NTNU – Trondheim
Norwegian University of
Science and Technology

Marine Telepresence System

Eirik Valle

Marine Technology
Submission date: June 2015
Supervisor: Roger Skjetne, IMT

Norwegian University of Science and Technology
Department of Marine Technology

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

PROJECT DESCRIPTION SHEET

Name of the candidate: Eirik Valle

Field of study: Marine control engineering

Thesis title (Norwegian): Marint Telerobotisk System

Thesis title (English): Marine Telepresence System

Background

Telepresence refers to a set of technologies that give a person the feeling of being present at a remote location away from the true location of the person. Telerobotics is an extension of telepresence, by allowing a person to perform a set of tasks at the remote location.

This thesis will investigate a system for telepresence and telerobotics using a head-mounted display (HMD) in the Marine Cybernetics laboratory (MC-Lab) at the Department of Marine Technology. In this setting, an HMD is a wearable video display with the capabilities to track the head movements of the user. The HMD technology is originally developed for Virtual Reality (VR) purposes in video games or simulators. However, as this technology has matured, it is now brought into new application areas, including real-time monitoring and control systems.

The platform for this thesis will be the model ship C/S Enterprise I (CSE1), and tentatively the ROV SF 30k. The CSE1 model ship in combination with the MC-lab has a real-time control system based on National Instruments and Compact RIO (cRIO), including a dynamic positioning (DP) function. The main task will be to implement, interface, and test a telerobotic control system for CSE1 using an HMD to control a ship-mounted camera to achieve telepresence from the ship. This involves controlling the ship to track the head movement of the operator. If successful, the thesis will also investigate implementing it on the ROV SF 30k for underwater telepresence in the sea.

Work description

1) Perform a literature review to provide background and relevant references on:
   - Telepresence, telerobotics, Virtual Reality and Augmented Reality
   - The HMD in question, Oculus Rift.
   - Setup in the MC-lab
   - Guidance and control of ocean vehicles, including relevant literature/reports on CSE1 and ROV SF 30K.

   Write a list with abbreviations and definitions of terms and concepts, explaining relevant concepts

2) Write PC interface software to manage the HMD. This should consider two vessel platforms, the CSE1 and the ROV SF 30k (both based on cRIO control systems), with the functions:
   - For CSE1 with a normal camera: Head rotations turn the vessel to the corresponding heading
   - For ROV SF 30k with normal camera: Turn the ROV heading with the head movements
   - Send head attitude information to control system.
   - Make sure software is easily convertible to other systems than CSE1.

   This involves the following work tasks:
   - Image processing using OpenCV.
   - Render images for the HMD, using the Oculus Rift SDK and OpenGL.
   - Send head attitude information to vessel (attitude) control systems.
   - Make sure the software is generic and easily transferable to other vessel control systems.

3) Interface PC software with CSE1 control system:
    - Instrumentation:
        i) Transmitting signals from PC software to cRIO vessel controller.
        ii) Receiving feedback signals and video stream from cRIO vessel controller.
    - Design the guidance (and control) system for CSE1 in Simulink.
    - Update the vessel HMI if necessary.

4) Verify the new “telerobotic” control function for CSE1:
    - Perform a set of relevant tests using HIL simulation with cRIO and NI Veristand.
    - Test the real system in MC-lab for CSE1.

5) Interface the PC software to the ROV SF 30k control system:
    - Instrumentation:
        i) Transmitting signals from PC software to cRIO vessel controller.
        ii) Receiving feedback signals and video stream from cRIO vessel controller
    - Design the new guidance functions for ROV SF 30k based on the above functional specifications.
    - Update the vessel HMI if necessary.

6) Verify the new “telerobotic” control function for ROV SF 30k:
    - Perform a set of relevant tests using HIL simulation with cRIO and NI Veristand.

Tentatively:
7) Test the ROV SF 30k telepresence function in sea.

Guidelines

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present his personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the various steps in the deduction.

The report shall be organized in a rational manner to give a clear exposition of results, assessments, and conclusions. The text should be brief and to the point, with a clear language. The report shall be written in English (preferably US) and contain the following elements: Abstract, acknowledgements, table of contents, main body, conclusions with recommendations for further work, list of symbols and acronyms, references, and optional appendices. All figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. *natbib* Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct. Such practice is taken very seriously by the university and will have consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed upon.

The thesis shall be submitted with a printed and electronic copy to 1) the main supervisor and 2) the external examiner, each copy signed by the candidate. The final revised version of this thesis description must be included. The report must appear in a bound volume or a binder according to the NTNU standard template. Computer code, pictures, videos, data series, and a PDF version of the report shall be included electronically.

Start date:	14 January 2015	**Due date:**	10 June 2015

Supervisor: Prof. Roger Skjetne
Co-advisor(s): PhD candidates Andreas Reason Dahl and Mauro Candeloro

Roger Skjetne
Supervisor

Abstract

Maneuvering remotely operated vehicles (ROV) can be challenging, especially when the mission requires the use of a manipulator arm. Today such operations are usually conducted by two operators: a pilot in charge of keeping the position of the ROV and one operating the manipulator arm. This thesis proposes a telepresence head-mounted system that allows the user to both have a first-person perspective of the operation and to control the vehicle position at the same time, leaving both hands free.

To accomplish the aforementioned goal, a head mounted display (HMD) called Oculus Rift is utilized. This device is equipped with sensors that track the users head movement, and a display that can provide 3D images to the user as well as sensor feedback, providing a small degree of augmented reality.

A gesture based guidance and control system is designed: Gesture based guidance transforms measured head movement to a speed reference, the control system then ensures that the vehicle follows the reference. Mathematical models that allow to design control systems for the vehicles are also elaborated.

The software solution for Oculus Rift and its interface with the vehicle control systems, along with technical specifications and solutions for the equipment involved is presented in the thesis.

Simulations, hardware in the loop, model scale and full scale test are conducted to collect data, verify the system and establish conclusions about this technology. Model scale tests are conducted at the Marine Cybernetics laboratory at the Department of Marine Technology, with the model ship Cybership Enterprise 1, while full scale tests are performed with the NTNU ROV SUB-fighter 30k from deployed from R/V Gunnerus.

Model and full scale tests shows that a head mounted display telepresence system can be successfully used to control and maneuver a ROV. The system is suggested as an alternative way of conducting combined ROV - manipulator operations. For this purpose, further development and testing is needed.

Sammendrag

Undervannsoperasjoner kan være utfordrende, spesielt når posisjonering av undervannsfarkoster må kombineres med bruk av manipulatorarm, noe som er en vanlig operasjon i undervannsindustrien. I dag blir slike operasjoner gjennomført ved at en pilot er ansvarlig for å holde farkosten i posisjon, mens en annen operatør er ansvarlig for manipulatorarmen. I denne masteroppgaven blir det foreslått et system som har mulighet til å styre undervannsfarkoster med hodebevegelser samtidig som det gir førstepersons perspektiv fra farkosten, noe som vil frigjøre operatørens hender.

For å oppnå dette blir en hodemontert monitor av typen Oculus Rift benyttet. Denne monitoren er utstyrt med sensorer som måler hodevinklene til brukeren, og en monitor kan vise frem 3D video samt grafikk til brukeren.

Et kontrollsystem der hodevinkler blir transformert til ønsket hastighet og posisjon for farkosten er implementert. Matematiske modeller av involverte farkoster som underbygger designet av kontrollsystemer er presentert.

Programvaren skrevet for Oculus Rift for å hente sensormålinger, hente sanntids video, samt dens integrering med kontrollsystemer for de relevante farkostene og tekniske spesifikasjoner og løsninger blir presentert.

Simuleringer, modellforsøk og fullskalaforsøk er gjennomført for å verifisere systemet. Modellforsøk er gjennomført med skipet Cybership Enterprise 1, og fullskala forsøk med NTNUs SUB-fighter 30k.

Forsøkene viser at det er mulig å kontrollere en undervannsfarkost ved hjelp av hodebevegelser der armene er fullstendig fri til andre operasjoner en styring. Flere kontrollmetoder er beskrevet og sammenlignet. Videre forsøk og utvikling av systemet anbefales for å undersøke om dette virkelig er en praktisk måte å utføre undervannsoperasjoner med farkost og manipulatorarm.

Acknowledgments

The work for this thesis has been conducted for the Department of Marine Technology at the Norwegian University of Science and Technology, with the use of their laboratory facilities at Marine Cybernetics laboratory (MC lab) and R/V Gunnerus, using their model ship Cybership Enterprise 1 and remotely operated vehicle (ROV) SUB-fighter 30k.

I would like to thank my supervisor Roger Skjetne for giving me the opportunity to work on such an exciting master thesis.

My co-supervisors Andreas Reason Dahl and Mauro Candeloro for great supervision and guidance throughout this final year at NTNU. Andreas for all the help with model scale tests and work in the MC lab, and Mauro for his guidance with regards to the full scale tests.

Others who have contributed to the thesis are: Torgeir Wahl, his custom device driver was of great importance for the model tests. The HAROV group, especially Martin Ludvigsen for contributing with ideas and feedback and the crew of R/V Gunnerus and TBS for their contribution to the full scale test.

At last I wish to thank my co-students, especially Christan Møgster and Fredrik Sandved for contributing with constructive discussion and support throughout my final years of study.

Contents

Nomenclature

API	Application programming interface
AR	Augmented reality
AUR	Applied Underwater Robotics laboratory
CB	Center of buoyancy
CG	Center of gravity
cRIO	CompactRIO
CSE1	Cybership Enterprise 1
DK2	Development Kit 2
DOF	Degree of freedom
DP	Dynamic positioning
ESC	Electronic speed control
FPGA	Field-programmable gate array
GNC	Guidance, navigation and control
GPS	Global positioning system
GUI	Guided user interface
HIL	Hardware in the loop
HMD	Head mounted display
HUD	Heads up display
I/O	Input/output
IMU	Inertial measurement unit
MC lab	Marine Cybernetics laboratory
NED	North east down
NI	National Instruments
NTNU	Norwegian University of Science and Technology

OpenCV	Open source computer vision
OpenGL	Open graphics library
OS	Operating system
PD	Proportional derivative
PID	Proportional integral derivative
PWM	Pulse-width modulation
QTM	Qualisys Track Manager
RAM	Random access memory
ROV	Remotely operated vehicle
SDK	Software development kit
SF-30k	SUB-fighter 30k ROV
SNAME	Society of Naval Architects and Marine Engineers
TCP	Transmission control protocol
UDP	User datagram protocol
VI	Virtual instrument
VR	Virtual reality
VSP	Voith Schneider propeller

Symbols and Notation

Forces and moments	
X	Force in surge
Y	Force in sway
Z	Force in Heave
K	Moment in roll
M	Moment in pitch
N	Moment in yaw
$\boldsymbol{\tau}$	Force vector
Positions and attitude	
N	North position
E	East position
D	Down Position
x	x position MC lab
y	y position MC lab
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\boldsymbol{\eta}$	Position vector
Velocities	
u	Surge velocity
v	Sway velocity
w	Heave velocity
p	Roll angular velocity
q	Pitch angular velocity
r	Yaw angular velocity
$\boldsymbol{\nu}$	Velocity vector
Other	
m	Mass
x_g	Center of gravity in x direction
y_g	Center of gravity in y direction
z_g	Center of gravity in z direction
I_{ij}	Moment of inertia around axis. $i, j = x, y, z$
W	Weight
B	Buoyancy
$\bigtriangledown$	Vehicle displacement
g	Gravity constant
ρ	Density of water

Subscripts and superscripts

$\{b\}$	Body reference frame
$\{n\}$	NED reference frame
$\{mc\}$	MC lab reference frame
$\{oc\}$	Oculus Rift measurement
$\{d\}$	Desired state
$\{k\}$	Discrete state
$\{\hat{}\}$	Estimated state
$\{\tilde{}\}$	Error state
$\{\cdot\}$	Time derivative

Chapter 1

Introduction

The idea of using head mounted displays (HMD) for various purposes has long been around in science fiction culture, as in Figure 1.1 where Michael J. Fox is seen wearing a HMD to watch TV in the movie Back to the future II (1989), where the plot is a time travel to year 2015.

The movie got this part of futuristic technology quite right, though they missed with predicting hoovering skateboards and time machines. Today, some of the largest technology companies in the world (Google, Facebook, Sony, Samsung and Microsoft) are looking into producing HMDs.

The idea for this master thesis originally occurred after seeing a video (Urke, 2014) of a tank from the Norwegian Army being driven with hatches closed, but able to have a 360 degree view around the tank using an Oculus Rift and a camera system.

The idea of using the same type of setup for remotely operated vehicles (ROV) was a natural consequence. Operating ROVs can be challenging, especially when combining the use of a manipulator arm and maneuvering the ROV at the same time. Nowadays such operations are usually conducted by two operators: a pilot

Figure 1.1: Michael J. Fox wearing a HMD in the movie Back To the Future II (1989) (courtesy: Zemeckis (1989))

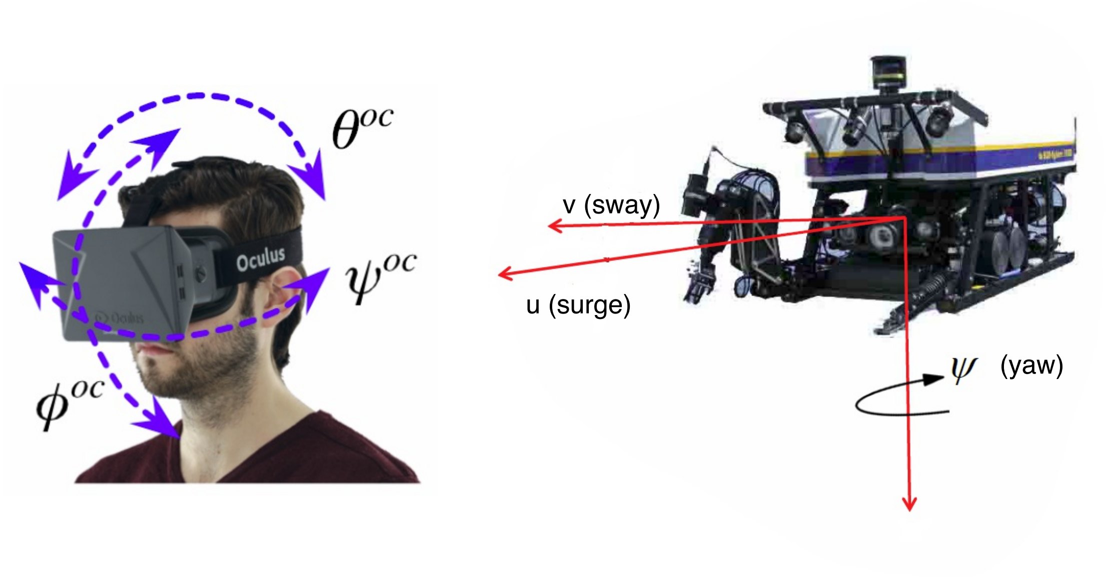


Figure 1.2: Head motion transferred to ROV motion

is tasked with keeping the ROV in position, while another operator is tasked with operating the manipulator arm. The fact that the operator of the manipulator arm has no direct control over the ROV position, is part of why this is challenging.

With a HMD that senses head movements, the ROVs position could be controlled, while at the same time providing the operator with a first person perspective of high quality and both hands free to operate the manipulator arm. In Figure 1.2 a sketch of the setup is visible, with degrees of freedom (DOF) of the head used to control the ROV.

1.1 Head Mounted Display

A head mounted display is, as the name implies, a display that is attached to your head. Usually it comes with one display for each eye, which makes it possible to display 3D-images. In addition it is common that a HMD has the ability to track the head movement of the user (see Figure 1.2) in at least roll, pitch and yaw. This combination of 3D displays and head tracking has made HMDs very popular for virtual reality (VR) simulators and video games. This is still an emerging field, as there are no mass consumer version of a HMD available for the public, though this will change soon, as Oculus Rift has announced that their consumer version will be launched in the first quarter of 2016 (Robertson, 2015).

1.1.1 Definitions

Before going further with explaining what a HMD is, a few terms should be clarified.

- **Virtual Reality (VR):**
  "The computer-generated simulation of a three-dimensional image or environment that can be interacted with in a seemingly real or physical way by a person using special electronic equipment, such as a helmet with a screen inside or gloves fitted with sensors.", (Oxford University Press, 2015)
- **Augmented Reality (AR):**
  "A technology that superimposes a computer-generated image on a user's view of the real world, thus providing a composite view." (Oxford University Press, 2015)
- **Telepresence**:
  "The use of virtual reality technology, especially for remote control of machinery or for apparent participation in distant events."(Oxford University Press, 2015)

1.1.2 Applications

The concepts of VR, AR and telepresence are all really relevant for HMDs.

VR is typically used for computer games and simulators, and this is perhaps the most known use for a HMD. Steuer (1992) defines VR as a combination of interactivity and vividness, where the fictional Star Trek Holodeck[1] is used as an example that has full score on both scales, and a regular book has the lowest possible score on both scales.

Typical AR concepts are environments where there is a wish to improve the user environment digitally. An example of an AR application is the new F35 fighter

[1] A room that simulates objects and people with holograms and material

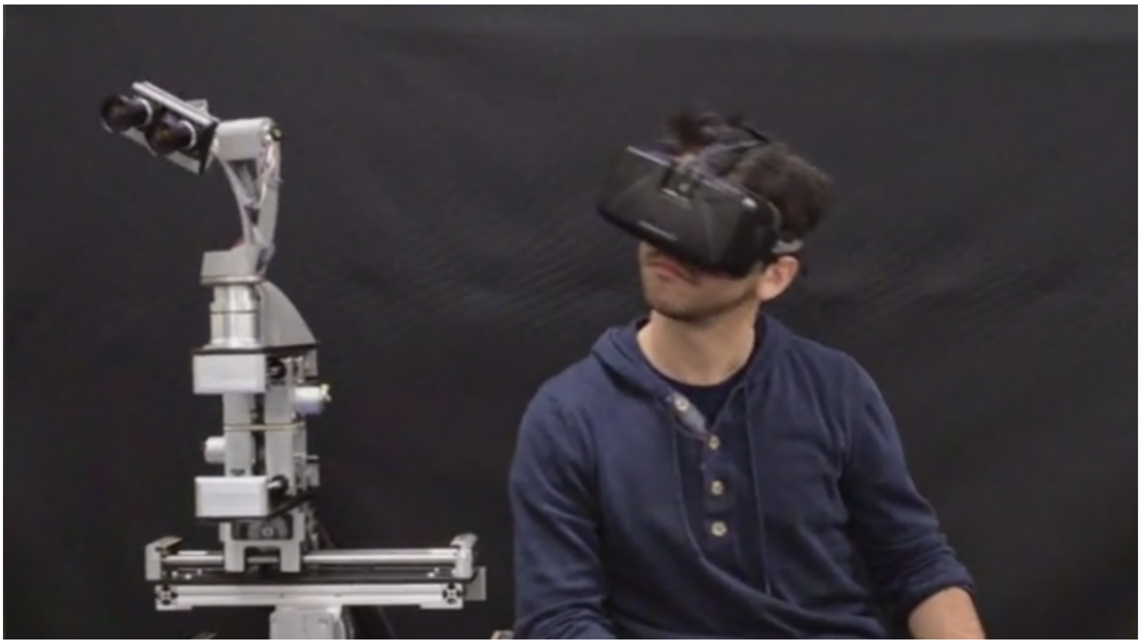


Figure 1.3: DORA project personal avatar, courtesy of DORA (2015)

pilot helmet shown by Lockheed Martin (2015). This helmet HMD has the ability to augment the pilot environment displaying flight info, tracking targets and even make the fuselage see through for the pilot combining cameras and sensors that track the head movement.

A more affordable approach to AR using the Oculus Rift HMD is shown by Steptoe et al. (2014). The demonstration video of his multipurpose AR system is available in the video Steptoe (2014).

Ackerman (2015) shows an implementation of a telepresence avatar, see Figure 1.3. This robot will mimic the movements of the operator, following every move with low latency, a demonstration video is available by DORA (2015).

Another way to implement telepresence is with a filming in all directions simultaneously. This is typically achieved by filming with several camera lenses and then digitally stitching the images together to a spherical image that virtually encircles the user, cameras of this type are referred to as 360 cameras. The Norwegian company Making View, which made the Oculus Rift system for the tanks (Urke, 2014) also produces their own 360 camera, as described by Drangeland (2015).

On May 28, 2015 Google launched their virtual reality investment called Google Jump (Bohn, 2015). Here they announced that they will combine their already available HMD Google Cardboard (Figure 1.4d), with a system of 16 cameras mounted in a circle, filming in different directions simultaneously, the 16 videos is then uploaded to Google servers where advanced algorithms will stitch together and synchronize the raw video into one spherical video. As the picture is wrapped around the user virtually, no system has to physically move in order to allow the user to look around with 360 camera telepresence.

1.1.3 Manufacturers

Today, several producers are developing HMDs. With the most well known being the Oculus Rift system that we are using as a test platform in this thesis, Oculus Rift will be thoroughly presented in Chapter 2.1. Sony, Google and Microsoft alternatives can be seen in Figure 1.4

The Sony HMD is really more of a wearable display, meant for watching traditional media like movies and TV. It does not feature any sensors to capture the users head movement, thus it would not be able to provide a VR experience in sense of interactivity.

Google Glass is a more experimental wearable technology that interacts with the user in everyday life, displaying information and graphics to the user as an overlay of real life. In January 2015 Google Inc. (2015) announced that they would stop selling Google Glass, but hinted that they might release a new version at a later stage.

Google cardboard is actually just an empty shell made from cardboard. Here the VR technology is provided by an Android smart phone that contains the processors, sensors and display.

Microsoft (2015*a*) Hololens is an AR HMD that creates holographic images that the operator can interact with in their the users environment. Microsoft claims that this is not traditional AR or VR but mixed reality. However, it clearly satisfies the definition of an AR device. The release date is not known as of May, 2015.

(a) Sony HMZ-T3W

(b) Google Glass

(c) Microsoft Hololens

(d) Google Cardboard

Figure 1.4: HMDs commercial solutions (courtesy: Sony, Google and Microsoft)

1.2 Methodology

The control systems in this thesis are designed using the 3-step process described by Sørensen (2012) with simulation, hardware-in-the-loop (HIL) testing conducted before model and full scale testing.

1.2.1 Mathematical Modeling and Simulation

To test and implement control systems, mathematical modeling and simulations are commonly used to accomplish the best results possible. A typical design process starts with creating a mathematical model for the physical system or finding an existing model based on first principles (Kirkeby, 2010).

The next step is then to implement this model for computer simulations, typically using either MATLAB or Simulink software (Svendby, 2007). In these offline[2] simulations, the control design is validated.

The simulation models may vary in complexity: A vessel may either be simulated with a model taking only the basic kinetics and kinematics into account or with a model of increased complexity, by also considering environmental disturbances, thruster dynamics, etc.

[2] Offline meaning completely disconnected from real-operations and hardware/software of the actual system

1.2.2 Hardware In the Loop

The next step in the design process is to conduct HIL tests. These simulations are conducted with the control system running on the vessel specific computer hardware. The models tested can have varying complexity, however, the goal is to simulate the vessel and environment as accurately as possible, including sensor characteristics, etc. This method of testing the control system, as opposed to going directly from simulations to full scale testing, reduces commissioning time and risk of failure.

1.2.3 Model Scale Tests

If a thorough HIL test has been successfully accomplished, the implementation of the control system on the actual vessel should be seamless. Ideally, the only difference from a HIL test is that instead of a simulation model, the vessel actuators are connected to the controller.

The model scale tests of this thesis are conducted in the Marine Cybernetics laboratory (MC lab) at Department of Marine Technology, with the model vessel Cybership Enterprise 1 (CSE1).

1.2.4 Full Scale Tests

At last, full scale tests at sea with NTNU's ROV SUB-fighter 30k (SF 30k) are conducted to commission and verify the control systems previously simulated and tested on a model scale. Data is collected and analyzed to evaluate the control algorithms and software.

1.3 Contributions

- A software that shows a live video feed on the Oculus Rift video system, while sending the Oculus Rift sensor data to a control system.
- Guidance systems, for feed-forward force control, velocity control and position control. Takes the measurements from the Oculus Rift and turns it into desired position and velocity.
- Development of new control software and other technical solutions for model scale tests in the MC lab with the model CSE1.
- Implementation of Oculus Rift as an input device on the AUR-lab control system for ROVs, leading to full scale tests with NTNU's ROV SF 30k.
- Critical analysis of results in connection to possible employment of the system for commercial use and suggestion for further development of the technology in connection with the underwater vehicles field.

1.4 Thesis Outline

- **Chapter 2:** describes the HMD in use, the model ship CSE1, the general technical MC lab setup and the ROV SF 30k.

- **Chapter 3:** presents the software produced, Oculus Rift software is elaborated, setup of the CSE1 software and integration of Oculus Rift as an input device for SF 30k is presented.

- **Chapter 4:** focuses on the mathematical modeling of ocean vehicles. A model for the vehicles CSE1 and the ROV SF 30k is presented.

- **Chapter 5:** presents guidance and control theory for the setup. Emphasis is put on the guidance for transforming head motion into desired states for the system to be controlled.

- **Chapter 6:** presents the results of simulations, HIL testing, model scale tests and full scale tests.

- **Chapter 7:** sums up the findings and make recommendations for further work.

Chapter 2

Equipment

In this chapter, the different set of platforms utilized in the implementation of a telepresence systems is presented. This mainly consists of the HMD Oculus Rift, the model ship CSE1 from the MC lab and ROV SF 30k NTNU's own work class ROV.

The main emphasis of the chapter is to present the technical specifications and limitations of the different pieces of equipment. CSE1 is presented more thoroughly than SF 30k since the latter is well documented in Berg (2012) and Dukan (2014).

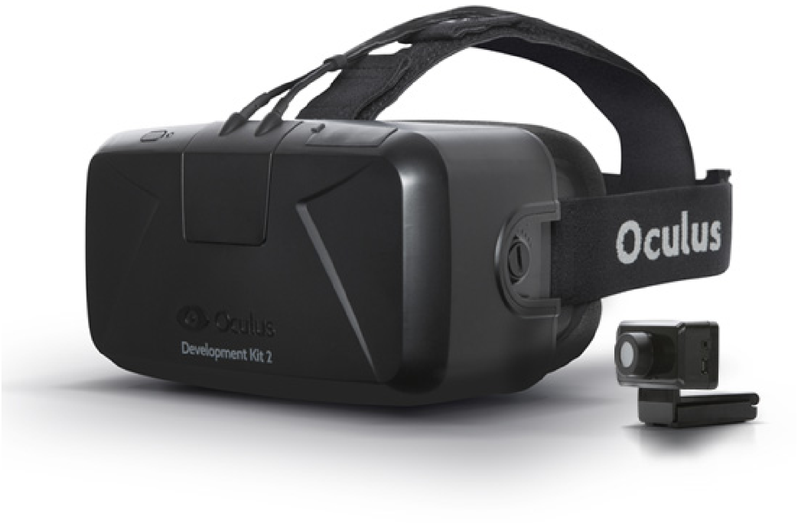


Figure 2.1: Oculus Rift DK2 (courtesy: Oculus VR LCC)

2.1 Oculus Rift

The HMD that is used for this thesis is the Oculus Rift system, as seen in Figure 2.1. This is one of the most well known systems, though it is still only available as a development kit. The consumer edition is due for release in the first quarter of 2016 (Robertson, 2015)

This section is closely based on the what was written in the project thesis Follestad et al. (2014).

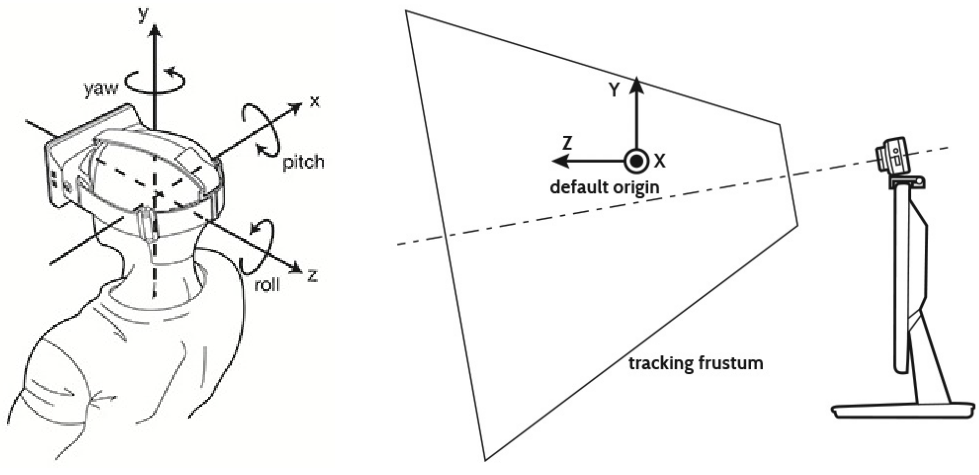
2.1.1 Hardware

The HMD hardware is divided into two main groups. Sensors for tracking head movement and monitors for displaying video to the user.

Sensor System

The Oculus Rift has magnetometers, accelerometers and gyroscopes for tracking head attitude, at a high refresh rate (as seen in Figure 2.2a). This is required for all VR applications, as the ability to look around in the virtual world is essential for the interactivity aspect.

In addition to the attitude information, it is possible to measure the relative position in the x, y and z plane. This is accomplished by an infrared tracker, seen in Figure 2.2b. This also helps for the accuracy of the attitude measurements. For more info about the technical specifications see the development kit specification Oculus VR, LCC (2014*b*).



(a) Oculus Rift attitude sensors

(b) Oculus Rift positioning

Figure 2.2: Oculus Sensors (Courtesy: Oculus VR, LCC (2014*a*))

Video

The video delivery system works by showing two different videos individually to each eye, supplying each eye with its own high definition (HD) monitor.

Oculus Rift monitors need to cover the whole field of view of the user, to create realistic VR environments. To accomplish this, the producers apply wide angle lenses. The lenses pincushion distort the video image, see Figure 2.3a. To counteract this effect, the video is digitally barrel distorted, see Figure 2.3b. In this way, the image shown to the Oculus Rift operator does not have any distorting effects.

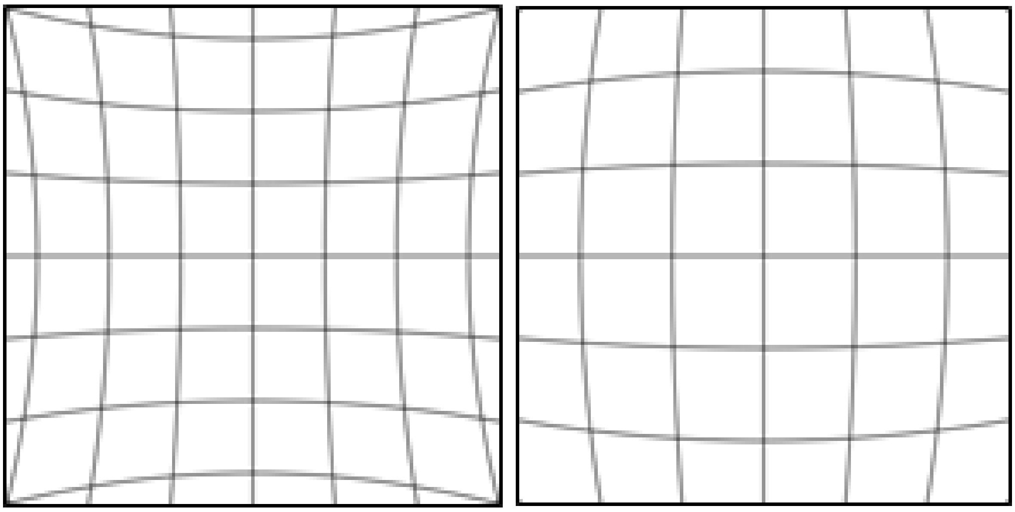
For the Oculus Rift to show 3D video; stereo video signal should be sent to the displays. This requires two cameras filming, one for each eye.

2.1.2 Software Development

There is an extensive software development kit (SDK) accompanying the Oculus Rift. The core of the SDK consists of source code and binary libraries. Moreover, documentation, sample programs and other tools such as test applications are available.

Very much of the emphasis in the SDK is on programming three dimensional graphical environments. The work described here; in fact what is most relevant for our project is to deliver video feed to the monitors and read output from the sensors. Sensor output is read by a dedicated piece of code, in which there are implemented functions for. There are functions for reading both Euler angles and quaternions. Software for Oculus Rift is written in C++ language.

The software development in this thesis is based on the method outlined and examples given by Davies et al. (2015*b*). The specific software developed for this thesis is covered in Section 3.1.



(a) Pincushion (b) Barrel

Figure 2.3: Illustration of pincushion and barrel distortion (Oculus VR, LCC, 2014*a*).

2.1.3 Simulator Sickness

Since HMDs and VR are relatively new commercial products, they have probably not been tested for long time usage. One of the unknown factors concerning operation with a HMD is simulator sickness. A few paragraphs about the phenomenon is included here, for future reference.

The sickness stems from conflict between the visual and bodily senses. Symptoms are very similar to what is normally known as motion sickness, and they could be described as disorientation, nausea and eyestrain. It is very difficult to predict who will get affected by simulator sickness as it is very individual, as for example car sickness also is. Since developers will get more and more used to wearing the Oculus, they are unsuitable test subjects for simulator sickness (Yao et al., 2014).

A factor that suggest that the problem will not be as big for this setup, is that the head movement will determine the movement of the system hopefully this will have the same effect as driving a car versus being a passenger with regards to car sickness. However, it is difficult to conclude, and formal structured series of tests should be conduced to commission the system.

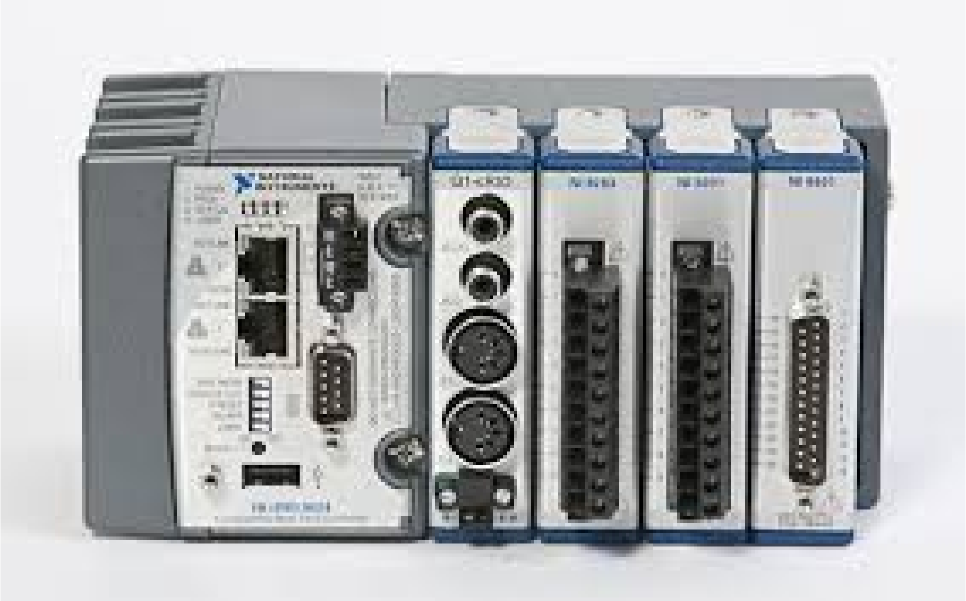


Figure 2.4: CompactRIO-9024 (courtesy: National Instruments (2015))

2.2 CompactRIO Embedded Controller

The real-time controller fitted on CSE1 is a National Instruments (NI) CompactRIO (cRIO), as seen in Figure 2.4. This is described as a reconfigurable embedded control and acquisition system by the producer National Instruments (2015). The cRIO features interchangeable input/output (I/O) modules that can be reached by a field-programmable gate array (FPGA). The system runs real-time control systems, programmed either as National Instruments own program LabVIEW, or Simulink code via the software NI VeriStand.

The cRIO fitted on CSE1 is of the model 9024 and is configured with two I/O modules. One NI-9215 analogue in module and NI-9474 digital out module. This allows measure analogue signals and send digital signals, like pulse-width modulation (PWM) commands. Other input is available through the two Ethernet ports.

The cRIO is also utilized for the HIL tests conducted. When this is the case, the I/O is replaced by a simulation model.

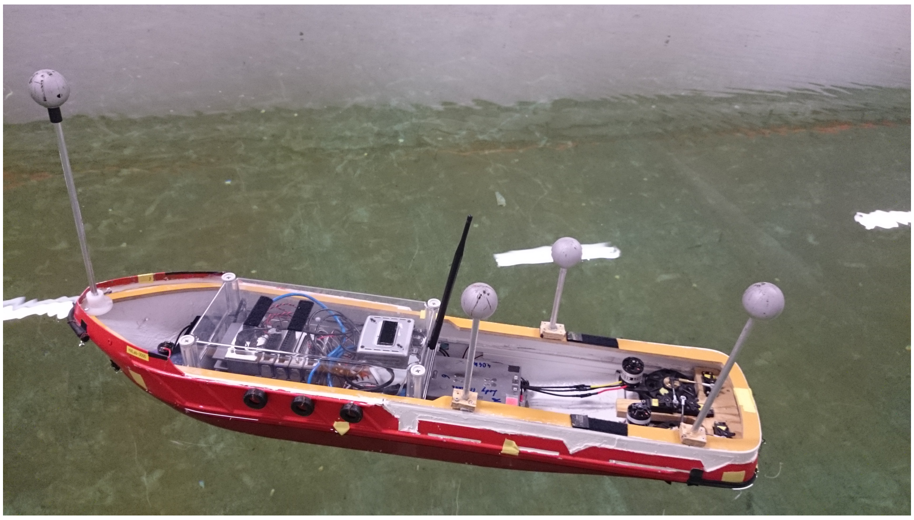


Figure 2.5: CSE1 in the MC lab

Main Dimensions		
Displacement	∇	0.01479 [m^3]
Mass	m	14.79 [kg]
Length o.a	L	1.10 [m]
Breadth	B	0.24 [m]
Depth	D	0.07 [m]

Table 2.1: CSE1 main dimensions

2.3 Cybership Enterprise 1

CSE1 is a model ship stationed at the MC lab. It was originally built and commissioned as a part of a master thesis (Skåtun, 2011) and further developed by (Tran, 2014).

Recently CSE1 has been updated (NTNU, 2015) as it is now being used as a part of the laboratory tutoring in the course TMR4243 - Marine Control Systems 2 at NTNU, were students have the opportunity to perform model tests of control applications, such as dynamic positioning (DP).

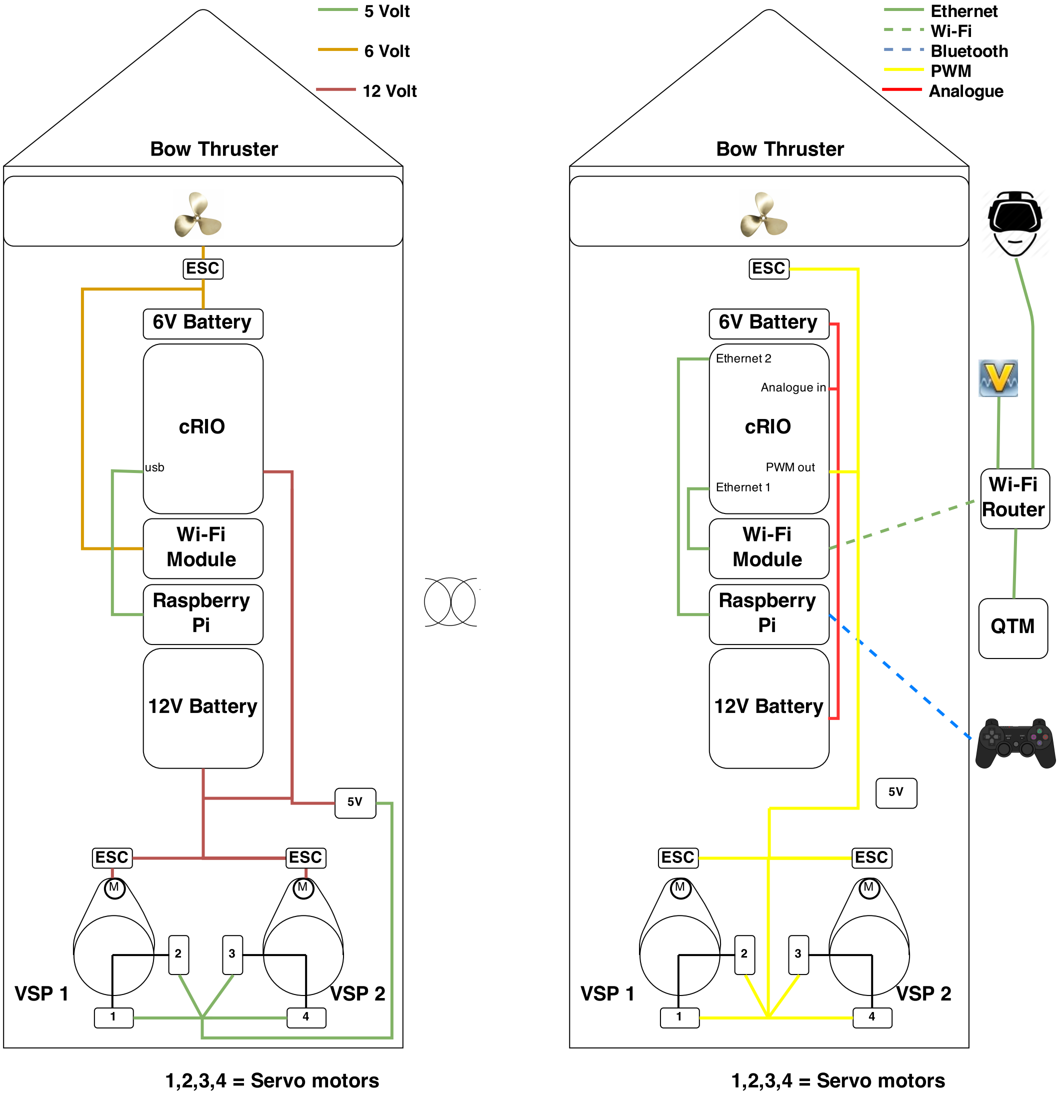
2.3.1 Technical Specifications

The main dimensions of CSE1 are described in Table 2.1

Power and signal topology

The electrical wire topology of CSE1 can be seen in Figure 2.6 (a).

CSE1 is powered by two batteries. A 12V battery powers the main thrusters and the on-board computer, while a 6V battery powers the bow thruster and the Wi-Fi module.



(a) Power

(b) Signal flow

Figure 2.6: Wiring and network topology

The compactRIO (see Section 2.2) embedded controller is commanding the electronic speed controllers (ESC) and servos with pulse-width modulation (PWM) signals.

2.3.2 Propulsion

The propulsion units of CSE1 consists of a bow thruster and two Voith Schneider propellers (VSP) for main propulsion, see Figure 2.5. The ship thrusters are severely more powerful than they need to be for a ship of this size. Thus, the thruster effect has to be saturated.

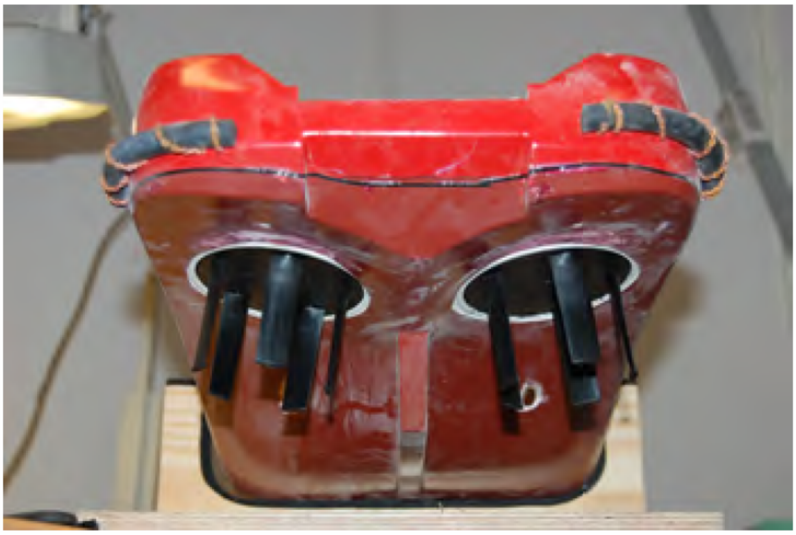


Figure 2.7: VSP propellers from as installed in CSE1 (courtesy: Skåtun (2011))

Voith Schneider Propeller

VSPs is a propeller type invented by Ernst Schneider and patented in the US by Schneider and Mueller (1941). This propeller design has its advantages in the ability to deliver propulsion power in any direction almost instantaneously. This makes it a popular choice for tug boats, some offshore supply vessels also feature VSPs. Advantages claimed by producer Voith GmbH (2015) are:

- *"prompt and precise steering."*
- *"stepless, prompt and precise control of direction and magnitude of thrust."*
- *"low downtime due to low-maintenance technology."*
- *"roll stabilization during standstill as well as when sailing."*

The history of VSPs and how they work are well described by Skåtun (2011).

Bow Thruster

The bow thruster is built in a tunnel in the bow of the vessel. This thruster is used for inducing yaw motion and maneuvering in sway direction. The thruster looses effect with increasing forward speed and is thus best used for stationary operations.

2.3.3 Camera System

For setting up an Oculus Rift telepresence system, a camera system streaming live video is a requirement. For a ROV this is usually in place but for CSE1 a system has to be installed. The camera system for CSE1 is comprised of an Android mobile phone. The free application IP Webcam (Khlebovich, 2015) turns the mobile phone

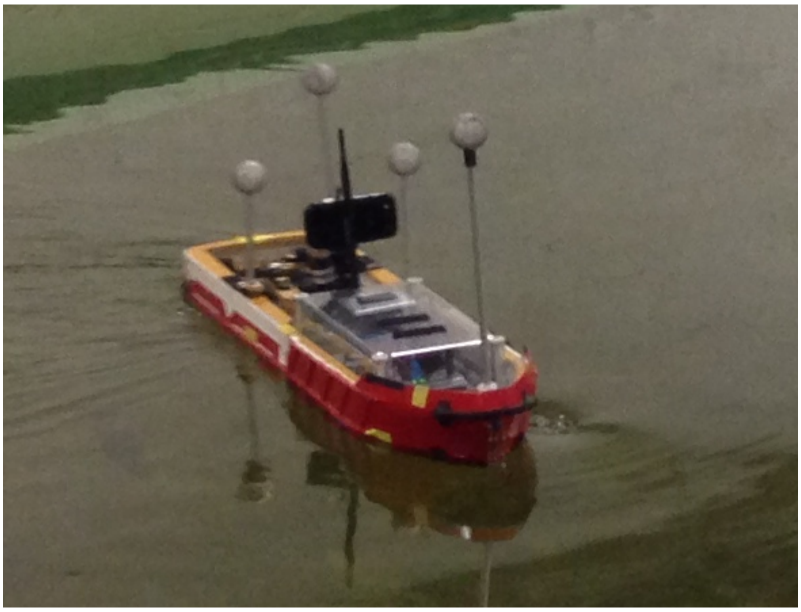


Figure 2.8: CSE1 fitted with mobile phone camera system

into an IP camera. The phone can now stream video through Wi-Fi, either on a local network or through the internet. Considering this thesis work, internet streaming gave the best result, but that could vary with a different local network setup.

This solution was chosen because of the simplicity, since an Android phone was available. However, the system has a decent performance even with a three year old Samsung 3s. For a more permanent system, a fixed IP camera should be employed.

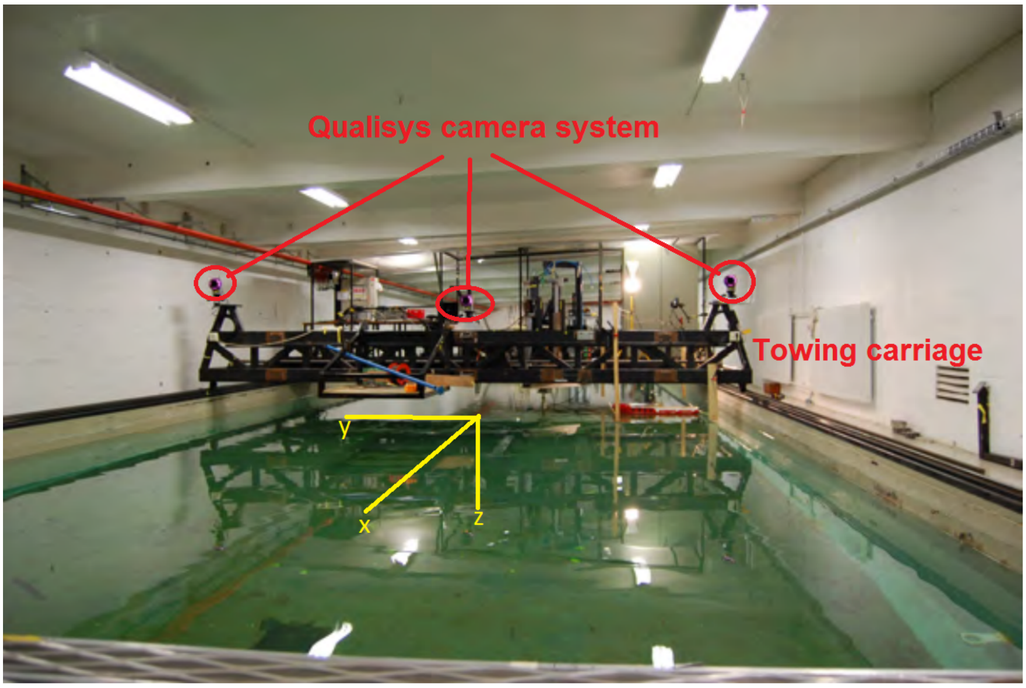


Figure 2.9: MC lab Oqus cameras used in the Qualisys mtion capture system

2.4 Marine Cybernetics Laboratory

The MC lab at the Department of Marine technology NTNU, is a laboratory equipped for testing motion control systems for marine vessels. It features a wave generator, a towing carriage and a positioning system among other things (NTNU, 2015).

2.4.1 Qualisys Motion Capture System

The positioning system in the MC lab is handled by Qualisys motion capture system. This is an optical positioning system. It works by using three infrared cameras, seen in Figure 2.9. These cameras then recognizes the silver spheres on CSE1 visible in Figure 2.5. If all three cameras have a clear line of sight to the spheres, the position of the vessel can be calculated by triangulation. The position is available with 1mm precision in six degrees of freedom.

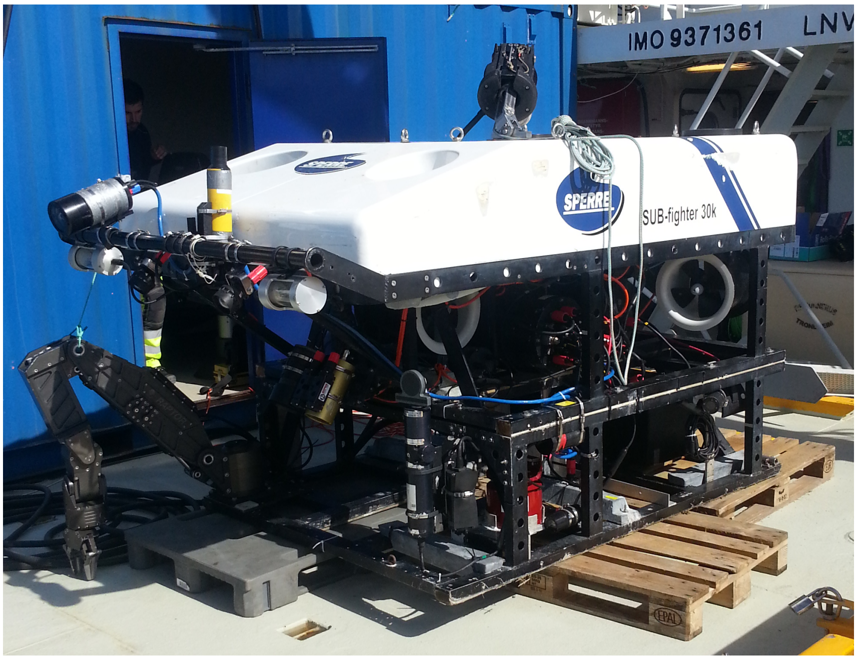


Figure 2.10: SUB-fighter 30k

Specifications	
Mass, Δ	1800 [kg]
Length	2.6 [m]
Width	1.5 [m]
Height	1.6 [m]
Depth rating	100 [m]
Payload	60 [kg]

Table 2.2: SF 30k specifications (Dukan, 2014)

2.5 SUB-fighter 30k

The ROV used for testing control systems (in this thesis) is the NTNU owned SF 30k, seen in Figure 2.10. This is a working class ROV produced by the Norwegian manufacturer Sperre. Table 2.2 shows the main specifications of the ROV. SF 30k is controlled either by manual control or through dynamic positioning. SF 30k capabilities and technical specifications are presented more in detail by Berg (2012) and Dukan (2014).

2.5.1 Equipment and Sensors

Some of the equipment and sensors that is featured on SF 30k are listed in Table 2.3. It is also possible to fit the ROV with custom sensors, as long as they are depth rated and do not exceed the ROV payload. The camera solution presented in the next subsection is an example of this.
The schematics of our setup for on the ROV mission is shown in Figure 2.11.

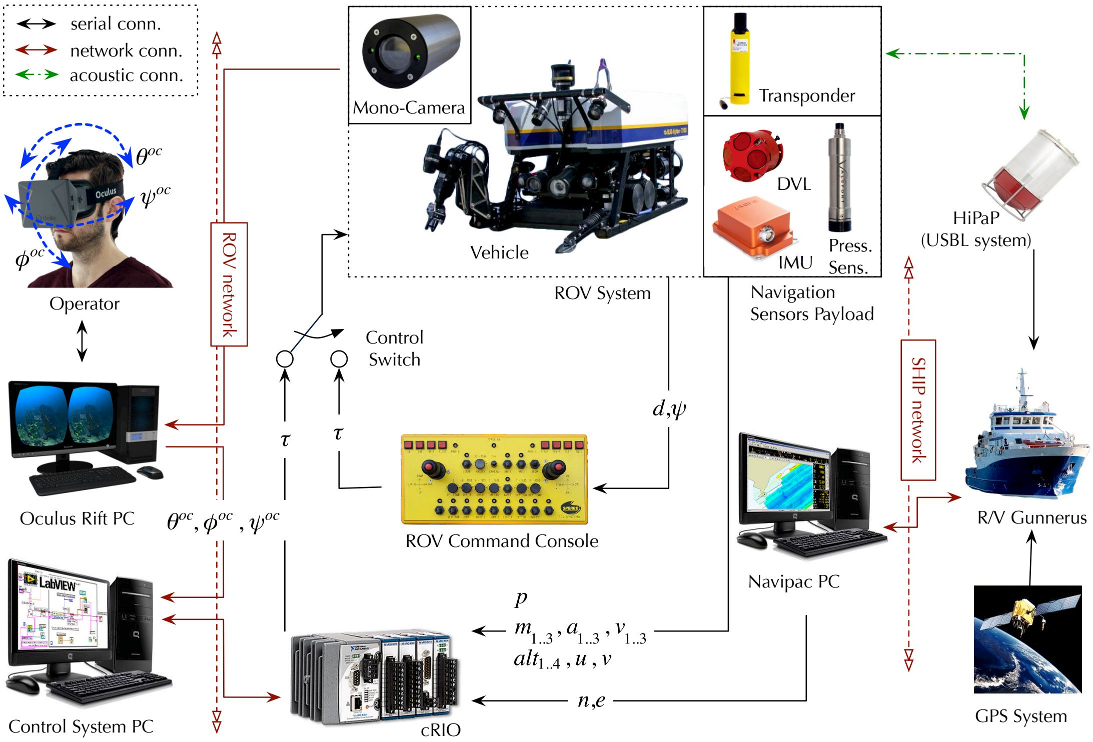
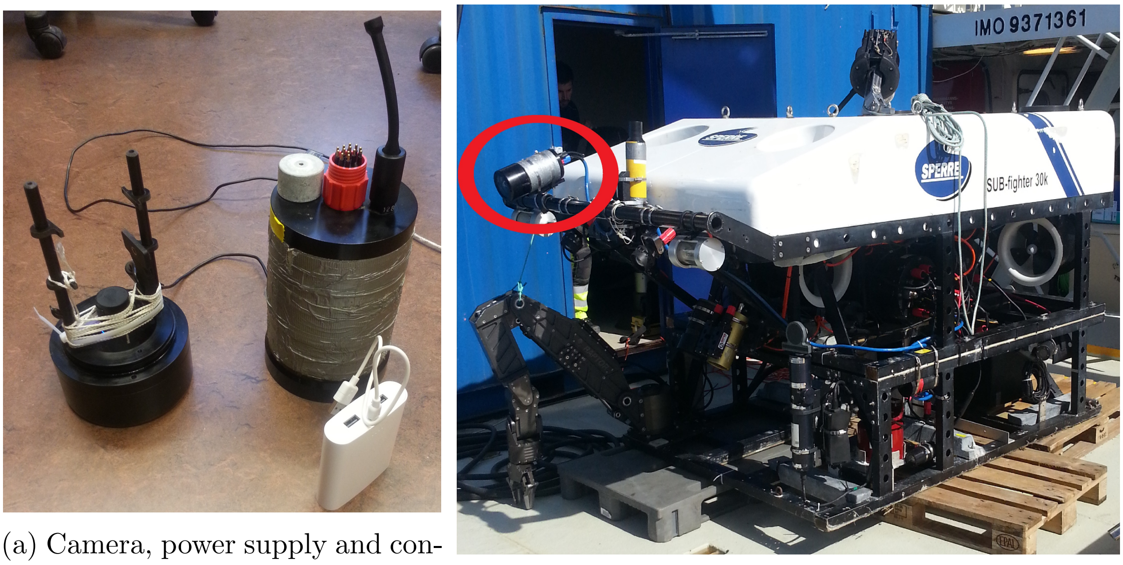


Figure 2.11: SF 30k controls system including Oculus Rift control (courtesy of Mauro Candeloro)

Sensors	Equipment
Sonar	4 Cameras
Multibeam	4 lights
Pressure Sensor	Manipulator arm
Cable tracker	Laser
Side scan	
Altimeter	
Doppler velocity log (DVL)	
Motion reference unit (MRU)	
Hydroacoustic position reference (HPR)	

Table 2.3: Sensors and equipment



(a) Camera, power supply and container

(b) Camera solution as mounted on SF 30k

Figure 2.12: Wiring and network topology

2.5.2 Camera System Modification

The Oculus Rift software demands a video source that is compatible with the OpenCV programming library. On the other hand, the video system for SF 30k runs on the SDI protocol which would require additional hardware to be compatible. This hardware was not available during the development of the this thesis. So, an alternative solution was established.

Follestad et al. (2014) runs the mini-ROV Neptunus with an OpenCV compatible video stream as a satellite through SF-30k. This same setup was used to acquire video by placing a single board Beaglebone computer in the camera bottle along with a power supply (see Figure 2.12a). The Beaglebone computer runs the ROV software from OpenROV, described in. This software sets up a video stream that the Oculus Rift software is able to process (Follestad et al., 2014)

Figure 2.12b shows the camera hardware and how it was mounted on the ROV. The camera was positioned such that it should be able to look forward and down so that the sea bottom could be a visual reference. In Figure 2.13 the schematics of the ROV-camera connections are shown.

2.5.3 R/V Gunnerus

SF 30k is operated from the research vessel Gunnerus. The ROV is controlled from a container placed on the aft deck of the vessel. The vessel has a DP system from Kongsberg Maritime and provides position measurements to the ROVs through a hull mounted Ultra Short Base Line (USBL) positioning system.

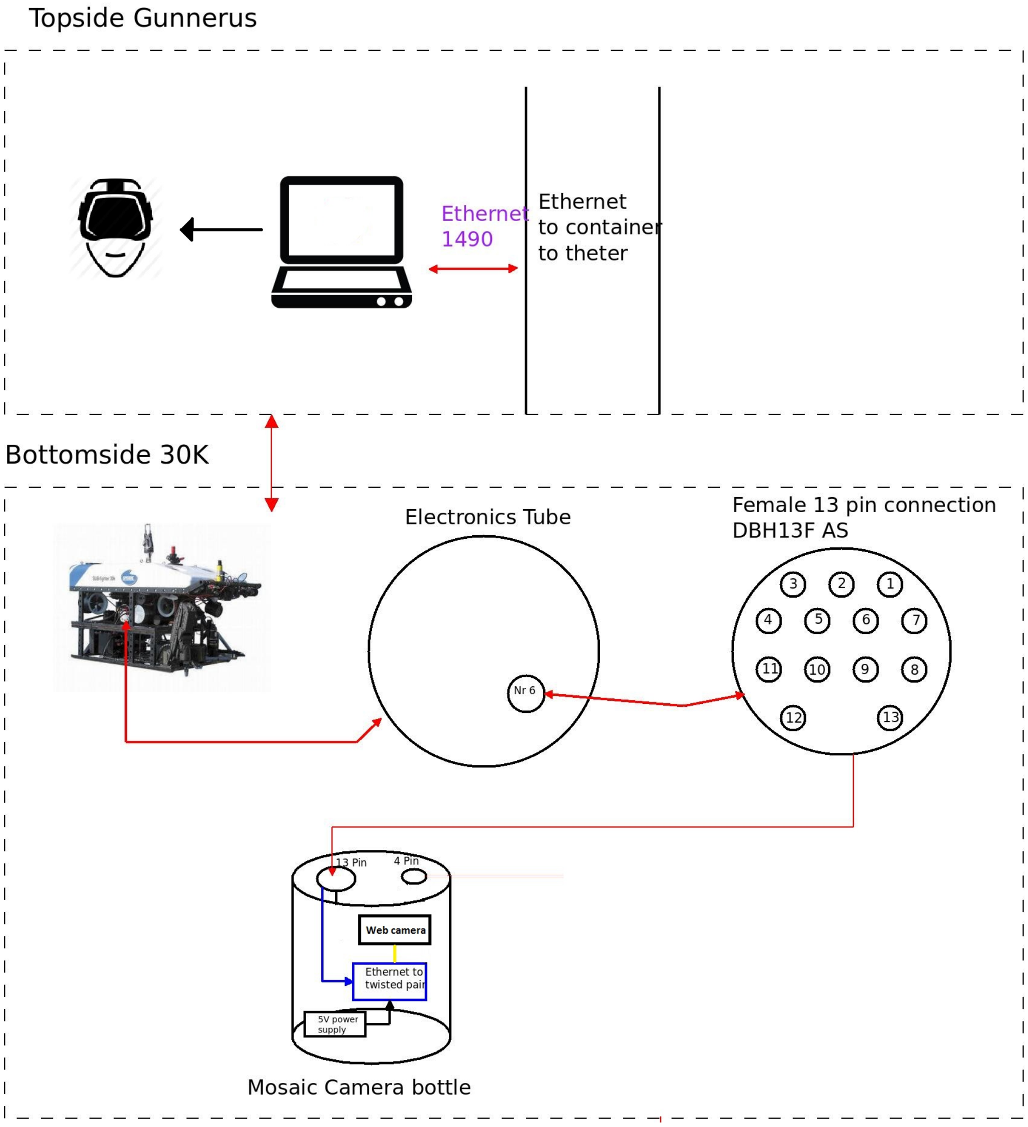


Figure 2.13: Camera-ROV connections for Oculus Rift

Chapter 3

Software

In this chapter the software developed for this thesis is presented. The main components are: Oculus Rift software written in C++, the Simulink control algorithms and NI VeriStand guided user interface (GUI) for CSE1 and the new interface software for Oculus Rift to SF 30k.

3.1 Oculus Rift

The purpose of this software is to show live video on the Oculus Rift monitor and to make Oculus Rift sensor data available for the remote control system.

One of its main goals is that the software should be compatible with multiple systems. The only input necessary for the program to run is a video stream, however, the system that is going to be controlled must be able to set up a client to receive the sensor data from the program.

3.1.1 Development Process

The Oculus Rift software is written in C++ and comes with an extensive software development kit where functions for accessing the hardware is available.

For the development of Oculus Rift software Davies et al. (2015*a*) and the accompanying example code Davies et al. (2015*b*). This software is available through the Apache License[1] (Apache License, 2007).

For the main Oculus Rift software used in this thesis, a demo program called *HighResWebcamDemo*, changed to include a TCP server that sends sensor data to the control system and takes in other video sources than a web-camera.

3.1.2 Program

The purpose of this section is to explain the different sections and functions of the C++ Oculus Rift software. The program relies on the following libraries in addition to standard C++:

- Oculus Rift SDK, for working with hardware
- OpenGL, for rendering graphics to the Oculus Rift screen
- OpenCV, for live video and image processing
- Winsock, setting up TCP server in windows

A flow chart for the program can be seen in Figure 3.1

Basic Oculus Rift programming

Once the Oculus Rift SDK is properly installed on a PC it is quite simple to work with the hardware. Davies et al. (2015*a*) created a macro that handles initializing and terminating interactions with the hardware.

[1] The Apache license basically allows the user to change the code and even use it commercially, as long as the original creators are not held responsible, their trademarked names or logos are not used and include copyright, license, notice and state changes are included.

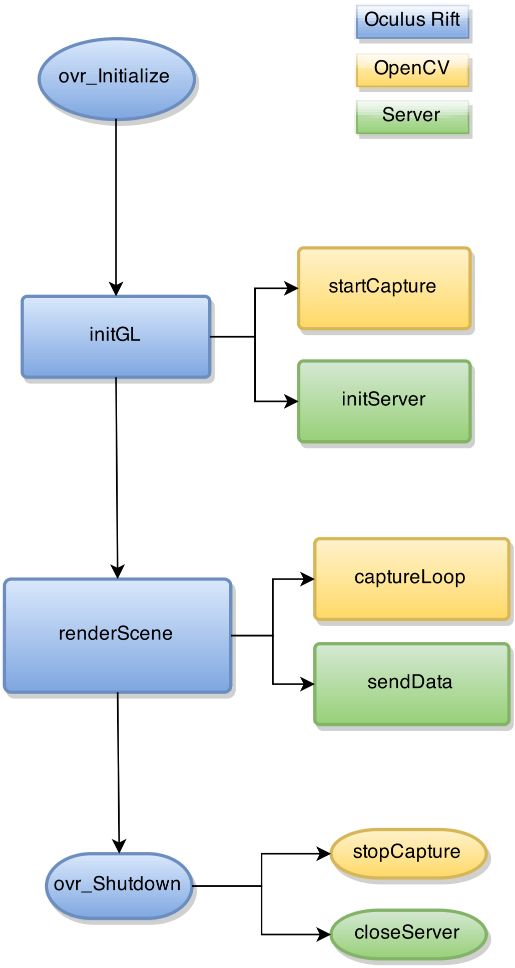


Figure 3.1: Oculus Rift application function flowchart

Sensor data is reached from basic functions. Head attitude is available both in Euler angles and quaternions. How to work with the Oculus Rift SDK is well explained in Davies et al. (2015*a*) and Oculus VR, LCC (2014*a*).

The code was modified to collect the Euler angles and make it available for the server side of the program, moreover, the head attitude is displayed on the screen to the operator of the Oculus Rift.

OpenGL (Open Graphics Library) is the graphics engine used to render images to the Oculus Rift. This part of the code is unchanged as it is already set up by Davies et al. (2015*a*).

OpenCV, Image Processing

OpenCV (Open Source Computer Vision) is a library of functions that evolves mainly around real-time computer vision. OpenCV handles the video part of the software. Video can be captured from most sources that are available on a computer, such as web-cameras and IP camera streams, then the video is then processed such that OpenGL can render the image on the Oculus Rift screen.

As video is usually captured at around 30 frames per second and the Oculus Rift renders an image at up to 75 Hz, it is important that OpenCV and OpenGL work on different loops, where OpenGL can collect the available image from OpenCV and does not have to wait for a new image from the camera before rendering graphics to the Oculus Rift screen.

Data transfer

To add server capability in order to be able to send sensor data over TCP protocol, Microsoft (2015*b*) the Windows Socket library has been utilized. Here three functions are implemented into the example code to initialize the server, send the data and terminate the connection. The function to send data is inserted together with the render scene function thus sending data at the same rate as the screen is updated, which can be as much as 75 Hz. Notice that with the server up and running 50 Hz is the experienced performance.

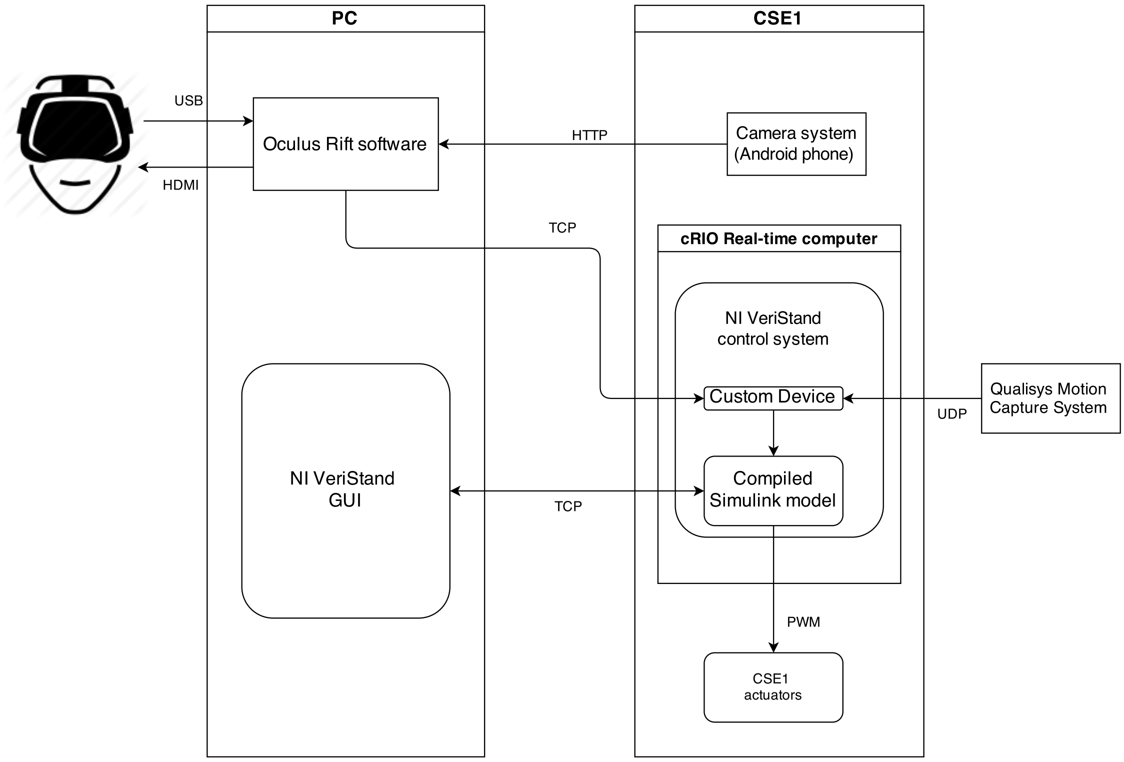


Figure 3.2: Software/hardware overview with protocols for CSE1

3.2 Cybership Enterprise 1 Control Software

The control systems for CSE1 are written in Simulink and the National Instruments software Ni VeriStand.

Simulink takes care of the control algorithms and is then compiled into a model that NI VeriStand can use for real-time control.

In Figure 3.2 the topology of the system can be seen, along with the communication protocols utilized.

3.2.1 Simulink Control Algorithms

For CSE1 the control algorithms are implemented in Simulink. Which is a block diagram environment for simulation and model based design as described in MathWORKS (2015).

The greatest advantage is the combination of the ability to run offline simulations, that can lead to the creation of an embedded system by reusing almost the same code.

The main difference between an offline simulation and an embedded system (with respect to the block diagram) is that the embedded system has to receive its inputs from external sources (e.g position measurements). When generating code for NI VeriStand special input blocks are used.

These blocks can be connected to different I/O or to the user interface, to ensure

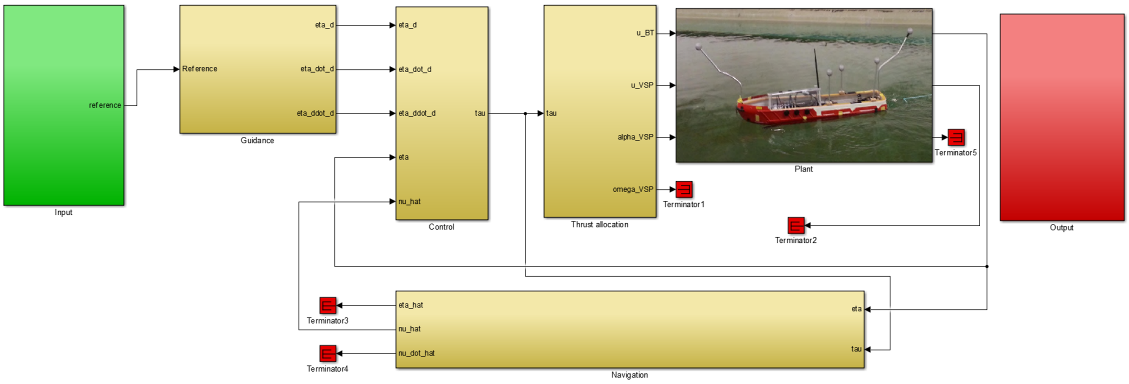
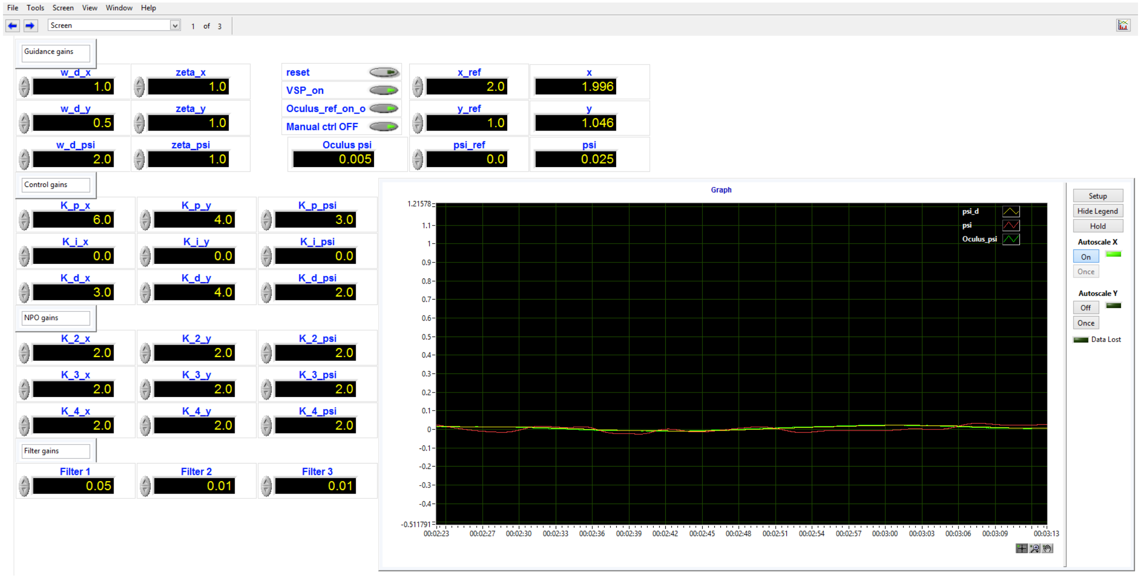


Figure 3.3: Simulink diagram for simulation of CSE1 model



Guidance gains

w_d_x 1.0
zeta_x 1.0
w_d_y 0.5
zeta_y 1.0
w_d_psi 2.0
zeta_psi 1.0

reset
VSP_on
Oculus_ref_on_o
Manual ctrl OFF
Oculus psi 0.005

x_ref 2.0
x 1.996
y_ref 1.0
y 1.046
psi_ref 0.0
psi 0.025

Control gains

K_p_x 6.0
K_p_y 4.0
K_p_psi 3.0
K_i_x 0.0
K_i_y 0.0
K_i_psi 0.0
K_d_x 3.0
K_d_y 4.0
K_d_psi 2.0

NPO gains

K_2_x 2.0
K_2_y 2.0
K_2_psi 2.0
K_3_x 2.0
K_3_y 2.0
K_3_psi 2.0
K_4_x 2.0
K_4_y 2.0
K_4_psi 2.0

Filter gains

Filter 1 0.05
Filter 2 0.01
Filter 3 0.01

Figure 3.4: NI VeriStand User Interface example

that the Simulink diagram used in simulation need as little change as possible, all input blocks are implemented in the green input box and output blocks in the red output box. Signals are then sent by goto blocks to the main system. With this design only the green and red blocks are altered when changing from an offline simulation to a embedded design, in addition to removing the simulation plant and replacing it with out ports to the actuators.

3.2.2 NI VeriStand Ship Interface

User interface and physical I/O is handled by NI VeriStand. This is a National Instruments software used for real time testing, monitoring and HIL simulations.

A VeriStand project is setup and with embedded Simulink models, I/O and a GUI. It is then compiled for running on the cRIO VxWorks operating system. VeriStand then loads the compiled project onto the cRIO where the software runs locally.

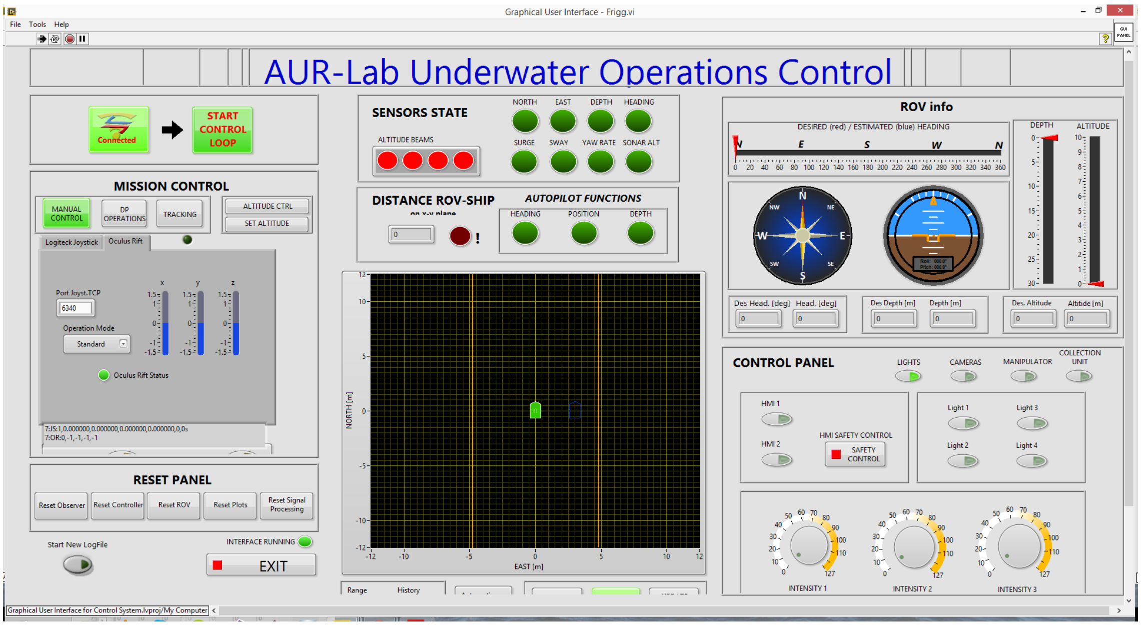
Input and output ports for the different software and hardware pieces are setup in the project file, through the FPGA unit on the cRIO physical I/O is available in the NI VeriStand menu, and the simply connected to the Simulink model.

User interface is made with simple drag and drop instruments in NI VeriStand. It is also possible to create custom widgets in LabVIEW and then import them into VeriStand. In Figure 3.4 shows an example of a GUI with a simple graph, some switches and displays. Here, the Simulink model can be tuned and monitored and the GUI can even be edited while the control system is running, which is useful.

Custom device

To bridge the Oculus Rift software and the control system, some additional software is needed. Custom devices take I/O from any source that can be integrated with LabVIEW.

The need for custom devices is one of the drawbacks for NI VeriStand compared with regular LabVIEW. The documentation for custom devices is limited and working with them is complicated, LabVIEW is often an easier option. However, once a custom device is implemented it is reliable.



Port Joyst.TCP: 6340

Operation Mode: Standard

```
7:JS:1,0.000000,0.000000,0.000000,0.000000,0,0s
7:OR:0,-1,-1,-1,-1
```

Figure 3.5: Frigg GUI for SF 30k

3.3 AUR-lab Control Software

The AUR lab control systems runs both of the NTNU ROVs, Minerva and SF 30k. The development of the system is described by Dukan (2014, Ch. 3) and Tolpinrud (2012).

The system is written in object-oriented LabVIEW and consists of two main modules: The control system *Njord* and the GUI *Frigg*, seen in Figure 3.5.

The user interaction is handled from the GUI while the control actions are handled by the control system. It is also possible to connect the system with a HIL simulator instead of a ROV, intended to simulate all aspects of ROV operation, like for example sensor characteristics.

3.3.1 Integration of Oculus Rift as Input Device

When integrating the Oculus Rift as an input device, the setup for joystick control has been followed. As the method in Section 5.1.1 states, joystick control and Oculus Rift control is very similar.

This means that Oculus Rift signals are first picked up in the GUI then sent to the control system. In Figure 3.6 the topology for the setup can be seen.

An extra LabVIEW program is used to interface the Oculus Rift signals. This was necessary since the synchronization of the signal flow has been crucial for the proper functioning of the system.It was also discovered that it was desirable to send signals over the UDP protocol, since the TCP protocol waits for confirmation that a sent package is received, and this caused delay and lag in our Oculus Rift software due to a bug.

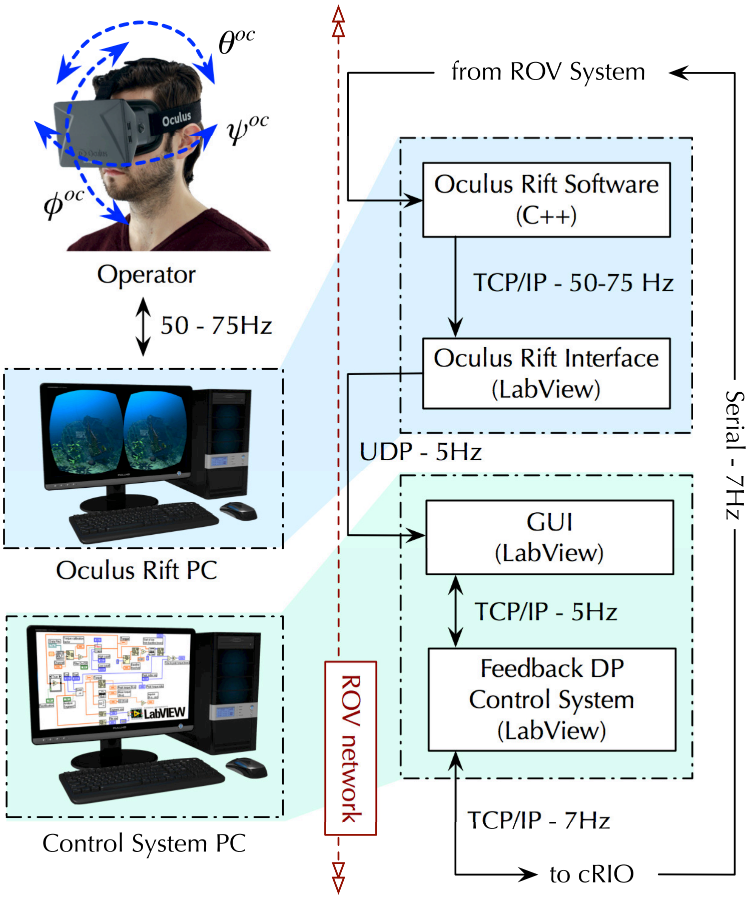


Figure 3.6: Software/hardware overview with protocols for SF 30k, courtesy of Mauro Candeloro

The video processing is completely separate from the LabVIEW control system. To connect to the video system described in Section 2.5.2, the Oculus Rift PC needs to be on the ROV subnet, and connect with the correct IP address.

Chapter 4

Mathematical Modeling of Ocean Vehicles

In order to predict the behavior of a dynamic system, without having the need of testing it full scale, a mathematical model of the system can be established. There are different types of models, based on their purpose and complexity. Sørensen (2012) categorizes the mathematical models in two groups:

- *Process Plant Model* (PPM) is a comprehensive description of the actual dynamic system. The PPM includes as much information as possible, with its purpose of simulation of the dynamic system in an accurate way.
- *Control Plant Model* (CPM) is a simplified version of the PPM, and hence a simplified model of the real dynamic system. Here, it is normal to include the main forces and effects of interest, and the rest is neglected. It is important to mention that, in order to simplify a model, care must be taken, in order not neglect relevant terms for the vehicles dynamics.

In this chapter, a CPM for CSE1 is developed. It deals with both the kinematic and kinetic part of the dynamic behavior. The kinematics treat the geometrical aspects of motion, while the kinetics models forces causing the motion. The underwater vehicle is free to rotate and translate in all directions, meaning that it has six degrees of freedom. The Society of Naval Architects and Marine Engineers (SNAME, 1950) notation will be used together with Fossen (2011) "robot like" way of representing the equation of motions in a vectorial setting.

DOF	Definition	Linear and Angular Velocities	Position Attitude	Forces and moments
1	*surge*	u [m/s]	x [m]	X [N]
2	*sway*	v [m/s]	y [m]	Y [N]
3	*heave*	w [m/s]	z [m]	Z [N]
4	*roll*	p [rad/s]	ϕ [rad]	K [Nm]
5	*pitch*	q [rad/s]	θ [rad]	M [Nm]
6	*yaw*	r [rad/s]	ψ [rad]	N [Nm]

Table 4.1: Motion variables notation (SNAME, 1950)

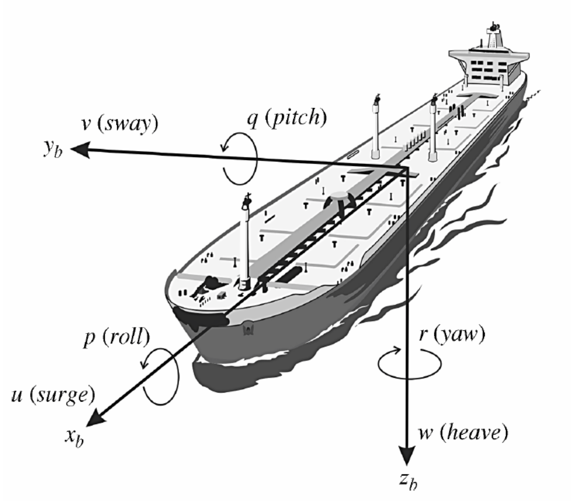


Figure 4.1: DOF illustration (Fossen, 2011)

4.1 Kinematics

Fossen (2011, Ch. 2) explains kinematics for ocean vehicles. Kinematics deal with the geometrical relationship of the variables in the various frames in which those can be defined. A vessel can move in 6 DOF which are denoted in Table 4.1 and illustrated in Figure 4.1.

4.1.1 Reference Frames

In order to establish proper results from analyzing the motion of a vehicle, it is convenient to define two or more reference frames. In most cases, one of these reference frames are fixed to the vehicle. Then, to represent the motion of the vehicle in space, a reference frame that is fixed to a specific point on earth is needed. Fossen (2011) presents several reference frames:

1. **MC lab**:It is defined with axis as seen in Figure 2.9. For local navigation it is assumed that the MC lab-frame (x, y, z) is constant and inertial, such that Newton's second law holds.

2. **BODY**: this reference frame is fixed to the vehicle. The first axis (x_b) is directed from aft to fore on the vehicle, the second axis (y_b) is pointing towards starboard, while the third axis (z_b) is pointing downwards.

3. **NED** (North-East-Down): this reference frame is earth fixed. It is defined such that the x,y and z axis points straight north, east and down respectively. For local navigation it is assumed that the NED-frame (N, E, D) is constant and inertial, such that Newton's second law holds.

CSE1 will move in the MC lab. Hence it it sufficient to use the MC lab frame as the inertial frame where the vehicle is navigating within. The ROV SF 30k will use the NED frame as the inertial frame.

4.1.2 Vectorized Notation

In the notation the subscripts $\{n\}$, $\{b\}$ and $\{mc\}$ denotes respectively NED frame, body-frame and MC-lab frame. Bold symbols denotes vectors.

$$\text{NED position - } \boldsymbol{p}^n_{b/n} = \begin{bmatrix} N \\ E \\ D \end{bmatrix} \tag{4.1}$$

$$\text{MC-lab position - } \boldsymbol{p}^{mc}_{b/mc} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.2}$$

$$\text{Attitude in Euler angles - } \boldsymbol{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{4.3}$$

$$\text{Body fixed linear velocity - } \boldsymbol{v}^b_{b/n} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.4}$$

$$\text{Body fixed angular velocities - } \boldsymbol{\omega}^b_b = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{4.5}$$

$$\text{Body fixed force - } \boldsymbol{f}^b_b = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.6}$$

$$\text{Body fixed moments - } \boldsymbol{m}^b_b = \begin{bmatrix} K \\ M \\ N \end{bmatrix} \tag{4.7}$$

Hence, motion in the general reference frame is denoted as

$$\text{positions } \boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}^n_{b/n} \\ \boldsymbol{\Theta}_{nb} \end{bmatrix}, \tag{4.8}$$

$$\text{velocities } \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}^n_{b/n} \\ \boldsymbol{\omega}^b_b \end{bmatrix} \tag{4.9}$$

$$\text{and forces } \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{f}^b_b \\ \boldsymbol{m}^b_b \end{bmatrix}. \tag{4.10}$$

If the MC lab frame is used $\boldsymbol{p}^{mc}_{b/mc}$ will obviously replace $\boldsymbol{p}^n_{b/n}$ in $\boldsymbol{\eta}$.

4.1.3 Transformation Between Reference Frames

The kinematic relationship is described by Fossen (2011, Ch. 2) as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_\Theta(\boldsymbol{\eta})\boldsymbol{\nu} \tag{4.11}$$

where the transformation matrix $\boldsymbol{J}_\Theta(\boldsymbol{\eta})$ is given by

$$\boldsymbol{J}_\Theta(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{\Theta}) & \boldsymbol{0}_{3x3} \\ \boldsymbol{0}_{3x3} & \boldsymbol{T}(\boldsymbol{\Theta}) \end{bmatrix} \tag{4.12}$$

with the linear transformation matrix

$$\boldsymbol{R}(\boldsymbol{\Theta}) = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{x,\phi} = \begin{bmatrix} \mathrm{c}\psi\mathrm{c}\theta & -\mathrm{s}\psi\mathrm{c}\phi + \mathrm{c}\psi\mathrm{s}\theta\mathrm{s}\phi & \mathrm{s}\psi\mathrm{s}\phi + \mathrm{c}\psi\mathrm{c}\phi\mathrm{s}\theta \\ \mathrm{s}\psi\mathrm{c}\theta & \mathrm{c}\psi\mathrm{c}\phi + \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi & -\mathrm{c}\psi\mathrm{s}\phi + \mathrm{s}\theta\mathrm{s}\psi\mathrm{c}\phi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\phi & \mathrm{c}\theta\mathrm{c}\phi \end{bmatrix} \tag{4.13}$$

and the angular transformation matrix

$$\boldsymbol{T}(\boldsymbol{\Theta}) = \begin{bmatrix} 1 & \mathrm{s}\phi\mathrm{t}\theta & \mathrm{c}\phi\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \frac{\mathrm{s}\phi}{c\theta} & \frac{\mathrm{c}\phi}{c\theta} \end{bmatrix}. \tag{4.14}$$

Here, the notation is abridged as $\mathrm{c}(\cdot) = \cos(\cdot)$, $\mathrm{s}(\cdot) = \sin(\cdot)$ and $\mathrm{t}(\cdot) = \tan(\cdot)$. Since the angular transformation matrix has $\cos(\theta)$ expressions below the fraction bar the matrix is singular for $\theta = \pm\ \frac{\pi}{2}$ and therefore not defined in this point. Since none of the vehicles in this thesis have anything near this much roll, this is not a problem. However, a fix for this problem would be to change the angle representation from Euler angles to unit quaternions (Fossen, 2011).

4.2 Kinetics

The kinetics describe the forces acting on the rigid body. The forces considered are mass and added mass forces denoted in $\boldsymbol{M}$, Coriolis and centripetal forces due to rotation of the body frame about the inertial frame, denoted $\boldsymbol{C}(\boldsymbol{\nu})$, hydrodynamic damping forces are expressed in $\boldsymbol{D}(\boldsymbol{\nu})$, and restoring forces due to hydrostatics are given in the vector $\boldsymbol{g}(\boldsymbol{\eta})$. Forces acting on the vessel are given on the right side of the equation, with thruster forces given by $\boldsymbol{\tau}$ and environmental forces from wind, waves and current are given by $\boldsymbol{\tau}_{env}$

This is put together to Fossen's robot like vectorial model for marine crafts(Fossen, 2011).

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{4.15}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{env} \tag{4.16}$$

General kinetic modeling will not be presented in further detail here. However, the models for CSE1 and SF 30k are presented in the following sections.

4.3 Cybership Enterprise 1

The mathematical model for CSE1 was first established by Skåtun (2011). For the course TMR4243 - Marine Control Systems 2 the model was updated with new towing tank tests to identify damping parameters (NTNU, 2015).

4.3.1 Kinematics

Since the vessel cannot be actuated in roll, pitch and heave and will only move in the MC lab. The vessel model is reduced from 6 DOF to 3 DOF. With $\boldsymbol{\eta}$ denoting positions, $\boldsymbol{\nu}$ denoting velocities and $\boldsymbol{\tau}$ denoting thruster forces.

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \tag{4.17}$$

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & r \end{bmatrix}^T \tag{4.18}$$

$$\boldsymbol{\tau} = \begin{bmatrix} X & Y & N \end{bmatrix}^T \tag{4.19}$$

According to Fossen (2011) a simplified kinematic relationship for 3 DOF can be described as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{4.20}$$

Where rotation matrix $\boldsymbol{R}(\psi)$ is given by

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \mathrm{c}(\psi) & -\mathrm{s}(\psi) & 0 \\ \mathrm{s}(\psi) & \mathrm{c}(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.21}$$

4.3.2 Kinetics

According to Faltinsen (1990, Ch. 3) there is no restoring forces in surge, sway and yaw. The fact that CSE1 will only be operated in the MC lab with little (or almost none) environmental disturbances, this simplifies (4.11) and (4.16) to

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{4.22}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} \tag{4.23}$$

On component form the matrices are given as follows:
Mass and added mass:

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{4.24}$$

Coriolis and centripetal forces:

$$\boldsymbol{C(\nu)} = \begin{bmatrix} 0 & -mr & Y_{\dot{v}}v + (Y_{\dot{r}} - mx_g)r \\ mr & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - (Y_{\dot{r}} - mx_g)r & X_{\dot{u}}u & 0 \end{bmatrix} \tag{4.25}$$

Hydrodynamic damping:

$$\boldsymbol{D(\nu)} = \begin{bmatrix} d_{11}(u) & 0 & 0 \\ 0 & d_{22}(v,r) & d_{23}(v,r) \\ 0 & d_{32}(v,r) & d_{33}(v,r) \end{bmatrix} \tag{4.26}$$

$$d_{11}(u) = -X_u - X_{|u|u}\,|u| - X_{uuu}u^2 \tag{4.27}$$

$$d_{22}(v,r) = -Y_v - Y_{|v|v}\,|v| - Y_{vvv}v^2 - Y_{|r|v}\,|r| \tag{4.28}$$

$$d_{23}(v,r) = -Y_r - Y_{|v|r}\,|v| - Y_{|r|r}\,|r| - Y_{rrr}r^2 \tag{4.29}$$

$$d_{32}(v,r) = -N_v - N_{|v|v}\,|v| - N_{vvv}v^2 - N_{|r|v}\,|r| \tag{4.30}$$

$$d_{33}(v,r) = -N_r - N_{|v|r}\,|v| - N_{|r|r}\,|r| - N_{rrr}r^2 \tag{4.31}$$

Damping and added mass parameters are listed in Appendix A.1.1.

Assumptions and simplifications

This model ignores all environmental forces like wind, waves and current. In the MC lab this is a good assumption as it is an indoor pool where any environmental loads like wind, waves or current are user generated.

The model does not include a bias term, this is usually included to account for environmental disturbances and modeling errors, and for more advanced designs this should be included.

As a result of reducing from 6 DOF to 3 DOF the hydrostatic restoring forces are ignored (as surge, sway and yaw does not have restoring forces).

4.4 SUB-fighter 30k

The model for SF 30k presented in this thesis is collected from Berg (2012).

4.4.1 Kinematics

The ROV moves in the NED reference frame and all 6 DOF are modeled, even though it is only possible to actuate surge, sway, heave and yaw.

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T \tag{4.32}$$

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T \tag{4.33}$$

$$\boldsymbol{\tau} = \begin{bmatrix} X & Y & Z & N \end{bmatrix}^T \tag{4.34}$$

4.4.2 Kinetics

For the kinetics, the following assumptions are made by Dukan (2014) for the SF 30k model:

- Coriolis and centripetal forces are neglected due to the assumption of low speed maneuvering. Velocities are less than 1 [m/s].

- Ocean currents are assumed constant and slowly varying. Thus, they are included in the bias estimate, modeled as a 1^{st} order Markov process, as in (4.37).

- Movement in roll and pitch is less than 10 degrees, the center of buoyancy (CB) is located straight above the center of gravity (CG) and the ROV is neutrally buoyant. When this is the case the restoring forces $\boldsymbol{g}(\boldsymbol{\eta})$ from (4.16) can be linearized to $\boldsymbol{G}$, see (4.40)

These assumptions leads to the following CPM:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}_{\Theta}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{4.35}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{D}\boldsymbol{\nu} + \boldsymbol{G}\boldsymbol{\eta} = \boldsymbol{\tau} + \boldsymbol{J}^T(\boldsymbol{\eta})\boldsymbol{b} \tag{4.36}$$

$$\dot{\boldsymbol{b}} = -\boldsymbol{T}_b^{-1}\boldsymbol{b} + \boldsymbol{w}_b \tag{4.37}$$

where the mass matrix consisting of mass, inertia and added mass terms are given as:

$$\boldsymbol{M} = \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{v}} & -X_{\dot{w}} & -X_{\dot{p}} & -X_{\dot{q}} & -X_{\dot{r}} \\ -Y_{\dot{u}} & m - Y_{\dot{v}} & -Y_{\dot{w}} & -Y_{\dot{p}} & -Y_{\dot{q}} & -Y_{\dot{r}} \\ -Z_{\dot{u}} & -Z_{\dot{v}} & m - Z_{\dot{w}} & -Z_{\dot{p}} & -Z_{\dot{q}} & -Z_{\dot{r}} \\ -K_{\dot{u}} & -K_{\dot{v}} & -K_{\dot{w}} & I_x - K_{\dot{p}} & -I_{xy} - K_{\dot{q}} & -I_{xz} - K_{\dot{r}} \\ -M_{\dot{u}} & -M_{\dot{v}} & -M_{\dot{w}} & -I_{yx} - M_{\dot{p}} & I_y - M_{\dot{q}} & -I_{yz} - M_{\dot{r}} \\ -N_{\dot{u}} & -N_{\dot{v}} & -N_{\dot{w}} & -I_{zx} - N_{\dot{p}} & -I_{zy} - N_{\dot{q}} & I_z - N_{\dot{r}} \end{bmatrix} \tag{4.38}$$

here m is the mass of the ROV, while I_x, I_y, I_z are the moments of inertia around the body axis. The damping matrix is made up of linear terms only, given as:

$$\boldsymbol{D} = -\mathrm{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\}. \tag{4.39}$$

The linearized restoring forces are expressed on G as:

$$\boldsymbol{G} = \mathrm{diag}\{0, 0, 0, -z_b B, -z_b B, 0\}, \tag{4.40}$$

where B denotes buoyancy and z_b denotes the distance from CG to CB.

$\boldsymbol{T}_b$ is a made up of positive constants, while $\boldsymbol{w}_b \in \mathbb{R}^{6x1}$ is a zero mean Gaussian white noise process:

$$\boldsymbol{T}_b = \mathrm{diag}\{b_1, b_2, b_3, b_4, b_5, b_6\} \tag{4.41}$$

The bias estimate covers slowly varying disturbances such as ocean current as well as modeling errors.

The values for the parameters used are given in Appendix A.2

Chapter 5

Control Design

In this chapter the guidance, navigation and control (GNC) system for CSE1 and SF 30k is presented. Previous work on CSE1 control system is presented by Skåtun (2011), Tran (2014) and Orsten (2014). While development of the control system for SF 30k is discussed by Dukan (2014) and Berg (2012).

In Figure 5.1 the general setup of a GNC system with user input is presented. The plant is in this case CSE1 or SF 30k.

Here the guidance system has the task to transform the user input to a feasible task for the control system. In this instance, a signal processing algorithm and a reference filter is utilized.

The motion control module compares the desired states with the actual states and calculated necessary force. For example the desired position is compared with the estimated position provided by the navigation module.

The thrust allocation calculated force needs to be transformed into vessel specific commands that fits the specific thrusters of the vessel, this is handled by the thruster allocation block.

The main contribution of this chapter is the gesture based guidance system developed to transform Oculus Rift signals into usable control inputs.

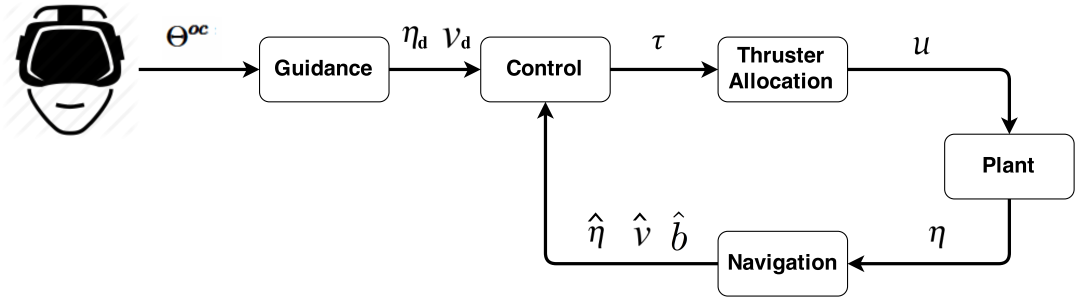


Figure 5.1: Guidance, navigation and control blocks

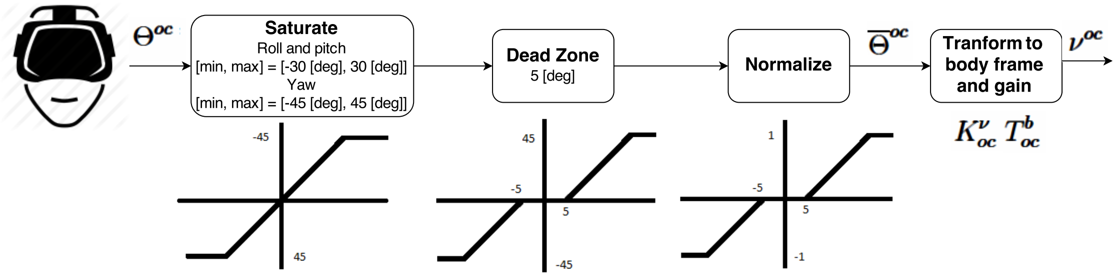

Θ^{oc}
Saturate
Roll and pitch
[min, max] = [-30 [deg], 30 [deg]]
Yaw
[min, max] = [-45 [deg], 45 [deg]]
Dead Zone
5 [deg]
Normalize
$\overline{\Theta}^{oc}$
Tranform to body frame and gain
ν^{oc}
K^{ν}_{oc} T^{b}_{oc}
-45
45
45
-5
5
-45
1
-5
5
-1

Figure 5.2: Oculus Rift frame to body frame transformation

5.1 Guidance

The purpose of guidance is to produce a feasible desired state for the control system. In this case, a desired speed reference $\boldsymbol{\nu}^{oc}$ is delivered to the reference model which then produces the desired states $\boldsymbol{\eta}_d$ and $\boldsymbol{\nu}_d$.

5.1.1 Gesture Based Guidance

This module, a gesture inspired by the control design based on feedback joystick control as shown by Dukan (2014, Ch. 5.3). The system designed here gives guidance in the plane.

First, the head angles are described

$$\boldsymbol{\Theta}^{oc} = \begin{bmatrix} \phi^{oc} & \theta^{oc} & \psi^{oc} \end{bmatrix}^T . \tag{5.1}$$

For practical reasons the angles are normalized, as that makes them easier to work with.

$$\overline{\boldsymbol{\Theta}}^{oc} = \begin{bmatrix} \overline{\phi}^{oc} & \overline{\theta}^{oc} & \overline{\psi}^{oc} \end{bmatrix}^T \tag{5.2}$$

Where $\overline{\phi}^{oc}$, $\overline{\theta}^{oc}$, $\overline{\psi}^{oc} \in \mathbb{R} \subseteq [-1, 1]$. The Oculus Rift signals $\boldsymbol{\Theta}^{oc}$ are also saturated at respectively 30 degrees for roll and pitch and 45 degrees for yaw. This is done since larger angles would lead to unnatural head movements, the saturation limits have been found experimentally during simulations. A dead-zone of five degrees is also implemented to help the operator keeping the vehicle in DP, avoiding unwanted movements deriving from small oscillations of the head, this whole process is illustrated in Figure 5.2.

The commands that move the vehicle is as follows. Tilting the head forwards will push the vehicle forwards along the surge direction, tilting the head sideways will induce sway motion and turning the head clockwise/counterclockwise will induce yaw. These commands are mapped to Oculus Rift velocity commands denoted $\boldsymbol{\nu}^{oc}$ that will be fed to the reference model. This mapping is given by;

$$\boldsymbol{\nu}^{oc} = \boldsymbol{K}_{oc}^{\nu}\boldsymbol{T}_{oc}^{b}\overline{\boldsymbol{\Theta}}^{oc} \tag{5.3}$$

Here $\boldsymbol{K}_{oc}^{\nu}$ is a tuning matrix and $\boldsymbol{T}_{oc}^{b}$ is the transformation matrix from the Oculus Rift reference frame to the ship fixed body-frame.

$$\boldsymbol{K}_{oc}^{\nu} = \begin{bmatrix} K_u & 0 & 0 \\ 0 & K_v & 0 \\ 0 & 0 & K_r \end{bmatrix}, \boldsymbol{T}_{oc}^{b} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{5.4}$$

Note that the transformation matrix is defined with negative values as the Oculus Rift reference frame is defined opposite of the body-frame.

5.1.2 Cybership Enterprise 1 Reference Model

A smooth reference trajectory for CSE1 is generated by a the reference model as described by Fossen (2011, Ch. 10). This model assures that the desired trajectory fed to the controller is smooth and bounded. The model is inspired by a mass-spring-damper system with damping ζ and natural frequency ω_n which can be written in a vectorial setting as (5.6).

$$\dot{\boldsymbol{\eta}}_{\boldsymbol{d}} = \boldsymbol{R}(\psi_d)\boldsymbol{\nu}_{\boldsymbol{d}} \tag{5.5}$$

$$\boldsymbol{\nu}_{\boldsymbol{d}}^{(3)} + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}\ddot{\boldsymbol{\nu}}_{\boldsymbol{d}} + (2\boldsymbol{\Delta} + \boldsymbol{I})\boldsymbol{\Omega}^{2}\dot{\boldsymbol{\nu}}_{\boldsymbol{d}} + \boldsymbol{\Omega}^{3}\boldsymbol{\nu}_{\boldsymbol{d}} = \boldsymbol{\Omega}^{3}\boldsymbol{\nu}^{oc} \tag{5.6}$$

where vectorial damping and natural frequency is written as:

$$\boldsymbol{\Delta} = \text{diag}\{\zeta_1, \zeta_2, \zeta_3\} > 0,\ \boldsymbol{\Omega} = \text{diag}\{\omega_{n1}, \omega_{n2}, \omega_{n3}\} > 0 \tag{5.7}$$

For the special case of direct heading control, meaning the CSE1 yaw angle should copy the yaw angle of the Oculus Rift operators head exactly. The reference model is set up as:

$$\psi_d^{(3)} + (2\zeta_3 + 1)\omega_3\ddot{\psi}_d + (2\zeta_3 + 1)\omega_3^2\dot{\psi}_d + \omega_3^3\psi_d = \omega_3^3(-\psi^{oc}) \tag{5.8}$$

For tuning of the reference model damping and natural frequency parameters are set: $\zeta_i = 1$ (critical damping) is set in all degree of freedoms, ω_{n3}, (natural frequency in yaw) is set to a high value to ensure that ψ_d aggressively follows the reference, this is important since the head movement has much faster dynamics than the vehicle movements.

5.1.3 SUB-fighter 30k Reference Model

For SF 30k, a second order low pass filter is implemented as a reference model, as where CSE1 has a third order filter.

$$\dot{\boldsymbol{\eta}}_d = \boldsymbol{J}(\psi_d)\boldsymbol{\nu}_d \tag{5.9}$$

$$\ddot{\boldsymbol{\nu}}_d + 2\boldsymbol{\Delta\Omega}\dot{\boldsymbol{\nu}}_d + \boldsymbol{\Omega}\boldsymbol{\nu}_d = \boldsymbol{\Omega}^2\boldsymbol{\nu}^{oc} \tag{5.10}$$

where

$$\boldsymbol{\Delta} = \text{diag}\{\zeta_1, \zeta_2, \zeta_3, \zeta_4, \zeta_5, \zeta_6\} > 0 \tag{5.11}$$

$$\boldsymbol{\Omega} = \text{diag}\{\omega_{n1}, \omega_{n2}, \omega_{n3}, \omega_{n4}, \omega_{n5}, \omega_{n6}\} > 0 \tag{5.12}$$

Tuning parameters for CSE1 and SF 30k reference models are available in Appendix A.1.2 and A.2.2

5.2 Navigation

The purpose of a navigation system is to feed the control system with accurate state information. A typical marine navigation system consists of sensors providing measurements, typically compass, positioning system and inertial measurements. And an observer that filters the measurements and estimates unmeasured states. The most famous observer used for marine vehicles is the Kalman (1960) filter that NTNU professor Jens Glad Balchen (Breivik and Sand, 2009) used to revolutionize DP for marine application, recently voted as greatest Norwegian engineering achievement since world war II (Qvale et al., 2015).

Candeloro et al. (2012) presents the sector-based Kalman filter observer utilized by SF 30k in our mission, this will not be presented in detail here.

5.2.1 Cybership Enterprise 1

For the CSE1, position ($\boldsymbol{\eta}$) measurements are available, while velocities ($\boldsymbol{\nu}$), are not measured and have to be estimated. To accomplish this a discrete filter is implemented.

The filter works by low-pass filtering the measurements of $\boldsymbol{\eta}$ before taking the discrete derivative . The resulting $\dot{\boldsymbol{\eta}}$ is then filtered again before the state is rotated into the body frame:

$$\bar{\boldsymbol{\eta}}_k = \alpha_1 \boldsymbol{\eta}_k + (1-\alpha_1)\bar{\boldsymbol{\eta}}_{k-1} \tag{5.13}$$

$$\dot{\boldsymbol{\eta}}_k = \frac{\bar{\boldsymbol{\eta}}_k - \bar{\boldsymbol{\eta}}_{k-1}}{h} \tag{5.14}$$

$$\dot{\bar{\boldsymbol{\eta}}}_k = \alpha_2 \dot{\boldsymbol{\eta}}_k + (1-\alpha_2)\dot{\bar{\boldsymbol{\eta}}}_{k-1} \tag{5.15}$$

$$\hat{\boldsymbol{\nu}} = \boldsymbol{R}(\psi)^T \dot{\bar{\boldsymbol{\eta}}}_k \tag{5.16}$$

Here α_1 and α_2 are the low pass filter time constants and h is the system step size, that is $h = 0.01$. The time constants assume values between 0.01 and 0.1. Depending on how much filtering is desired: smaller α_i will give a smoother signal at the cost of time delay, while higher α_i will give a more aggressive tuning at the cost of a more noisy signal.

It is worth mentioning that position measurements in the MC lab are very accurate compared with underwater positioning measurements. So the CSE1 control system is less dependent of an advanced signal processing algorithm, than what would otherwise be necessary.

5.3 Motion Control

Control algorithms compare measured and estimated states from the navigation system with desired states from guidance, and then produces the necessary force to make them converge.

5.3.1 Cybership Enterprise 1

The control design for CSE1 with Oculus Rift reference signal is chosen as a proportional-derivative (PD) station keeping controller, this is motivated by the fact that, CSE1 will only need to stay in a fixed position and in the MC lab there are no external disturbances such as wind or current. This makes integral effect negligible in the for our experiments . The control algorithm Fossen (2011, Ch. 12)

$$\boldsymbol{\tau} = -\boldsymbol{K}_P \boldsymbol{R}^T(\psi)\tilde{\boldsymbol{\eta}} - \boldsymbol{K}_D\tilde{\boldsymbol{\nu}}, \tag{5.17}$$

where the error states $\tilde{\boldsymbol{\eta}}$ and $\tilde{\boldsymbol{\nu}}$ are

$$\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta} - \boldsymbol{\eta}_d \text{ and} \tag{5.18}$$

$$\tilde{\boldsymbol{\nu}} = \hat{\boldsymbol{\nu}} - \boldsymbol{\nu}_d, \tag{5.19}$$

$$\tag{5.20}$$

the gain matrices are

$$\boldsymbol{K}_P = \text{diag}\{K_{Px}, K_{Py}, K_{P\psi}\} > 0 \tag{5.21}$$

$$\boldsymbol{K}_D = \text{diag}\{K_{Du}, K_{Dv}, K_{Dr}\} > 0 \tag{5.22}$$

The control law has different gains for different scenarios, but the overall control layout is the same for all CSE1 control scenarios in this thesis.

5.3.2 SUB-fighter 30k

Details for the SF 30k control algorithm are collected from Dukan (2014, Appendix B).

Feed-forward force control

The simplest form of control is the feed-forward force control. This turns operators head movements directly into thrust commands and involves no feedback terms, using the control law

$$\boldsymbol{\tau} = \boldsymbol{K}_{oc}^{\tau}\boldsymbol{T}_{oc}^{b}\bar{\boldsymbol{\Theta}}^{oc}, \tag{5.23}$$

the gains are set as the maximum available thrust:

$$\boldsymbol{K_{oc}^{\tau}} = diag\{X_{max}, Y_{max}, Z_{max}, 0, 0, N_{max}\}. \tag{5.24}$$

Speed reference, position and velocity control

For position control of the ROV, a nonlinear PID control law is implemented. This has two working modes: position control and velocity control.

For position control the following algorithm is used:

$$\boldsymbol{\tau} = -\boldsymbol{J}^T(\boldsymbol{\eta})(\boldsymbol{K}_p\tilde{\boldsymbol{\eta}} + \boldsymbol{K}_i \int_0^t \tilde{\boldsymbol{\eta}} d\boldsymbol{\tau}) + \boldsymbol{K}_d\boldsymbol{\nu} \tag{5.25}$$

When velocity control is activated, the force elements corresponding to surge and sway are alternated to:

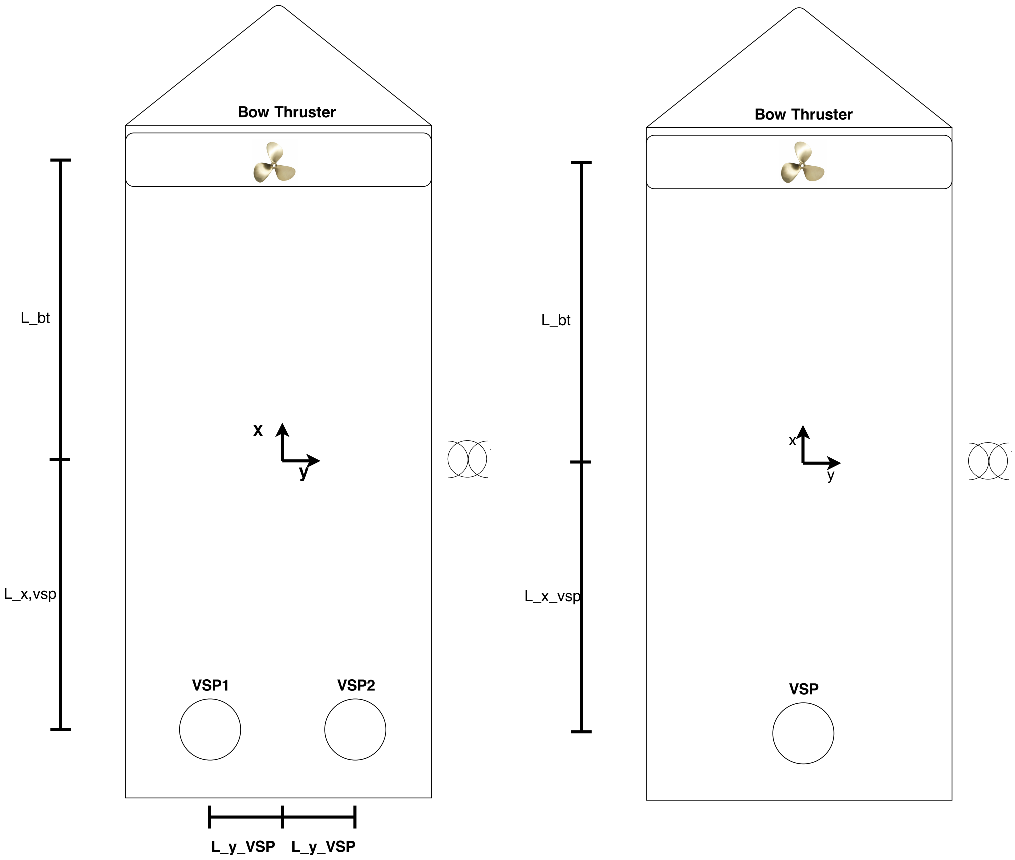
$$X = -K_{p,u}(u - u_d) - K_{i,u} \int_0^t (u - u_d) d\tau \tag{5.26}$$

$$Y = -K_{p,v}(v - v_d) - K_{i,v} \int_0^t (v - v_d) d\tau \tag{5.27}$$

For the integral effect of both controllers, an anti wind up algorithm is implemented:

$$\tau_i = (K_i - K_{\text{anti,i}}(\text{sat}(\tau_i) - \tau_i)) \int_0^T \tilde{\eta}(t) d\tau \tag{5.28}$$

The anti wind up gain $K_{\text{anti,i}}$ is set to 80% of the corresponding DOF available thrust.



(a) Thrust allocation (b) Thrust allocation simplified

Figure 5.3: Thrust allocation

5.4 Thrust Allocation

The control force $\boldsymbol{\tau}$ is calculated in the body frame. This force command has to be redistributed to commands specified for the vessel actuators. Usually denoted with the vector $\boldsymbol{u}$ such that the actuators produce the desired force. Thrust allocation for marine crafts is described by Fossen (2011, Ch. 12.3) and Sørensen (2012, Ch. 8.3).

CSE1 thrust allocation is elaborated here, as this is new a design. Thrust allocation for SF 30k is presented by (Berg, 2012, Ch. 2.4 and 3.2).

5.4.1 Cybership Enterprise 1

Thrust allocation is based on the configuration shown in Figure 5.3b (NTNU, 2015). The thrust is thus

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{K}\boldsymbol{u} \tag{5.29}$$

Where the configuration matrix is

$$\boldsymbol{T}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos(\alpha_{\text{VSP1}}) & \cos(\alpha_{\text{VSP2}}) & 0 \\ \sin(\alpha_{\text{VSP1}}) & \sin(\alpha_{\text{VSP2}}) & 1 \\ l_{x,\text{VSP1}}\cos(\alpha_{\text{VSP1}}) - l_{y,\text{VSP1}}\sin(\alpha_{\text{VSP1}}) & l_{x,\text{VSP2}}\cos(\alpha_{\text{VSP2}}) - l_{y,\text{VSP2}}\sin(\alpha_{\text{VSP2}}) & l_{\text{BT}} \end{bmatrix}, \tag{5.30}$$

where the thruster force angles are

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{\text{VSP1}} \\ \alpha_{\text{VSP2}} \end{bmatrix}, \tag{5.31}$$

$$\tag{5.32}$$

the force coefficient matrix is

$$\boldsymbol{K} = \begin{bmatrix} K_{\text{VSP1}} & 0 & 0 \\ 0 & K_{\text{VSP2}} & 0 \\ 0 & 0 & K_{\text{BT}} \end{bmatrix} \tag{5.33}$$

$$\tag{5.34}$$

and

$$\boldsymbol{u} = \begin{bmatrix} u_{\text{VSP1}} \\ u_{\text{VSP2}} \\ u_{\text{BT}} \end{bmatrix} \text{ are the control forces.} \tag{5.35}$$

Since solving the thrust equation for u and α would be not trivial, a virtual azimuth VSP thruster representing the joint forces from VSP1 and VSP2 is considered instead, as illustrated in Figure 5.3b. It is further assumed that $\alpha_{\text{VSP1}} = \alpha_{\text{VSP2}}$, $K_{\text{VSP1}} = K_{\text{VSP2}}$ and $u_{\text{VSP1}} = u_{\text{VSP2}}$. Considering an extended control force vector

$$\boldsymbol{u_e} = \begin{bmatrix} u_{\text{VSP},x} \\ u_{\text{VSP},y} \\ u_{\text{BT}} \end{bmatrix}, \tag{5.36}$$

where the VSP control forces are decomposed, yields

$$\underbrace{\begin{bmatrix} X \\ Y \\ N \end{bmatrix}}_{\tau_e} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & l_{x,\text{VSP}} & l_{\text{BT}} \end{bmatrix}}_{T_e} \underbrace{\begin{bmatrix} K_{\max,\text{VSP}} & 0 & 0 \\ 0 & K_{\max,\text{VSP}} & 0 \\ 0 & 0 & K_{\max,\text{BT}} \end{bmatrix}}_{K_e} u_e. \tag{5.37}$$

This is solved for u_e by simple inversion. Finally, the actual control forces are

$$\begin{aligned} u_{\mathrm{VSP1}} = u_{\mathrm{VSP2}} &= \sqrt{\left(u_{\mathrm{VSP},x}\right)^2 + \left(u_{\mathrm{VSP},y}\right)^2}, \text{ and} && (5.38)\\ \alpha_{\mathrm{VSP1}} = \alpha_{\mathrm{VSP2}} &= \operatorname{arctan2}\left(u_{\mathrm{VSP},y}, u_{\mathrm{VSP},x}\right). && (5.39) \end{aligned}$$

Chapter 6

Results and discussion

This chapter present the results obtained from simulation, HIL testing, model scale tests (collected at the MC lab) and full scale tests (collected with ROV SF 30k). Since this is a new untested system with a human in the control loop, an operator assessment of the user experience is provided as well as analysis of the control performance.

For simulations, HIL and model scale tests with CSE1, the following results are presented:

- direct heading control;
- gesture based control:
    - Speed reference position control;

For Full scale tests with SF 30k the following results are presented:

- Gesture based control:
    - Feed forward force control;
    - Speed reference, velocity control;
    - Speed reference, position control.

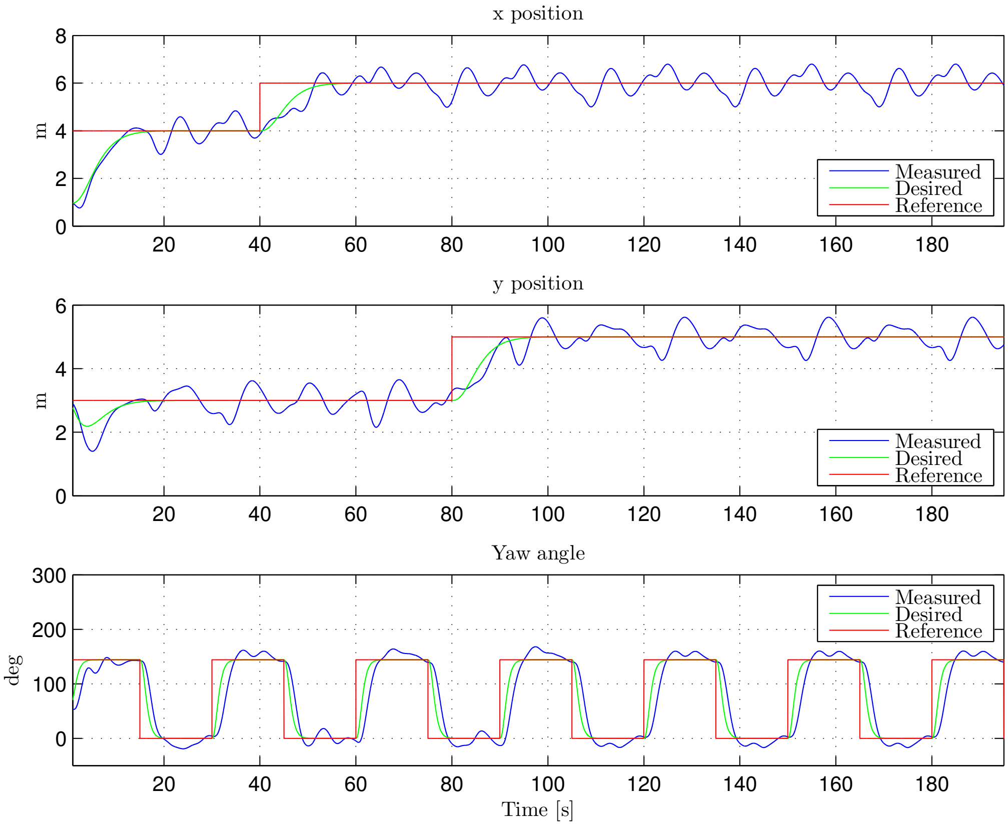


Figure 6.1: Simulation results: x and y position and yaw angle

6.1 Cybership Enterprise 1 Simulation

The offline simulations are mostly used in order to verify the CSE1 mathematical model and control system, with the main objective of verifying software before starting with real-time control. Thus, not a lot of effort has gone into fine tuning of the controller.

In Figure 6.1 the ship performing station keeping with regards to x and y, while the Oculus Rift reference, that controls heading is simulated as an impulse signal, impersonating sharp head movements. The green line represents the desired states (output from the reference model).

The ship is able to keep position and follow the reference, though it is oscillating in x and y, probably due to the rapid change of heading.

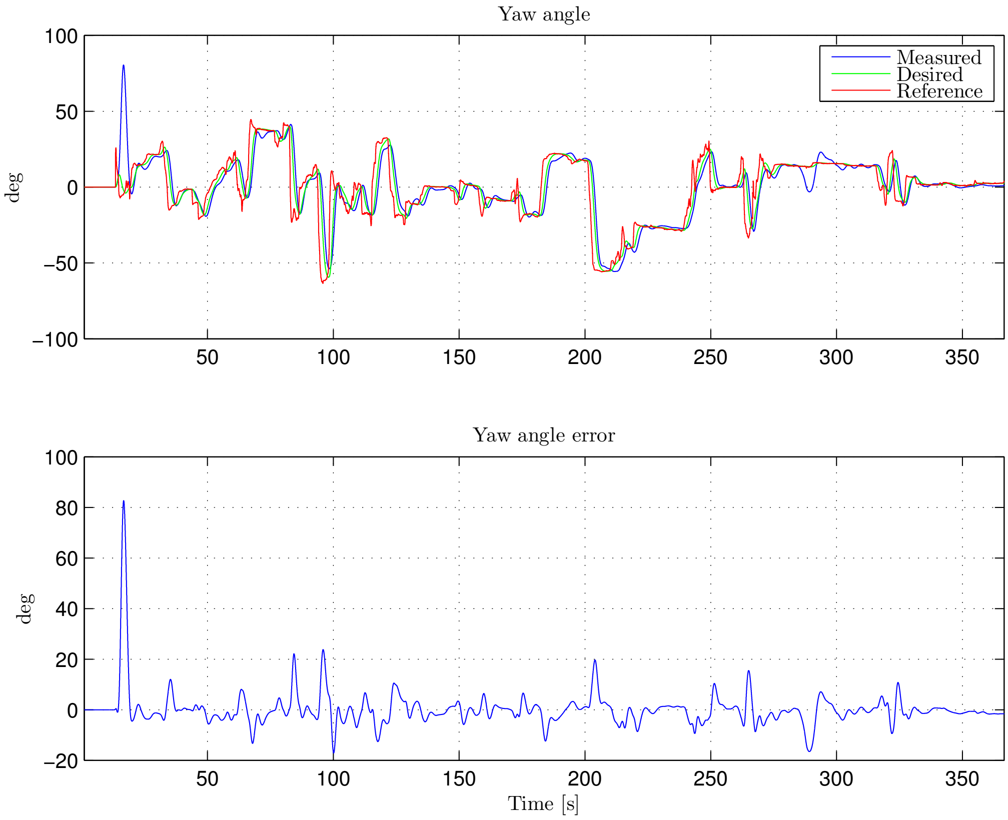


Figure 6.2: HIL: Direct heading control, yaw angle and error

6.2 Cybership Enterprise 1 HIL

For the HIL tests, very similar performance as in the simulation study were expected. The main difference from the simulation study was that Oculus Rift is giving the input; moreover, the test is in real-time. For this section both the direct heading control and gesture control of CSE1 is included.

The HIL test has been valuable for debugging of the system. The problems regarding different sampling periods were discovered at an early stage, and was corrected before the model scale tests.

6.2.1 Direct Heading control

Results for heading control are shown in Figure 6.2. The reference is now given directly by the head angle ψ^{oc}.

As seen in the plot, desired yaw angle follows the reference with a slight delay due to the reference filter. It is possible to see that the vessel then tracks the desired yaw angle with good performance.

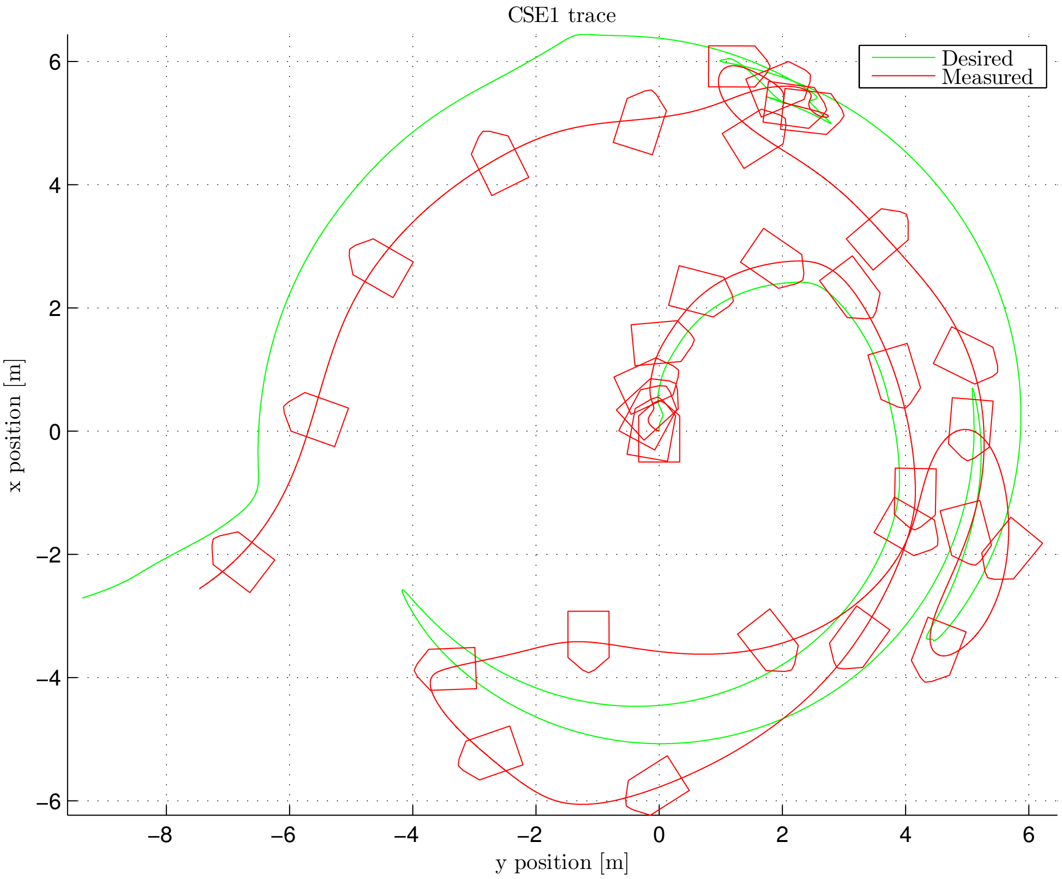


Figure 6.3: HIL: Gesture control, x-y position trace

6.2.2 Gesture control

This test is less suited for HIL testing than the previous one, since the control system is much more dependent on the operator visual feedback. Figure 6.3 shows the simulated path.Tests were run with speed reference and position control. Figure 6.4 shows that the desired position is followed correctly, though not very well, while the velocities seen in 6.5 are more unstable. Although the performance of these tests are not optimal, they were useful to correct bugs and getting more familiar with the system.

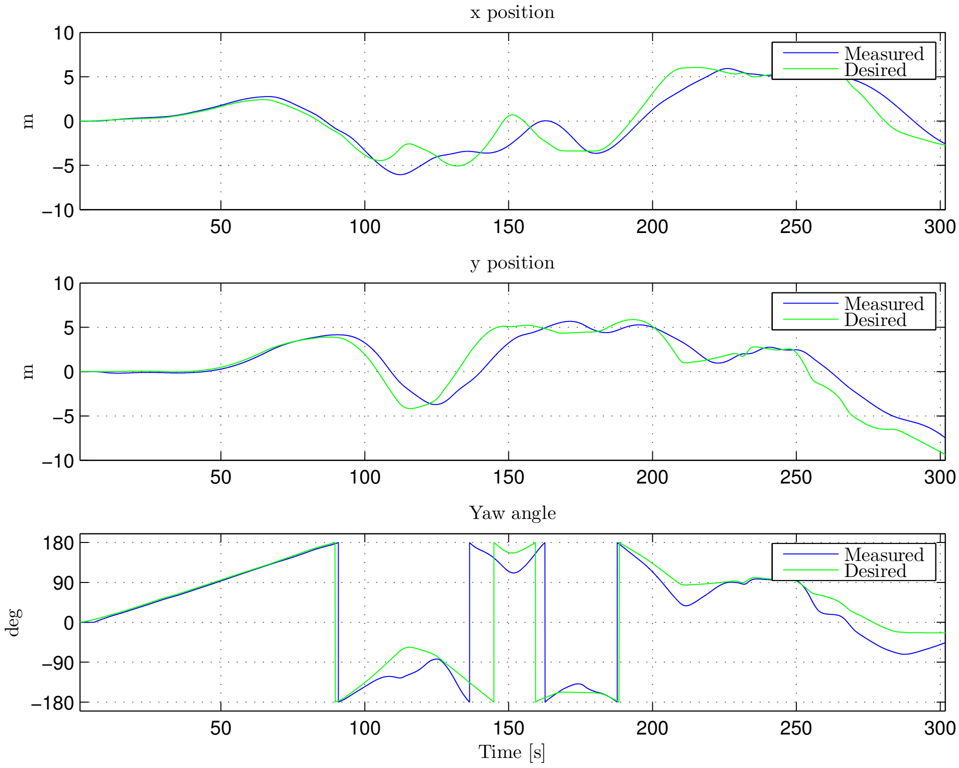


Figure 6.4: HIL: Gesture control, x,y position and yaw angle

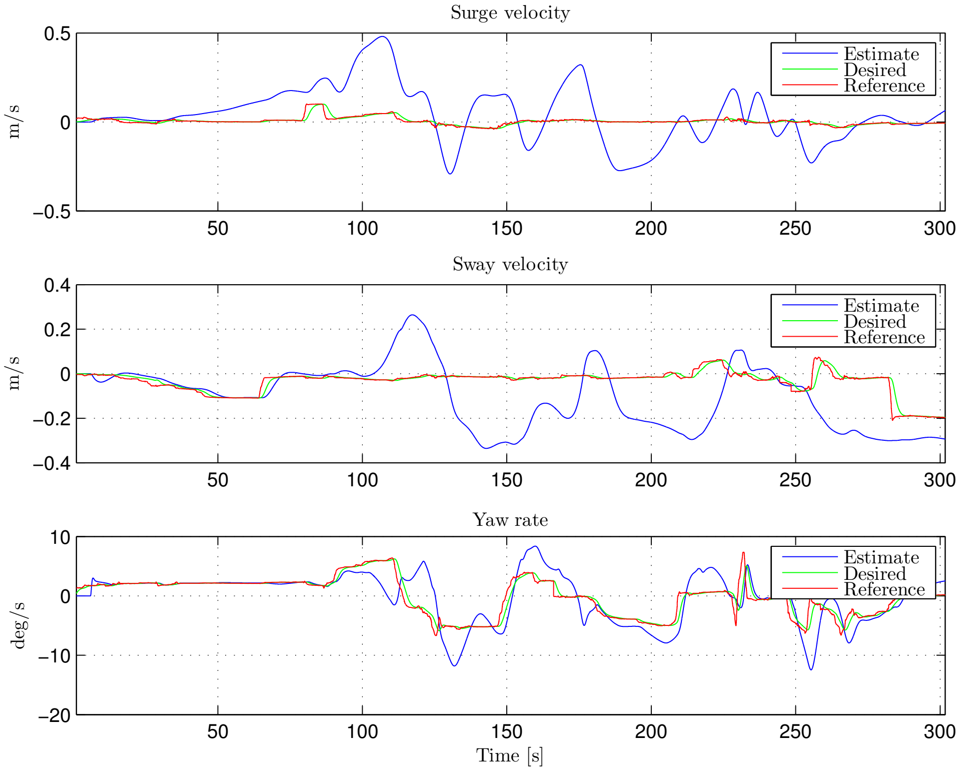


Figure 6.5: HIL: Gesture control, surge, sway velocities and yaw rate

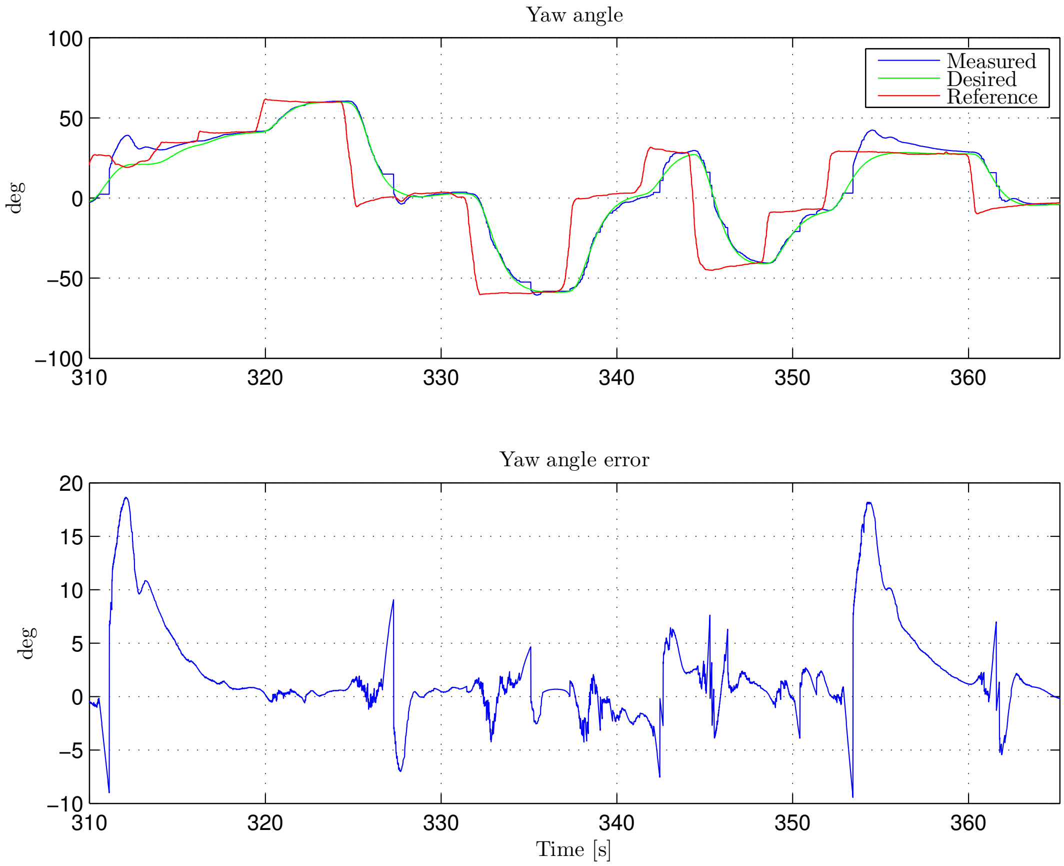


Figure 6.6: CSE1: Direct heading control, yaw angle and error

6.3 Cybership Enterprise 1 Model Scale

This section threats the results in two parts:

1. Control performance: Analyzing how the control algorithm performed, in terms of making the vessel reach its control objective.

2. User experience: Providing a more subjective view on the user experience..

A YouTube video of these results being collected is available (Valle, 2015).

6.3.1 Control Performance - Direct Heading Control

For direct heading control the system is quite responsive. The reference in Figure 6.6 is the input signal (ψ^{oc}), the heading follows the reference properly. As seen in Figure 6.7, the vessel holds its position in x and y within 10 cm. This is a good result, considering the fact that the vessel has sharp changes in heading. The estimated velocities are available from Figure 6.8.

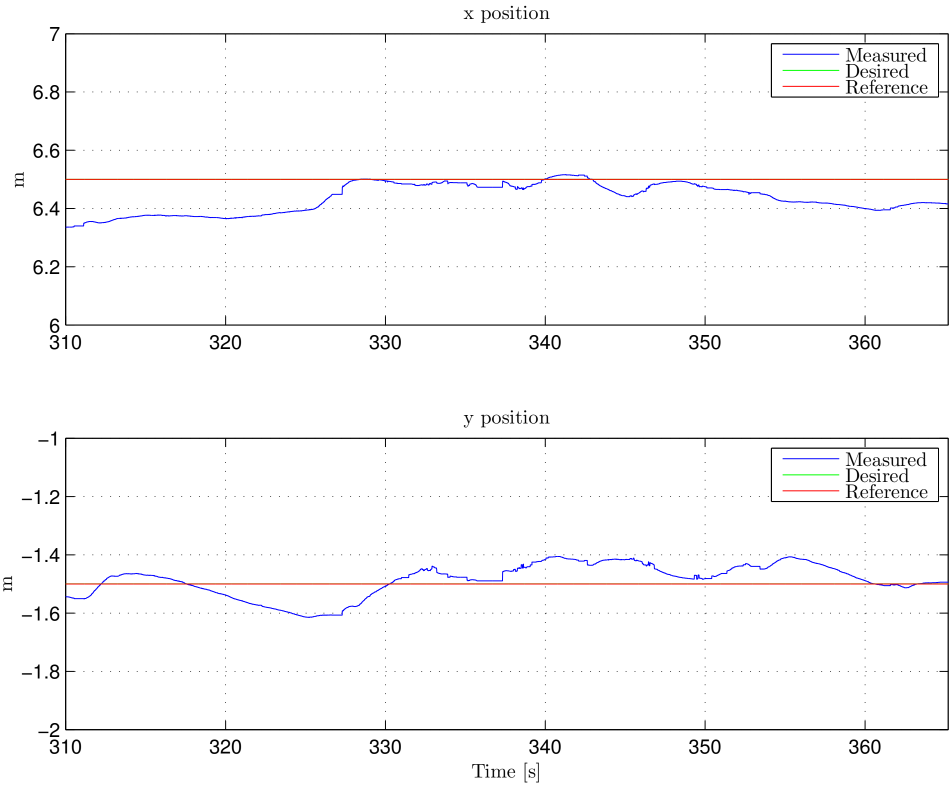


Figure 6.7: CSE1: Direct heading control, x and y position

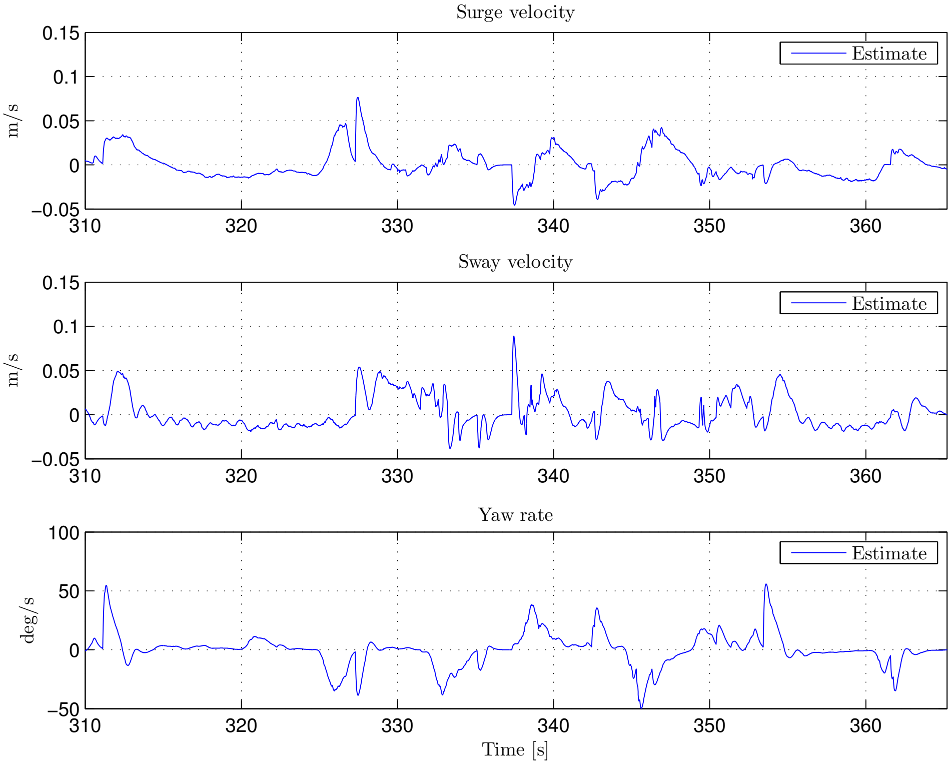


Figure 6.8: CSE1: Direct heading control, estimated surge, sway velocities and yaw rate

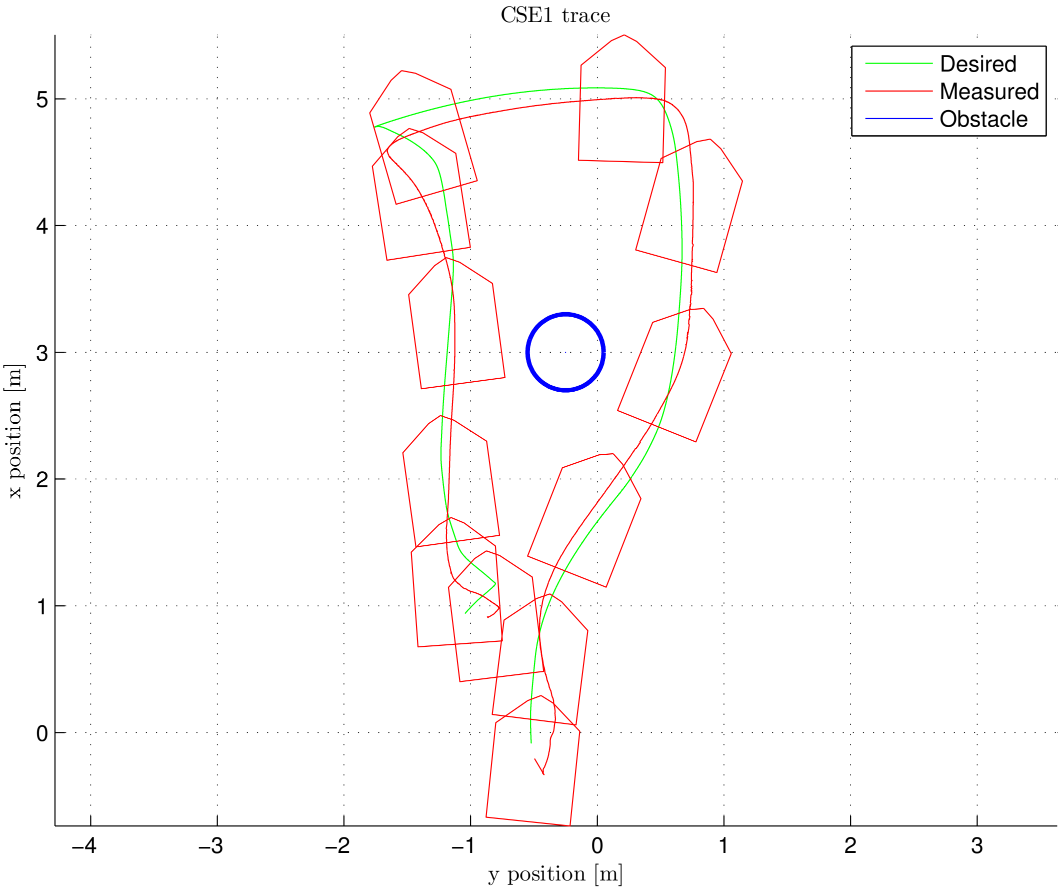


Figure 6.9: CSE1: Gesture control, x-y position trace. CSE1 is moving around an obstacle in the MC lab

6.3.2 Control Performance - Gesture Control

For the gesture control mode, an obstacle was placed in the MC lab and the operator was asked to navigate around it. As seen in Figure 6.9 and Valle (2015) the task is successfully completed. CSE1 is moving counter-clockwise around the obstacle.

In this experiment speed reference with position control has been used. Figure 6.10 shows how the measured position tracks the desired position while Figure 6.11 shows the reference $\boldsymbol{\nu}^{oc}$ with the desired and estimated velocities, respectively $\boldsymbol{\nu}_d$ and $\hat{\boldsymbol{\nu}}$.

The tracking is not perfect, especially in sway direction. The reason is probably related with the not perfect tuning of the controller. Moreover the aggressive tuning for yaw rate ($\hat{r}$) makes the estimates noisy, as seen in Figure 6.5.

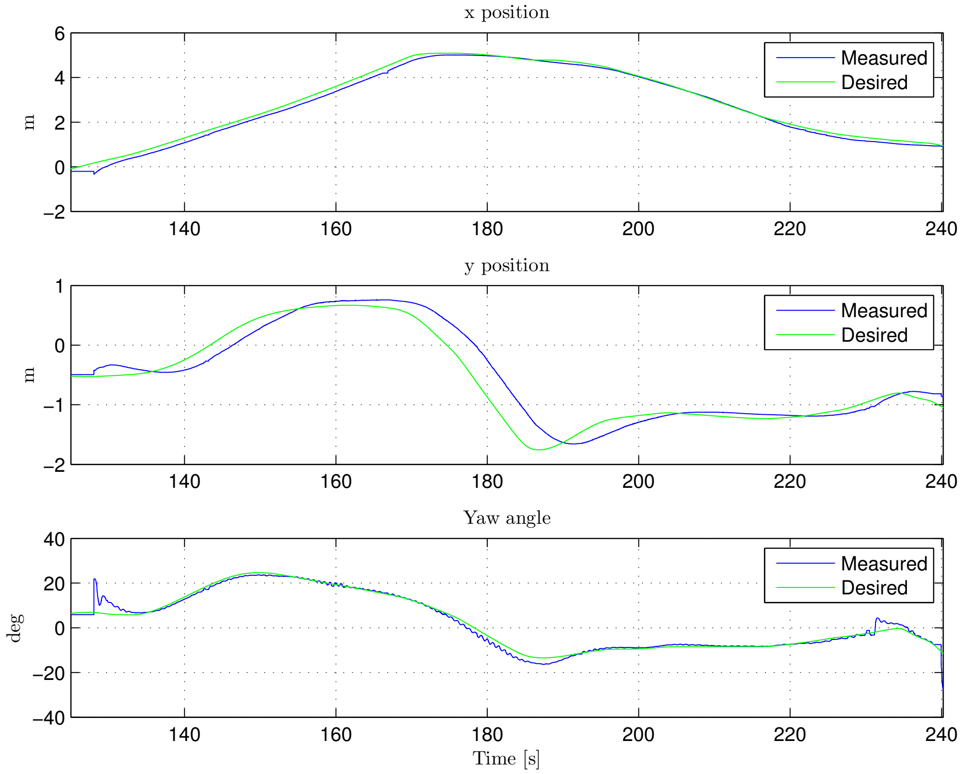


Figure 6.10: CSE1: Gesture control, x, y position and yaw angle

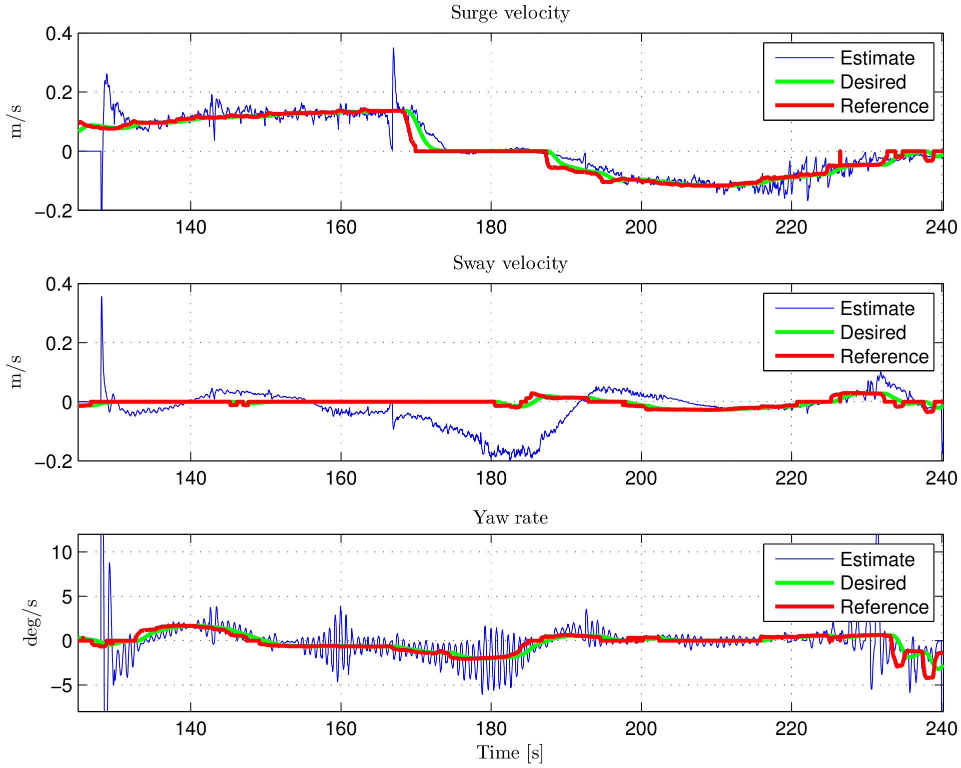


Figure 6.11: CSE1: Gesture control, surge, sway velocities and yaw rate. Estimates are noisy in yaw, and some wild points escape the plot

6.3.3 User experience: CSE1

Here an assessment of the user experience is included. Operations have been conducted with the author (Eirik Valle) as Oculus Rift operator.

The video system struggled with high latency for periods, and this had a negative impact on the user experience, as the visual reference was impaired.

The operator did not experience any simulation sickness during these trials. However, as the operator has much experience with Oculus Rift, and is not prone to motion sickness, this does not necessarily mean that the system does not cause simulator sickness.

Other aspects specific to the control modes that are worth mentioning are listed in the following:

Direct heading control

The direct heading control mode was very responsive to head movements when it was tuned aggressively, this improved the user experience compared to the previous tests with a less aggressive tuning.

Gesture control

When maneuvering the vehicle in the MC lab, it was difficult for the operator to pinpoint the position and velocity of the vessel. To complete the maneuver the operator was dependent on receiving oral commands from someone else in the room.

For this experiment, there was not any information available for the operator of the exact head attitudes. It was then difficult to know if the head was pitching 5 or 10 degrees. The dead-zone from -5 to 5 degrees reduces oscillations, and the observed vessel motion gives an indication. For delicate maneuvers, however, the operator must be aware of what his exact head attitude is.

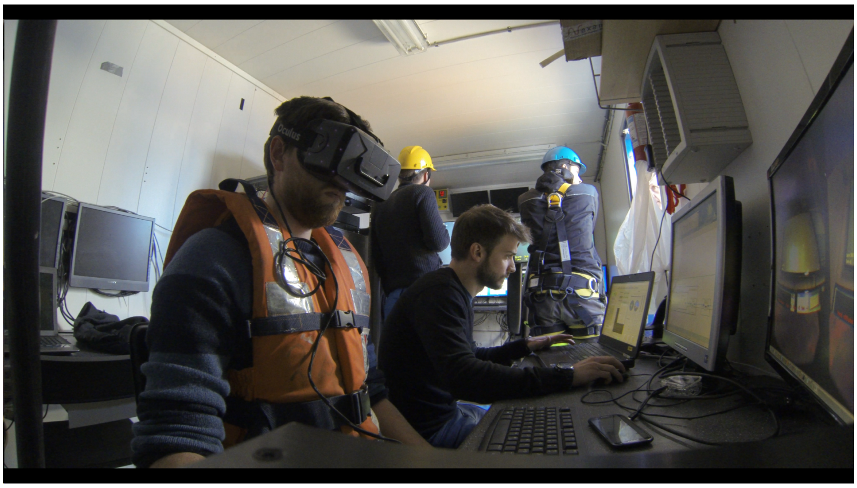


Figure 6.12: Setup in the control container, Eirik Valle as Oculus Rift operator and co-supervisor Mauro Candeloro operating the control system

6.4 SUB-fighter 30k Full Scale

Full scale test of this system was conducted on May 18, 2015. The ROV SF 30k was deployed from R/V Gunnerus.

The ROV was controlled from the control container as seen in Figure 6.12.

More control modes are tested out with the ROV, namely:

1. Feed-forward force control:
   - Straight line maneuver;
2. Speed reference with velocity control:
   - Straight line maneuver;
3. Speed reference with position control:
   - Square test;
   - Object inspection, maneuvering on visual reference.

The vertical DOF value is either kept stable with fixed depth control or altitude[1] control, depth is shown in the position plot even though it is not directly controlled by the Oculus Rift.

In the plots displaying north and east position, the axis are scaled down to local coordinates for readability purposes. The real position can be retrieved by translating the north, east values with ΔN and ΔE.

$$\Delta N = 7036850[m] \text{ and } \Delta E = 570120[m]. \tag{6.1}$$

[1] distance to the seabed

6.4.1 Mission issues

When conducting the full scale tests numerous practical problems were encountered. After an initial postponement of the sea trials due to software issues, the system was launched late in the day. This reduced the experimental time window, hence, position control is tested more thoroughly than force control and velocity control.

Two main issues that have been encountered are explained:

Camera system

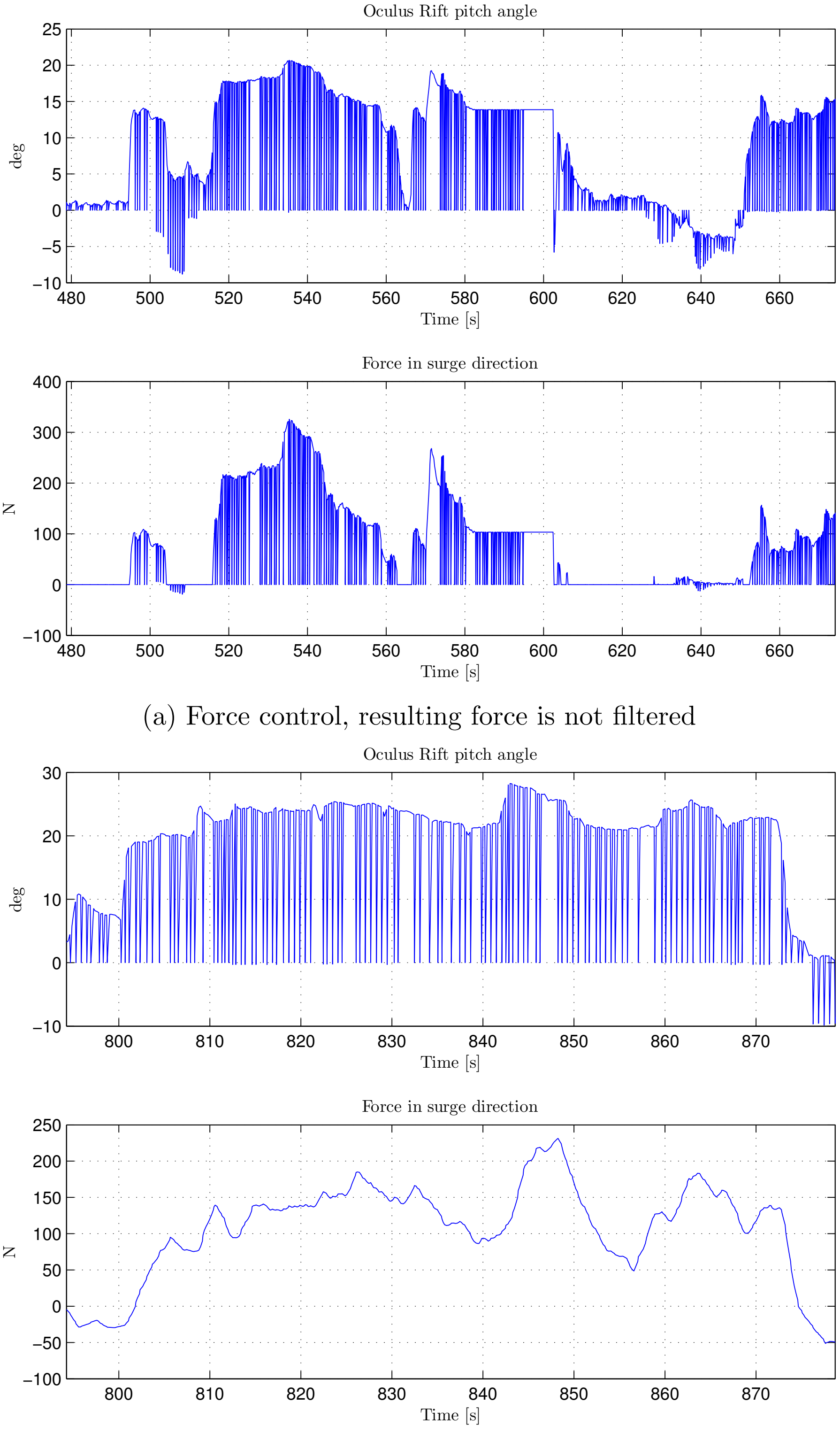
The camera solution failed when submerged most likely due to a faulty seal on the waterproof bottle. However, an ad hoc solution was established, were a web camera was set up to film the main screen. This worked well enough, the Oculus Rift operator had a visual reference when maneuvering the ROV. Obviously this impaired the overall performance of the system.

Measurement wild-points

The Oculus Rift measurements suffered from numerous wild-points when introduced to the control system (this problem should have been anticipated and dealt with by introducing the measurements to for example a low pass filter).

The problem was mainly affecting the force control law, since the two other modes, velocity and position control with speed reference sends the Oculus Rift measurements through the reference model which is a second order low pass filter.

As seen in Figures 6.13a and 6.13b the commanded force is fluctuating violently for force control, but in velocity control the force is stable even though the input angle has the same fluctuations as with force control.



(a) Force control, resulting force is not filtered

(b) Velocity control, resulting force is filtered by reference model

Figure 6.13: Wild points: Oculus Rift pitch angle and the resulting force in surge

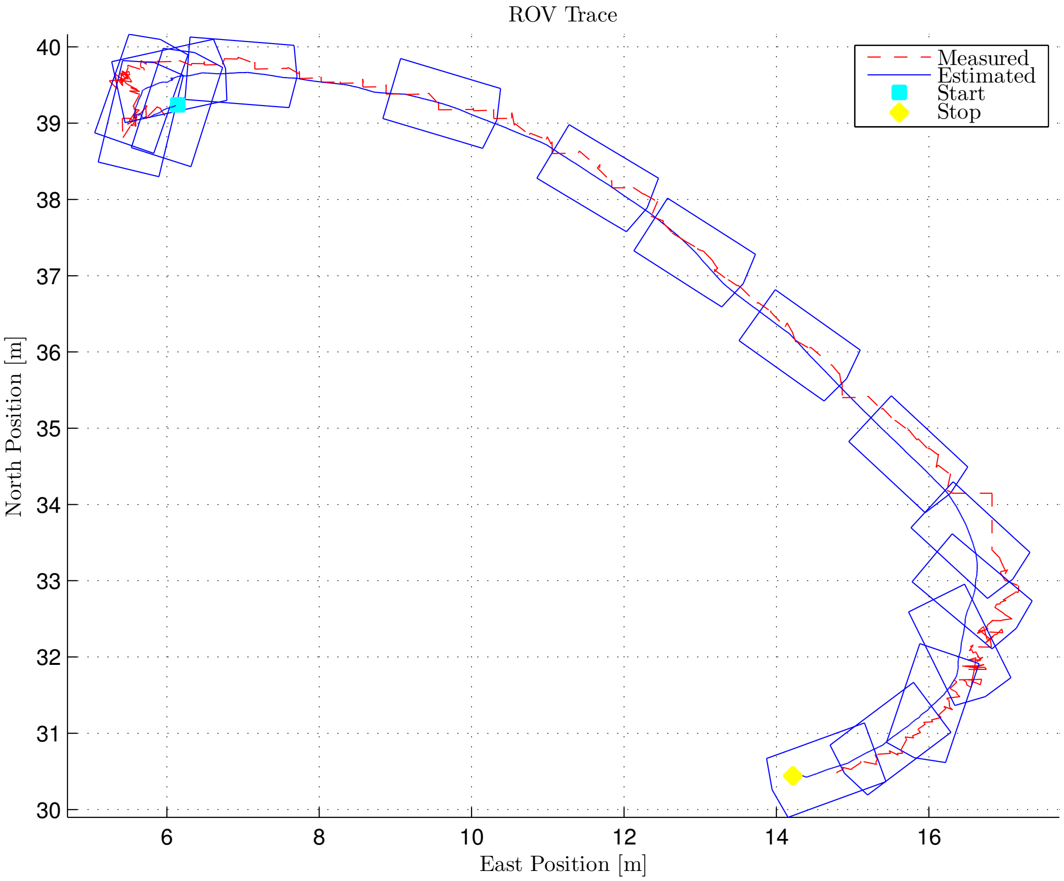


Figure 6.14: SF 30k: Force control, north-east position trace

6.4.2 Feed Forward Force Control

The objective was to move in a straight line. As seen in Figure 6.14 this is accomplished, with a turn at the end and at the beginning. The wild-point problem was not perceived as an issue while operating the ROV.

Figure 6.15 and Figure 6.16 show the estimated and measured positions and velocities with regards to time.

As there is no feedback control, it is difficult to quantify the performance of the control system, since there is no desired states to compare with actual states. However, the operator was able to follow the required maneuver, achieving the goal.

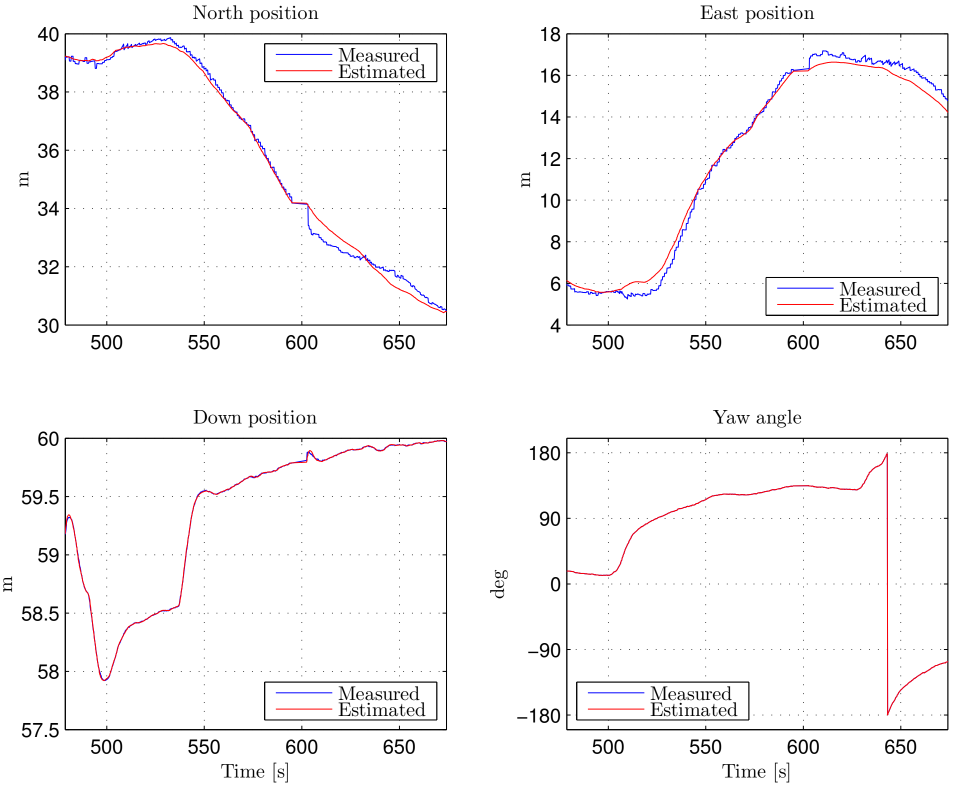


Figure 6.15: SF 30k: Force control, north, east, down position and yaw angle

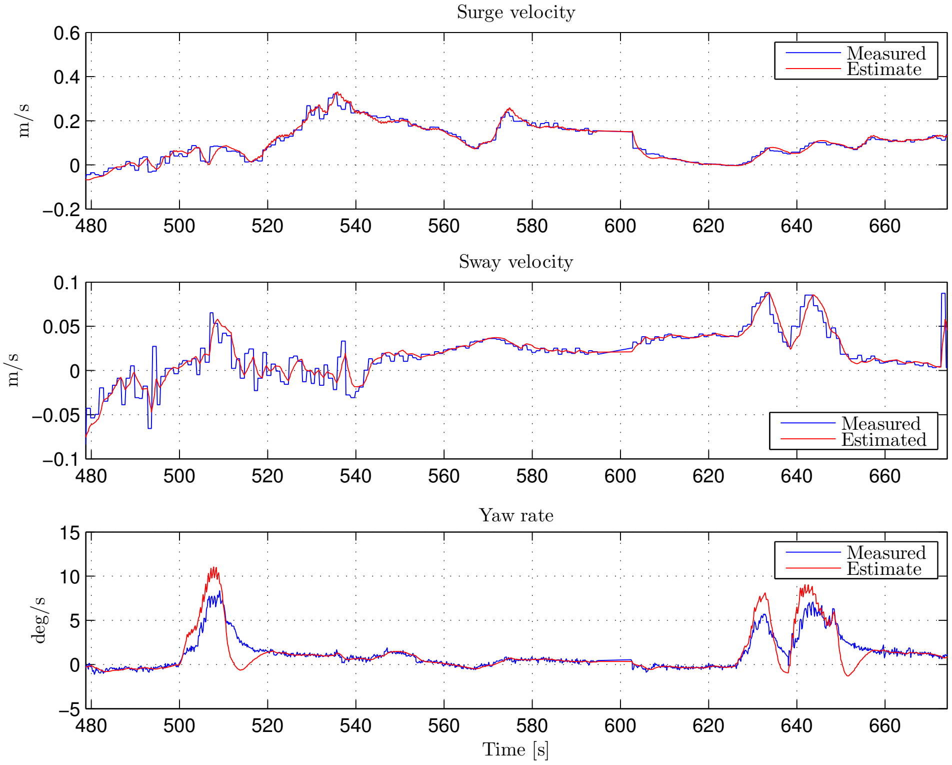


Figure 6.16: SF 30k: Force control, surge, sway velocities and yaw rate

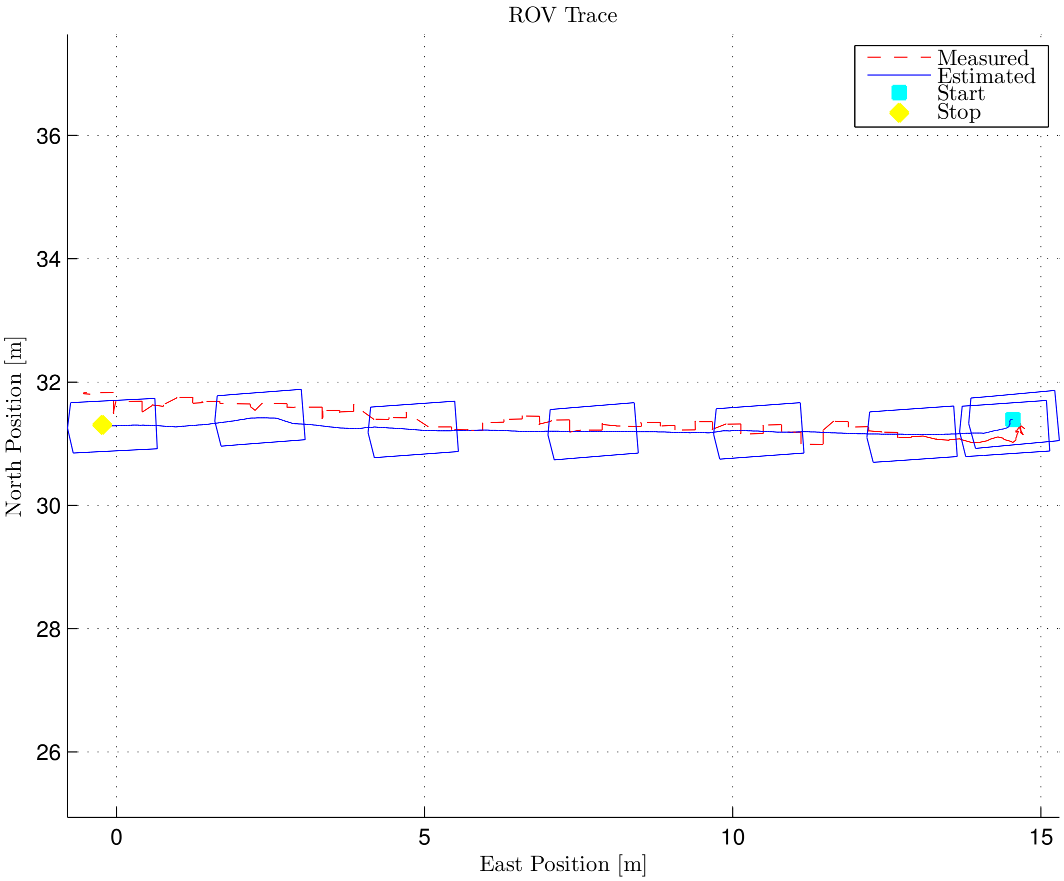


Figure 6.17: SF 30k: Velocity control, north-east position trace

6.4.3 Speed Reference, Velocity Control

For velocity control mode a straight line test was also conducted. As seen in Figures 6.17 the ROV is able perform the maneuver in a better way than with force control.

From Figure 6.18 and Figure 6.19 the individual positions and velocities can be seen. Notice that only the velocities have desired values in this control mode.

The ROV is able to follow the desired values in surge, moving forwards and too keep the velocities in sway and yaw at zero until when the ROV is decelerating. Then it is possible to observe the ROV loosing control in sway and yaw. At this time it is also possible to observe the measured and estimated values in north and east diverging, meaning this could be the cause of the problem.

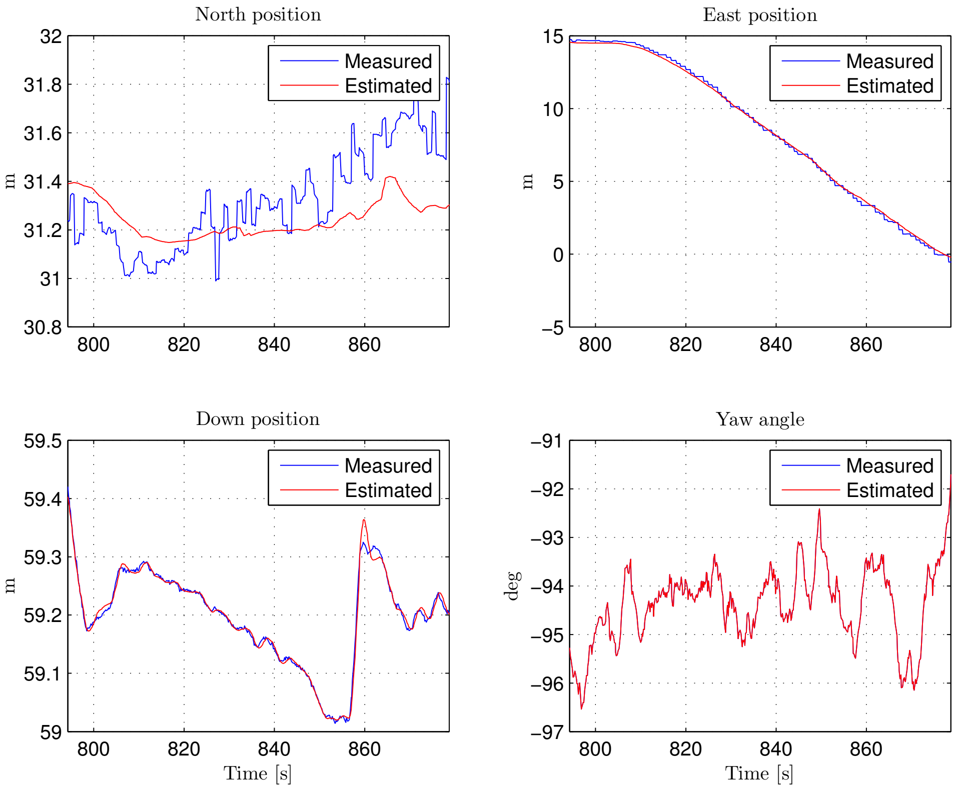


Figure 6.18: SF 30k: Velocity control, north, east, down position and yaw angle

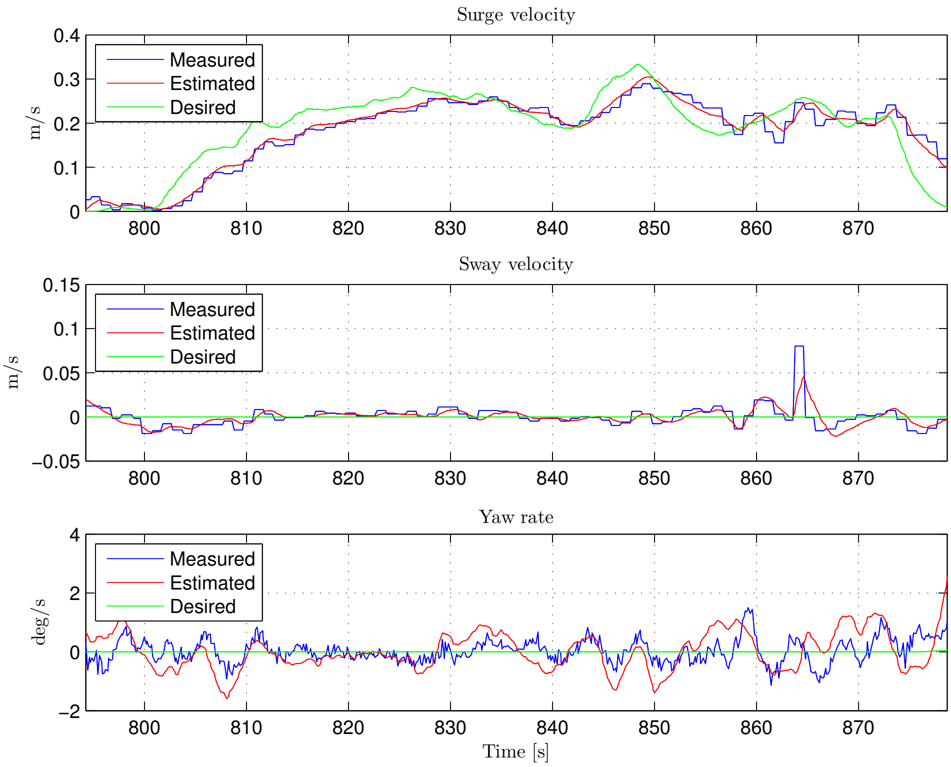


Figure 6.19: SF 30k: Velocity control, surge, sway velocities and yaw rate

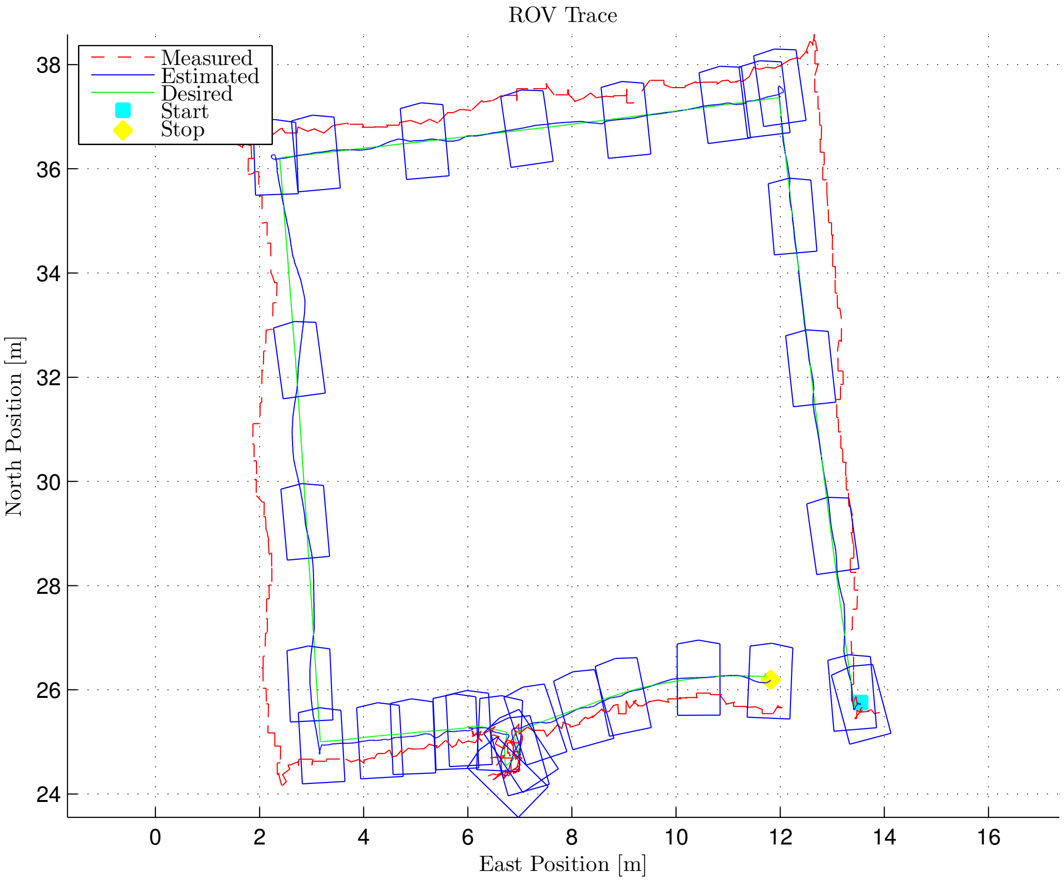


Figure 6.20: SF 30k: Square maneuver, north-east position trace

6.4.4 Speed Reference, Position Control

Two tests are presented here: a square maneuver and an inspection maneuver of an object discovered on the sea bottom.

Square maneuver

For the square maneuver, the objective was to follow a square path using the ROV. For this test oral commands were given from the operators of the AUR lab control system to the Oculus Rift operator. This was totally necessary to pull of the maneuver as the visual feedback was inadequate to know the position in the desired square path.

As seen in Figure 6.20 a nice square was drawn on the map using the ROV. The square maneuver is done counterclockwise.

In Figures 6.21 and Figure 6.22, the individual states can be observed following their desired values with good performance, some oscillations are seen in sway and yaw.

At about time = 1640s, seen in the trace plot (Figure 6.22) in the middle of south border. The Oculus Rift software lost connection for unknown reasons. After the problem was fixed, the vehicle was driven back to the path to complete the task.

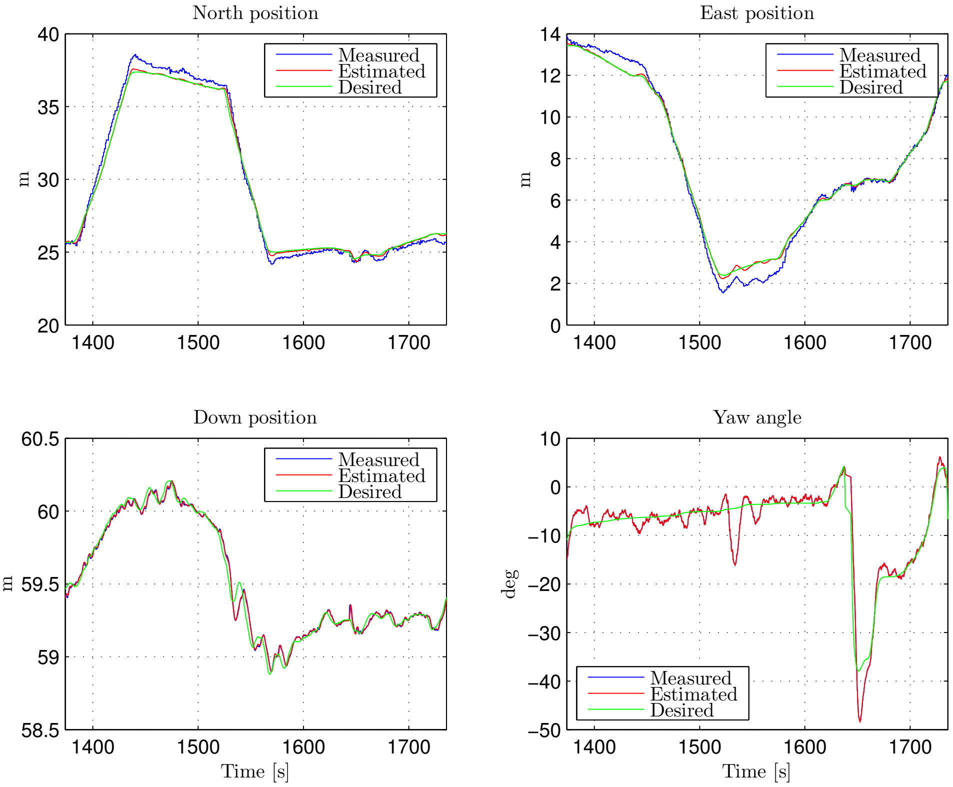


Figure 6.21: SF 30k: Square maneuver, north, east, down position and yaw angle

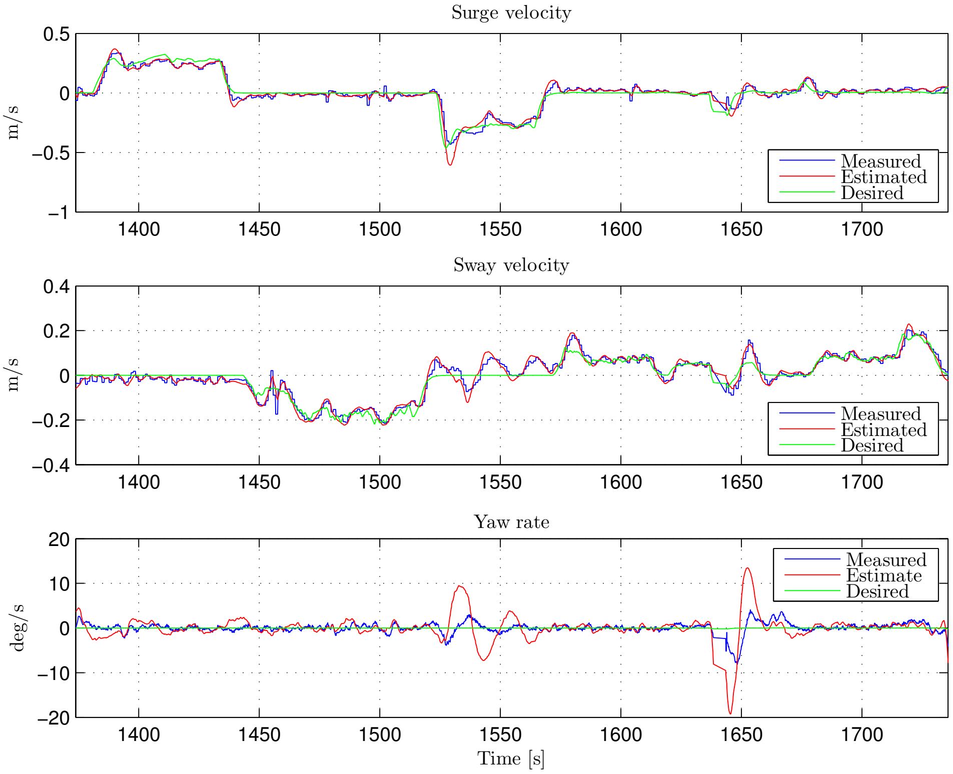


Figure 6.22: SF 30k: Square maneuver, surge, sway velocities and yaw rate

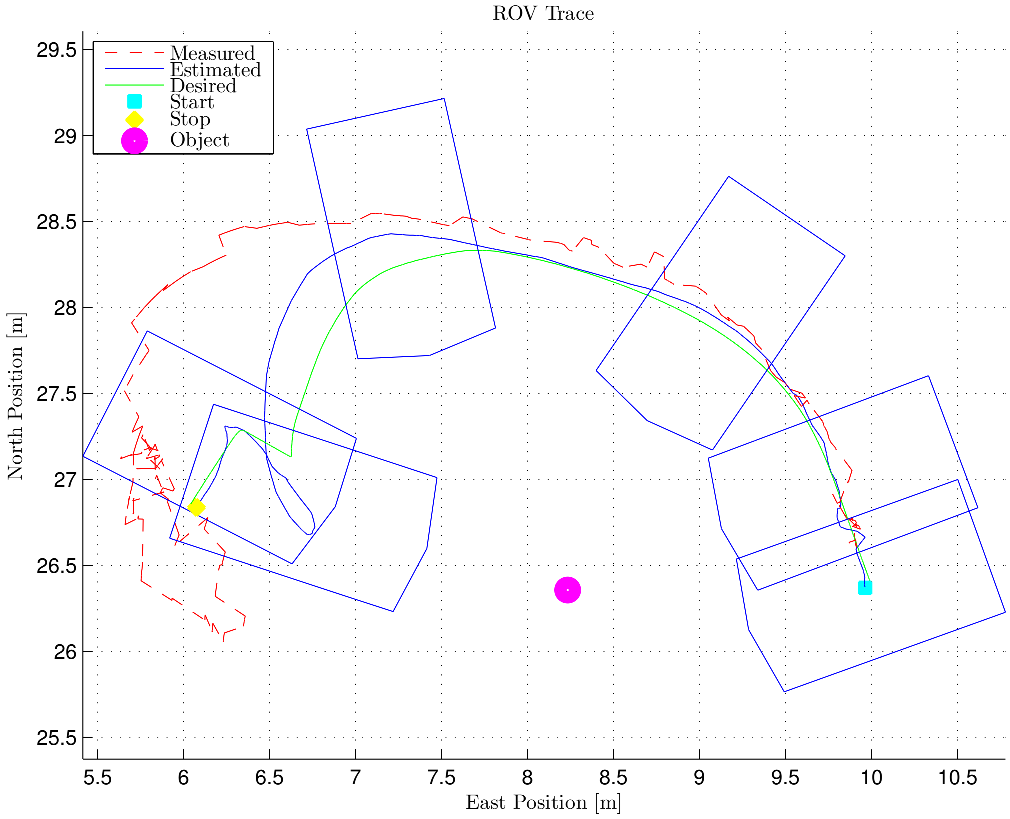


Figure 6.23: SF 30k: Object inspection, north-east position trace. The circle represents the inspected object

Object inspection

A maneuver based on visual feedback was attempted, since the previous maneuver was carried out with the help of oral commands from someone with live updates from the North-East plot.

A bottle was located on the seabed and the task was to maneuver around the bottle keeping it in sight. In Figure 6.23 the bottle is marked with the purple circle and the ROV hovering in a half circle around it can be observed.

In Figures 6.24 and 6.25 the individual states can be observed following their desired values.

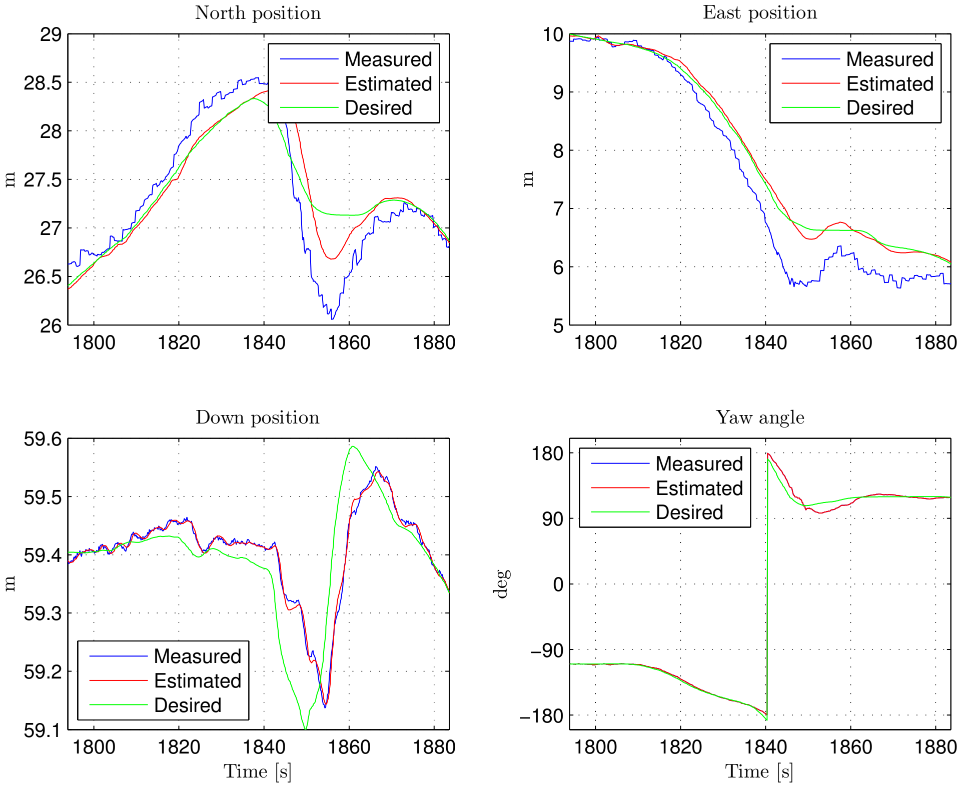


Figure 6.24: SF 30k: Object inspection, north, east, down position and yaw angle

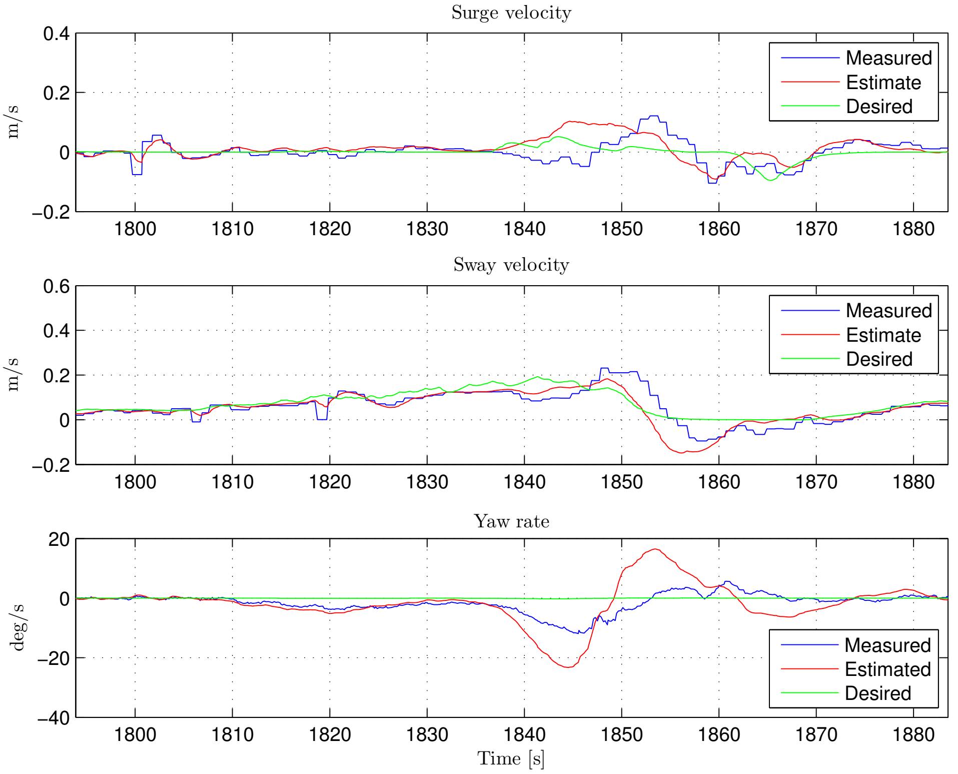


Figure 6.25: SF 30k: Object inspection, surge, sway velocities and yaw rate

6.4.5 User Experience

As mentioned above the original camera system failed, and reducing drastically the quality of the visual input for the operator. However, the web camera filming the screen had low latency and was still comfortable. It was also possible to observe the heading and depth of the ROV through the AUR lab camera system, which was very useful when the altitude of the vehicle was too high to have a visual reference of the sea bottom.

For the full scale tests, the head attitude of the operator was available on the heads up display. This was a significant improvement from the earlier version used for model scale tests, especially in terms of station keeping and fine maneuvering.

When it comes to the different control modes the operator felt little difference between direct force, velocity- or position control. However, the testing of velocity and force control were given less time of testing due to mission time constraints.

Also this time no motion sickness was experienced by Oculus Rift operator. Even though the continuous operation time with Oculus Rift on went close to 30 minutes.

Chapter 7

Conclusion

This thesis shows that a HMD telepresence system can be used to control and maneuver a ROV using measured head attitude. The system is recommended for further development and testing, in order to obtain a reliable system to conduct coordinated ROV and manipulator operations. This will allow one operator to to control both ROV position and arm simultaneously.

Gesture based guidance, transforming head attitude to a speed reference is implemented. Feed forward force, velocity and position control is able to maneuver SF 30k, where position control performed best. The Oculus Rift interface software is connected both with CSE1 and the AUR lab control system.

The model and full scale test found that the following elements are important for the user experience:

- visual reference points are important when maneuvering with the system;
- low latency on the video;
- system must be responsive to operator input (i.e head movement);
- information about head attitude.

7.1 Further Work

Further work recommendations are divided in two groups: High priority improvements that should be implemented for the system to function properly and lower priority improvements that would improve the system performance. Ideas that could be investigated are also presented.

7.1.1 High priority

- **Integrate ROV camera system with Oculus Rift**
  If the Oculus Rift is going to be used with the AUR lab ROVs, there should be a priority to integrate the AUR lab video system with Oculus Rift. Since this is already supplying high quality video, even with possibilities of stereo video, this would greatly improve the user experience.

- **Oculus Rift software sample rate sensitivity**
  The Oculus Rift software needs to be adjusted so that it is not sensitive to the sampling rate of the system receiving signals from it. This would make the software more versatile and eliminate the need for quick fixes of the problem.

  Two ways of fixing this can be proposed:

  - Either by sending information over the UDP protocol, instead of TCP. Meaning that the system would not have to wait for confirmation that the package is received before sending the next package.
  - Create a separate thread for the server sending signals within the C++ Oculus Rift software.

- **Crash prevention**
  The Oculus Rift system sometimes crash for no apparent reason. This needs to be investigated and fixed.

7.1.2 Lower priority

- **Stereo Vision**
  Expand the software to handle stereo vision and acquire stereo camera. This would enhance the user experience greatly. There is example code by Davies et al. (2015*a*) for this purpose.

- **Heads up display**
  As mentioned in the user experience section, information on the vehicle being controlled could be useful to display on the Oculus Rift screen. Like for example heading, depth, velocities, alarms, altitude etc.

- **Motion sickness**
  Motion sickness with Oculus Rift is not experienced by the operator during our tests. However, to conclude on this matter, a higher number of test, with many operators (with different level of experience) should be performed, since all the conclusion that this thesis may report would only be describing a special case, that could not be generalized.

- **Permanent CSE1 camera**
  If CSE1 is to be used as a telepresence system, CSE1 should be fitted with a permanent camera system.

- **CSE1 observer**
  CSE1 requires a new and improved observer in order to receive proper velocity estimates.

- **Other opportunities**
  Just to mention some of the many opportunities that are made possible with the a telepresence system:

  - Augmented Reality possibilities with OpenCV.
  - New input devices, hand sensors etc. Utilize the fact that the operators hands are free.
  - Cameras filming in 360 degrees of your surroundings. This way it would be possible to look around without having to move ROV and telepresence would be enhanced.

Bibliography

Ackerman, E. (2015), 'Oculus Rift-Based System Brings True Immersion to Telepresence Robots'. IEEE Spectrum, viewed 2015-06-08
URL: `http://goo.gl/6oAUB5`.

Apache License (2007), 'Apache license version 2.0'. Viewed 2015-06-08
URL: `http://www.apache.org/licenses/LICENSE-2.0.html`.

Berg, V. (2012), 'Development and Commissioning of a DP system for ROV SF 30k', *Master thesis, Department of Marine Technology, NTNU* .

Bohn, D. (2015), 'Google's big jump into virtual reality'. The Verge, viewed 2015-06-08
URL: `http://goo.gl/Ckr7IE`.

Breivik, M. and Sand, G. (2009), 'Jens Glad Balchen: A Norwegian Pioneer in Engineering Cybernetics'.

Candeloro, M., Sørensen, A. J., Longhi, S. and Dukan, F. (2012), 'Observers for Dynamic Positioning of ROVs with Experimental Results Manoeuvring and Control of Marine Craft', *Proc. MCMC'2012, 9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano, Italy, 19-21 September* .

Davies, B. A., Bryla, K. and Benton, A. (2015*a*), *Oculus Rift In Action*, Manning Publications Co., Shelter Island, USA.

Davies, B. A., Bryla, K. and Benton, A. (2015*b*), 'Oculus Rift In Action GitHub repository'. GitHub.com, viewed 2015-06-08
URL: `https://github.com/OculusRiftInAction/OculusRiftInAction`.

DORA (2015), 'Dora: The future of telepresence'. Video, viewed 2015-06-08
URL: `https://vimeo.com/124993083`.

Drangeland, Å. (2015), 'Nytt 3D-kamera skal filme i 360 grader'. Gemini (Norwegian), viewed 2015-06-08
URL: `http://gemini.no/2015/04/3d-kamera-som-filmer-i-360-grader/`.

Dukan, F. (2014), 'ROV Motion Control Systems', *PhD thesis, NTNU, Trondheim, Norway* .

Faltinsen, O. (1990), *Sea loads on ships and offshore structures*, Cambridge university press, Cambridge, United Kingdom.

Follestad, J., Sandved, F. and Valle, E. (2014), 'Low cost ROV design, based on testing, simulations and analysis of OpenROV', *Project Thesis, Department of Marine Technology, NTNU* .

Fossen, T. I. (2011), *Handbook of marine craft hydrodynamics and motion control*, John Wiley & Sons, Hoboken, NJ, USA.

Google Inc. (2015), 'We're graduating from Google[x] labs'. Viewed 2015-06-08
URL: `https://plus.google.com/+GoogleGlass/posts/9uiwXY42tvc`.

Kalman, R. E. (1960), 'A new approach to linear filtering and prediction problems', *Journal of Fluids Engineering* **82**(1), 35–45.

Khlebovich, P. (2015), 'Ip webcam, google play store'. Viewed 2015-06-08
URL: `https://goo.gl/9TtG38`.

Kirkeby, M. (2010), 'Comparison of controllers for dynamic positioning and tracking of ROV Minerva', *Master thesis, Department of Marine Technology, NTNU* .

Lockheed Martin (2015), 'F35 helmet'. Viewed 2015-06-08
URL: `https://www.f35.com/about/capabilities/helmet`.

MathWORKS (2015), 'Simulink home page'. Viewed 2015-06-08
URL: `http://se.mathworks.com/products/simulink/`.

Microsoft (2015*a*), 'Microsoft hololens'. Video, viewed 2015-06-08
URL: `https://www.microsoft.com/microsoft-hololens/`.

Microsoft (2015*b*), 'Windows sockets 2'. Viewed 2015-06-08
URL: `https://goo.gl/Zq5Ms2`.

National Instruments (2015), 'NI CompactRIO'. Viewed 2015-06-08
URL: `http://www.ni.com/compactrio/`.

NTNU (2015), 'Handbook of Marine HIL simulation and Marine cybernetics laboratories'. Viewed 2015-08-02
URL: `http://www.ntnu.no/imt/lab/cybernetics`.

Oculus VR, LCC (2014*a*), *Oculus Developer Guide, SDK Version 0.4*, Oculus VR LCC, Menlo Park, CA, USA.

Oculus VR, LCC (2014*b*), 'Oculus rift dk2 specifications'. Viewed 2015-06-08 URL: `http://www.oculus.com/dk2/`.

Orsten, A. (2014), 'Automatic Reliability-based Control of Iceberg Towing in Open Waters', *Master thesis, Department of Marine Technology, NTNU* .

Oxford University Press (2015), 'Oxford Dictionaries'. Online Dictionary: Virtual reality, augmented reality and telepresence, viewed 2015-06-08, URL:
`http://www.oxforddictionaries.com/definition/english/augmented-reality`
`http://www.oxforddictionaries.com/definition/english/virtual-reality`
`http://www.oxforddictionaries.com/definition/english/telepresence`.

Qvale, P., Dalløkken, P. E., Ramsdal, R. and Lygre, E. T. (2015), 'Norske Ingeniørbragder'. Teknisk Ukeblad, viewed 2015-06-06
URL: `http://goo.gl/ZyU3Lx`.

Robertson, A. (2015), 'The finished Oculus Rift is shipping early next year'. The Verge, viewed 2015-06-08
URL: `http://goo.gl/rP4tk7`.

Schneider, E. and Mueller, H. (1941), 'Blade wheel'. US Patent 2,250,772
URL: `https://www.google.com/patents/US2250772`.

Skåtun, H. N. (2011), 'Development of a DP system for CS Enterprise I with Voith Schneider thrusters', *Master thesis, Department of Marine Technology, NTNU* .

Skjetne, R. (2005), 'The Maneuvering Problem', *PhD-thesis, NTNU, Trondheim, Norway* .

SNAME (1950), The Society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid, *in* 'Technical and Research Bulletin No 1-5'.

Sørensen, A. J. (2012), 'Marine Control Systems Propulsion and Motion Control of Ships and Ocean Structures Lecture Notes', *TMR4240 Marine control systems, NTNU* .

Steptoe, W. (2014), 'AR demonstration video'. Video, viewed 2015-06-08
URL: `https://goo.gl/1v31ZT`.

Steptoe, W., Julier, S. and Steed, A. (2014), Presence and discernability in conventional and non-photorealistic immersive augmented reality, *in* 'Mixed and Augmented Reality (ISMAR)', IEEE, pp. 213–218.

Steuer, J. (1992), ‘Defining virtual reality: Dimensions determining telepresence’, *Journal of communication* **42**(4), 73–93.

Svendby, E. (2007), ‘Robust control of ROV/AUVs’, *Master thesis, Department of Marine Technology, NTNU* .

Tolpinrud, E. (2012), ‘Development and Implementation of Computer-Based Control System for ROV with Experimental Results’, *Master thesis, Department of Marine Technology, NTNU* .

Tran, N. D. (2014), ‘Line-Of-Sight-based maneuvering control design, implementation, and experimental testing for the model ship C/S Enterprise I.’, *Master thesis, Department of Marine Technology, NTNU* .

Urke, E. H. (2014), ‘Norwegian Army driving armored vehicle using Oculus Rift’. Teknisk Ukeblad, viewed 2015-06-08
URL: `http://goo.gl/9F9aK8`.

Valle, E. (2015), ‘Demonstration of CSE1 and Oculus Rift implementation’. Video, viewed 2015-06-08
URL: `https://goo.gl/ASvPFY`.

Voith GmbH (2015), ‘Voith Schneider Propeller’. Viewed 2015-06-08
URL: `http://goo.gl/Y5MD3o`.

Yao, R., Heath, T., Davies, A., Forsyth, T., Mitchell, N. and Hoberman, P. (2014), ‘Oculus VR Best Practices Guide’.

Zemeckis, R. (1989), ‘Back to the Future Part II’, Universal Pictures. [Film].

Appendix A

Vessel Parameters

In this Appendix parameters for CSE1 and SF 30k are found.

A.1 Cybership Enterprise 1

A.1.1 Parameter values

Parameter	Value	Parameter	Value
m	14.79	$X_{\dot{u}}$	-2
I_z	1.76	$Y_{\dot{v}}$	-10
x_g	0.0375	$Y_{\dot{r}}$	0
y_g	0	$N_{\dot{r}}$	-1

Table A.1: Rigid body and added mass parameters

Note the added mass therms are Cybership 2 values used in Skjetne (2005), this vessel is significantly larger and thus the values are probably not accurate for CSE1.

Parameter	Value	Parameter	Value	Parameter	Value
X_u	-0.6555	Y_v	-1.33	N_v	0
X_{uu}	0.3545	Y_{vv}	-2.776	N_{vv}	-0.2088
X_{uuu}	-3.787	Y_{vvv}	-64.91	N_{vvv}	0
X_v	0	Y_r	-7.25	N_r	-1.9
X_{vv}	-2.433	Y_{rr}	-3.45	N_{rr}	-0.75
X_{vvv}	0	Y_r	0	N_r	0
		Y_{rv}	-0.805	N_{rv}	0.13
		Y_{vr}	-0.845	N_{vr}	0.08

Table A.2: Hydrodynamic damping parameters in surge, sway and yaw

A.1.2 Gains

$$\boldsymbol{K}_p = \text{diag}\{3,\ 3,\ 2\} \tag{A.1}$$

$$\boldsymbol{K}_d = \text{diag}\{7,\ 7,\ 5\} \tag{A.2}$$

$$\boldsymbol{K}_{oc}^{\nu} = \text{diag}\{0.3,\ 0.2,\ 0.1\} \tag{A.3}$$

$$\boldsymbol{\Delta} = \text{diag}\{1,\ 1,\ 1\} \tag{A.4}$$

$$\boldsymbol{\Omega} = \text{diag}\{0.5,\ 0.5,\ 2\} \tag{A.5}$$

A.2 SUB-fighter 30k

A.2.1 Parameter values

$$\boldsymbol{M}_{RG}^{CG} = \begin{bmatrix} 1862.9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1862.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1862.9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 525.4 & 1.4 & 33.4 \\ 0 & 0 & 0 & 1.4 & 794.2 & 2.6 \\ 0 & 0 & 0 & 33.4 & 2.6 & 691.2 \end{bmatrix} \tag{A.6}$$

$$\boldsymbol{M}_A = \begin{bmatrix} 779.8 & -6.9 & -103.3 & 8.5 & -165.5 & -7.8 \\ -6.9 & 1222 & 51.3 & 409.4 & -5.8 & 62.7 \\ -103.3 & 51.3 & 3659.9 & 6.1 & -386.4 & 10.7 \\ 8.5 & 409.4 & 6.1 & 534.9 & -10.0 & 21.0 \\ -165.5 & -5.8 & -386.4 & -10 & 2.69 & -1.1 \\ -7.8 & 62.7 & 10.7 & 21.0 & -1.1 & 224.3 \end{bmatrix} \tag{A.7}$$

$$\boldsymbol{M} = \boldsymbol{M}_{RG}^{CG} + \boldsymbol{M}_A \tag{A.8}$$

$$\boldsymbol{D} = -\text{diag}\{74.82,\ 69.48,\ 728.40,\ 268.8,\ 309.77,\ 105\} \tag{A.9}$$

A.2.2 Gains

$$\boldsymbol{K}_p = \mathrm{diag}\{1000,\ 1000,\ 3500,\ 0,\ 0,\ 2000\} \tag{A.10}$$

$$\boldsymbol{K}_i = \mathrm{diag}\{10,\ 10,\ 20,\ 0,\ 0,\ 50\} \tag{A.11}$$

$$\boldsymbol{K}_d = \mathrm{diag}\{1000,\ 1000,\ 2000,\ 0,\ 0,\ 1000\} \tag{A.12}$$

$$\boldsymbol{K}_{oc}^{\nu} = diag\{0.8,\ 0.3,\ 0.5\} \tag{A.13}$$

$$\boldsymbol{K}_{oc}^{\tau} = diag\{1187,\ 838,\ 1965,\ 0,\ 0,\ 1270\} \tag{A.14}$$

$$\boldsymbol{\Delta} = \mathrm{diag}\{1,\ 1,\ 1,\ 1,\ 1,\ 1\} \tag{A.15}$$

$$\boldsymbol{\Omega} = \mathrm{diag}\{0.8,\ 0.8,\ 0.8,\ 0.8,\ 0.8,\ 0.8\} \tag{A.16}$$