# Test Protocol

## Remotely Operated Underwater Vehicle

Version 0.8

Author: Alaa Saeed
Date: December 13, 2017


### Status

| | | |
|---|---|---|
| Reviewed | Amanda Andersson | 2017-12-13 |
| Approved | | |


| | | | |
|---|---|---|---|
| Course name: | Automatic Control Project Course | E-mail: | tsrt10rov2017@gmail.com |
| Project group: | ROV2017 | Document responsible: | Alaa Saeed |
| Course code: | TSRT10 | Author's E-mail: | alasa433@student.liu.se |
| Project: | Remotely Operated Underwater Vehicle | Document name: | TestProtocol.pdf |

## Project Identity

| | |
|---|---|
| **Group E-mail:** | tsrt10rov2017@gmail.com |
| **Homepage:** | http://www.isy.liu.se/edu/projekt/tsrt10/2017/rov/ |
| **Orderer:** | Jonas Linder, Linköping University<br>**Phone:** +46 13 28 28 04<br>**E-mail:** jonas.linder@liu.se |
| **Customer:** | Rikard Hagman, Combine Control Systems AB<br>**Phone:** +46 72 964 70 59<br>**E-mail:** rikard.hagman@combine.se |
| **Course Responsible:** | Daniel Axehill, Linköping University<br>**Phone:** +46 13 28 40 42<br>**E-mail:** daniel@isy.liu.se |
| **Project Manager:** | Amanda Andersson<br>**Phone:** +46 76 843 40 79<br>**E-mail:** amaan181@student.liu.se |
| **Advisors:** | Kristoffer Bergman, Linköping University<br>**Phone:** +46 73 847 31 51<br>**E-mail:** kristoffer.bergman@liu.se |

## Group Members

| Name | Responsibility | Phone | E-mail (@student.liu.se) |
|---|---|---|---|
| Amanda Andersson | Project manager | +46 76 843 40 79 | amaan181 |
| Marcus Homelius | Documentation | +46 70 245 90 96 | marho949 |
| George Jajji | Design | +46 70 790 16 17 | geoja551 |
| Martin Johannesson | Hardware | +46 76 949 79 69 | marjo790 |
| Mattias Mucherie | Information | +46 76 238 53 06 | matmu715 |
| Fredrik Nilsson | Software | +46 73 575 49 11 | freni169 |
| Anton Nordlöf | Modelling & Simulation | +46 76 145 04 90 | antno848 |
| Alaa Saeed | Tests | +46 76 207 57 86 | alasa433 |

## Document History

| Version | Date | Changes made | Sign | Reviewer(s) |
|---|---|---|---|---|
| 0.1 | 2017-11-08 | First draft. | All | AA |
| 0.2 | 2017-11-09 | Second draft. | All | AA |
| 0.3 | 2017-11-09 | Third draft. | All | AA |
| 0.4 | 2017-11-22 | Fourth draft. | All | MH |
| 0.5 | 2017-11-27 | Fifth draft. | All | AA |
| 0.6 | 2017-12-04 | Sixth draft. | All | AA, MH |
| 0.7 | 2017-12-05 | Seventh draft. | All | MH |
| 0.8 | 2017-12-13 | Eighth draft. | All | AA |

# Contents

# Notations

**DOF** Degrees of Freedom

**ESC** Electronic Speed Controller

**GUI** Graphical User Interface

**IMU** Inertial Measurement Unit

**I/O** Input/Output

**ROS** Robot Operating System

**ROV** Remotely Operated Underwater Vehicle

# 1 Introduction

This document contains descriptions of all tests that were performed during this project. The results from the tests and whether the tested requirements were fulfilled is documented here. A description of the planned tests can be found in [3] and a description of the tested requirements can be found in [2].

The following format will be used to describe the different tests:

**Test number:** Which test that was performed.
**Requirement number:** Which requirements that the test was supposed to verify.
**Test description:** A detailed description of how the the test was performed.
**Test result:** What the result of the test was.
**Approved:** Whether the requirements were fulfilled.
**Executed by:** Persons who performed the test.
**Date:** Which date the test was performed.

## 1.1 Tests not Performed

The following tests were not performed due to their requirements having priority 2: 14, 30, 31, 47, 52, 58, 59, 60, 62, 69

# 2 General System and Interface Tests

**Test number:** 1
**Requirement number:** 1
**Test description:** In order to verify that ROS is used as the main framework and for communication with the ROV, the source code was inspected.
**Test result:** The result shows that the ROV is communicating through ROS and it is also the main framework used in the code.
**Approved:** The requirement is fulfilled.
**Executed by:** FN
**Date:** November 28, 2017

**Test number:** 2
**Requirement number:** 7, 8
**Test description:** ROS messages from a run were logged and afterwards inspected in order to verify that the GUI could communicate with other modules using ROS topics.
**Test result:** The GUI can visualize any topic in real time and all active topics can be monitored via the topic monitor. It can also send new parameters to modules via the dynamic reconfiguration tab. The results show that the GUI can send and receive information from correct modules via ROS topics.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** November 26, 2017

**Test number:** 3
**Requirement number:** 21, 22, 23

**Test description:** ROS messages from a run were logged and inspected in order to verify that the control module was subscribing to reference topics for position, attitude, linear and angular velocity, reference trajectory and publishing thruster signals.
**Test result:** The result shows that the controller is receiving references for position, attitude, linear velocities and angular velocities and trajectory and publishing thruster signals.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** December 5, 2017

**Test number:** 4
**Requirement number:** 49, 50
**Test description:** The ROS messages that were logged from a run were inspected in order to verify that the sensor fusion module were subscribing to sensor topics and publishing the estimated states.
**Test result:** The result shows that the sensor fusion is receiving sensor data by subscribing to correct topics and transmitting the estimated states to the correct topic.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** November 20, 2017

**Test number:** 5
**Requirement number:** 97, 99
**Test description:** The ROS messages that were logged from a run were inspected in order to verify that the I/O-module could publish sensor data and receive control signals. Data was logged from different topics associated with different sensors connected to the I/O-module. The ROV was operated by using manual control with the XBOX controller, showing that it is subscribing to the thruster control signal topic.
**Test result:** The result shows that the I/O-module is communicating over ROS by subscribing and publishing to correct ROS Topics.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** November 20, 2017

**Test number:** 6
**Requirement number:** 104, 105, 106, 107
**Test description:** The ROS massages published from the pathfinder module were logged during a short test run, and afterwards inspected. The topic which it publishes the trajectory to should be the same as the controller and GUI subscribe to. The pathfinder module should also subscribe to the state topic, published by the sensor fusion module.
**Test result:** The result showed that the pathfinder published data on appropriate topics and subscribes to correct topics.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** November 08, 2017

**Test number:** 7
**Requirement number:** 2
**Test description:** The output logs were inspected after a run to verify that there exists functionality to log raw sensor data, reference signals, control signals and state estimates.
**Test result:** The output logs contain raw sensor data, reference signals, control signals

and state estimates which shows that the there exists functionality to log raw data.
**Approved:** The requirement is fulfilled.
**Executed by:** MM
**Date:** November 08, 2017

**Test number:** 8
**Requirement number:** 3, 4
**Test description:** To verify that the correct thruster is responding to the correct references from the XBOX-controller a simulation was made controlled from the XBOX-controller. Each of the linear and angular velocities were excited, one at a time, and the thruster forces and reference messages from the XBOX-controller were logged and inspected.
**Test result:** The result showed that the correct thrusters responds when sending a velocity reference from the XBOX-controller.
**Approved:** The requirements are fulfilled
**Executed by:** AA
**Date:** December 05, 2017

**Test number:** 9
**Requirement number:** 92, 93, 94, 95, 96, 98
**Test description:** In order to verify that the sensors could communicate with the I/O module, the logged ROS topics from a telegraph test were inspected. If they contained reasonable data, the communication between the I/O module and the sensors is successful.
**Test result:** The topics contained reasonable data from all the sensors. The barometer measurements are not sent to a topic, but it is used to set a pressure offset and produces an error message if it failed. Since no error message appeared during the run, the barometer could also communicate with the I/O-module.
**Approved:** The requirements are fulfilled.
**Executed by:** MH
**Date:** November 03, 2017

# 3 Design Tests

**Test number:** 10
**Requirement number:** 17, 18, 19, 20
**Test description:** In order to verify that the code is written according to the ROS coding standard, the source code was inspected. To verify that all comments in the source code was written in english, the source code including MATLAB scripts was inspected. Since no non-ROS C++ code were written, the requirement 18 is automatically fulfilled. The code has been version controlled in Git.
**Test result:** The result shows that all ROS code fulfills the ROS coding standard and all source code has comments written in english. The code has been version controlled.
**Approved:** The requirements are fulfilled.
**Executed by:** FN
**Date:** December 8, 2017

**Test number:** 11
**Requirement number:** 24
**Test description:** To verify that it exist a controller for linear and angular velocity,

inspection of the controller code was made.
**Test result:** A controller for linear and angular velocity exist in the controller code.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** December 5, 2017

**Test number:** 12
**Requirement number:** 25
**Test description:** To verify that it exist a controller for attitude and position, inspection of the controller code was made.
**Test result:** A controller for attitude and position exist in the controller code.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** December 5, 2017

**Test number:** 13
**Requirement number:** 26
**Test description:** To verify that it exist a controller for following a reference trajectory, inspection of the controller code was made.
**Test result:** A controller for following a reference trajectory exist in the controller code.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** December 5, 2017

**Test number:** 15
**Requirement number:** 64
**Test description:** To verify that a mathematical model exists that describes the dynamic of the ROV, an inspection of the Matlab and Simulink model was made.
**Test result:** The implemented mathematical model correspond with the theoretical model described in the Design Specification [1].
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** November 27, 2017

**Test number:** 16
**Requirement number:** 71
**Test description:** To verify that there exists a GUI for the simulation environment the code was inspected.
**Test result:** The inspection showed that the simulator can be controlled using the Simulink environment and Matlab script as a GUI.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** November 08, 2017

**Test number:** 17
**Requirement number:** 78
**Test description:** To check that the simulator is built in Matlab/Simulink the simulator environment was inspected.
**Test result:** The simulator is built in Matlab and Simulink, see Figure 1.
**Approved:** The requirement is fulfilled.
**Executed by:** AN

**Date:** November 08, 2017

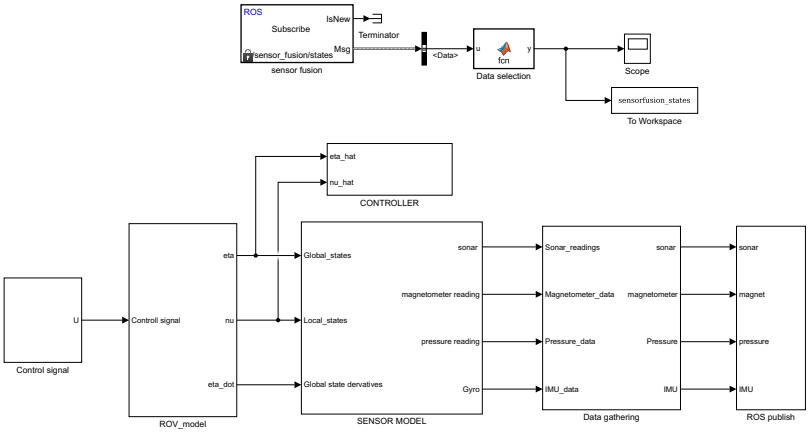
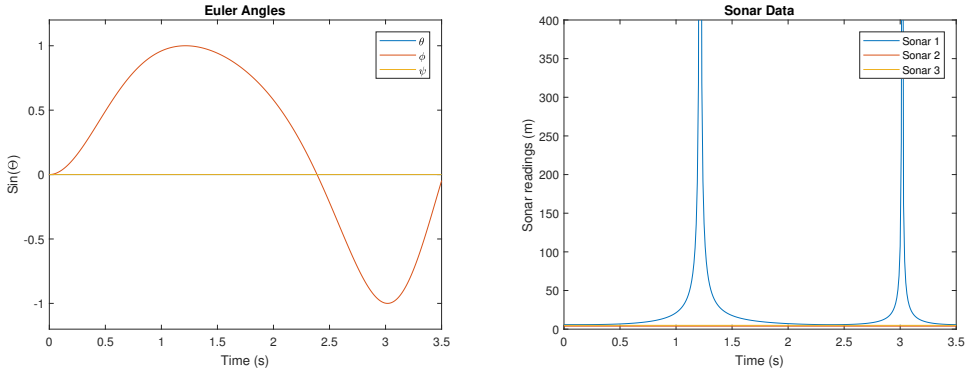



Figure 1: An overview of the simulator environment.

**Test number:** 18
**Requirement number:** 81
**Test description:** To inspect that the ROV is simulated in an environment without a visible bottom or surface a simulation test was done. The ROV is fed with a constant control in $\tau(\theta)$ signal making it pitch from the starting position a full 360°. This will result in a sonar reading going towards infinity when the angle closes ± 90°
**Test result:** The resulting Euler angles and sonar readings from the test is shown in Figure 2. The test shows that when the sinus value for the $\theta$ angle is ±1 i.e. for 90° and 270° the sonar readings spike towards infinity .
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** December 05, 2017

**Test number:** 19
**Requirement number:** 82
**Test description:** In order to verify that the sensors have been implemented in the simulator, a simulation when the ROV is traveling with constant thruster force in different directions was performed. The output from the sensor module was inspected in order to verify that they were implemented in the simulator.
**Test result:** The output from the sensor module is shown in Figures 3 and 4. All sensor values have been scaled down to fit into the same figure.
**Approved:** The requirement is fulfilled.
**Executed by:** FN
**Date:** November 20, 2017

**Test number:** 20




(a) The sinus values of the Euler angles during the simulation

(b) Sonar readings during simulation sonar 1 is forward looking 2 is right and 3 is left.

Figure 2: The sonar and angular readings during simulation for test 18.

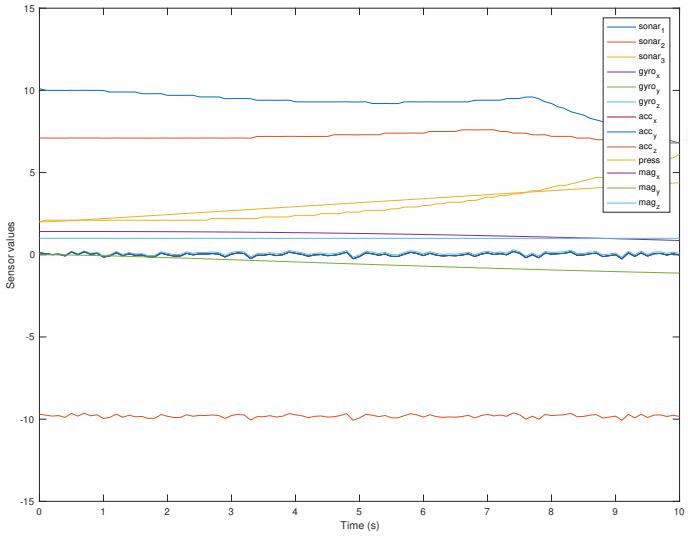



Figure 3: Output from the sensor module when the ROV is traveling forward while slightly turning to the right about its $z$-axis.

**Requirement number:** 79
**Test description:** The test was done by running a simulation twice with the same control signals but different parameter values to see that the state estimates were effected. The test was done with a step input in $\tau(u)$ at time 1 s and the initial Euler angles set to zero. This simulates a movement forward in the $x$-direction of the pool. The parameter that include the mass of the ROV was changed between the two simulations to double the normal weight.
**Test result:** The test result is shown in Figure 5 and it can be observed that the different parameter value gives different results.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** November 08, 2017

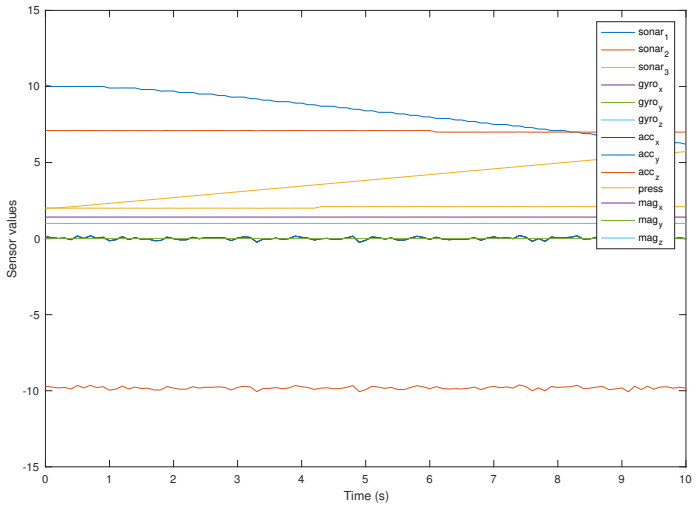



Figure 4: Output from the sensor module when the ROV is traveling in $x$- and $z$-direction.

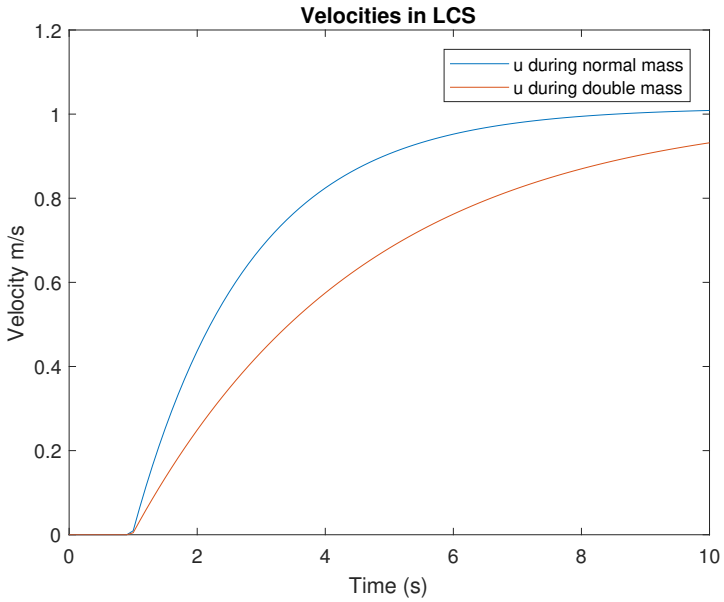



Figure 5: The state of $u$ plotted over time with the mass changed from normal to double.

**Test number:** 21
**Requirement number:** 80
**Test description:** To see that the simulation can run without any error messages one simulation was done.
**Test result:** The simulator runs without any error messages.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** December 03, 2017

# 4 Graphical User Interface Tests

**Test number:** 22
**Requirement number:** 9
**Test description:** References for the linear and angular position and velocity were set from a script and then the GUI was inspected to verify that the shown references were the same as the sent.
**Test result:** The plot in the GUI showed the same references as the commanded references in the script.
**Approved:** The requirement is fulfilled.
**Executed by:** AA
**Date:** December 04, 2017

**Test number:** 23
**Requirement number:** 10
**Test description:** To verify that their exist functionality in the GUI to present references, state estimates, control signals and operating modes, the GUI was inspected.
**Test result:** The GUI can show references, state estimates, control signals and the current operating mode, see Figure 6.
**Approved:** The requirement is fulfilled.
**Executed by:** AA
**Date:** December 04, 2017

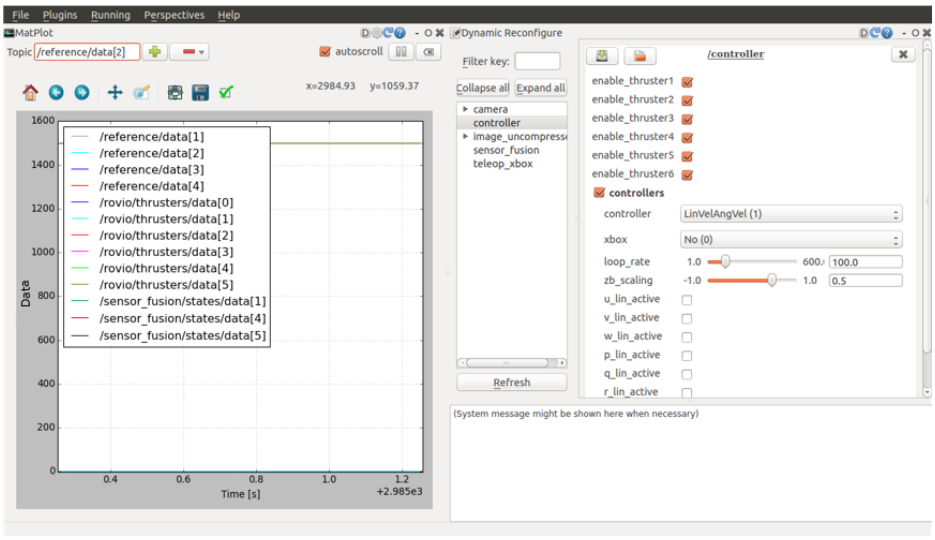



Figure 6: Screen-shot of the GUI. The active operating mode is shown to the right, the references, state estimates and control signals are shown in the plot.

**Test number:** 24
**Requirement number:** 11
**Test description:** To verify that the GUI is able to present a graphical representation of the reference trajectory, start and end points for the pathfinder were set in the GUI.
**Test result:** Figure 9 shows that a graphical representation of the path can be presented

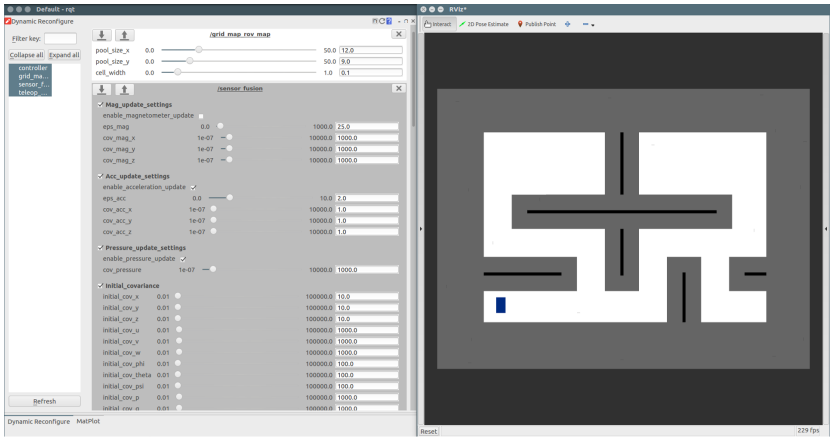


Figure 7: In the GUI, the map is created with RViz. Black illustrates obstacles, dark grey illustrates zones that are to close to the obstacles and light grey the generated path.

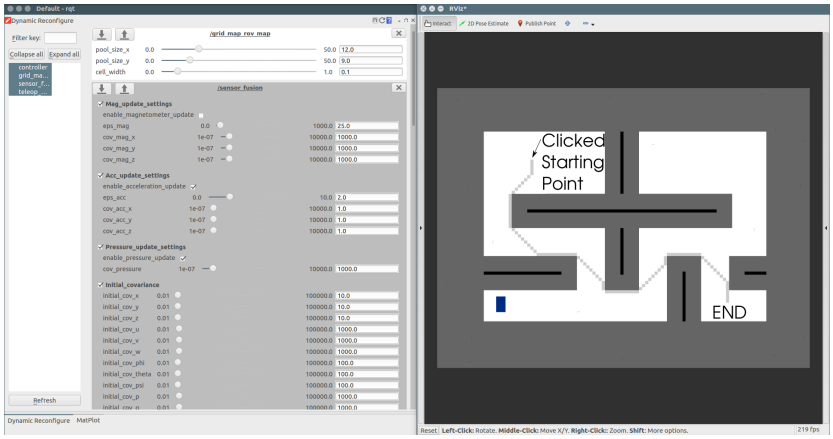



Figure 8: The path's start point is set by clicking on the map. The path is then recalculated to start at set start point.

in the GUI.
**Approved:** The requirement is fulfilled.
**Executed by:** MM
**Date:** November 27, 2017

**Test number:** 25
**Requirement number:** 12, 106
**Test description:** To verify that the path's start and end points were equal to the point set in the GUI, start and end points for the pathfinder were set in the GUI by clicking on the map. The ROS messages from the pathfinder was logged and the map on the GUI showed the new calculated path.
**Test result:** ROS messages were sent from the pathfinder module. Figure 7, 8 and 9 showed that the path's start and end coordinates were equal to the clicked points.
**Approved:** The requirements are fulfilled.
**Executed by:** AA
**Date:** November 08, 2017

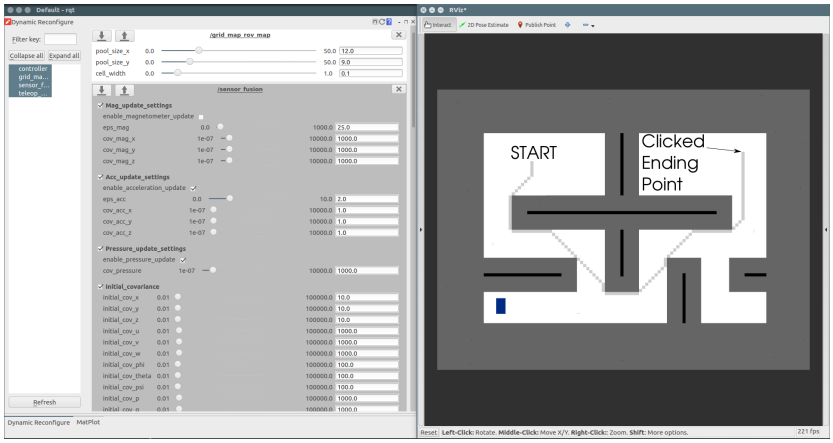



Figure 9: The path's end point is set by clicking on the map. The path is then recalculated to end at set end point.

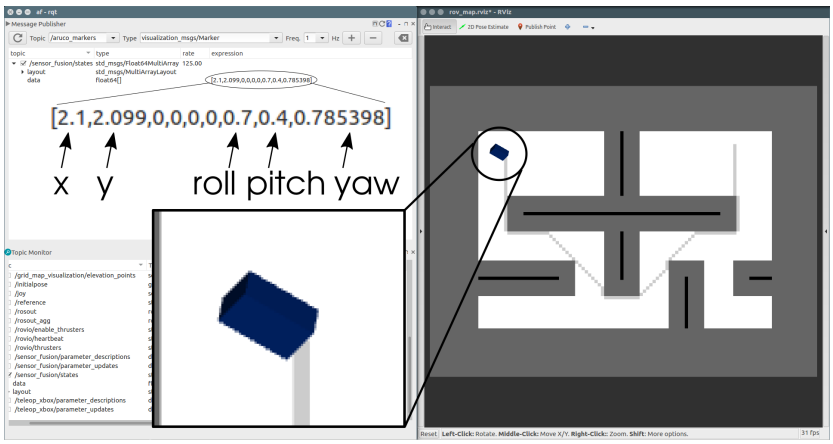



Figure 10: The ROV's position and attitude correspond to the simulated state estimate.

**Test number:** 26
**Requirement number:** 13, 14
**Test description:** To verify that the GUI is able to present a graphical representation of the map, the ROV's position and the ROV's position estimates. The estimates were simulated by sending a message with a position and an attitude on the corresponding topic.
**Test result:** Figure 10 shows the graphical view of the ROV and its position corresponds to the simulated state estimates.
**Approved:** The requirements are fulfilled.
**Executed by:** MM
**Date:** November 27, 2017

**Test number:** 27
**Requirement number:** 15
**Test description:** To verify that the GUI is able to change mode of the controller. The mode was changed in the GUI while being logged on the controller node.
**Test result:** The mode is changing, see Figure 11 and it is working as intended.
**Approved:** The requirement is fulfilled.

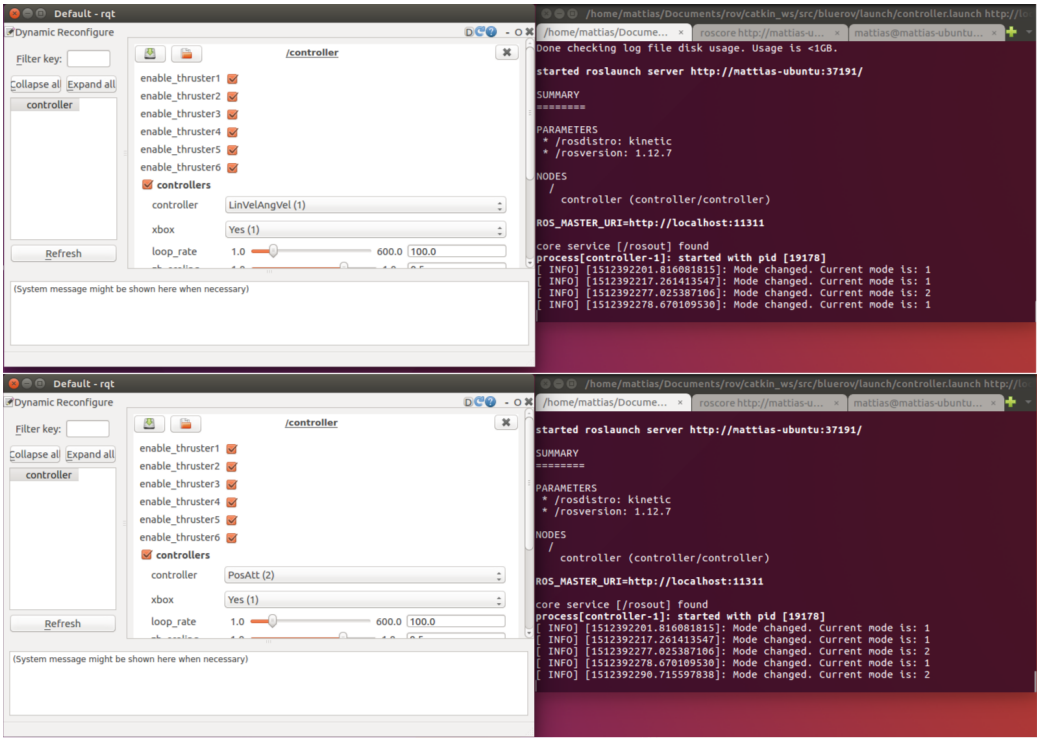



Figure 11: The controller mode was changed and a ROS message was logged to ensure that the mode was changed.

**Executed by:** MM
**Date:** December 4, 2017

**Test number:** 28
**Requirement number:** 16
**Test description:** To verify that parameters can be changed from the GUI, two simulations were made with the same control signals. Between the two simulations, initial $y$-position was changed from the GUI.
**Test result:** Figure 12 show that the initial condition of $y$-position was changed between the two simulations.
**Approved:** The requirement is fulfilled.
**Executed by:** AA
**Date:** December 4, 2017

**Test number:** 29
**Requirement number:** 72
**Test description:** The screen was recorded in slow motion to evaluate the time it takes for the GUI to update the visualization of the ROV when the states are updated.
**Test result:** It takes around 60 ms for the GUI to present the position and attitude of the ROV.
**Approved:** The requirement is fulfilled
**Executed by:** MM
**Date:** November 29, 2017

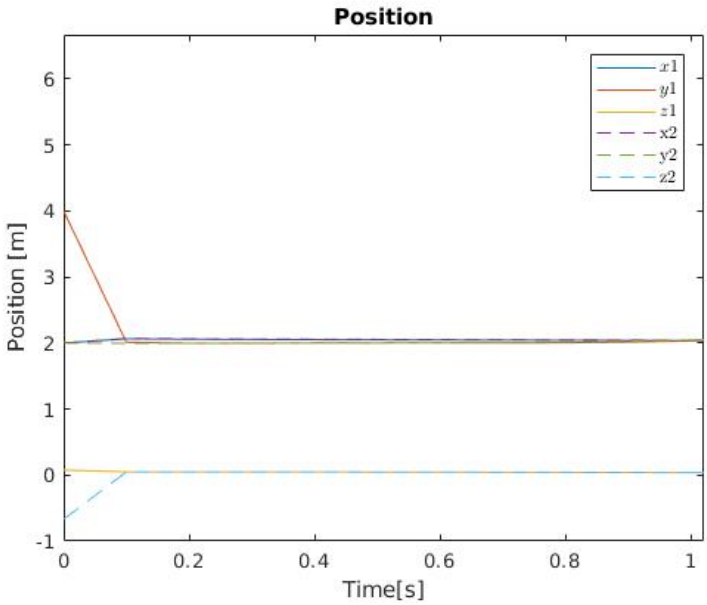



Figure 12: The $x$-, $y$- and $x$-position from the two different simulations.

# 5 Control System Tests

**Test number:** 32
**Requirement number:** 27, 31, 32, 33, 34, 35
**Test description:** The roll performance was verified by performing a step in the roll angle ($\phi$) from 0 rad to 0.52 rad (30°) and then inspecting the plots with respect to stationary error, rise time, overshoot and settling time.
**Test result:** The step response for the roll angle ($\phi$) can be seen in Figure 13.
**Approved:** The requirements are fulfilled.
**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 33
**Requirement number:** 27, 31, 32, 33, 34, 35
**Test description:** The pitch performance was verified by performing a step in the pitch angle ($\theta$) from 0 rad to 0.52 rad (30°) and then inspecting the plots with respect to stationary error, rise time, overshoot and settling time, see Figure 14.
**Test result:** The step response for the pitch angle ($\theta$) can be seen in Figure 14.
**Approved:** The requirements are fulfilled.
**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 34
**Requirement number:** 27, 31, 32, 33, 34, 35
**Test description:** The yaw performance was verified by performing a step in the yaw angle ($\psi$) from 0 rad to 0.52 rad (30°) and then inspecting the plots with respect to stationary error, rise time, overshoot and settling time, see Figure 15.
**Test result:** The step response for the yaw angle ($\psi$) can be seen in Figure 15.
**Approved:** The requirements are fulfilled.

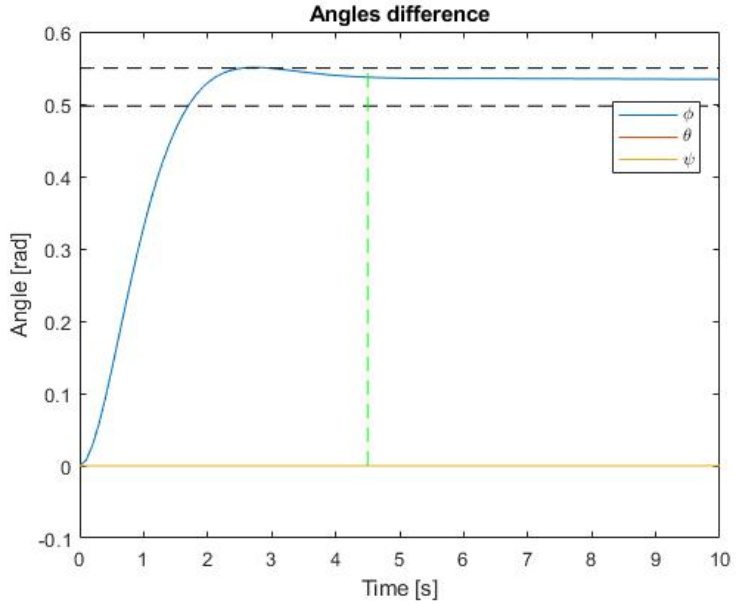



Figure 13: Step response when doing a step in roll angle ($\phi$) from 0 rad to 0.52 rad (30°). The rise time is 1.2 s, the settling time is 3 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 35
**Requirement number:** 28, 36, 37, 38, 39
**Test description:** The angular velocity performance about the LCS $x$-axis ($p$) was verified by performing a step in $p$ from 0 rad/s to 0.5 rad/s (28.6°/s) and inspecting the plots with respect to stationary error, rise time, overshoot and settling time.
**Test result:** The step response for the angular velocity about LCS $x$-axis ($p$) can be seen in Figure 16.
**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

**Test number:** 36
**Requirement number:** 28, 36, 37, 38, 39
**Test description:** The angular velocity performance about the LCS $y$-axis ($q$) was verified by performing a step in $q$ from 0 rad/s to 0.5 rad/s (28.6°/$s$) and inspecting the plots with respect to stationary error, rise time, overshoot and settling time.
**Test result:** The step response for the angular velocity about LCS $y$-axis ($q$) can be seen in Figure 17.
**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

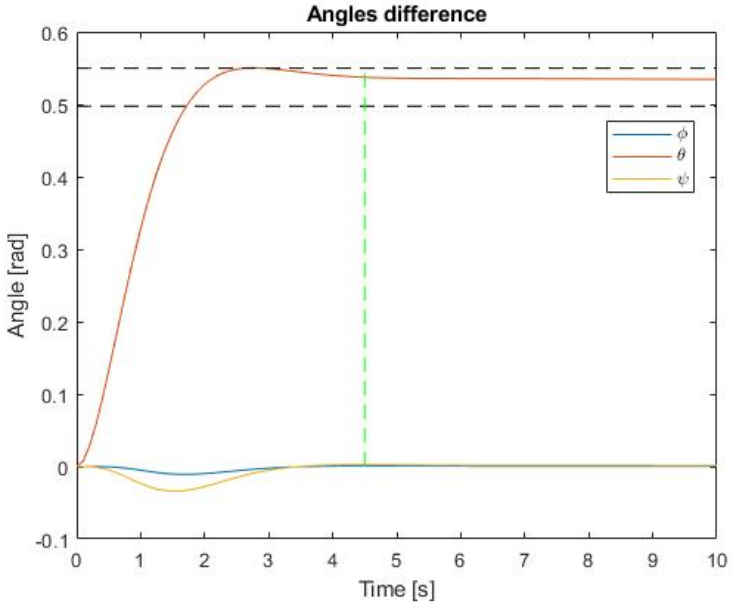



Figure 14: Step response when doing a step in roll angle ($\theta$) from 0 rad to 0.52 rad (30°). The rise time is 1.3 s, the settling time is 3 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

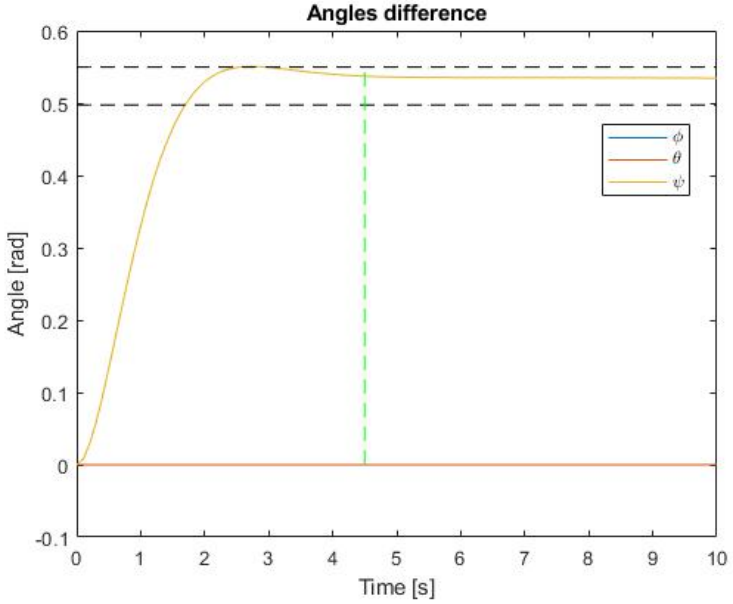



Figure 15: Step response when doing a step in yaw angle ($\psi$) from 0 rad to 0.52 rad (30°). The rise time is 1.3 s, the settling time is 3.6 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

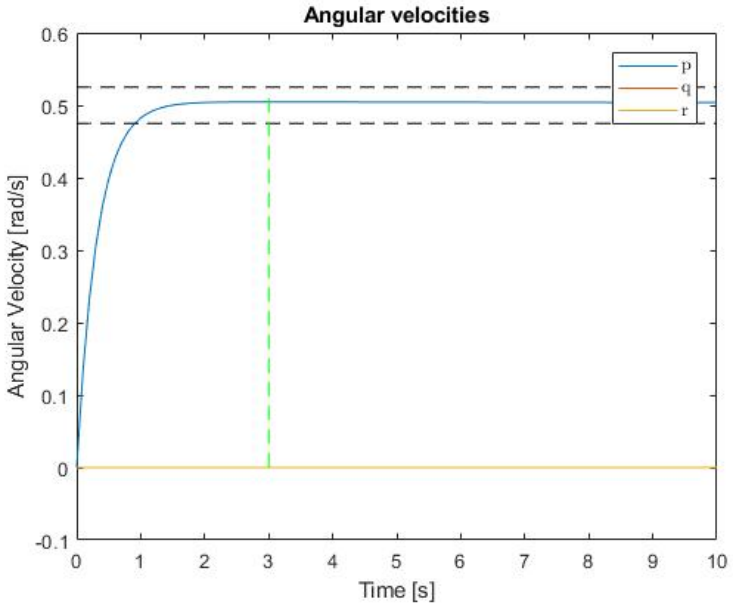



Figure 16: Step response when doing a step in angular velocity about the LCS $x$-axis ($p$) from 0 rad/s to 0.5 rad/s (28.6°/s). The rise time is 0.6 s, the settling time is 1.1 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is about 0 rad/s. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

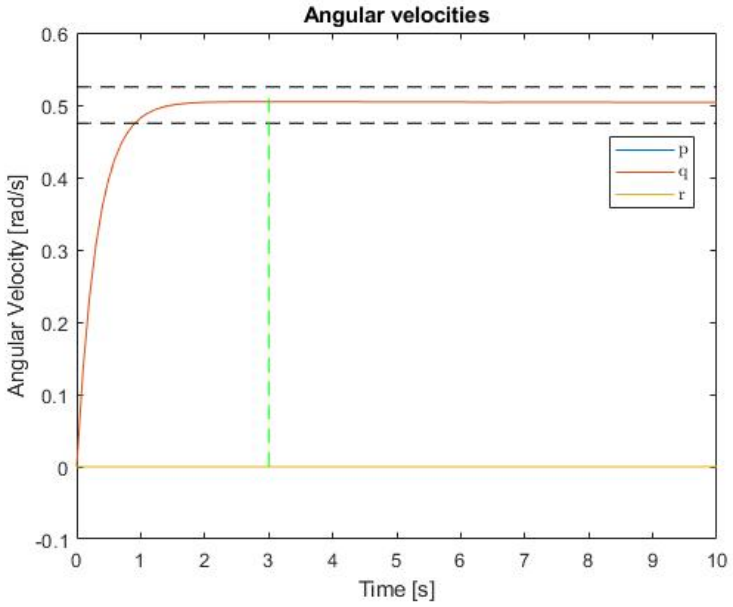



Figure 17: Step response when doing a step in angular velocity about the LCS $y$-axis ($q$) from 0 rad/s to 0.5 rad/s (28.6 °/s). The rise time is 0.6 s, the settling time is 1.1 s with respect to an error less than 5 %, the overshoot is less than 10 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

**Test number:** 37
**Requirement number:** 28, 36, 37, 38, 39
**Test description:** The angular velocity performance about the LCS $z$-axis ($r$) was verified by performing a step in $r$ from 0 rad/s to 0.5 rad/s (28.6°/s) and inspecting the plots with respect to stationary error, rise time, overshoot and settling time.
**Test result:** The step response for the angular velocity about LCS $z$-axis ($r$) can be seen in Figure 18.
**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

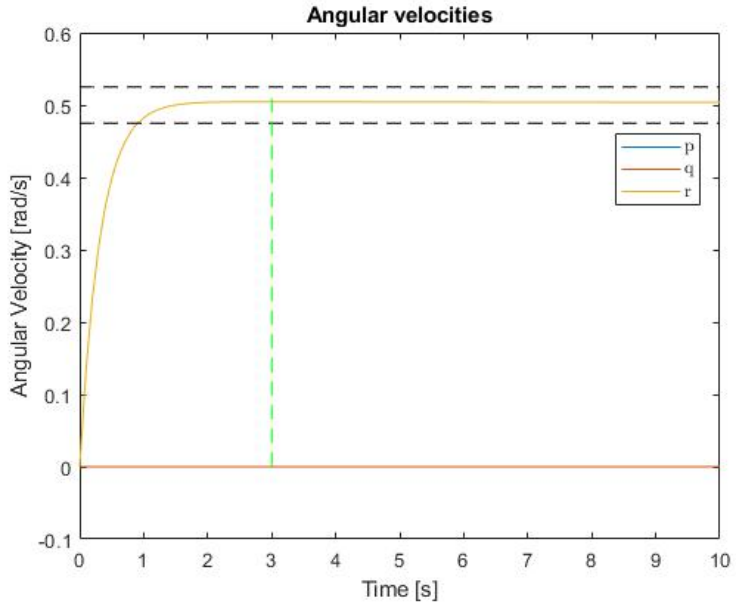



Figure 18: Step response when doing a step in angular velocity about the LCS $z$-axis ($r$) from 0 rad/s to 0.5 rad/s (28.6 °/s). The rise time is 0.6 s, the settling time is 1.1 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is about 0 rad/s.The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

**Test number:** 38
**Requirement number:** 30, 40, 41, 42, 43
**Test description:** To validate the functionality of the motion in PCS $x$-axis, a step in $x$ from 0 m to 1 m was taken and the plots were inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 19.
**Test result:** The step response for the position in PCS $x$-axis can be seen in Figure 19.
**Approved:** The requirements are fulfilled.
**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 39
**Requirement number:** 30, 40, 41, 42, 43
**Test description:** To validate the functionality of the motion in PCS $y$-axis a step in $y$

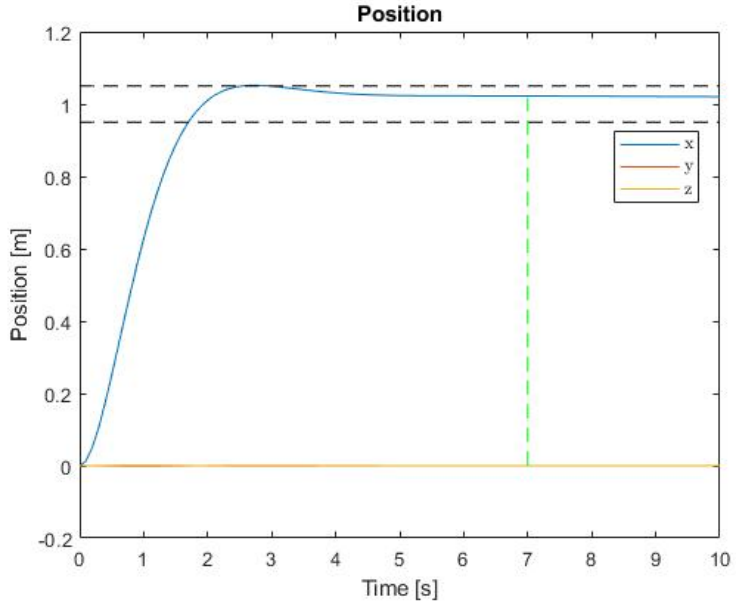



Figure 19: Step response when doing a step in position in the PCS $x$-axis from 0 m to 1 m. The rise time is 1.3 s, the settling time is 3.2 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

from 0 m to 1 m was taken and the plots was inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 20.
**Test result:** The step response for the position in PCS $y$-axis can be seen in Figure 20.
**Approved:** The requirements are fulfilled.
**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 40
**Requirement number:** 30, 40, 41, 42, 43
**Test description:** To validate the functionality of the motion in PCS $z$-axis a step in $x$ from 0 m to 1 m was taken and the plots was inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 21.
**Test result:** The step response for the position in PCS $x$-axis can be seen in Figure 21.
**Approved:** The requirements are fulfilled.
**Executed by:** AS
**Date:** December 05, 2017

**Test number:** 41
**Requirement number:** 29, 44, 45, 46, 47, 48
**Test description:** The linear velocity performance along the LCS $x$-axis ($u$) was verified by performing a step in $u$ from 0 m/s to 0.2 m/s and the plots was inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 22. Tests for requirement 48 are presented in Figures 23 and 24.
**Test result:** The step response for the linear velocity along LCS $x$-axis ($u$) can be seen in Figure 22.

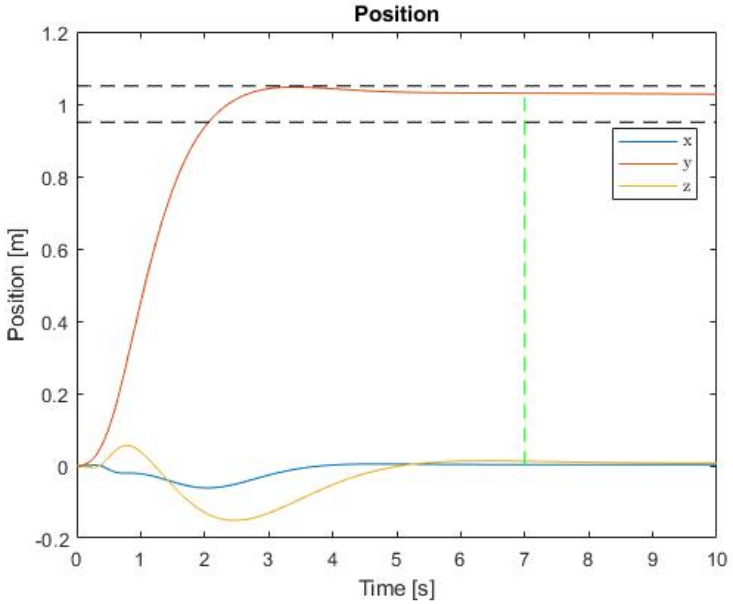



Figure 20: Step response when doing a step in position in the PCS $y$-axis from 0 m to 1 m. The rise time is 1.4 s, the settling time is 3.5 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

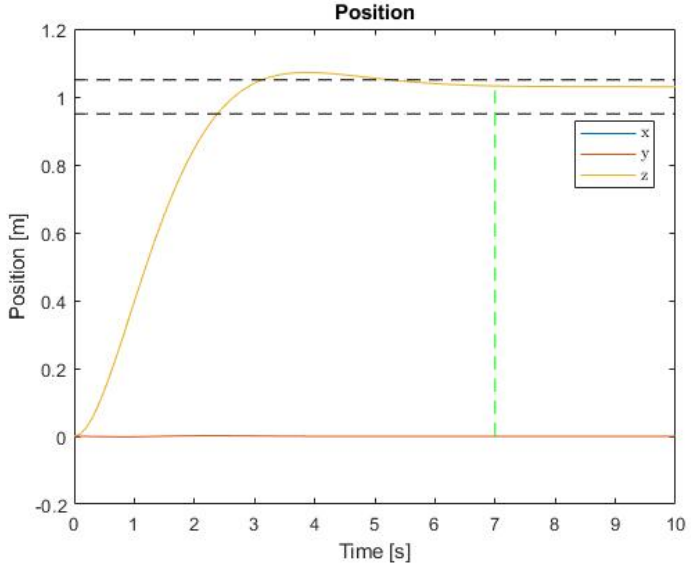



Figure 21: Step response when doing a step in position in the PCS $z$-axis from 0 m to 1 m. The rise time is 1.6 s, the settling time is 5 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is less than 5 %. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

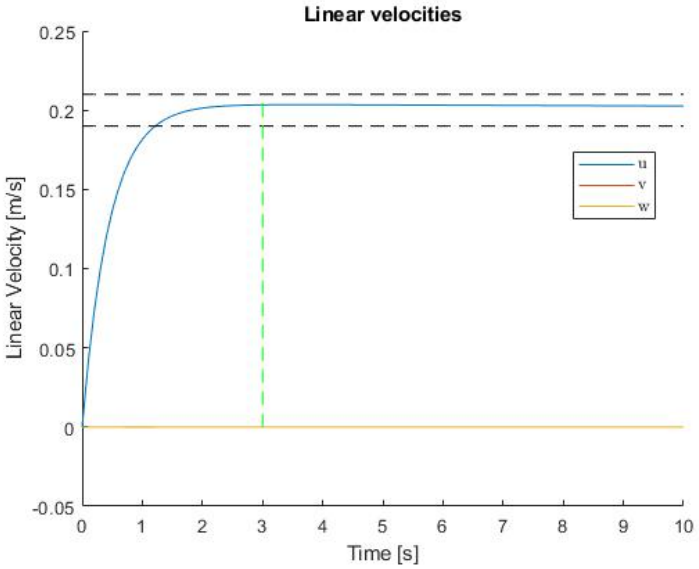



Figure 22: Step response when doing a step in linear velocity along the LCS $x$-axis ($u$) from 0 m/s to 2 m/s. The rise time is 1 s, the settling time is 1.3 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is about 0 rad/s. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

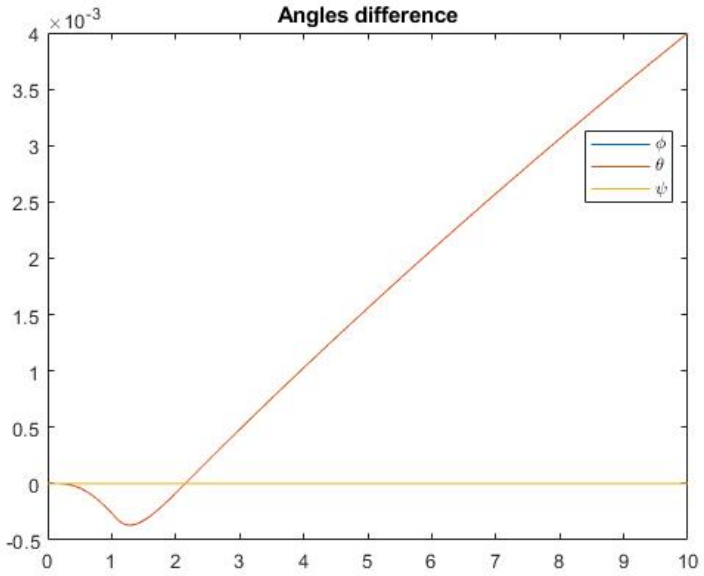



Figure 23: A step in linear velocity along $x$-axis ($u$) does not effect the angles with more than $4 \cdot 10^{-3}$ rad.

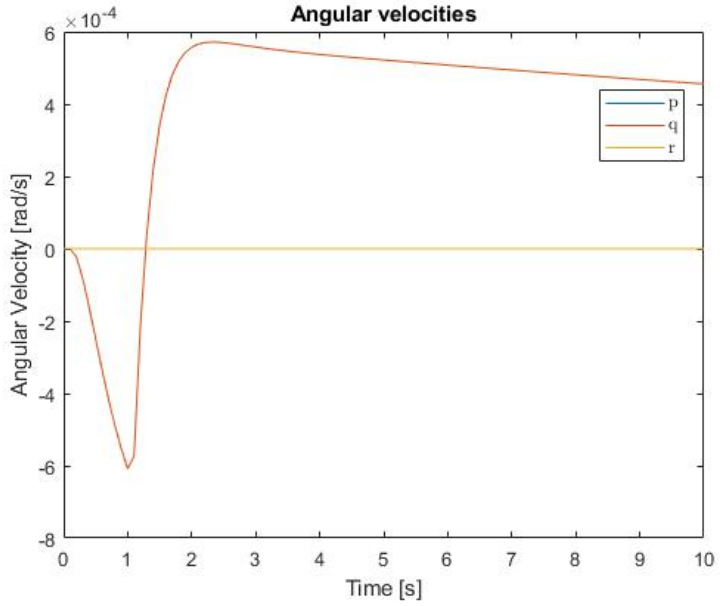



Figure 24: A step in linear velocity along $x$-axis ($u$) does not effect the angular velocities with more than $6 \cdot 10^{-4}$ rad/s.

**Test number:** 42
**Requirement number:** 29, 44, 45, 46, 47, 48
**Test description:** The linear velocity performance along the LCS $y$-axis ($v$) was verified by performing a step in $v$ from 0 m/s to 0.2 m/s and the plots was inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 25. Tests for requirement 48 is presented in Figures 26 and 27.
**Test result:** The step response for the linear velocity along LCS $y$-axis ($v$) can be seen in Figure 25.
**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

**Test number:** 43
**Requirement number:** 29, 44, 45, 46, 47, 48
**Test description:** The linear velocity performance along the LCS $z$-axis ($w$) was verified by performing a step in w from 0 m/s to 0.2 m/s and the plots was inspected with respect to stationary error, rise time, overshoot and settling time, see Figure 28. Tests for requirement 48 is presented in Figures 29 and 30.
**Test result:** The step response for the linear velocity along LCS $z$-axis ($w$) can be seen in Figure 28.
**Approved:** The requirements are fulfilled.
**Executed by:** AS, AA
**Date:** December 05, 2017

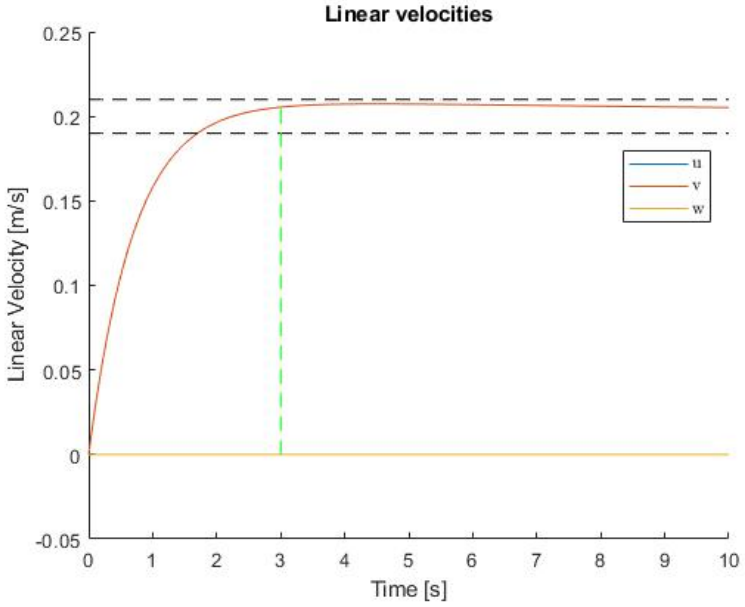



Figure 25: Step response when doing a step in linear velocity along the LCS $y$-axis ($v$) from 0 m/s to 2 m/s. The rise time is 1.3 s, the settling time is 1.5 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is about 0 rad/s. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

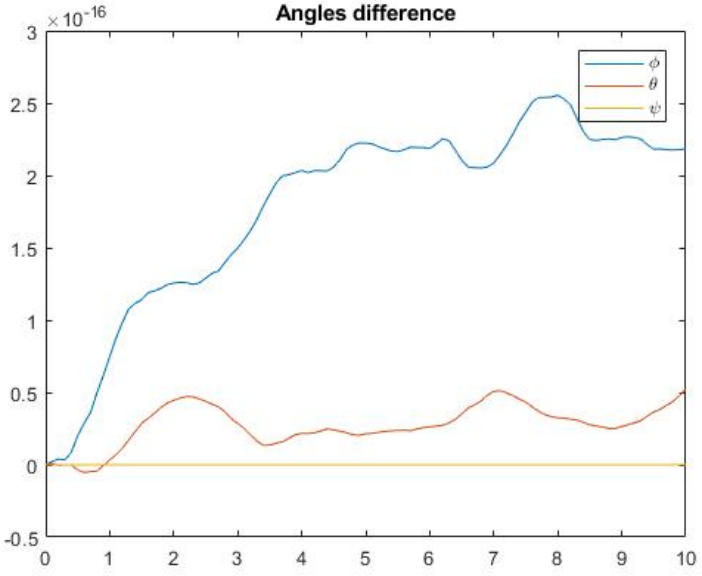



Figure 26: A step in linear velocity along $y$-axis ($v$) does not effect the angles with more than $2.5 \cdot 10^{-16}$ rad.

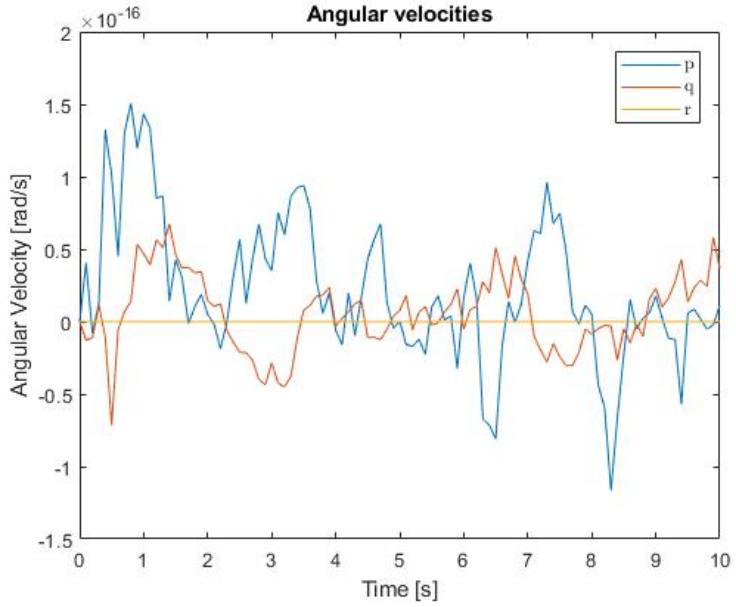



Figure 27: A step in linear velocity along $y$-axis ($v$) does not effect the angular velocities with more than $1.5 \cdot 10^{-16}$ rad/s.

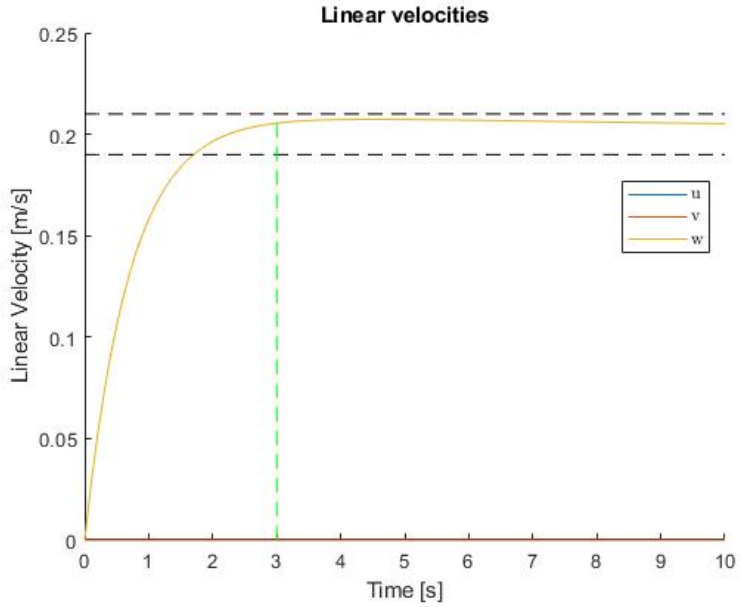



Figure 28: Step response when doing a step in linear velocity along the LCS $z$-axis ($w$) from 0 m/s to 2 m/s. The rise time is 1.5 s, the settling time is 1.6 s with respect to an error less than 5 %, the overshoot is less than 10 %. The stationary error is about 0 rad/s. The black dashed lines represent the range allowed for the step to be within after the settling time is achieved. The green line is the maximum value of settling time to have.

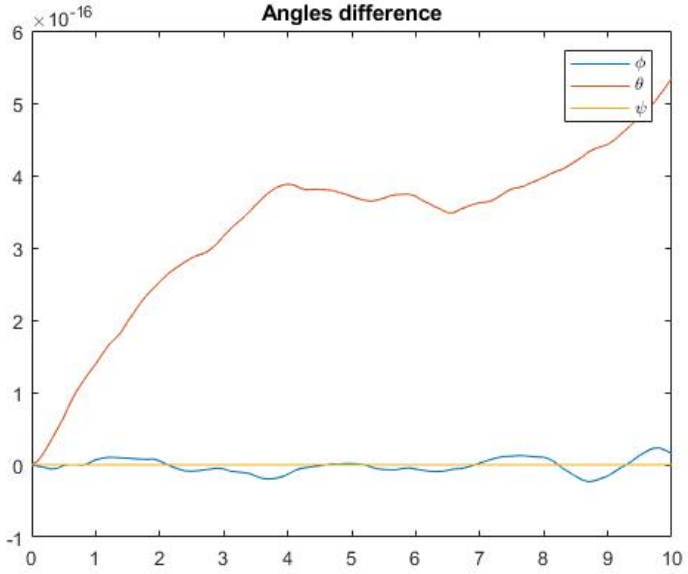



Figure 29: A step in linear velocity along $z$-axis ($w$) does not effect the angles with more than $6 \cdot 10^{-16}$ rad.

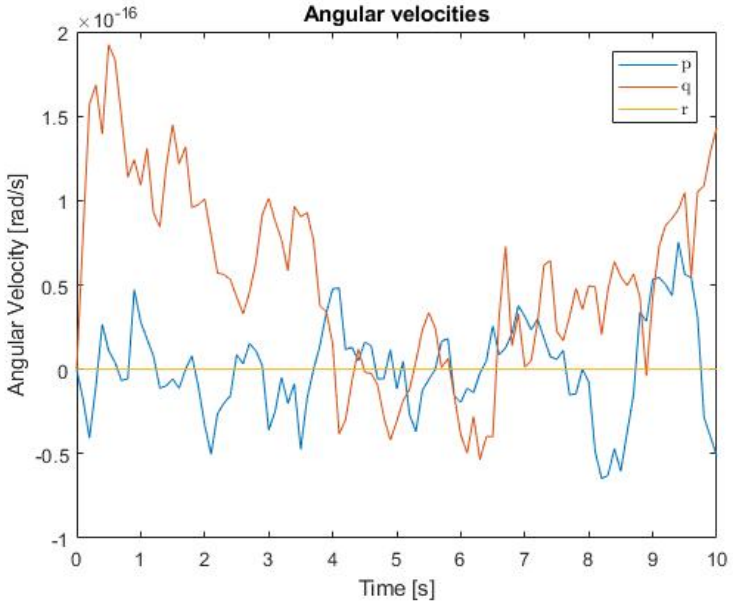



Figure 30: A step in linear velocity along $z$-axis ($w$) does not effect the angular velocities with more than $2 \cdot 10^{-16}$ rad/s.

# 6 Sensor Fusion Tests

**Test number:** 44
**Requirement number:** 52, 56
**Test description:** To verify that the ROV can estimate its position after it has been initialized with a 0.6 m error, a simulation was run where the position states were initialized to 4.0 m in $x$, 4.0 m in $y$ and 2.0 m in $z$. In the sensor fusion the position states were initialized to 3.4 m in $x$, 3.4 m in $y$ and 2.6 m in $z$.
**Test result:** The estimated states compared to the simulated states can be seen in Figure 31. The estimates are initially wrong, but eventually manages to estimates its position.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 8, 2017

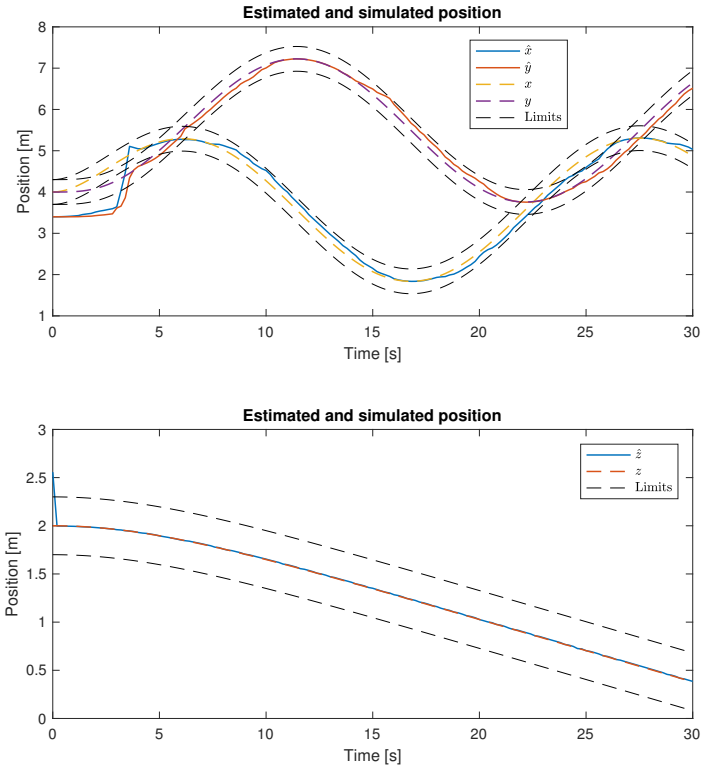



Figure 31: The estimated position states compared to the simulated position states.

**Test number:** 45
**Requirement number:** 53, 57
**Test description:** To verify that the ROV can estimate its attitude after it has been initialized with a 15° error, a simulation was run where all angles were initialized to 0°. In the sensor fusion all angles were initialized to 15°. In the simulation the ROV was moving in a circle.
**Test result:** The estimated angles, $\phi$ and $\theta$, compared to the simulated angles can be

seen in Figure 32 and the difference between the estimated angle, $\psi$, and the simulated angle can be seen in Figure 33. The estimated states quickly reaches the limits of 1° difference and stays inside those limits.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 8, 2017

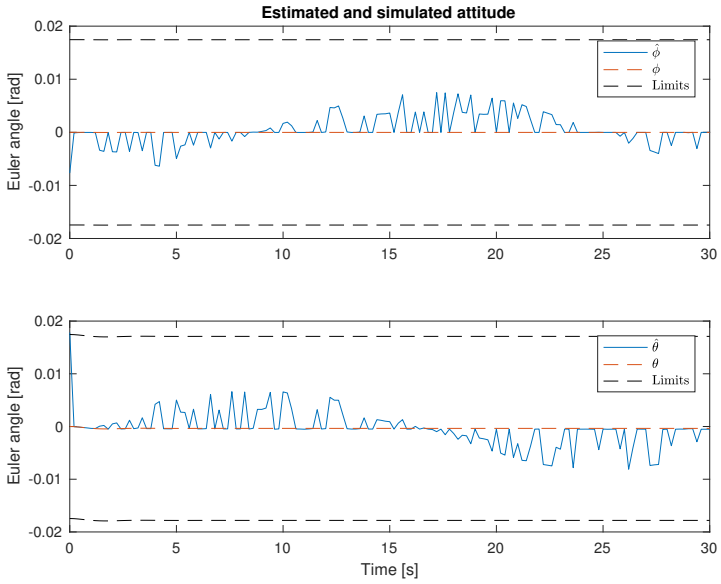



Figure 32: The estimated angles, $\phi$ and $\theta$, compared to the simulated angles.

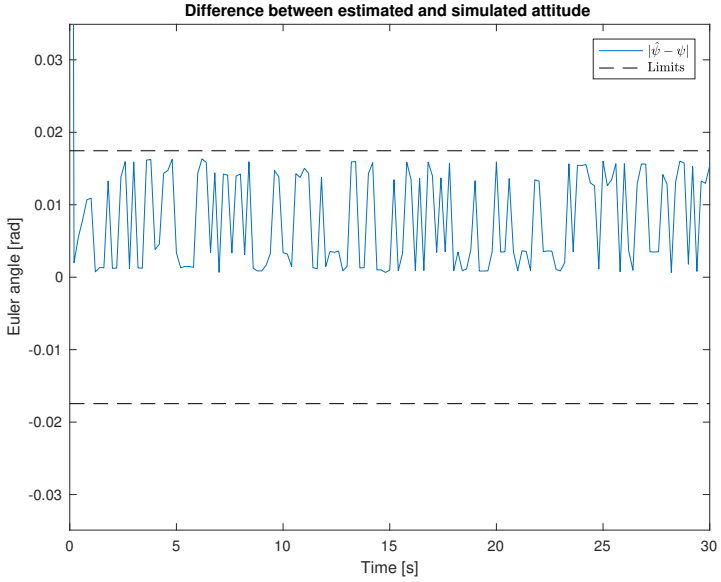



Figure 33: The difference between the estimated angle, $\psi$, and the simulated angle.

**Test number:** 46
**Requirement number:** 52, 58
**Test description:** To verify that the position estimates are maintained within 0.3 m when this limit has been reached, a simulation was run where the position states were initialized to 4.0 m in $x$, 4.0 m in $y$ and 2.0 m in $z$. In the sensor fusion the position states were initialized to 3.4 m in $x$, 3.4 m in $y$ and 2.6 m in $z$. In the simulation the ROV was moving in a circle.
**Test result:** The estimated states compared to the simulated states can be seen in Figure 31. When the estimated states after 4 seconds reaches the limits of 0.3 m from the true position, the estimates stays inside the limits.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 8, 2017

**Test number:** 48
**Requirement number:** 54, 60
**Test description:** To validate the estimates of linear velocities, an XBOX-controller was used to run the ROV with a constant velocity over an interval. The tests were filmed and the tiles in the pool were measured. In that way the distance and time could be measured afterwards and the real velocity was then calculated.
**Test result:** In Figure 34, the estimated value of $u$ can be seen compared to the real value of $u$. The average of the estimated velocity is 0.38 m/s and that is inside the allowed limits. In Figure 35, the estimated value of $v$ can be seen compared to the real value of $v$. The average of the estimated velocity is -0.26 m/s and that is inside the allowed limits. The estimated value of $w$ can be seen in Figure 36 compared to the real value of $w$. The average of the estimated velocity is -0.21 m/s and that is also inside the allowed limits.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** November 29, 2017

**Test number:** 49
**Requirement number:** 55, 61
**Test description:** To validate the estimate of angular velocity in yaw, a simulation was run where the ROV was turning about its $z$-axis with a constant angular velocity.
**Test result:** In Figure 37 the estimated value of $r$ can be seen compared to the simulated value of $r$. The estimate is inside the allowed limits at all times.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 7, 2017

**Test number:** 50
**Requirement number:** 55, 61
**Test description:** To validate the estimates of angular velocities in roll and pitch, a simulation was run where the ROV was making a step from -25° to 25° about the $x$-axis and the angular velocity corresponding to when $\phi$ was between -20° and 20° was used to verify the performance of the estimate of $p$. Another simulation was run where the ROV was making a step from -40° to 40° about the $y$-axis and the angular velocity corresponding to when $\theta$ was between -30° and 30° was used to verify the performance of the estimate of $q$.
**Test result:** The result of the estimated values of $p$ and $q$ from the simulations can be seen in Figure 38 and Figure 40. The part of the actual step can be seen in Figure 39 and

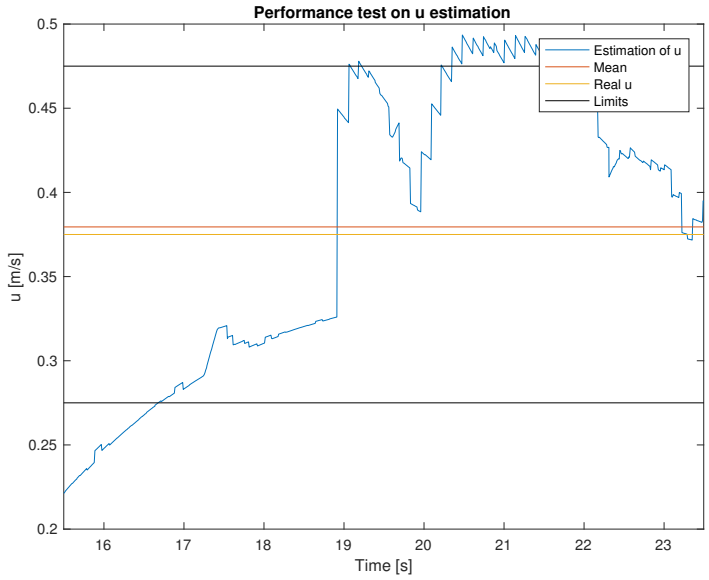



Figure 34: The estimated state of $u$ compared to the real $u$. The behaviour around 19 seconds might be explained by fact that the test were performed with manual control of the thrusters.

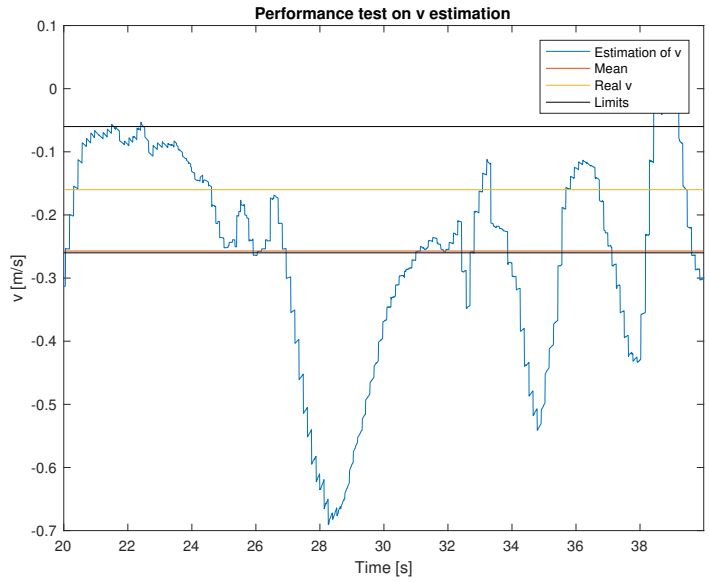



Figure 35: The estimated state of $v$ compared to the real $v$.

41.
**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 3, 2017

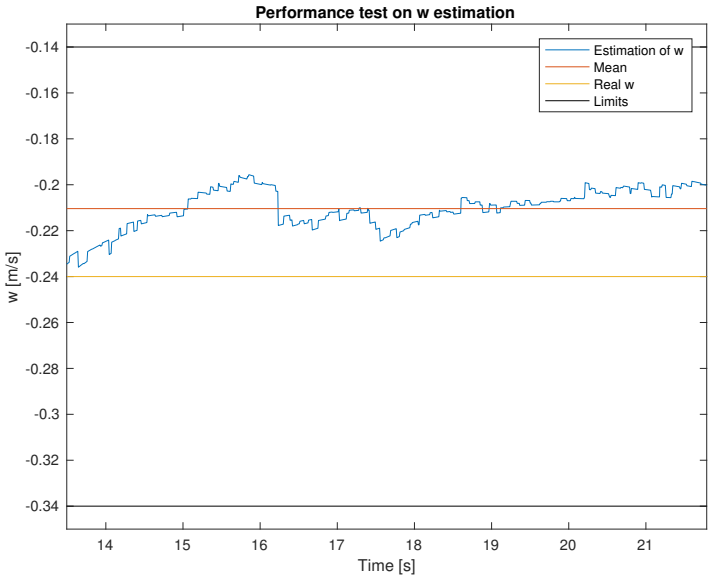



Figure 36: The estimated state of $w$ compared to the real $w$.

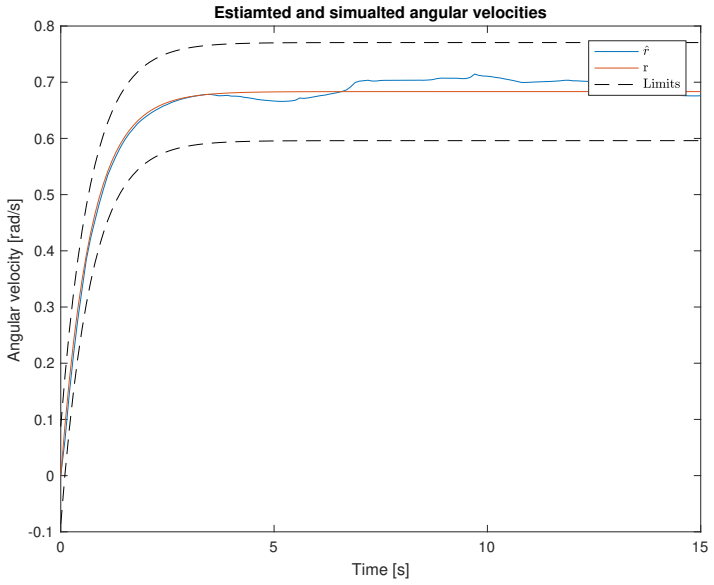



Figure 37: The estimated state of $r$ compared to the real $r$.

**Test number:** 51
**Requirement number:** 53, 62
**Test description:** To validate the estimates of the attitude, a simulation was run where the ROV was standing still and tilting its roll angle to approximately 50°. Another simulation was run where the ROV was standing still and tilting its pitch angle to approximately 50°. To verify the estimation of the yaw angle, a simulation was run where the ROV was standing still and rotating with constant angular velocity about its $z$-axis.

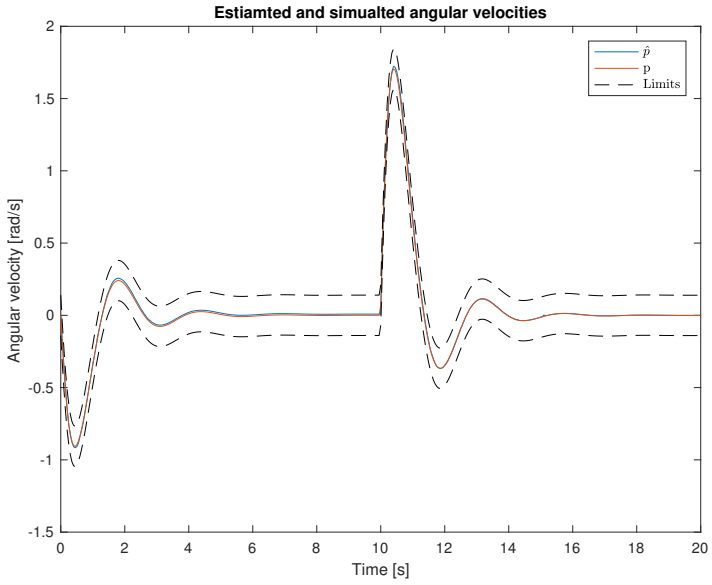



Figure 38: The estimated state of $p$ compared to the real $p$.

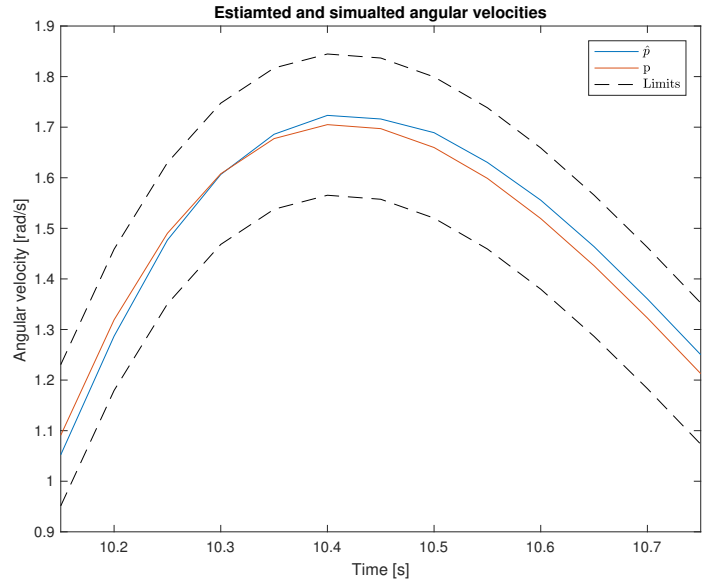



Figure 39: The estimated state of $p$ compared to the real $p$, zoomed in on the part where the step is taken.

**Test result:** The result of the estimated value of $\phi$ from the simulation can be seen in Figure 42 compared to the simulated value. The maximum difference is 0.17°. The result of the estimated value of $\theta$ from the simulation can be seen in Figure 43 compared to the simulated value. The maximum difference is 0.29°. The result of the estimated value of $\psi$ from the simulation can be seen in Figure 44 compared to the simulated value. The maximum difference is 0.14°.

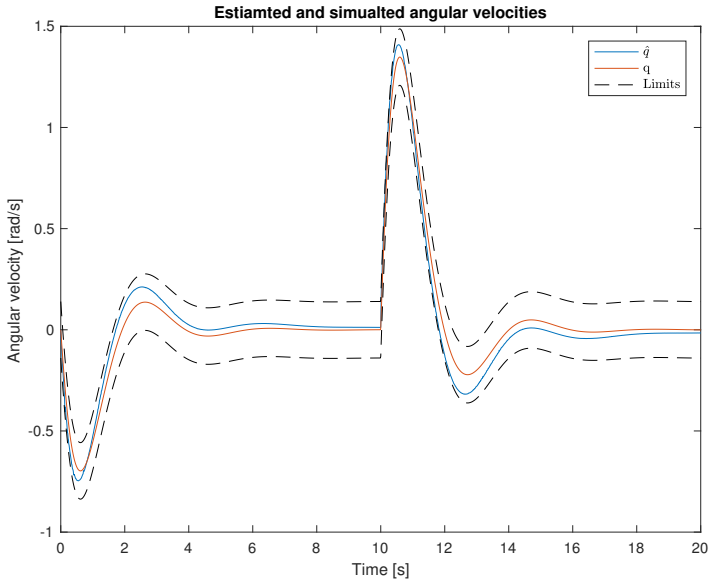



Figure 40: The estimated state of $q$ compared to the real $q$.

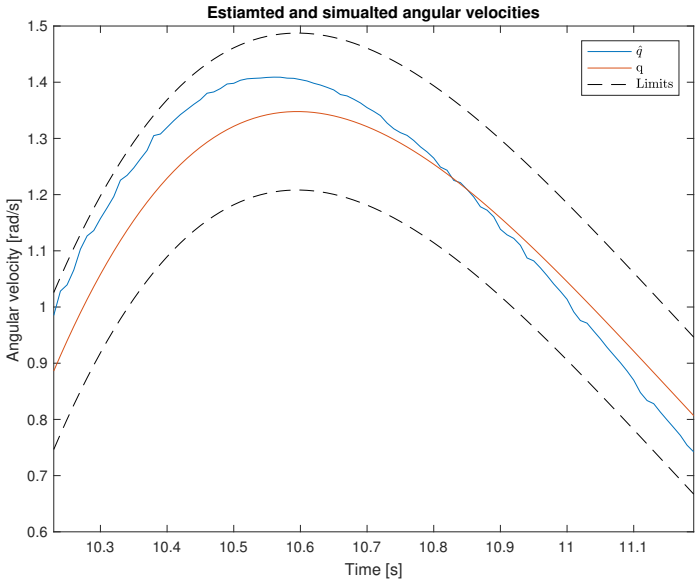



Figure 41: The estimated state of $q$ compared to the real $q$, zoomed in on the part where the step is taken.

**Approved:** The requirements are fulfilled.
**Executed by:** MH, FN
**Date:** December 1, 2017

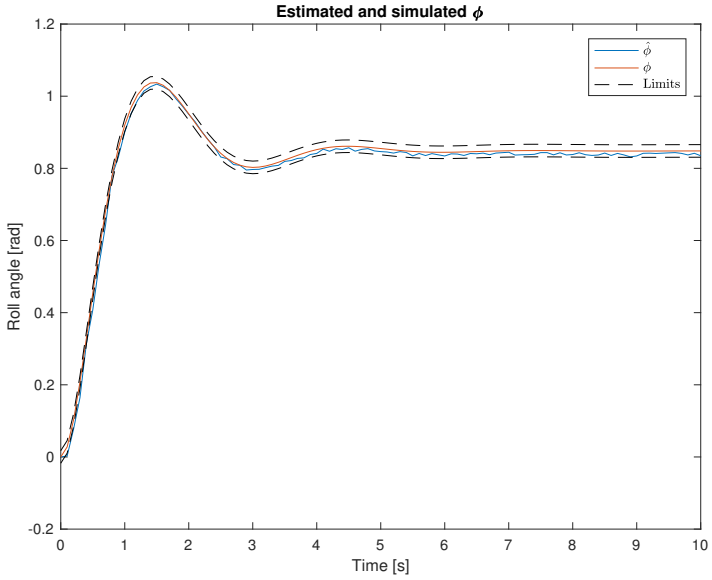



Figure 42: The estimated state of $\phi$ compared to the real $\phi$.

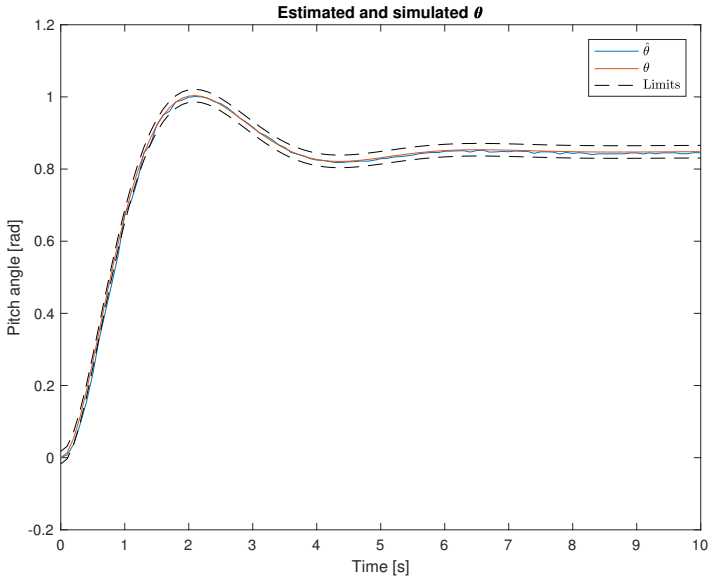



Figure 43: The estimated state of $\theta$ compared to the simulated $\theta$.

# 7 Modelling Tests

**Test number:** 53
**Requirement number:** 66, 68, 69
**Test description:** Validation data was collected by running a telegraph test for each of the angular DOF's. The validation data was then compared to the estimated model to verify that the model fit the data to at least 75 % in each DOF.

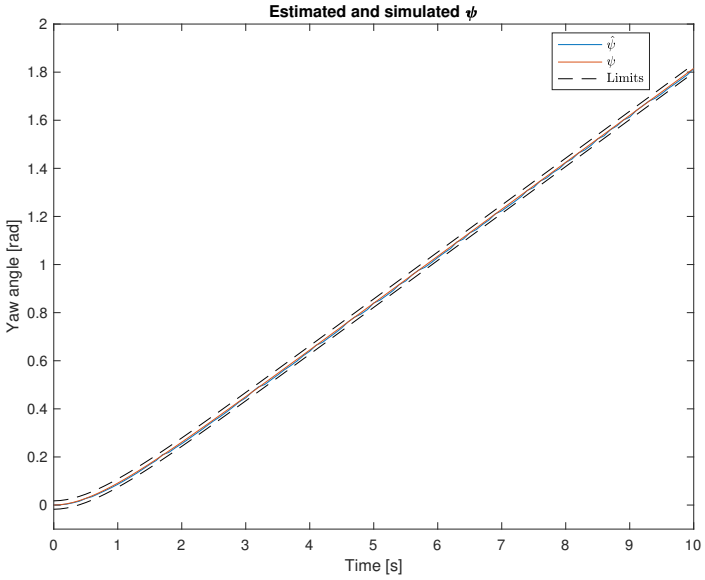



Figure 44: The estimated state of $\psi$ compared to the simulated $\psi$.

**Test result:** The result of the model excited in roll can be seen in Figure 45 compared to the validation data. The model fit the data to 83.21 % and 76.10 %. In Figure 46 the result of the model compared to validation data can be seen when the ROV was excited in pitch. The model fit the data to 80.77 % and 88.56 %. In Figure 47 the result of the model compared to validation data can be seen when the ROV was excited in yaw. The model fit the data to 78.52 %.
**Approved:** The requirements are fulfilled.
**Executed by:** MH
**Date:** November 24, 2017

**Test number:** 54
**Requirement number:** 65, 67, 70
**Test description:** Validation data was collected by running a telegraph test for each of the linear DOF's. Outliers in the sonar measurements were removed by replacing these measurements with an average of the measurements before and after the outliers. The validation data was then compared to the estimated model to verify that the model fit the data to at least 80 % in each DOF.
**Test result:** The result of the model excited in $x$ can be seen in Figure 48 compared to the validation data. The model fit the data to 81.41 %. In Figure 49 the result of the model compared to validation data can be seen when the ROV was excited in $y$. The model fit the data to 89.51 %. In Figure 50 the result of the model compared to validation data can be seen when the ROV was excited in $z$. The model fit the data to 93.49 %.
**Approved:** The requirements are fulfilled.
**Executed by:** MH
**Date:** November 24, 2017

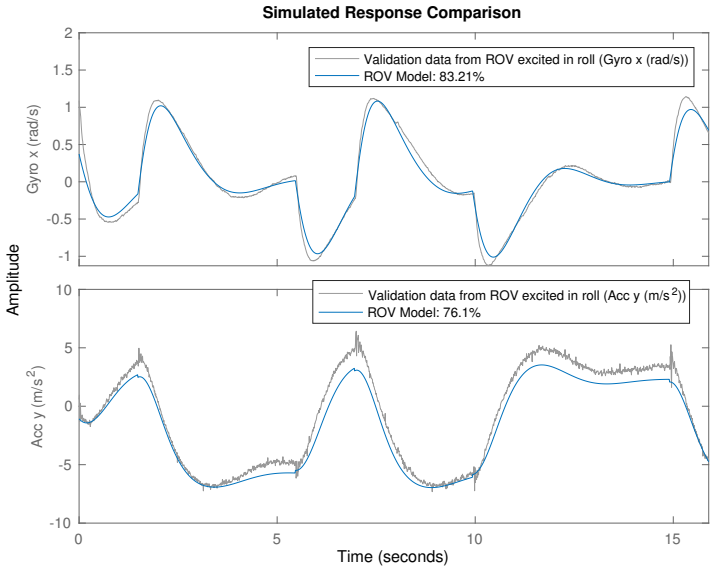



Figure 45: Validation of model excited in roll.

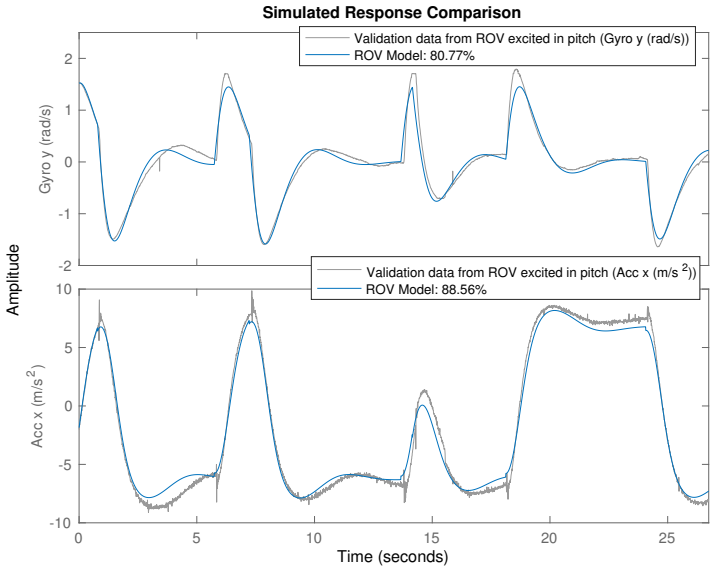



Figure 46: Validation of model excited in pitch.

# 8 Simulation Tests

**Test number:** 55
**Requirement number:** 73
**Test description:** The test was done to verify that the initial conditions set were realized by the simulator as initial states. The test was done by setting the initial $xyz$-coordinates to (5,6,3) and running the simulator without any control signals. The resulting plot of

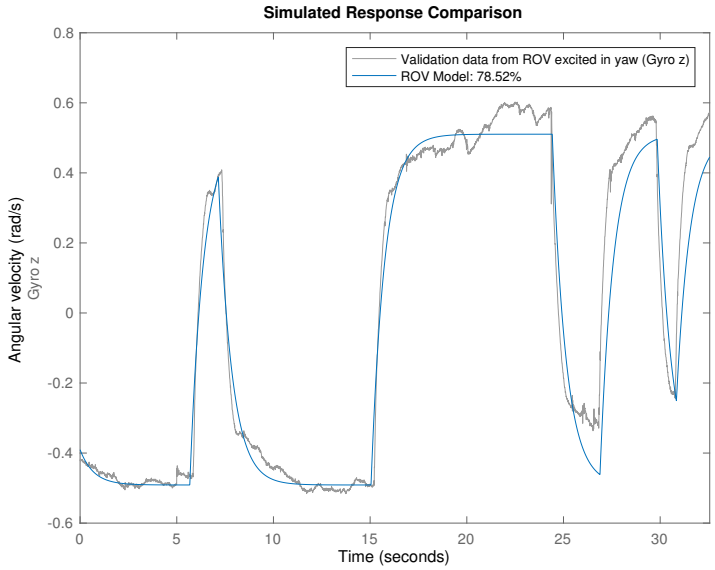



Figure 47: Validation of model excited in yaw.

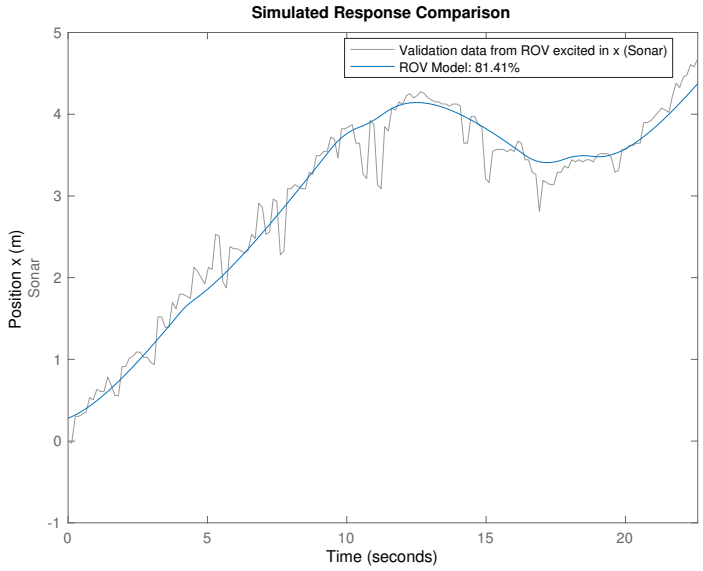



Figure 48: Validation of model excited in $x$.

the ROV location should then be constant at the initial set coordinates in X and Y and slowly diminishing in Z since the ROV is rising slowly when submerged.
**Test result:** The resulting plot of the position of the ROV is shown in Figure 51 where the resulting position are shown to be the same as initially entered.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** November 07, 2017

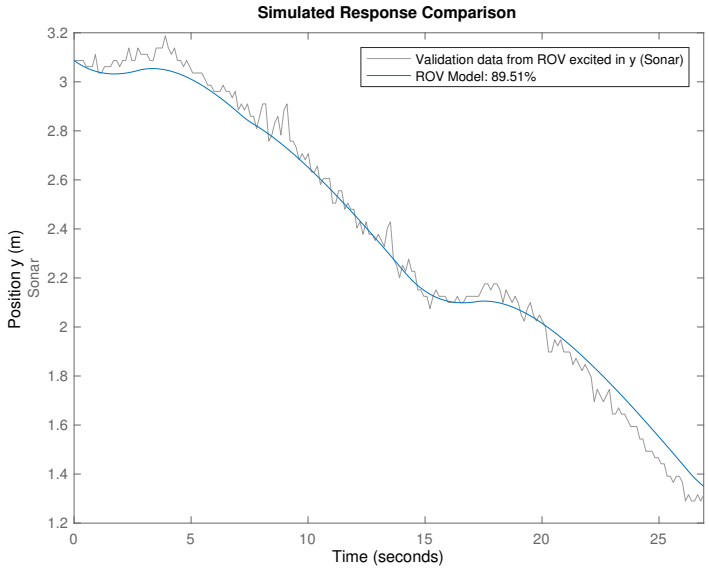



Figure 49: Validation of model excited in $y$.

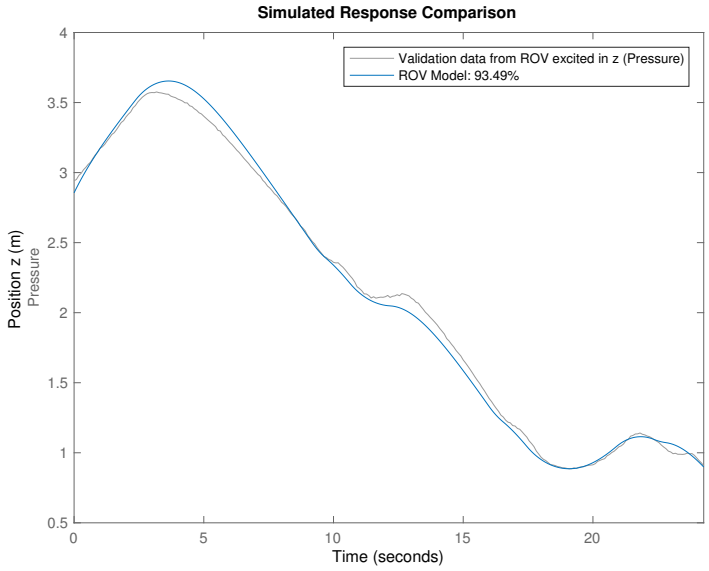



Figure 50: Validation of model excited in $z$.

**Test number:** 56
**Requirement number:** 74, 75, 76, 77
**Test description:** The test was performed to verify that the GUI for the simulator can show the angular/linear velocities and the attitude of the ROV, and that there exist a 3D visualization of the ROV during simulation. The test was done by performing a simulation with a linear movement along the $x$-axis of the pool while not rotating the ROV thus the

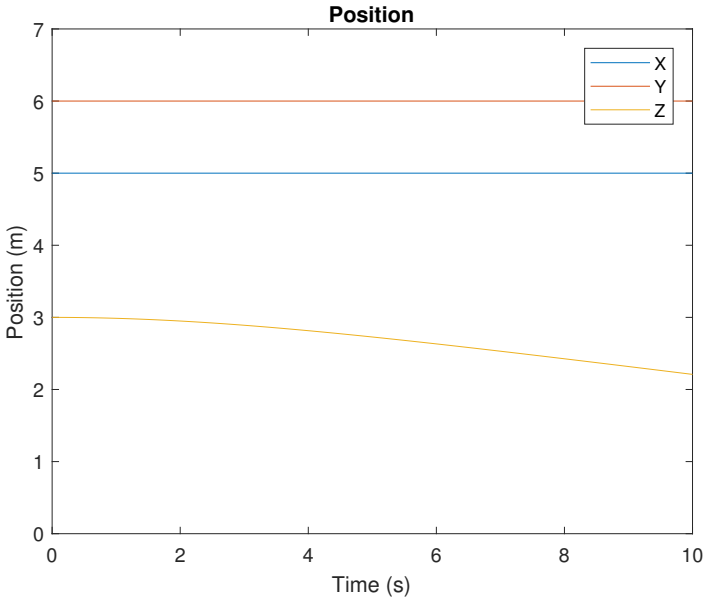
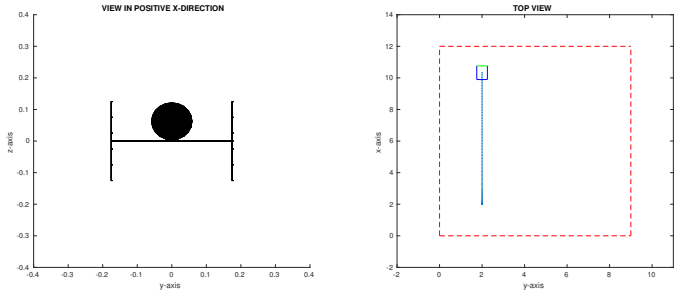



Figure 51: Plot of the position of the ROV.

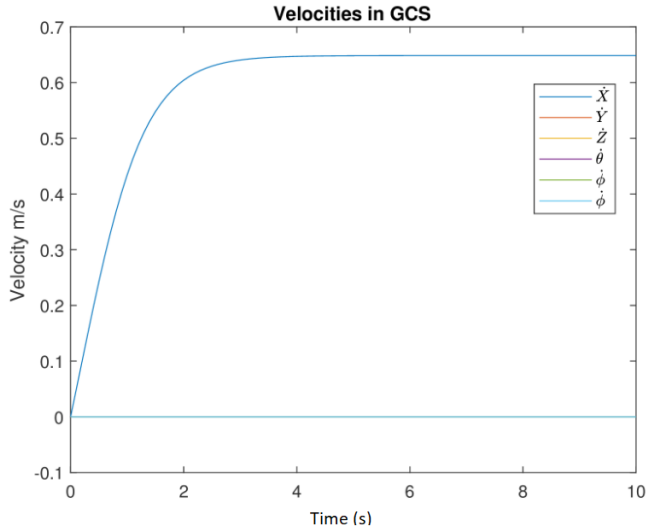
angular velocities should be zero.
**Test result:** The resulting visualization is shown in Figure 52 where the trace of the ROV's path can be seen in Figure 52a and the velocities during the movement are shown in Figure 52b.
**Approved:** The requirements are fulfilled.
**Executed by:** AN
**Date:** November 07, 2017

**Test number:** 57
**Requirement number:** 83,91
**Test description:** The test was performed to verify that the simulated states from the simulator are estimated within 85% of the sensor fusion estimates when using the same control signals. The test used logged data from a one minute run with the XBOX-controller. The simulation and sensor fusion both used this real data as input and creating simulated states from the simulator and estimated states from the sensor fusion and comparing them in the same plots.
**Test result:** The resulting plots are shown in Figure 53, were the simulated and estimated Euler angles are compared, and in Figure 54 were the simulated and estimated linear velocities are shown. It is noted that the simulated data is not even close to 85% fit for the Euler angles compared to the estimated data. The velocities are much better and the worst fit is $R_w^2 = 61\%$. Also noted is that during steady state the level is almost never within the limits but it converges during translation.
**Approved:** The requirements are not fulfilled.
**Executed by:** AN
**Date:** December 05, 2017

**Test number:** 61
**Requirement number:** 87
**Test description:** To verify that the simulator is able to store and display the simulated




(a) In the left figure, the ROV attitude is visualized in pool $x$-direction. In the right figure, the ROV's position and yaw angle is visualized in a 12 x 9 m pool.




(b) The angular and linear velocities of the ROV during the simulated run.

Figure 52: Visualization of the ROV's path and its angular/linear velocities during said path.

data, a Matlab script was used to initialize and run the simulation. The script contained initial states as well as the control signals to the thrusters. The control signals for the test were a constant input in $\tau(r)$ for movement in yaw. The Matlab script then runs the simulation, store and plot the logged states.
**Test result:** The states are logged and can be displayed as intended, one example of logged data is shown in Figure 55 where the simulated magnetometer data is displayed.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** November 07, 2017

**Test number:** 63
**Requirement number:** 89
**Test description:** This test is done to verify that the regulator module can produce thruster signals when running a SIL simulation. To verify this, a simulation is done where the control module is running in c++ and the simulator reads the control signals from

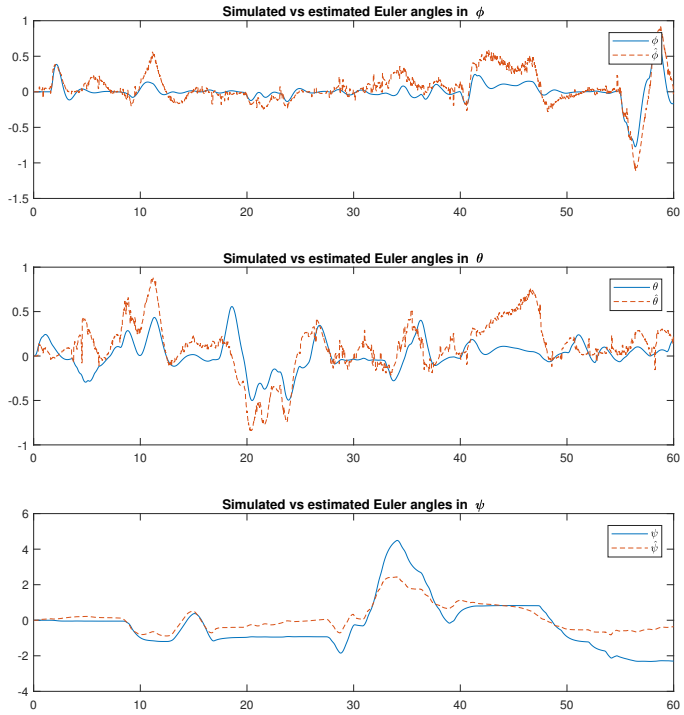



Figure 53: Simulated and estimated Euler angles sampled from one minute of XBOX-controller data

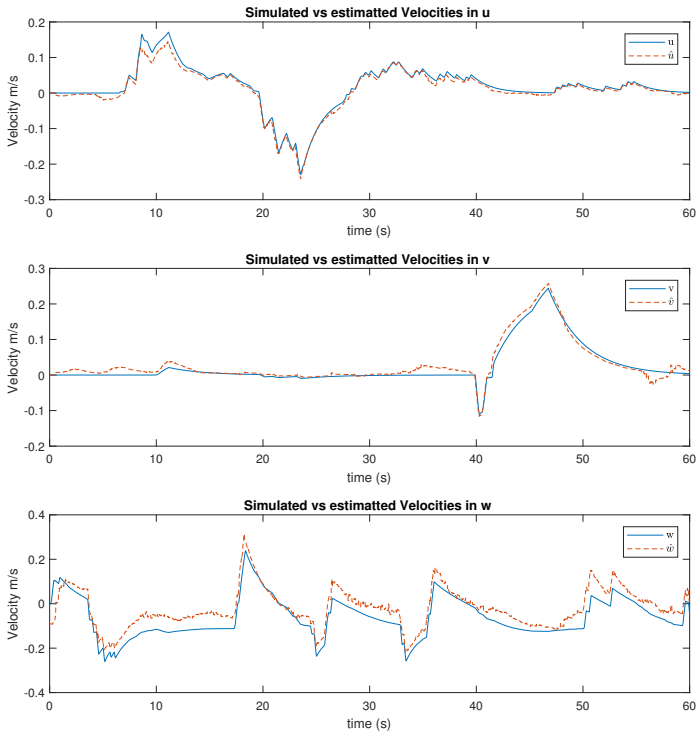



Figure 54: Simulated and estimated linear velocities in LCS sampled from one minute of XBOX-controller data

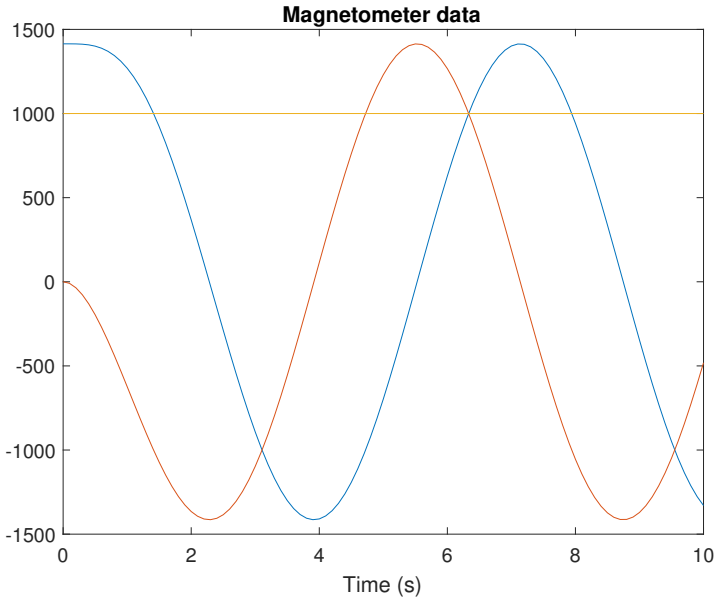



Figure 55: Magnetometer reading during a simulated motion in yaw ($\psi$).

the corresponding ROS topic. The simulator was running on OSX while the sensor fusion and the controller module was running via a Linux virtual machine.
**Test result:** The result when taking a step from 0 m/s to 0.2 m/s can be seen in Figures 56 and 57.
**Approved:** The requirement is fulfilled.
**Executed by:** AN
**Date:** November 07, 2017

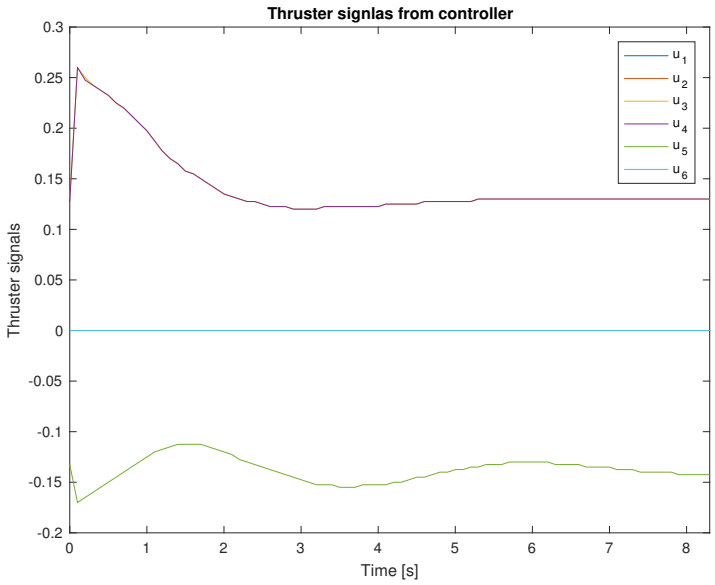



Figure 56: Thruster signals from SIL simulation of the controller.

**Test number:** 64
**Requirement number:** 90
**Test description:** To verify that the sensor fusion module could produce state estimates when running a SIL simulation, a test when the ROV was traveling in different direction were done. The simulator was running on OSX while the sensor fusion module was running via a Linux virtual machine.
**Test result:** Figure 58 and Figure 59 show the state estimates published from the sensor fusion module.
**Approved:** The requirement is fulfilled.
**Executed by:** FN
**Date:** November 09, 2017

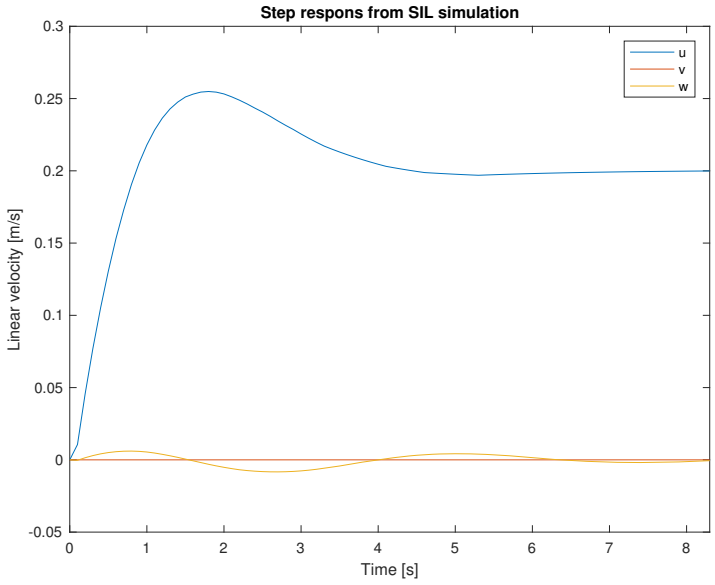



Figure 57: Linear velocity when running a SIL simulation.

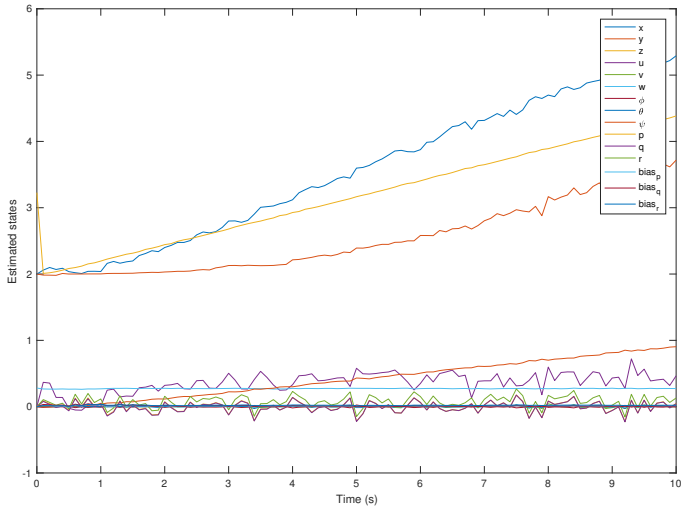



Figure 58: Sensor fusion estimates when running a SIL simulation. The ROV was excited in the $x$- and yaw-direction.

# 9 Hardware Tests

**Test number:** 65
**Requirement number:** 100
**Test description:** To test that the ROV is waterproof, the ROV was placed at the bottom of a pool at Ljungsbro bathhouse. The depth of the pool is 3.8 meters, see Figure 60b. The ROV was placed there for 2 minutes and then visually inspected.
**Test result:** After the ROV was placed at the bottom it was visually inspected and no

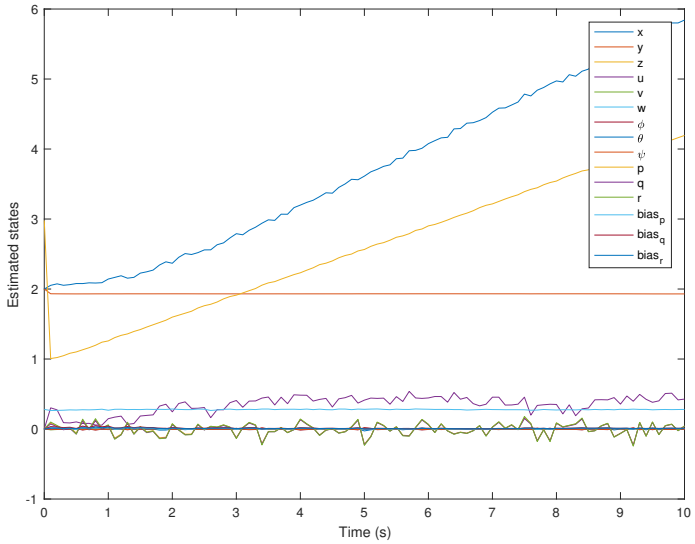
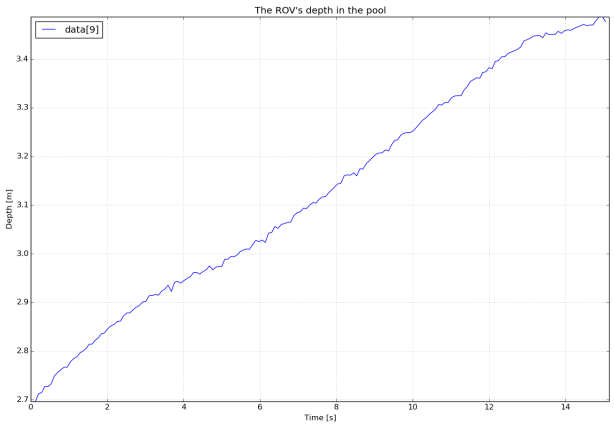
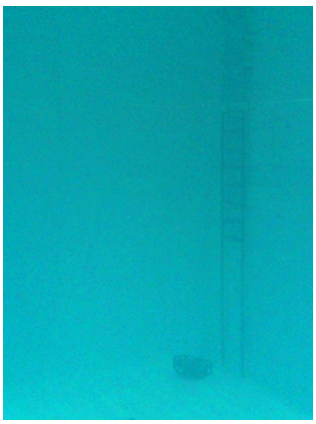



Figure 59: Sensor fusion estimates when running a SIL simulation. The ROV was excited in the $x$- and $z$-direction.

sights of water was found in the acrylic tube or in the sonar tubes. In Figure 60a the maximum depth is 3.5 meters, the depth measurement was calibrated when the top of the ROV was at the water surface. Thereby the maximum depth of the center of the ROV can be increased by 0.1 meters. The other 0.2 meters that is missing to achieve 3.8 meters is due to that the pressure sensor is 0.2 meters above the bottom surface when the ROV is standing at the bottom.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** November 06, 2017




(a) Plot of the ROV depth.



(b) The ROV shown at 3.8 meters depth in Ljungsbro bathhouse.

Figure 60: Tests to verify that the ROV is waterproof.

**Test number:** 66
**Requirement number:** 101
**Test description:** A test to verify battery replacement can be done in a maximum time of 30 minutes. This test was done by opening the acrylic tube and disconnecting the current battery and then connecting the new battery.
**Test result:** The battery change was done in less that 10 minutes.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** November 06, 2017

**Test number:** 67
**Requirement number:** 102
**Test description:** A test to verify that it was possible to physically access the hardware inside the acrylic tube. The hardware inside the acrylic tube was accessed by opening the lid and removing the acrylic tube.
**Test result:** It is possible to physically access the hardware inside the acrylic tube.
**Approved:** The requirement is fulfilled.
**Executed by:** MJ
**Date:** November 06, 2017

# 10 Pathfinder Tests

**Test number:** 68
**Requirement number:** 108, 110
**Test description:** Start and end position for the path is set in the GUI and publish as ROS messages. The planned path is plotted on the map and can be inspected. The obstacles are displayed with a dark grey area around them with a margin of 0.5 meters to ensure that the ROV would not collide with them.
**Test result:** Figures 61 and 62 show a path that is generated from the input given by the GUI and the path does not collide with any obstacles or walls.
**Approved:** The requirements are fulfilled.
**Executed by:** MM
**Date:** November 08, 2017

# 11 Security Tests

**Test number:** 70
**Requirement number:** 115
**Test description:** A test to verify that the ROV detects a lost connection to the workstation. The ROV was set in forward motion and the ethernet cable was disconnected from the workstation. The logged data was inspected and the movement of the ROV filmed.
**Test result:** When the ethernet cable was disconnected the subscribing and publishing of topics from the ROV stopped. The ROV stopped its motion when disconnected.
**Approved:** The requirement is fulfilled.
**Executed by:** AA
**Date:** November 06, 2017

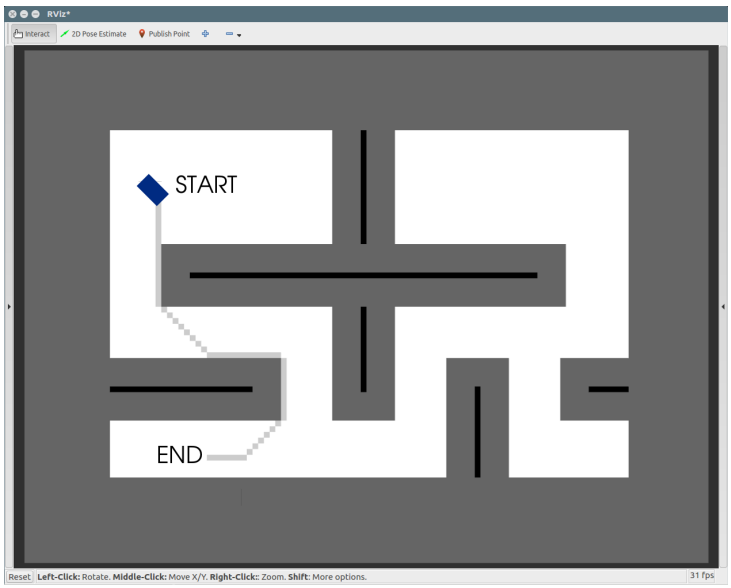



Figure 61: Only the visualization part of the GUI is presented in this figure. The black parts of the map are obstacles, the dark grey parts of the map are areas that are close to obstacles that the ROV should try to avoid and the light grey path is the generated path.

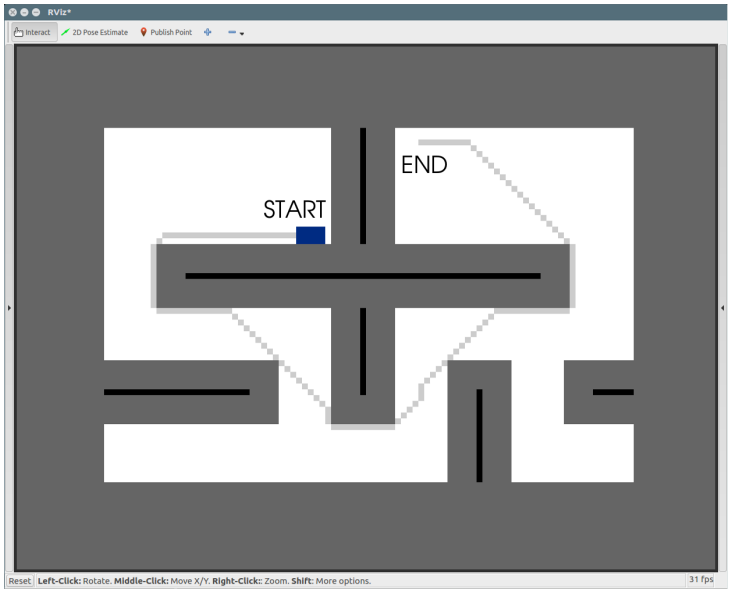



Figure 62: Only the visualization part of the GUI is presented in this figure. The black parts of the map are obstacles, the dark grey parts of the map are areas that are close to obstacles that ROV should try to avoid and the light grey path is the generated path.

# References

[1] Marcus Homelius. *Design Specification, Remotely Operated Underwater Vehicle*. 2017.

[2] Marcus Homelius. *Requirement Specification, Remotely Operated Underwater Vehicle*. 2017.

[3] Alaa Saeed. *Test Plan, Remotely Operated Underwater Vehicle*. 2017.