

MARINE CONTROL SYSTEM II

PROJECT REPORT



Department of Marine Technology
Norwegian University of Science and Technology

April 12, 2019

Contents

1 Case A: DP Thrust Allocation and Joystick Control	2
1.1 Theoretical development	2
1.1.1 Thruster allocation in polar coordinates	3
1.1.2 Thruster allocation in rectangular coordinates	4
1.2 Simulink and model scale testing	6
1.3 Discussion of Results	6
2 Part B: DP Observer Design - State estimation and filtering	7
2.1 Theoretical development	7
2.2 Simulink and model scale testing	9
2.3 Discussion of Results	10
3 Part C: DP Feedback Control and Maneuvering	11
3.1 Theoretical development	11
3.1.1 Path parametrization	11
3.1.2 Kinematic model control design	12
3.1.3 Update law	14
3.2 Simulink and model scale testing	17
3.3 Discussion of Results	17
4 Conclusion	18
Appendix	19
A MATLAB Code	19
B System matrices & parameters	23

C Simulink diagrams	25
D Graphical user interface	32
E Figures of results	35
E.1 Case A - Simulink and HIL estimation	35
E.2 Case A - Model scale experiment	37
E.3 Case B - Simulink and HIL estimation	42
E.4 Case B - Model scale experiment	47
E.5 Case C - Simulink and HIL estimation	49
E.6 Case C - Model scale experiment	50

Introduction

In this project, three cases are involved, consisting of thruster allocation, observer design and adaptive backstepping controller design with maneuvering task. Each case has an associated theory part that is addressed in this report. Based on the simulation, HIL testing and model testing is then done.

For the first part, the joystick is used to input control force to the designed thruster allocation, from which the force is calculated and distributed to each thruster. The force is bounded to avoid breaking thrusters. Also, both rectangular and polar coordinates are used in thruster allocation. Then, the observer is implemented to give estimation of the position and velocity of the ship. The observer is also modified to work for dead reckoning. Next, the backstepping controller with maneuvering under given path is designed.

Finally, the task is to test the maneuvering problem by combining the thruster allocation, observer and the new designed backstepping controller.

Project objective

The main objective of this project is to design a plant model and use it to realize the maneuvering task of the ship.

Project scope

This project consists of three part: design of thrust allocation, observor and maneuvering. In each part, theoretical development, Simulink and HIL simulation and model scale test are included.

1 Case A: DP Thrust Allocation and Joystick Control

1.1 Theoretical development

Assuming that the CS Enterprise only operates at low speeds in calm waters, a simple 3 DOF control model for the vessel's horizontal motion (i.e motion in surge, sway and yaw) is

$$\begin{aligned}\dot{\eta} &= R(\psi)v \\ M\dot{v} &= -C(v)v - D(v)v + \tau,\end{aligned}\tag{1}$$

with the position and velocity vectors as

$$\eta = [x \ y \ \psi]^\top, \quad v = [u \ v \ r]^\top\tag{2}$$

where x and y is NED-position in surge and sway, while ψ is the body-frame heading. u , v and r are surge and sway velocities and yaw rate in body frame. $\tau = [X, Y, N]^\top$ is thrust force and moment, also in body-frame. $R(\psi)$ is the three dimensional rotation matrix for transformations between vessel frame and NED frame, M is the vessel inertia matrix, C the coriolis and centripetal matrix and D is the damping matrix. These system matrices are assumed known to the reader, but are included in the appendix for reference together with relevant parameters.

Given a set of commanded forces and moments from the controller, the thrust allocation algorithm assigns a normalized force vector to each individual thruster. The thruster's PWM modules then translate this vector into an actuator assignment that will achieve the desired force. The thrust load τ_i from thruster i generating thrust $f = [f_x \ f_y \ f_z]^\top$, at body-frame location $l = [l_x \ l_y \ l_z]^\top$ is $\tau_i = [f \ l \times f]^\top$. Because the C/S Enterprise has three thrusters in the surface plane, it is fully actuated in the working space. Hence we can directly manipulate surge, sway and yaw. The thrust load vector thus becomes $\tau_i = [f_x \ f_y \ l_x f_y - l_y f_x]^\top$. Here, we have neglected heave, roll and pitch motion (we assume that the vessel is self-stabilizing in these DOFs).

Figure 1 shows a sketch of the CS Enterprise with its two Voith-Schneider propellers and bow tunnel thruster. Relevant data for the thrusters are found in Table 1.1. Note that the angles are defined by 0 pointing towards the bow with a clockwise rotation giving a positive angle.

The generalized thrust loads τ are related to the thruster setpoints by

$$\tau = \sum_{i=1}^m \tau_i = B(\alpha)Ku\tag{3}$$

where $B(\alpha)$ is the thrust configuration matrix, $\alpha = [\alpha_1 \ \alpha_2 \ \alpha_3]^\top$, $K = \text{diag}(k_1, k_2, k_3)$ is a scaling gain matrix and u the (normalized) thruster forces. The normalized thrust vector must satisfy $u_i \in [0, 1]$

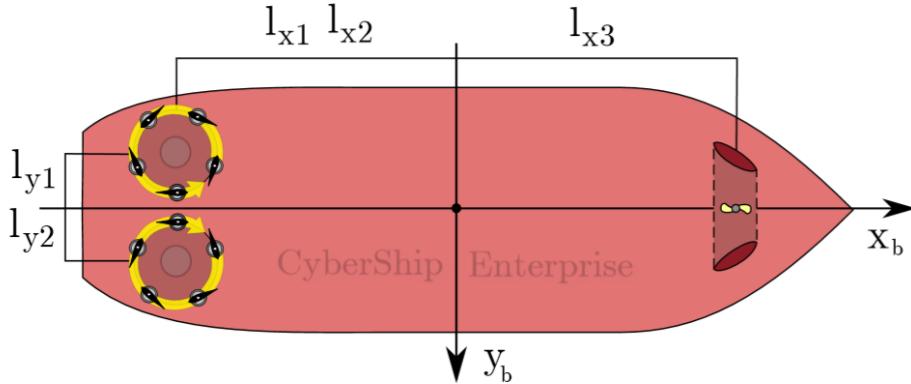


Figure 1: Ship with positive directions and thruster locations

Thruster	L_x [m]	L_y [m]	α [rad]	F_{max} [N]	α_{min} [rad]	α_{max} [rad]
1	-0.4574	-0.055	α_1	1.03	$-\pi$	π
2	-0.4574	0.055	α_2	1.03	$-\pi$	π
3	0.3875	0	$\frac{\pi}{2}$	2.629	-	-

Table 1: Thruster data.

for the VSPs, while for the bow thruster $u_i \in [-1, 1]$. Assuming the linear map $F_i = k_i u_i$ for the thrust produced by thruster i , it follows from these u_i and the tabulated max thruster forces that $k_1 = k_2 = 1.03$ N and $k_3 = 2.629$ N.

1.1.1 Thruster allocation in polar coordinates

For the thruster allocation in polar coordinates, returning to (3), and noting that $Ku = f$ is the vector containing the force produced by each thruster i having orientation α_i , we have that

$$\tau = \sum_{i=1}^m \begin{bmatrix} f_{x,i} \\ f_{y,i} \\ l_{x,i}f_{y,i} - l_{y,i}f_{x,i} \end{bmatrix} = B(\alpha)f.$$

The thrust configuration matrix is therefore

$$B(\alpha) = \begin{bmatrix} c(\alpha_1) & c(\alpha_2) & c(\alpha_3) \\ s(\alpha_1) & s(\alpha_2) & s(\alpha_3) \\ l_{x,1}s(\alpha_1) - l_{y,1}c(\alpha_1) & l_{x,2}s(\alpha_2) - l_{y,2}c(\alpha_2) & l_{x,3}s(\alpha_3) - l_{y,3}c(\alpha_3) \end{bmatrix},$$

where we have denoted $\cos(\cdot)$ and $\sin(\cdot)$ by $c(\cdot)$ and $s(\cdot)$, for compactness.

If we assume fixed angles, which in later tasks would be 0 or π , and a commanded thrust τ_{cmd} , (3) implies that we could calculate the thruster setpoints u from

$$u = K^{-1}B^{-1}\tau_{\text{cmd}}, \quad (4)$$

since B would be constant.

Note that if B was not square, e.g if the vessel was over- or under-actuated, its inverse would not exist. In this case, B would be replaced by its (Moore-Penrose) pseudo-inverse in the above equation. A singular configuration of B is also possible, making it noninvertible. This is the case if

$$|B| = 0. \quad (5)$$

Now

$$\begin{aligned} |B| &= l_{x,3} \cos \alpha_1 \sin \alpha_2 + \cos \alpha_2 [l_{x,1} \sin \alpha_1 - l_{y,1} \cos \alpha_1] \\ &\quad - \cos \alpha_1 [l_{x,2} \sin \alpha_2 - l_{y,2} \cos \alpha_2] - l_{x,3} \sin \alpha_1 \cos \alpha_2 \\ &= [l_{x,3} - l_{x,2}] \cos \alpha_1 \sin \alpha_2 + [l_{x,1} - l_{x,3}] \sin \alpha_1 \cos \alpha_2 + [l_{y,2} - l_{y,1}] \cos \alpha_1 \cos \alpha_2 \\ &= [l_{x,3} - l_{x,1}] \cos \alpha_1 \sin \alpha_2 + [l_{x,1} - l_{x,3}] \sin \alpha_1 \cos \alpha_2 + 2l_{y,2} \cos \alpha_1 \cos \alpha_2 \end{aligned}$$

This is zero under the conditions

$$\cos \alpha_1 \sin \alpha_2 = \sin \alpha_1 \cos \alpha_2 \quad (6)$$

$$\cos \alpha_1 \cos \alpha_2 = 0. \quad (7)$$

If we first assume that $\cos \alpha_1 = 0$, this implies that $\sin \alpha_1 \cos \alpha_2 = 0$. Let us now assume $\cos \alpha_2 \neq 0$. Therefore, we must have $\sin \alpha_1 = 0$, which is impossible by our first assumption. Hence both $\cos \alpha_1$ and $\cos \alpha_2$ must be zero to attain a singular configuration of B , i.e these angles must be odd multiples of $\pm\pi/2$. The physical interpretation of this is that in a singular configuration, the vessel is under-actuated in the working space - if the C/S Enterprise's thrusters are in a singular configuration as indicated above, the actuators cannot control surge motion.

1.1.2 Thruster allocation in rectangular coordinates

For the thrust mapping in rectangular coordinates, we can instead choose to decompose each thruster into its x - and y - components and work in rectangular coordinates. This gives us the thrust vector

$$f = \begin{bmatrix} f_{1,x} & f_{1,y} & f_{2,x} & f_{2,y} & f_3 \end{bmatrix}^\top. \quad (8)$$

For the VSPs, we can then calculate the corresponding angle α_i as

$$\alpha_i = \tan^{-1} \frac{f_{i,y}}{f_{i,x}}, \quad (9)$$

while the thrust input $u_i \in [0, 1]$ becomes

$$u_i = \left| \frac{\sqrt{f_{i,x}^2 + f_{i,y}^2}}{F_{i,max}} \right|. \quad (10)$$

For the bow tunnel thruster, u is simply $\frac{f_3}{F_{3,max}}$

By decomposing the thrust vector of a VSP and regarding it as two independent thrusters, we can extend the thrust configuration matrix into a 3×5 matrix that is not dependent on α . This gives us

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ -l_{1,y} & l_{1,x} & -l_{2,y} & l_{2,x} & l_{3,x} \end{bmatrix}, \quad (11)$$

so we can write the thrust mapping as

$$\tau = BKu = Bf. \quad (12)$$

where now $K = \text{diag}(k_1, k_1, k_2, k_2, k_3)$.

The rows in B represent the force coefficients in surge, sway and yaw. The columns are the coefficients for force and moment from each thruster. Since (12) defines three equations in 5 unknown quantities, it follows that the system of equations is underdetermined - i.e, having an infinite number of solutions. Solving the thrust allocation problem for a given commanded force f_{cmd} can in this case be done by posing it as an optimization problem. We can try finding its minimum norm solution, found from the minimization problem

$$\min \quad f_{cmd}^\top f_{cmd} = |f_{cmd}|^2 \quad (13a)$$

$$\text{s.t.} \quad Bf_{cmd} = \tau \quad (13b)$$

An example for finding the optimal solution for f_{cmd} is for instance *Quadratic Programming* which is an algorithm that solves a special type of mathematical optimization problem. This kind of optimization algorithm will take care of the different constraints which occurs in the thrust allocation problem. Specially a good optimization tool when a azimuth propeller or a Voith Schneider propeller are applied to the ship. An example is included in the appendix.

1.2 Simulink and model scale testing

For the practical part, a Simulink model is first implemented and then tested and tuned via HIL. Finally, the model scale test is done in the MC lab. The thruster allocation in both polar and rectangular coordinates are involved.

The Simulink diagrams used in HIL estimation and scale experiment in case A are shown in Figure 2 to Figure 8. In Figure 7, we use joystick *PosXLeft* to control the surge force and *PosYLeft* to control the yaw moment. In Figure 8, we use joystick *PosXLeft* to control the surge force, use *PosYLeft* to control the sway force and use *L2* and *R2* to control the yaw moment. We use *L2* and *R2* together because we want yaw moment changes in $[-1, 1]$ those two bottoms change linearly from -1(not pressing) to 0(pressing to the end). In order to reach the maximum velocity of three thruster, two gain blocks are added in those force conversion block, and velocity constraints are added in thrust allocation in Figure 6. Besides, a series of blocks are used in the upper part of Simulink diagram shown in Figure 2. *ArrowUp* and *ArrowDo* bottoms are used to switch between two force modes. The same function is also used to switch between polar coordinates (*ArrowLeft*) and rectangular coordinates (*ArrowRig*).

A graphical user interface used in HIL estimation and scale experiment is shwon in Figure 20. The information about thruster velocities, VPS thrusters angles, mode of force and mode of coordinates are included in the interface. The diagram shows the real-time data of the vessel position $[x, y, \psi]$. And a data logging control is used to log data.

1.3 Discussion of Results

From the result of HIL simulation in Figure 26 and 27, the model can follow command almost perfectly. And the result from Lab in Figure 28, 29, 30, 31, 32, 33 shows that the ship can follow the joystick's command properly under most condition. However, flaws of the model are also evident—The ship cannot run perfectly under the control of joystick. By comparing the testing in HIL box and scale model, it is clear that the main problem lies in the deviation between the simplified model and the real model, such as the asymmetry of the real model and the property of thrusters. Therefore, in the scale model testing, although the desired path is surge motion, the deviation leads to a little sway and yaw motion.

What is more, in the scale testing, the sudden change exists in rotation angle under both polar and rectangular coordinates. Also, for the thruster allocation in polar coordinates, the ship cannot rotate due to the flaw of thruster allocation matrix. Given $\alpha_1 = 0$ and $\alpha_2 = 0$, when the desired motion is

rotation, the first and second raw of the B matrix becomes $[1, 1, 0]$ and $[0, 0, 1]$, which means that the force of 3 thrusters must be zero. This is impossible to make the ship model rotate. In the following cases, rectangular coordinates is the default coordinates in thrust allocation.

2 Part B: DP Observer Design - State estimation and filtering

2.1 Theoretical development

The low-speed dynamics of the CS Enterprise are

$$\begin{aligned}\dot{\eta} &= R(\psi)v \\ M\dot{v} &= -Dv + R(\psi)^\top b + \tau \\ \dot{b} &= 0\end{aligned}$$

The position and heading vector, η , and the bias vector are given in NED-frame, while the velocity vector v and the body forces τ are defined in the ship's body frame. For position and velocity estimates $\hat{\eta}$ and \hat{v} , let us define the estimation errors as

$$\begin{aligned}\bar{\eta} &= \eta - \hat{\eta} \\ \bar{v} &= v - \hat{v}.\end{aligned}$$

The classical observer for the low-speed dynamics is made from copying the dynamics and adding injection terms, under the assumption that only position measurements are available:

$$\begin{aligned}\dot{\hat{\eta}} &= R(\psi)\hat{v} + L_1\bar{\eta} \\ M\dot{\hat{v}} &= -D\hat{v} + R(\psi)^\top \hat{b} + \tau + R(\psi)^\top L_2\bar{\eta}\end{aligned}$$

The matrices L_1 and L_2 are assumed to be diagonal matrices. If we assume no bias, the closed loop error dynamics thus become

$$\begin{aligned}\dot{\bar{\eta}} &= R(\psi)\bar{v} - L_1\bar{\eta} \\ M\dot{\bar{v}} &= -D\bar{v} - R(\psi)^\top L_2\bar{\eta}\end{aligned}$$

For $D + D^T > 0$, the equilibrium $(\bar{\eta}, \bar{v}) = 0$ is UGES. We can prove this by considering the Lyapunov function:

$$p_m|x|^2 \leq V(x) = x^\top Px \leq p_M|x|^2 \quad (15)$$

Where $x = [\bar{\eta}^\top \bar{v}^\top]^\top$ and $P = P^T = \text{diag}[L_2, M]$. p_m and p_M denote the minimum and maximum eigenvalues of matrix P . The function $V(x)$ is positive definite if L_2 and M are positive definite. Then we can take the time derivative of $V(x)$ along with the two equations from exercise 1.2. This will give us

$$\dot{V} = -\bar{\eta}^T [L_1 L_2 + L_2 L_1] \bar{\eta} - \bar{v}^T (D + D^T) \bar{v} \quad (16)$$

Now, for the conditions $D + D^T > 0$ and $L_1 L_2 + L_2 L_1 > 0$, \dot{V} is negative definite, and the equilibrium is UGES. The assumption that $(D + D^\top) > 0$ is reasonable because the hydrodynamic damping matrix D is strictly positive. This is due to the connection between the damping coefficients and the energy radiated away from the vessel, the latter of which is nonnegative.

Returning to the observer design, we consider the case when the bias b is nonzero. For a constant bias, we extend the observer (14) by adding the equation

$$\dot{\bar{b}} = L_3 \bar{\eta}. \quad (17)$$

This yields the new error dynamics

$$\dot{\bar{\eta}} = R(\psi) \bar{v} - L_1 \bar{\eta} \quad (18a)$$

$$M \dot{\bar{v}} = -D \bar{v} + R(\psi)^T \bar{b} - K_2 R(\psi)^T L_2 \bar{\eta} \quad (18b)$$

$$\dot{\bar{b}} = -L_3 \bar{\eta}. \quad (18c)$$

For the equilibrium $(\bar{\eta}, \bar{v}, \bar{b}) = 0$, we now show UGES. Let the Lyapunov function be

$$\begin{aligned} V &= \bar{x}^\top P \bar{x} \\ &= \begin{bmatrix} \bar{\eta}^\top & \bar{v}^\top & \bar{b}^\top \end{bmatrix} \begin{bmatrix} L_2 & 0 & -I \\ 0 & M & 0 \\ -I & 0 & L_3^{-1} L_1 \end{bmatrix} \begin{bmatrix} \bar{\eta} \\ \bar{v} \\ \bar{b} \end{bmatrix}, \end{aligned}$$

and let L_3 be a diagonal matrix with L_1, L_3 commutative. The time derivative of the Lyapunov function is

$$\dot{V} = \bar{x}^\top P \dot{\bar{x}} + \dot{\bar{x}}^\top P \bar{x} \quad (19)$$

Now, from (18) we have that

$$\dot{\bar{x}} = \begin{bmatrix} -L_1 & R & 0 \\ -M^{-1} R^\top L_2 & -M^{-1} D & R^\top \\ -L_3 & 0 & 0 \end{bmatrix} \bar{x} = \Omega \bar{x},$$

$$\begin{aligned} \Rightarrow \quad \bar{x}^\top P \dot{\bar{x}} &= \bar{x}^\top \begin{bmatrix} L_2 & 0 & -I \\ 0 & M & 0 \\ -I & 0 & L_3^{-1}L_1 \end{bmatrix} \begin{bmatrix} -L_1 & R & 0 \\ -M^{-1}R^\top L_2 & -M^{-1}D & M^{-1}R^\top \\ -L_3 & 0 & 0 \end{bmatrix} \bar{x} \\ &= \bar{x}^\top \begin{bmatrix} -L_2L_1 + L_3 & L_2R & 0 \\ -R^\top L_2 & -D & R^\top \\ L_1 - L_3^{-1}L_1L_3 & -R & 0 \end{bmatrix} \bar{x} \\ &= \bar{x}^\top \begin{bmatrix} -L_2L_1 + L_3 & L_2R & 0 \\ -R^\top L_2 & -D & R^\top \\ 0 & -R & 0 \end{bmatrix} \bar{x}. \end{aligned}$$

In the same way we get

$$\begin{aligned} \dot{\bar{x}}^\top P \bar{x} &= \bar{x}^\top \Omega^\top P \bar{x} \\ &= \bar{x}^\top \begin{bmatrix} -L_1 & -L_2RM^{-1} & -L_3 \\ R^\top & -D^\top M^{-1} & 0 \\ 0 & RM^{-1} & 0 \end{bmatrix} \begin{bmatrix} L_2 & 0 & -I \\ 0 & M & 0 \\ -I & 0 & L_3^{-1}L_1 \end{bmatrix} \bar{x} \\ &= \bar{x}^\top \begin{bmatrix} -L_1L_2 + L_3 & -L_2R & 0 \\ R^\top L_2 & -D^\top & -R^\top \\ 0 & R & 0 \end{bmatrix} \bar{x} \end{aligned}$$

Inserting this into (19) yields

$$\dot{V} = \bar{x}^\top \begin{bmatrix} -(L_1L_2 + L_2L_1 - 2L_3) & 0 & 0 \\ 0 & -(D + D^\top) & 0 \\ 0 & 0 & 0 \end{bmatrix} \bar{x} \quad (20)$$

$$= -\bar{\eta}^\top [L_1L_2 + L_2L_1 - 2L_3] \bar{\eta} - \bar{v}^\top (D + D^\top) \bar{v}, \quad (21)$$

which is negative definite provided $L_1L_2 + L_2L_1 - 2L_3$ is positive definite and $D + D^\top$ positive definite. In this case UGES holds. Assuming $D + D^\top > 0$, stability can thus be ensured by taking L_1, L_2 to be e.g $I_{3 \times 3}$ and $L_3 = \frac{1}{2}I_{3 \times 3}$.

2.2 Simulink and model scale testing

Two observers implemented are shown in Figure 11 and 12. A manual switch is added in Figure 10 to change between to observers.

In Simulink part, two observers are compared. The results are shown in Figure 34, 35, 36 and 37. In HIL simulation, a graphical user interface is built. The observer parameters L_1, L_2, L_3 and be

directly tuned in the interface. We start with bias $b = 0$ and increase b while tuning until a stable estimate was achieved. Resulting parameter values are given in Table 4. Plots of real and estimated position for varying b are seen in Figures 38, 39 and 40.

Noise is added to the signal to test the performance of observer, shown in Figure 41 and 42. The variance of noise is up to 0.4.

The observer in Figure 13 is modified to include dead reckoning under loss-of-signal. This scenario is simulated and the estimate of observer is in Figure 43.

In model test, observer with bias is used and tested, with the result shown in Figure 44. The vessel enters the dead reckoning manually and the plot of estimated and real position is present in Figure 45.

The graphical user interfaces for observer without and with bias are shown in Figure 21 and Figure 22. Matrices are tuned in the graphical interface. We use *ArrowUp* and *ArrowDo* in joystick to switch between normal and dead reckoning. After testing, it is found that the observer does not work very well under loss-of-signal.

In scale model test, we use a graphical user interface shown in Figure 24. Manual dead reckoning switch is keep, and we add auto dead reckoning switch. When the vessel lose signal in 4 subsequent cycles, the observer will switch to dead reckoning mode automatically, the manual change is blocked and the indicator light changes from *Normal* to *Auto dead reckoning*. After detecting signal change, the observer and indicator light will automatically switch back.

2.3 Discussion of Results

In simulation, both observers are able to estimate η well with model bias $b = 0$. However, when model bias is added, they have different performance. The observer without bias always have deviation between estimation and real position, while the observer with bias is able to decrease the deviation.

The observer can filter out most of the noise. It can be found that increasing elements in matrices \mathbf{L}_1 and \mathbf{L}_2 the observer will have better performance, but too large \mathbf{L}_1 and \mathbf{L}_2 will bring noise into $\hat{\eta}$.

In terms of dead reckoning, the observer would slowly drift away from the real position, and finally would have stable error term. It has been noticed that large \mathbf{L}_2 relative to \mathbf{L}_1 can decrease the error, because large \mathbf{L}_2 leads to smaller estimation error in acceleration. Otherwise the error would become very large. The auto dead reckoning function works well in scale model test, the observer can switch between two mode as expected.

3 Part C: DP Feedback Control and Maneuvering

3.1 Theoretical development

3.1.1 Path parametrization

In the maneuvering problem, the desired parametrized trajectory is denoted as $\eta_d(s(t)) = [p_x \ p_y \ \psi_d]$, where $s(t)$ is an absolutely continuous function. The two first derivatives of η_d with respect to s are

$$\eta_d^s = \begin{bmatrix} p_x^s & p_y^s & \psi_d^s \end{bmatrix}, \quad \eta_d^{2s} = \begin{bmatrix} p_x^{2s} & p_y^{2s} & \psi_d^{2s} \end{bmatrix}$$

The desired heading is tangential to the desired path, hence we have

$$\begin{aligned} \psi_d &= \arctan \frac{p_y^s}{p_x^s}, & \psi_d^s &= \frac{p_x^s p_y^{2s} - p_y^s p_x^{2s}}{(p_x^s)^2 + (p_y^s)^2}, \\ \psi_d^{2s} &= \frac{p_x^s p_y^{3s} - p_y^s p_x^{3s}}{(p_x^s)^2 + (p_y^s)^2} - \frac{(p_x^s p_y^{2s} - p_y^s p_x^{2s})(2p_x^s p_x^{2s} + 2p_y^s p_y^{2s})}{((p_x^s)^2 + (p_y^s)^2)^2}. \end{aligned}$$

There are 4 methods described in the lecture notes to design the speed assignment, in order to solve the dynamic task. We choose the simplest one; using Tracking update law to design \dot{s} :

$$\begin{aligned} \dot{s} &= v_0 \\ U(s, t) &= u_{ref} / \sqrt{p_x^{s2} + p_y^{s2}} \\ U(s, t)^s &= -u_{ref} (p_x^s p_x^{2s} + p_y^s p_y^{2s}) / (p_x^{s2} + p_y^{s2})^{3/2}. \end{aligned}$$

For a straight line parametrization of $p_d(s) = (x_d(s), y_d(s))$ we can take:

$$p_x = (1-s)x_0 + sx_1 \quad p_y = (1-s)y_0 + sy_1,$$

where (x_0, y_0) is initial position and (x_1, y_1) is final position. With this choice the s -derivatives of p_d become

$$p_x^s = x_1 - x_0 \quad p_y^s = y_1 - y_0$$

with higher derivatives in s being identically zero.

An alternate, ellipsoidal parametrization would yield

$$\begin{aligned}
p_x &= c_x + r_x \cos 2\pi s & p_y &= c_y + r_y \sin 2\pi s \\
p_x^s &= -2\pi r_x \sin 2\pi s & p_y^s &= 2\pi r_y \cos 2\pi s \\
p_x^{2s} &= -4\pi^2 r_x \cos 2\pi s & p_y^{2s} &= -4\pi^2 r_y \sin 2\pi s \\
p_x^{3s} &= 8\pi^3 r_x \sin 2\pi s & p_y^{3s} &= -8\pi^3 r_y \cos 2\pi s \\
&\quad = -4\pi^2 p_x^s & &\quad = -4\pi^2 p_y^s
\end{aligned}$$

3.1.2 Kinematic model control design

Let us look at the kinematic model control design. A backstepping controller model is designed. For the kinematic model, the body-fixed position/heading error vector z_1 and its derivative is

$$\begin{aligned}
z_1 &= R(\psi)^\top (\eta - \eta_d(s)) \\
\implies \dot{z}_1 &= R(\psi)^\top (\dot{\eta} - \eta_d^s(s)\dot{s}) \\
&= v - R(\psi)^\top \eta_d^s(s)\dot{s}
\end{aligned} \tag{22}$$

We have the CLF

$$\begin{aligned}
V_1(\eta, s(t)) &= \frac{1}{2} z_1^\top z_1 \\
&= \frac{1}{2} [R^\top (\eta - \eta_d)]^\top R^\top (\eta - \eta_d) \\
&= \frac{1}{2} (\eta - \eta_d)^\top R R^\top (\eta - \eta_d) \\
&= \frac{1}{2} (\eta - \eta_d)^\top (\eta - \eta_d),
\end{aligned} \tag{23}$$

and the derivative of (23) is

$$\begin{aligned}
\dot{V}_1 &= \frac{1}{2} \dot{z}_1^\top z_1 + \frac{1}{2} z_1^\top \dot{z}_1 \\
&= \frac{1}{2} [v - R^\top \eta_d^s \dot{s}]^\top R^\top (\eta - \eta_d) + \frac{1}{2} [R^\top (\eta - \eta_d)]^\top (v - R^\top \eta_d^s \dot{s}) \\
&= [R^\top (\eta - \eta_d)]^\top (v - R^\top \eta_d^s \dot{s}) \\
&= z_1^\top (v - R^\top \eta_d^s \dot{s}).
\end{aligned} \tag{24}$$

Assuming that $v = \alpha_1$ is our control law, with

$$\alpha_1(\eta, s, t) = -K_p z_1 + R(\psi)^\top \eta_d^s U_s(s, t), \quad K_p = K_p^\top > 0,$$

and inserting this in (24) we get

$$\begin{aligned}\dot{V}_1 &= z_1^\top (-K_p z_1 + R^\top \eta_d^s U_s(s, t) - R^\top \eta_d^s \dot{s}) \\ &= z_1^\top (-K_p z_1 + R^\top \eta_d^s (U_s - \dot{s}))\end{aligned}\quad (25)$$

Expanding (25) yields

$$\begin{aligned}\dot{V}_1 &= -z_1^\top K_p z_1 + z_1^\top R^\top \eta_d^s (U_s - \dot{s}) \\ &= -z_1^\top K_p z_1 + [R^\top (\eta - \eta_d)]^\top R^\top \eta_d^s (U_s - \dot{s}) \\ &= -z_1^\top K_p z_1 + (\eta - \eta_d)^\top R R^\top \eta_d^s (U_s - \dot{s}) \\ &= -z_1^\top K_p z_1 + (\eta - \eta_d)^\top \eta_d^s (U_s - \dot{s}).\end{aligned}\quad (26)$$

Noting that

$$\begin{aligned}V_1^s &= (\eta - \eta_d)^\top \frac{d}{ds}(\eta - \eta_d) \\ &= -(\eta - \eta_d)^\top \eta_d^s,\end{aligned}\quad (27)$$

it follows that (26) can be written as

$$\dot{V}_1 = -z_1^\top K_p z_1 - V_1^s (U_s - \dot{s}), \quad (28)$$

which gives us the inequality

$$\dot{V}_1 \leq -\lambda_{min}(K_p)|z_1|^2 - V_1^s (U_s - \dot{s}).$$

The expression for V_1^s changes according to chosen parametrization. We have

$$\begin{aligned}V_1^s &= -(\eta - \eta_d)^\top \eta_d^s \\ &= -\begin{bmatrix} x - x_d & y - y_d & \psi - \psi_d \end{bmatrix} \begin{bmatrix} p_x^s \\ p_y^s \\ \psi_d^s \end{bmatrix} \\ &= -[p_x^s(x - x_d) + p_y^s(y - y_d) + \psi_d^s(\psi - \psi_d)] \\ &= -[p_x^s(x - x_d) + p_y^s(y - y_d) + \frac{p_x^s p_y^{2s} - p_y^s p_x^{2s}}{(p_x^s)^2 + (p_y^s)^2}(\psi - \psi_d)]\end{aligned}$$

For the straight line path parametrization, the factor $\frac{p_x^s p_y^{2s} - p_y^s p_x^{2s}}{(p_x^s)^2 + (p_y^s)^2}$ vanishes since second order derivatives are 0, while in the ellipsoidal parametrization we get

$$\begin{aligned}\frac{p_x^s p_y^{2s} - p_y^s p_x^{2s}}{(p_x^s)^2 + (p_y^s)^2} &= \frac{-2\pi r_x \sin(2\pi s)(-4\pi^2 r_y \sin(2\pi s)) - 2\pi r_y \cos(2\pi s)(-4\pi^2 r_x \cos(2\pi s))}{(-2\pi r_x \sin(2\pi s))^2 + (2\pi r_y \cos(2\pi s))^2} \\ &= \frac{2\pi r_x r_y}{r_x^2 \sin^2(2\pi s) + r_y^2 \cos^2(2\pi s)}.\end{aligned}$$

3.1.3 Update law

We now look at some of the update laws described in the lectures. First, the tracking update law

$$\dot{s} = U_s(s, t)$$

obviously satisfies the dynamic task of the Maneuvering Problem,

$$\lim_{t \rightarrow \infty} |\dot{s} - U_s(s(t), t)| = 0$$

for all t . It is called the tracking update law because it forms a tracking problem where $s(t)$ just becomes a time signal generated by a ecosystem.

Then, the gradient update law,

$$\dot{s} = U_s(s, t) - \mu V_1^s(\eta, s), \quad \mu \geq 0$$

solves the Maneuvering Problem in the limit since

$$\begin{aligned} \dot{V}_1 &\leq -\lambda_{\min}(K_p)|z_1|^2 - V_1^s(U_s - \dot{s}) \\ &= -\lambda_{\min}(K_p)|z_1|^2 - V_1^s(U_s - U_s + \mu V_1^s) \\ &= -\lambda_{\min}(K_p)|z_1|^2 - \mu(V_1^s)^2 \\ &\leq 0 \end{aligned} \quad (29) \quad (30)$$

The name of the law comes from the gradient term $\frac{d}{ds}V_1$, which takes feedback from the gradient of Lyapunov function with respect to s . By the same consideration as in the gradient update law the modified gradient update law

$$\dot{s} = U_s(s, t) - \frac{\mu}{|\eta_d^s|} V_1^s, \quad (31)$$

also solves the maneuvering problem. This update law normalizes V_1^s since

$$\begin{aligned} \frac{\mu}{|\eta_d^s|} V_1^s &= \frac{\mu}{|\eta_d^s|} (\eta - \eta_d)^\top \eta_d^s \\ &= \mu \frac{(\eta_d^s)^\top}{|\eta_d^s|} (\eta - \eta_d) \end{aligned} \quad (32)$$

For the filtered gradient update law, define the new CLF

$$W_1 := V_1 + \frac{1}{2\lambda\mu} \omega_s^2.$$

Then, given the speed assignment error $\omega_s = U_s - \dot{s}$ and

$$\dot{s} = U_s - \omega_s \quad (33)$$

$$\dot{\omega}_s = -\lambda(\omega_s - \mu V_1^s), \quad (34)$$

we get by applying (19), (33) and (34) that

$$\begin{aligned} \dot{W}_1 &= \dot{V}_1 + \frac{1}{\lambda\mu}\omega_s\dot{\omega}_s \\ &= \dot{V}_1 + V_1^s\omega_s - \frac{1}{\mu}\omega_s^2 \\ &= -z_1^\top K_p z_1 - V_1^s(U_s - \dot{s}) + V_1^s(U_s - \dot{s}) - \frac{1}{\mu}\omega_s^2 \\ &= -z_1^\top K_p z_1 - \frac{1}{\mu}\omega_s^2 \\ &\leq -\lambda_{min}(K_p)|z_1|^2 - \frac{1}{\mu}|\omega_s|^2 \\ &\leq 0 \end{aligned}$$

The reason for this name is because the gradient term V_1^s is input to and filtered by a first-order lowpass filter before entering the pass speed dynamics \dot{s} .

Given the 3DOF DP control design

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

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

To simplify the different derivations in this task we define $\dot{R}(\psi) = rRS$ and $\dot{R}(\psi)^\top = -rSR^\top$. Letting $z_1 = R(\psi)^\top(\eta - \eta_d)$ be the position error and $z_2 = \nu - \alpha_1(\eta, s, t)$ be the error between ν and its desired control α_1 . Calculating \dot{V} now with z_2 and α_1 included in the equations. First we find

$$\begin{aligned} \dot{z}_1 &= \dot{R}(\psi)^\top[\eta - \eta_d] + R(\psi)^\top[\dot{\eta} - \dot{\eta}_d] \\ &= -rSR^\top(\eta - \eta_d) + R(\psi)^\top\dot{\eta} - R(\psi)^\top\dot{\eta}_d \\ &= -rSz_1 + z_2 + \alpha_1 - R(\psi)^\top\dot{\eta}_d \end{aligned}$$

Furthermore we know that $V_1 = \frac{1}{2}z_1^\top z_1$ which gives

$$\begin{aligned} \dot{V}_1 &= z_1^\top \dot{z}_1 \\ &= -z_1^\top rS z_1 + z_1^\top z_2 + z_1^\top \alpha_1 - z_1^\top R(\psi)^\top \dot{\eta}_d \\ &= z_1^\top z_2 + z_1^\top (\alpha_1 - R(\psi)^\top \dot{\eta}_d) \end{aligned}$$

$$= -z_1^\top K z_1 + z_1^\top z_2,$$

by choosing the desired control

$$\alpha_1 = -K_1 z_1 + R(\psi)^\top \dot{\eta}_d.$$

Now, to be able to calculate $\dot{\alpha}_1$ we redefine \dot{z}_1

$$\begin{aligned}\dot{z}_1 &= -rS z_1 + z_2 - K_1 z_1 + R^\top \dot{\eta}_d - R^\top \ddot{\eta}_d \\ &= -(K_1 + rS) z_1 + z_2\end{aligned}$$

Then we can calculate $\dot{\alpha}_1$

$$\begin{aligned}\dot{\alpha}_1 &= -K_1 \dot{z}_1 + R^\top \dot{\eta}_d + R^\top \ddot{\eta}_d \\ &= -K_1 \dot{z}_1 - rS R^\top \dot{\eta}_d + R^\top \ddot{\eta}_d \\ &= K_1 (K_1 + rS) z_1 - K_1 z_2 - rS R^\top \dot{\eta}_d + R^\top \ddot{\eta}_d\end{aligned}$$

then calculate \dot{z}_2

$$\dot{z}_2 = \dot{v} - \dot{\alpha}_1$$

Now we let $V_2(v, \eta, s, t) = V_1(\eta, s, t) + \frac{1}{2} z_2^\top M z_2$ be step 2 CLF. We have

$$\begin{aligned}M \dot{z}_2 &= M \dot{v} - M \dot{\alpha}_1 \\ &= M \dot{v} + M K_1 \dot{z}_1 + M r S R^\top \dot{\eta}_d - M R^\top \ddot{\eta}_d \\ &= -(C + D)v + \tau + R^\top b - M \dot{\alpha}_1 \\ &= -(C + D)(z_2 + \alpha_1) + \tau + R^\top b - M \dot{\alpha}_1\end{aligned}$$

and given in the assignment paper

$$V_2 = V_1 + \frac{1}{2} z_2^\top M z_2 + \frac{1}{2} \tilde{b}^\top K_0^{-1} \tilde{b}$$

Then differentiate V_2

$$\begin{aligned}\dot{V}_2 &= -z_1^\top K_1 z_1 + z_2^\top (-(C + D)(z_2 + \alpha_1) + \tau + R^\top b - M \dot{\alpha}_1) + b^\top K_0^{-1} \dot{b} \\ &= -z_1^\top K_1 z_1 + z_2^\top (-(C + D)\alpha_1 + \tau + R^\top b - M \dot{\alpha}_1 + z_1) - z_2^\top (C + D) z_2 + \tilde{b}^\top K_0^{-1} (\dot{b} - \hat{b}) \\ &= -z_1 K_1 z_1 + z_2^\top (z_1 - (C + D)\alpha_1 + \tau + R^\top \hat{b} - M \dot{\alpha}_1) - z_2^\top (C + D) z_2 + z_2^\top R^\top \tilde{b} - \tilde{b} K_0^{-1} \dot{b}\end{aligned}$$

Purposing a control law for τ that renders \dot{V}_2 negative definite. Choosing first τ to be

$$\tau = -z_1 + (C + D)\alpha_1 - R^\top \hat{b} + M\dot{\alpha}_1 - K_2 z_2$$

Also, choosing the \hat{b} since we included the bias in the calculations.

$$\hat{b} = K_0 R z_2$$

This will simplify the \dot{V}_2 as follows and making it negative definite

$$\dot{V}_2 = -z_1^\top K_1 z_1 - z_2^\top (K_2 + C + D) z_2$$

3.2 Simulink and model scale testing

The Simulink diagrams used in Case C are shown in Figure 14 to Figure 19. The vessel follows the two desired trajectories hence joystick is not used.

The graphical user interface in Case C is shown in Figure 25. We use it to observe the change of vessel positions and log data. Because the diagram function in the interface can show the position change over time only, it is hard to see the real trajectory of the vessel. Therefore, we use Simulink to tune control parameters , which are present in Table 5.

Straight line maneuvering was simulated in Simulink, with two different initial positions. The result is presented in Figure 46 and 47. To prepare for the scale model test, the initial position of the ship is changed, the result in Figure 47 shows that the ship moves towards the desired path first then along the straight line. Then, the ellipsoid maneuvering is tested and shown in Figure 48. The parameter μ is tuned to be very small due to large position tracking error.

Both straight line maneuvering and ellipsoid maneuvering was performed in the model scale test, results of which are seen in figures 49 and 50, respectively.

3.3 Discussion of Results

In the HIL testing, the ship is able to follow the desired straight line perfectly. Even when the initial position is not in the desired path. But for the ellipsoid maneuvering, the result shows a little deviation from the desired path, while we notice that it can be reduced if the speed dynamics slows down, which enables the ship to follow the path better.

In the model scale testing, there is a small deviation between desired position and measured ship position. Since the result looks similar to the result in HIL simulation, it is reasonable to say that for

the maneuvering problem, the main deviation relies on the lack of propulsion. Also in model test, lost-of-signal happens from time to time, and it is hard to test the ship where the ship can always have the position signal. The lost-of-signal would result in poor performance in maneuvering, for example, the ship suddenly increases the propulsion.

4 Conclusion

The project is finished with theoretical derivation, simulation design based on simulink in Matlab, HIL testing and model scale testing.

In case A, the advantages and disadvantages of 2 kinds of thruster allocation in polar and rectangular coordinates are discussed. For the later tasks, the rectangular coordinate is chosen since the angle of thrusters can be changed to greatly fulfill the control requirement.

In case B, the observer performs well after tuning L2 to be much larger than L1. This is because a larger L2 can reduce the deviation of acceleration. Also, the bias and noise is also implemented in the observer and parameters are tuned respectively.

In case C, the task of maneuvering is finished well under both straight line and ellipsoid path. The small deviation exists due to the weakness of thrusters.

A MATLAB Code

Listing 1: Code snippet showing how to calculate the corresponding u_{cmd}

```
1 clear all;
2 clc;
3
4 % tau
5 tau1 = [1 ; 1; 0.5];
6 tau2 = [2 ; 0; 0.5];
7
8 %% Allocation with thrust vectors in rectangular coordinates
9 B_rec = [1 0 1 0 0;
10      0 1 0 1 1;
11      0.055 -0.4574 -0.055 -0.4574 0.3875];
12
13 K_rec = diag([1.03,1.03,1.03,1.03,2.629]);
14
15 % f_cmd
16 u1 = pinv(B_rec*K_rec)*tau1;
17 u2 = pinv(B_rec*K_rec)*tau2;
18
19 u_rec1 = [(u1(1)^2+u1(2)^2)^0.5 (u1(3)^2+u1(4)^2)^0.5 u1(5)];
20 u_rec2 = [(u2(1)^2+u2(2)^2)^0.5 (u2(3)^2+u2(4)^2)^0.5 u2(5)];
21
22 %% Allocation with thrust vectors in polar coordinates
23 K_pol = diag([1.03,1.03,2.629]);
24
25 % tau1 = [1 ; 1; 0.5];
26 alpha_11 = 0;
27 alpha_21 = pi;
28
29 B_pol1 = [ cos(alpha_11) cos(alpha_21) 0;
30            sin(alpha_11) sin(alpha_21) 1;
31            -0.4574*sin(alpha_11)+0.055*cos(alpha_11) -0.4574*sin(alpha_21)-0.055*cos(alpha_21)
32            0.3875];
33
34 u_pol1 = inv(K_pol)*inv(B_pol1)*tau1
35
36 % tau2 = [2 ; 0; 0.5];
37 alpha_12 = 0;
38 alpha_22 = pi;
39
40 B_pol2 = [ cos(alpha_12) cos(alpha_22) 0;
41            sin(alpha_12) sin(alpha_22) 1;
42            -0.4574*sin(alpha_12)+0.055*cos(alpha_12) -0.4574*sin(alpha_22)-0.055*cos(alpha_22)
43            0.3875];
44
45 u_pol2 = inv(K_pol)*inv(B_pol2)*tau2
```

The angles of VSPs are set to be 0 or π , depending on which give positive u . For $\tau = [1, 1, 0.5]^T$, the results are $u = [0.5046, 0.4746, 0.4302]^T$ in rectangular and $u = [1.4784, 0.5075, 0.3804]^T$ in polar coordinates. For $\tau = [2, 0, 0.5]^T$, the results are $u = [1.0354, 0.9887, 0.2239]^T$ in rectangular and $u = [5.3839, 3.4422, 0.0000]^T$ in polar coordinates. From the results, it seems that constraints are more easily to be violated using polar coordinates.

Listing 2: MATLAB code used in HIL simulation and model scale experiment

```

1  %% Thruster Allocation
2  % position
3  Lx1=-0.4574;
4  Lx2=-0.4574;
5  Lx3=0.3875;
6  Ly1=-0.055;
7  Ly2=0.055;
8  Ly3=0;
9
10 %% C1
11 B1 = [ 1 0 1 0 0;
12      0 1 0 1 1;
13      -Ly1 Lx1 -Ly2 Lx2 Lx3];
14 K1 = diag( [1.03, 1.03, 1.03, 1.03, 2.629]);
15 C1 = pinv(B1*K1);
16
17 %% C2
18 alpha1 = 0;
19 alpha2 = 0;
20 B2 = [ cos(alpha1) cos(alpha2) 0;
21         sin(alpha1) sin(alpha2) 1;
22         Lx1*sin(alpha1)-Ly1*cos(alpha1) Lx2*sin(alpha2)-Ly2*cos(alpha2) Lx3 ];
23 K2 = diag( [1.03, 1.03, 2.629]);
24 C2=inv(K2)*inv(B2);
25
26
27
28 %% Observor
29 % M-matrix :
30 m     = 14.11;
31 I_z   = 1.7600;
32 x_g   = 0.0375; % x_g   = 0.0460; % Table B.1
33
34 % Added mass
35 X_ud  = -2.0;
36 Y_vd  = -10.0;
37 N_vd  = -0.0;
38 Y_rd  = -0.0;
39 N_rd  = -1.0;
40
41 %% Total mass matrix

```

```

42 m_11 = m-X_ud;
43 m_22 = m-Y_vd;
44 m_23 = m*x_g-Y_rd;
45 m_32 = m*x_g-N_vd;
46 m_33 = I_z-N_rd;
47
48 M = [m_11 0 0
49 0 m_22 m_23
50 0 m_32 m_33];
51
52 inv_M = inv(M);
53
54 % Damping coefficients
55 X_u = -0.6555;
56 X_v = 0;
57 Y_v = -1.33;
58 Y_r = -7.250;
59 N_v = 0;
60 N_r = -1.900;
61
62
63 % Assembly of the damping matrix
64 % should be abs(u), abs(v) some places, only positive values?
65
66 d_11 = -X_u ;
67 d_22 = -Y_v;
68 d_33 = -N_r;
69
70 d_12 = -X_v;
71 d_23 = -Y_r;
72 d_32 = -N_v;
73
74 D = [d_11 0 0
75 0 d_22 d_23
76 0 d_32 d_33];
77
78 % Defining L's
79 L_1 = diag([2 2 2]);
80 L_2 = diag([12 12 12]);
81 L_3 = diag([1 1 1]);
82
83
84
85
86 %% Maneuvering
87 % maneuvering_straight_line
88 Umax_1 = 0.2;
89 pd_initial = [0;0]; %the initial position
90 pd_final = [5;-0.5]; % the final position
91 mu = 10e-8; % for tracking update law: mu = 0;% for unit-tangent gradient update law ,tune

```

```
92
93 % maneuvering_ellipsoid
94 Umax_2 = 0.1;
95 center = [2; 0];
96 R = [2; 1.5];
97
98 % tuning parameters
99 K_1 = 10*diag([1 1 1]);
100 K_2 = 10*diag([1 1 1]);
101 b = [0;0;0]; % constant bias
```

B System matrices & parameters

Rotation matrix:

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

Inertia matrix:

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_r \\ 0 & mx_g - Y_r & I_z - N_{\dot{r}} \end{bmatrix}$$

Coriolis matrix:

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

Damping matrix:

$$D(v) = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}$$

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

Table 2: CSE1 rigid body and added mass parameters

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

Table 3: CSE1 damping parameters

Parameter	Value
L_1	$diag([2, 2, 2])$
L_2	$diag([12, 12, 12])$
L_3	$diag([1, 1, 1])$

Table 4: Observer parameters

Parameter	Value
K_1	$diag([10, 10, 10])$
K_2	$diag([10, 10, 10])$
μ	10^{-8}

Table 5: Controller parameters

C Simulink diagrams

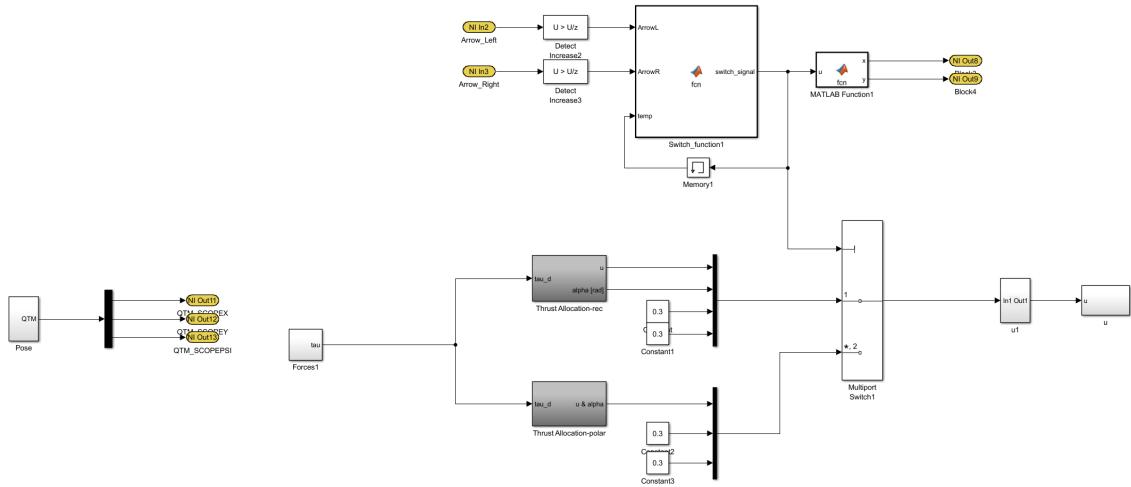


Figure 2: Simulink diagram in Case A

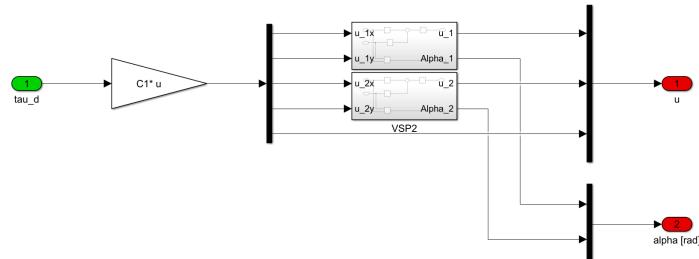


Figure 3: Simulink diagram of rectangular coordinates thrust allocation

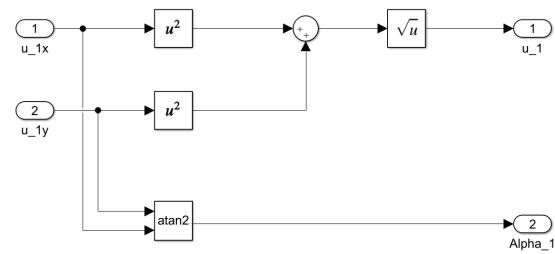


Figure 4: Simulink diagram of velocity and angle calculations in rectangular coordinates



Figure 5: Simulink diagram of polar coordinates thrust allocation

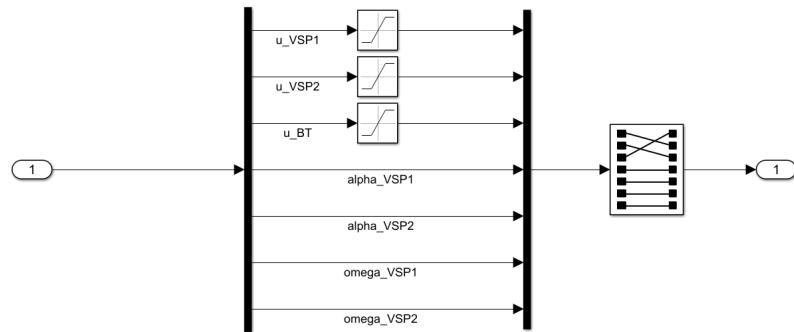


Figure 6: Simulink diagram of velocity constraints in thrust allocation

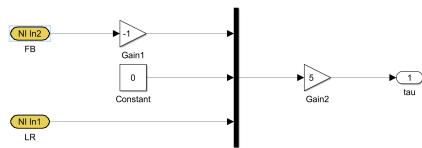


Figure 7: Simulink diagram of surge and yaw control by joystick

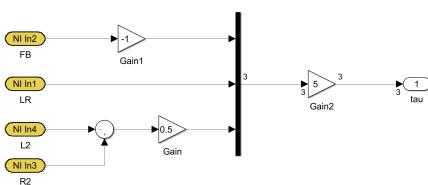


Figure 8: Simulink diagram of surge, sway and yaw control by joystick

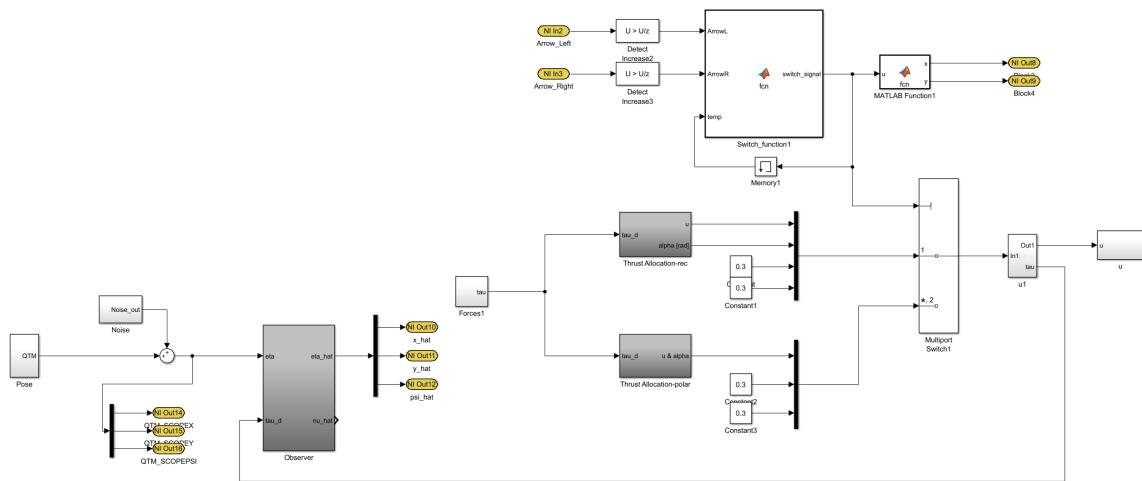


Figure 9: Simulink diagram in Case B

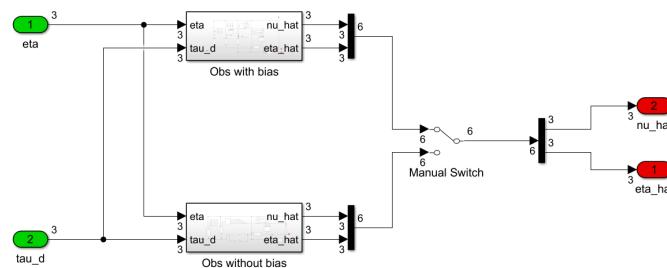


Figure 10: Simulink diagram of observer

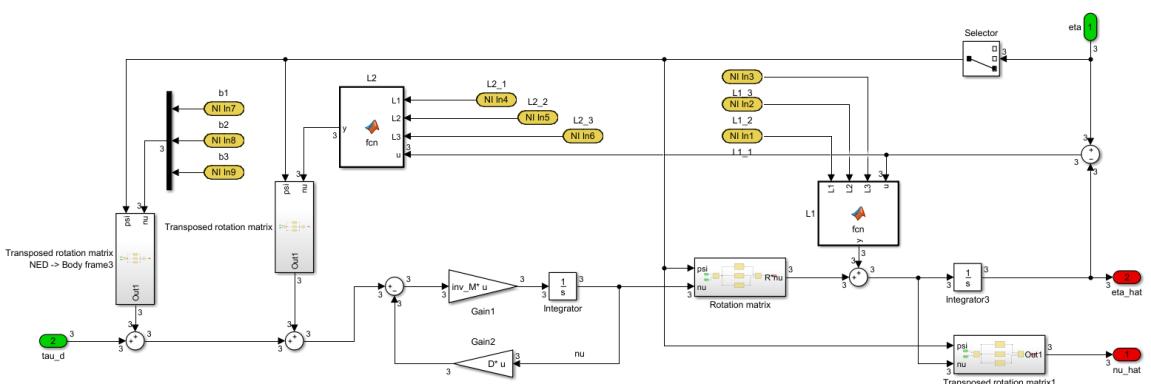


Figure 11: Simulink diagram of observer without bias

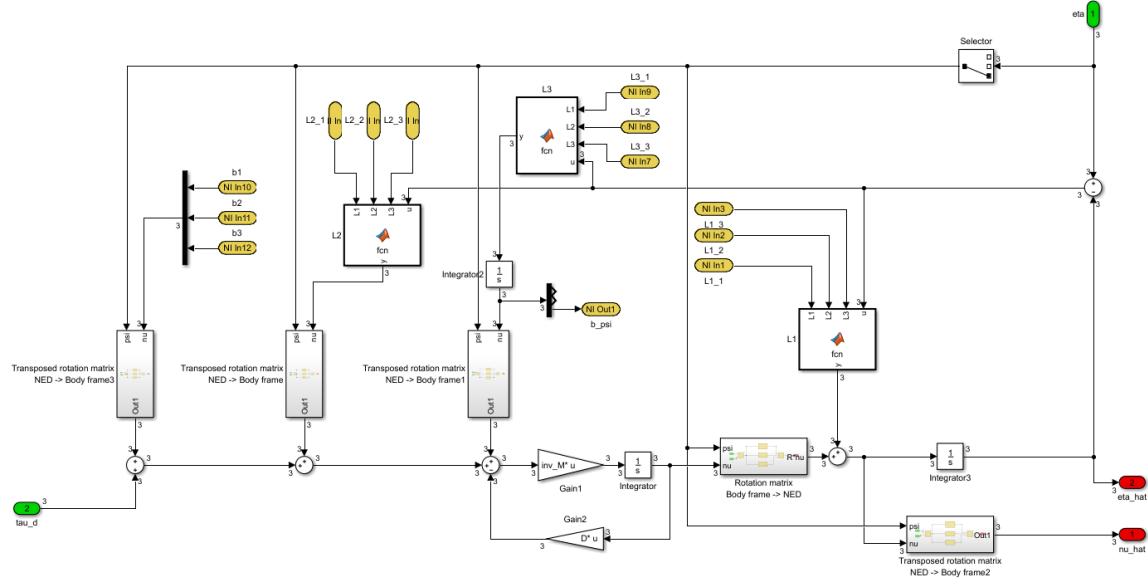


Figure 12: Simulink diagram of observer with bias

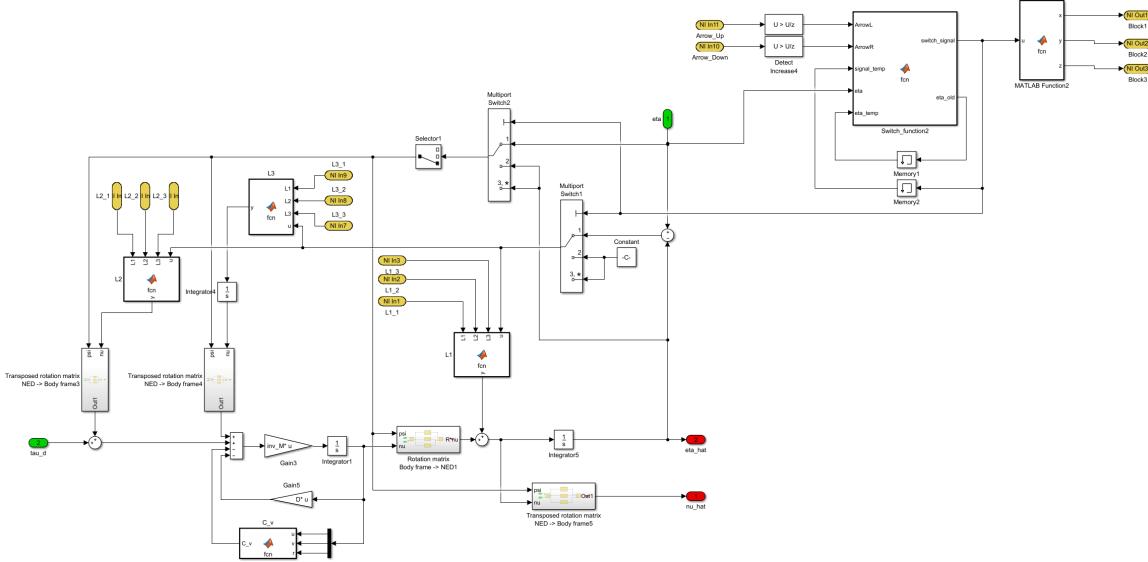


Figure 13: Simulink diagram in model scale experiment of observer with manual and auto dead reckoning

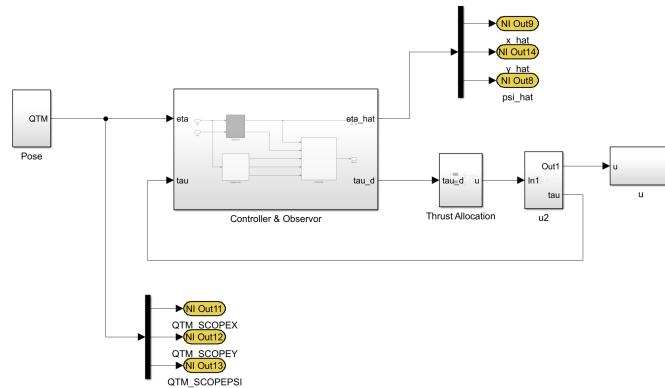


Figure 14: Simulink diagram in Case C

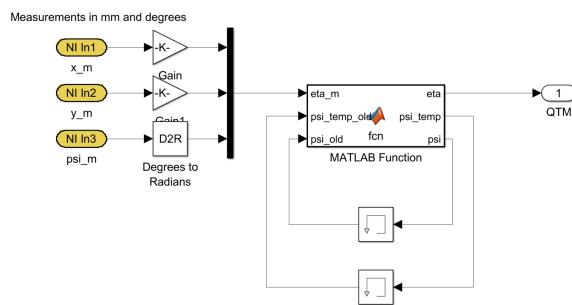


Figure 15: Simulink diagram in model scale experiment of pose

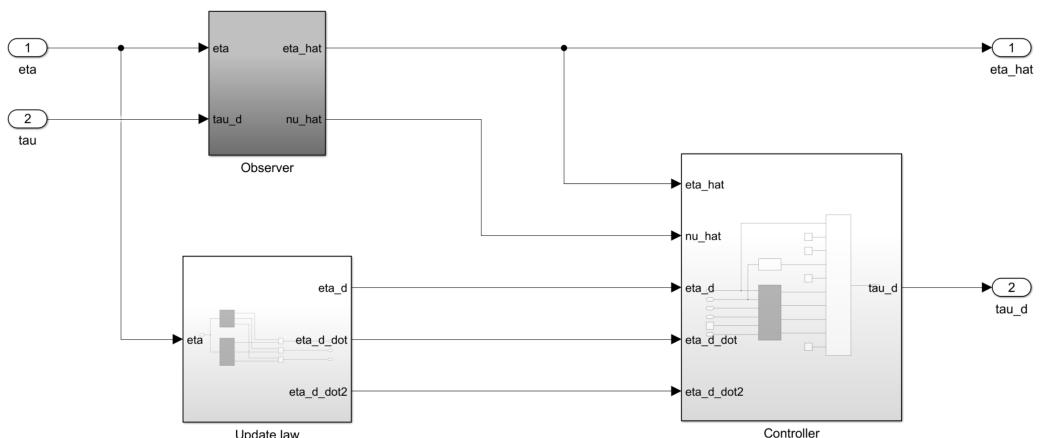


Figure 16: Simulink diagram of controller and Observer

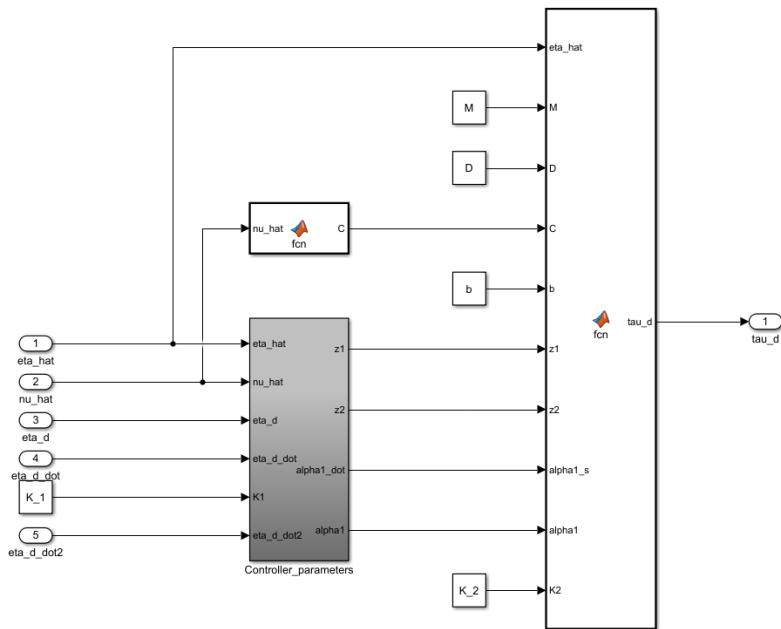


Figure 17: Simulink diagram of controller

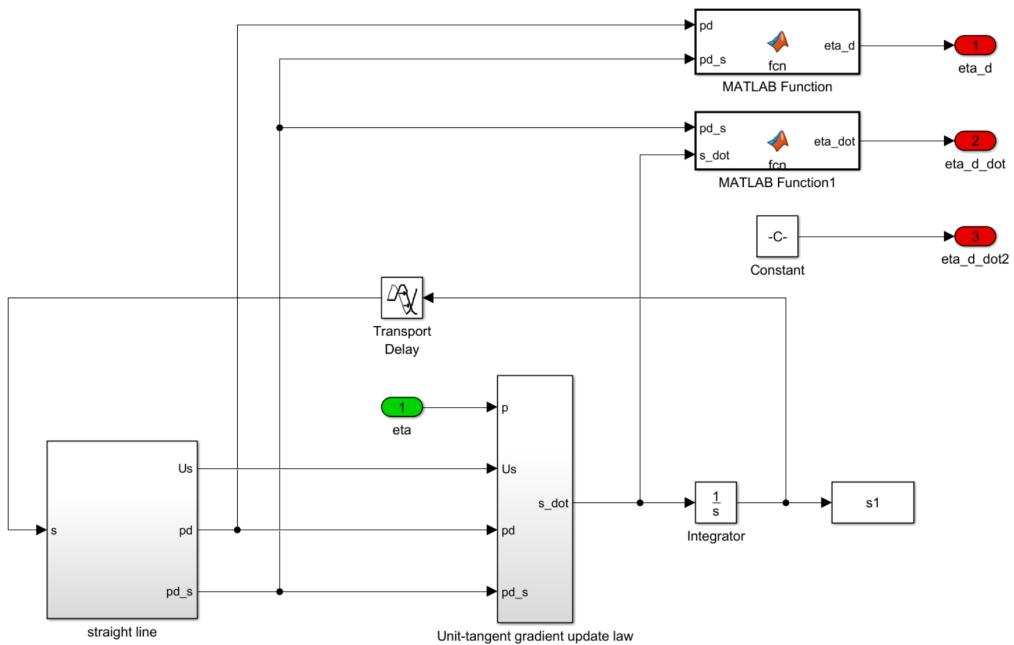


Figure 18: Simulink diagram of straight line update law

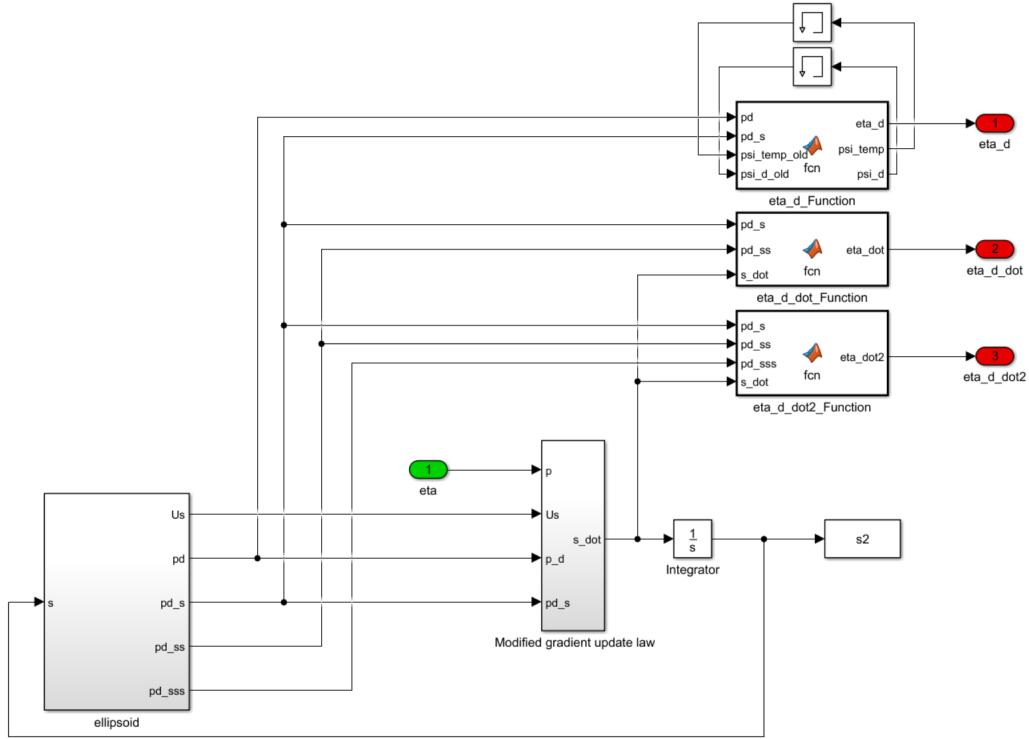


Figure 19: Simulink diagram of ellipsoid update law

D Graphical user interface



Figure 20: Graphical user interface in Case A

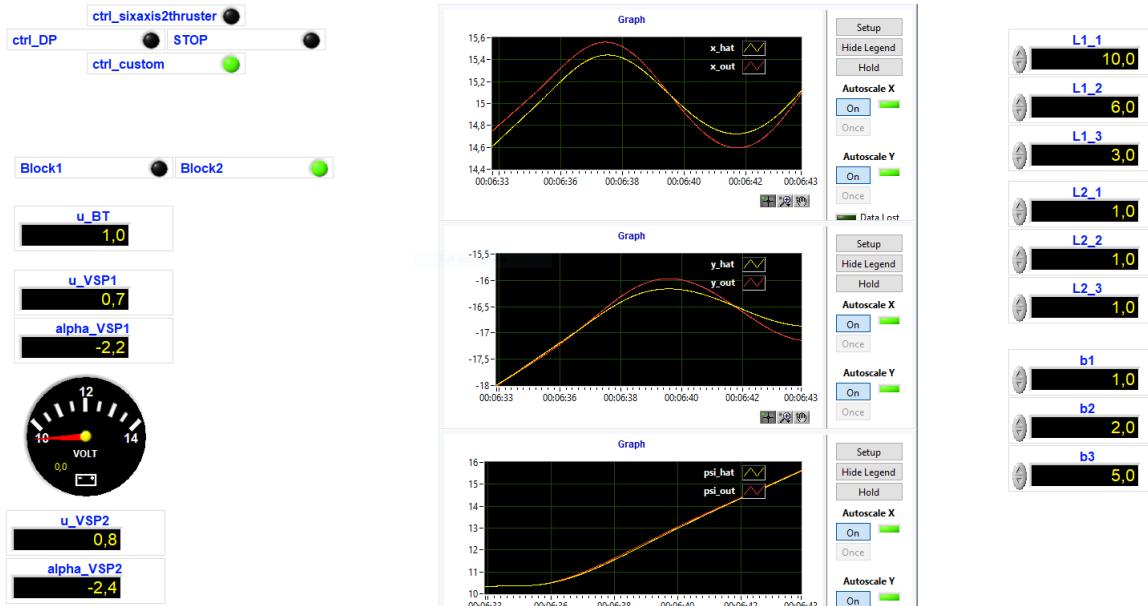


Figure 21: Graphical user interface in Case B of observer without bias

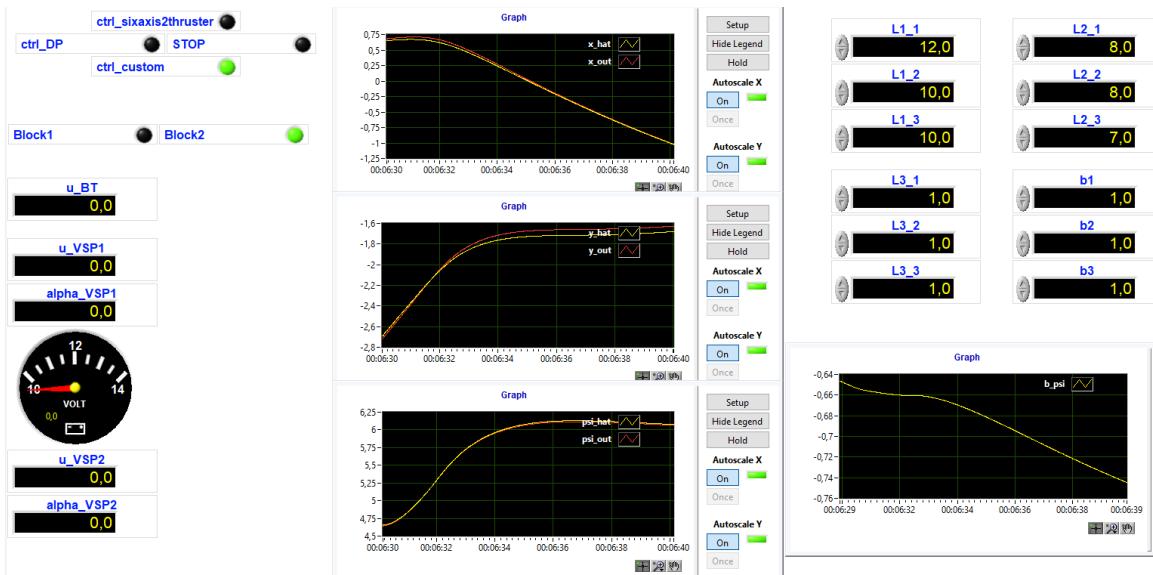


Figure 22: Graphical user interface in Case B of observer with bias



Figure 23: Graphical user interface in Case B HIL simulation of observer with dead reckoning

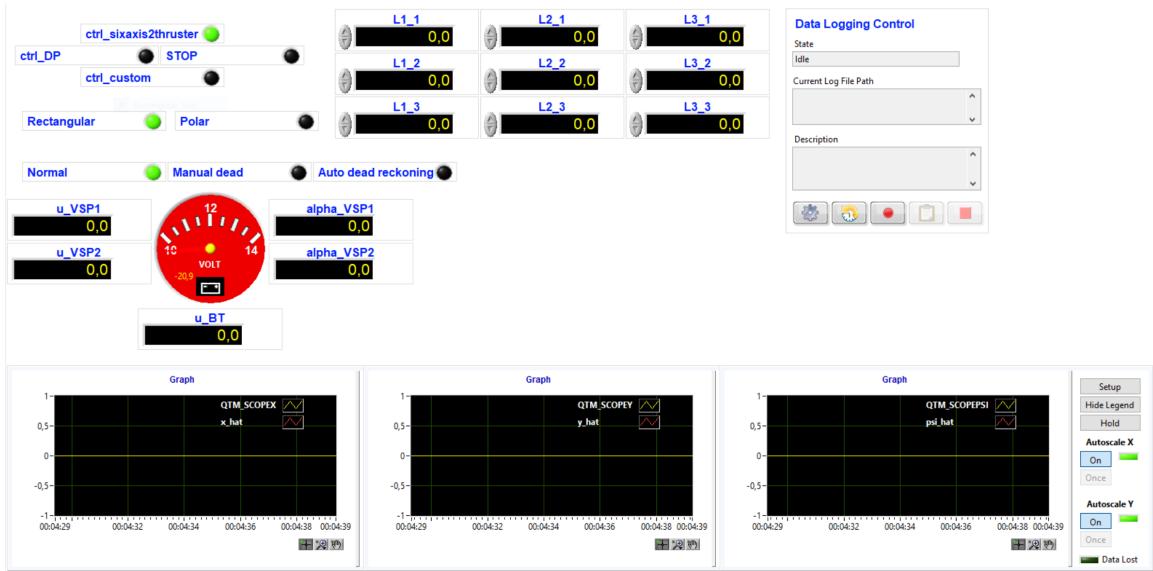


Figure 24: Graphical user interface in Case B model scale experiment of observer with manual and auto dead reckoning

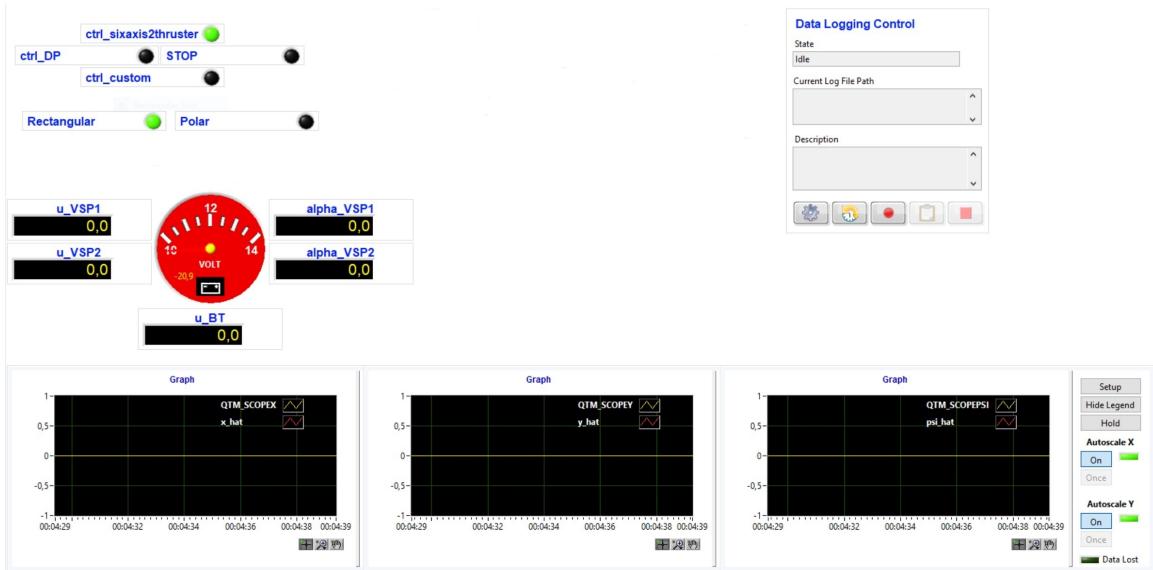


Figure 25: Graphical user interface in Case C

E Figures of results

E.1 Case A - Simulink and HIL estimation

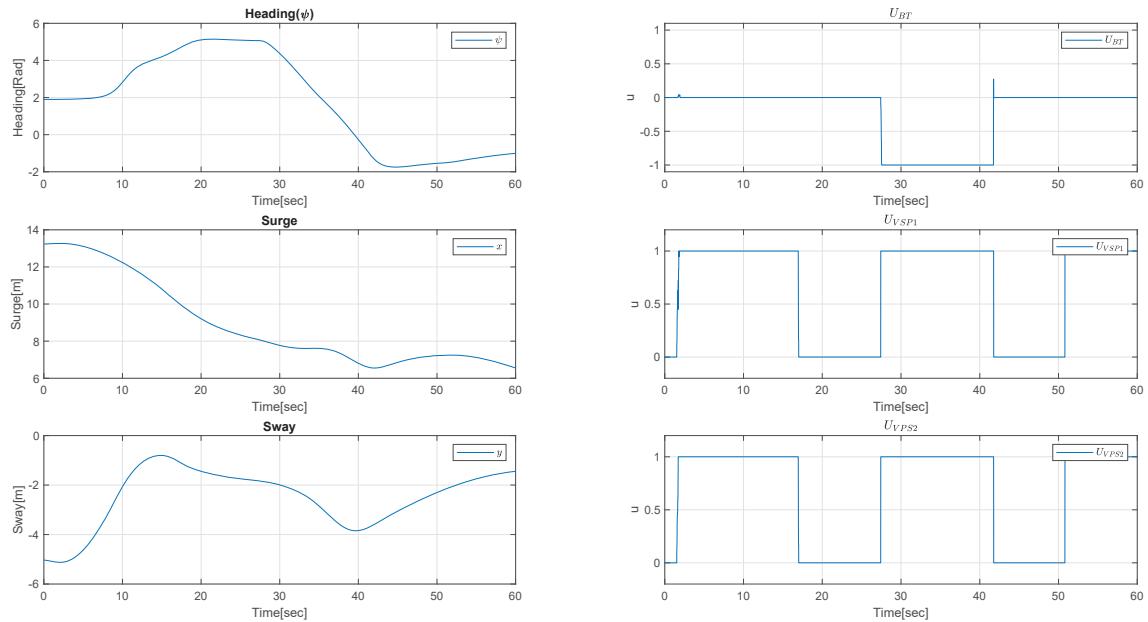


Figure 26: Vessel position in polar thrust allocation

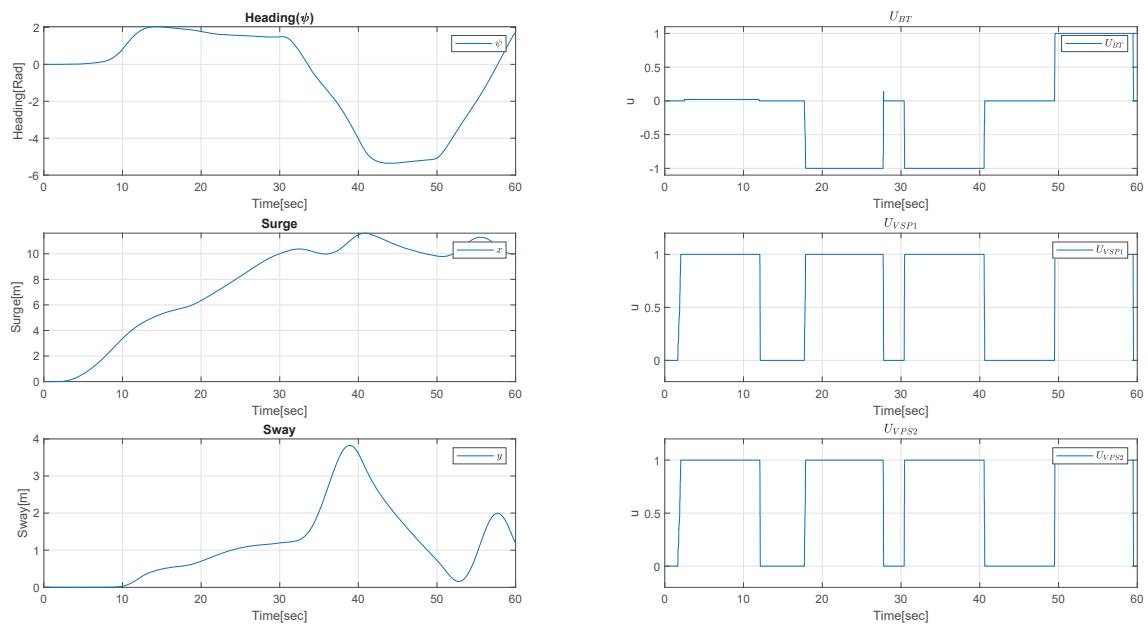


Figure 27: Vessel position in rectangular thrust allocation

E.2 Case A - Model scale experiment

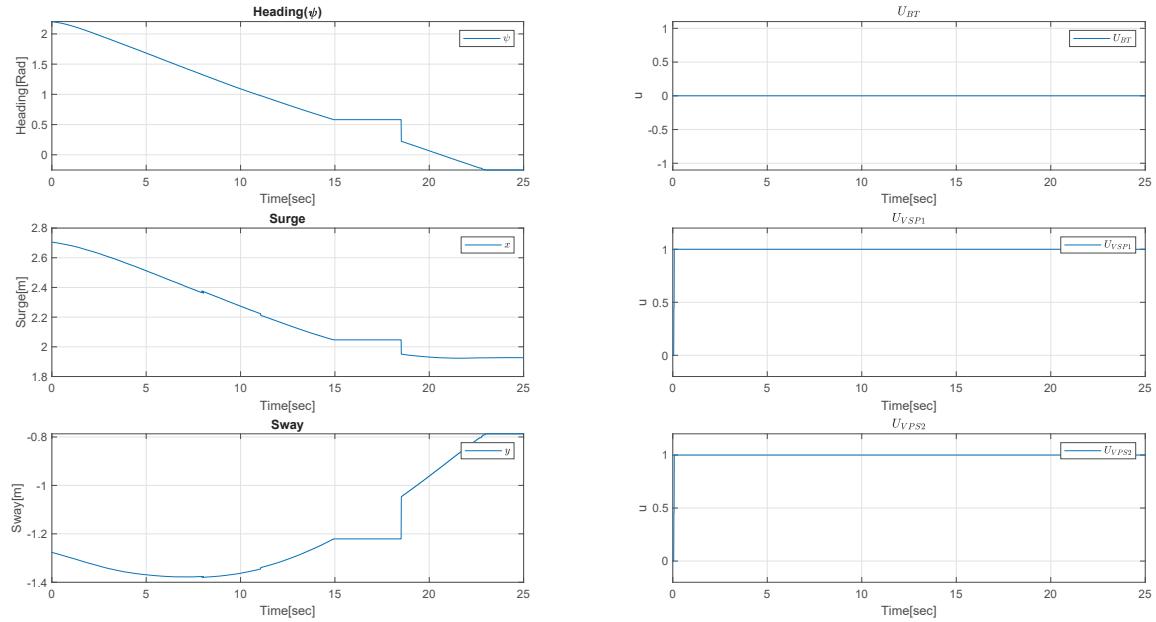


Figure 28: Vessel position with only yaw moment using polar thrust allocation

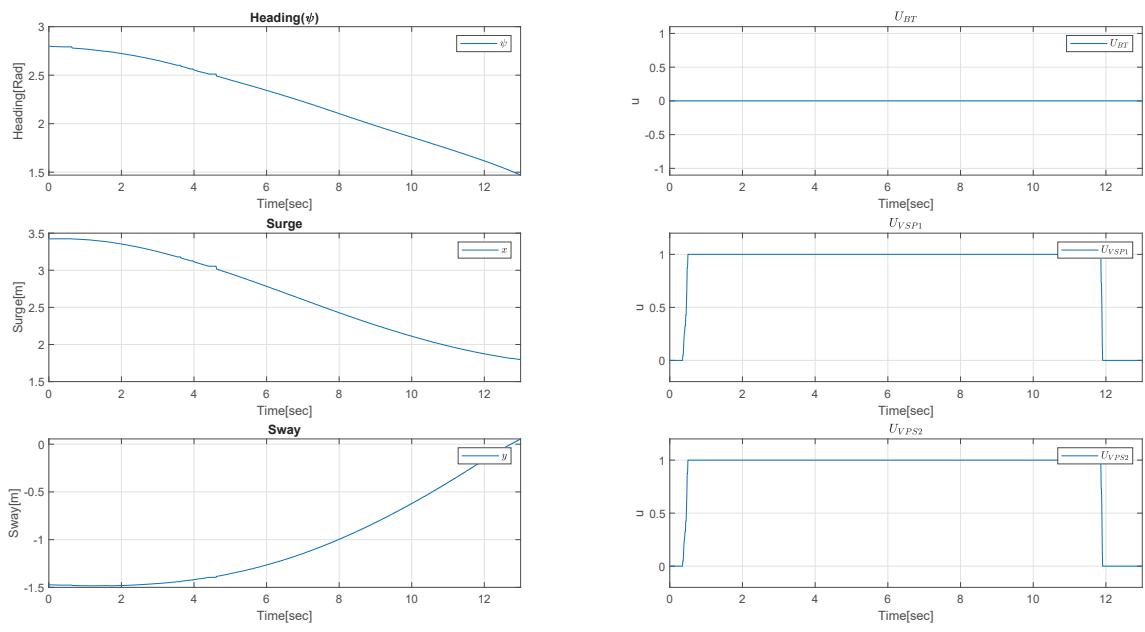


Figure 29: Vessel position with only surge force using polar thrust allocation

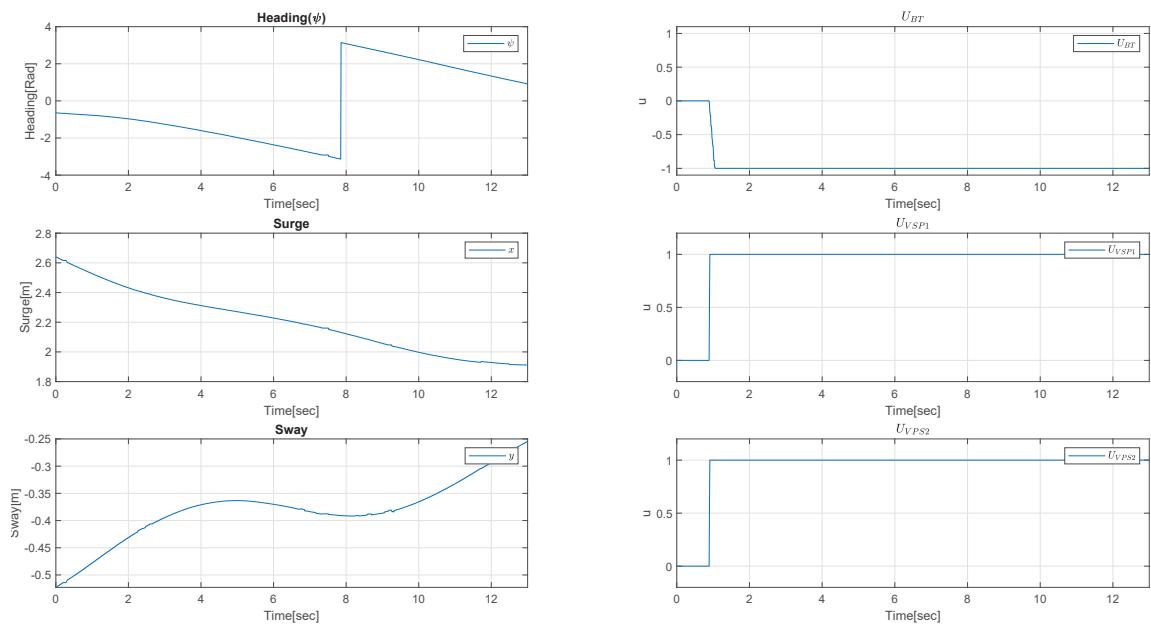


Figure 30: Vessel position with only surge sway using polar thrust allocation

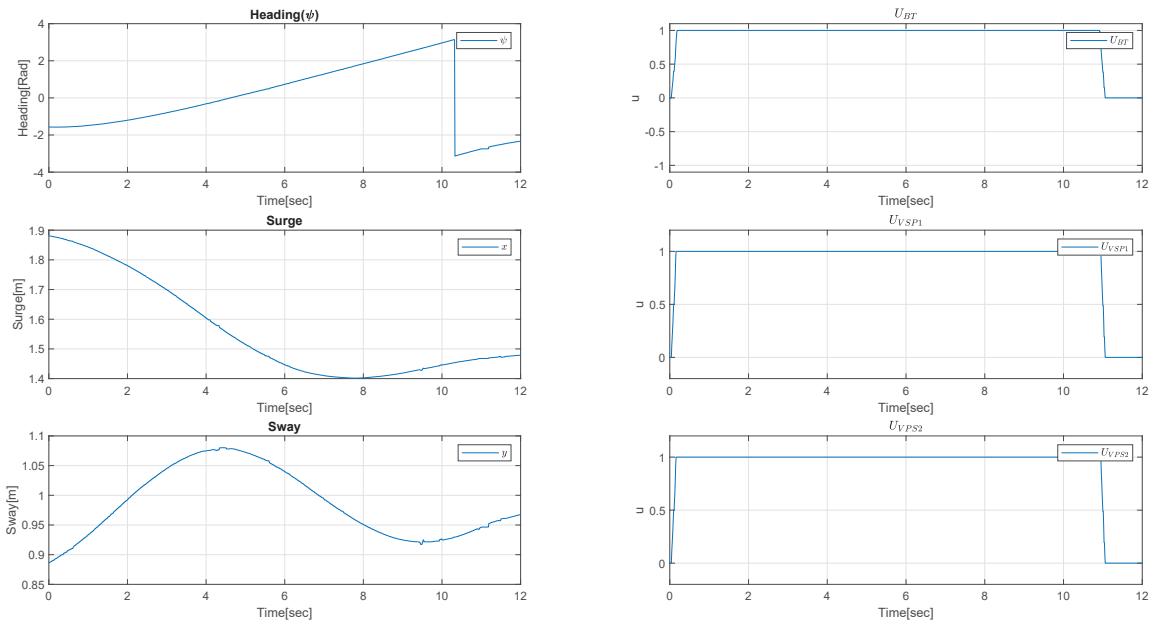


Figure 31: Vessel position with only yaw moment using rectangular thrust allocation

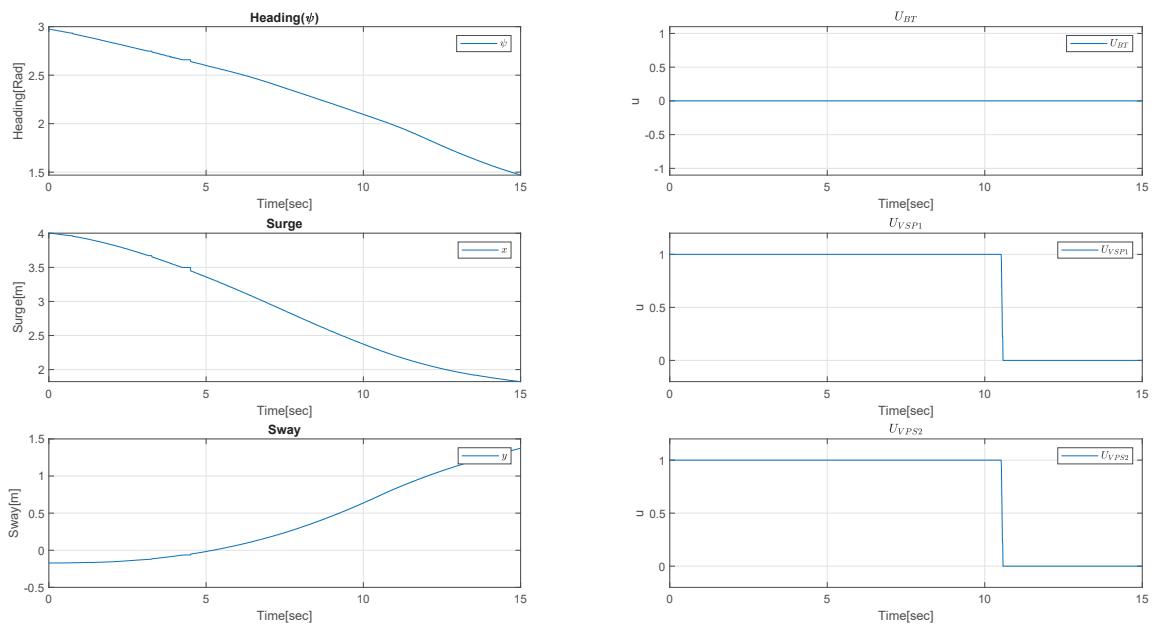


Figure 32: Vessel position with only surge force using rectangular thrust allocation

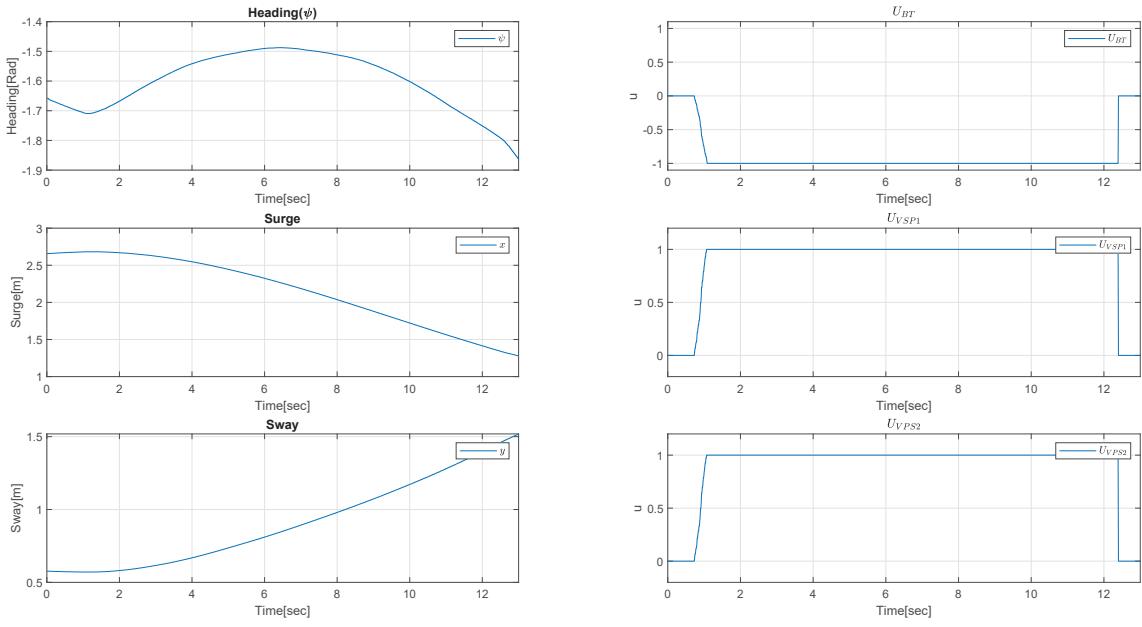


Figure 33: Vessel position with only sway force using rectangular thrust allocation

E.3 Case B - Simulink and HIL estimation

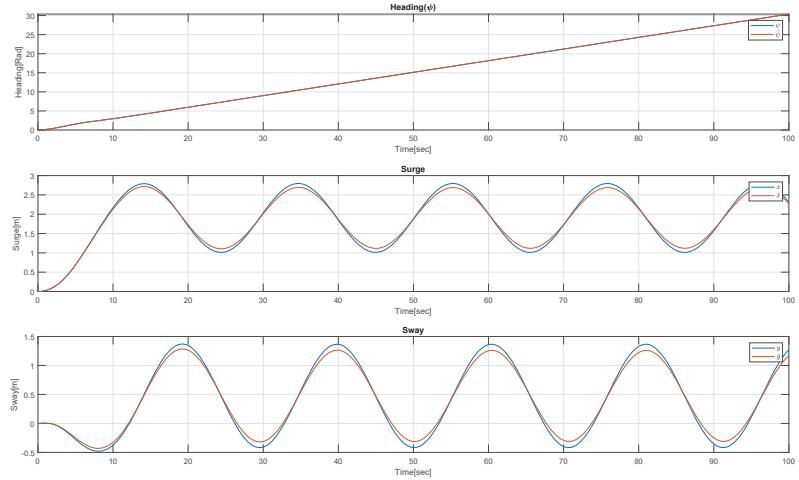


Figure 34: Position with model bias $b = 0$ for observer without bias

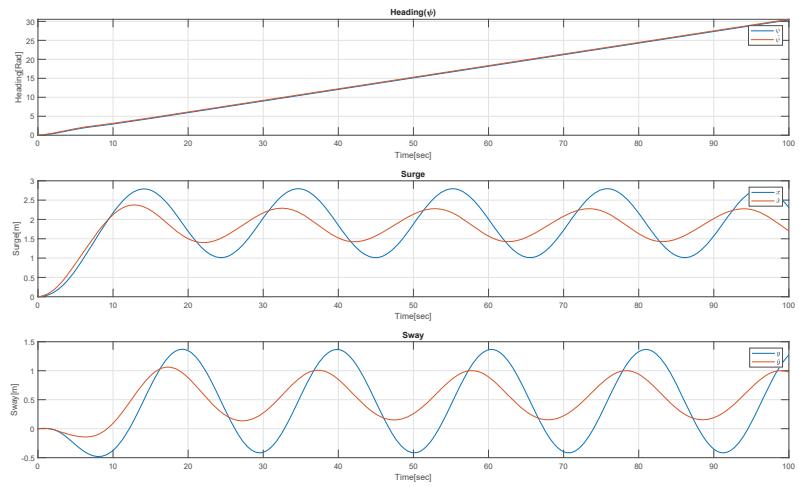


Figure 35: Position with model bias $b = 5$ for observer without bias

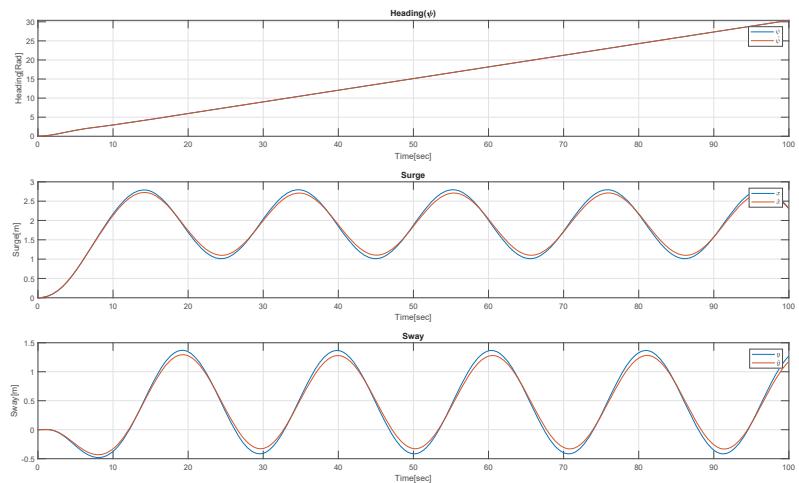


Figure 36: Position with model bias $b = 0$ for observer with bias

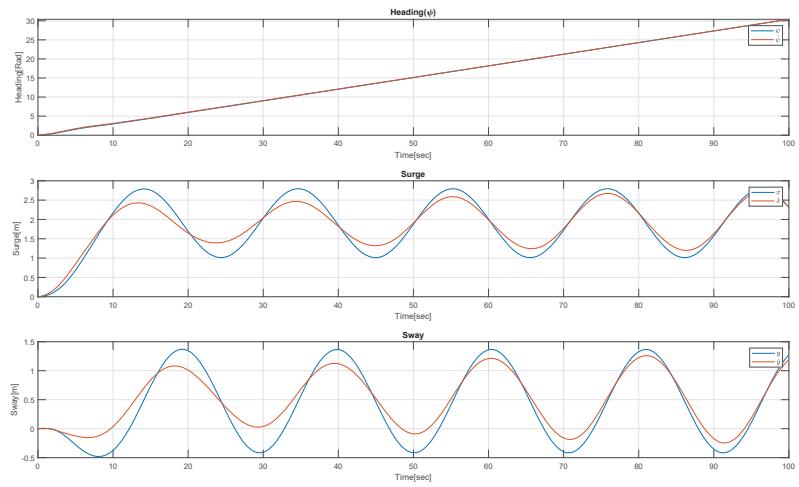


Figure 37: Position with model bias $b = 5$ for observer with bias

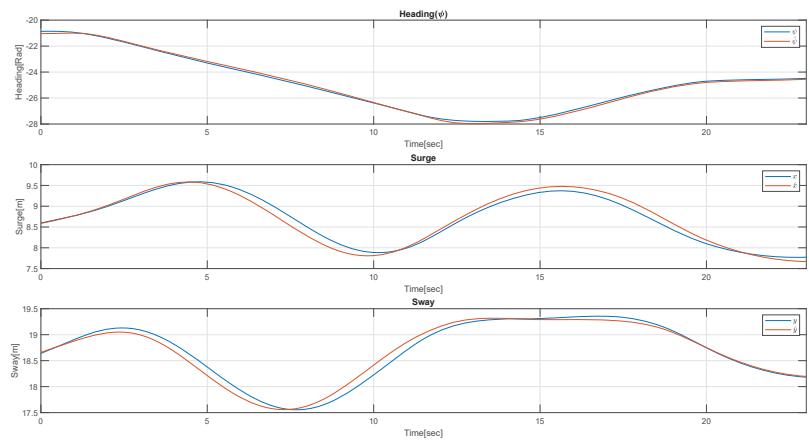


Figure 38: Plot of real and estimate position with $b = 0$

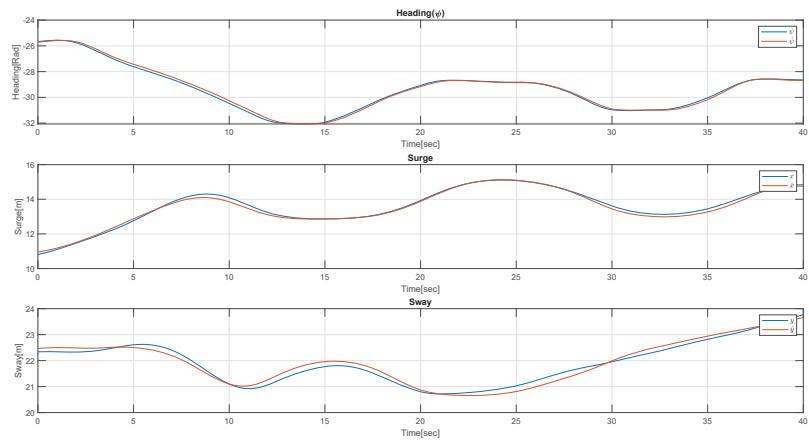


Figure 39: Plot of real and estimate position with $b = 1$

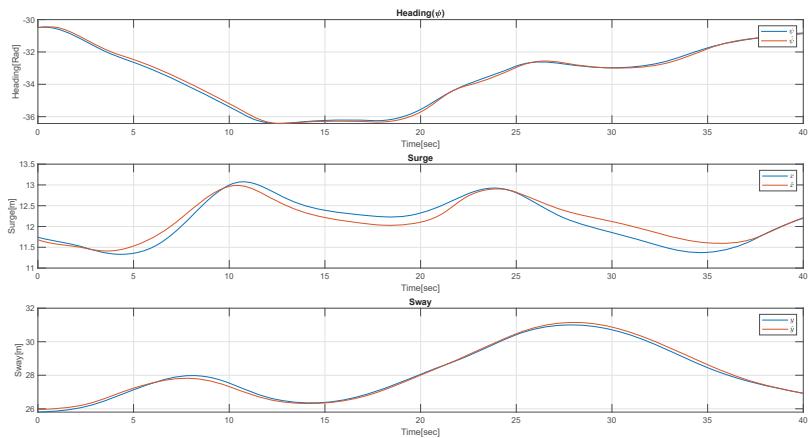


Figure 40: Plot of real and estimate position with $b = 2$

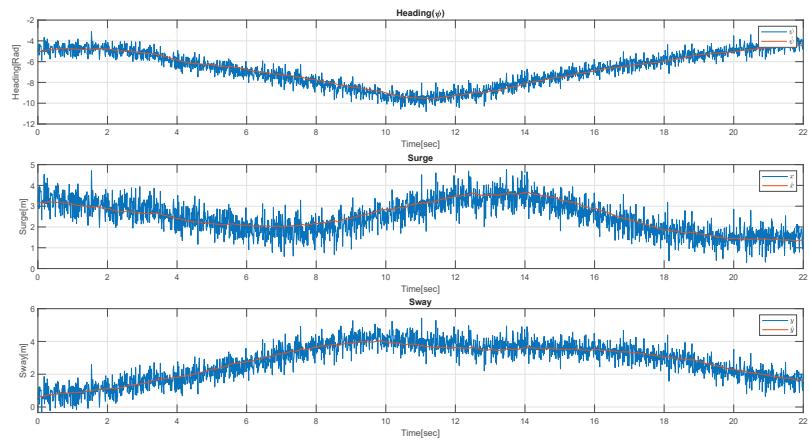


Figure 41: Plot of real and estimate position with 0.2 variance noise

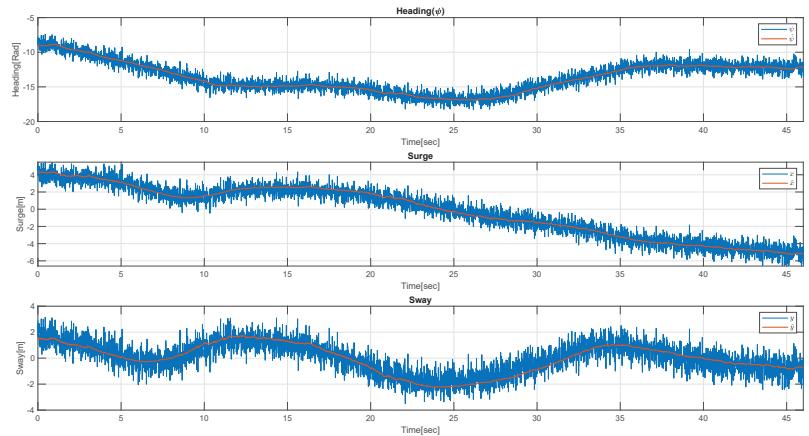


Figure 42: Plot of real and estimate position with 0.4 variance noise

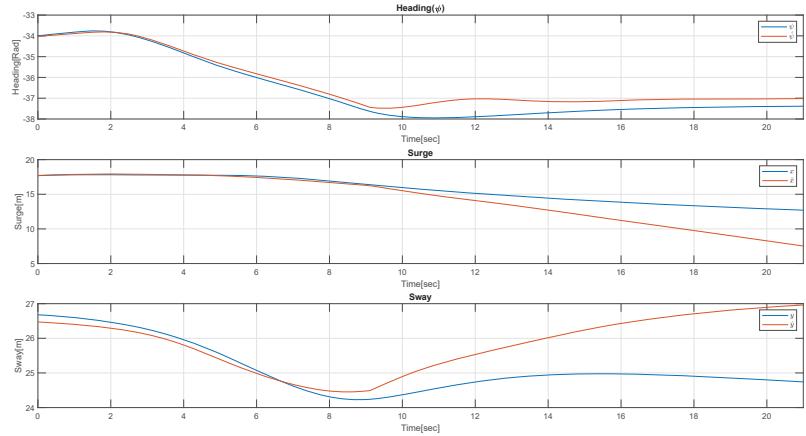


Figure 43: Plot of position in dead reckoning

E.4 Case B - Model scale experiment

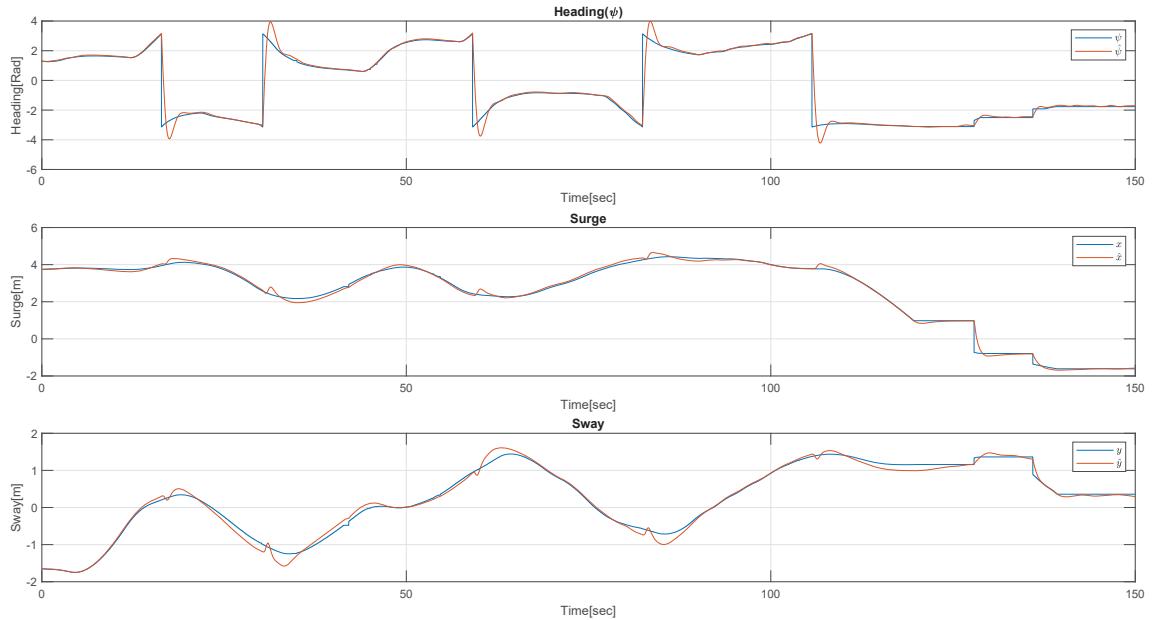


Figure 44: Plots of real and estimated position

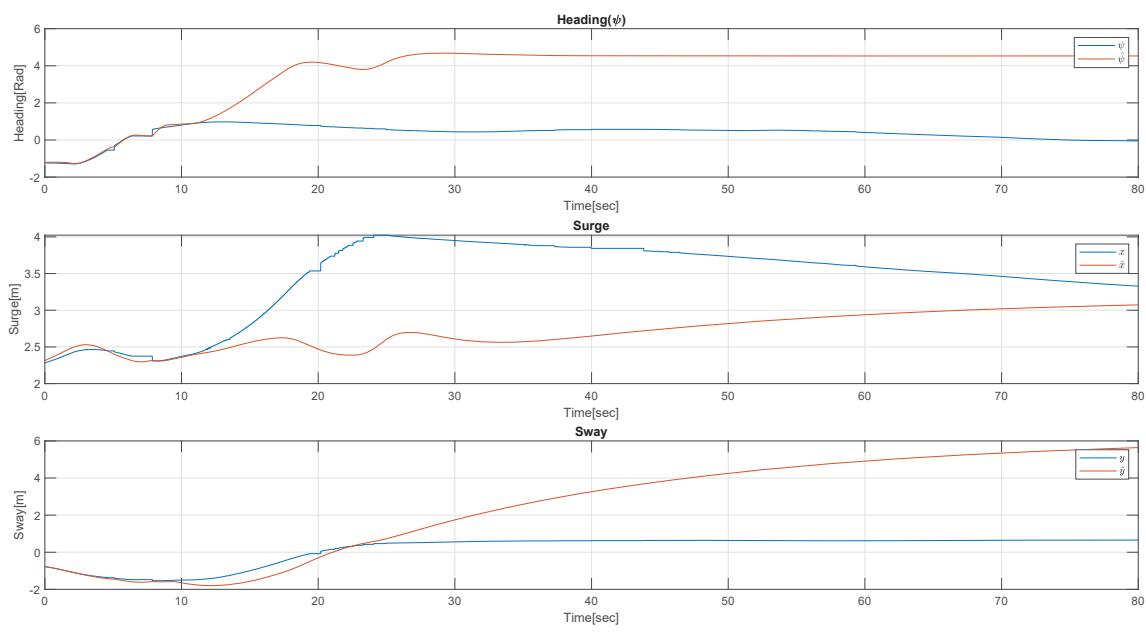


Figure 45: Plots of real and estimated position under dead reckoning

E.5 Case C - Simulink and HIL estimation

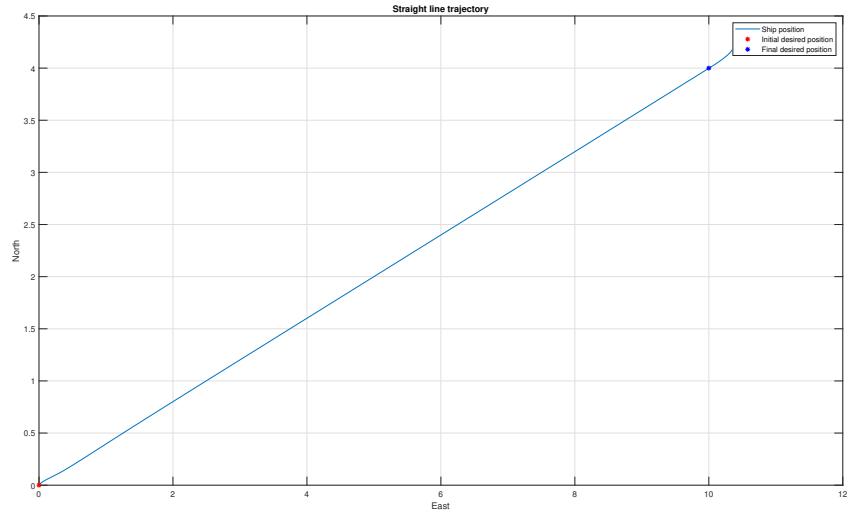


Figure 46: Straight line maneuvering with initial position at the origin

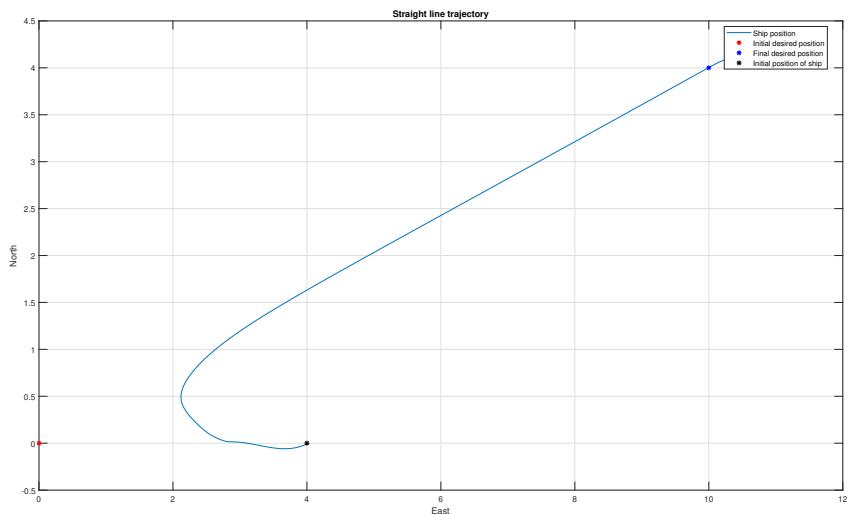


Figure 47: Straight line maneuvering with initial position at (4,0)

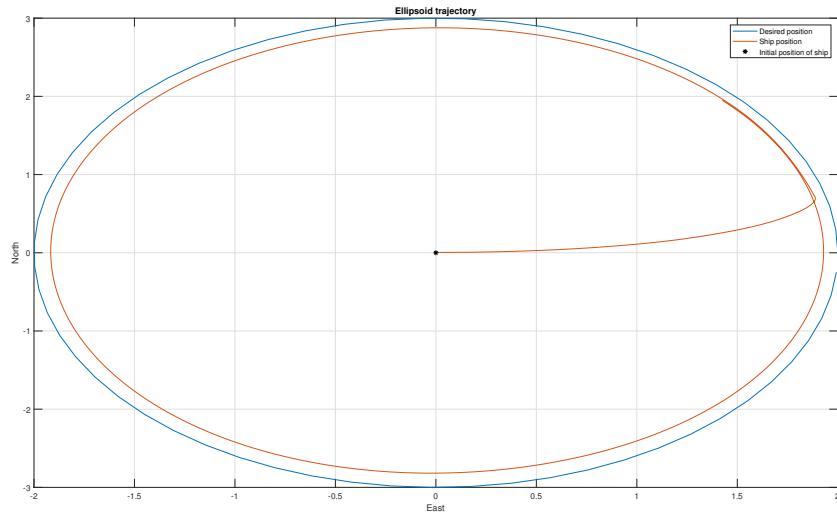


Figure 48: Ellipsoid maneuvering with initial position at the origin

E.6 Case C - Model scale experiment

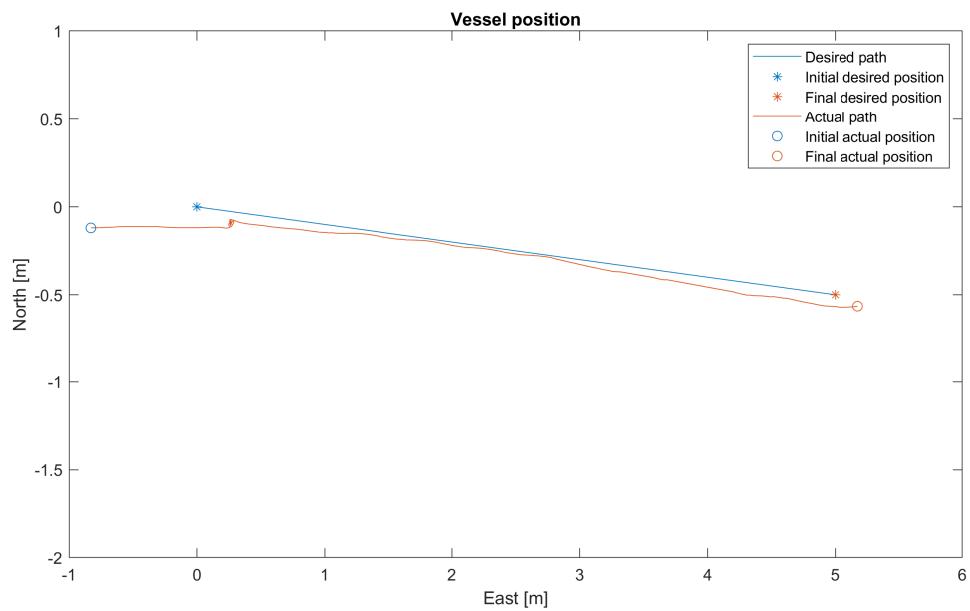


Figure 49: Straight line maneuvering

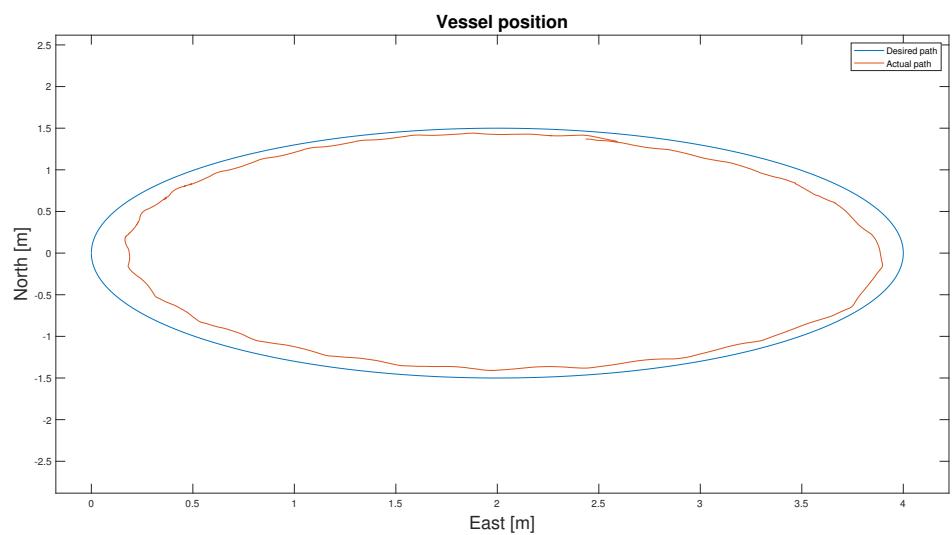


Figure 50: Ellipsoid maneuvering