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 ease the process of using CyberShip Enterprise I(CSE1), and concerns only software and hardware of CSE1 specifically. 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.

Structure of document

This User Manual is divided in two parts:

- Technical description(hardware, software, mathematical models etc.)
- Operation manual(software deployment, launching/demolition instructions etc.)

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 is not correct.

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

Preface i

Structure of document i

Further work i

CSE1 main data i

I Technical description 2

1 Hardware 3

1.1 Introduction to CSE1 3

1.1.1 Literature 3

1.2 Actuators 4

1.3 Power system 5

1.4 IMU 6

1.5 Control system 7

1.5.1 cRIO 7

1.5.2 RPi 8

1.5.3 ESC 9

1.6 High-level communication 9

2 Software 12

2.1 Introduction 12

2.2 Control system 12

2.2.1 ctrl_custom 12

2.2.2 ctrl_DP 13

2.2.3 ctrl_sixaxis2thruster 13
2.2.4 ctrl_sixaxis2direction 14
2.2.5 u2pwm 14
2.2.6 Oqus 18
2.2.7 WL_Joystick 18
2.2.8 FPGA 18
2.3 Connecting software 19

3 Modeling 20

II User Manual 23

4 Launching 25
4.1 Vessel and lab preparations 25
4.2 Updating custom control system 27
4.3 Upload VeriStand project to the vessel 27

5 Operation 29
5.1 Workspace 29

6 Demolition 33

Bibliography 34

Part I

Technical description

Chapter 1

Hardware

1.1 Introduction to CSE1

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

The vessel is a 1:50 scale model of a tug boat, and is fitted with two Voith Schneider propellers(VSP) astern and one bow thruster(BT). The main dimensions of the vessel are:

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 much 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 C/S Enterprise I.(Tran, 2014)
- Remote Control and Automatic Path-following for C/S 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 illustrate the position of the actuators, and their distance from CO is given in Table 1.2. The BT and VSP motor speeds are controlled by an Electronic Speed Control(ESC). The ESC receive their setpoints as pulse-width modulated (PWM) signals from the cRIO digital output module. The VSP blade pitches are controlled by servos. The servos also receive their setpoint as PWM signals.

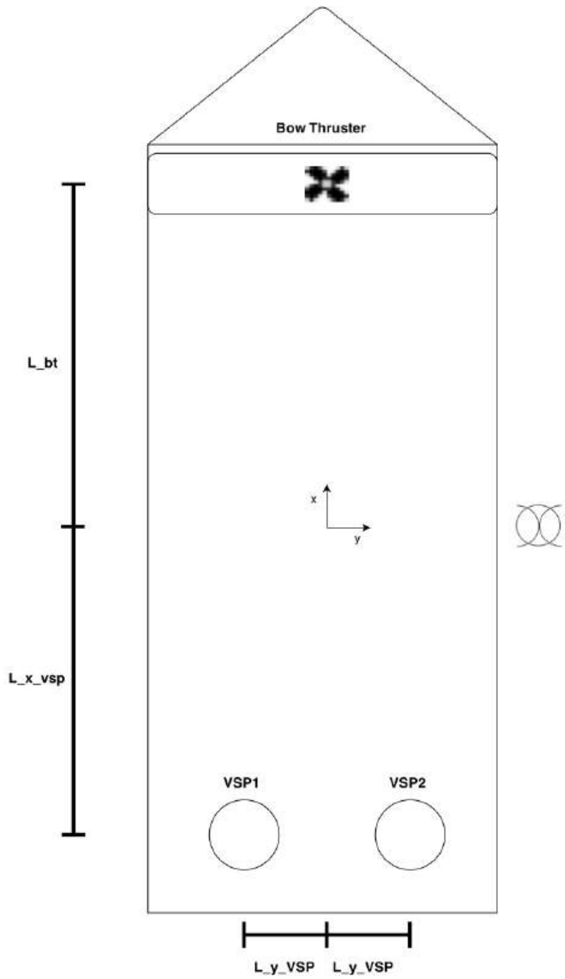


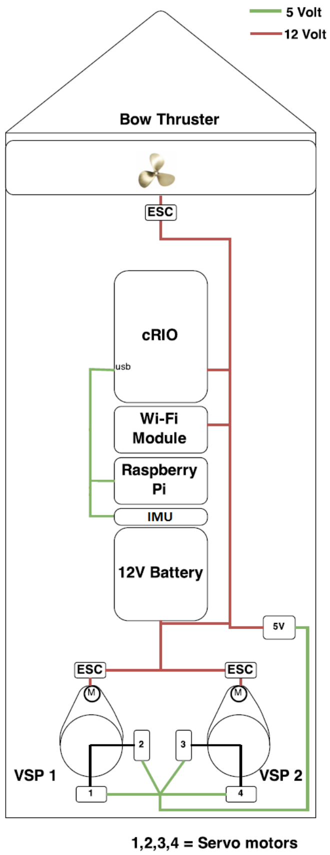
Figure 1.1: Position of actuators. Adapted from Valle (2015)

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

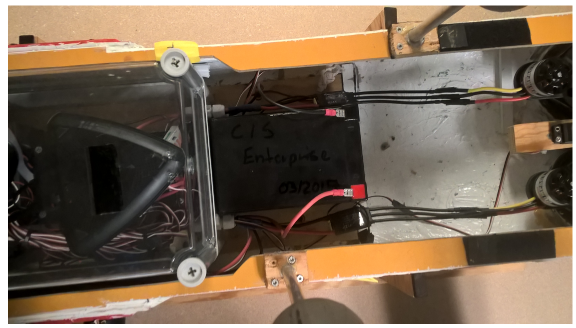
Table 1.2: Position of actuators

1.3 Power system

CSE1 is powered with one 12V 12Ah battery on-board. Some of the components require different voltage, and thus some voltage converters are mounted. However, the setup works as it is, and by connecting the battery to the wires, the whole system is powered. A schematic of the power grid is illustrated in Figure 1.2a, and Figure 1.2b show a photo of the battery mounted and connected.



(a) CSE1 power system



(b) Battery mounted and connected

Figure 1.2: Battery system

1.4 IMU

CSE1 is equipped with one 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 thus provides accurate measurements over a temperature range of -10 to +70 degrees Celsius. The sampling rate is set to 100 Hz. The most relevant data is 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 illustrated by arrows. As seen, the standard coordinate frame for linear accelerations uses left-hand orientation, while the angular rates uses right-hand orientation. It is advised to change the coordinate frame of accelerations to right-hand, which is achieved by multiplying the accelerations with -1. Further, the sensor is mounted with a different orientation than the body frame, as can be seen in Figure 1.3b. Using the *zyx*-convention, the sensor frame has an orientation relative body frame: $(\phi, \theta, \psi) = (\pi, 0, 0)$. Hence, by using the rotation matrix with these values, the measured accelerations and angular rates can be rotated to 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 the following parts:

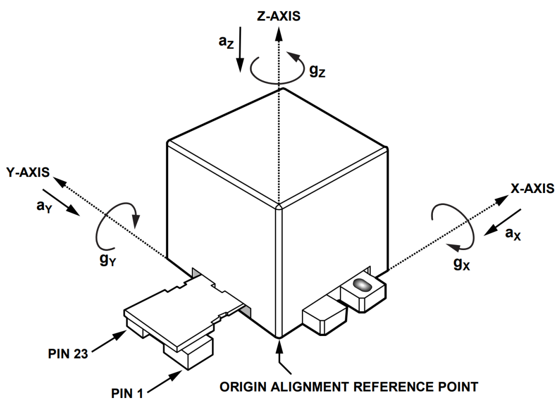
- a National Instruments compact reconfigurable input/output (cRIO) embedded controller
- a Raspberry Pi (RPi) single-board computer
- three electronic speed controllers (ESC)
- four servos

A short description of the cRIO and RPi is given in the latter.

1.5.1 cRIO

The model on-board is the cRIO-9024, and it is connected to 4 FPGA modules for analogue and digital I/O:

- NI-9215, used for analog input such as measuring voltage



(a) IMU reference frame from manufacturer

(b) IMU mounted in the vessel

Figure 1.3: Inertial Measurement Unit in CSE1

- NI-9263, used for reading IMU measurements
- NI-9401, not used
- NI-9474, used for sending pwm signal

1.5.2 RPi

The Raspberry Pi provides communication with the Siaxaxis controller, as described in 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 ESC's are controlled with PWM signals, based on PWM tick signals. Table 1.4 describe the setup for all ESC's on-board CSE1, and Table 1.5 gives the pwm signal range for each ESC.

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	3.12	3.12
neutral	7.55	5.01	5.01
max	8.10	6.90	6.90

1.6 High-level communication

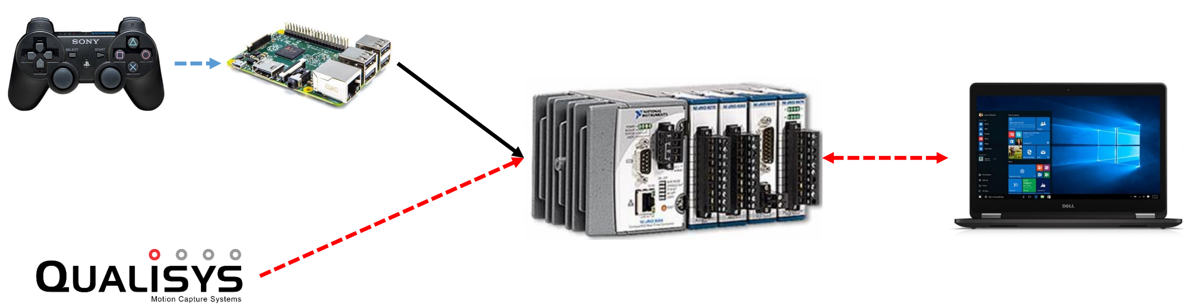


Figure 1.4: CSE1 communication diagram

Following Figure 1.4 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).

cRIO reads QTM broadcast positioning data through the Wi-Fi bridge on Ethernet port 1, 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 MC Lab Wi-Fi.

Hence, there are two possible methods for controlling the vessel:

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

Figure 1.5 depicts an overview of the whole communication structure of CSE1, from user input to actuator control.

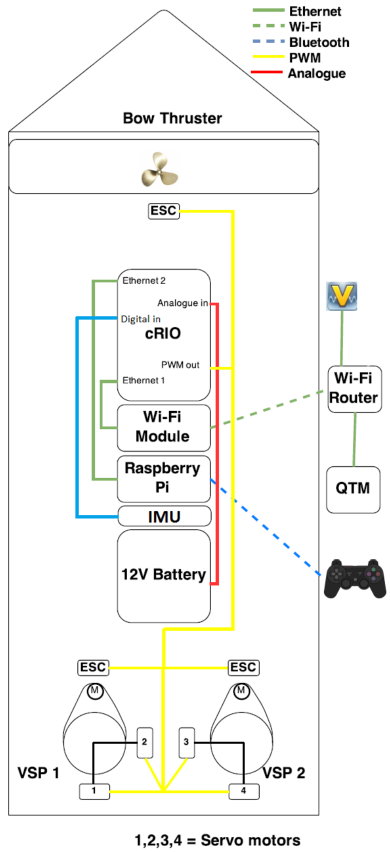


Figure 1.5: CSE1 signal paths

Chapter 2

Software

2.1 Introduction

In order to control CSE1, there are several software parts that runs. 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 is not necessary(except modifying ctrl_custom, as described in Part II).

2.2 Control system

Figure 2.1 illustrate the software architecture, and gives an overview of how the different modules are connected and the 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 and u2pwm

LabVIEW generated parts: IMU, Oqus, WL_Joystick and FPGA

All of these modules are described in the latter.

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.

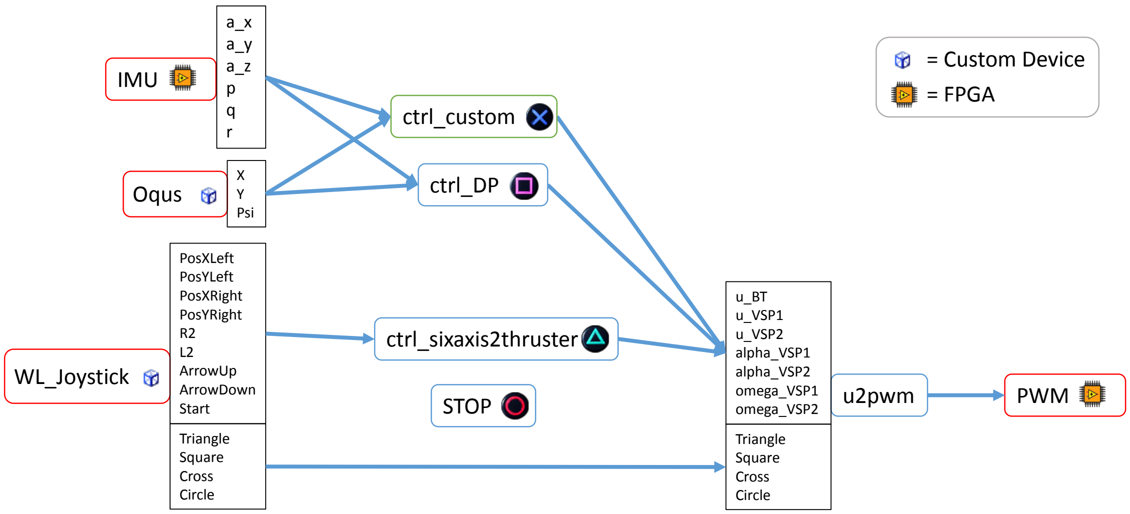


Figure 2.1: CSE1 control software

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

Voith Schneider Propellers

The left and right joysticks, repectively, give the VSP deflections, u_{VSP1} and u_{VSP2}, and angles, α_{VSP1} and α_{VSP2}. The joystick coordinates PosX and PosY axes point right and down, as seen in Figure 2.2. The deflection is

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

The $\min(\cdot)$ ensures constraining $u_{\text{VSP}i} \in [0, 1]$. The angle is

$$\alpha_{\text{VSP}i} = \text{arctan2}\,(\text{PosX}, -\text{PosY})\,.$$

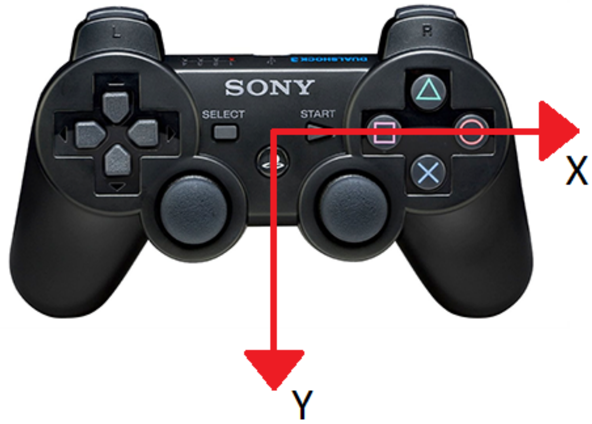


Figure 2.2: Sixaxis coordinate system

The VSP rotational speeds, ω_{VSP1} and ω_{VSP2}, are set 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 increasing to 1 when fully pushed. The thruster input

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

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

2.2.4 ctrl_sixaxis2direction

The 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 sets the thruster limits.

2.2.5 u2pwm

This code transforms the control input to PWM signals, which are sent to the FPGA module. There are 2 groups of inputs, namely the control signal and a switch signal. The switch signal is used to switch between the 4 control systems described in the previous. Switching is simply achieved by pressing either one of the four symbols, and the mapping between the buttons and models is shown in Figure 2.1. The code is not supposed to be altered, and should work as it is. The code performs four operations on the input signal, as illustrated in Figure 2.3. All four operations are explained in the latter.

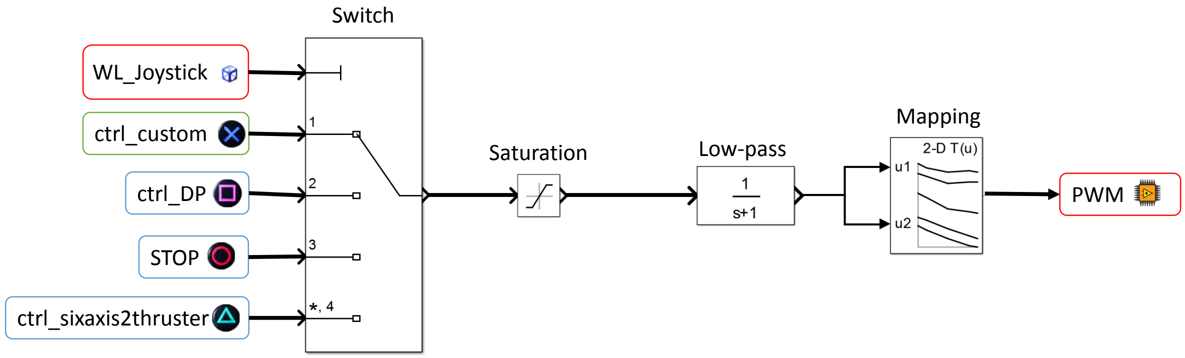


Figure 2.3: Operations in u2pwm

Switch

The switch simply forward the input from the desired model, 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

Saturation

The input is saturated as given in Table 2.1.

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

Low-pass

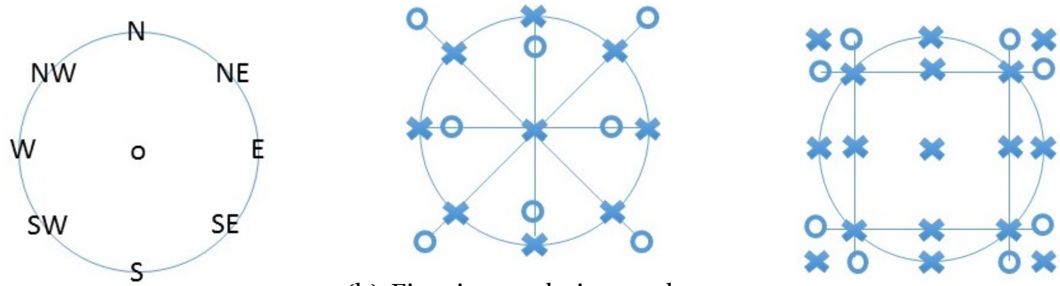
The Low-pass block provides an optional simulation of a mechanical system. Modeling the system as a mechanical system is initialized in VeriStand, and must be activated by the user(default is no mechanical simulation). The time constant is equal for all parameters, and is set to 1.

Mapping

This block converts the 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 for each. There is a nonlinear relation between the input and the PWM signal, and thus a mapping is utilized. The constants found here are manually tuned, and if the thrusters are not operating as desired, it may be necessary with to tune the servos again. The tuning process is described below.

Table 2.2: Servo PWM ranges (tuned July 2017)

Position	VSP1		VSP2	
	servo1 [%]	servo2 [%]	servo3 [%]	servo4 [%]
N	6.80	7.20	6.00	4.90
NE	7.60	7.10	5.90	4.30
E	7.80	6.50	5.00	3.80
SE	7.60	5.60	4.30	4.50
S	6.40	5.10	4.00	5.40
SW	5.50	5.30	4.00	6.20
W	5.20	5.90	4.60	6.50
NW	5.90	6.80	5.50	5.80
Origo	6.59	6.19	4.79	5.01



(a) Measurments

(b) First interpolation and extrapolation

(c) Second extrapolation

Figure 2.4: Servo, rod position tuning

Tuning of servos One need to carry out two processes when tuning the servos, namely finding origin with the vessel floating and determining extrema with the vessel in its rack. The pwm mapping is given in Section 2.2.8. Finding servo extrema:

1. In VeriStand, disconnect all mappings from the simulation model u2pwm to the FPGA pwm, such that thrusters and servos don't run.

2. In the Workspace (CSE1.nivsscreen), make 4 new numerical controls linked to the pwm signal for the servos(pwm4-pwm7). Set the scale to 100 in these controls, for higher precision when tuning.

3. Deploy the project, and set all 4 servos to neutral position in the numerical control. For example, if servo1 has origin in pwm=4.90, then set the value to 490 in the control.

4. Start seeking all 8 extrema manually, and write down the servo gains for all extrema. This is a manual process, and the direction of the rod corresponding to for example North is determined by eye. Keep in mind that the directions for each VSP are opposite. Thus, North corresponds to the rod pointing in negative y-direction for VSP1 and positive y-direction for VSP2(in body-frame).

5. When all extrema are found(N, NE, E,... NW) for the servos, continue to finding origin with the vessel in the basin.

Finding origin(neutral position) is done in the following way:

1. In the Workspace, place 2 new numerical controls and connect them to the FPGA pwm signal for VSPs(pwm1 and pwm2). Set the scale to 100.

2. In the u2pwm_init.m file, find the VSP_zero_pwm and VSP_u2pwm_gain values. Find the pwm value corresponding to omega=0.3 (VSP_zero_pwm+0.3*VSP_u2pwm_gain = 5.57?)

3. Launch CSE1 in the basin, and deploy the project. In the Workspace, set the servos to old neutral position, and then set the pwm signal (557?).

4. Once the VSPs are running, start tuning the servos to find the position when the vessel is at rest with running VSPs. This may take some time, but it should be possible to get a position where the vessel does not move.

When all 8 extrema positions and origin is found, insert the new values in *u2pwm_init.m*. The mapping function should work as it is, and Figure 2.4c illustrate the mapping. Open *u2pwm.slx*, build the model with new servo mapping and update the simulation model in VeriStand. Finally, map the pwm output from u2pwm to FPGA pwm again. Deploy the project and verify the new servo setup.

2.2.6 Oqus

The Qualisys Track Manager software broadcast the position data of the vessel over MC Lab Wifi. Reading these data is done on the cRIO through a Custom Device module named Oqus. The Oqus software simply listen to the network for data from QTM, and once it receives data it forward the position and orientation to the models as given in Figure 2.1. The Custom Device is programmed by Torgeir Wahl, and can be found on GitHub.

2.2.7 WL_Joystick

This is the second Custom Device that runs on the cRIO, which listen to the Ethernet 2 port for input from the RPi. It forwards this data to the respective models 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, 3 FPGA modules are in use(1 for analog signals, 1 for digital signals and 1 for reading IMU data). The FPGA software is described in the MC Lab 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 signal is mapped as described in Table 2.3.

Table 2.3: 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

All the different software parts described in the previous Section are connected together in VeriStand. On CSE1, VeriStand 2017 is used. In the system definition file *CSE1.nivssdf*, all necessary mappings of variables are done. In addition, here the different Custom Devices, FPGA code and Simulink Models can be included. However, the standard setup should not be altered, as all necessary code and mappings is already taken care of. For description on how to implement the modified ctrl_custom Simulink model, the reader is referred to Part II.

Chapter 3

Modeling

The mathematical model of CSE1 is given here, based on system identification done in previous Master's Theses. The model is valid for low-speed. The proposed control design model is

$$\dot{\eta} = R(\psi)\nu \tag{3.1}$$

$$M\dot{\nu} = -C(\nu)\nu - D(\nu)\nu + \tau \tag{3.2}$$

where

- the pose and velocity vectors are

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

  respectively. (x, y) is the position and ψ the yaw angle or heading in the basin frame. (u, v) are the surge and sway velocities in the CSE1 vessel frame, and r is the yaw rate.

- the thrust force and moment vector is

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

	Max
Surge X	1.03 N
Sway Y	2.50 N
Yaw N	0.98 N

Table 3.1: CSEI forces and moments given $\omega_{\text{VSP}} = 0.3$

where (X, Y) is the surge and sway force vector, and N is the yaw moment. The thrust forces and moments ranges are listed in Table 3.1.

- the three degrees of freedom (3 DOF) rotation matrix is

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

- 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

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

- the damping matrix is

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

where the damping components are

$$
\begin{aligned}
d_{11}(u) &= -X_u - X_{|u|u}|u| - X_{uuu}u^2 && (3.3)\\
d_{22}(v,r) &= -Y_v - Y_{|v|v}|v| - Y_{vvv}v^2 - Y_{|r|v}|r| && (3.4)\\
d_{23}(v,r) &= -Y_r - Y_{|v|r}|v| - Y_{|r|r}|r| - Y_{rrr}r^2 && (3.5)\\
d_{32}(v,r) &= -N_v - N_{|v|v}|v| - N_{vvv}v^2 - N_{|r|v}|r| && (3.6)\\
d_{33}(v,r) &= -N_r - N_{|v|r}|v| - N_{|r|r}|r| - N_{rrr}r^2 && (3.7)
\end{aligned}
$$

The rigid body inertia and hydrodynamic added mass parameters are given in Table 3.2, and the hydrodynamic damping parameters are given in Table 3.3.

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 3.2: CSE1 rigid body and added mass 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

Table 3.3: CSE1 damping parameters

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 limitation and HIL testing before application of new control algorithms.

Propeller dry running: 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. 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

Launching

Describe hardware preparations, establishing connection, building and including simulink model, uploading code to vessel

4.1 Vessel and lab preparations

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 the code. Hence, do as follows:

1. Place the vessel in the basin

2. Check that the 1kg weight in the bow is placed at its position, in front of the cRIO box.

3. Place the 12V 12Ah battery(marked CSE1) in its dedicated position. Connect red/positive first, then black/negative.

4. Once the Bluetooth dongle (connected to the RPi) starts blinking(blue light with frequency approx. 1 Hz), press the PS-button on the sixaxis controller. On the controller, indicator 1 lights continuously red when successfully connected.

5. Place the vessel inside the region of sight for Qualisys(check on the computer that the 4 reflectors are visible). Align the vessel with 0 degrees heading in the basin frame, i.e. with the bow pointing towards the command center.

6. On the Qualisys computer(labeled *QTM Surface*), start Qualisys Track Manager. In the upper left corner, press to start new measurement, then press to open the setting. In the left pane, navigate to 6DOF Tracking. Remove existing bodies and then press Aqcuire Body(verify CSE1 has 0°heading when pressing). Qualisys should find the 4 markers on the vessel(check 3D window to verify the body. If extra markers/points are found, remove them from the body in settings). 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]. See Figure 4.1 for illustration, or see the MCLab Handbook for further explanation and debugging.

7. Go to 3D visualization window, and verify that the body is defined correct (body frame position and orientation relative the 4 markers).

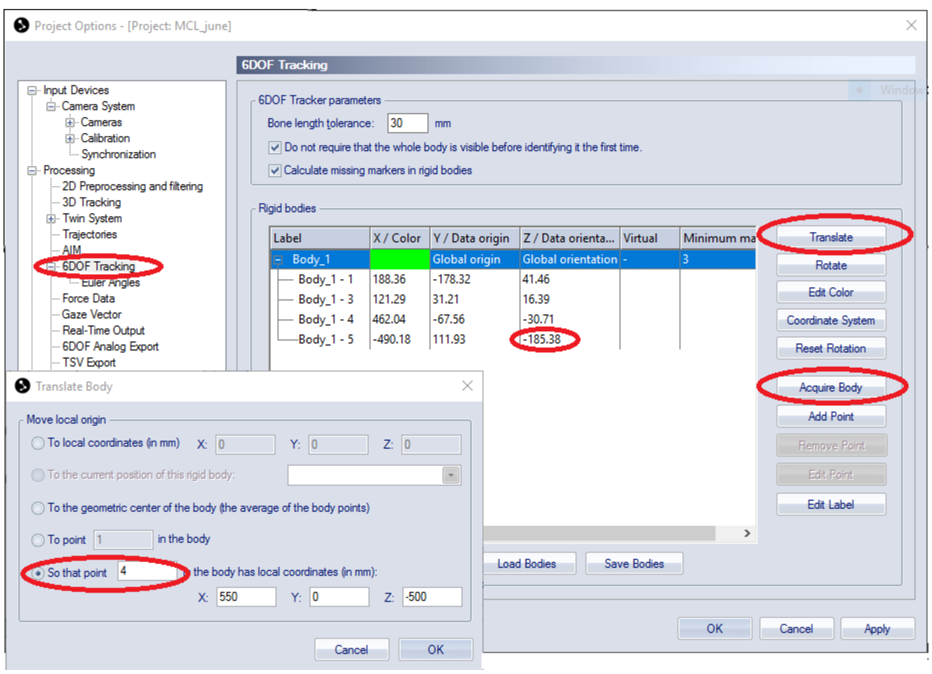


Figure 4.1: QTM window

CSE1 and the lab is now set up for experiments. However, before continuing to controlling the vessel, make sure the vessel does not take in water(there have been some issues with leakage in the hull opening around the bow thruster...)

4.2 Updating custom control system

See the MC-Lab 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, continue here with uploading the code to the vessel.

4.3 Upload VeriStand project to the vessel

With the hardware set up and ready, continue to preparing the software.

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

2. Check the communication between the laptop and CSE1. Open command prompt(cmd.exe), 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 uploading 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 Qualisys. Try deploying again. If it still does not work, try restarting the vessel(either disconnecting the battery and connecting it again, or restart the cRIO in NI MAX). Also check that the Qualisys 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 the next Chapter for instructions on operating the vessel.

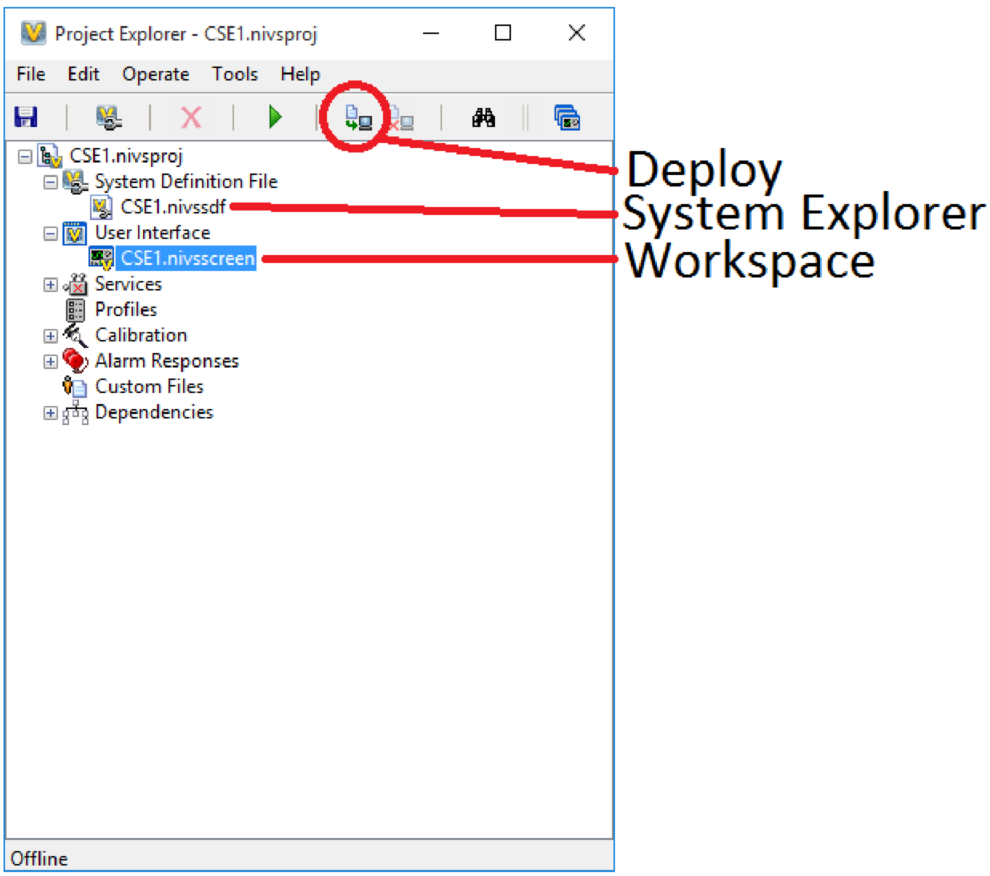


Figure 4.2: User interface in VeriStand Project Explorer

Chapter 5

Operation

After successfully deploying the VeriStand project, you control the vessel with the sixaxis-controller and/or the laptop. Use the sixaxis-controller to switch between the different operation modes:

- △ - ctrl_sixaxis2thruster
- □ - ctrl_custom
- × - ctrl_DP
- ○ - STOP

Data logging can be done in two ways, as described in the MC-Lab Handbook.

5.1 Workspace

On the laptop, use the Workspace to monitor the different variables. There are 3 screens, one for each operating mode:

ctrl_sixaxis2thruster In Figure 5.1 the workspace for sixaxis2thruster is illustrated. As the control mode is simply based on the sixaxis controller, the workspace only indicates the different variables on the vessel.

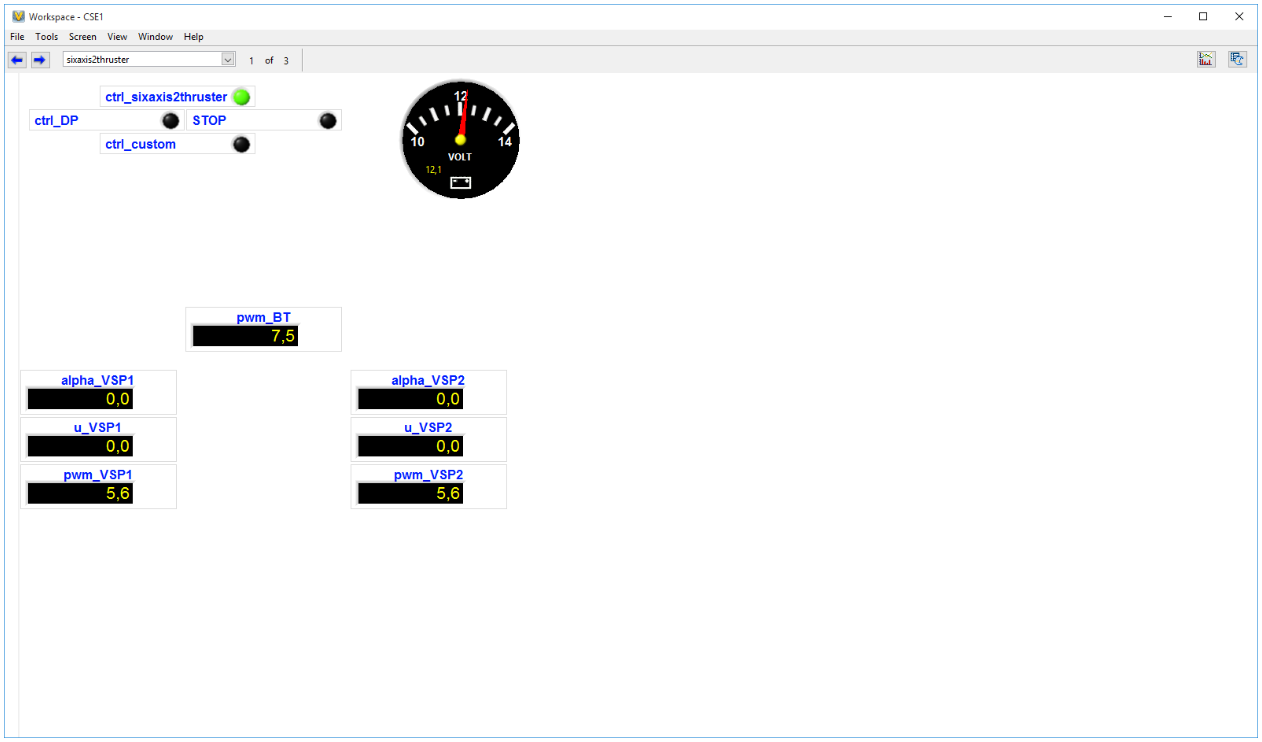


Figure 5.1: sixaxis2thruster workspace

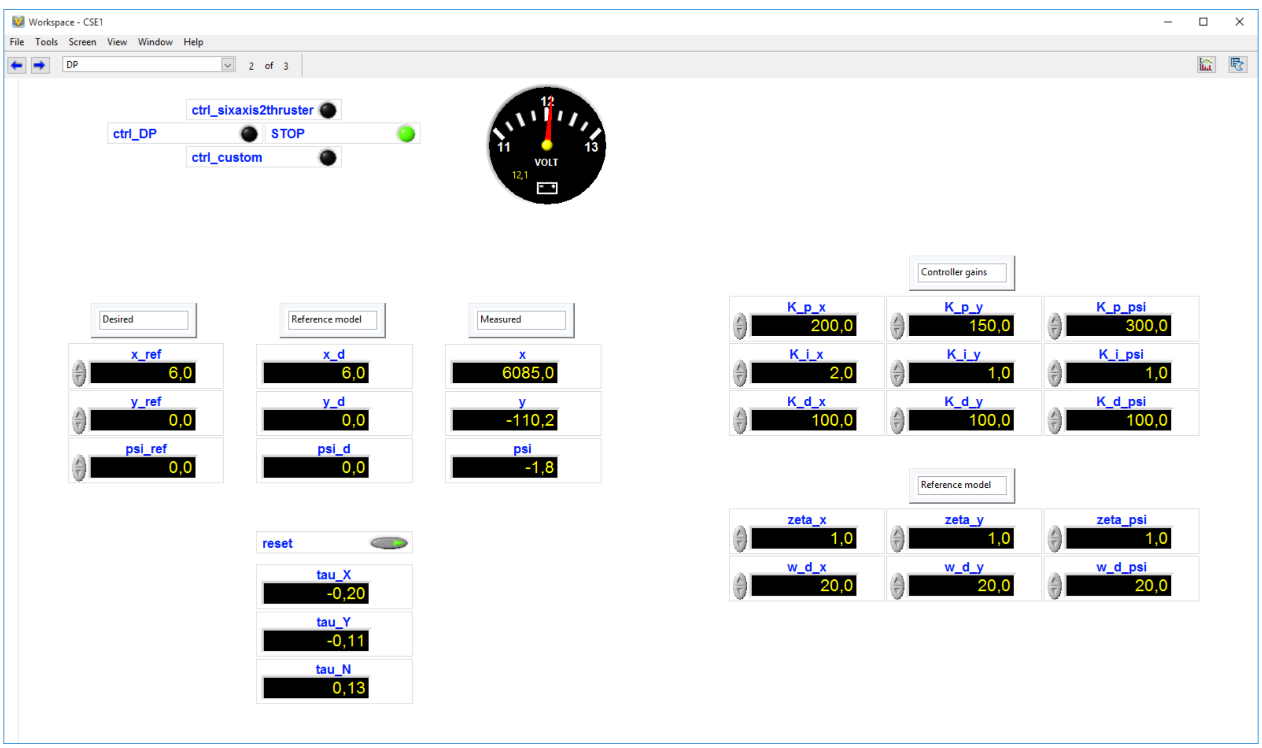


Figure 5.2: DP workspace

ctrl_DP In Figure 5.2 the workspace for the DP code is shown. Here, you define the set point for the reference model, and it's current position and orientation is given. On the right side, the model is tuned. There are some initial values, which provides a functional DP-system.

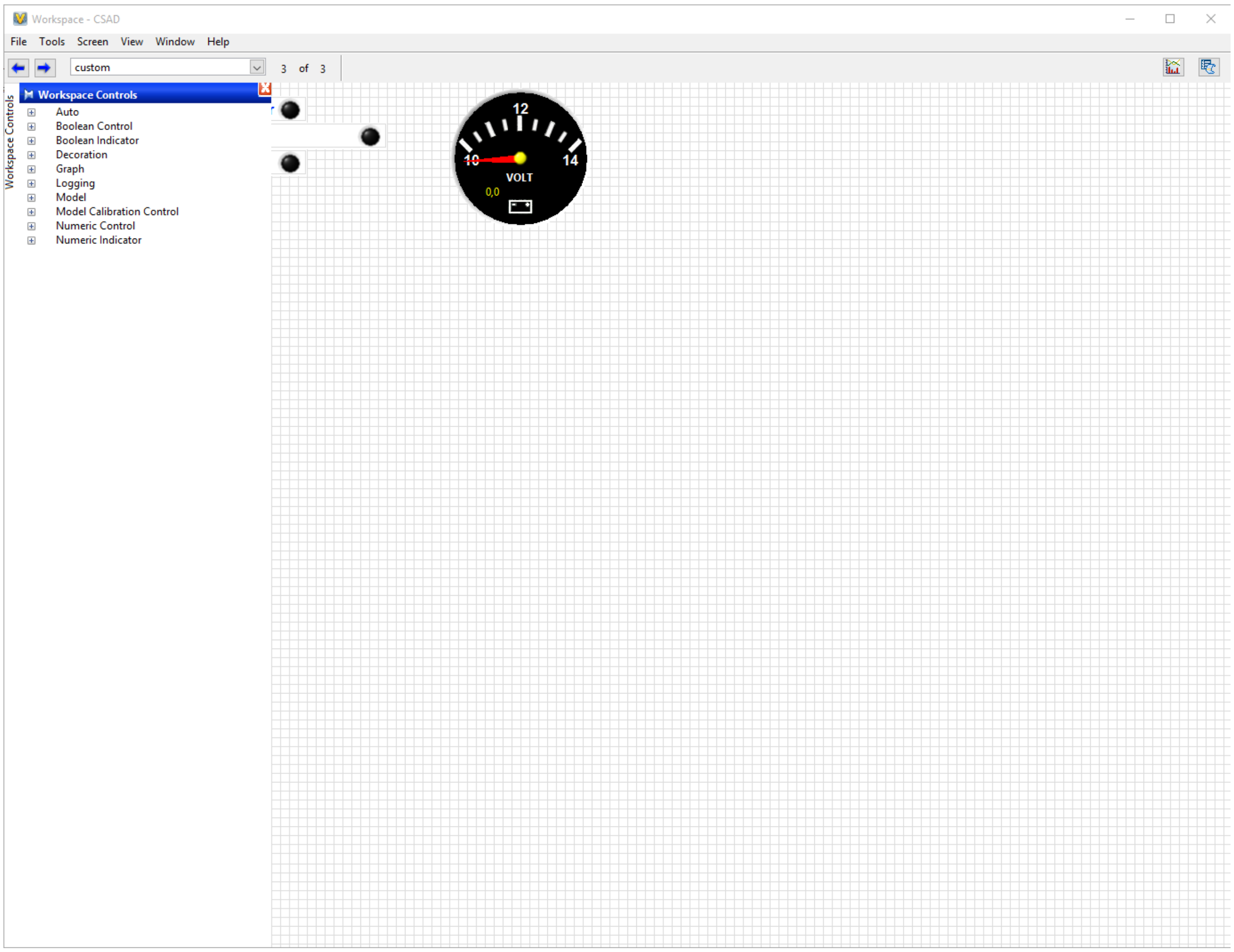


Figure 5.3: custom workspace

ctrl_custom In Figure 5.3 the workspace window for the customized control mode is illustrated. This is the one you should use when testing your control mode design. If you need more screens, press Screen -> Add Screen. To edit one screen, press Screen -> Edit Mode. On the left side(Workspace Controls), you can add different controls or indicators to monitor variables in the simulation model. Press and 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->Meter and linking it to the variable. Use the scale option to enable higher precision when tuning.

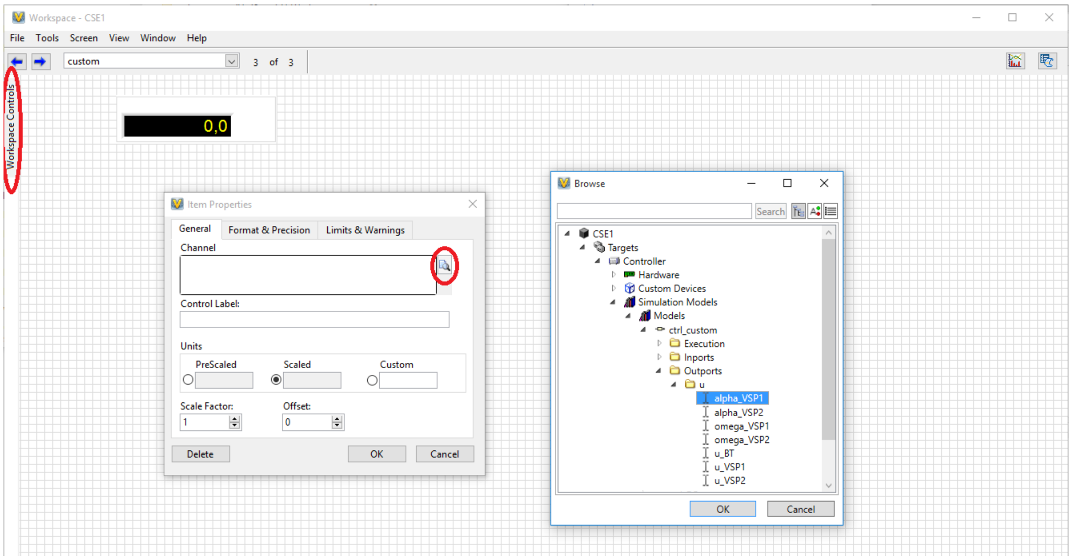


Figure 5.4: Workspace window in Edit Mode

Chapter 6

Demolition

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

1. Switch to *ctrl_sixaxis2thruster*, and navigate CSE1 near the basin wall
2. In the Project Explorer window, press to 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 from the basin, and put it in its rack
6. Leave the sixaxis controller in the vessel
7. On the Qualisys computer, quit Qualisys Track Manager
8. If you recorded any videos with the Camera System, export these videos to a memory stick, quit the software and turn of the TV-monitor
9. Do a general clean up, bring all your personal belongings with you when you leave

Bibliography

Analog Devices, 2017. Adis16364 data sheet. Tech. rep.
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.