CyberShip Enterprise I

User Manual

NTNU – Trondheim
Norwegian University of
Science and Technology

Faculty of Engineering Science and Technology

Department of Marine Technology

Preface

The purpose of this document is to provide a manual that makes it easier to utilize CyberShip Enterprise I (CSE1), and in particular to present software and hardware specific to CSE1. For information about the Marine Cybernetics Laboratory (MCLab) and its systems, the reader is referred to the MCLab Handbook, which can be found on GitHub.

This manual was revised during spring 2020 by E. Gauslaa, J. S. Jensen and C. Fleischer, and Chapter 7–8 were added.

Structure of document

This user manual is divided in two parts:

- Technical description (hardware, software, mathematical models)
- Operation manual (software deployment, launching/demolition instructions, tuning of hardware, force measurements)

Further work

CSE1 has some known errors. The following list is suggested for further work:

- Carry out a new bollard pull test, as the maximum thrust and moment are not correct. *This was carried out January 2020.*

Table 1: CSE1 main data

Parameter	Value
Length over all	1.105 [m]
Beam	0.248 [m]
Weight	14.11 [kg]
Scale	1:50
IP-address(port 1)	192.168.0.75
IP-address(port 2)	192.168.1.21
RPi IP-address	192.168.1.22
RPi Port Number	51717
Qualisys body	(550, 0, -500) [mm]
MATLAB Version	2016b
LabVIEW Version	2017
VeriStand Version	2017

Contents

Part I

Technical description

Chapter 1

Hardware

1.1 Introduction to CSE1

The CSE1 was initially bought in 2009, as a fully configured model boat named "Aziz" and built by Model Slipway. Due to requirements for master and PhD experiments, the model was refitted by Skåtun (2011). The work performed on CSE1 includes, but is not limited to, dynamic positioning systems, maneuvering systems and path following, as well as navigation with virtual reality.

The vessel is a 1:50 scale model of a tug boat. It is fitted with two Voith Schneider propellers (VSP) astern and one bow thruster (BT). The main dimensions of the vessel in terms of length over all (LOA), breadth (B) and weight displacement (Δ) are presented in Table 1.1.

Table 1.1: Main dimensions of CSE1

LOA	1.105[m]
B	0.248 [m]
Δ	14.11 [kg]

1.1.1 Literature

The development of CSE1 is a product of research from several theses, which contain complementary information on the theory applied to the system.

Journals and conferences

- LOS guidance for towing an iceberg along a straight-line path (Orsten et al., 2014)

Specialization projects and master theses

- Development of a DP system for CS Enterprise I with Voith Schneider thrusters (Skåtun, 2011)
- Development of a modularized control architecture for CS Enterprise I for path-following based on LOS and maneuvering theory (Tran, 2013)
- Automatic Reliability-based Control of Iceberg Towing in Open Waters (Orsten, 2014)
- Line-Of-Sight-based maneuvering control design, implementation, and experimental testing for the model ship CS Enterprise I (Tran, 2014)
- Remote Control and Automatic Path-following for CS Enterprise I and ROV Neptunus (Sandved, 2015)
- Marine Telepresence System (Valle, 2015)
- Nonlinear Adaptive Motion Control and Model-Error Analysis for Ships-Simulations and MCLab experiments (Bjørne, 2016)
- Low-Cost Observer and Path-Following Adaptive Autopilot for Ships (Mykland, 2017)

Other

- YouTube video (Skåtun, 2014)

1.2 Actuators

Figure 1.1 illustrates where the actuators are positioned, and the distances to the Coordinate Origin (CO) are given in Table 1.2. BT and VSP motor speeds are controlled by an Electronic Speed Controls (ESC). The ESCs receive setpoints as Pulse-Width Modulated (PWM) signals from the digital output module of the on-board embedded controller (see Section 1.5.1). Pitch of the VSP blades are controlled by servos attached to a steering rod. These servos also receive their setpoints as PWM signals.

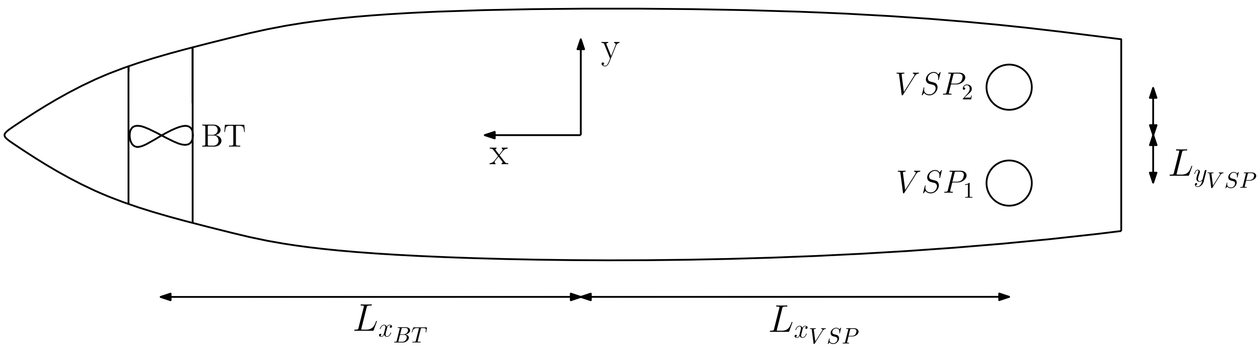


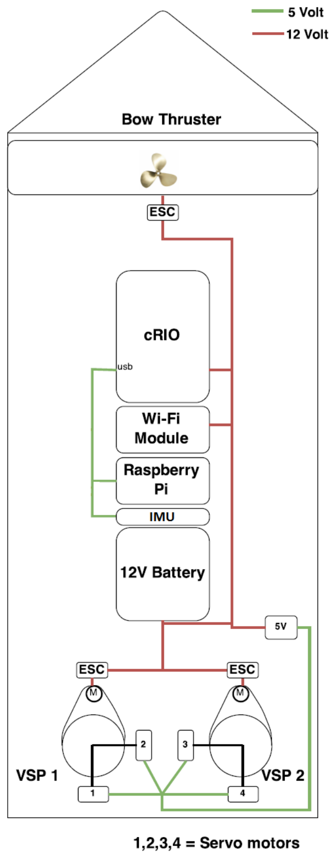
Figure 1.1: Position of actuators. Adapted from Valle (2015)

Table 1.2: Position of actuators

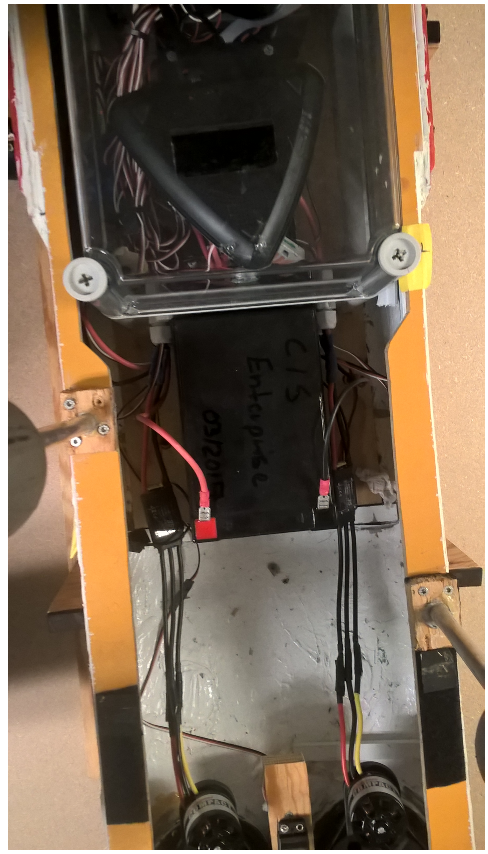
Parameter	**Symbol**	**Value**[m]
x length to VSP	$L_{x,VSP}$	-0.4574
x length to BT	$L_{x,BT}$	0.3875
y length to VSP	$L_{y,VSP}$	0.055

1.3 Power system

CSE1 is powered by a 12V, 12Ah battery on-board. Some of the components require different voltages, and for these components voltage converters are introduced. The setup works as it is, and by connecting the battery to the wires, the whole system will be powered. A schematic of the power grid is displayed in Figure 1.2a. Figure 1.2b shows an installed and connected battery on CSE1.



(a) CSE1 power system



(b) Battery mounted and connected

Figure 1.2: Battery system

1.4 IMU

CSE1 is equipped with an Inertial Measurement Unit (IMU) from Analog Devices. The sensor mounted on-board is the ADIS16364 and includes a triaxis gyroscope and triaxis accelerometer. The sensor has built-in compensation for bias, alignment and sensitivity, and provides accurate measurements over temperatures from -10 to +70 degrees Celsius. The sampling rate is set to 100 Hz. Relevant specifications are presented in Table 1.3, and for supplementary information the reader is referred to the data sheet Analog Devices (2017).

The coordinate frame of the sensor is illustrated in Figure 1.3a, with positive directions marked by arrows. Linear accelerations are defined in a left-hand coordinate frame, while the angular rates are defined according to a right-hand orientation. It is advised to use the right-hand definition, which is achieved by multiplying the linear accelerations with a factor of -1. Further, the

IMU is not aligned with the body frame, as can be seen in Figure 1.3b. According to the *zyx*-convention, the sensor frame has an orientation relative to the body frame of $(\phi, \theta, \psi) = (\pi, 0, 0)$. Hence, by using the corresponding rotation matrix, the measured accelerations and angular rates can be rotated into the body-frame.

Table 1.3: IMU specifications

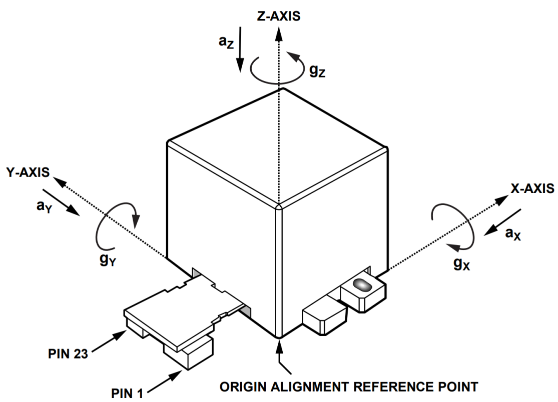
	Parameter	**Typical value**	**Unit**
Gyroscopes	Dynamic range	±350	°/sec
	Sensitivity	0.0125	°/sec/LSB
	Bias stability, σ	0.007	°/sec
	Angular random walk	2.0	$°/\sqrt{hr}$
	Output noise	0.8	°/sec rms
Accelerometers	Dynamic range	±5.25	g
	Sensitivity	1.00	mg/LSB
	Bias stability, σ	0.1	mg
	Velocity random walk	0.12	$m/sec/\sqrt{hr}$
	Output noise	5	mg rms
Power supply	Operating voltage	5.0 ± 0.25	V

1.5 Control system

The on-board control system consists of:

- a National Instruments compact reconfigurable input/output (cRIO) embedded controller
- a Raspberry Pi (RPi) single-board computer
- three electronic speed controls (ESC)
- four servos

A short description of the cRIO, RPi and ESCs is given in the following sections.



(a) IMU reference frame from manufacturer

(b) IMU mounted in the vessel

Figure 1.3: Inertial Measurement Unit in CSE1

1.5.1 cRIO

The Input/Output (I/O) controller on-board CSE1 is the cRIO-9024. It is connected to four Field-Programmable Gate Arrays (FPGA) modules for analogue and digital I/O:

- NI-9215: Used for analog input such as measuring voltage
- NI-9263: Used for reading IMU measurements
- NI-9401: Not used
- NI-9474: Used for sending PWM signals

1.5.2 RPi

The RPi computer provides communication with the Sixaxis controller (see Section 1.6). It works as an embedded system, and once powered it will start searching for the wireless controller. When connection is established, it continuously sends the Sixaxis controller output to the cRIO over Ethernet. To successfully connect the Sixaxis controller to the RPi, wait for the Bluetooth dongle to start blinking before pressing the PS-button on the controller.

If there are problems establishing connection between Sixaxis and RPi, contact Torgeir Wahl or see the MCLab Handbook on Github.

1.5.3 ESC

The ESCs are controlled with PWM tick signals. Table 1.4 gives the set-up for the ESCs on-board CSE1, and PWM signal ranges for each ESC are given in Table 1.5.

Table 1.4: PWM specification for ESC

Initial value	**Scaling**	**Offset**	**PWM period** [Ticks]
0	100	0	800.000

Table 1.5: PWM ranges for ESC

	ESC_BT[%]	**ESC_VSP1**[%]	**ESC_VSP2**[%]
min	7.00	5.28	5.28
neutral	7.55	-	-
max	8.10	6.91	6.91

1.6 High-level communication

Figure 1.4 presents a communication diagram of hardware involved in controlling CSE1. Following the figure from left to right:

Sixaxis transmits its Joystick information to the RPi over Bluetooth communication.

RPi receives Sixaxis data through the USB dongle and forwards it using Ethernet connection (TCP).

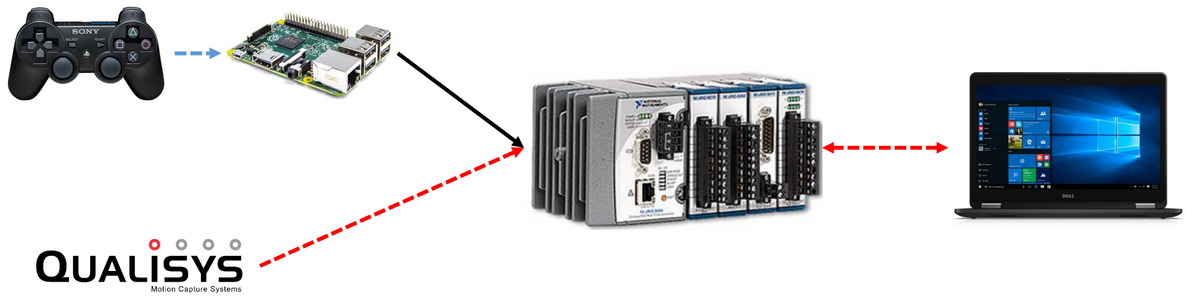


Figure 1.4: CSE1 communication diagram

cRIO reads QTM (see Section 2.2.6) broadcast positioning data through the Wi-Fi bridge on Ethernet port 1 and Sixaxis data on Ethernet port 2. Online data and laptop input is transmitted and received on Ethernet port 1 by the VeriStand Engine.

Laptop reads simulation data and sends input to the cRIO over the MCLab WiFi.

There is a two-way link between the laptop and the I/O controller on-board CSE1. Hence, there are two possible ways for controlling the vessel:

- a laptop connected to the MCLab wireless network
- a Sony Sixaxis wireless gamepad for PlayStation 3

Figure 1.5 gives an overview of the whole communication structure on-board CSE1, from user input to actuator control.

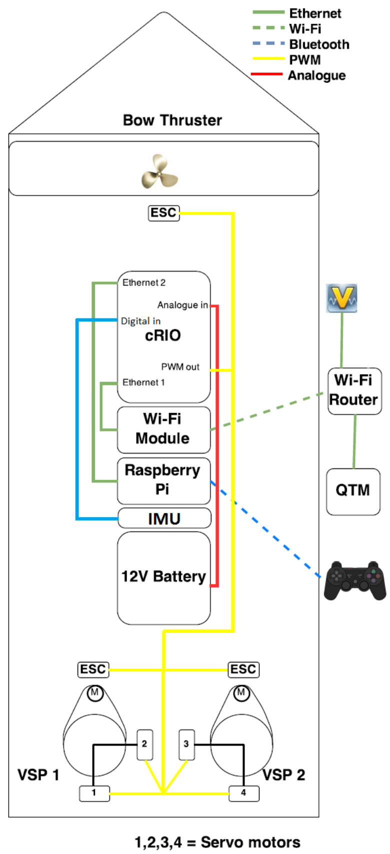


Figure 1.5: CSE1 signal paths

Chapter 2

Software

2.1 Introduction

In order to control CSE1, several software parts have to run. This chapter gives a description of the software hierarchy on CSE1. Note that the software is ready to use, and alterations in the software described here are not necessary (except for modifying `ctrl_custom`, see Part II).

2.2 Control system

Figure 2.1 illustrates the software architecture, and gives an overview of how the different modules are connected as well as I/O from the Simulink models. In general, the software can be divided into 2 groups:

> MATLAB generated parts: `ctrl_custom`, `ctrl_DP`, `ctrl_sixaxis2thruster`, `u2pwm`
>
> LabVIEW generated parts: IMU, Oqus, WL_Joystick, FPGA

All of these modules are described in the following.

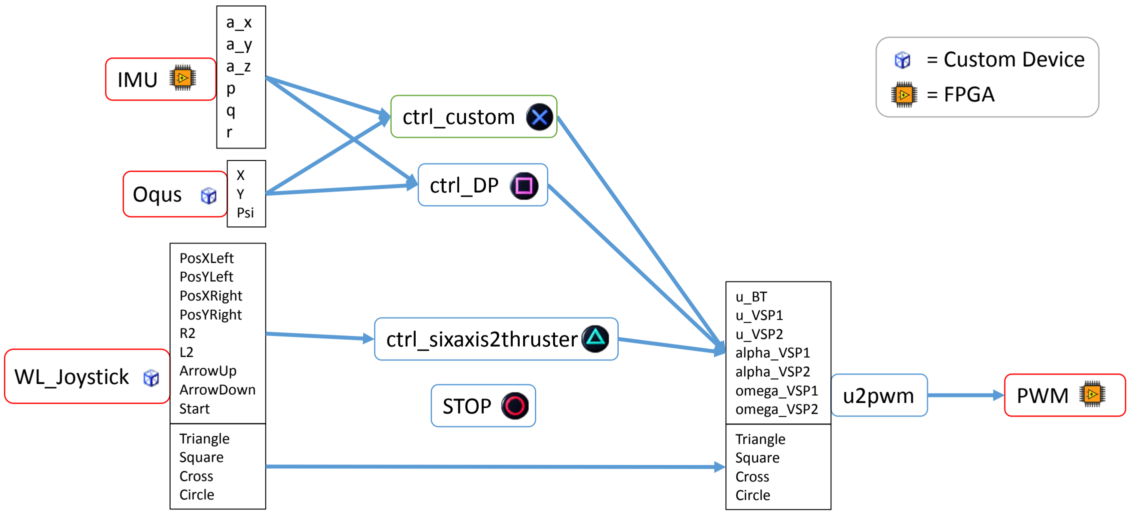


Figure 2.1: CSE1 control software

2.2.1 ctrl_custom

This is the only reconfigurable software, and does not consist of any control system. In Part II a description on how to configure and upload the code to CSE1 is given.

2.2.2 ctrl_DP

This code is provided as a black-box DP system, intended for demonstrations of the vessel. The desired position of the vessel is modified in VeriStand.

2.2.3 ctrl_sixaxis2thruster

All three thrusters can be controlled manually using the Sixaxis controller. The right joystick controls the starboard VSP, the left joystick controls the port VSP and R2/L2 control the BT. The thrust limits are set with ArrowUp and ArrowDown, while Start is a reset button.

Voith Schneider Propellers

The left and right joysticks, respectively, are used to control the VSP deflections, u_{VSP1} and u_{VSP2}, and angles, α_{VSP1} and α_{VSP2}. The joystick coordinates axes, PosX and PosY, point to right and

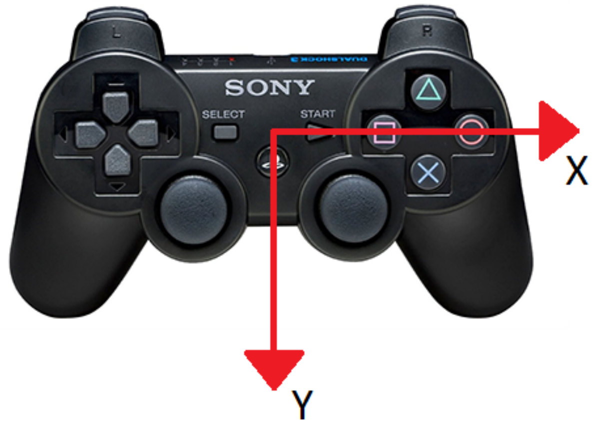


Figure 2.2: Sixaxis coordinate system

downwards, as seen in Figure 2.2. Deflections are given by

$$u_{\mathrm{VSP}i} = \min\left(\sqrt{(\mathrm{PosX})^2 + (\mathrm{PosY})^2}, 1\right),$$

where $\min(\cdot)$ ensures constraining $u_{\mathrm{VSP}i} \in [0,1]$. Angles are defined as

$$\alpha_{\mathrm{VSP}i} = \arctan 2\,(\mathrm{PosX}, -\mathrm{PosY})\,.$$

VSP rotational speeds, ω_{VSP1} and ω_{VSP2}, can be increased or decreased in ± 0.1 increments by use of the directional pad up and down buttons.

Bow thruster

BT is controlled by L2 and R2. Both buttons output -1 when released and increase to 1 when fully pushed. The thruster input

$$u_{\mathrm{BT}} = -\frac{L2 - R1}{2}$$

maps to the interval $u_{\mathrm{BT}} \in [-1,1]$ with positive direction towards starboard.

2.2.4 ctrl_sixaxis2direction

This final control mode is a basin fixed manual control of the vessel. The reference coordinate frame is defined with origin in the command center, positive x-direction towards the basin and positive y-direction towards the large towing tank. The position of the vessel is controlled with the right joystick and yaw is controlled with R2/L2. ArrowUp and ArrowDown set the thruster limits.

2.2.5 u2pwm

This script transforms control input to PWM signals that are sent to the FPGA module. There are 2 types of input: The control signal and a switch signal. The latter is used to switch between the 4 control modes described in Section 2.2.1–2.2.4. Switching is achieved by pressing either one of the four symbols (△,□,×,○). Mapping between buttons and modes is shown in Figure 2.1. This code is not supposed to be altered, and should work as it is. The input signal is subject to the four operations shown in Figure 2.3. All four operations are explained in the following.

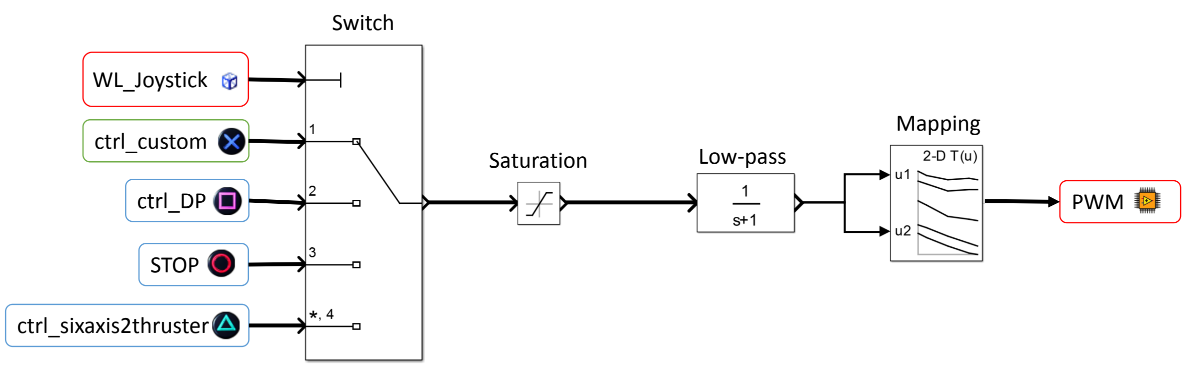


Figure 2.3: Operations in u2pwm

Switch

The switch forwards input from the desired mode, as given by the switch signal:

- `ctrl_sixaxis2thruster` when △ is pushed
- `ctrl_DP` when □ is pushed
- `ctrl_custom` when × is pushed
- STOP when ○ is pushed

Table 2.1: Control input ranges

min	control input	max
$-1 \leq$	u_BT	≤ 1
$0 \leq$	u_VSP1	≤ 1
$0 \leq$	u_VSP2	≤ 1
$-\pi \leq$	alpha_VSP1	$\leq \pi$
$-\pi \leq$	alpha_VSP2	$\leq \pi$
$0 \leq$	omega_VSP1	≤ 0.4
$0 \leq$	omega_VSP2	≤ 0.4

Saturation

Input is saturated as specified in Table 2.1.

Low-pass

The *Low-pass* block provides the option to simulate a mechanical system. Modeling the system as a mechanical system is initialized in VeriStand, and must be activated by the user (not activated by default). The time constant is equal for all parameters, and is set to 1.

Mapping

This block converts controller inputs to signals suitable for PWM output to the ESC. The position of the VSP steering rods are controlled by a pair of servos each. There is a nonlinear relation between the input and the PWM signal, and thus a mapping is utilized. The constants given in Table 7.1 are manually tuned. Thus, if the thrusters are not operating as desired, it might be necessary to retune the servos. The tuning process is described in Chapter 7.

2.2.6 Oqus

Qualisys Track Manager (QTM) software broadcast position data of the vessel over the MCLab WiFi. Reading of data is done on the cRIO through a Custom Device module named Oqus. Once the Oqus software receives data, the position and orientation information is forwarded to the modes as presented in Figure 2.1. The Custom Device is programmed by Torgeir Wahl, and can be found on GitHub.

2.2.7 WL_Joystick

Similar to Oqus, WL_Joystick is a Custom Device that runs on the cRIO. It listens to Ethernet port 2 for input from the RPi. When receiving data, it is forwarded to the respective modes as given in Figure 2.1. The software is designed by Torgeir Wahl, and can be found on GitHub.

2.2.8 FPGA

For the CSE1, three Field-Programmable Gate Arrays (FPGA) modules are in use: One for analog signals, another for digital signals and a third for reading IMU data. The FPGA software is described in the MCLab Software Handbook, and provides a guide on how to create an FPGA module. The modules can be found on GitHub, but are as standard. The PWM signals are related as given in Table 2.2.

Table 2.2: PWM connections

u2pwm	FPGA
pwm_{BT}	pwm0
pwm_{VSP1}	pwm1
pwm_{VSP2}	pwm2
pwm_{servo1}	pwm4
pwm_{servo2}	pwm5
pwm_{servo3}	pwm6
pwm_{servo4}	pwm7

2.3 Connecting software

The different software parts described in Section 2.2 are connected together in VeriStand. The version used is NI VeriStand 2017. Necessary mappings of variables are done in the system definition file `CSE1.nivssdf`. In addition, this is where the different Custom Devices, FPGA code and Simulink Models can be included. However, the standard set-up should not be altered, as all necessary code and mappings are already taken care of. For a description on how to implement the modified Simulink model `ctrl_custom`, the reader is referred to Part II.

Chapter 3

Modeling

This chapter is dedicated to the mathematical model of CSE1. The model is valid for low-speed applications and is based on system identification done by former master students. The proposed control design model is

$$
\begin{aligned}
\dot{\eta} &= R(\psi)\nu && (3.1) \\
M\dot{\nu} &= -C(\nu)\nu - D(\nu)\nu + \tau && (3.2)
\end{aligned}
$$

where

- the pose and velocity vectors are defined as

$$\eta = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \in \mathbb{R}^3, \text{ and} \qquad \nu = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \in \mathbb{R}^3,$$

  respectively. Positions (x, y) and heading ψ are given in the basin frame, whereas the lin-

Table 3.1: CSEI forces and moments given $\omega_{\mathrm{VSP}_1} = 0.199$, $\omega_{\mathrm{VSP}_2} = 0.235$

	Max
Surge X	0.91 N
Positive sway Y	1.24 N
Negative sway Y	1.18 N
Positive yaw N	0.52 Nm
Negative yaw N	0.55 Nm

ear velocities in surge and sway (u, v) as well as the yaw rate r are given in the CSE1 vessel frame.

- the thrust force and moment vector is

$$\tau = \begin{bmatrix} X \\ Y \\ N \end{bmatrix} \in \mathbb{R}^3,$$

where (X, Y) denote the surge and sway forces, and N is the yaw moment. The thrust forces and moment maxima are listed in Table 3.1.

- the three Degrees Of Freedom (DOF) rotation matrix is given by

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

- the vessel inertia matrix is

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & m x_g - Y_{\dot{r}} \\ 0 & m x_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} = M^\top > 0.$$

- the Coriolis and centripetal matrix is

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

- the damping matrix is

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

where the damping components are

$$d_{11}(u) = -X_u - X_{|u|u}|u| - X_{uuu}u^2 \tag{3.3}$$

$$d_{22}(v,r) = -Y_v - Y_{|v|v}|v| - Y_{vvv}v^2 - Y_{|r|v}|r| \tag{3.4}$$

$$d_{23}(v,r) = -Y_r - Y_{|v|r}|v| - Y_{|r|r}|r| - Y_{rrr}r^2 \tag{3.5}$$

$$d_{32}(v,r) = -N_v - N_{|v|v}|v| - N_{vvv}v^2 - N_{|r|v}|r| \tag{3.6}$$

$$d_{33}(v,r) = -N_r - N_{|v|r}|v| - N_{|r|r}|r| - N_{rrr}r^2 \tag{3.7}$$

Rigid body inertia and hydrodynamic added mass parameters are given in Table **??**. Table **??** lists the hydrodynamic damping parameters.

Table 3.2: CSE1 rigid body and added mass parameters

Rigid body		Added mass	
Parameter	Value	Parameter	Value
m	14.11	$X_{\dot{u}}$	−2
I_z	1.76	$Y_{\dot{v}}$	−10
x_g	0.0375	$Y_{\dot{r}}$	0
y_g	0.0	$N_{\dot{r}}$	−1

Table 3.3: CSE1 damping parameters

Hydro surge		Hydro sway		Hydro yaw	
Parameter	Value	Parameter	Value	Parameter	Value
X_u	-0.6555	Y_v	-1.33	N_v	0.0
X_{uu}	0.3545	Y_{vv}	-2.776	N_{vv}	-0.2088
X_{uuu}	-3.787	Y_{vvv}	-64.91	N_{vvv}	0.0
X_v	0.0	Y_r	-7.25	N_r	-1.9
X_{vv}	-2.443	Y_{rr}	-3.45	N_{rr}	-0.75
X_{vvv}	0.0	Y_{rrr}	0.0	N_{rrr}	0.0
.	.	Y_{rv}	-0.805	N_{rv}	0.130
.	.	Y_{vr}	-0.845	N_{vr}	0.080

Part II

User Manual

It is assumed that the reader has studied the MCLab Handbook before using CSE1, and has knowledge about lab equipment, procedures and safety precautions. In addition, keep the following in mind when using CSE1:

Water damage: CSE1 is not waterproof and has excessive thrust capability which can inflict large roll angles. The risk of water on deck is reduced through thrust limitations and HIL testing before application of new control algorithms.

Propeller dry running: The BT must only be run in water. Before removing the vessel from the water, the control system must be stopped and the VeriStand project undeployed.

Loss of laptop control: Wireless network instability may result in loss of connection between the laptop user interface and the cRIO. In this event, fall back to manual thruster control by pushing △ on the Sixaxis controller. Alternatively, press ○ to stop the vessel.

Total loss of control: Pull CSE1 with a boat hook, and keep the vessel in water while disconnecting batteries.

Chapter 4

Preparations

4.1 Launching the vessel

The gear in the bow thruster is lubricated with water, and thus it is **important** that the vessel is always launched in the basin when starting up/deploying code. Prior to deploying code, do the following steps:

1. Place the vessel in the basin.
2. Check that the 1kg weight in the bow is properly placed in front of the cRIO box.
3. Place the 12V, 12Ah battery (marked CSE1) in its dedicated position. Connect red/positive first, then the black/negative wire.
4. Once the Bluetooth dongle (connected to the RPi) starts blinking (blue light with frequency of about 1Hz), press the PS-button on the Sixaxis controller. When successfully connected, indicator 1 on the controller will light red without blinking.

Following the above procedure, CSE1 will be ready for carrying out experiments. However, before continuing, make sure the vessel does not take in water (there have been some issues with leakage in the hull opening around the bow thruster). NB! Change battery if the battery level gets below 11.5V (battery status is displayed in the VeriStand interface).

4.2 Positioning system

Qualisys Track Mangager (QTM) is a camera-based system for broadcasting position data of the vessel over WiFi. Three cameras, mounted on a towing carriage spanning the basin, perform optical measurements of the vessel position and orientation (x,y,z,ϕ,θ,ψ). At least two cameras must cover the area where the vessel is located in order to obtain its position. Hence, position measurements cannot be provided if the vessel is closer than 6 meters to the carriage. QTM is set up according to the following procedure:

1. Turn on the computer labeled "Mclabb-1010" which is connected to the screen "QTM SURFACE", and open *Qualisys Track Manager*. Choose the desired project.

2. In the upper left corner, press ("New") to start a new measurement.

3. Click on to open the settings. A window similar to the one displayed in Figure 4.1 will pop up. Then, in the left pane, navigate to "6DOF Tracking". Remove existing bodies and press "Aqcuire Body" (verify that CSE1 has 0° heading before pressing). QTM should now identify the 4 markers on the vessel.

4. Navigate to the 3D visualization window, and verify that the body is defined correctly (body frame position and orientation relative the 4 markers). If extra markers/points are found, remove these. Press "Translate", and define CO from the highest marker (i.e. the one with lowest z-coordinate, typically -150mm) such that it has the body coordinates (x,y,z)=(550,0,-500)[mm]. For further explanation and debugging instructions see the MCLab Handbook.

4.3 Updating custom control system

See the MCLab Handbook for instructions on how to update and include your customized control system in the VeriStand project. Once you have completed the steps described there, follow the steps in Section 4.4.

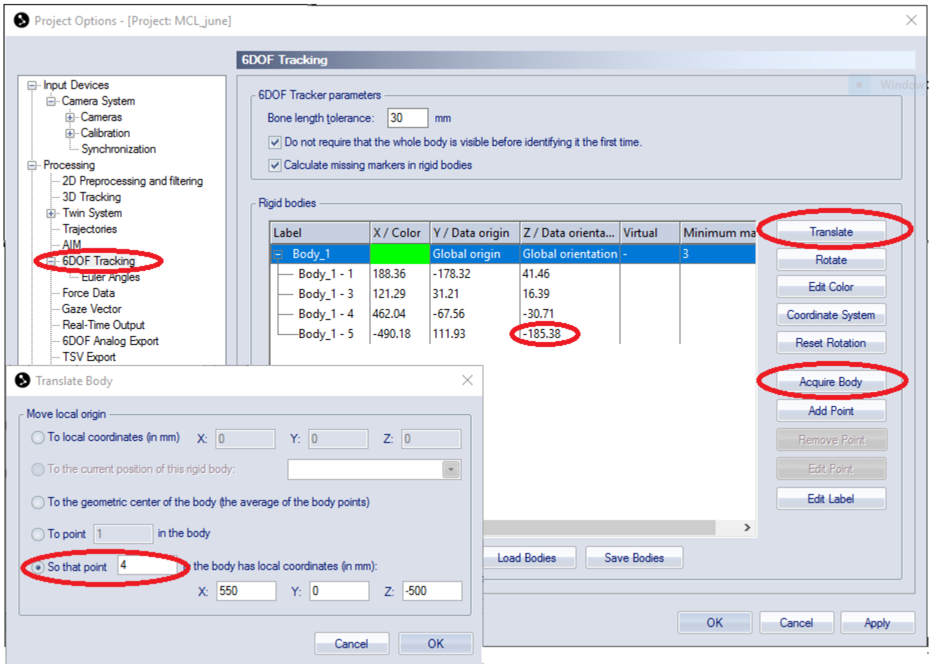


Figure 4.1: QTM window

4.4 Upload VeriStand project to the vessel

With the hardware set up and ready, continue with preparing the software:

1. Make sure the computer is connected to the MCLab network (either by Ethernet cable or over WiFi).

2. Check the communication between the laptop and CSE1. Open the command prompt (`cmd.exe`) and write the following: `ping 192.168.0.75`. Make sure the command returns 0% loss.

3. Open the VeriStand project (`CSE1.nivsproj`). Press the deploy button (see Figure 4.2) or F6. The project is now being uploaded to the cRIO on-board CSE1. If deployment is not successful, make sure the Sixaxis controller is still connected to the RPi, and that the vessel body is shown in QTM. Deploy again. If it still does not work, try to restart the vessel (either by reconnecting the battery or restarting the cRIO in NI MAX). Also check that the QTM computer is connected to MCLab (`ping 192.168.0.10`).

4. When successfully deployed, open the Workspace (`CSE1.nivsscreen`). The code is now running on the cRIO, continue to Chapter 5 for instructions on how to operate the vessel.

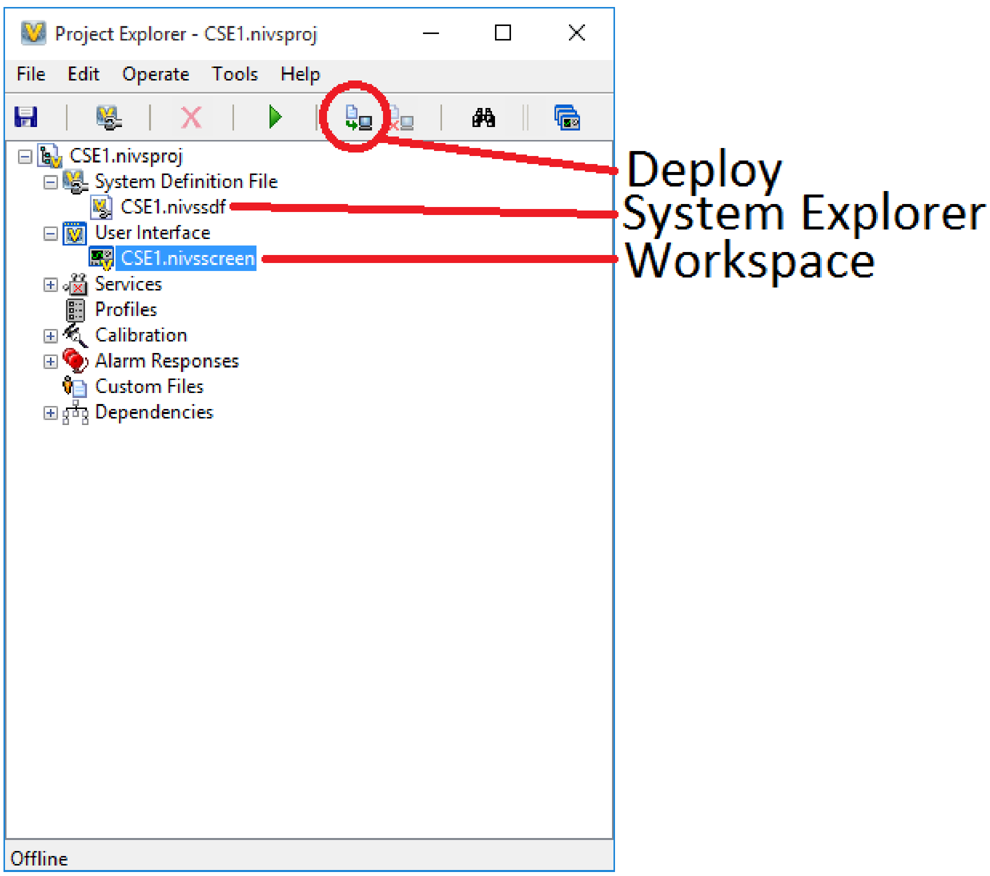


Figure 4.2: User interface in VeriStand Project Explorer

Chapter 5

Operation

After successfully deploying a VeriStand project, you can control the vessel with the Sixaxis controller and/or the laptop. Use the Sixaxis controller to switch between the different operation modes presented in Section 2.2.1–2.2.4:

- △ for `ctrl_sixaxis2thruster`
- □ for `ctrl_custom`
- × for `ctrl_DP`
- ○ to STOP

Data logging can be done in two ways, as described in the MCLab Handbook.

5.1 Workspace

On the laptop, use the Workspace in VeriStand to monitor and alter desired variables and parameters. There are 3 screens, one for each operating mode:

ctrl_sixaxis2thruster Figure 5.1 is a screenshot of the Workspace for `sixaxis2thruster`. As the control mode is based on input from the Sixaxis controller, the Workspace is used for moni-

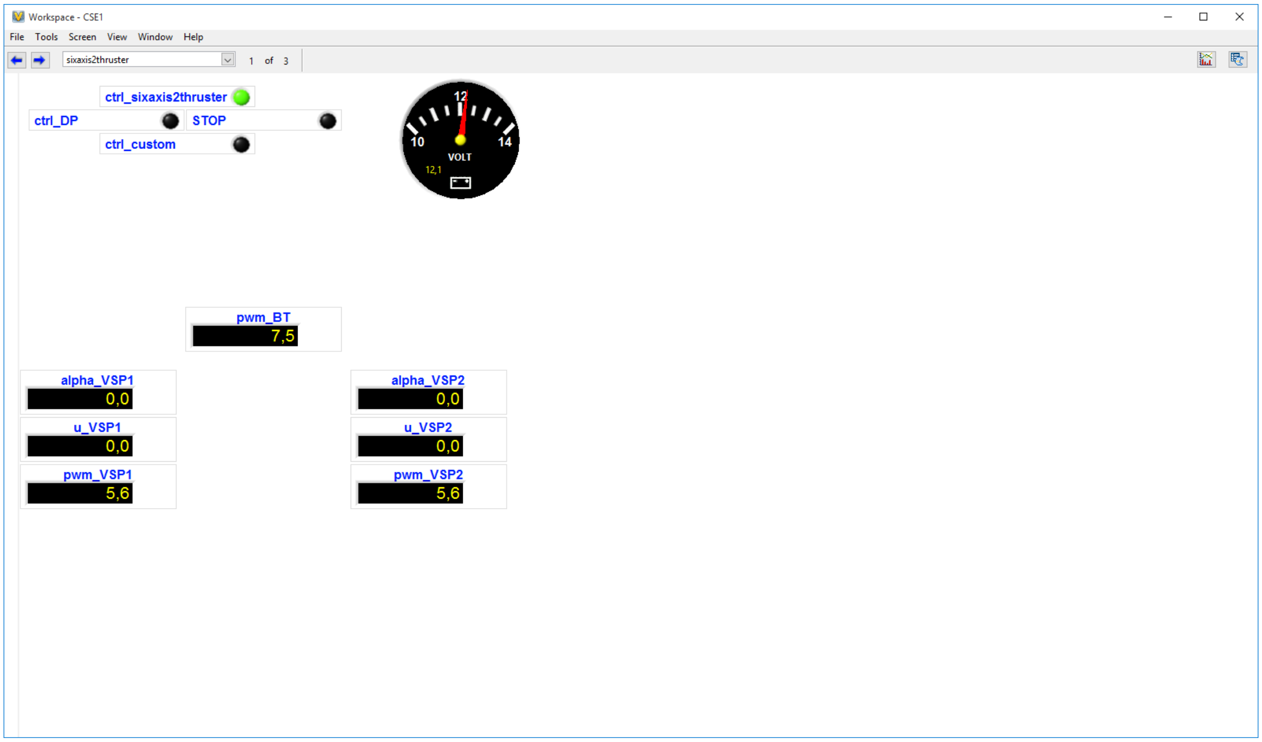


Figure 5.1: sixaxis2thruster workspace

toring only.

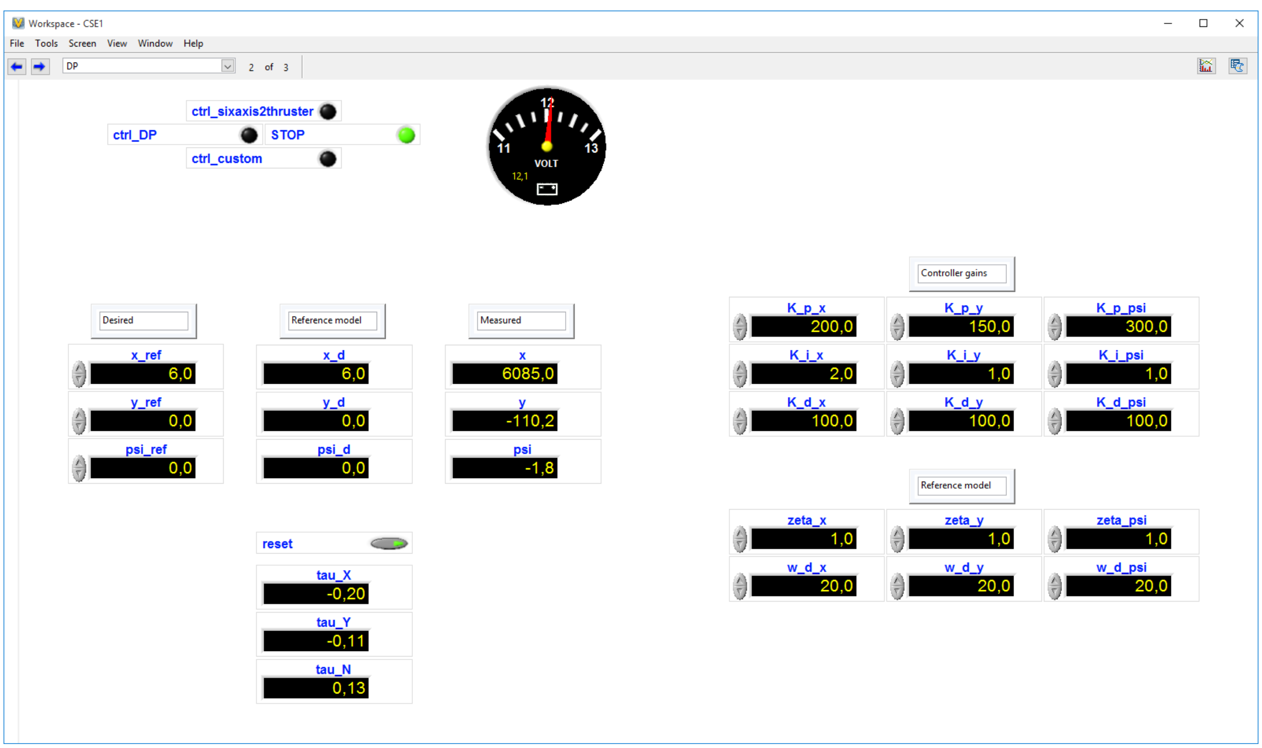


Figure 5.2: DP workspace

ctrl_DP Figure 5.2 shows the Workspace for the DP mode. Here, you define setpoints for the reference model, and it's current position and orientation are displayed. On the right-hand side, the model can be tuned. The initial values give a functional DP-system.

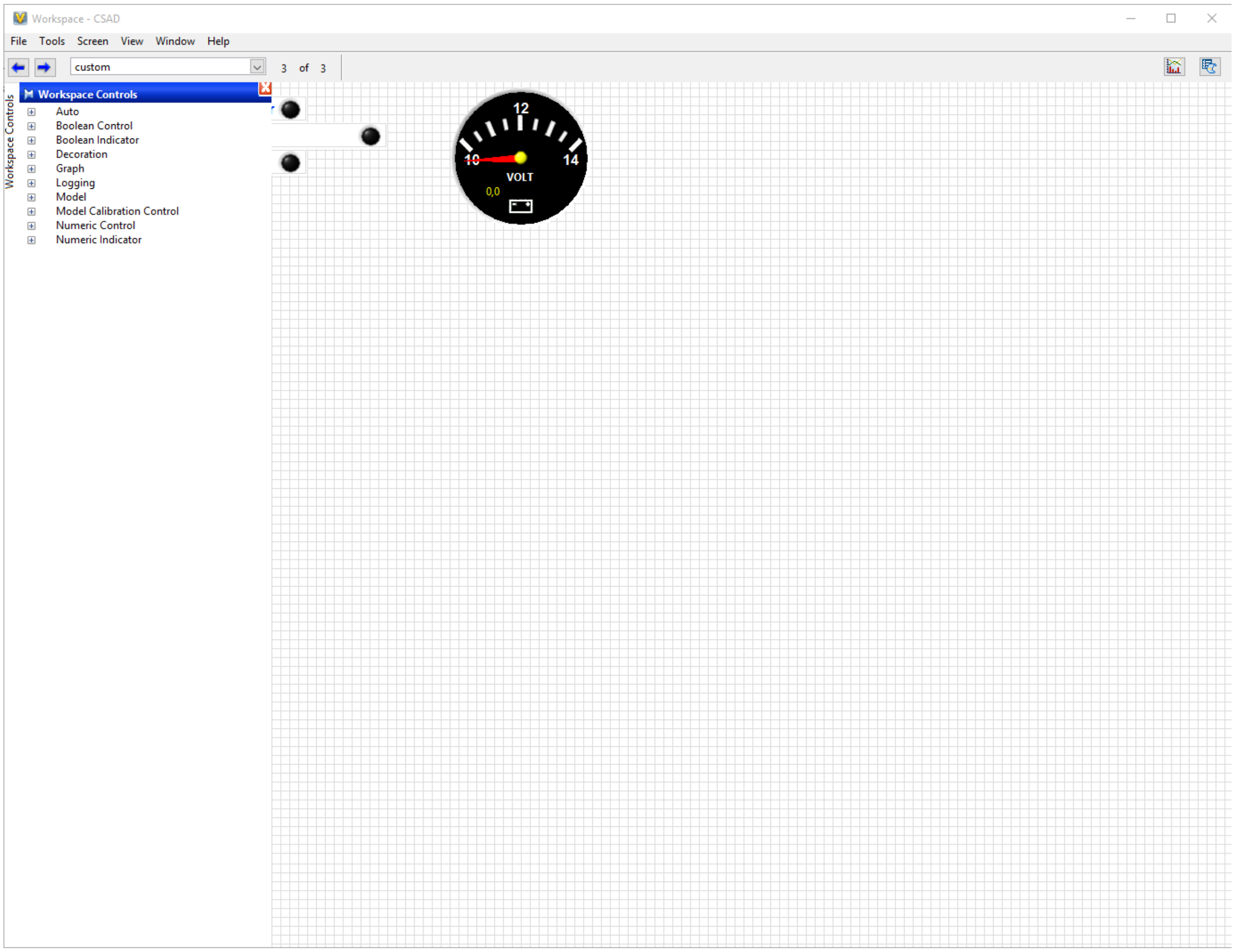


Figure 5.3: custom workspace

ctrl_custom The Workspace for the customized control mode is depicted in Figure 5.3. This mode should be used when testing your own control design. If you need more screens, press Screen and Add Screen. To edit one screen, press Screen and Edit Mode. On the left side (Workspace Controls), you can add different controls or indicators to monitor variables and parameters in the simulation model. Press to browse for the parameter you want to control/-monitor, see Figure 5.4. For example, real-time tuning of controller gains can be done by adding a Numeric Control, selecting Meter and linking it to the desired variable. Use the scale option to enable higher precision when tuning.

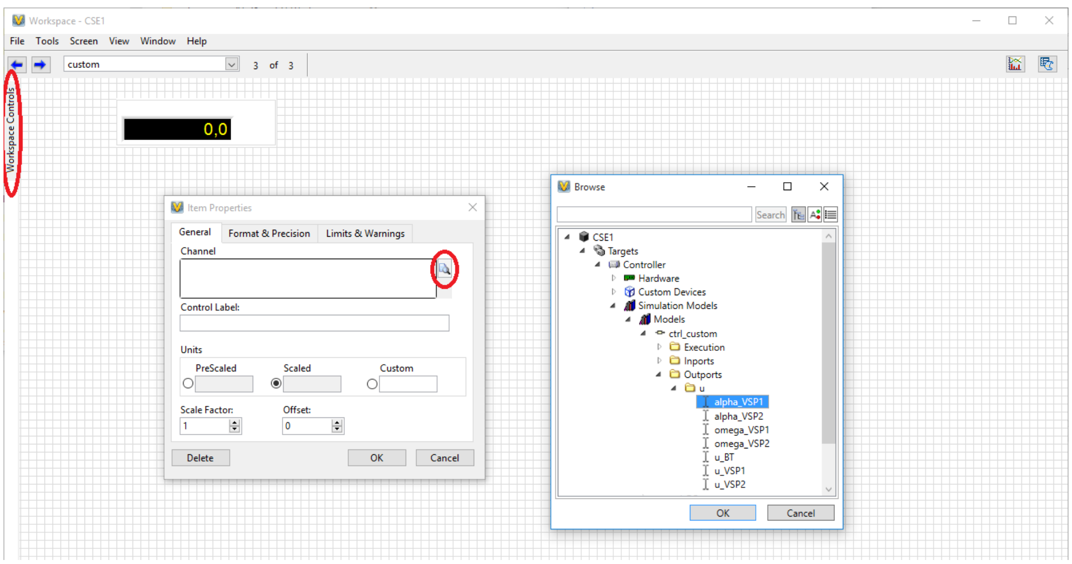


Figure 5.4: Workspace window in Edit Mode

Chapter 6

Demolition

When the experiments are finished, follow the procedure given here to shut down:

1. Switch to `ctrl_sixaxis2thruster` in the Workspace, and steer CSE1 towards the basin wall.
2. In the Project Explorer window, undeploy the code.
3. Disconnect the battery (negative first, then positive).
4. Remove the battery, and set it to charge in the storage.
5. Lift CSE1 up from the basin, and put it in its rack.
6. Leave the Sixaxis controller in the vessel.
7. On the QTM computer, quit the QTM program.
8. If you recorded any videos with the Camera System, export these videos to a memory stick, quit the software and turn off the TV-monitor.
9. Leave the lab clean and tidy, and bring all your personal belongings with you.

Chapter 7

Tuning of the servos

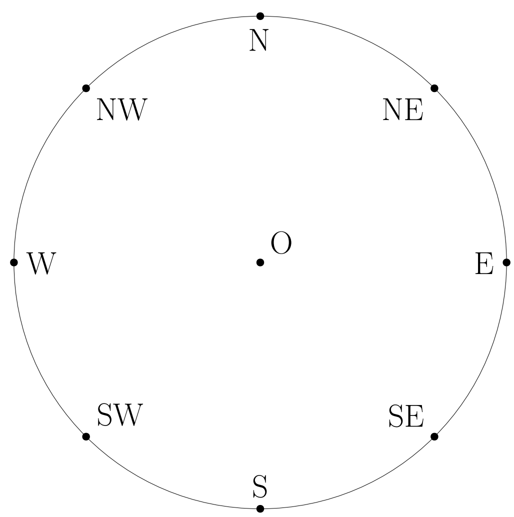


Figure 7.1: Tuning of servo and rod position

Tuning the servos controlling the steering rods of the two VSPs can be divided into two tasks: First, the extreme value positions of the rod depicted in Figure 7.1 have to be determined. These can be estimated on dry land with the vessel safely placed in its rack. The other task is to determine the origin (O in Figure 7.1). This corresponds to the rod positions that will keep the vessel at rest while the VSPs are running. A method of how to carry out the servo tuning is presented below. The PWM mapping is given in Section 2.2.8, Table 2.2.

Finding the origin and servo extreme values The origin, or the neutral position, where the vessel stays at rest with both VSPs running as well as servo extreme values can be determined by test and trial in the following way:

1. In VeriStand, disconnect all mappings from the simulation model `u2pwm` to the FPGA, such that thrusters and servos do not run. See Figure 7.2.

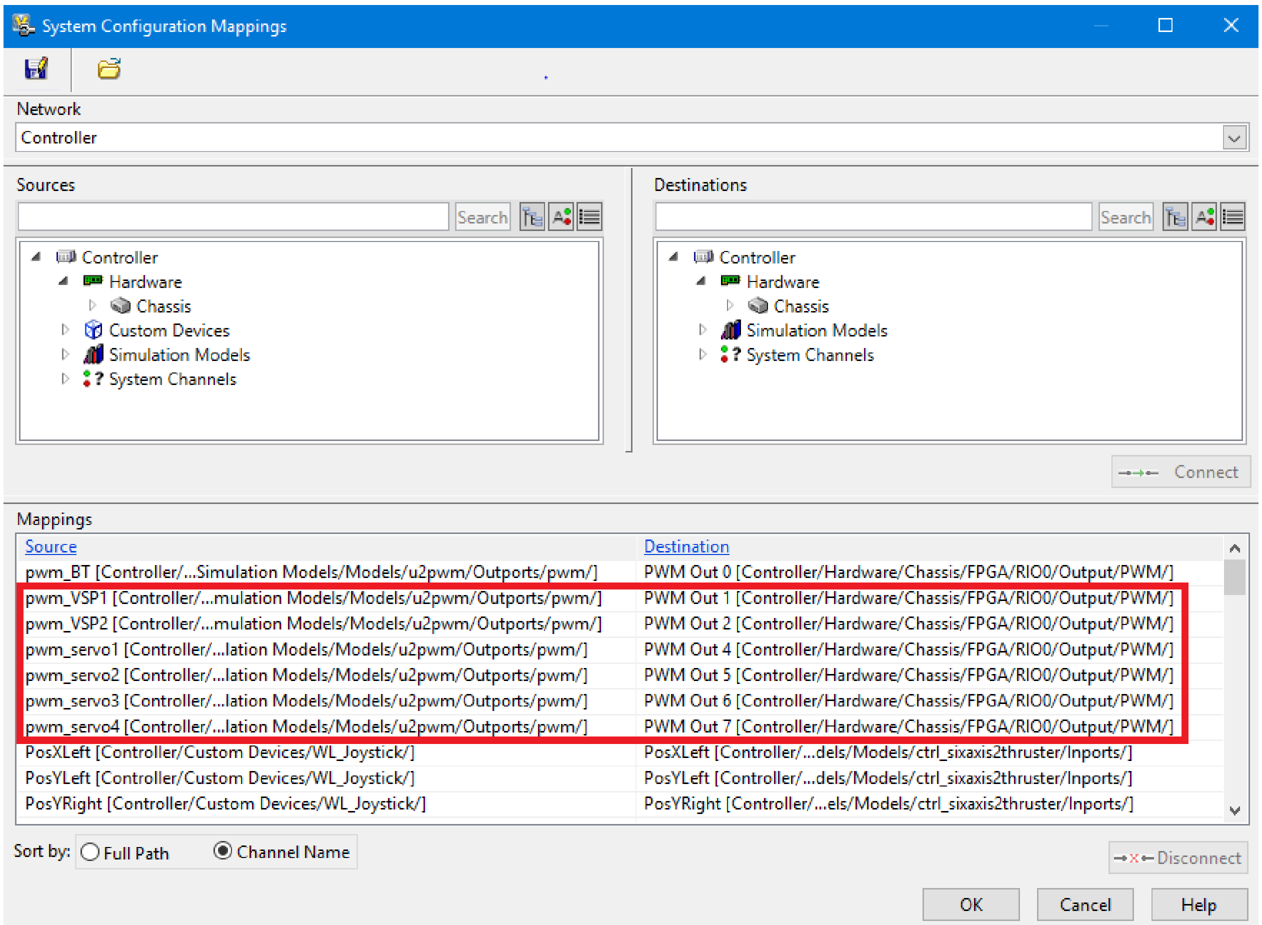


Figure 7.2: Mapping between `u2pwm` and `FPGA`.

2. In the Workspace (`CSE1.nivsscreen`), place 4 new numerical controls linked to the pwm signal for the steering rod servos (`pwm4-pwm7`). Set up further 2 numerical controls and connect them to the FPGA PWM signals controlling the rotation rate of the VSPs (`pwm1` and `pwm2`). Set the scale of all six controls to 100 for higher precision when tuning (e.g. a numerical control value of 520 equals PWM=5.20). A Veristand project file for step 1 and 2, named `servo_tuning`, can be found on https://github.com/NTNU-MCS/CS_EnterpriseI_

archive/tree/master/Servo%20tuning%20and%20force%20measurements.

3. On dry land, estimate the origin by eye. This will help in finding the order of magnitude of where to expect the origin to be when launching CSE1 in step 7.

4. Deploy the project with all 4 servos set in neutral position in the numerical controls.

5. To start the VSPs, find the PWM value that activates them. A sound can be heard when the VSPs are activated. This value was 520 as of January 2020. Now, start increasing the PWM values to find the minimum and maximum rotation speeds of the VSPs. The interval at which the VSPs are running is [528, 691] as of January 2020. Thrust force measurements revealed that the VSPs gave approximately the same force output with `pwm1=554` and `pwm2=560`. These pwm are obtained by giving 0.199 and 0.235 as omega inputs.

6. Start seeking all 8 extreme values manually by eye, and write down the servo gains for all extreme values. Keep in mind that the directions for the two VSPs are opposite (e.g. North corresponds to the rod pointing in negative y-direction for VSP1 and positive y-direction in body-frame for VSP2, see Figure 7.3).

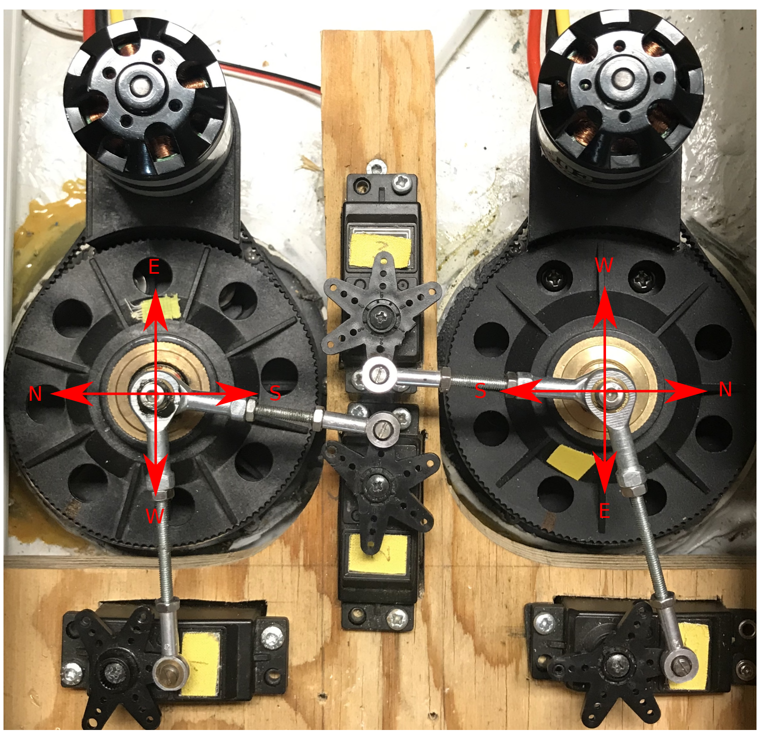


Figure 7.3: North, East, South and West directions for the servos.

7. Launch CSE1 in the basin, and deploy the project. In the Workspace, set the servos to their

estimated origin positions. Then, set `pwm1` and `pwm2` to the activation value and run the motors by increasing `pwm1` and `pwm2` to a desired level (e.g. those proposed in step 5).

8. Once both VSPs are running, start tuning the servos to find the position where the vessel stays at rest. It might be easier to fine-tune the origin of one VSP at a time, and thereafter test both together. The tuning may take some time, but it should be possible to get a position where the vessel does not move. Continue with fine-tuning the extreme values for North and South. The vessel is supposed to move in positive surge direction when both servos are set to their respective North-directions and negative surge direction with the servos in their respective South-directions. East and West positions can also be verified by looking at surge motion, which should be zero. Keep in mind that small changes on the servos can give relatively large impacts on the vessel motions, and the tuning may not at all be intuitive but rather a process of trial-and-error.

When all 8 extreme values and the origin is found, update the values in `u2pwm_init.m`. The mapping function should work as it is. Open `u2pwm.slx`, build the model with new servo mapping and update the simulation model in VeriStand. It is **very** important to build a new version of `u2pwm.slx` after updating `u2pwm_init.m`, else the updated values will not be used by `u2pwm.slx`. Make sure that the updated versions of `u2pwm_init.m` and `u2pwm.slx` are distributed to the users of CSE1, and uploaded to GitHub. Also make sure that Table 7.1 in this document is up to date.

Table 7.1: Measurements and data for VSPs as of January 2020

Position	VSP_1		VSP_2	
	servo 1 [%]	servo 2 [%]	servo 3 [%]	servo 4 [%]
N	4.45	7.20	5.75	4.54
NE	5.30	7.15	5.80	4.10
E	5.70	6.85	4.72	3.88
SE	5.25	5.45	4.30	4.30
S	4.38	5.20	3.90	5.47
SW	3.48	5.33	4.05	6.05
W	3.20	6.05	5.20	6.30
NW	3.58	6.87	5.60	5.70
O	4.50	5.95	4.62	5.12

Chapter 8

Force measurements

Before using CSEI for experimentation it might be useful to calibrate the control input to force mapping, as this can change over time. The following general guideline is suggested to obtain a new control input to force mapping for CSEI:

1. Attach 3 force sensors to the vessel: One at the bow and two on port side. Force measurements will be sent to the catman® data aquisition software.
2. Constrain the vessel by using strings and springs such that it cannot move significantly.
3. For each thruster individually, increase the control input step-wise while measuring the generated force.
4. The VSPs may also be tested simultaneously to see if there are any interacting forces between the two.
5. Plot the recorded data and extract an appropriate input to force mapping.

This procedure was proposed by the PhD candidate Einar S. Ueland. Details on the procedure are explained in section 8.1.

8.1 Control Input to Thrust Mapping

8.1.1 Setup

In order to measure the thruster generated forces, the vessel has to be restrained from moving in any direction. For this reason, attach a total of six strings to the vessel, as displayed in Figure 8.2. Four of these, two on port side and the other two on starboard side, are fastened perpendicular to the basin walls with clamps. The string going out of the bow can be fastened to the front wall of the basin and the string in the aft to a plank mounted on the towing carriage. Two force sensors are connected to the vessel port side and one in the bow, hence enabling force measurements in both surge and sway directions. The strings on starboard side and in the aft are linked to the vessel through pre-tensioned springs. These provide elasticity to the system to obtain accurate measurements. A sketch of the ship and the constraints, as well as the sensors and springs, can be seen in Figure 8.1. The actual setup in the MCLab is shown in Figure 8.2 and Figure 8.3. Rather than connecting the force sensors directly to the vessel, it might be a good idea to attach them close to the basin walls instead, as they are sensitive to water.

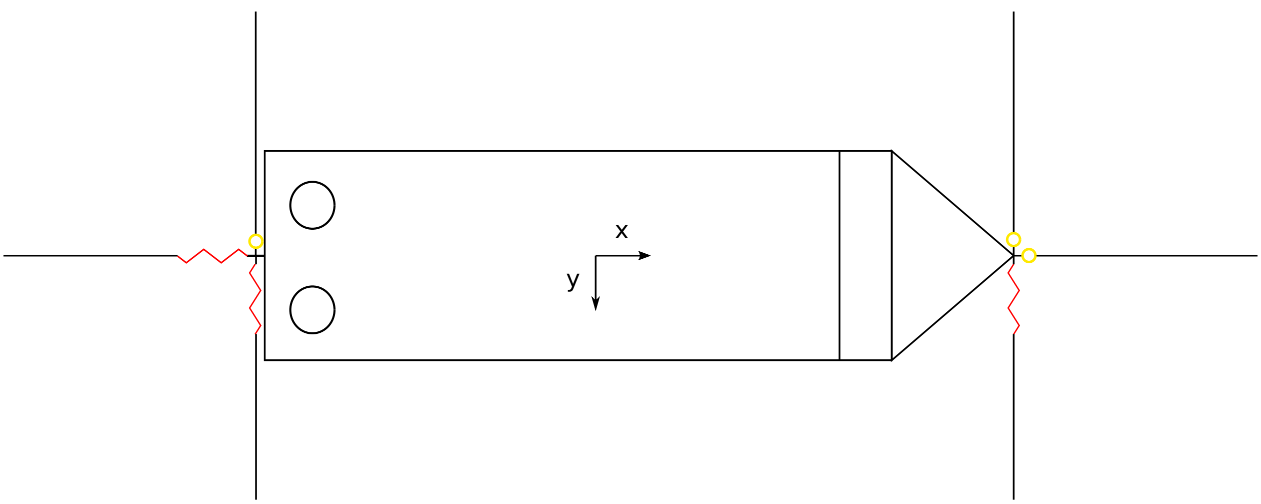


Figure 8.1: Setup with sensors and constraints

8.1.2 Step Program

A modified file of `ctrl_custom.slx` has been made for gradually increasing the control input applied to one thruster at a time. It can be found in the folder named `Thruster steps 2020.zip` on GitHub, https://github.com/NTNU-MCS/CS_EnterpriseI_archive/tree/master/

Figure 8.2: Setup with sensors and constraints

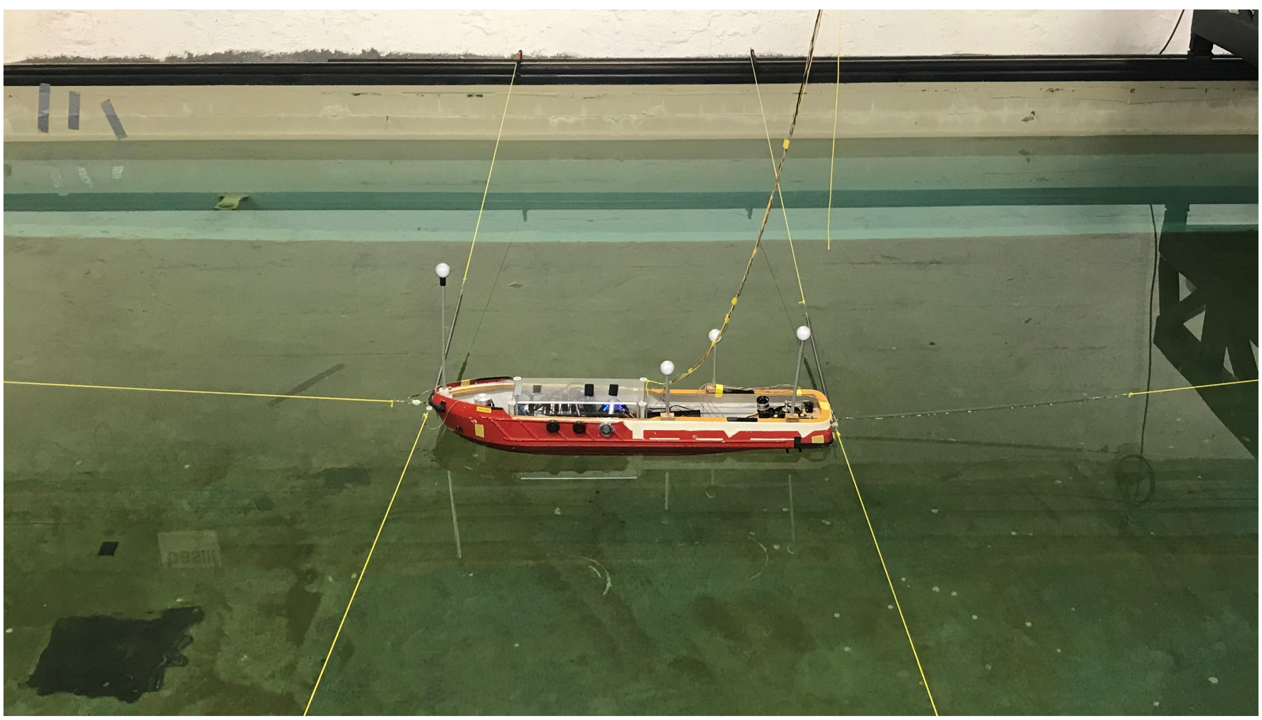


Figure 8.3: Close up of setup with sensors and constraints

Servo%20tuning%20and%20force%20measurements. The program consists of several successive steps of increasing magnitude, from minimum to maximum control input in steps of 10%. Table 8.1 gives the bounds on the control input.

Table 8.1: Upper and lower bounds on input commands

Control input	Interval
u_{BT}	$[-1, 1]$
u_{VSP_1}	$[0, 1]$
u_{VSP_2}	$[0, 1]$

The bow thruster should be tested in both directions. The angles of attack of the two VSPs must be adjusted according to the desired test direction. For instance, to obtain thrust in positive surge direction, set α_{VSP_1} and α_{VSP_2} to 0.

8.1.3 Force Measurements and catman

The generated thrust forces are measured by sensors attached to the vessel starboard side and bow. Measurements are sent to the catman® data aquisition software. catman allows for visualization of the measured force while testing, and stores the measured time series in dedicated `.bin`-files. The MATLAB file `catman_read_dt.m`, from https://github.com/NTNU-MCS/CS_EnterpriseI_archive/tree/master/Servo%20tuning%20and%20force%20measurements, can be used to open and read the `.bin`-file as a struct in MATLAB. Transform the time series into MATLAB arrays and link these with the known control inputs. By evaluating this data, a proper input command to thrust force mapping can be established. An example with data and measurements as of January 2020 is given in the following section.

8.1.4 Tables of Data and Measurements as of January 2020

The following tables present the sensor measurements obtained from applying the step program outlined in subsection 8.1.2. Average values of the measured forces in surge X and sway Y corresponding to a commanded step input, together with the start time of when each step was initiated t relative to the first step are presented. Table 8.2 and Table 8.3 give the values obtained for the VSP$_1$ and VSP$_2$, respectively, whereas Table 8.4 and Table 8.5 presents the same data for the bow thruster BT for positive and negative sway force.

Table 8.2: Measurements and data for VSP_1

Input, u	Force [N]	t [s]
0.1	0.0153	0
0.2	0.0128	20
0.3	0.0233	40
0.4	0.0468	60
0.5	0.1250	80
0.6	0.1947	100
0.7	0.2572	120
0.8	0.3313	140
0.9	0.4034	160
1.0	0.4771	180

Table 8.3: Measurements and data for VSP_2

Input, u	Force [N]	t [s]
0.1	0.0048	0
0.2	0.0060	20
0.3	0.0043	40
0.4	0.0512	60
0.5	0.1092	80
0.6	0.1843	100
0.7	0.2616	120
0.8	0.3414	140
0.9	0.4233	160
1.0	0.4328	180

Table 8.4: Measurements and data for BT in positive sway direction

Input, u	Force [N]	t [s]
0.1	0.0224	0
0.2	0.0459	20
0.3	0.0777	40
0.4	0.1021	60
0.5	0.1277	80
0.6	0.1390	100
0.7	0.1797	120
0.8	0.2297	140
0.9	0.2762	160
1.0	0.3326	180

Table 8.5: Measurements and data for BT in negative sway direction

Input, u	Force [N]	t [s]
-0.1	-0.0476	0
-0.2	-0.0699	20
-0.3	-0.0864	40
-0.4	-0.1137	60
-0.5	-0.1406	80
-0.6	-0.1762	100
-0.7	-0.1892	120
-0.8	-0.2075	140
-0.9	-0.2280	160
-1.0	-0.2677	180

Figure 8.4 and Figure 8.5 display the command inputs and corresponding force measurements presented in Tables 8.2–8.5.

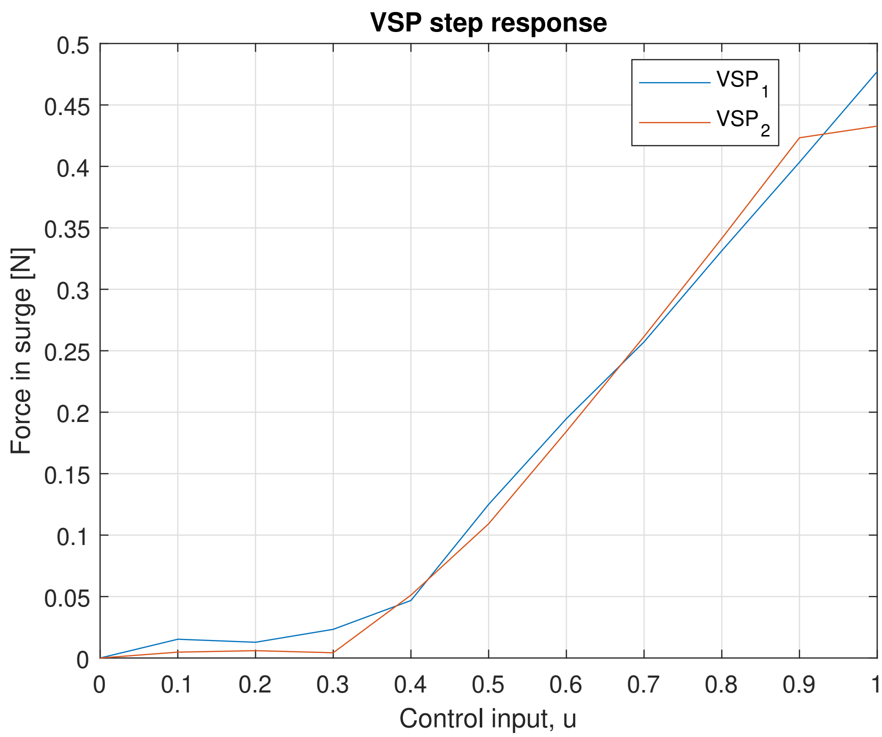


Figure 8.4: VSP mapping

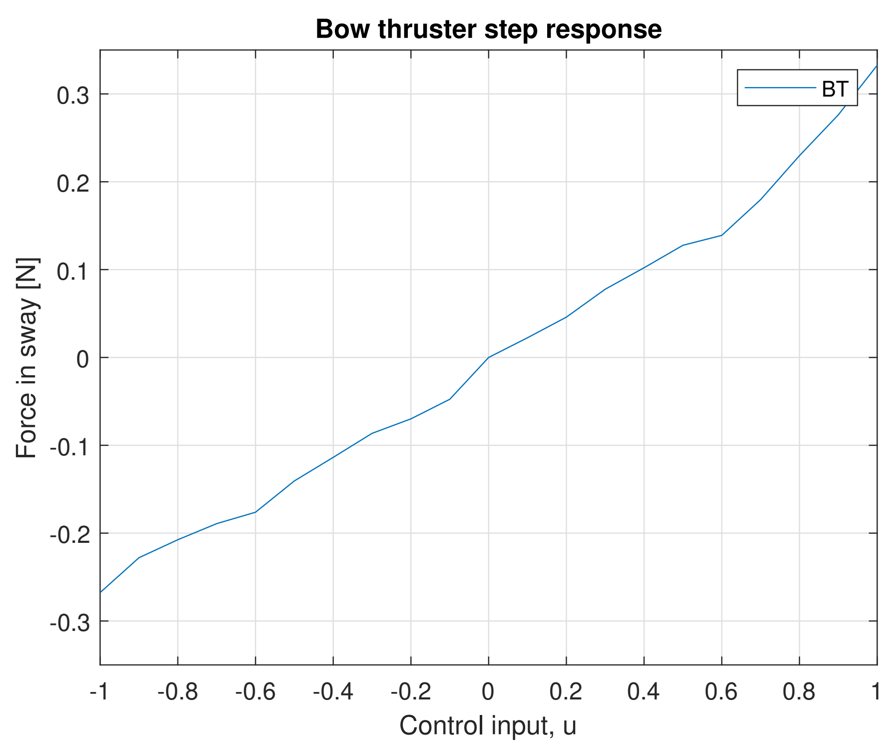


Figure 8.5: BT mapping

Bibliography

Analog Devices, 2017. ADIS16364 Data Sheet. Technical Report. URL: http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16364.pdf.

Bjørne, E.S., 2016. Nonlinear Adaptive Motion Control and Model-Error Analysis for Ships-Simulations and MCLab experiments. Master's thesis. NTNU.

Mykland, A., 2017. Low-Cost Observer and Path-Following Adaptive Autopilot for Ships. Master's thesis. NTNU.

Orsten, A., 2014. Automatic Reliability-based Control of Iceberg Towing in Open Waters. Master's thesis. Norwegian University of Science and Technology.

Orsten, A., Norgren, P., Skjetne, R., 2014. LOS guidance for towing an iceberg along a straight-line path, in: Proceedings of the 22nd IAHR International Symposium on ICE 2014 (IAHR-ICE 2014).

Sandved, F., 2015. Remote Control and Automatic Path-following for C/S Enterprise I and ROV Neptunus. Master's thesis. Norwegian University of Science and Technology.

Skåtun, H.N., 2011. Development of a DP system for CS Enterprise I with Voith Schneider thrusters. Master's thesis. Norwegian University of Science and Technology.

Skåtun, H.N., 2014. CS Enterprise I. Video. Http://www.youtube.com/watch?v=MiESJsIZO04.

Tran, N.D., 2013. Development of a modularized control architecture for CS Enterprise I for path-following based on LOS and maneuvering theory. Specialization project.

Tran, N.D., 2014. Line-Of-Sight-based maneuvering control design, implementation, and experimental testing for the model ship C/S Enterprise I. Master's thesis. Norwegian University of Science and Technology.

Valle, E., 2015. Marine Telepresence System. Master's thesis. Norwegian University of Science and Technology.