

Simplified Marine Cybernetics

Laboratory Handbook



NTNU – Trondheim
Norwegian University of
Science and Technology

Faculty of Engineering Science and Technology
Department of Marine Technology

Introduction

This handbook is a simplified version of the Marine Cybernetics Laboratory Handbook, intended for use in TMR4243 - *Marine Control Systems II*. The full Handbook can be found on Github(https://github.com/NTNU-MCS/MC_Lab_Handbook), with extensive information on all vessels and technical description of the different software.

Structure

Chapter 1 describes the different equipment found in the Laboratory, and how to use it.

Chapter...

Contents

1	Marine Cybernetics Laboratory equipment	1
1.1	Introduction	1
1.1.1	Safety	1
1.2	Qualisys Motion Capture System	3
1.2.1	User Manual	3
1.2.2	Importing Data from the Qualisys System into ROD and MATLAB(Linux)	8
1.3	Towing carriage	11
1.3.1	Preparation before startup	11
1.3.2	Manual Operation of the Carriage	11
1.3.3	Operation Controlled automatically from PC	14
1.3.4	Troubleshooting	15
1.3.5	Note	15
1.4	Wave Generator	16
1.4.1	User Manual	16
1.5	Video-Camera	17

Chapter 1

Marine Cybernetics Laboratory equipment

1.1 Introduction

The laboratory is equipped for experimental testing of marine control systems and hydrodynamic tests. It consists of a wave basin with an advanced instrumentation package and a towing carriage. The basin, depicted in Figure 1.1, has dimensions 40m x 6.45m x 1.5m (LxBxD). The laboratory consists of the following fixed equipment:

- Qualisys Motion Capture System
- Towing Carriage
- Wave Generator
- Video-camera

These parts are thoroughly described in the next sections, with a user manual and technical description of the equipment.

1.1.1 Safety

1.1.1.1 Personnel injury

Drowning It is required to have two or more persons present when using the basin.

Electric shock The towing catenary should not be approached or touched.

Carriage collision It is forbidden to run the towing carriage when there are people alongside the basin.

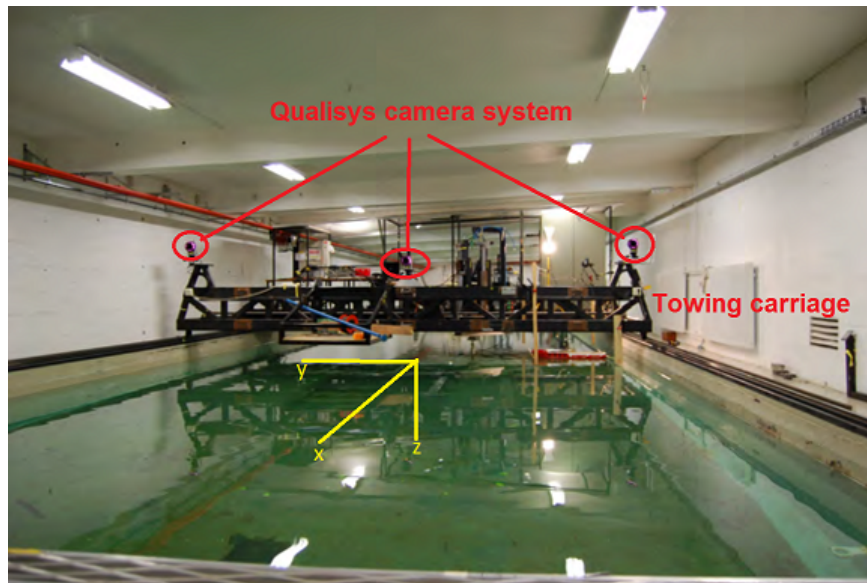


Figure 1.1: Marine cybernetics laboratory basin

Thruster blade cuts Vessels must stay in the water as long as actuators are active. Before removing the vessels from the water, the control system must be stopped and disabled, for instance by undeploying in the VeriStand project.

1.1.1.2 Material damage

Cybership Enterprise 1

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 Δ on the Sixaxis.

Loss of position measurement: -

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

Towing carriage Stop before automatic stop at high speeds.

1.2 Qualisys Motion Capture System

Qualisys provides 6 degrees of freedom data tracking. The system has millimeter precision, works in real time and is configured to 50Hz.

The positioning system consists of three Oqus high speed infrared cameras registering infrared reflectors placed on the vessel. Peer-to-peer (P2P) networking is used to transmit camera data to a dedicated computer running Qualisys Track Manager (QTM) software. QTM performs triangulation and broadcasts the vessel position over the wireless network. The system is operated from a dedicated PC in the Lab, labeled with *QTM surface vehicle*.

Add info
on zyx-
convention

1.2.1 User Manual

Start Qualisys Track Manager

1. Execute the program. The first displayed window is as in Figure 1.2.

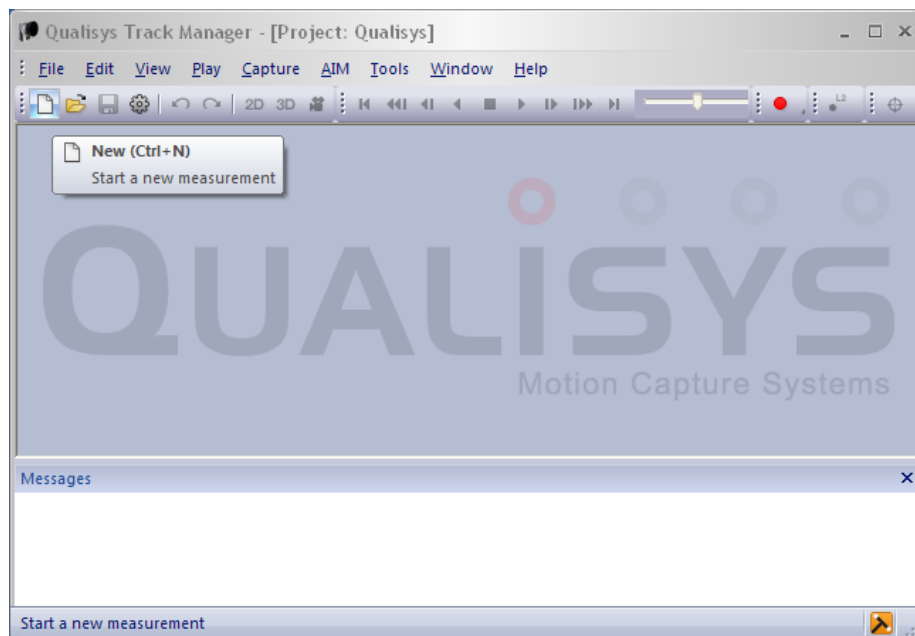


Figure 1.2: Qualisys Track Manager start window

2. Push the white sheet icon to start a new measurement. The main window should then display the 2D view, as in Figure 1.3. The squares numbered #1, #2 and #3 show the basin as seen from the respective cameras. The white dots are the vessel reflectors. A minimum of three reflectors must be visible in each camera.

Acquire body

1. Push the gear icon to access Project Options. Navigate to 6DOF Tracking, as in Figure 1.4.

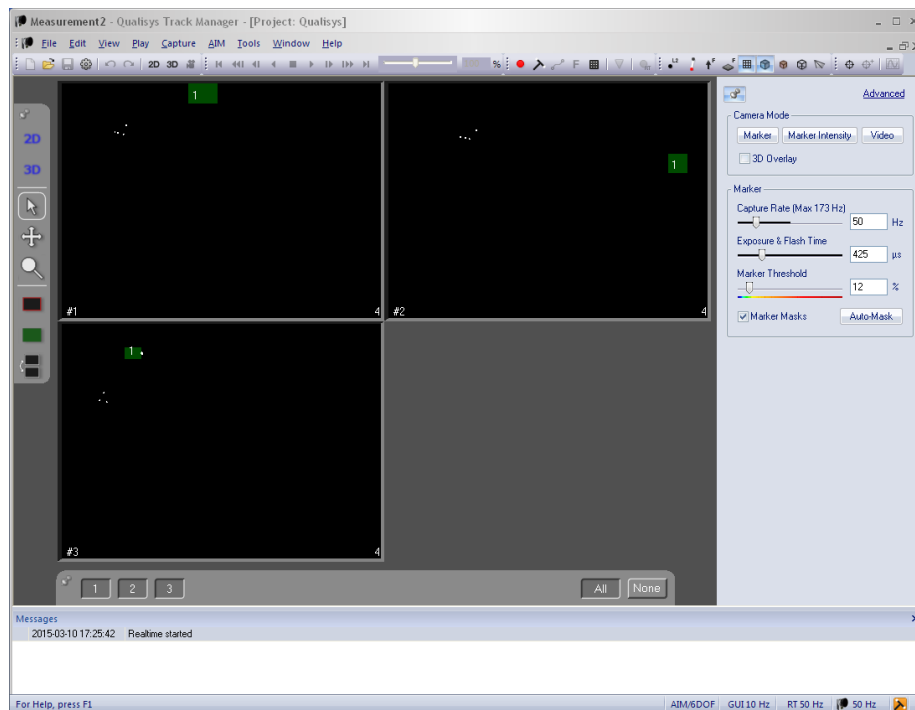


Figure 1.3: Qualisys Track Manager 2D view

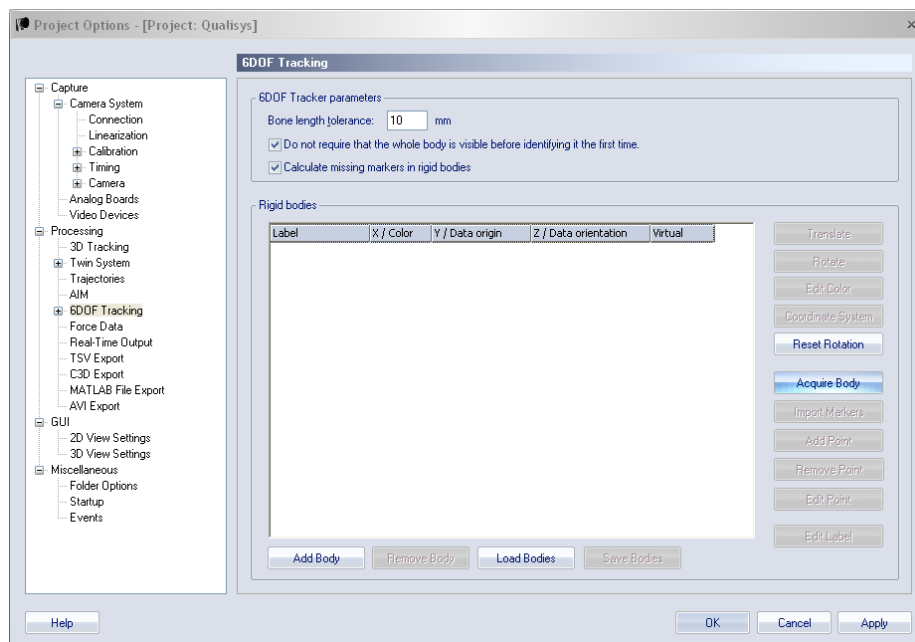


Figure 1.4: Qualisys Track Manager 6 DOF Tracking

2. Remove previous bodies, if any.
3. Align the vessel with desired heading 0 and push “Acquire Body” to get the position of the reflectors. A list appears, as in Figure 1.5.

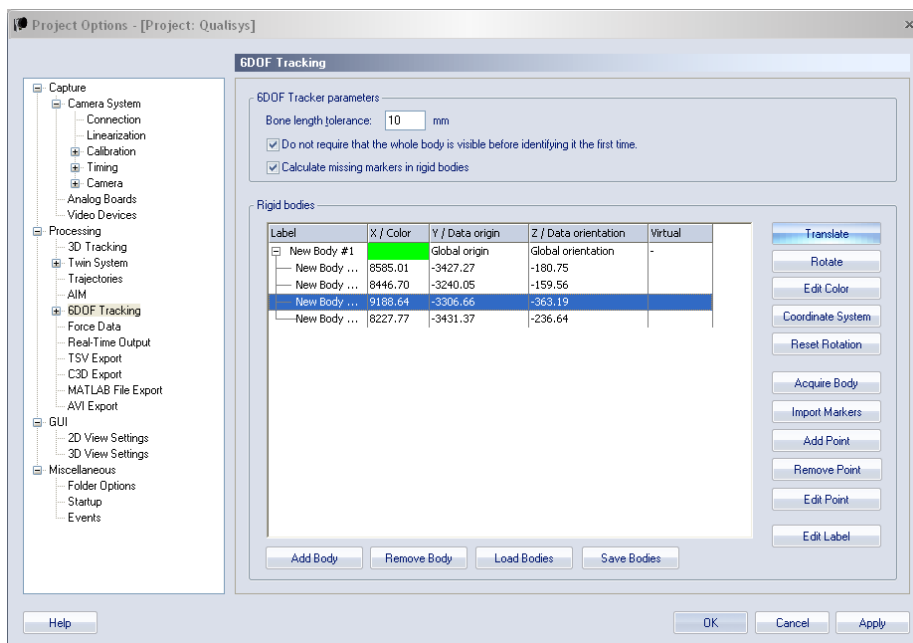


Figure 1.5: Qualisys Track Manager acquired body

4. To redefine the body fixed coordinate frame
 - (a) Chose a reference reflector. As highlighted in Figure 1.5, it may be practical to choose the highest-most, in this case reflector 3.
 - (b) Push “Translate” and enter the coordinates of the chosen reflector in the desired frame. In Figure 1.6, reflector 3 is set at $x, y, z = (0.550, 0, -0.500)$. Figure 1.7 shows the resulting coordinates after translation.
5. Finally, select 3D view to confirm that the body-fixed frame is indeed located as desired, as in Figure 1.8.

Troubleshooting

Long waiting for camera If the operation depicted in Figure 1.9 is unsuccessful after a couple of minutes, reboot the cameras by unplugging and reconnecting the power cord on the rack.

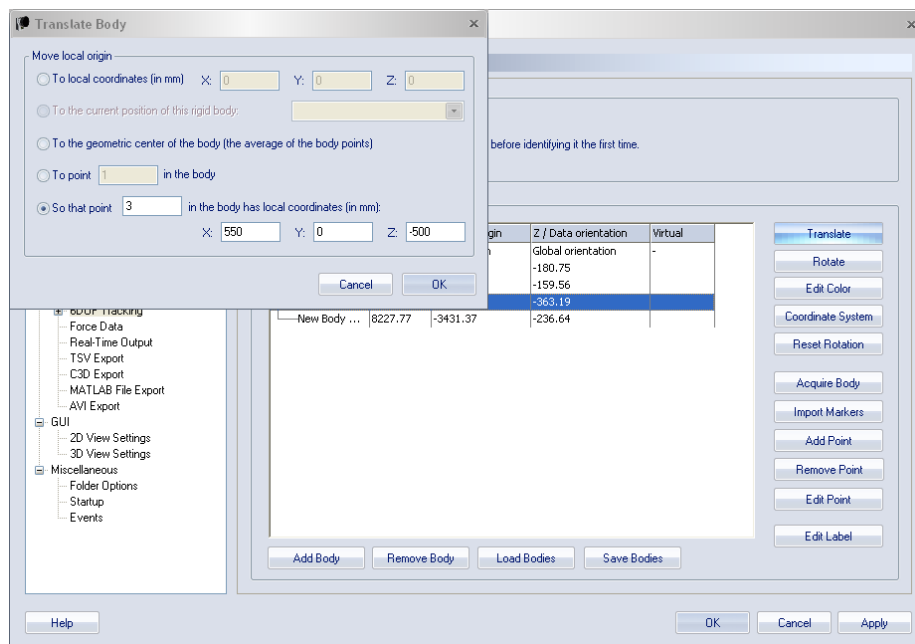


Figure 1.6: Qualisys Track Manager translate body

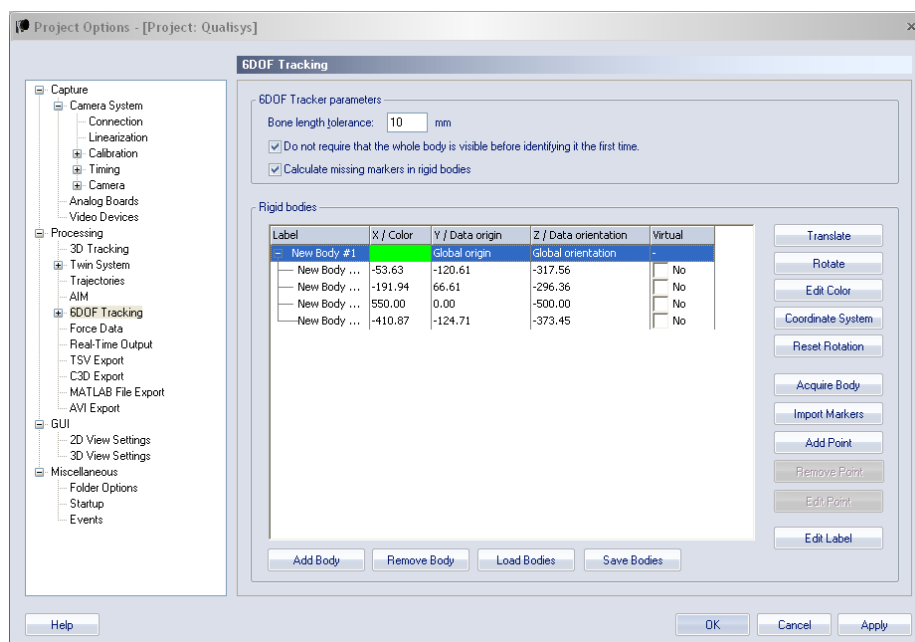


Figure 1.7: Qualisys Track Manager translated body

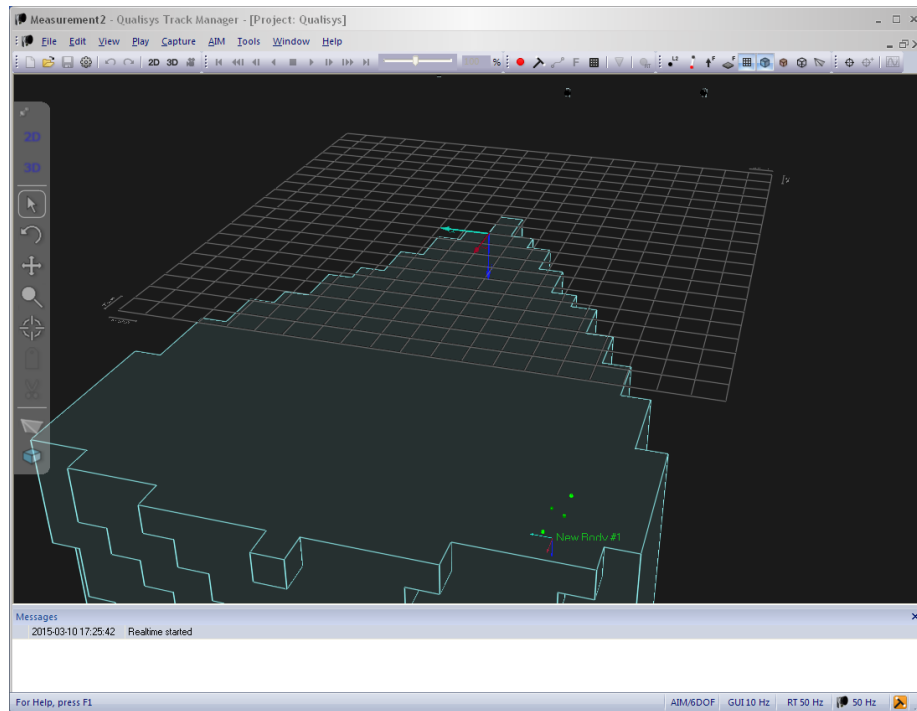


Figure 1.8: Qualisys Track Manager 3D view



Figure 1.9: Qualisys Track Manager waiting for cameras

1.2.2 Importing Data from the Qualisys System into ROD and MATLAB(Linux)

The following approach may be used to read Qualisys data into ROS and MATLAB. The method is convenient for Qualisys data into MATLAB independent on whether the rest of the system use ROS or not. The method, as described is limited to Linux-operating systems.

The first part of the manual describe how import Qualisys data into ROS, while the second part describe how to get the data from ROS into MATLAB/Simulink.

Please note the following:

- In the manual, the dollar sign \$ indicate a line of text that should be written in the Linux- terminal window.
- In the manual gedit is used as text editor. This can be replaced with the readers favourite text editor.
- The manual is written and tested for ROS-Indigo and MATLAB 2015b. It is based on MATLAB's manual for importing custom messages (Math-Works, 2016), which is and adapted and expanded to fit that of the MC lab and the Qualisys system.
- You need MATLAB version 2015a or newer in order to proceed with the MATLAB section of the manual.

1.2.2.1 Manual

If not already installed on the machine you should start by installing ROS. Follow the instructions on the ROS download page: <http://wiki.ros.org/indigo/Installation/Ubuntu>.

You should now make a ROS workspace in your home directory:

```
$mkdir -p ~/catkin_ws/src
$cd ~/catkin_ws/src
$catkin_init_workspace
```

You now need to make sure that one are sourcing the setup.bash file in your ROS workspace each time you open your terminal window. This can be done by changing the bash file with the following command:

```
$ echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

Now import the Qualisys driver from GitHub. (The driver (KumarRobotics, 2016) is available through the Apache License)

```
$ cd ~/catkin_ws/src
$ git clone https://github.com/KumarRobotics/qualisys
$ cd ~/catkin_ws
$ catkin_make
```

Open the qualisys.launch file in a text-editor

```
$ sudo gedit ~/catkin_ws/src/qualisys/launch/qualisys.launch
```

Edit the ip address and port number for the Qualisys system. (As of March 2016 the IP is: 192.168.0.10 and the port is 22222)

The driver should now be set for interfacing with Qualisys in ROS. To test it, first check that you are able to ping the Qualisys system over the MC lab WiFi.

```
$ ping 192.168.0.10
```

If you successfully pinged the qualisys system it should now be possible to listen to the data from the Qualisys system.

(Note that the Qualisys system need to recognize the IR-markers in the MC lab in order to transmit data. It may be smart to first to check that the computer running Qualisys software in the MC-Lab sees the marker)

```
$ roslaunch qualisys qualisys.launch
$ rostopic list
```

The command “rostopic list” prints the ROS active ROS topics. It should now be printed a qualisys topic in terminal. The name will depend on the name set on the Qualisys computer. In this manual the topic is named /qualisys/CSE1.

You can now listen to the data as it is published to ROS

```
$ rostopic echo /qualisys/CSE1
```

1.2.2.2 Getting Qualisys data to MATLAB

The message sent from the Qualisys system is a custom message that MATLAB does not recognize (most messages in ROS is not custom, and will be recognized by MATLAB). In order to get the Qualisys data into MATLAB you one to facilitate so that MATLAB recognize the custom message.

Start by creating a new folder ~/qualisysDir. Now copy the folder named qualisys, located in ~/catkin_ws/src and paste it into the folder ~/qualisysDir

Now one want to edit the package file so that MATLAB recognizes the messages.

```
Sudo gedit ~/qualisysDir/qualisys/package.xml
```

Add the following two lines somewhere in the main body of the package.xml file.

```
<build_depend>geometry_msgs</build_depend>
<build_depend>std_msgs</build_depend>
```

Now open MATLAB. The first step in MATLAB is to download the ROS custom message package. Type the following lines into the MATLAB command window, and follow instructions to download the ROS custom message package.

```
roboticsAddons (in MATLAB 2016)
roboticsSupportPackages (in MATLAB 2015)
```

When the download is finished paste the following commands in the MATLAB command window.

```
folderpath= '~/qualisysDir'  
rosgenmsg(folderpath)
```

Now follow the instructions generated by MATLAB in order generate the needed message type. In this process you may need allow writing permission to the file “pathdef.m”

You are now ready to get the data into MATLAB.

Remember that the Qualisys node always need to be launched before reading signals in MATLAB.

```
roslaunch qualisys qualisys.launch
```

You can now get the data into Simulink by the Subscriber block, or to MATLAB workspace by typing the following commands:

```
Subb = rossubscriber('/qualisys/CSE1');  
posedata = receive(Subb,10);
```

1.3 Towing carriage



Figure 1.10: Towing carriage

The scope of this section is to explain how to safely operate the carriage without any damage towards humans or equipment.

1.3.1 Preparation before startup

To start with, you must make sure that any items mounted or fixed to the carriage are securely fitted, so they don't prevent the operation of the carriage. All personnel must stay on the operation platform during the travel of any axis.

Locate the Emergency Button and place it so that you can easily reach it from where you are sited. **DO NOT USE THE EMERGENCY BUTTOM AS A BRAKE. YOU MUST ONLY OPERATE IT WHEN YOU ARE IN REAL EMERGENCY SITUATIONS.**

Operation console

The operation console is an All-in-one PC. The Power button is on the bottom right side of the screen. If the operation panel is not on the desktop, you can start it by double clicking on desktop Icon, shown in Figure 1.11.

1.3.2 Manual Operation of the Carriage

Setup

It is very important to select the Setup tab first before you start any operation of the carriage. As you can see in Figure 1.14, it is possible to change the travel parameters for all available axes. In principle, all axis parameters have different range limits. These are listed in Table 1.1.

All axes can be activated or deactivate by using the ON button.



Figure 1.11: Towing console

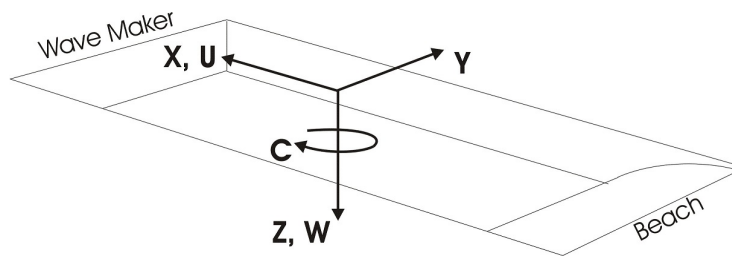


Figure 1.12: Coordinate system

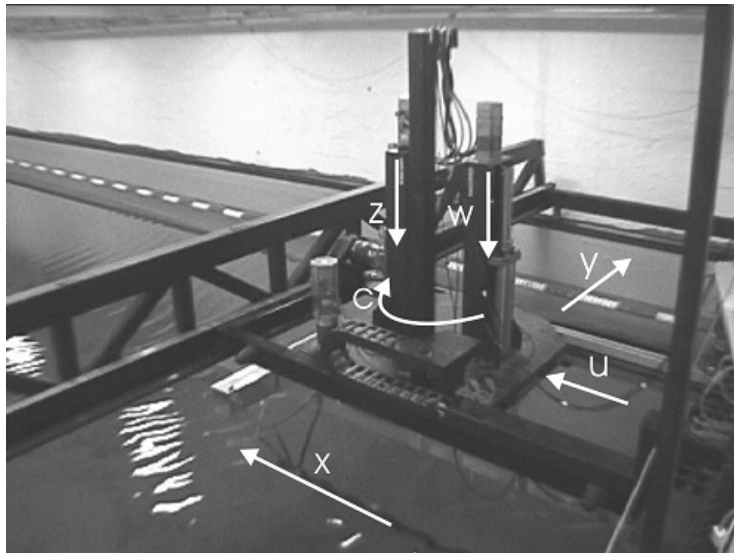


Figure 1.13: Coordinate system

For the X axis it is possible to activate a list of predefined Forward speeds. You will then have the ability to automatically change to a different speed on the next run. The list can be edited in the Main Tab Window.

Axis	Forward/Backward Speed		Acceleration Deceleration		Position Pos./Neg. Limit	
X	0 - 2.0	[m/s]	% of 0 - 0.5	m/s^2	0 - 22	[m]
Y	0 - 1.0	[m/s]	% of 0 - 1.0	m/s^2	0 - 4.5	[m]
U	0 - 1.0	[m/s]	% of 0 - 1.0	m/s^2	0 - 1	[m]
C	0 - 10	[deg/s]	% of 0 - 20	deg/s^2	0 - 255	[deg]
Z	0 - 1.0	[m/s]	% of 0 - 2.0	m/s^2	0 - 0.5	[m]
W	0 - 1.0	[m/s]	% of 0 - 2.0	m/s^2	0 - 0.5	[m]

Table 1.1: Operation Limit

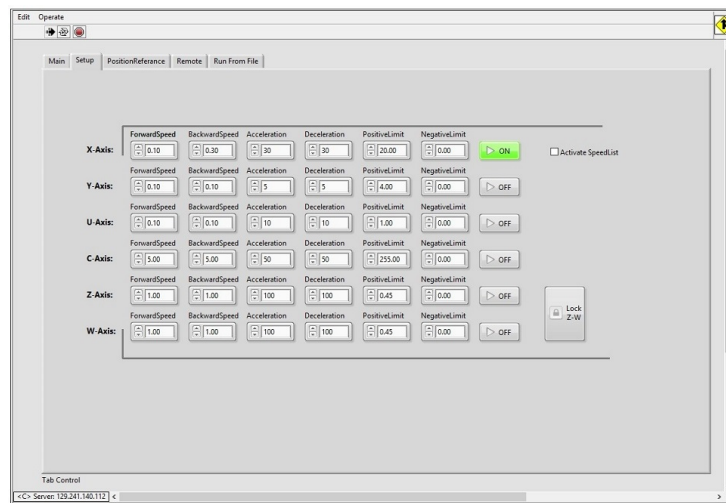


Figure 1.14: Setup

By selecting the “Lock Z-W” button the Z-W axis will operate in parallel. They will use the Z-axis parameter setup.

Main/Standard Operation

All the activated axes will operate within the limits set in the Setup. Only one axis can operate at a time. If you hit the button for another axis than the running one, it will instantly stop and new one will start running. To stop the running axis, simply hit the stop button. If no buttons are operated carriage axis will run it hit limit position of the current axis.

If an error occurs, for some reason, it can be cleared by hitting the “Power” button. If the error keeps reoccurring, please look at the Troubleshooting section of this document or contact responsible MC Lab personnel.

The current speed and position of the active axis are displayed referred to the selected limits.

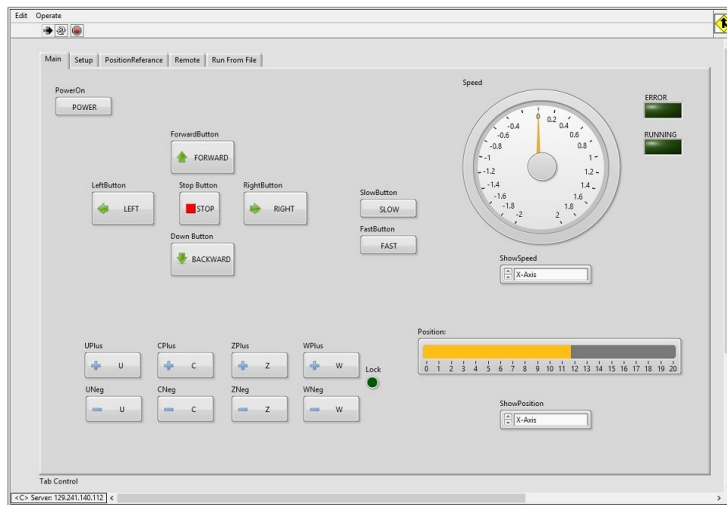


Figure 1.15: Main

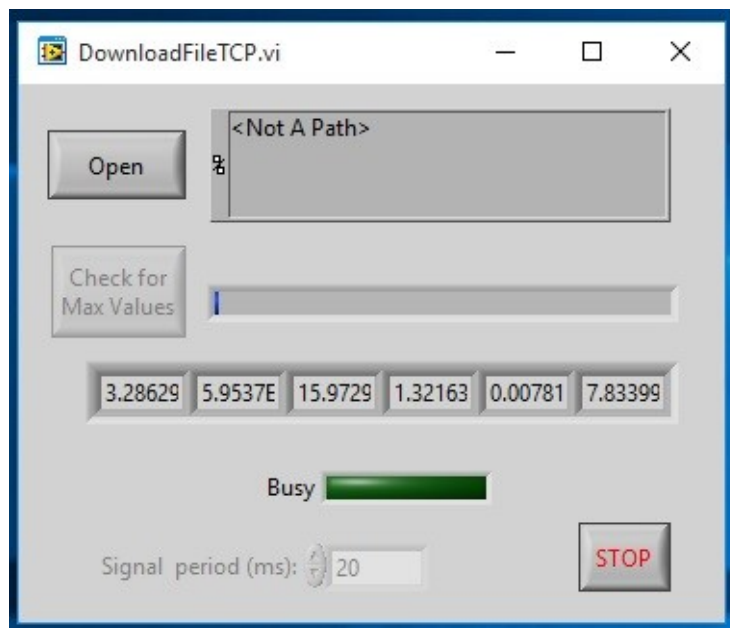


Figure 1.16: Run From File

1.3.3 Operation Controlled automatically from PC

The Trajectory Input File

All trajectories must be defined in a .mcl input file. The format of the file is slightly more general than allowed here and is the same as for the sloshing rig input. The entries in the file are

1. Time Step in ms, double precision integer (int32). Must be set to 10.

2. Number of channels, double precision integer (int32). Must be 6.
3. Position references in sequence: X(1),Y(1),U(1),C(1),Z(1),W(1), X(2),Y(2), double precision real (float32).

The following MTALAB lines write the matrix body (6xN) to file on the correct format:

```
fid=fopen(filename,'wb');
head=[10;6];
count=fwrite(fid,head,'int32');
count=count+fwrite(fid,body,'float32');
fclose(fid);
```

The resulting input file must be transferred to the realtime computer at /home/ntuser/inputpos.mcl. Normally this is done automatically when the Load button on the LabVIEW GUI is pressed.

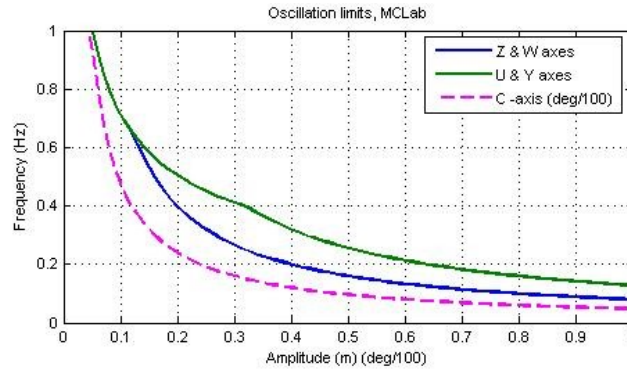


Figure 1.17: Operation Limit Amplitude vs. Frequency

1.3.4 Troubleshooting

1.3.5 Note

When the wagon is moving, no one is allowed to move on the sides of the basin.

1.4 Wave Generator

The wave maker is a single paddle wave making machine with a width of 6 meter and it is equipped with an Active Wave Absorption Control System (AWACS 2). The single paddle wave generator is controlled by a dedicated computer. The machine can produce both regular and irregular waves because of the DHI Wave Synthesizer the system has. Available spectrum are first order Stoke, JONSWAP, Pierson-Moskowitz, Bretschneider, ISSC and ITTC. Table 1.2 summarizes the generation capacity.

	Height [m]	Period T [s]
Regular waves	$H < 0.25$	0.3 - 3.0
Irregular waves	$H_s < 0.15$	0.6 - 1.5

Table 1.2: Wave generator capacity

1.4.1 User Manual

Add a description on how to use the system

1.5 Video-Camera

The laboratory is equipped with 2 high-resolution cameras for recording of activity in the basin.

Bibliography

KumarRobotics, 2016. Kumarrobotics/qualisys source code. Retrieved 10th of March 2016. <https://github.com/KumarRobotics/qualisys>. 1.2.2.1

MathWorks, 2016. Create custom messages from ros package. Retrieved 10th of March 2016. <http://se.mathworks.com/help/robotics/ug/create-custom-messages-from-ros-package.html>. 1.2.2