docking station and the cross-track error, from Figure 4.9, is read to be approximately 1 meter.

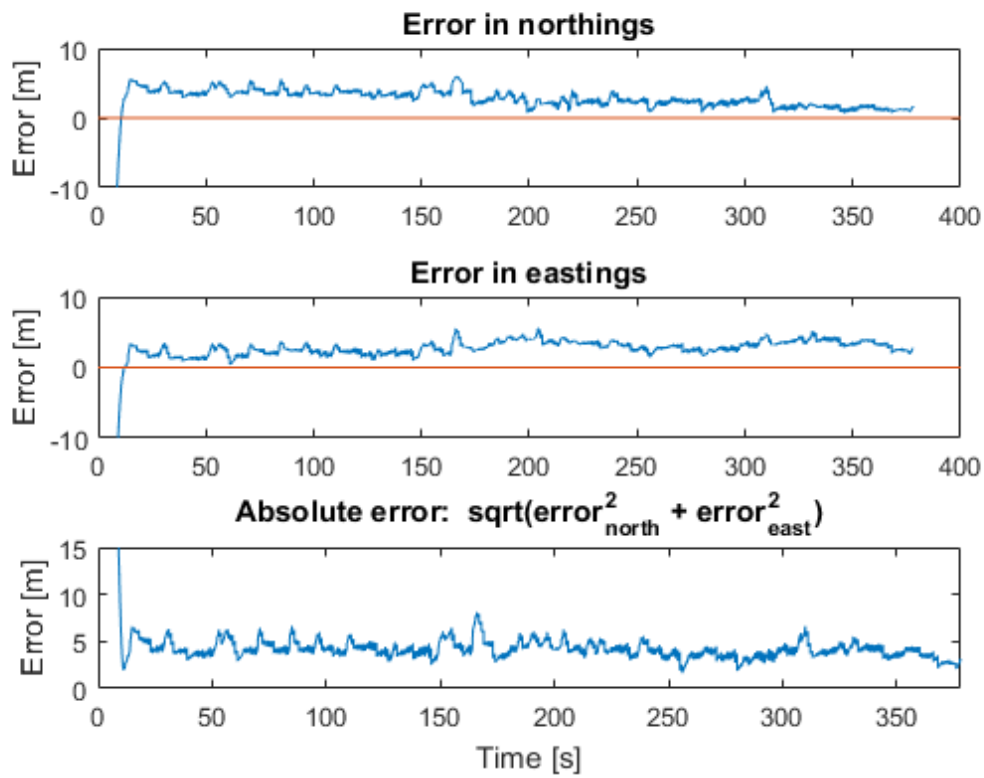


Figure 4.8: The error between the real position and the estimated position by the PF.

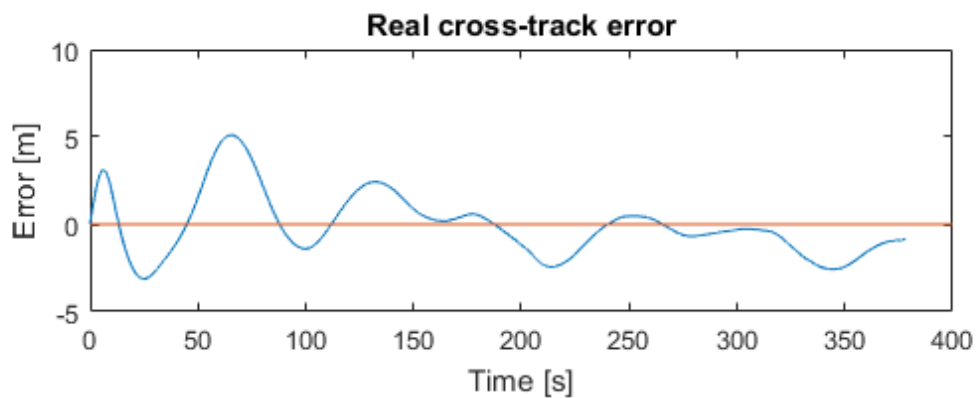


Figure 4.9: The real cross-track error during homing/docking with multinomial resampling.

The error at the dock is 0.80 meters, seen in Figure 4.9.

4.2.3 Residual Resampling

From Figure 4.10, 4.11 and 4.12 one clearly sees that residual resampling is more inaccurate than multinomial. The AUV travels in the direction of the dock, but consequently thinks it is more east than it really is. This results in a fairly large cross-track error, 19 meters at the worst, and an error of 5 meters at the dock.

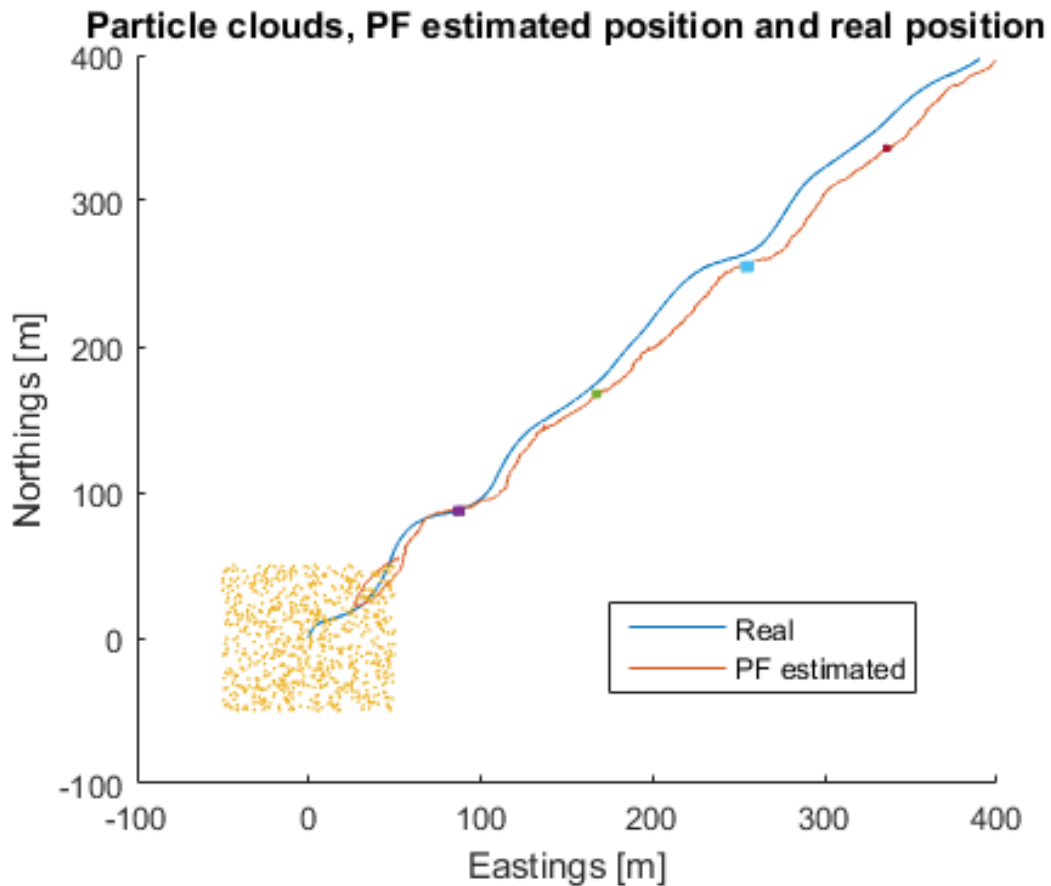


Figure 4.10: The estimated position overlays the actual position well.

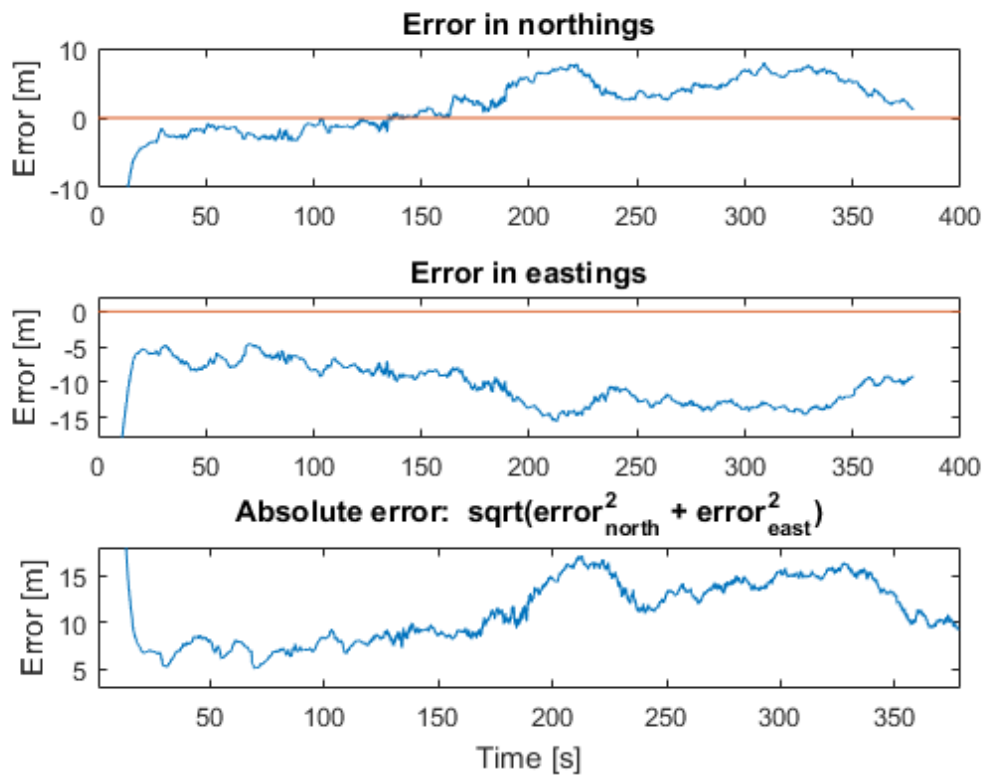


Figure 4.11: The error between the real position and the estimated position by the PF.

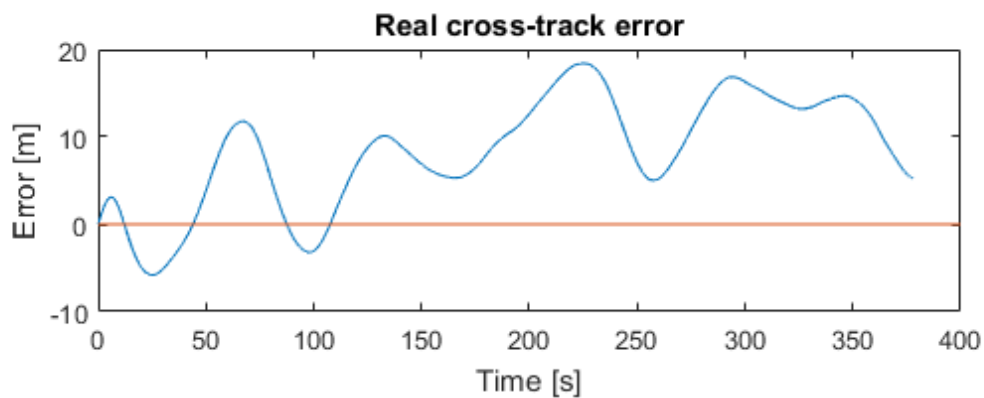


Figure 4.12: The real cross-track error during homing/docking with systematic resampling.

4.2.4 Systematic Resampling

Figure 4.13, 4.14 and 4.15 show that systematic resampling yields the best results, with a cross-track error of just 30 centimeters at the dock. The system seems to behave more stable than the two others, with a more dampened and slowly oscillating cross-track error.

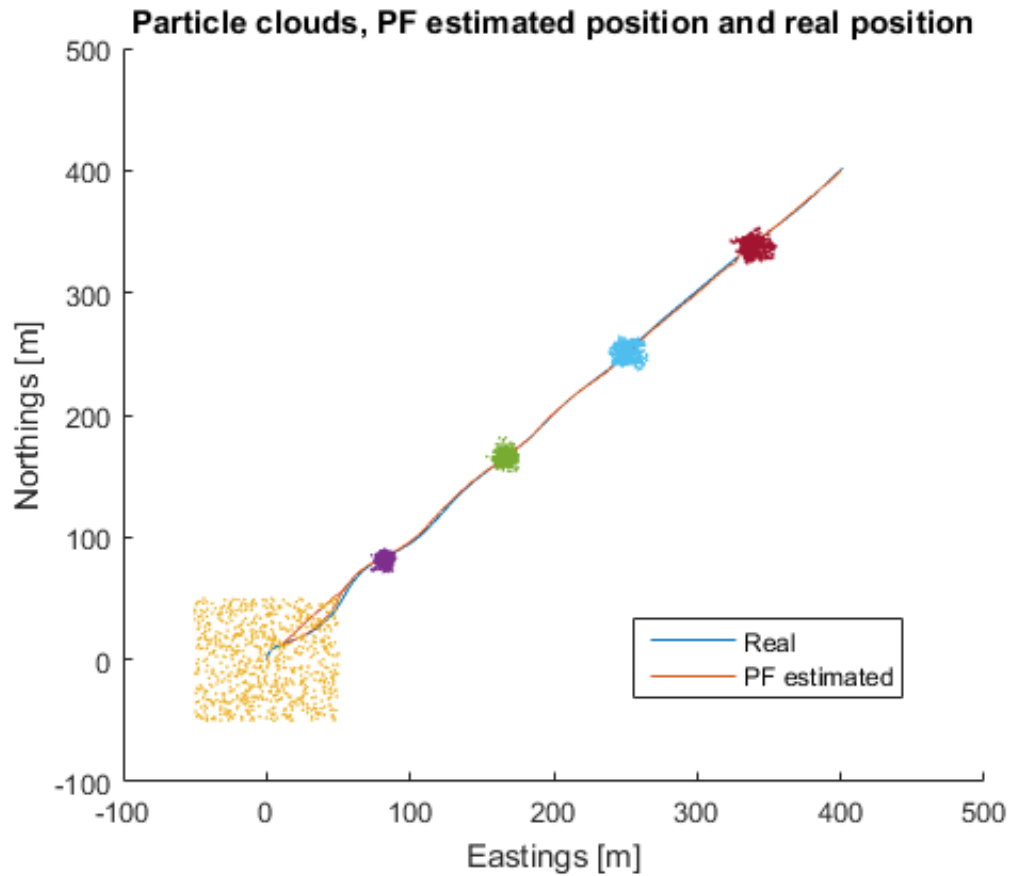


Figure 4.13: The estimated position overlays the actual position very well.

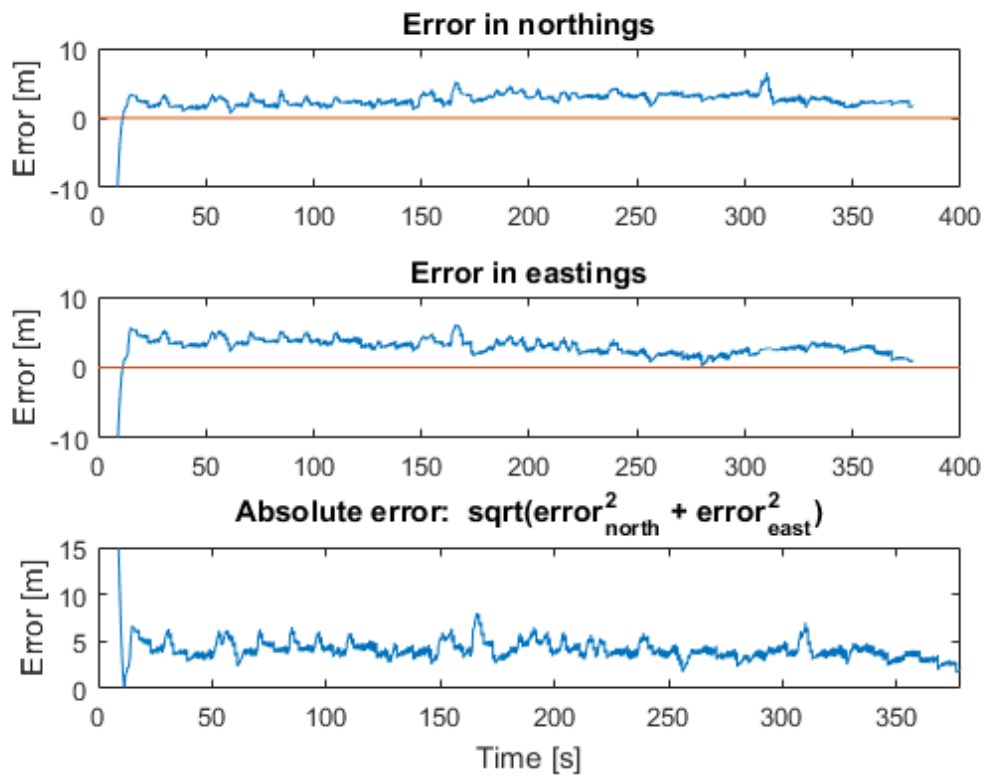


Figure 4.14: The error between the real position and the estimated position by the PF.

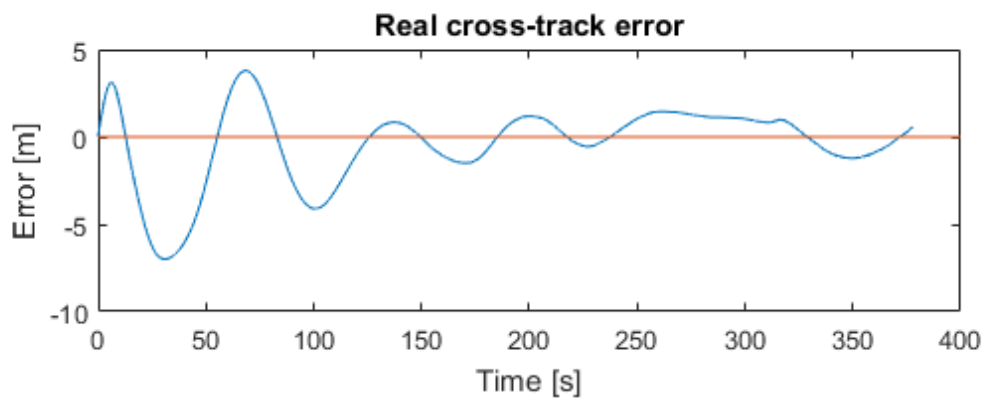


Figure 4.15: The real cross-track error during homing/docking with systematic resampling.

4.3 Depth Keeping

The homing and docking algorithm with DUSBL uses the vertical angle measurements to calculate the depth of the docking station. Since the angle measurements are noisy, even after the low-pass filter, the calculated reference signal is noisy as well. Figure 4.16 shows the error plot.

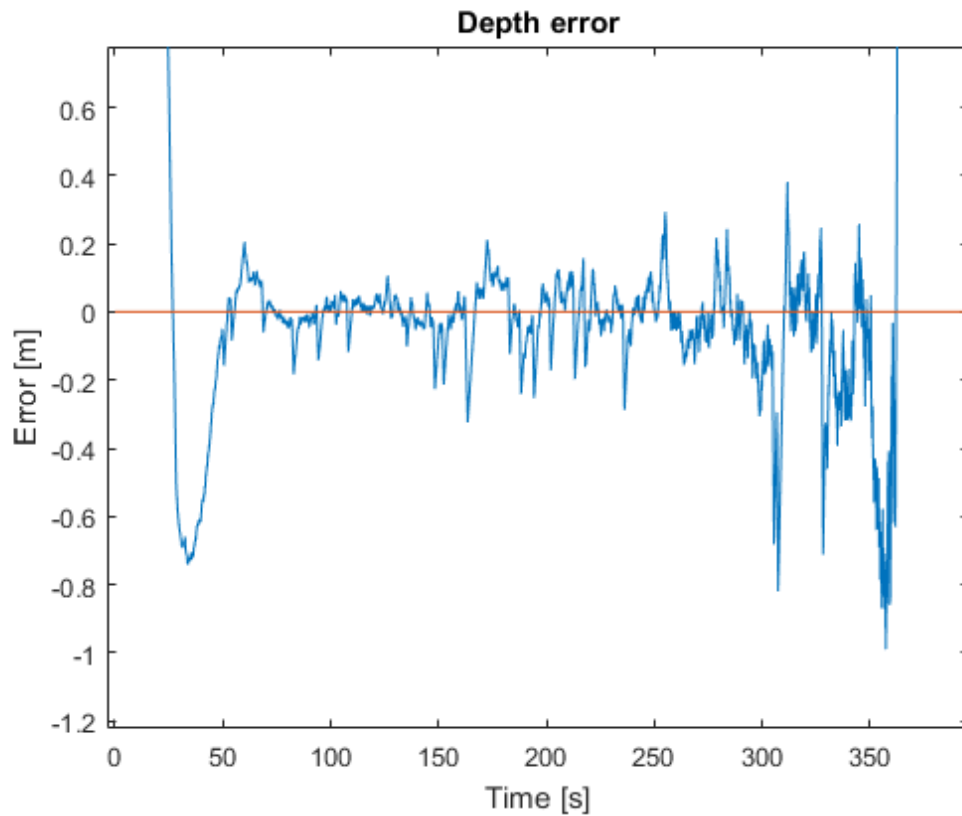


Figure 4.16: Depth error during homing and docking with DUSBL.

During LBL homing and docking, the attitude above sea bed is set to the same as of the dock. This means that no noise terms are influencing the reference signal. The error plot in Figure 4.17 shows almost perfect depth keeping, in contrast to that of DUSBL homing and docking.

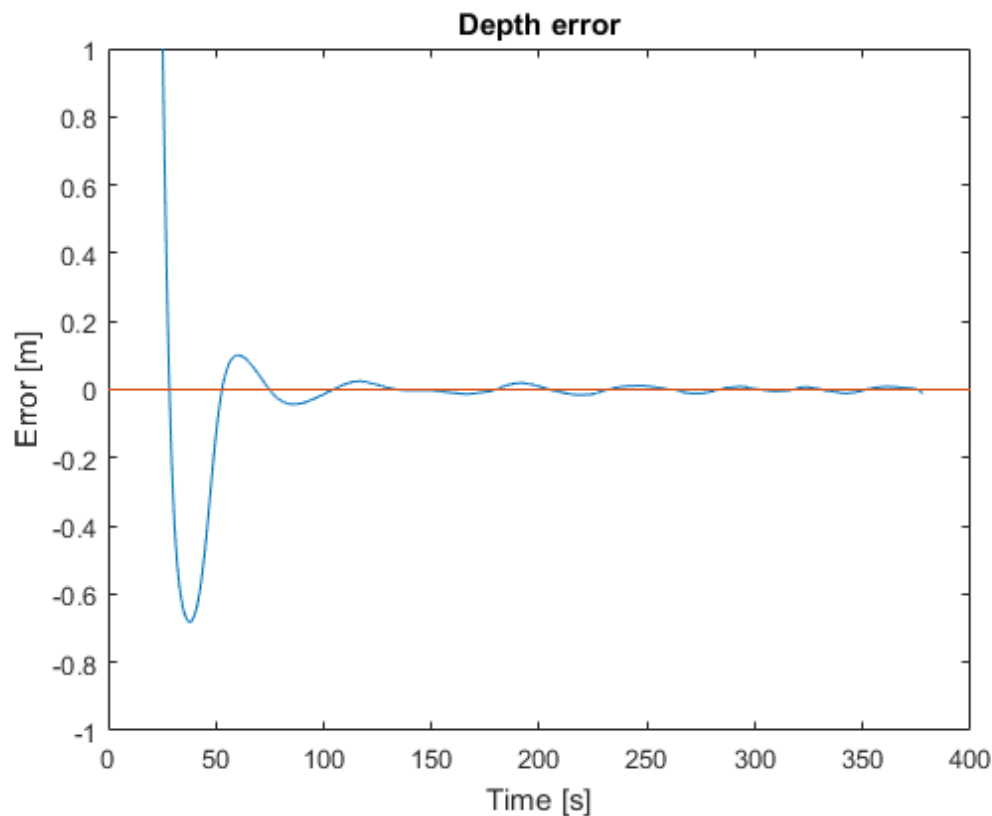


Figure 4.17: Depth error during homing and docking with LBL.

4.4 Results from the LBL Test

This section presents the results from the LBL tests done in Trondheimsfjorden 2. June, 2016. The results from each test are represented by similar plots, making comparison easy.

4.4.1 Line

The upper plot in Figure 4.18 shows that a single LBL measurement was taken at the very start of the mission. After this, the AUV starts to navigate itself to the start of the straight leg that crosses the transponder. The green circles are the points where LBL measurements were made. The data indicates that the maximum frequency of taking measurements is one every third second. The implications of this are discussed in Chapter 5.

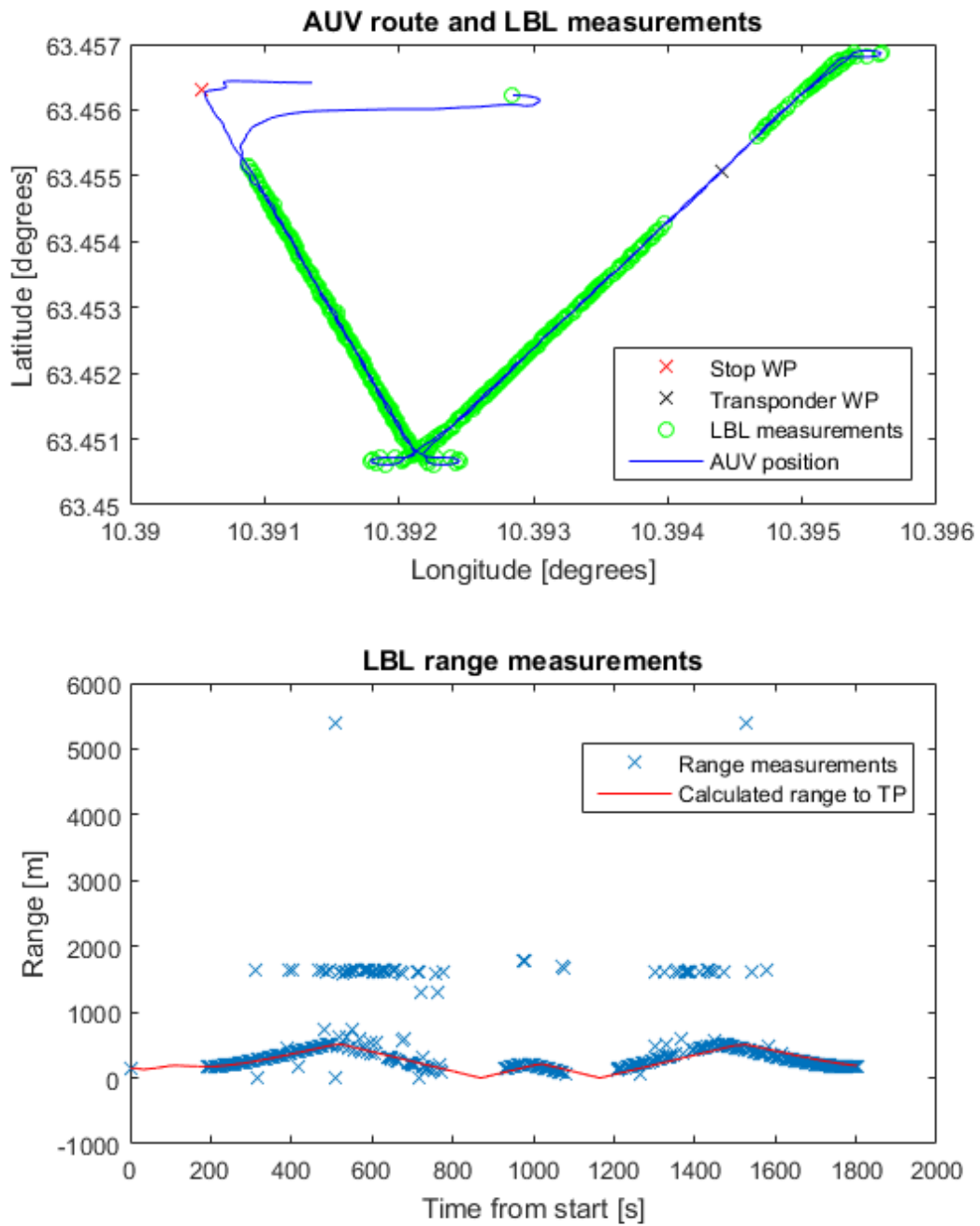


Figure 4.18: Results when traveling over the transponder.

The LBL measurements clearly stop when the AUV is close to the transponder.

The distance is not consistent as can be seen in the bottom plot of Figure 4.18, but in the interval between 70 and 115 meters no measurements are made. Also, there is a lot of wild points in the range measurements. The blue line is the estimated position the AUV uses in its internal control system. The red line in the bottom plot is the distance from the AUV to the transponder position (from the box-in test). This position is calculated with the haversine formula, which is commonly used in navigation. The formula finds the distance d between two positions on a sphere and is given by

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (4.1)$$

where r is the radius of the earth and (φ_1, λ_1) and (φ_2, λ_2) are two coordinates (lat,long) on the sphere. If this distance was correct because of no INS drift, and the box-in test gave the exact transponder position, all range measurements should optimally lay on the red line.

4.4.2 Box #1

The *Box #1* mission has very similar results as *Line*. The LBL is activated in the beginning of the mission, slightly longer in this case, and activated again when the AUV is closer to the dock. Far less LBL measurements are made during this mission, but it seems that the total percentage of wild points are less than the *Line* mission. The range measurements in Figure 4.19 differs from that of Figure 4.18 because measurements are only made when the AUV has a distance of 100 meters or greater from the dock.

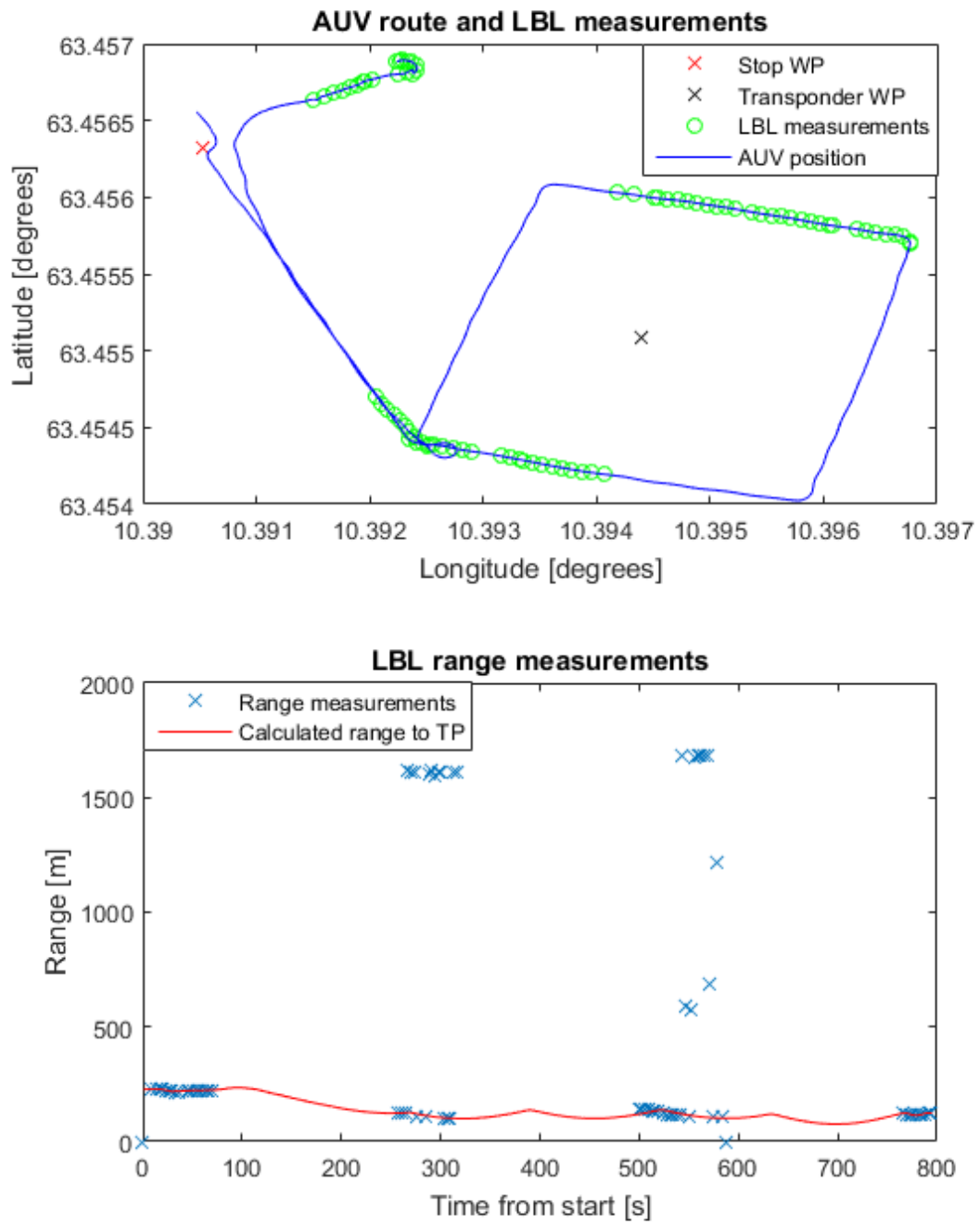


Figure 4.19: Results when traveling in a 100x100 [m] box.

4.4.3 Box #2

The *Box #2* mission is more or less a copy of the *Box #1* mission, but with a larger square. Significantly more LBL measurements are made, a single one at the start and the rest when the AUV starts to travel the square. The wild points are distributed similar as the two previous missions. Figure 4.20 show that no measurements are made when the distance to the transponder is smaller than 160 meters. Thus, the AUV replicates the behaviour of not making LBL measurements in the parts of the path with the closest proximity to the transponder.

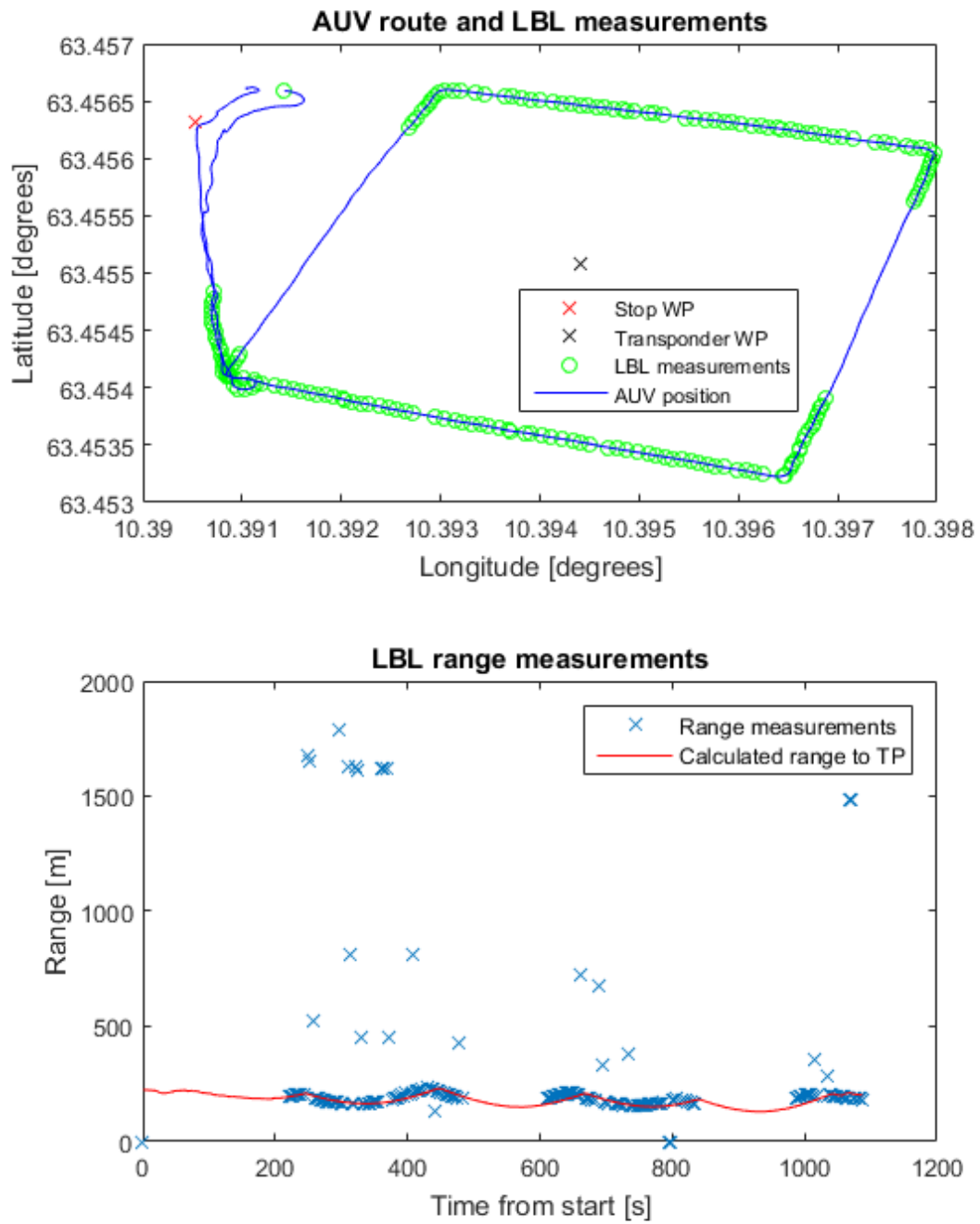


Figure 4.20: Results when traveling in a 200x200 [m] box.

4.4.4 Side Scan Sonar Data

At the point the AUV passed the transponder in the *Line* mission it was detected by the SSS. By doing measurements directly on the figure, the transponder was 3.36 meters from the AUV. Since the build-up of drift in the inertial navigation system is limited on such short missions, it is explained by currents dragging the transponder slightly horizontally. This is to be expected. The transponder location found by the box-in test was 63N27.305 10E23.664. The transponder location found by the SSS was 63N27.3052 10E23.6641.

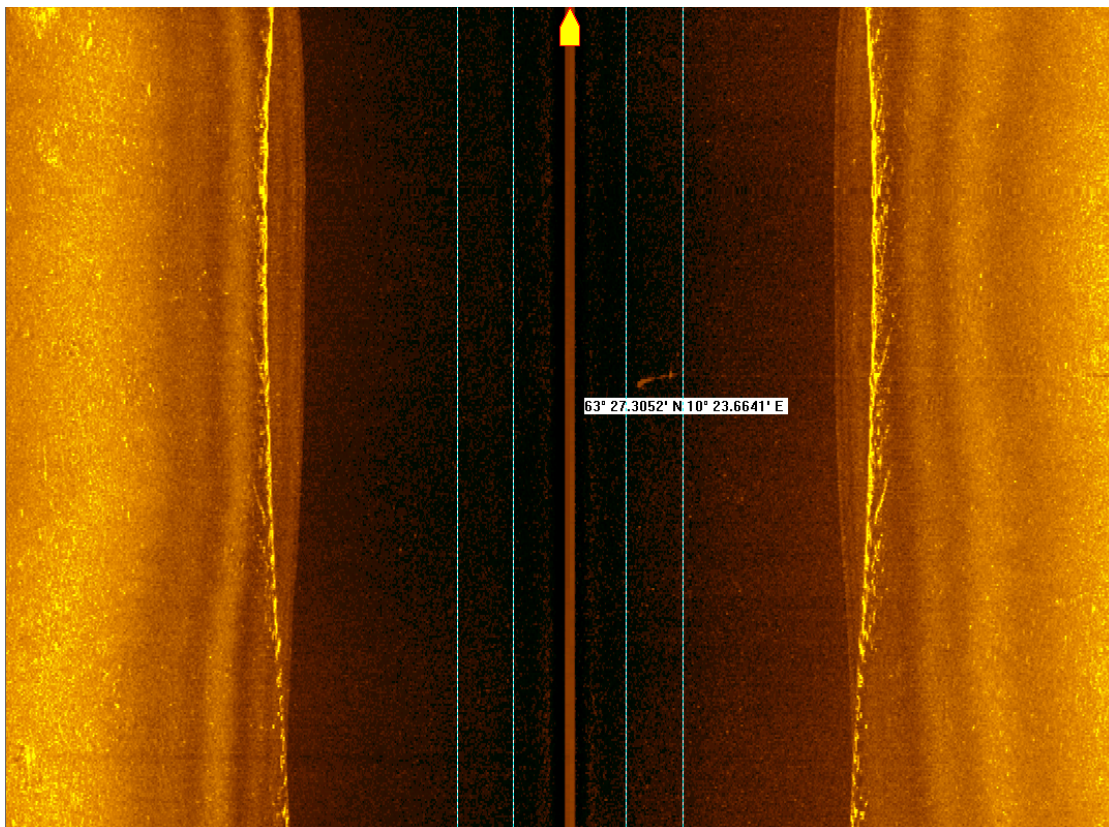


Figure 4.21: The SSS show that the AUV is close to the transponder.

Chapter 5

Discussion

The rate limiter used to remove wild points from the DUSBL/LBL range measurements is not an optimal solution. Figure 3.11 shows that the filtered signal experiences a small peak after a high wild point and vice versa for a negative wild point. The fact that the rate limiter is placed *after* the low-pass filter, just adds to this. A better solution is to add a proper wild point filter which uses a sliding windows to discover wild points. Here, two common algorithms for wild point detection is interesting to simulate; namely *distance-based* and *local metrics-based* (Subramaniam, Palpanas, Papadopoulos, Kalogeraki, & Gunopoulos, 2006). The distance-based algorithm is fast and removes points with sufficient distance to the other points. The local metrics-based is more robust and considers local density variations to remove outliers.

During the DUSBL homing and docking simulation, the cross-track error is reduced to just around 30 centimeters. This means that a docking procedure into a cone dock is possible, without the need of an impractically large "trumpet" entry. Figure 4.3 shows an oscillating AUV behaviour when the PID controller is initiated after the repositioning phase. One way of removing the oscillations is to apply a initial value to the I-term in the controller. This is not very practical because it would have to vary with the current speed and angle. Therefore the oscillations are considered to be acceptable. For very narrow entry paths (e.g. dock placed under a pier) problems might arise.

With an maximum error of one meter, the depth error (Figure 4.16) is governing when deciding if docking is possible. The dock should be placed in a way so that the center line has a flat, non varying surface the last stretch of the docking phase. This way the altitude controller used to keep the depth would have the best conditions for good depth keeping. By setting the AUV to operate at a constant altitude, the vertical angel measurements will not add to the total

error. Figure 4.17 shows that this method would hold the AUV at the set depth, with very small errors. Unfortunately, this depth simulation has to be rendered as unrealistically good, as real life data from Holsen (2015) show otherwise. Altitude oscillations of up to ± 0.5 meters is realistic and docking into a cone dock with a entry diameter of 1 meter is still possible. The pressure sensor in REMUS 100 is very accurate and the reason behind the poor depth keeping performance is a missing damping term in the controller. This is a problem that must be addressed by Hydroid. The simulation depth controller includes a damping term, hence the good performance.

The results favour systematic resampling as the preferred resampling method, when used with 1500 particles. The error at the dock is close to 30 centimeters, which enables docking - at least from simulations. The error in Figure 4.15 oscillates around ± 1 [m] just before the error drops to 30 centimeters. Therefore it could very well be a coincidence that the error was low at the dock. The other simulations show similar oscillating error plots. An implementation on the real system may yield other results. A particle filter is computationally intensive and seizes much of a CPU's resources. It is therefore important to check the real system performance to verify that the lag in the system is not causing position error. If docking is not of paramount importance, the number of particles can be reduced by half (Appendix B) and still get good performance in homing. One application of this can be a "hybride PF algorithm" that has two PFs, where one has a higher number of particles than the other. The PF with most particles is only activated in a docking scenario, whereas the other is used for general, non-critical navigation. This way CPU resources are saved.

The LBL sensor data gathered from the mission in Trondheimsfjorden gave some insight in the REMUS 100. No matter how close the AUV was to the transponder, the measurement rate was one every three seconds. The expected result was that the rate would increase with decreasing distance to the transponder. The reason why this did not happen was that the REMUS 100 listened for the "dummy transponder" which was programmed in the VIP software, but not deployed in the ocean. After time out of three seconds, a new measurement was made. This is important since the good results from the LBL homing and docking simulations demanded a LBL measurement every second. Figure 5.1 shows the estimated and real position when TOF is set to 3 seconds. $\sigma_{noise} = 0.07$. It is more than sufficient for homing, but docking is out of the question.

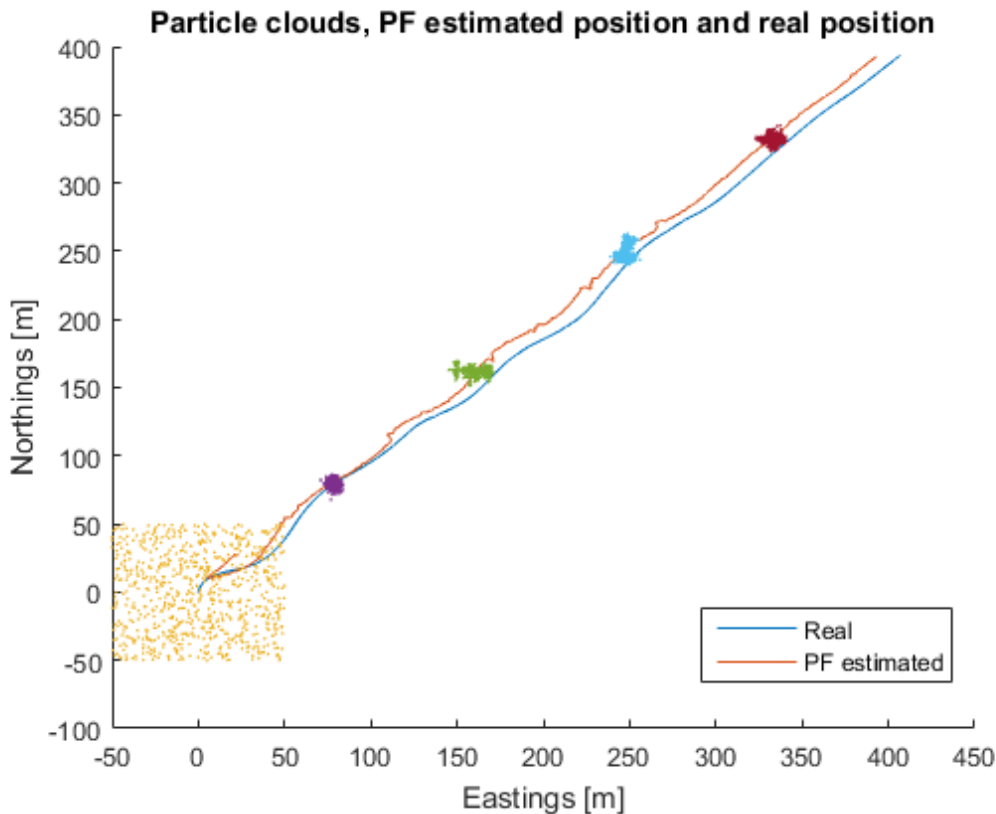


Figure 5.1: PF with LBL measurements every three seconds.

The other problem with the LBL not taking measurements close to the transponder has no "quick fix". When only two transponders are used, there exists a rectangular area with center line equal to the line between the two transponders. Inside this area the position can not be accurately determined because the AUV could be on both sides of the line. Therefore the transponder is not pinged here. A "hack" to overcome this, is to program two transponders in the VIP software, but only deploy one of them in the programmed position. The other is deployed in the area of interest. This is the transponder that we take range measurements to. The other transponder is deployed at the programmed position, and makes sure that two signals are received by REMUS 100, thus gets rid of the time out problem also.

The gathered data also verifies that the magnitude and frequency of the simulated wild points are realistic. Also, it became very clear that the LBL system is disengaged in close proximity to the transponder. The data is not consistent on the exact distance, but no data should be expected when the AUV is closer than 100 meters. This has a very negative impact on LBL homing and docking.

It is in the docking phase, the last hundred meters, fast LBL measurements are the most important. Without these measurements, docking is not possible. However, to optimize the acoustic range and quality of the signal, the transponder placement is not arbitrary. The sound speed profile of the water column and the depth of the transponder have great impact on the acoustic wave field. In future missions an acoustic model could help to plan the optimal path inside a LBL network (Siddiqui, Ludvigsen, & Dong, 2015).

Chapter 6

Conclusions and Further Work

This master's thesis presents a homing and docking algorithm for both a DUSBL and LBL sensor on the REMUS 100 AUV. Homing and Docking with the DUSBL sensor are based on standard PID controllers, line-of-sight guidance and the use of REMUS' internal controllers. A particle filter is created that is used in conjunction with an LOS algorithm to perform homing and docking with the LBL sensor. A homing/docking block is added to AUVSim which is able to simulate a single-transponder DUSBL and LBL signal. The block also implements a noise filter, particle filter and a guidance algorithm to steer the AUV into a docking station.

Homing and docking with the DUSBL sensor show promising results and might be suitable for docking. Real life performance has to be tested before this can be said for certain. Simulation results of the depth controller, when set to a constant depth, is not consistent with reality and show too good results. If the docking station is placed in an ocean environment with demanding terrain it is reasonable to believe that the docking task is hard to realize consistently. A flat terrain around the dock will on the other hand successfully dock the REMUS 100.

Homing and docking with the line-of-sight algorithm combined with position estimates from a particle filter are possible according to the simulation results. Of course, this demands a robust depth controller as in the case of the DUSBL sensor. The particle filter resampling strategy that worked best was systematic resampling. However, real life LBL data from REMUS 100 show that the LBL measurement rate is too slow to perform docking. Also, a critical investigation was that the LBL measurements stopped when the AUV was closer to the transponder than approximately 100 meters. Even though this can be solved, the solution is not elegant or practical.

Despite all this, and if docking had to be done, the diameter of the entrance would have to be impractically large. A better solution would be a simple net suspended between some weights and floaters. This would yield much better chances of retrieval. The DUSBL and LBL sensor show promising results and can be used to control the AUV to the area around a transponder with an accuracy of around a few meters. In most cases this would be more than enough for a safe and controlled vessel retrieval, even in demanding environments.

6.1 Further Work

Real-life ocean tests need to be carried out and the algorithm adjusted and tuned according to the results. This means that the homing and docking algorithms need to be implemented. DUNE is a good facilitator for such implementation. After this, a natural next step is to fuse the position estimates calculated from the LBL and DUSBL sensors together with inertial sensor data and other measurements into a Kalman filter. This way the REMUS 100 will get a more accurate position estimate (position relative to the transponder) which is needed in the docking phase. The DUSBL sensor needs to be interfaced with the on-board REMUS 100 computer so that measurement data can be used in custom algorithms. Also, the DUSBL characteristics (signal strength and quality as a function of angle to the transponder) has to be tested in order to improve the simulations.

The LBL measurement rate on the REMUS 100 has to increase before the particle filter will work optimally. A possible hack to make this work was described in Chapter 5, but it is favorable with a solution that gives more flexibility. Preferably the ping rate should be able to set manually. It is very likely that changes are needed in the internal REMUS code and that only Hydroid has access. If change is not possible, one might consider improving the algorithm with SSS data. This is tightly coupled to the SLAM problem and is an advanced topic. If a map of the seabed exists, a particle filter can be used to give an estimate of the most likely AUV position. This way the initial guess of the AUV position can be tightened. Since the SSS is low resolution and noisy, potential geographical features must be very distinct and visible. It may not even be possible. In the situation where transponders are tossed from a plane to create an interim local search area at random depths, 3D range-only localization must be investigated.

References

- Allen, B., Vorus, W., & Prestero, T. (2000). Propulsion system performance enhancements on REMUS AUVs. In *Oceans 2000 mts/ieee conference and exhibition* (Vol. 3, p. 1869-1873). doi: 10.1109/OCEANS.2000.882209
- AUSI. (2015). *Autonomous Oceanographic Sampling Networks*. Retrieved from <http://ausi.org/research/sauv/aosn/> (Online; accessed 20-October-2015)
- AUVAC. (2013). *Summary Report: 18th International Symposium on Unmanned Untethered Submersible Technology*. Retrieved from <http://auvac.org/newsitems/view/2242> (Online; accessed 14-October-2015)
- Baccou, P., & Jouvencel, B. (2002). Homing and Navigation using One Transponder for AUV, Post-Processing Comparisons Results with Long BaseLine Navigation. In *International conference on robotics and automation* (pp. 4004–4009). doi: 10.1109/ROBOT.2002.1014361
- Bahr, A. (2009). *Cooperative Localization for Autonomous Underwater Vehicles* (Doctoral Thesis).
- Balchen, J. G., Andresen, T., & Foss, B. A. (2003). *Reguleringsteknikk*. Institutt for teknisk kybernetikk, Norges Teknisk-naturvitenskapelige Universitet, (5. utgave)
- Beard, R. W., & McLain, T. W. (2012). *Small unmanned aircraft : theory and practice*. Princeton, N.J. Princeton University Press. (Chapter 6)
- Bellingham, J. G., Goudey, C., Consi, T., Bales, J., Atwood, D., Leonard, J., & Chryssostomidis, C. (1994, Jul). A second generation survey AUV. In *Autonomous underwater vehicle technology, 1994. auv '94., proceedings of the 1994 symposium on* (p. 148-155). doi: 10.1109/AUV.1994.518619
- Bellingham, J. G., McEwen, R. S., Hobson, B. W., & McBride, L. (2008). Docking Control System for a 54-cm-Diameter (21-in) AUV [Journal Article]. *Oceanic Engineering, IEEE Journal of*, 33(4), 550-562. Retrieved from <http://ieeexplore.ieee.org/ielx5/48/4769672/04769696.pdf?tp=&arnumber=4769696&isnumber=4769672> doi: 10.1109/JOE.2008.2005348
- Blidberg, D. R. (2001, May). *The Development of Autonomous Underwater Vehicles (AUV); A Brief Summary*.
- Breivik, M., & Fossen, T. I. (2009). Guidance laws for autonomous underwater vehicles. In *Underwater vehicles : (4th ed., pp. 51–76)*. IN-TECH.
- Brighenti, A. (1990, Jul). Parametric analysis of the configuration of autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 15(3), 179-188. doi: 10.1109/48.107146
- Brutzman, D. P. (1994). *A Virtual World for an Autonomous Underwater Vehicle*. Naval Postgraduate School Monterey, California.

- Burdinsky, I. N., & Otcheskiy, S. A. (2014). AUV Localization Using a Single Transponder Acoustic Positioning System.
- Carlton, J. (2007). 11 - Propeller-ship interaction. In J. Carlton (Ed.), *Marine Propellers and Propulsion (Second Edition)* (Second Edition ed., p. 264 - 283). Oxford: Butterworth-Heinemann. Retrieved from <http://www.sciencedirect.com/science/article/pii/B9780750681506500139>
doi: <http://dx.doi.org/10.1016/B978-075068150-6/50013-9>
- Casey, T., Guimond, B., & Hu, J. (2007, Sept). Underwater vehicle positioning based on time of arrival measurements from a single beacon. In *Oceans 2007* (p. 1-8). doi: 10.1109/OCEANS.2007.4449186
- Cruz, N. A., & Matos, A. C. (2008, Sept). The mares auv, a modular autonomous robot for environment sampling. In *Oceans 2008* (p. 1-6). doi: 10.1109/OCEANS.2008.5152096
- Curtin, T. B., Bellingham, J. G., Catipovic, J., & Webb, D. (1993). Autonomous Oceanographic Sampling Networks. In *Oceanography* (Vol. 6, p. 86-94). DeepOceanAndDeepSpace.com. (2015). *Shades of EAVE*. Retrieved from <http://deepoceananddeepspace.com/2015/09/15/shades-of-eave/> (Online; accessed 26-October-2015)
- Donovan, G. T. (2012, July). Position error correction for an autonomous underwater vehicle inertial navigation system (ins) using a particle filter. *IEEE Journal of Oceanic Engineering*, 37(3), 431-445. doi: 10.1109/JOE.2012.2190810
- Doucet, A., de Freitas, N., & Gordon, N. (2001). Sequential monte carlo methods in practice. In A. Doucet, N. de Freitas, & N. Gordon (Eds.), (pp. 3-14). New York, NY: Springer New York. Retrieved from http://dx.doi.org/10.1007/978-1-4757-3437-9_1 doi: 10.1007/978-1-4757-3437-9_1
- Evans, J., Keller, K., Smith, J., Marty, P., & Rigaud, O. (2001). Docking techniques and evaluation trials of the SWIMMER AUV: An Autonomous Deployment AUV for Work-Class ROVs. In *Oceans, 2001. mts/ieee conference and exhibition* (Vol. 1, p. 520-528 vol.1). doi: 10.1109/OCEANS.2001.968776
- Ferreira, B., Matos, A., & Cruz, N. (2010, Sept). Single beacon navigation: Localization and control of the mares auv. In *Oceans 2010 mts/ieee seattle* (p. 1-9). doi: 10.1109/OCEANS.2010.5664518
- Fossen, T. I. (2011). Seakeeping theory. In *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd. doi: 10.1002/9781119994138.ch5
- Francois, R. (1977). *High Resolution Observations of Under-Ice Morphology*. Applied Physics Laboratory, University of Washington.
- Francois, R., & Nodland, W. (1972). *Unmanned Arctic Research Submersible (UARS) System Development and Test Report*. Applied Physics

- Laboratory, University of Washington.
- Funnell, C., & Group, J. I. (1998). *Jane's Underwater Technology*. Jane's Information Group.
- Havforskningsinstituttet. (2013). *The Norwegian Coastal Current*. Retrieved from http://www.imr.no/temasider/kyst_og_fjord/den_norske_kyststrommen/en (Online; accessed 23-May-2016)
- Hewage, T., Huynh, J. Q. T., Hugo, A. P., Scheide, A. W., Ruud, F. J., Arnesen, B. O., ... Mossige, J. C. (2015). TMR4585 Underwater Technology Project work report.
- Hol, J. D., Schon, T. B., & Gustafsson, F. (2006, Sept). On resampling algorithms for particle filters. In *Nonlinear statistical signal processing workshop, 2006 ieee* (p. 79-82). doi: 10.1109/NSSPW.2006.4378824
- Holsen, S. A. (2015). *DUNE: Unified Navigation Environment for the REMUS 100 AUV* (Master Thesis).
- Honeywell. (2015). *HG1700 Inertial Measurement Unit*. Retrieved from <https://aerospace.honeywell.com/~media/Products/Navigation%20Systems%20and%20Sensors/Sensors%20and%20Inertial%20Products/HG1700%20Datashet.ashx> (Brochure download. Online; accessed 26-October-2015)
- Hydroid. (2012). *REMUS 100 Operations and Maintenance Manual V.1.0; REMUS 100 sn01417*.
- Imahori, G., Lanerolle, L., Brodet, S., Downs, R., Xu, J., Becker, R., & Robidoux, L. (2008, September). An integrated approach to acquiring sound speed profiles using the Regional Ocean Modeling System (ROMS) and an Autonomous Underwater Vehicle (AUV).
- Kongsberg Maritime. (2012a). *Hugin 1000 Payload SDK Reference Manual*.
- Kongsberg Maritime. (2012b). *REMUS REmote CONTrol PROTOCOL KMHYD-Man-0002*.
- Kongsberg Maritime. (2015). *HiPAP High Precision Acoustic Positioning*. Retrieved from <http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/FF57C18363FAD917C1256A7E002B9F2F?OpenDocument#tab-4> (Brochure download. Online; accessed 26-October-2015)
- Kukulya, A., Plueddemann, A., Austin, T., Stokey, R., Purcell, M., Allen, B., ... Pietro, J. (2010, Sept). Under-ice operations with a REMUS-100 AUV in the Arctic. In *Autonomous underwater vehicles (auv), 2010 ieee/oes* (p. 1-8). doi: 10.1109/AUV.2010.5779661
- McEwen, R., Thomas, H., Weber, D., & Psota, F. (2005, April). Performance of an AUV navigation system at Arctic latitudes. *Oceanic Engineering, IEEE Journal of*, 30(2), 443-454. doi: 10.1109/JOE.2004.838336
- NavalDrones.com. (2015). *Special Purpose Underwater Research Vehicle (SPURV)*. Retrieved from <http://www.navaldrones.com/SPURV.html>

- (Online; accessed 22-September-2015)
- Norgren, P., & Skjetne, R. (2014). Using Autonomous Underwater Vehicles as Sensor Platforms for Ice-Monitoring. *Modeling, Identification and Control*, 35(4), 263-277.
- Norgren, P., & Skjetne, R. (2015). Line-of-sight iceberg edge-following using an {AUV} equipped with multibeam sonar. *IFAC-PapersOnLine*, 48(16), 81 - 88. Retrieved from <http://www.sciencedirect.com/science/article/pii/S2405896315021539> (10th {IFAC} Conference on Manoeuvring and Control of Marine Craft {MCMC} 2015Copenhagen, 24–26 August 2015) doi: <http://dx.doi.org/10.1016/j.ifacol.2015.10.262>
- Olson, E., Leonard, J. J., & Teller, S. (2006, Oct). Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, 31(4), 949-958. doi: 10.1109/JOE.2006.880386
- Orhan, E. (2012). Particle filtering. Retrieved from <http://www.cns.nyu.edu/~eorhan/notes/particle-filtering.pdf>
- Prestero, T. (2001). *Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle*. Massachusetts Institute of Technology and Woods Hole Oceanographic Institution. (M.Sc. Thesis)
- Purcell, M., von Alt, C., Allen, B., Austin, T., Forrester, N., Goldsborough, R., & Stokey, R. (2000). New capabilities of the REMUS autonomous underwater vehicle. In *Oceans 2000 mts/ieee conference and exhibition* (Vol. 1, p. 147-151 vol.1). doi: 10.1109/OCEANS.2000.881250
- Saúde, J., & Aguiar, A. P. (2009). Single beacon acoustic navigation for an {AUV} in the presence of unknown ocean currents. *{IFAC} Proceedings Volumes*, 42(18), 298 - 303. Retrieved from <http://www.sciencedirect.com/science/article/pii/S1474667016319115> (8th {IFAC} Conference on Manoeuvring and Control of Marine Craft) doi: <http://dx.doi.org/10.3182/20090916-3-BR-3001.0057>
- Siddiqui, S. I., Ludvigsen, M., & Dong, H. (2015, Oct). Analysis, verification and optimization of auv navigation using underwater acoustic models. In *Oceans 2015 - mts/ieee washington* (p. 1-6).
- Singh, H., Bellingham, J. G., Hover, F., Lemer, S., Moran, B., von der Heydt, K., & Yoerger, D. (2001, Oct). Docking for an autonomous ocean sampling network. *Oceanic Engineering, IEEE Journal of*, 26(4), 498-514. doi: 10.1109/48.972084
- SNAME. (1950). *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid: Report of the American Towing Tank Conference*. Society of Naval Architects and Marine Engineers.
- Stokey, R., Purcell, M., Forrester, N., Austin, T., Goldsborough, R., Allen, B., & von Alt, C. (1997). A docking system for REMUS, an autonomous underwater vehicle [Conference Proceedings]. In *Oceans '97. mts/ieee*

- conference proceedings* (Vol. 2, p. 1132-1136 vol.2). Retrieved from <http://ieeexplore.ieee.org/ielx3/4919/13571/00624151.pdf?tp=&arnumber=624151&isnumber=13571> doi: 10.1109/OCEANS.1997.624151
- Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., & Gunopulos, D. (2006). Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd international conference on very large data bases* (pp. 187–198).
- Särkkä, S. (2012). Lecture 6: Particle Filtering — Sequential Importance Resampling and Rao-Blackwellized Particle Filtering. Retrieved from https://users.aalto.fi/~ssarkka/course_k2012/handout6.pdf
- Sørensen, A. J. (2013). *Lecture Notes: Marine Control Systems, Propulsion and Motion Control of Ships and Ocean Structures*. Norwegian University of Science and Technology, Department of Marine Technology.
- Sørensen, A. J., & Ludvigsen, M. (2015). Towards Integrated Autonomous Underwater Operations.
- Tate, J. W., & Israel, B. (2014). Synthetic Aperture Sonar: Improved Technology for Improved Mine Hunting. *Defence Systems Information Analysis Center*, 1(2). Retrieved from https://www.dsiac.org/resources/dsiac_journal/fall-2014-volume-1-number-2/synthetic-aperture-sonar-improved-technology (Online; accessed 21-October-2015)
- Teledyne RD Instruments. (2011). Acoustic Doppler current profiler: Principles of operation: A practical prime. *Oceanic Engineering, IEEE Journal of*. Retrieved from <http://www.rdinstruments.com/support/SoftwareFirmware/x/cs/files/Manuals/BBPRIME.pdf>
- Teledyne RD Instruments. (2015). Retrieved from <http://www.rdinstruments.com/monitor.aspx> (Online; accessed 20-October-2015)
- Tena Ruiz, I., Petillot, Y., & Lane, D. (2003, Sept). Improved AUV navigation using side-scan sonar. In *Oceans 2003. proceedings* (Vol. 3, p. 1261-1268 Vol.3). doi: 10.1109/OCEANS.2003.178034
- Vallicrosa, G., Ridao, P., Ribas, D., & Palomer, A. (2014, Sept). Active range-only beacon localization for auv homing. In *2014 ieee/rsj international conference on intelligent robots and systems* (p. 2286-2291). doi: 10.1109/IROS.2014.6942871
- Vickery, K. (1998). Acoustic Positioning Systems "A Practical Overview of Current Systems". In *Dynamic Positioning Conference*.
- WetLabs. (2015). *ECO Puck Miniature Optical Sensor*. Retrieved from http://wetlabs.com/sites/default/files/documents/datasheet_ECO%20Puck_0.pdf (Online; accessed 19-October-2015)
- WHOI. (2015). Retrieved from <http://www.whoi.edu/main/remus> (Online;

accessed 4-November-2015)

Wynn, R. B., Huvenne, V. A., Bas, T. P. L., Murton, B. J., Connelly, D. P., Bett, B. J., ... Hunt, J. E. (2014). Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, 352, 451-468. Retrieved from <http://www.sciencedirect.com/science/article/pii/S0025322714000747>

Appendix A

Reduced-order AUV Models

For slender symmetrical bodies, as an AUV, the 6 DOF equations of motion can be decomposed into two non-interacting subsystems (Fossen, 2011). That is, a longitudinal and a lateral subsystem. This is reasonable to do when developing AUV controllers. The two subsystems are expressed as the following (Fossen, 2011):

A.1 Longitudinal subsystem:

$$\begin{aligned}
 & \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & mz_g - X_{\dot{q}} \\ X_{\dot{w}} & m - Z_{\dot{w}} & -mx_g - Z_{\dot{q}} \\ mz_g - X_{\dot{q}} & -mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -X_u & -X_w & -X_q \\ -Z_u & -Z_w & -Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} \\
 & + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -(m - X_{\dot{u}})u \\ 0 & (Z_{\dot{w}} - X_{\dot{u}}) & mx_g u \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ W\overline{BG}_z \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_3 \\ \tau_5 \end{bmatrix}
 \end{aligned} \tag{A.1}$$

A.2 Lateral subsystem:

$$\begin{aligned}
 & \begin{bmatrix} m - Y_{\dot{v}} & -mz_g - Y_{\dot{p}} & mx_g - Y_{\dot{r}} \\ -mz_g - Y_{\dot{p}} & I_x - K_{\dot{p}} & -I_{zx} - K_{\dot{r}} \\ mx_g - Y_{\dot{r}} & -I_{zx} - K_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & -Y_p & -Y_r \\ -M_v & -M_p & -M_r \\ -N_v & -N_p & -N_r \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} \\
 & + \begin{bmatrix} 0 & 0 & (m - X_{\dot{u}})u \\ 0 & 0 & 0 \\ (X_{\dot{u}} - Y_{\dot{v}})u & 0 & mx_g u \end{bmatrix} \begin{bmatrix} v \\ p \\ r \end{bmatrix} + \begin{bmatrix} 0 \\ W\overline{BG}_z \sin(\phi) \\ 0 \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_4 \\ \tau_6 \end{bmatrix}
 \end{aligned} \tag{A.2}$$

A.3 Systems Simplifications

Further simplification of equation A.2 can be done by assuming small roll motions and constant forward speed, u_0 (which is controlled by a speed controller), thus obtaining a sway-yaw maneuvering model (Fossen, 2011):

$$\begin{aligned} & \begin{bmatrix} m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{v} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -Y_v & -Y_r \\ -N_v & -N_r \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} \\ & + \begin{bmatrix} 0 & (m - X_{\dot{u}})u_0 \\ (X_{\dot{u}} - Y_{\dot{v}})u_0 & mx_g u_0 \end{bmatrix} \begin{bmatrix} v \\ r \end{bmatrix} = \begin{bmatrix} \tau_2 \\ \tau_6 \end{bmatrix} \end{aligned} \quad (\text{A.3})$$

A simplification analog to the above is possible to apply to equation A.1 for dept/diving controller design:

$$\begin{aligned} & \begin{bmatrix} m - Z_{\dot{w}} & -mx_g - Z_{\dot{q}} \\ -mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -Z_w & -Z_q \\ -M_w & -M_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} \\ & + \begin{bmatrix} 0 & -(m - X_{\dot{u}})u_0 \\ (Z_{\dot{w}} - X_{\dot{u}})u_0 & mx_g u_0 \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ W\overline{BG}_z \sin \theta \end{bmatrix} = \begin{bmatrix} \tau_3 \\ \tau_5 \end{bmatrix} \end{aligned} \quad (\text{A.4})$$

This is not given more consideration in this project thesis, since the main focus is on the sway/yaw dynamics. A.3 has successfully proved capabilities of autonomous docking (Bellingham et al., 2008) and is discussed closer in Chapter 2.6.1.

Appendix B

Particle Filter Results

Results when experimenting with different numbers of particles using systematic resampling.

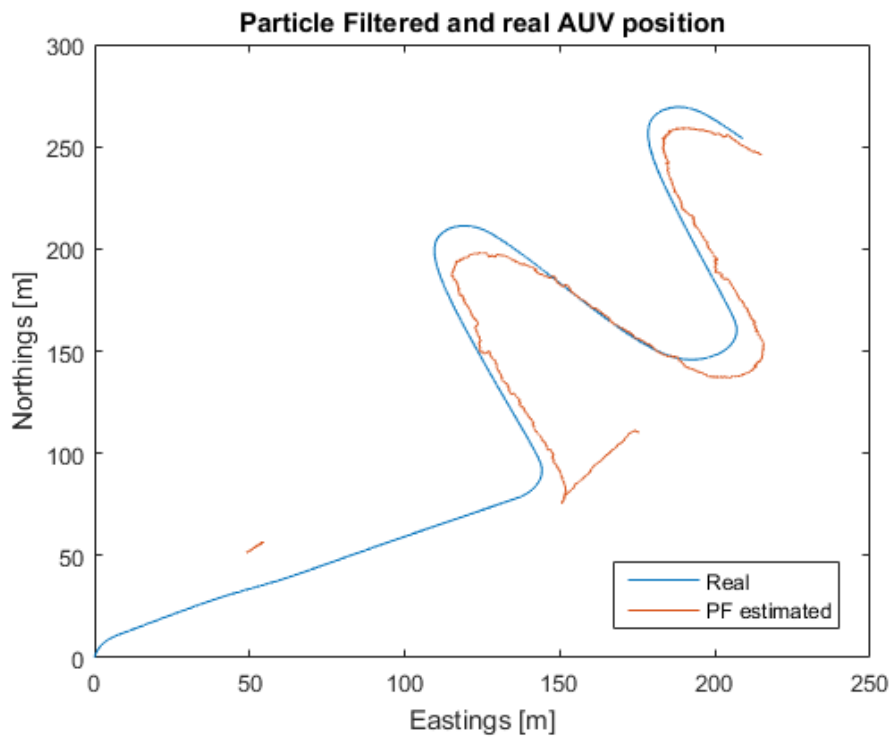


Figure B.1: LBL homing/docking with 100 particles.

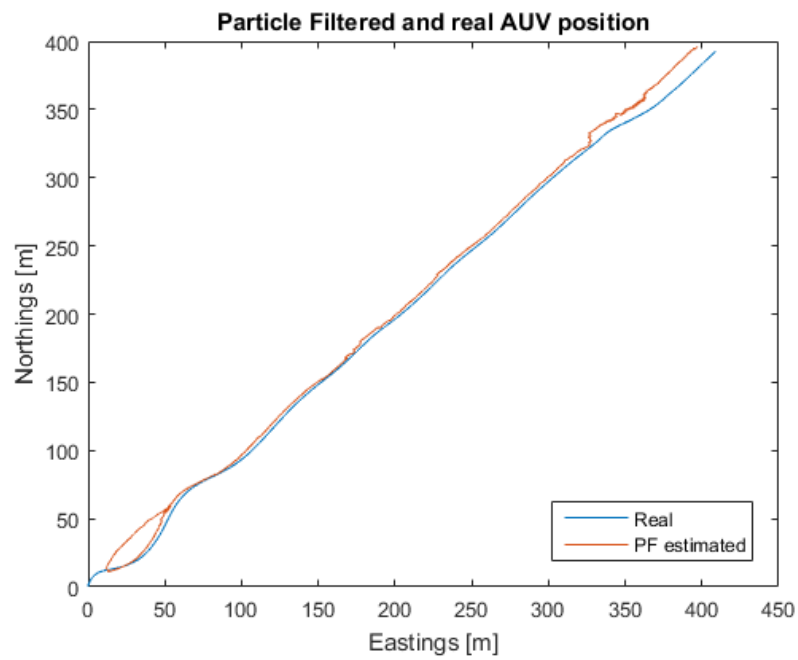


Figure B.2: LBL homing/docking with 700 particles.

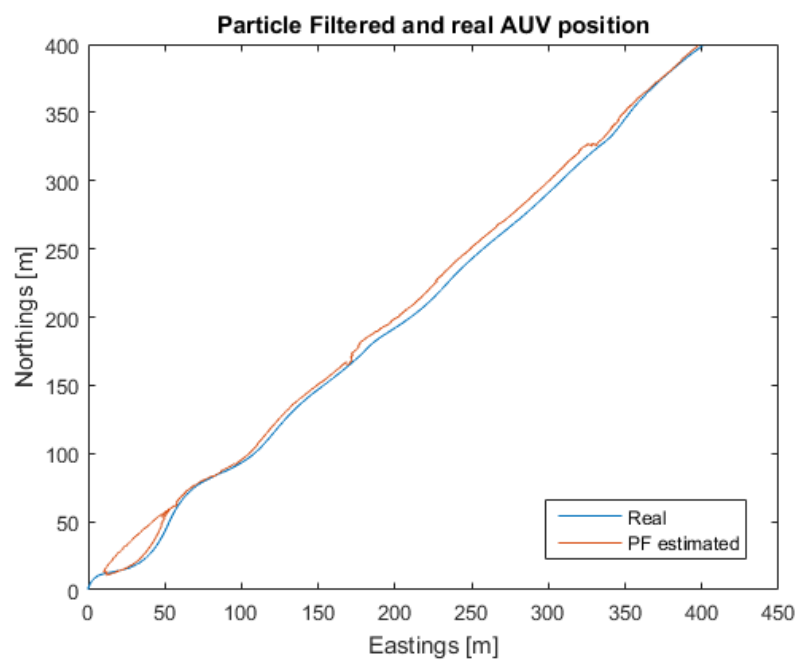


Figure B.3: LBL homing/docking with 1000 particles.

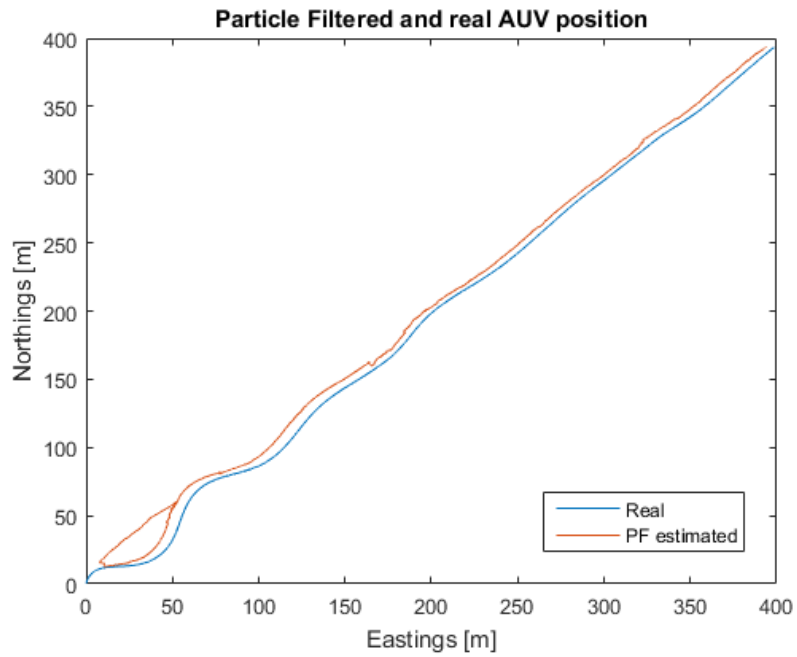


Figure B.4: LBL homing/docking with 1200 particles.

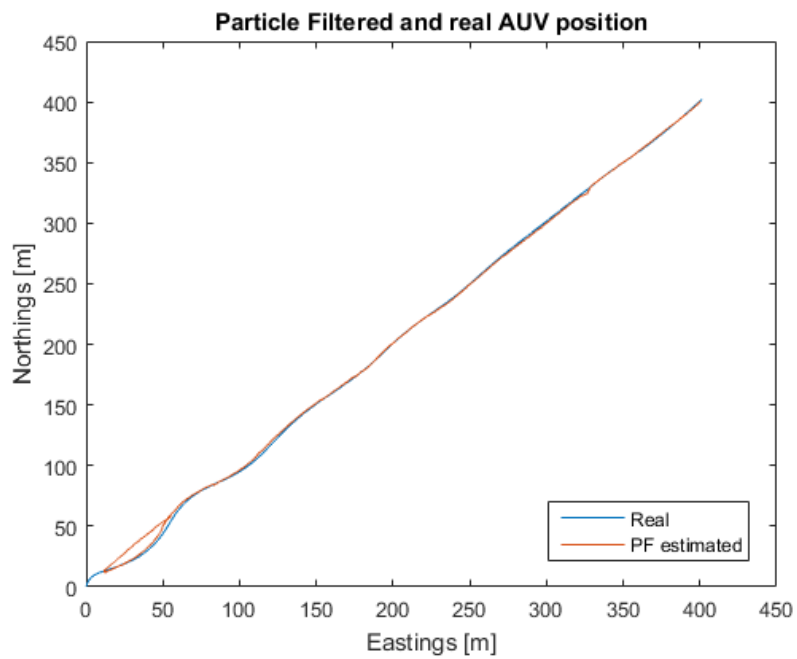


Figure B.5: LBL homing/docking with 1500 particles.

Appendix C

Additional Figures

This appendix holds additional figures that may not be practical to show in the main part of the thesis due to their level of detail and/or size.

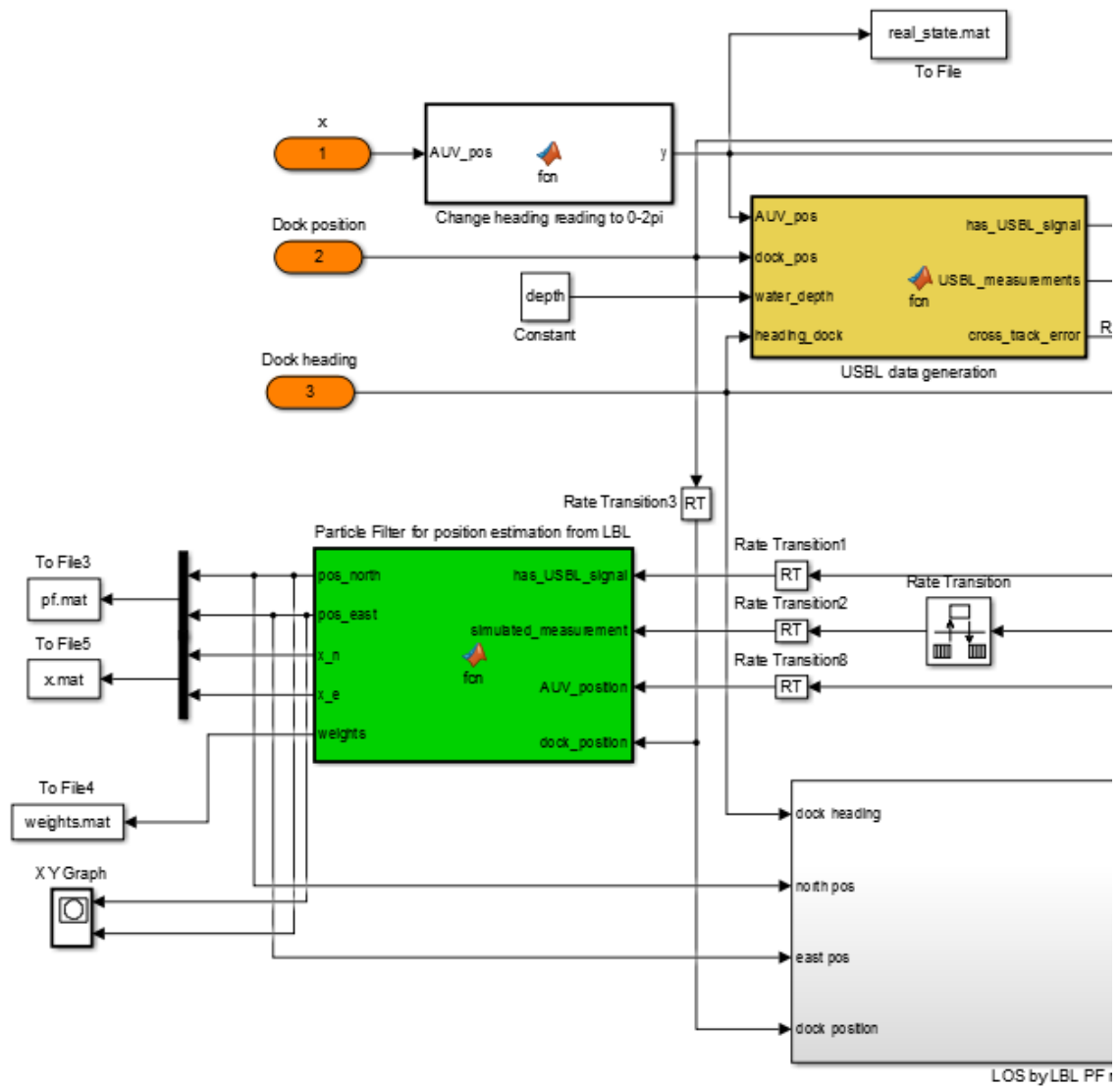


Figure C.1: Large diagram of the homing/docking Simulink block. 1/3

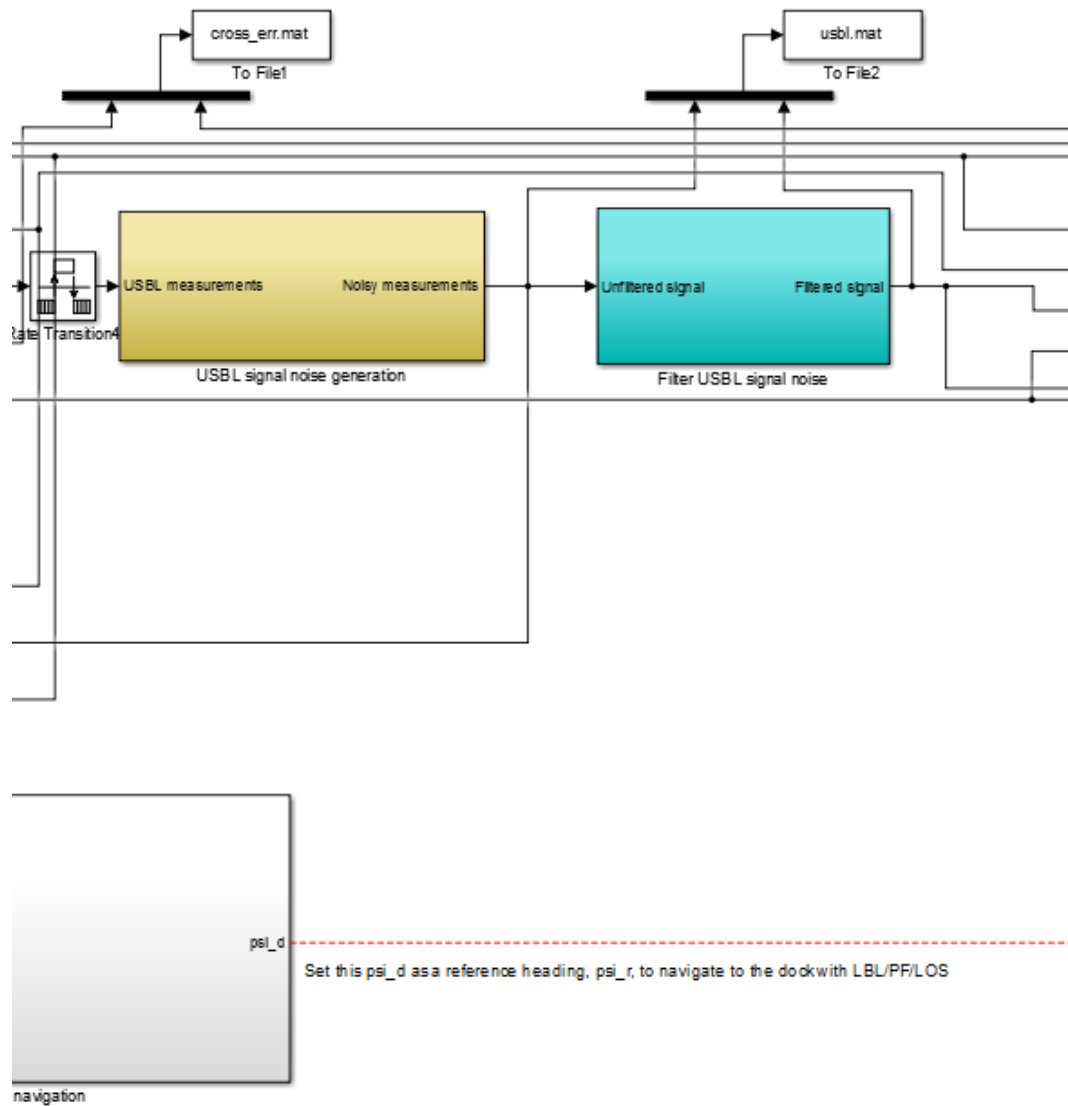


Figure C.2: Large diagram of the homing/docking Simulink block. 2/3

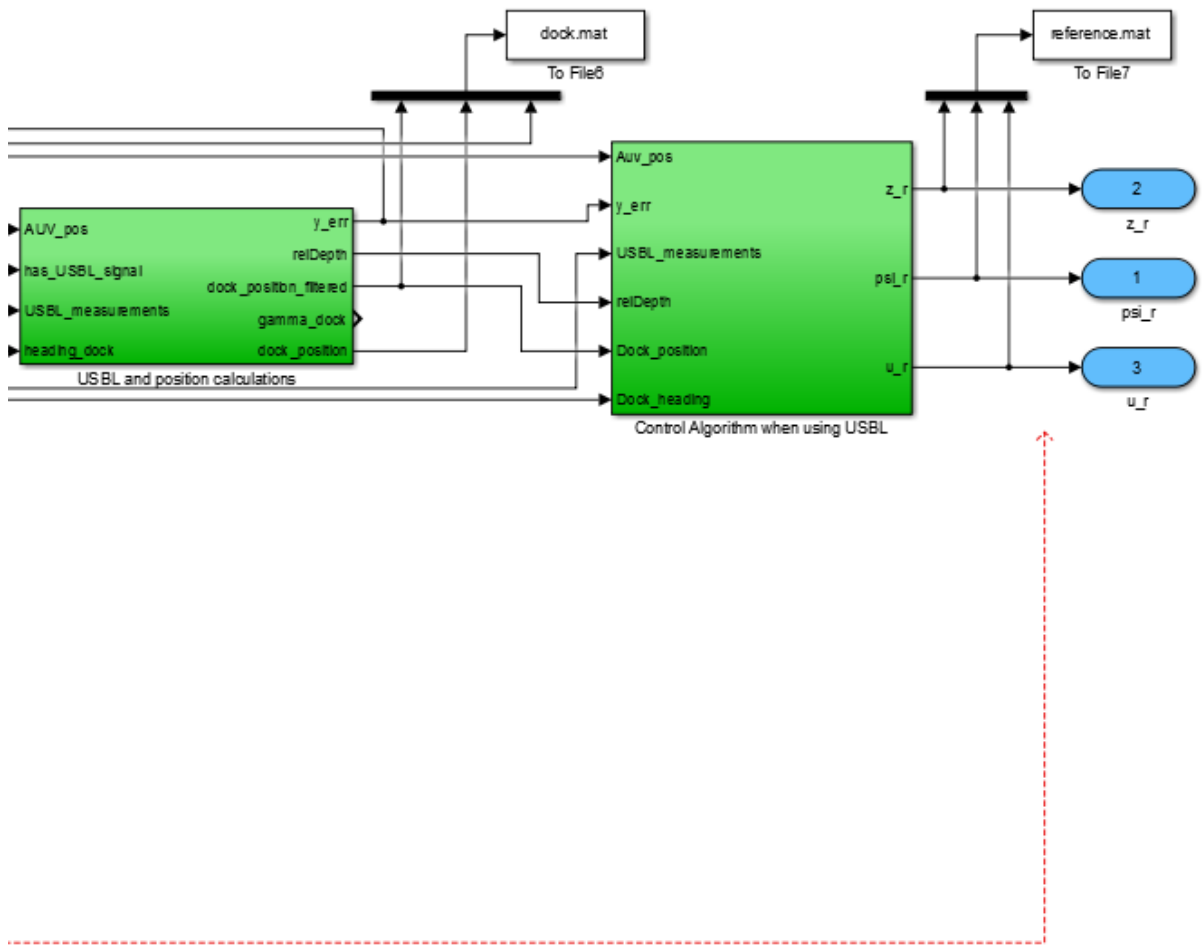


Figure C.3: Large diagram of the homing/docking Simulink block. 3/3

Appendix D

Attachments

This appendix lists the attachments. The attachment is a zip file that includes the following folders:

AUVSim

All files and source code necessary to run AUVSim, with the homing/docking block added.

LBL Test

Data captured from the LBL test in Trondheimsfjorden with the plot files used to generate the plots.

Msc poster

Mandatory scientific poster that presented the work done so far, at May 24.

