Design of Dynamic Positioning System for ROV Minerva

Carl Christoffer Lysdahl

June 18, 2010

NTNU – Trondheim
Norwegian University of
Science and Technology

1 Summary

This thesis focuses on the design and implementation of a DP system for the ROV Minerva. It features the implementation of receiving and sending signals and processing these. The thesis also includes the creation of controllers and thrust allocation. The making of a Human Machine Interface with a Grapichal User Interface is also done.

Fullscale seatrials of the system was done to test the viability of the design. The results where promising, but the systems needs more work and testing.

Contents

+

List of Figures

List of Tables

2 Introduction

This master thesis is a continuation of the work done in my project thesis: *Control System Architecture of the ROV Minerva*. This report presents the work done at my 10th semester of my Masters Degree in Marine Cybernetis at the Norwegian University of Science and Technology.

2.1 Background

2.1.1 ROV

A Remotely Operated Vehicle (ROV) is a tethered underwater robot. They are used for many underwater tasks, especially in deepwater. ROVs are controlled by an operator onboard a ship by the connection to the ROV called an umbilical (or tether). ROVs are very manouverable and usually carry camerae s, lights and manipulators. Other equipment found on ROVs may include sonars, water samplers and other measurement instruments.

2.1.2 Dynamic Positioning

Dynamic Positioning (DP) is a computer controlled system to automatically maintain the position and heading of a vessel. They can also be made to control other degrees of freedom including heave. DP uses position reference sensors combined with measurements, of for example wind, to calculate the necessary force and direction. This force is then converted to the required thrust and angle of the actuators.

2.2 Contributions

I would like to thank advisors Martin Ludvigsen, Fredrik Dukan and Professor Asgeir Sørensen for their assistance, help and guidance. I would also like to thank Marianne Kirkeby for her work and cooperation.

2.3 Master Thesis Objectives

The long term project objectives are to create a viable DP system for Minerva. The project objectives for this master thesis are:

- To propose a requirement specification and corresponding technical specification for DP system for the ROV Minerva.

- To propose and implement signal interfaces between sensors/instruments, ROV actuators and the ROV DP control system.
- To create a Human Machine Interface (HMI) so the regulator computer will be easily controlled.
- To design and implement a DP system for the ROV Minerva.
- To design and implement a graphical user interface (GUI) for the DP system.
- To propose a brief plan for full scale implementation and test of the DP system at the end of May/June.
- To conduct full scale sea trials in May/June.

3 Technical Specification of NTNU ROV - Minerva

3.1 Requirement Specification

The operating of Minerva requires quite a bit of consentration. The joystick control panel directly control the RPM of the thrusters and it is a challenging and tideous job to operate over longer periodes of time. The implementation of a Dynamic Positioning system will let the operator rest, and if sufficiently succesful, it should be able to hold the position more accuratly than an operator.

The overall requirements for a DP system is that is should be reliable and accurate. The system may also need to be operated by for instance research marine biologists. The system should then also be easy to use, even though a ROV operator would need to be present.

The system needs to be able to hold position in surge, sway, yaw and heave and be able to work under all depths and currents.

Figure 1: The NTNU ROV-Minerva.

3.2 Technical Specification

3.2.1 Minerva Specification

Minerva is a SUB-Fighter 7500 from Sperre AS. Minerva, as seen in figure 1, is currently only operated with an operator using two joysticks. The operator also has the ability to use auto-heading and auto-depth. The ROV is used in conjuction with the research vessel *RV Gunnerus*.

Minerva is complemented with five thrusters with frequency controlled alternating current engines. Four of the five thrusters are identical, with one single propeller. Two of them are positioned vertically, while the other two are positioned for forward thrust, with a 10 degree angle. The last thruster is a lateral thruster with two propellers, one on each side. The thrusters can be seen in figure 3.

The joysticks, which can be seen in figure 2 on the control panel send a direct proportional signal to the thrusters. Only the RPM of the thrusters is set and there is no feedback. If something would get stuck in one of the thrusters, or one of the thrusters is malfunctioning the operator will not get any response of this from the ROV. The operator would have to deduce that something is wrong and take the appropriate action.

Figure 2: Martin Ludvigsen using the joystick control console. Photo by Frida Holsten Gullestad.

The control panel also features a set of switches and buttons for control of lights, cameras and manipulators.

Minerva is also outfitted with a non factory-standard RDI Navigator 600kHz Doppler Velocity Log (DVL) which can be seen in figure 4. This can be used to measure both water velocity and bottom (seabed) velocity. The Doppler has a minimum altitude of 0.7

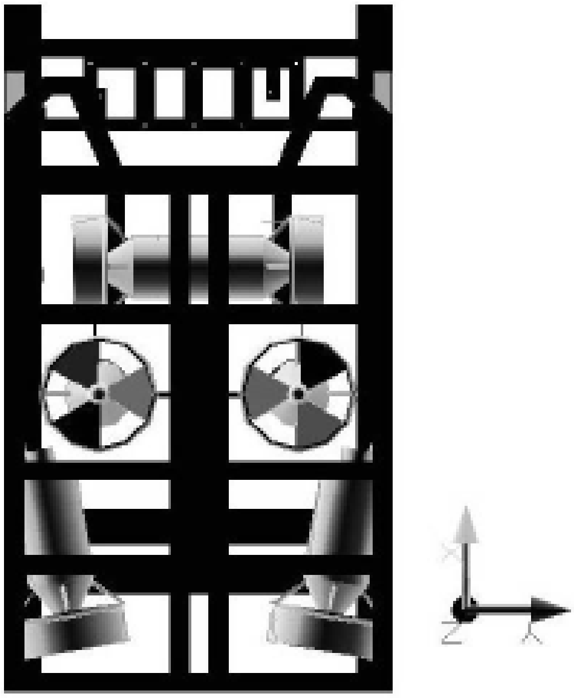

Figure 3: The thruster arrangment for Minerva.

Sub-fighter 7500	
Dimensions:	L*W*H = 152*82*84
Depth rating:	1500 MSW
Weight:	520kg with 30kg payload
Power:	2 Hp, 300-340 N thruster
Sonar:	MS1000
Compass:	Fluxgate

Table 1: Specifications of Minerva

[m] and a maximum altitude of 90 [m]. After seatrials it was registered that at altitudes specific altitudes the DVL performed very poorly.

Figure 4: The DVL mounted on the rear of Minerva.

Other specifications of Minerva can be seen in table 1.

3.2.2 Regulator Computer Specification

Since we are working with realtime implementation and running several embedded Matlab function simultaniously the computer should be quite powerful. The fullscale tests where run on a intel dual core e6750 processor with 4gb of DDR2 RAM with a high speed PCI serial card. This should have been more than enough, but in the fullscale tests the processor was sometimes working at 100%, which can create lags in the realtime loop.

4 ROV Control System Architecture

4.1 Kinematics

4.1.1 Degrees of Freedom

The 6 DOF standard definition from SNAME for marine vessels, which is used in this thesis, can be seen in table 2 (Fossen, 2010).

DOF		Velocities	Positions and Euler Angles
1	motions in the x-direction (surge)	u	x
2	motions in the y-direction (sway)	v	y
3	motions in the z-direction (heave)	w	z
4	rotation about the x-axis (roll)	p	ϕ
5	rotation about the y-axis (pitch)	q	θ
6	rotation about the z-axis (yaw)	r	ψ

Table 2: SNAME notation for marine vessels.

4.1.2 Reference Frames

In the project there are two reference frames that are used, the n-frame and the b-frame. Since the ROV only will operate within a small geographical area, we can assume that the Earth is flat in the operational sphere.

The n-frame originates at some point on the tangent plane to the Earths surface. The x-axis points north, the y-axis east and the z-axis down towards the center of the Earth.

The b-frame is the frame starting from the body of the ROV, also called the body-frame. The x-axis here points in the surge direction, the y-axis in sway direction and the z-axis in the heave direction, normal to the x-y plane.

The position and orientation is given by η, as can be seen in equation 1, which is in the n-frame.

$$\eta = \begin{bmatrix} x \\ y \\ z \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{1}$$

The translational and angular velocities are given by ν, as can be seen in equation 2, which is in the b-frame.

$$\nu = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \tag{2}$$

The translation between the reference frames is given by equation 3. This translation is used in the surge, sway and yaw controller described later in this chapter.

$$\dot{\eta} = J(\Theta)\nu \tag{3}$$

where $\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}$ is the vector which gives the orientation of Minerva.

4.2 System Architecture

To find the position and orientation of the ship *RV Gunnerus*, the ship is fitted with a onboard GPS (Seatex DPS116). The position of Minerva relative to the ship is done by using a HiPAP SSBL system. The velocity in the surge and sway directions are found by a RDI Navigator 600kHz DVL. The measurements are transmitted to the navigation program *Navipac* which transmits the measurements on the RS-232 protocol to the regulator computer.

The measurements from Minerva through the umbilical, the depth, altitude above sea bottom, heading and yaw rate, are transmitted to the ROV navigation computer. This raw data is then sent unprocessed to the regulator computer.

4.3 Platform for the ROV Control System

In the present configuration, the actuators receive the desired thrust directly from the control computer or joystick. The plan is to implement a regulator computer which will receive all the calculated sensor data and compute the desired thrust. The control computer will then send the calculated desired thrust over the RS-232 protocol to the ROV. This can be seen in figure 5. The change between manual and regulator control is done by a switch you manually flip to decide which of the systems sends signals to Minerva.

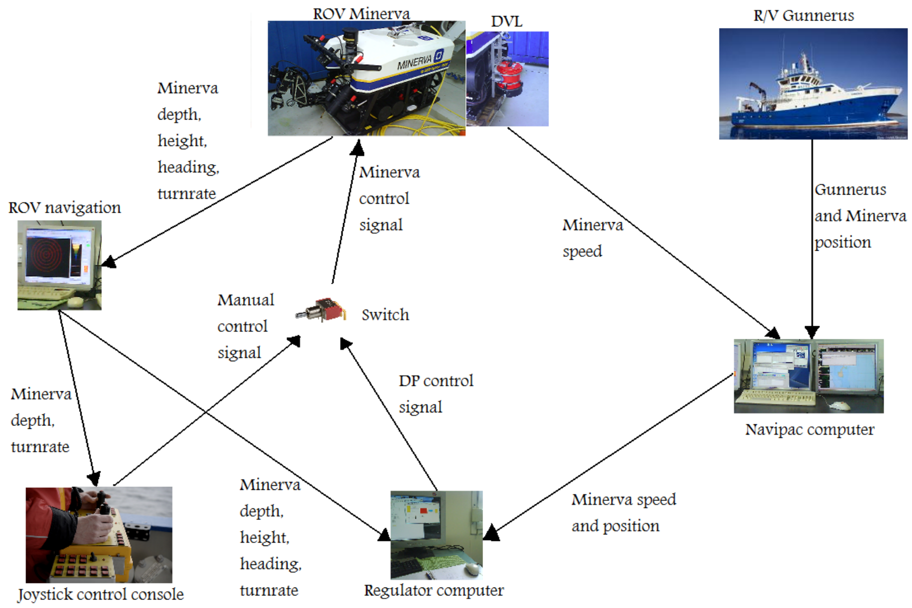

Figure 5: Communication set-up.

4.4 Control System

4.4.1 Underactuated or Fully Actuated Control

When creating a control system for Minerva, it was important to understand the difference between the control of underactuated vehicles and the underactuated control of vehicels. To clarify this we define the DOF as the independent displacements and rotations as seen in table 2. The configuration space (n) is the DOF where the craft is not subject to external constraints. The working space (m) is the space in which the control objective is defined. The number of independently controlled acturators is (r). The following statements makes the difference between underactuated or fully actuated control clear (Fossen, 2010).

- Fully actuation means that independent control forces and moments are simultaneously available in all DOF. Moreover, all positions in the configuration space have actuation such that r = n.
- An underactuated vehicle has independent control forces and moments in only some DOF. Moreover, r <= n. Stabilizing and tracking controllers for underactuated vehicles are usually designed by considering a working space of dimension m < n satisfying m = r (fully actuated in the working space but not in the configuration space).
- Underactuated control is a technical term used in control theory to describe a motion control system for a craft that has a lower numer of independently controlled

actuators than DOF (r < n). To design a control system that achieves stabilization and tracking for this case is nontrivial.

In the case for Minerva, since it is an under water vehicle, the configuration space equals the working space. The number of independently controlled actuators is 4. Since r = 4, the number of DOF we can can control without using underactuated control is 4. Since the 4 DOF we would most like to control are surge, sway, heave and yaw and because Minerva is inherently stable in roll and pitch, those 4 DOF where chosen to be controlled.

4.4.2 Nonlinear PID control

The controllers chosen where 4 independent nonlinear PID controllers. They can be a bit difficult to tune correctly, but if done so, they become very stable. The input/output of the controllers can be seen in figure 6. The surge, sway, yaw controller was taken from the Matlab/Simulink library Marine Systems Simulator (MSS). The heave controller seen in figure 7 was created using a single-input single-output (SISO) of the surge, sway, yaw controller.

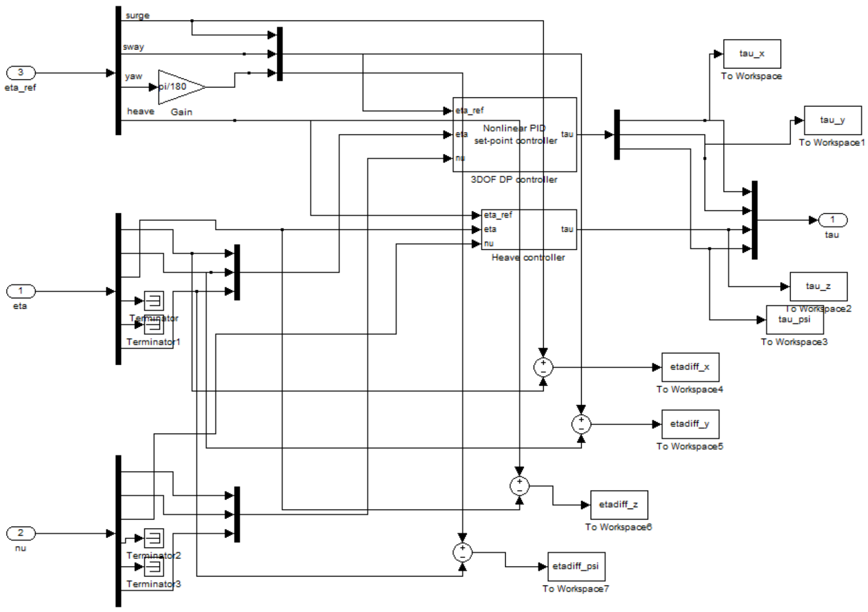

Figure 6: The controller overview.

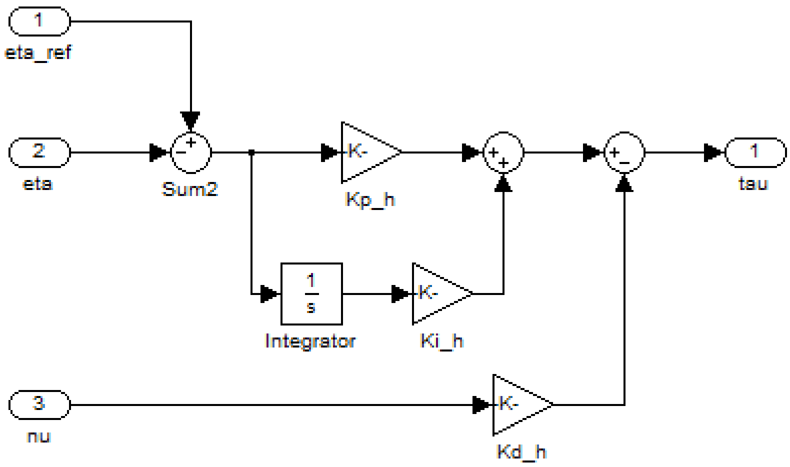

Figure 7: The heave controller.

4.4.3 Control Allocation

The controllers calculate a force and moment vector that should be applied to the vessel. This is done by translating the vector τ to the force vector u by using equation 4, where T(α) is the matrix seen in equation 5.

$$\tau = T(\alpha)u \tag{4}$$

The force vector u is composed of:

u1: force from the lateral thruster u2: force from both of the vertical thrusters u3: force from the port thruster u4: force from the starboard thruster

The thrust configuration of Minerva in 4DOF can be seen in equation 5. This reduction from the 6 DOF model can be made with the assumption that the roll and pitch DOF are stable.

$$T_{4DOF} = \begin{bmatrix} 0 & 0 & \cos(-10) & \cos(10) \\ 1 & 0 & \sin(-10) & \sin(10) \\ 0 & 2 & 0 & 0 \\ lx1 & 0 & lx4 * \sin(-10) - ly4 * \cos(-10) & lx5 * \sin(10) - ly5 * \cos(10) \end{bmatrix} \tag{5}$$

The values used in equation 5, which can be seen in equation 6, are approximated since the center of gravity (CG) is unknown. The thruster configuration can be seen in figure 8 and in figure 9. The CG is assumed to be in the centroid of the xy-plane of Minerva.

$$\begin{aligned} lx1 &= 0.163m \\ lx4 &= -0.570m \\ ly4 &= -0.300m \\ lx5 &= -0.570m \\ ly5 &= 0.300m \end{aligned} \tag{6}$$

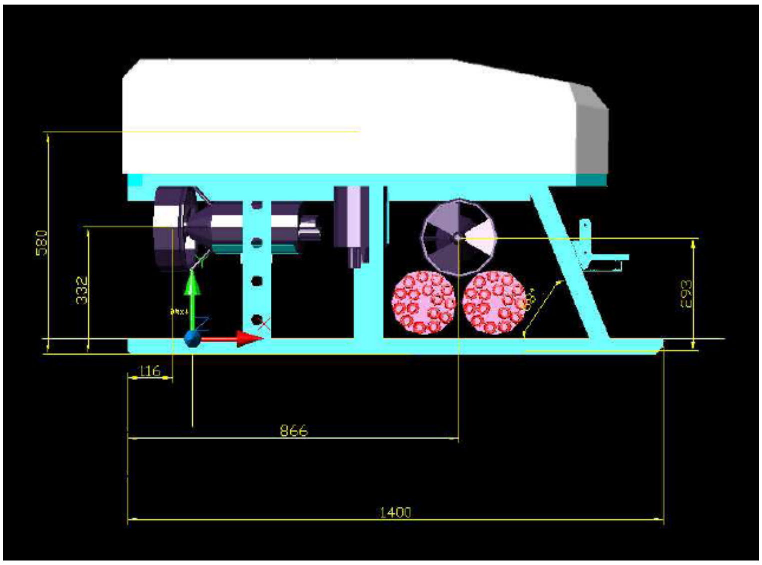

Figure 8: Thruster configuration of Minerva, sideview. Picture by Martin Ludvigsen.

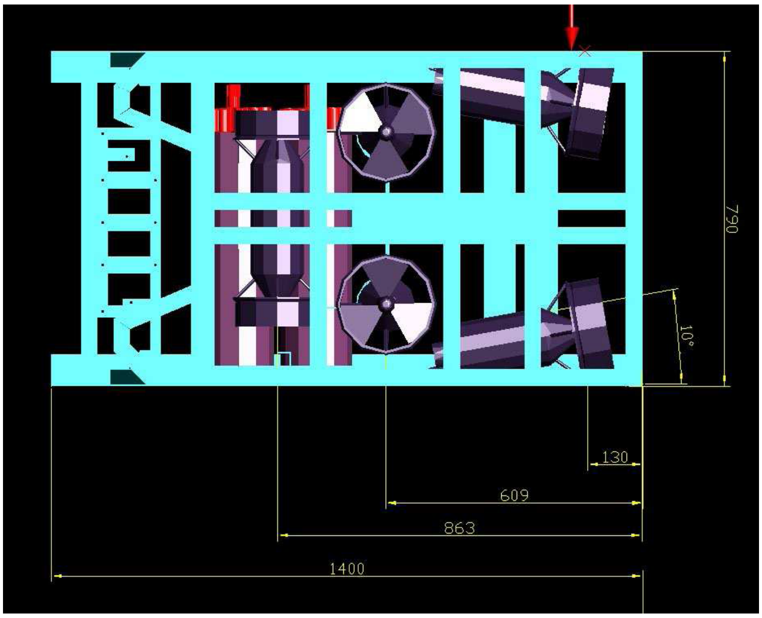

Figure 9: Thruster configuration of Minerva, overview. Picture by Martin Ludvigsen.

4.4.4 Calculating RPM From the Thrust Force vector

The calculation of RPM from the thrust force vector was done using equation 7.

$$u = K_t \rho D^4 \left| n \right| n \tag{7}$$

The diameter D, and the water density ρ are known. The variable n is here in RPS, but is later converted to RPM. The thrust coefficient K_t depends on the advance number. In equation 8 we see that the advance ratio, J, is dependent upon the inflow velocity to the propeller V_a.

$$J = V_a / nD \tag{8}$$

For the following calculation it was assumed that V_a is unknown and that means that J is 0. Then the maximum rotational speed and the maximum force, which was supplied by (Ludvigsen, 2006), was inserted. In this manner i linearised the mapping between the thrust force and RPM. I found the mapping values to be:

vertical: 103

port/starboard (u>=0): 66

port/starboard (u<0): 98

lateral: 73

Notice the difference in the mapping when the rotational speed is positiv and negativ. This is due to the obstruction of the waterflow when reversing. Since the maximum RPM of the thrusters is $\pm$ 1450, the controller saturates the output at this value.

4.5 Real Time Implementation

The realtime implementation was done using a Real Time timer function block. The RT block is connected to the DP system so that the system is dependant upon the RT block. This can be seen in figure 10. The whole simulation was run at 10 [Hz].

4.6 Discussion of Results

In the fullscale sea trials the controller was tested in heave and yaw. The results can be seen in chapter 8. The controllers in the 4DOF are all identical, with the same demands upon the thrusters. Since a small thrust difference in the port and starboard thrusters give a large yaw change, the heading controller should be give priority over the surge and sway controllers. Accurate heading control is usually more desired than accurate position.

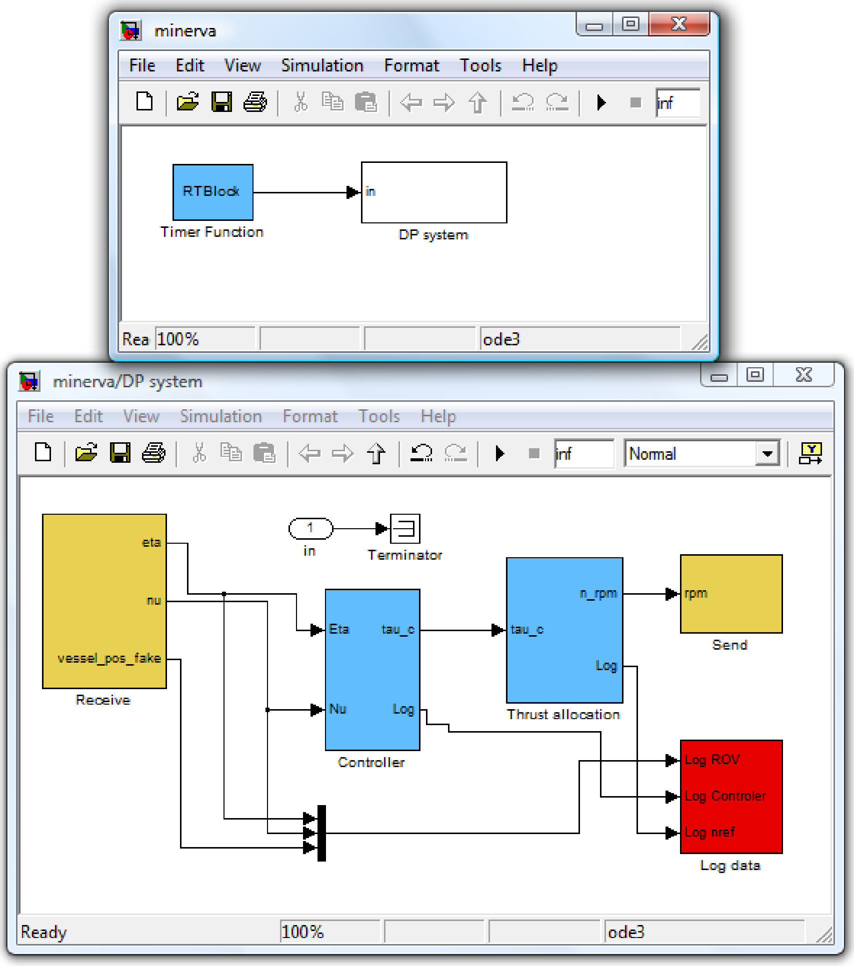

Figure 10: Overview of the realtime function controlling the DP system.

5 Signal Interfaces

To import, process and send the signals the inputdata had to be translated and split. The outputdata had to be calculated and created in the proper way.

5.1 Requirements

The signal interfaces require that the signals from the ROV navigation computer and the *Navipac* computer are received and sent at such a speed that the real-time requirements are satisfied. The interfaces also need to be stable and should be able to be implemented in both Matlab and Simulink.

5.2 Implementation

The implementation was made with a set of RS-232 blocks from (Daga, 2006).

5.2.1 Receiving the ROV Navigation String

The port configuration was set up by a RS232 setup block. The setup parameters where:

Port: COM2

Baudrate: 9600

Number of Databits: 8

Numbe of Stopbits: 1

Parity: None

The RS232 setup block is only active in the start and at the last step of each simulation. The block transmits the handle for the opened communication to the read block. The setup block also sends a flag which is used to communicate if the port is open or closed. The rate transition blocks ensures data integrity during the data transfer. The read block reads all the bytes in input from the serial port and sends it out as a string object. This is done at 2 [Hz]. An example of an input string received can be seen here:

$ 0.0,125.0,+0,101,2057, , , ,

Since we are only interested in the numerical values, the string is split into tokens and converted into double data values. From this string we get the depth from the pressure gauge, the altitude from the altometer, the heading from the compass and the turnrate from the gyro.

The Simulink subsystem created to receive the ROV navigation string can be seen in figure 11.

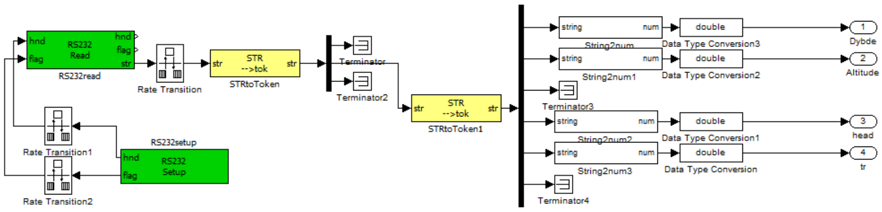

Figure 11: Receiving the ROV navigation string.

5.2.2 Receiving the Navipac String

The port configuration for *Navipac* was set up approximatly in the same way. The setup parameters where:

Port: COM3

Baudrate: 19200

Number of Databits: 8

Numbe of Stopbits: 1

Parity: None

An example of the input from *Navipac*:

!init;561833.90;7003833.73;0.00;0.01;

This input was handled in much the same way as the ROV data input. The output comes on the form of the eastern and northern coordinats, and the ROV speed in x and y direction.

During the sea trials in April it was discovered that the input string was sometimes lost for a sample. This made the subsystem output 0 for the coordinates. The DVL used in conjuction with *Navipac* also sended out a default value of -32.0 whenever the signal was lost. To correct this the inputsignal was checked and if outside a boundary of 1000 [m] the coordinates would be set to the setpoints and outside 5 [m/s] the speed would be set to 0.

The Simulink subsystem created to receive the Navipac string can be seen in figure 12.

5.2.3 The Receive Subsystem

After receiving the values from *Navipac* and Minerva they must be converted into the η and ν vectors used in the controller. This we can see in figure 13. The turnrate musts be altered using equation 9 to find the value in degrees wanted for the controller. The unkown values such as the roll angle, pitch angle, rollrate and pitchrate are set at 0.

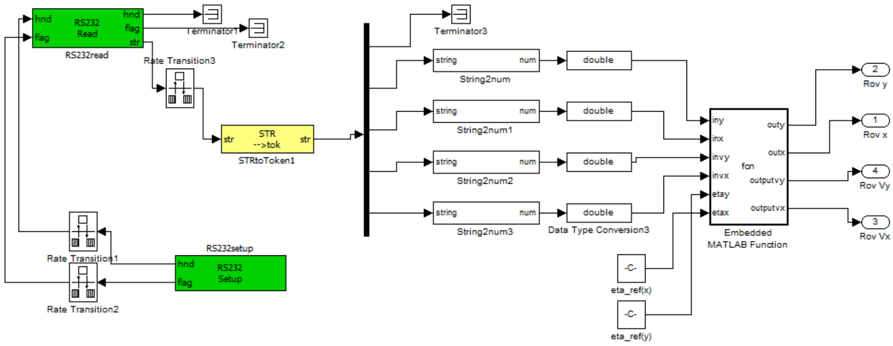

Figure 12: Receiving the Navipac string.

They are not a DOF that is controlled, so they are not necessary for our controller. Also notice that the altitude is not used.

$$\nu(6) = r = (tr - 2062)/16.4 \tag{9}$$

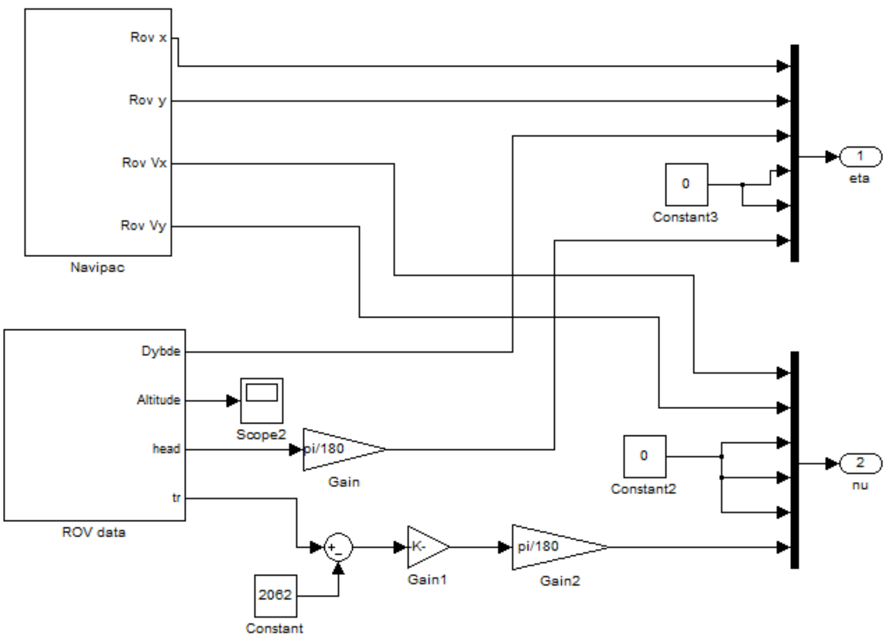

Figure 13: The receive subsystem.

5.2.4 Sending the Control String

After the controller has processed the data and the thrust has been allocated into the thrust vector, the data must be transmitted to Minerva.

The thrust vector is rearranged and the string built. This was done following an algorithm provided by *Sperre AS*. This was not very easy since it involved creating the different bytes and by rearranging the individual bits.

The same RS232 setup block used in receiving was used here with these parameters:

Port: COM4

Baudrate: 9600

Number of Databits: 8

Numbe of Stopbits: 1

Parity: None

The string is split into each of its bytes and written at 5 [Hz]. The Simulink send sybsystem can be seen in figure 14.

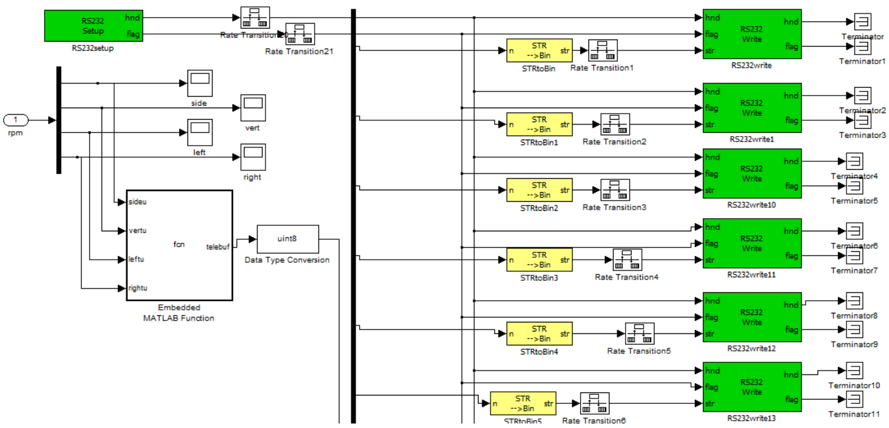

Figure 14: Subsystem showing the sending of the command string.

5.3 Testing

5.3.1 Receive Testing With GPS

The receive implementation was first tested with a Garmin etrex GPS. The regulator computer was installed with a ST-Lab high speed PCI Serial Card. The GPS was used using a baudrate of 9600, the same as used in the receiving from the ROV. The same baudrate was set for the RS-232 blockset and the GPS. The subsystem was then run and the outputdata was checked against the input data. A program called *Terminal*, which reads the whole inputdata from a com port, was used to verify that the data was received correctly. This program was very handy in that it can receive and show the data as both HEX, ASCII, Bin and decimal values.

5.3.2 Receive Testing With Minerva

The receive system was also tested at Trondheim Biological Station. Minerva was powered up and sent data in the regular fashion. The *Navipac* system however is dependent on the receiving of position data from *RV Gunnerus*. Navipac was then setup to transmit dummy position and speed values. *Terminal* was also used here to record the data sent from the joystick control panel. This was important, so that it would be easier to check that the algorithm from *Sperre AS* and the send subsystem worked correctly.

5.3.3 Send Testing

The send subsystem was first tested with the creation of a dummy control vector. The output datastring was then created, and using a nullmodem cabel sent from one com port to another. With *Terminal* it was a simple task to verify that the sent, and received, datastring was the same as the datastring from the joystick control panel.

5.3.4 Send Testing With Minerva

After confirming that the datastring sent was correctly built, the next step was to send a control vector to Minerva. Different control vectors, each with only giving values to one of the thrusters, where created and then run. It was then a simple task of checking that each of the thrusters operated according to the values.

5.4 Discussion of Results

The reading and sending took quite a lot of trial and error to get correct. The implementation of the send algorithm was especially challenging. The end result did work as intended. However during the last seatrial it was noticed that the received northern coordinates where only accurate to 0.5 [m]. This is of course not acceptable and will have to be corrected. The error could be in the receiving of the string, the splitting of the string into tokens or the transformation of the values from string to numerical values.

6 Human Machine Interface

The Human Machine Interface (HMI) is compromised of the manual control from the joystick and the Graphical User Interface (GUI), controlling the DP system. Since there where many changes made to the system under the fullscale seatrials, especially to the tuning variables, the GUI was not used during these tests.

6.1 Graphical User Interface

The Graphical User Interface (GUI) will control a Simulink model which will run the DP system for the ROV.

6.1.1 Requirements

The GUI should be easy to use and should have basic underlying safety precautions built in. For instance, you should not be able to input letters or input a setpoint that could cause damage to the ROV, or other dangerous situations. The GUI is made for the 4 DOF we want to control.

6.1.2 Implementation

The GUI was created with Matlabs built in GUI function *GUIDE*. Implementing a GUI with Simulink is not straightforward. Since Simulink uses its own workspace you either have to create all the necessary variables in your GUI, or run your GUI and Simulink in the main workspace, which is not advisable. For this GUI it was chosen to run Simulink in its own workspace. Other problems are that *GUIDE* is not inherently updateable. Hence, it is not straightforward to send and receive data from your Simulink model. It is possible to update the GUI in near real time with values from either the workspace or directly from the Simulink model. However, to implement this I would have to change the callbacks directly from the model and force the GUI to redraw the needed boxes. This is doable, but creates so much more processing power needed to run the Simulink model. Therefor I believe the best course of action is to let the Simulink model display the actual values of the ship in real time through scopes or plots.

In figure 15 we see the default start screen created when running the GUI.

- The 'ON' button send the values which are set to the workspace and starts the simulation.
- The 'OFF' button terminates the simulation.
- The 'SET' button sets the corresponding inputed value to the 'Values set:' column.
- The 'SET ALL' button sets all the inputed values at the same time.

- The 'DEFAULT' button sets all the valies in the 'Values set:' column to the defaults.

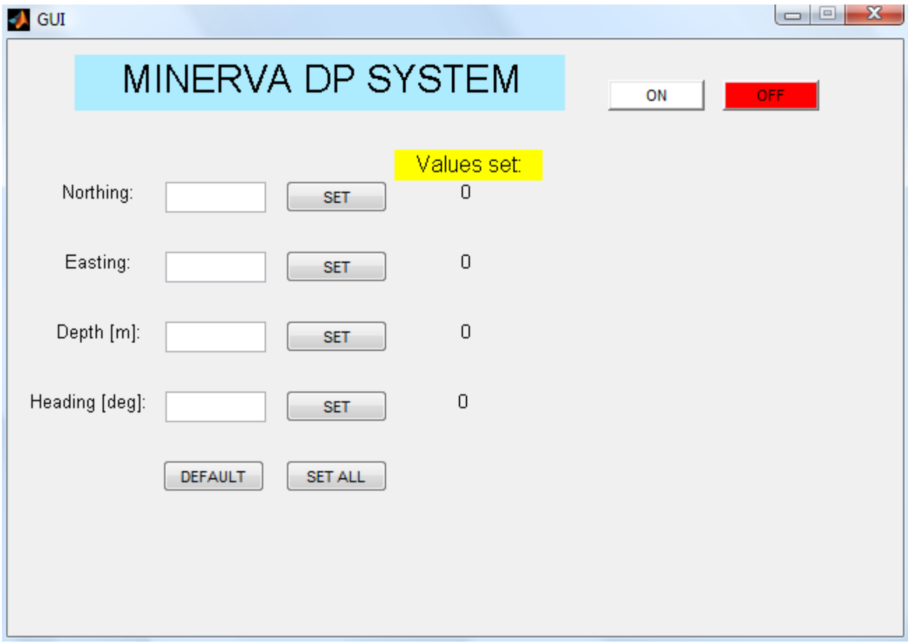

Figure 15: The default start screen for the GUI.

6.1.3 Testing

The GUI was tested with a Cargo ship model, fitted with an autopilot. The heading is sent from the GUI into the Simulink model. When pressing the 'ON' button the simulation starts and the course is plotted from the Simulink window. The default values are set at [0 0 0 30] for [Norting Easting Depth Heading] respectivly for this test. These values will be automatically inputted by pressing the 'DEFAULT' button. The 'OFF' button ends the simulation by calling the MATLAB code:

```
set_param('ship_kalman', 'SimulationCommand', 'stop');
```

This command sets the parameters of the simulation and commands the Simulink model to stop.

6.1.4 Discussion of Results

The GUI does work as predicted. The buttons all work correctly and interface with the Simulink model as desired. However there is the problem of updating the setpoints from the GUI to the model, while the model is running. This is doable, but creates many of the same problems as the updating of the GUI from model creates. The solution to this would be to create a small pause in the simulation, let the setpoints be updated, and continue the simulation.

7 Plan for Fullscale Implementation

7.1 Plan for April 26th and 27th

The first tests are planned to take place on April 26th and 27th. The main goals are to perform an initial test of the signal processing system and to try the different controllers.

7.1.1 April 26th

The schedule that was set up for this day was:

- 8:00. Departure from the docks at Brattøra.
- 8:00 - 10:00. Loading Minerva and the container from Trondheim Biological Station.
- 10:00 - 11:00. Setup of the container and lowering Minerva into the water.
- 11:00 - 15:00. Testing.
- 15:00 - 16:00. Retrieving Minerva and docking at Brattøra.

The testing planned for this day is the first fullscale communication test. This involved making sure all signals from *Navipac* and the ROV navigation computer was received properly and that the control string was sent and received by Minerva.

Secondly the first controller tests are to be made, firstly in 1 DOF.

7.1.2 April 27th

The schedule that was set up for this day was:

- 8:00. Departure from the docks at Brattøra.
- 8:00 - 9:00. Setup of the container and lowering Minerva into the water.
- 9:00 - 13:30. Testing.
- 13:30 - 15:30. Retrieving Minerva and offloading Minerva and the container at Trondheim Biological Station.
- 16:00. Arrive at Brattøra

The controllers are to be further tested and tuned. Marianne Kirkebys controller is also to be tested. The first DP test in 4 DOF is to be attempted.

7.2 Plan for June 7th and 8th

For the second set of seatrials the goals are to test the DP system as much as possible. The main goals are to create lots of data for a better foundation for further work.

7.2.1 June 7th

The schedule that was set up for this day was:

- 8:00. Departure from the docks at Brattøra.
- 8:00 - 10:00. Loading Minerva and the container from Trondheim Biological Station.
- 10:00 - 11:00. Setup of the container and lowering Minerva into the water.
- 11:00 - 19:00. Testing.
- 19:00 - 20:00. Retrieving Minerva and docking at Brattøra.

This is a long day and Mariannes controller is to be further tested and tuned. The implementation and testing of an autopilot, also made by Marianne Kirkby, is to be tested by sailing in a straight line with a fixed heading, speed and depth.

7.2.2 June 8th

The schedule that was set up for this day was:

- 8:00. Departure from the docks at Brattøra.
- 8:00 - 9:00. Setup of the container and lowering Minerva into the water.
- 9:00 - 13:30. Testing.
- 13:30 - 15:30. Retrieving Minerva and offloading Minerva and the container at Trondheim Biological Station.
- 16:00. Arrive at Brattøra

The testing for this day is to be mainly focused on step response tests in different DOF. This is to be done so that better simulation models can be created and that the response of Minerva is better known.

8 Fullscale Seatrials in April and June

Full scale tests with ROV Minerva were performed in the Trondheimsfjord from the NTNU research vessel *RV Gunnerus*. The initial testing took place on April 26-27, 2010, while more extensive testing of the DP system was carried out on June 7-8, 2010. The parts of this chapter, regarding the set-up ans the use of Marianne Kirkeby's controller, was co-written with Marianne Kirkeby. The controller made by Marianne was the one tested the most, since she will continue the work next semester.

8.1 Set-Up

To set up ROV Minerva for duty, several preparations are required. The container needs power from the ships' system and ROV Minerva needs power to the transformer in the container which in turn powers her. The video navigation system needs to be initialized, the regulator computer needs to be connected to the appropriate communication ports, and *NaviPac* and the *Kongsberg Simrad* ROV navigation system needs to be started. The set-up can be seen in figure 5. A small buoy was attached to the top of the ROV to give extra buoyancy. This was necessary due to the mounting of the Doppler Velocity Log (DVL), which is heavy and not part of the standard equipment of ROV Minerva. In figure 16 the container with all its screens is shown from the inside, and in figure 17 the crew of *RV Gunnerus* is lowering ROV Minerva into the sea.

Figure 16: The operator station of ROV Minerva on deck of *RV Gunnerus* with Martin Ludvigsen, Fredrik Dukan, Christoffer Lysdahl and Marianne Kirkeby during the sea trial April 26th. Photo by Frida Holsten Gullestad.

Due to the fact that Marianne Kirkeby had a project thesis of creating a DP controller

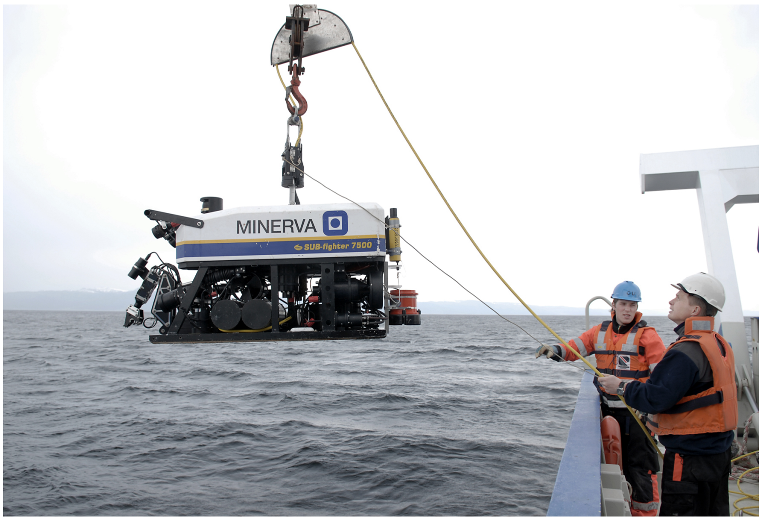

Figure 17: ROV Minerva beeing lowered into the Trondheimsfjord by the crew of *RV Gunnerus*. Photo by Frida Holsten Gullestad.

$$
\begin{array}{lcl}
K_p & = & \mathrm{diag}\{0 \quad 0 \quad 60 \quad 0 \quad 0 \quad 60\} \\
K_d & = & \mathrm{diag}\{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 100\} \\
K_i & = & \mathrm{diag}\{0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0\}
\end{array}
$$

Table 3: Controller gains in heave and yaw for the first test.

for Minerva, her controller was also to be tested. My controller described earlier was tested first. We had limited time and tuning the controller was time consuming, so it was decided that the second controller made by Marianne Kirkeby would be tuned and tested further.

8.2 Initial Testing of the DP System

8.2.1 Testing of My Controller

My controller was tested in the heave and yaw directions. My controller was never tested in full DP due to time constraints. The controller gains used for my controller can be seen in table 3. We can see that the heave controller was only set as a proportional controller. The yaw controller had quite a large relative dampening, but this was necessary due to the volitality of the thrusters compared to yawrate.

The desired depth and heading can be seen in equation 10, where the depth is in meters and the heading in degrees. Minerva was kept quite close to the seabottom which was approximatly 216 [m].

$$\eta_d = \begin{bmatrix} 0 & 0 & 213.0 & 0 & 0 & 90.0 \end{bmatrix}^T \tag{10}$$

In figure 18 we can see that the proportional controller in heave works pretty well. Since the controller does not have any integral action, it is not able to correct the offset error. Even so the max deviation is only 0.3 [m]. We also see that the heading overshoots with nearly 20 degrees, but later settles at around ± 4 [deg]. In figure 19 we observe the RPM of the thrusters. With max RPM for the thrusters beeing ± 1450 the thrust usage is calm and acceptable, especially in heave. With further tuning this controller would probably be acceptable in heave and yaw.

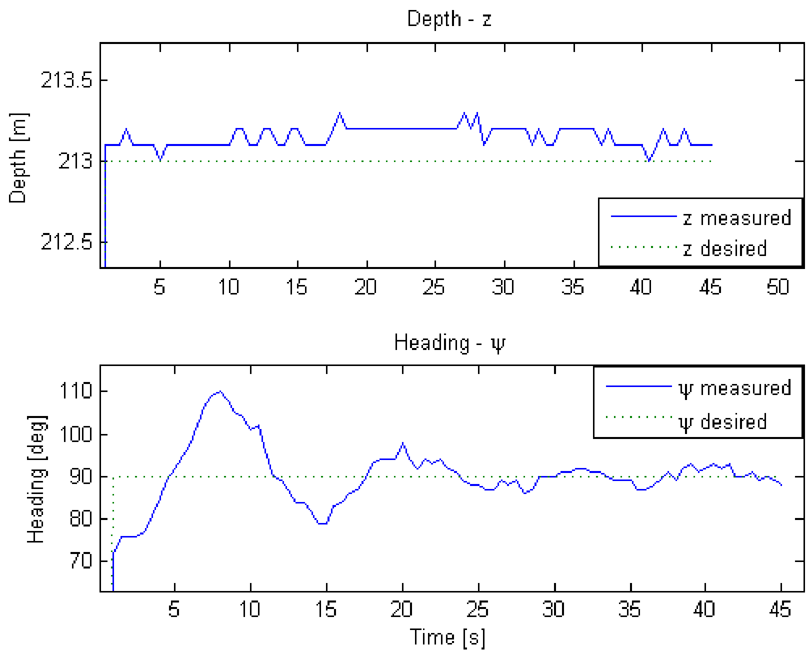

Figure 18: The desired depth/heading and actual depth/heading.

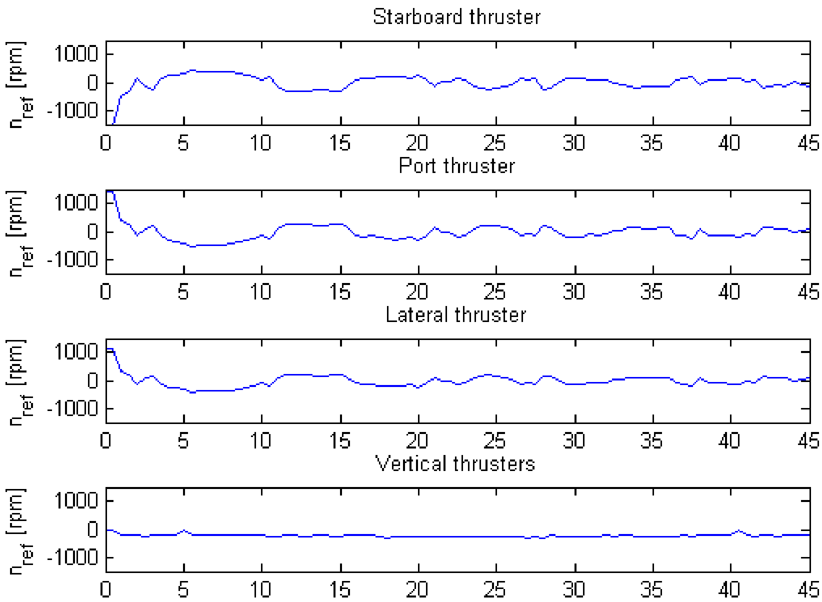

Figure 19: The RPM of the different thrusters.

All further testing was done using Marianne Kirkebys controller. Her controller was based on the same principle as mine, with 4 independent PID controllers working to control one DOF each. Her controller also had anti windup, making sure that the integral action did not come out of control.

8.2.2 Testing With Marianne Kirkebys Controller

The first test of the DP system in all 4 DOF was performed with the gains given in table 4. Note that there is no integral action in the north and east position. The anti wind-up gain was set to 0, such that there is no anti wind-up in this test. The surface current measured onboard *RV Gunnerus* was 0.5 knots (0.257 m/s) with direction 265 degrees. The ROV was kept close to the sea bottom, which was at about 221 meters. The heave velocity was not measured, hence there is no derivative action in the depth control.

The set-point and initial position are given by equation 8.2.2.

$$\eta_d = \begin{bmatrix}7041888.0 & 578433.0 & 216.0 & 0 & 0 & 90.0\end{bmatrix}^T$$
$$\eta_0 = \begin{bmatrix}7041880.5 & 578435.2 & 217.2 & 0 & 0 & 100.0\end{bmatrix}^T$$

In equation 8.4 the dimension is in meters, except for the heading, which is given in degrees. The initial velocity is close to zero.

In figure 20 the deviation between measured and desired position is shown. The set-point is 8 meters north and 2 meters west of the initial position. After 1 minute the deviation in both north and east position is 1 meter and then it varies between 0.5 and 1 meter. From the plots we see that there are drop-outs in the east position, 12 times during the 122 seconds of the test. The heading is oscillating around the set-point. Initially the deviation is 10 degrees and after 80 seconds it is still about 5 degrees. Later there is a jump at 116 seconds that gives 12 degrees deviation. This could be due to drop-outs in the east-position. The depth controller reduces the deviation in depth from 1.2 meters initially to 0.2 meters after 30 seconds. Later there is a jump in the depth to 0.5 [m] away from the set point.

Figure 21 displays the measured velocities. The same drop-outs of measurements as in the east position are found for surge and sway speed. Figure 22 shows the desired rotation speed of the five thrusters. We see that the drop-outs in position and velocity causes the longitudinal and lateral thrusters to jump temporarily to the maximum value. Apart from the jumps, the desired rotation speed does not give the thrusters hard work.

K_p	=	diag{10	10	60	0	0	30 }
K_d	=	diag{20	20	0	0	0	100}
K_i	=	diag{0	0	0.05	0	0	0.1 }

Table 4: Controller gains for DP full scale testing, first time.

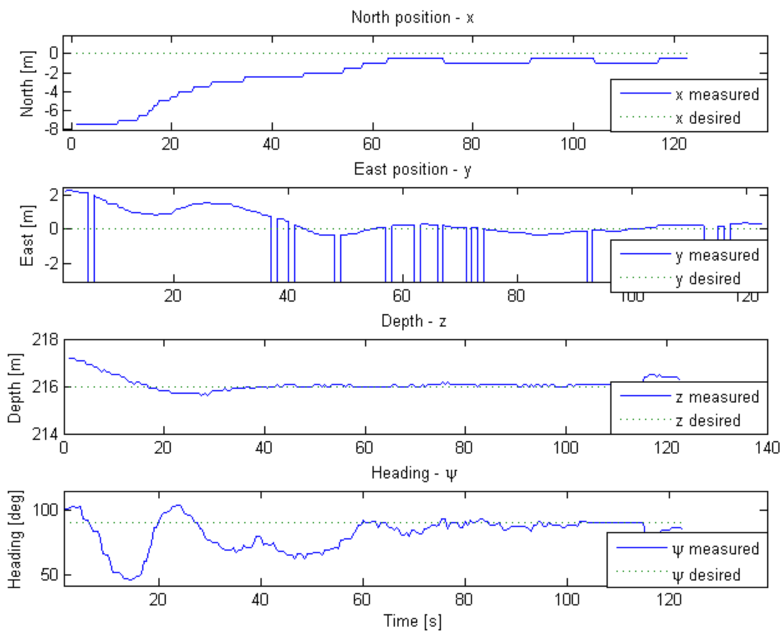

Figure 20: The deviation between position and desired position of ROV Minerva in the first full scale DP test.

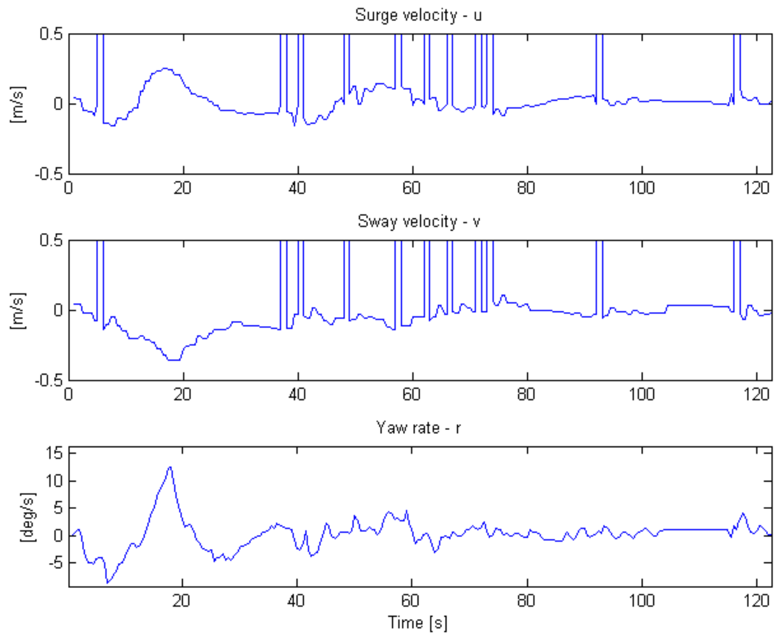

Figure 21: The measured velocity of ROV Minerva in the first full scale DP test.

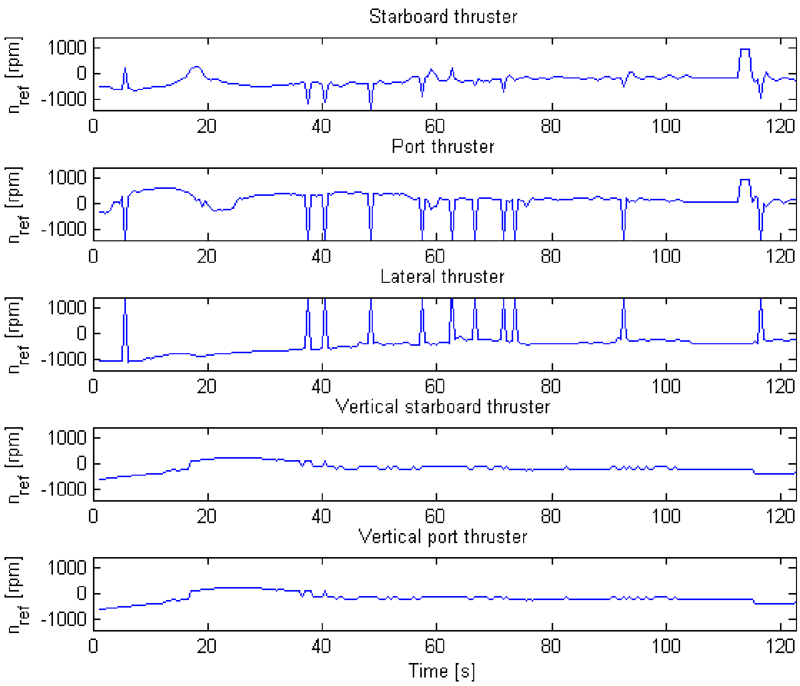

Figure 22: The desired rotation speed for the five thrusters of ROV Minerva in the first full scale DP test.

8.3 Tuning of the DP Controller

The gains were adjusted in the second sea trial after simulations done by Marianne Kirkeby. During the testing the ROV was kept close to the sea bottom, which was at about 35 meters. There was very little current, less than 0.1 [m/s] at this depth. As in the first test, the heave velocity was not measured, hence there is no derivative action in the depth control. Anti wind-up was not used. When the ROV was taken up after testing, it had taken in water such that it was no longer positive buoyant, but slowly sinking.

First, we show a test of the ROV without the DP system. The joystick controller from *Sperre AS* has a function for auto-depth which was used to keep the depth constant while letting the ROV drift in the horizontal plane. The drift of position and heading are shown in figure 23. This is to compare with the later results when the DP system is turned on. Because there was little current, the drift is only about 5 meters north and 4 meters west during the 200 seconds, but there is a large change in the heading, about 200 degrees.

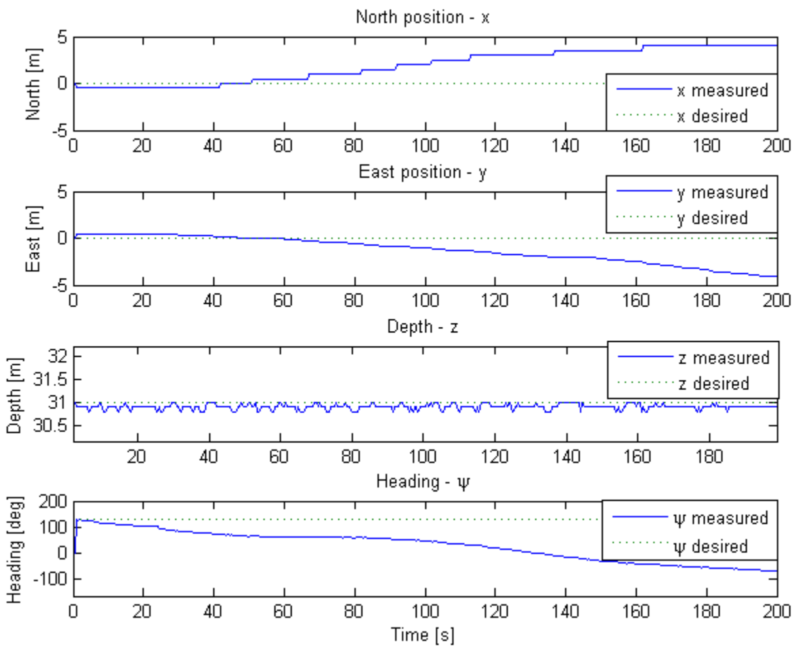

Figure 23: The position of ROV Minerva drifting in the horizontal plane without DP control.

The gain matrices used in the PID controller are diagonal, thus simplifying the tuning. The heading proved to be the hardest DOF to tune. This is due to very small thruster values giving large responses in yawrate. Here we will show 3 tests where the controller gains on the heading are varied.

- Heading test 1: With $K_p = 50$, $K_d = 80$, $K_i = 1$
- Heading test 2: With $K_p = 60$, $K_d = 96$, $K_i = 1$. The chosen alternative.
- Heading test 3: With $K_p = 50$, $K_d = 160$, $K_i = 1$

Disregarding the first 10 seconds of the tests, the standard deviation of the heading error was computed to be $\sigma = 4.4$ degrees, $\sigma = 3.8$ degrees and $\sigma = 4.3$ degrees for the 3 tests respectively. In figures 24, 25 and 26 the heading measured in the 3 tests are shown together with the desired rotation speed sent to the horizontal thrusters. Considering the thrusters, the results from the third test are not optimal because the desired rotation speed is oscillating very quickly due to the high derivative gain K_d, thus giving the thrusters hard work. The second test is best both regarding the thrusters and the standard deviation of the heading error, and gains from this test were chosen as final gains.

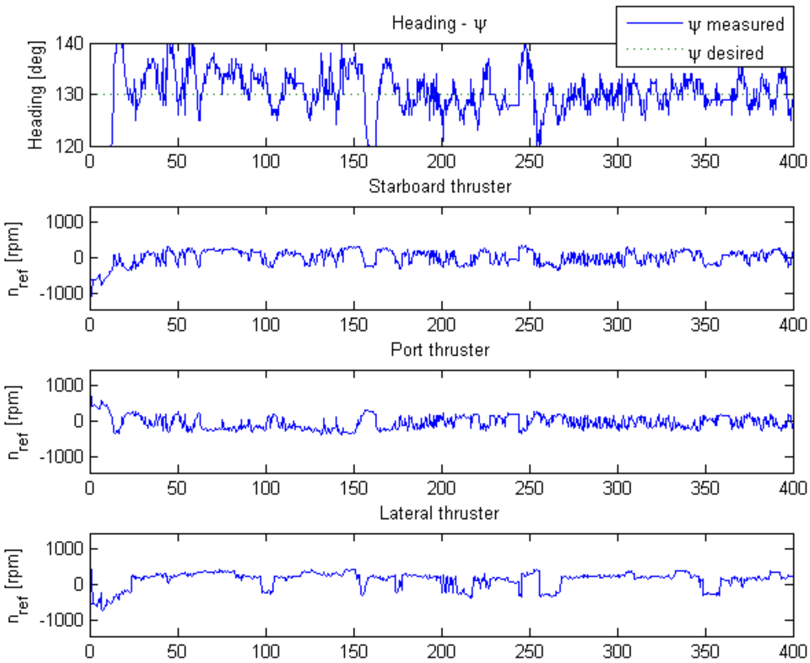

Figure 24: Heading and desired rotation speed of the horizontal plane thrusters. K_p=50, K_d=80 and K_i=1.

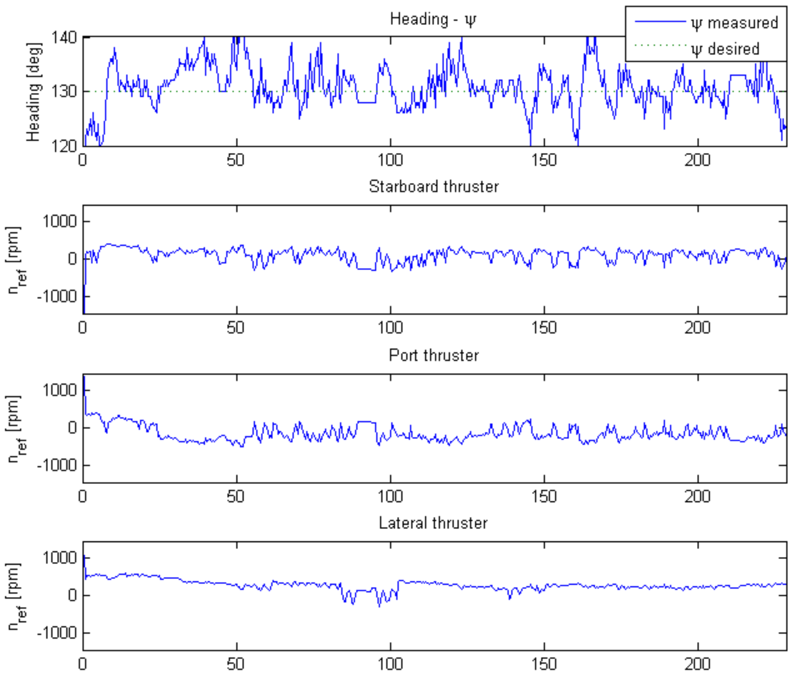

Figure 25: Heading and desired rotation speed of the horizontal plane thrusters. K_p=60, K_d=96 and K_i=1.

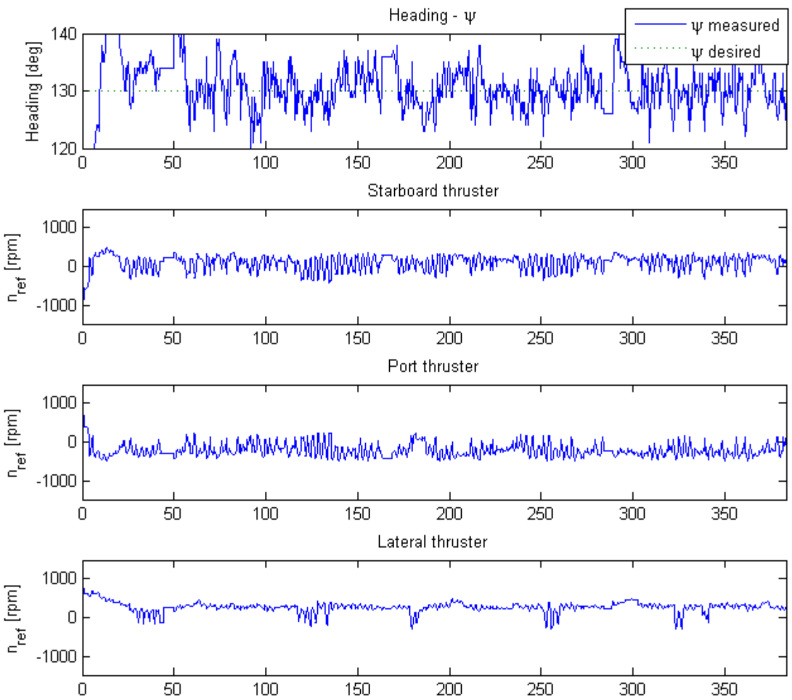

Figure 26: Heading and desired rotation speed of the horizontal plane thrusters. K_p=50, K_d=160 and K_i=1.

K_p	=	diag{20	20	100	0	0	60	}
K_d	=	diag{50	50	0	0	0	96	}
K_i	=	diag{0.5	0.5	2	0	0	1	}

Table 5: Controller gains for DP full scale testing

8.4 Testing of the DP Controller

The final controller gains are given in table 5. Figures 27, 28, 29 and 30 show the position, velocity, control forces and desired thruster speed of ROV Minerva during testing with these gains. The errors in the north and east positions are less then 0.5 meters after 40 seconds. The depth converges quicker, and the error is mostly less than 0.2 meters, but once grows to 0.5 meters. The heading error is generally about 4 degrees, but has peaks reaching 10 degrees. The velocity in surge and sway is generally $\pm$ 0.1 [m/s], while the yaw rate is oscillating with a higher frequency than the two, and has peaks up to 5 [deg/s]. The plot in figure 27 also shows that the accuracy of the north position is only in half meters. This was a problem experienced during the full scale tests, and arises when the signal is taken into the control computer. The accuracy of both the north and east position measurements should be in centimeters. As in the first full scale test, the position and velocity measurements drop out from time to time. To avoid large jumps, the velocity was set to zero every time the velocity signal failed, and the position was set equal to the set-point value.

The control forces displayed in figure 29 are divided in three parts: proportional effect, derivative effect and integral effect. The total forces are also shown. As expected the integral effect is quite stable, while the proportional and derivative parts of the force are oscillating around zero. There is no derivative effect for the depth, because the heave velocity is not measured. The derivative effect is smaller than the proportional effect in the other three DOFs, and especially in surge and sway. The rapid oscillations in heading and yaw rate are reflected in the yaw control moment. From figure 30 we see that the controller does not demand very high rotation speed from the thrusters except for the first seconds. However, the desired rotation speed changes very fast for the horizontal thrusters, especially the longitudinal ones.

The set-point and initial position of the test are given by equation 8.4.

$$\eta_d = \begin{bmatrix} 7036686.0 & 569610.0 & 31.5 & 0 & 0 & 130.0 \end{bmatrix}^T$$
$$\eta_0 = \begin{bmatrix} 7036687.5 & 569609.2 & 30.6 & 0 & 0 & 121.0 \end{bmatrix}^T$$

In equation 8.4 the dimension is in meters, except for the heading, which is given in degrees. The initial velocity is close to zero.

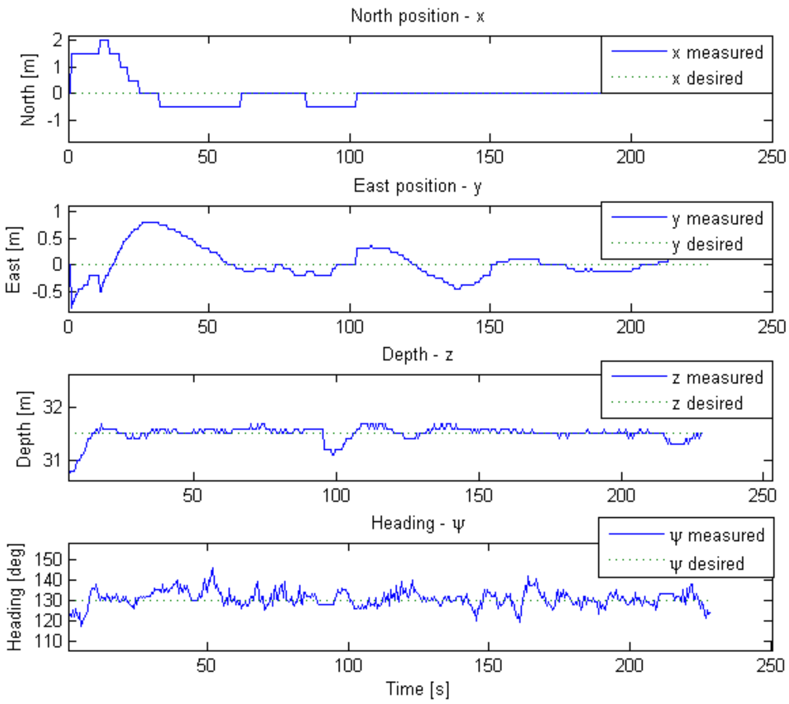

Figure 27: The deviation between position and desired position of ROV Minerva, tested with final controller.

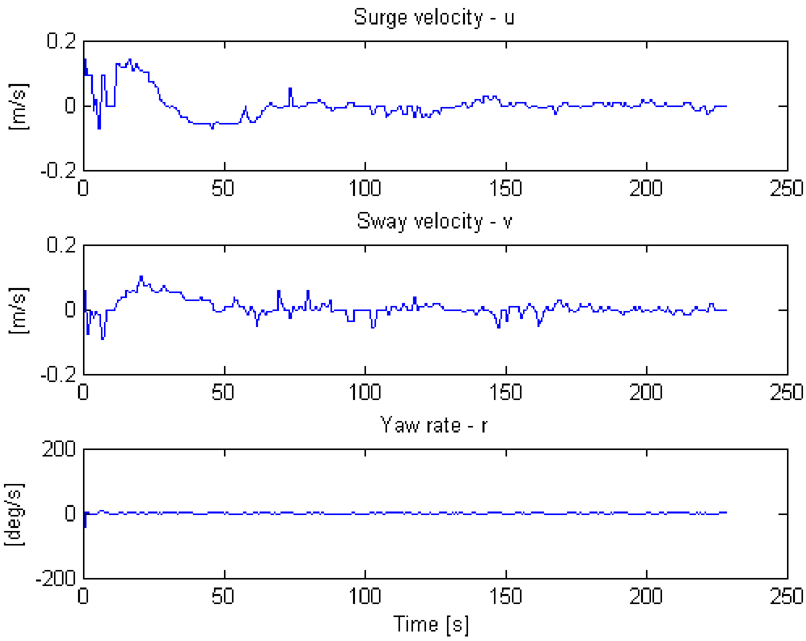

Figure 28: The measured velocity of ROV Minerva, tested with final controller.

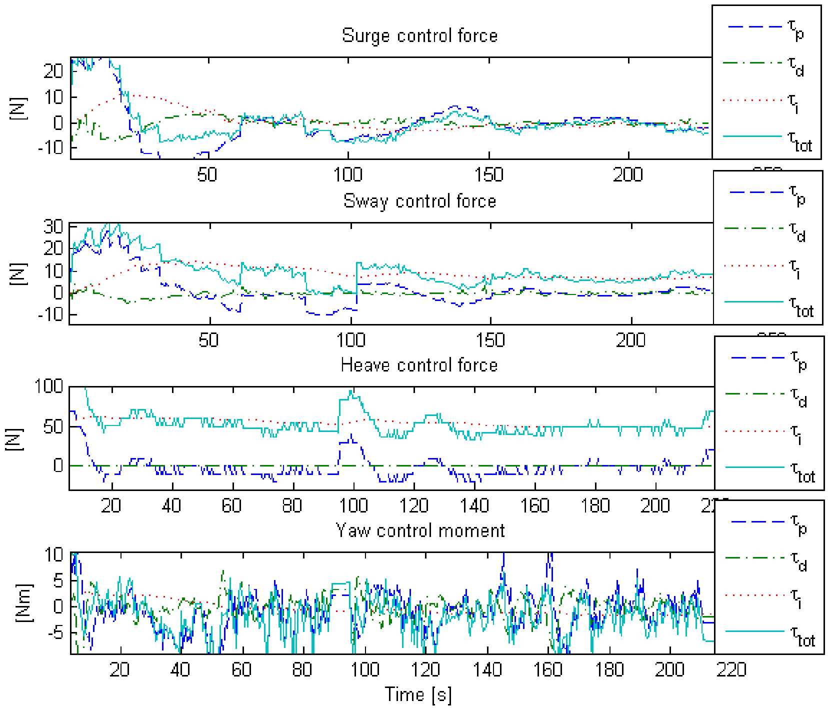

Figure 29: The desired forces and moments of ROV Minerva, tested with final controller.

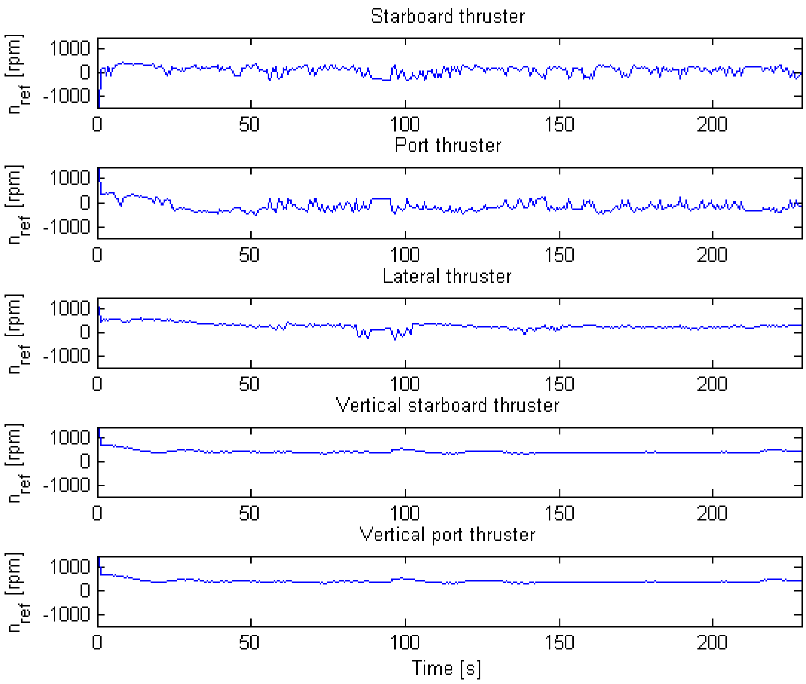

Figure 30: The desired rotation speed for the five thrusters of ROV Minerva, tested with final controller.

8.5 Step Response Testing of ROV Minerva in Closed Loop

ROV Minerva was tested in closed loop, that is, with the DP system working, for steps in the desired position. This was with parameter estimation for the simulation model in mind. Steps of 1, 2, 5 and 10 meters were applied successively in surge, sway and heave. In yaw the steps were 5, 10, 20, 45 and 90 degrees. Unfortunately, some of the data were lost. In figure 31 the response for both positive and negative steps in desired heading are shown together with the desired control force from the controller. In figure 32 the step response for downward steps in desired depth is shown together with the heave control force. When Minerva is moving backwards as in the step response shown in 33, her characteristics are known to be different than when she is moving forwards. This is because the two longitudinal thrusters will be sending water into the ROV body, hence loosing thrust as described in chapter 4.

8.6 Problems With the Regulator Computer

During the tests we experienced several crashes of Matlab. This was caused partly because Matlab works on a network licence and that the internet connection on *RV Gunnerus* was not very stable. There where also problems with the com-ports not closing properly after a crash. This caused problems when trying to initiate a simulation with the RS232 blocks tryin to open a port not properly closed. The problems where also aggrevated with the computer becoming very hot, something known to cause crashes and problems. The computer was located directly across from the electric radiator. During the last tests I set the fan speed on the computer to maximum. After this we experienced no crashes.

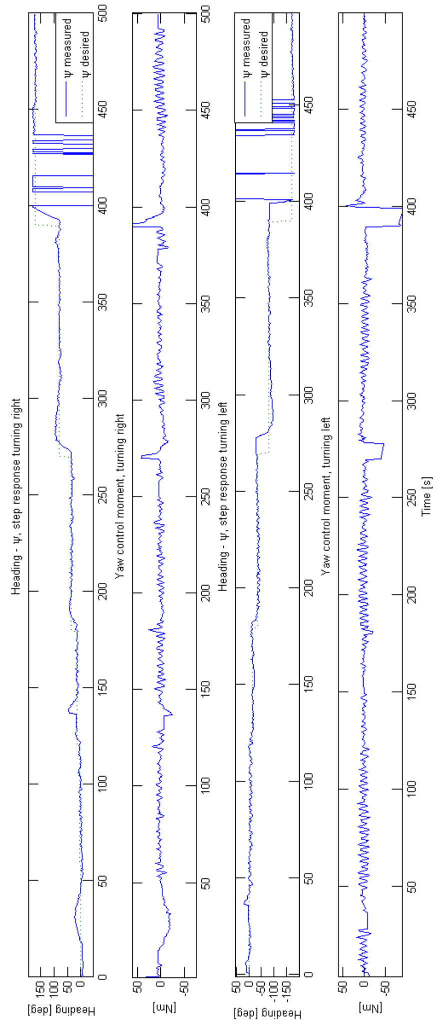

Figure 31: Heading step response of ROV Minerva in closed loop.

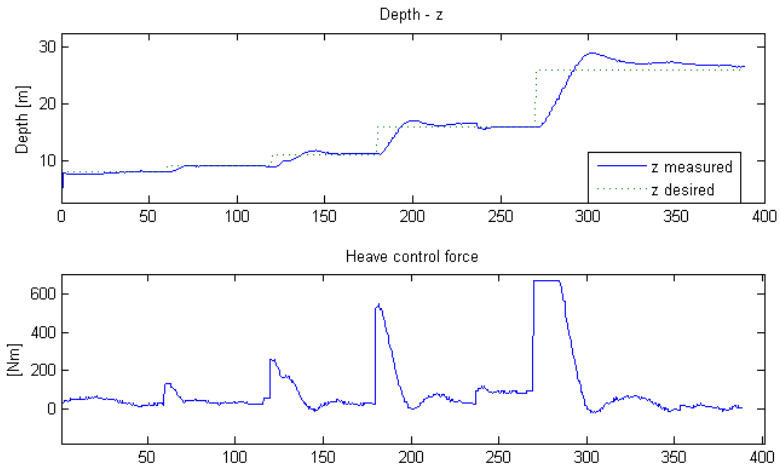

Figure 32: Depth step response of ROV Minerva in closed loop.

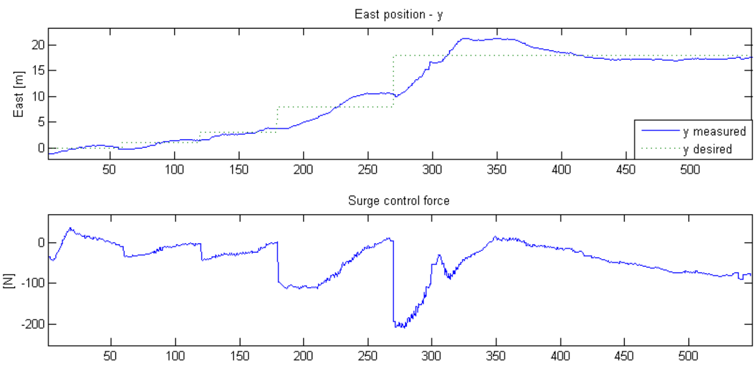

Figure 33: Step response of ROV Minerva going backwards, in closed loop.

9 Conclusion

The project has designed a viable DP system in 4 DOF. The design was implemented in Matlab/Simulink using a realtime function block over the RS232 interface. The system worked well in heave and yaw, but lacked precision in surge and sway. The controllers and the system does work, however more implementations and refinement of the Matlab/Simulink system is needed for it to be viable.

All the objectives set were met. With this work as a basis further experiments should be carried out. There is still many different areas of the system that needs work, but the main principles are sound. This project was inteded as a foundation for further implementation and in that regard the system is a success.

10 Further Work

The depth controller could be modified to use the altitude instead, creating an altitude controller. This would be preferable to a depth controller if an autopilot is created. The compass and turnrate measurements should also be used in a different manner. Since the turnrate is quite jittery, but has low drift and the compass is stable, but drifts, they should be used together with the turnrate updating the drift of the compass. With the current controller the turnrate is used for the derivative gain, while the compass is integrated and used in both the proportional and integral gains. The new controller should combine the compass and turnrate measurements into one heading and derive/integrate from that value.

The realtime implementation should maybe be altered somewhat. The system is now designed with one RT function block running the whole system. It might be better for the send and receive subsystems to work independently and the RT block only running the calculations. In this way the send and receive subsystems will receive/send data when they can, and not be driven by the RT block. The RT block really only needs to control the controllers.

The problem with accuracy must be corrected. The system of checking the signals and only transmitting a speed of 0 when the DVL signal was lost is not optimal. A kalman filter with a predictor should be implemented. This would vastly improve the signal quality and the end DP result. If an altitude controller is to be implemented the receive subsystem would have to be modified to output the altitude.

The GUI could be implemented with a complete set of buttons matching the control keyboard on the joystick control console. In this way, the lights, manipulators and cameras could be controlled from the same computer as the DP system. There should also be written a new subroutine which collects the position and heading of the ROV at that same moment. In this way you would not have to manually input the wanted coordinates. This would be easiest created using a duplicate of the *Navipac* receive system which would open the com port, collect the data over one sample, and close the port again. Then send this data to be displayed in the GUI and to be further used in the simulation.

There should be written a code for checking if the com-ports are closed before trying to initiate a connection. A new regulator computer needs to be purchased and a stand alone Matlab version, not requiring an internet connection, is also needed.

11 References

Svendby - Eirik Svendby *ROV MINERVA*, Master thesis report 2007.

Ludvigsen - Martin Ludvigsen *Sammenfatning av hydrodynamiske forsøk med ROVen Minerva*, 2006.

Ludvigsen - Martin Ludvigsen *An ROV-toolbox for optical and acoustic scientific seabed investigations*, 2009.

Daga - Leonardo Daga *RS-232 blockset, URL: http://digilander.libero.it/LeoDaga/*, 2006.

Sørensen - Asgeir J. Sørensen *Marine Cybernetics, Modelling and Control*, Department of Marine Technology, NTNU, 2006.

Lysdahl - Christoffer Lysdahl *Control System Architecture of the ROV Minerva*, Project thesis, 2009.

Fossen - Thor I. Fossen *Guidance and Control of Marine Craft*, Norwegian University of Science and Technology, 2010.