

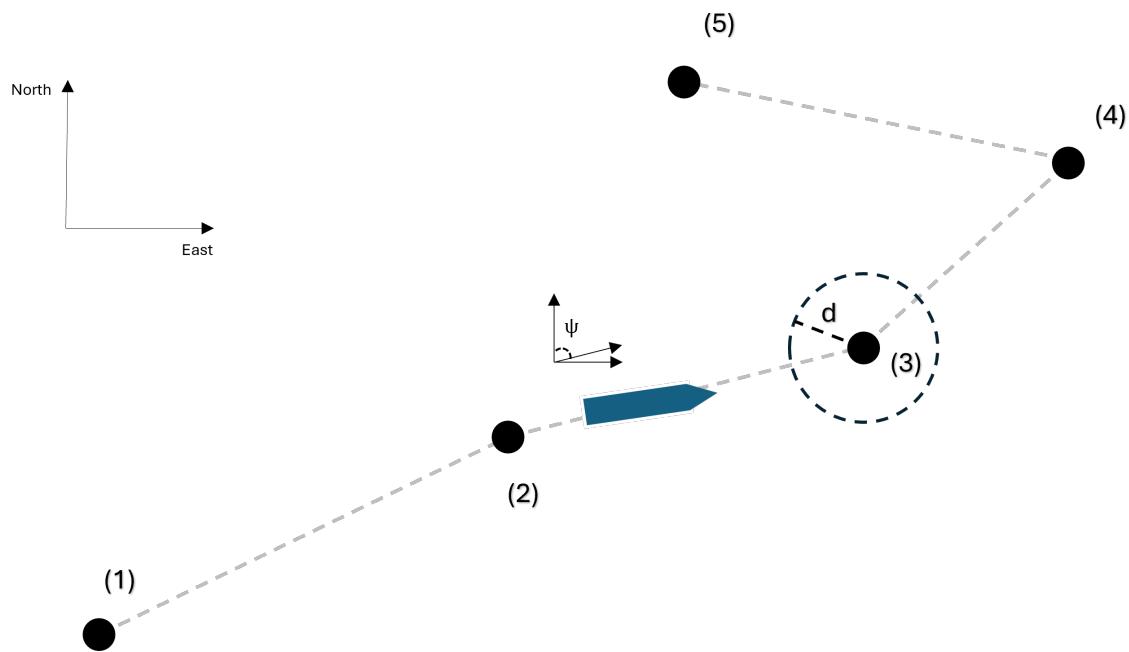


University of  
South-Eastern Norway

[www.usn.no](http://www.usn.no)

FMH606 Master's Thesis 2025  
Industrial IT and Automation

# Dynamic positioning, system identification and control of marine vessels - based on a vessel model



Tommy Sneltvedt

Faculty of Technology, Natural Sciences and Maritime Sciences  
Campus Porsgrunn



**Course:** FMH606 Master's Thesis 2025  
**Title:** *Dynamic positioning, system identification and control of marine vessels - based on a vessel model*  
**Pages:** 164  
**Keywords:** *Dynamic positioning, LQ optimal control, MPC, NMPC, path following, trajectory tracking, thruster allocation, system identification, DSR, DSR\_e, PEM, double integrators, and green DP.*  
**Student:** *Tommy Sneltvedt*  
**Supervisor:** *Associate Professor Dr. David Luigi Di Ruscio*  
**External partner:** *None*

**Summary:**

Dynamic positioning (DP) is the process of controlling position and attitude of a seafaring vessel. This is a key technology in the offshore oil and gas industry. In this thesis, setpoint tracking was performed using different control strategies; LQ optimal control (on deviation form), MPC, and NMPC. All three methods could successfully control the ship even when subjected to external disturbances from wind, waves, and current.

The thesis also touches on some more specialized topics, such as path-following and strategies for fuel reduction (green DP).

A novel approach to using system identification to identify mathematical models for vessel behavior (based on simulated data) is introduced. These identified linear models are then used in model-based controllers. It was shown that even nonlinear vessel models could be controlled using LQ optimal control based on identified models.

Three strategies for thruster allocation were implemented. It was shown that a good thruster configuration can reduce fuel consumption during DP operations.



# Preface

Keeping a seafaring vessel at a position relative to some fixed point is the fundamental problem solved by dynamic positioning systems. In modern times, the advent of computers and improved sensors have allowed for the development of increasingly complex systems for position-keeping.

This thesis aims to describe dynamic positioning from a control perspective, including state estimation (Kalman filter), setpoint tracking, and thruster allocation. Some other typical functionalities that go beyond simple position-keeping will be discussed; implementing a control strategy for reducing the thruster usage (minimizing fuel consumption and the environmental footprint) and testing two simple algorithms for path following and trajectory tracking. A part of the thesis is also concerned with the identification of mathematical models for different vessels using system identification methods. Dynamic positioning is in itself a large topic that encompasses a multitude of technologies, and this work can only hope to describe some of the features of such systems.

This work was submitted in the fall of 2025 and was a requirement for obtaining a master's degree in Industrial IT and Automation (IIA) at the University of South-Eastern Norway (USN).

I am indebted to Dr. David Luigi Di Ruscio for his insightful guidance and valuable feedback during the writing of this thesis. He kindly shared the DSR and DSR\_e toolbox with the author for use in this thesis. These are state-of-the-art subspace system identification methods. Two vessel models from the MSS toolbox by Thor Inge Fossen and the seminal Balchen model were used to simulate vessel behavior.

The author has some practical experience with DP systems as a former project engineer (2014-2016) at Kongsberg Maritime. This experience ignited an interest in understanding the underlying theory behind dynamic positioning.

It is my hope that this work might be of interest and use to others who share my interest in the topic of dynamic positioning.

Porsgrunn, 14th November 2025

Tommy Sneltvedt



# Contents

<b>Preface</b>	<b>5</b>
<b>Contents</b>	<b>9</b>
<b>1 Introduction</b>	<b>13</b>
1.1 History . . . . .	13
1.2 DP System . . . . .	15
1.3 Vessel types . . . . .	17
1.4 Sensors and thrusters . . . . .	18
1.5 DP and Norway . . . . .	19
1.6 More recent developments . . . . .	20
<b>2 Theory</b>	<b>21</b>
2.1 Kinematics and coordinate frames . . . . .	21
2.1.1 Modeling . . . . .	24
2.1.2 Determining position . . . . .	25
2.2 Wind, waves and current . . . . .	26
2.3 Supply model . . . . .	30
2.3.1 Eigenvalues of state transition matrix . . . . .	31
2.4 DP-model . . . . .	33
2.5 LQ optimal control with integral action . . . . .	34
2.5.1 LQ optimal control theory . . . . .	34
2.6 Kalman Filter . . . . .	36
2.7 Vessel models . . . . .	38
2.7.1 Supply model . . . . .	39
2.7.2 Offshore Support Vessel (OSV) model . . . . .	39
2.7.3 Balchen vessel model . . . . .	39
2.7.4 Simulating vessel models . . . . .	41
2.7.5 Discretizing continuous state space models and precalculate matrices . . . . .	41
2.8 MPC . . . . .	44
2.8.1 MPC algorithm . . . . .	44
2.8.2 MPC strategies . . . . .	51
2.9 NMPC . . . . .	52
2.9.1 NMPC algorithm . . . . .	52

2.9.2	Grouping strategy for inputs . . . . .	53
2.10	Applying system identification to identify ship dynamics . . . . .	54
2.10.1	Data generation and system identification . . . . .	54
2.10.2	Combined Deterministic and Stochastic System Identification and Realization (DSR) . . . . .	55
2.10.3	Prediction Error Method (PEM) . . . . .	56
2.10.4	Novel approach to identifying model for control . . . . .	56
2.10.5	Identifying supply model . . . . .	59
2.10.6	Identifying OSV model . . . . .	59
2.10.7	Identifying Balchen model . . . . .	61
2.10.8	Reducing model to three double integrators . . . . .	61
2.11	Tracking . . . . .	63
2.11.1	Simple path following . . . . .	64
2.11.2	Trajectory tracking . . . . .	64
2.12	Green DP . . . . .	65
2.13	Thruster allocation . . . . .	67
2.13.1	Thruster types . . . . .	67
2.13.2	Thruster configuration . . . . .	69
2.13.3	Thruster allocation: explicit solution . . . . .	70
2.13.4	Thruster allocation: quadratic programming approach . . . . .	72
2.13.5	Thruster allocation: Nonlinear optimization approach . . . . .	73
2.14	MISC . . . . .	74
2.14.1	Monte Carlo simulations . . . . .	75
<b>3</b>	<b>Results</b> . . . . .	<b>77</b>
3.1	LQ Optimal Control . . . . .	77
3.1.1	Simulating with and without disturbances . . . . .	77
3.1.2	Monte Carlo simulation . . . . .	82
3.1.3	Computational improvement when prestoring matrices . . . . .	84
3.2	MPC . . . . .	85
3.2.1	Comparison of MPC formulations . . . . .	86
3.2.2	Monte Carlo simulation . . . . .	91
3.3	NMPC . . . . .	94
3.3.1	Monte Carlo simulation . . . . .	97
3.4	System Identification . . . . .	99
3.4.1	Identifying vessel models . . . . .	99
3.4.2	Supply model simulation . . . . .	100
3.4.3	OSV model simulation . . . . .	102
3.4.4	Balchen model simulation . . . . .	105
3.4.5	Comparison of SI models . . . . .	107
3.4.6	Using double integrator model . . . . .	109

3.5	Tracking . . . . .	113
3.5.1	Simple path following . . . . .	114
3.5.2	More advanced tracking strategy . . . . .	117
3.5.3	Summary . . . . .	118
3.6	Green DP . . . . .	119
3.6.1	Comparison of green DP versus ordinary DP control when using reduction factor . . . . .	119
3.6.2	Comparison of green DP versus ordinary DP control when using no reduction factor . . . . .	122
3.7	Thruster allocation . . . . .	124
3.7.1	Thruster allocation applied to single scenario . . . . .	124
3.7.2	Summary . . . . .	136
3.7.3	Monte Carlo simulation . . . . .	136
<b>4</b>	<b>Discussion</b>	<b>139</b>
4.1	Control algorithms . . . . .	139
4.2	System identification . . . . .	141
4.3	Path following and trajectory tracking . . . . .	142
4.4	Green DP . . . . .	142
4.5	Thruster allocation . . . . .	143
<b>5</b>	<b>Conclusion</b>	<b>145</b>
5.0.1	Suggestions for future work . . . . .	146
<b>Bibliography</b>		<b>149</b>
<b>A</b>	<b>Programming code</b>	<b>153</b>
<b>B</b>	<b>Video demonstrations</b>	<b>157</b>
<b>C</b>	<b>Parameter values</b>	<b>159</b>
C.1	Wind model . . . . .	159
C.2	Balchen model . . . . .	159
C.3	Kalman filter . . . . .	160
C.4	LQ optimal control (MPC) . . . . .	160
C.5	Model predictive control . . . . .	161
C.6	Nonlinear Model predictive control (NMPC) . . . . .	161
C.7	System identification . . . . .	161
C.8	Tracking . . . . .	161
C.9	Green DP . . . . .	162
C.10	Thruster allocation . . . . .	162
<b>D</b>	<b>Signed agreement with USN supervisor concerning content of thesis</b>	<b>163</b>



# Nomenclature

## Symbol Explanation

DGPS	Differential Global Positioning System
DIPTD	Double Integrating Plus Time Delay (system)
DoF	Degrees of Freedom
DP	Dynamic Positioning
DSR	Combined <u>D</u> eterministic and <u>S</u> tochastic System Identification and <u>R</u> ealization
DSR_e	Extended version of DSR (developed for handling closed-loop data)
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IAE	Integrated Absolute Error
LQ	Linear Quadratic (optimal control)
LQG	Linear Quadratic Gaussian (controller)
MIMO	Multiple-Input Multiple-Output (system)
MPC	Model Predictive Control
N4SID	<u>N</u> umerical algorithms for <u>S</u> ubspace <u>S</u> tate <u>S</u> pace <u>I</u> Dentification
NMPC	Non-linear Model Predictive Control
OSV	Offshore Supply Vessel
PEM	Prediction Error Method
PID	Proportional–Integral–Derivative (controller)
PRBS	Pseudo Random Binary Signal
PRNG	Pseudo-Random Number Generator
RMSE	Root Mean Squared Error
RPM	Rotations Per Minute
RPS	Rotations Per Second
SISO	Single-Input Single-Output (system)



# 1 Introduction

It is often said that civilizations are shaped and formed by their surrounding geography. Since 71 % percent of Earth's surface is covered by water, it is hard to overestimate its significance to humanity. Today, it is an important source of food, a major highway for mercantile traffic, used for business and leisure travel, offshore oil and gas activities, and offshore wind turbines. The use of cargo ships to deliver goods across the world was and continues to be a major driver in globalization.

In ancient times, it was believed that storms on the open sea were a manifestation of divine wrath, though as time progressed, mankind learned to master the seas and weather the storms through emerging technology. Dynamic positioning is one of those technologies that played an important role in helping sailors manage the fickle nature of the seas.

This thesis will investigate control strategies for dynamic positioning of seafaring vessels. Such vessels are subjected to external forces and moments due to wind, waves, and current. To maintain some absolute or relative position<sup>1</sup>, it is necessary to balance the external disturbances with propellers, rudders, tunnel thrusters, and azimuths. The typical approach to control in dynamic positioning is a two-fold process; estimate the net forces and moment needed to maintain position, followed by calculating the individual forces needed from each thruster to realize these net forces and moment.

## 1.1 History

Some of the first forms of dynamic positioning were the use of anchors or general seamanship with respect to some visual reference. Modern dynamic positioning systems typically use GPS systems to determine position. One of the key technologies needed for dynamic positioning is some concept of position on the sea surface. It was the Greco-roman polymath Claudius Ptolemy, living in the second century AD, who divided the world into latitudes (parallels) and longitudes (meridians). One of the most significant historical challenges of the past was determining the position of a ship in the ocean. From ancient times it was known that the latitude could be determined from celestial observations.

---

<sup>1</sup>Absolute position would relate to latitude and longitude. The relative position would be with respect to some (potentially) moving object at sea

However, determining longitude (defined with respect to the Greenwich meridian) remained a major problem that challenged the brightest minds. This problem was solved by a carpenter turned clockmaker, John Harrison, who devised an ingenious clock, known as a chronometer, which was precise even when subjected to the conditions of the sea (temperature changes and ship movement). Knowing the local time and longitude at the port of departure, it was possible to estimate the current longitude by observing the position of the sun (and hence the local time) and comparing it with the chronometer watch. Before this, the common method was to perform dead reckoning, essentially measuring speed and heading at constant intervals, and integrating these observations to obtain an estimate of position. This method was obviously flawed and resulted in many accidents. Even today, with modern computers and sensors, it is known that dead-reckoning will degrade over time. However, dead-reckoning is fairly accurate for short time-spans and is hence used as backup in DP systems in the case of position sensor failure. An interesting treatment on the topic of navigation in an historical context can be found in the book "Longitude" [1].

One of the first automatic control systems [2] for seafaring vessels was a device known as the *Metal Mike*, a mechanical autopilot system guided by a gyroscope. This device used the measured heading angle from the gyroscope to correct the rudder angle in order to maintain the vessel heading setpoint. This device, developed by Elmer Sperry, was a simple single-input single-output (SISO) system, and the first trials were performed in 1922. Today, this kind of autopilot system, with rudder control alone, would typically be solved using a PID controller. An interesting recent and novel approach in this regard is the paper by Ruscio et al. [3] that identifies a SISO system from a more complicated model; the rudder angle is controlled to keep the heading setpoint. The transfer function is originally a seventh order model, but is reduced to a DIPTD process<sup>2</sup>. The model parameter values of the reduced model are then used to determine the PID tuning parameters. The general dynamic positioning problem is not concerned with the vessel heading alone, but also includes the vessel position. Information about position and attitude is needed. Furthermore, the controller calculates the net forces and moment needed to maintain position, making it a multiple-input multiple-output (MIMO) system. Not surprisingly, this is a much more complicated problem and automated systems did not become a reality before the second half of the 20th century.

The first automated dynamic positioning system was the brainchild of Howard Shatto [2][4], a Shell engineer tasked with developing automated positioning control for the ship named *Eureka*. He used a PID controller on each degree of freedom to maintain the position reference. He is known as the father of dynamic positioning. This vessel was

---

<sup>2</sup>DIPTD: Double Integrating Plus Time Delay systems are processes with transfer functions on the following form:

$$Y(s) = T(s)U(s) = K \frac{e^{-\tau s}}{s^2} U(s) \quad (1.1)$$

Here  $K$  is the acceleration gain and  $\tau$  represents the time delay.

constructed in 1961 and relied on a taut wire system for position measurements (GPS was not yet invented). Earlier attempts were based on manual thruster control, using some position reference system to track how well the vessel managed to stay close to the setpoint.

The first DP installations were mainly used for research vessels. Over time, as offshore oil and gas extraction became increasingly important, both in Norway and abroad, the need for automated position control became increasingly obvious. A collaboration between Jens Glad Balchen, Kongsberg Våpenfabrikk (a large Norwegian technology company) and SINTEF<sup>3</sup> resulted in a commercial DP system that used a Kalman filter for improved performance. Over time, this product became the "state-of-the-art" dynamic positioning system, dominating the world markets.

The development of dynamic positioning systems is tightly tied to other technologies developed at the same time. The move from shallow water to deep water required improved position reference systems. The GPS, Galileo and Glonass<sup>4</sup> systems were major breakthroughs. Today, global navigation satellite systems are standard equipment in every DP system.

Today, DP is ubiquitous in the offshore industry. The usage of DP is now strictly regulated by governments and classification societies [5]. What used to be highly custom-made products is now more and more resembling shelf-ware. However, operational needs are no longer limited to only simple station-keeping, new DP systems must often support complex operations like underwater cable laying or anchor handling. A list of operations performed using dynamic positioning is found in [4]. Among those mentioned are offshore drilling, anchor handling, pipe and cable laying, and diving support. Some typical vessels that use dynamic positioning are supply vessels, research vessels, yachts, cruise ships, mobile loading platforms, icebreakers, offshore rocket launchers, and mine hunters. This is by no means a complete list, but it does show the variety of tasks that require DP.

## 1.2 DP System

The IMO<sup>5</sup> states the following definition of a DP system [6]:

"The complete installation necessary for dynamically positioning a vessel comprising the following sub-systems: Power system, thruster system, and DP control system."

---

<sup>3</sup>SINTEF is a large research foundation headquartered in Norway.

<sup>4</sup>GPS: Global Positioning System (American). Galileo and GLONASS are the European and Russian equivalent of GPS, respectively.

<sup>5</sup>IMO: International Maritime Organization is a specialized agency under the United Nations (UN) umbrella founded in 1948. It concerns itself with the safety and security of seafaring vessels. It is also involved in improving the environmental footprint of shipping.

This definition is visually depicted in Figure 1.1. From this definition, it is clear that

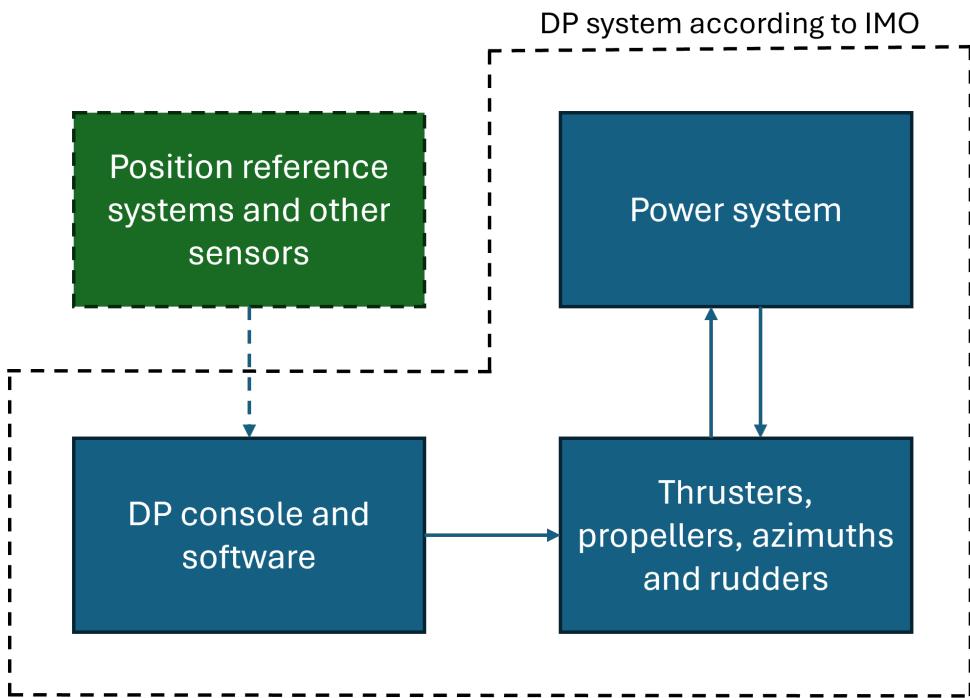


Figure 1.1: The figure outlines the relation between sub-systems belonging to a dynamic positioning system according to IMO. Position reference systems and other sensors are of vital importance, but is not explicitly mentioned in the short definition.

the DP system entails more than just the control software and user interface, and this fact is also reflected in the different DP classifications [5]. The different classification types depend on points of failure. Essentially, three DP classes are defined by the major classification societies. All classes assume automatic position and heading control, but the level depends on how the system reacts to internal failure.

- **DP Class 1:** A single point of failure can cause loss of position and heading.
- **DP Class 2:** No single point of failure in active components can cause loss of position and heading. Active components can be generators and thrusters, as opposed to static components such as pipes and cables.
- **DP Class 3:** The DP system should maintain its position and heading in the event of a single point failure. In addition, it should be able to resist flooding or fire that occurs in any single compartment on the ship.

PID controllers were used in the early days of DP systems, which were later replaced by more complicated algorithms. The paper by Balchen [7] outlines the use of a linear-quadratic controller combined with the Kalman filter for state estimation. This would be the approach pursued by Kongsberg Maritime when realizing their first DP system.

It is also known [5][4] that the green DP mode uses non-linear model predictive control (NMPC). Thus, there is a wide array of strategies that can be employed, from simple output feedback controllers to more complex model-based controllers.

The controller will perform two important tasks; it will calculate forces and moment needed for reducing the setpoint error, but will also need to calculate how these forces are to be realized in the available thrusters. The latter task is called thruster allocation. This twofold process is depicted in Figure 1.2.

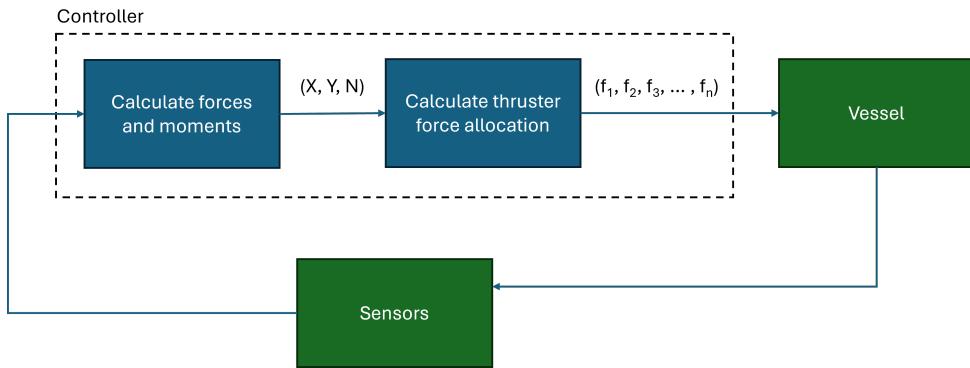


Figure 1.2: The figure depicts the two-fold control process. First it computes the forces and moment needed for reaching the position and heading setpoint. Here  $X$  and  $Y$  are the force demand in surge and sway respectively, while  $N$  represent the torque demand. These must be realized by assigning individual forces to each thruster:  $f_1, f_2, f_3, \dots, f_n$  where  $n$  represents the number of thrusters. If rudders or azimuths were present, their angles would also be calculated.

### 1.3 Vessel types

This thesis will consider ship models of displacement vessels. These vessels can be defined with respect to the Froude number [8], [2]:

$$F_n = \frac{u}{\sqrt{gL}} \quad (1.2)$$

where  $u$  is the speed of the vessel,  $g$  is the gravitational acceleration and  $L$  represents the submerged length of the craft. Displacement vessels have a Froude number below 0.4. These are the most common type of vessel, such as supply and cargo ships, characterized by buoyant forces dominating the hydrodynamic forces. Ships with Froude number above 0.5 are considered high-speed vessels. A high-speed catamaran is an example of such a vessel.

The vessel model types considered here will all be of the supply ship variety, landing them squarely in the displacement ship category.

## 1.4 Sensors and thrusters

Typical thrusters associated with seafaring vessels are: propellers, rudders, tunnel thrusters, and azimuths. The number, positioning, and size of the thrusters will determine the ability of the ship to withstand environmental conditions (imposed by wind, waves, and current) in order to maintain some fixed setpoint. A common tool for estimating the capability of a dynamic positioning system is the DP capability report/plot. A capability plot graphically shows how much external disturbance the ship can handle for different angles of attack relative to the vessel and still maintain position. They also consider different failure states, such as the loss of a single thruster. These plots are considered when class societies determine the DP class of a particular ship.

There are many types of sensors that can be used to improve the state estimate of the model. Here, it will be assumed that position sensor(s) and gyrocompass are available, that is, the position and heading measurements are fed to the Kalman filter. Wind sensors are also assumed to be available to measure wind speed. A Doppler log can be used to measure the velocity of the vessel with respect to water or the seabed. There are a multitude of position reference systems on the market. Some examples [5] are listed below:

- **Hydroacoustic position reference (HPR) systems:** Estimates position based on reference points on the seabed.
- **Global Navigation Satellite System:** Collection of satellites that broadcast information that can be used to determine position. A more refined version than the standard GPS is the Differential GPS (DGPS)<sup>6</sup>.
- **Taut-wire position reference system:** A wire is connected from the ship to the seabed (typically using an anchor). By measuring the angle and length of the wire used, it is possible to estimate the position of the ship relative to the seabed.
- **Microwave position reference system (Artemis):** Uses microwaves to estimate position based on communication with Artemis stations.

As the operational areas moved from shallow water to deep water (up to thousands of meters of vertical distance to the seabed), these new conditions put higher demands on the position reference systems. In the 1990s, GPS became a viable contender to other position reference systems, especially when using the DGPS service. Today, GPS is the main position reference system on most ships, offering reliable measurement data with reasonable accuracy.

---

<sup>6</sup>The DGPS calculates the position based on correction signals sent from stations monitoring the satellites. These signals are transmitted by radio signals. In Norway there are 12 such DGPS stations situated along the Norwegian coastline [9]. These services will be discontinued in 2026 as modern GPS are considered to perform well enough without the correction signal from the monitoring stations [10].

It should be stressed that many dynamic positioning systems use multiple sensors for the same type of measurement. This is to avoid a single point of failure that leads to loss of position. Sensors are considered active components and are of special interest when considering the DP class of a vessel.

## 1.5 DP and Norway

This historical section is based on [5][4]. Balchen, often considered the father of Norwegian cybernetics, was a seminal figure in the development of new DP technology. He brought with him new knowledge from his studies in America and introduced the concepts of linear quadratic optimal control and Kalman filter. His main idea was to combine these two technologies in automated position control. This was a novel approach at the time; other competitors used simple PID controllers for each degree of freedom. The new approach offered superior performance compared to other systems on the market.

Initially started as a small group named Albatross within the KV umbrella, they were tasked with developing a DP system for the ship *Seaway Eagle*. The system was approved onboard the vessel in 1977. A testament to the success experienced in this early period is the fact that 42 out of a total of 80 ships furnished with dynamic positioning technology in the year 1980 used the Albatross system. An impressive feat considering they faced stiff competition from the likes of Honeywell. Not only did Albatross over time dominate the domestic and international markets, but they would also become the leading innovator of new DP technology. They greatly influenced the way class societies defined and evaluated DP systems.

The development of dynamic positioning has been dictated to a great extent by the needs of the offshore industry. Thus, the move from shallow water to deep water created a need for improved position reference systems. Furthermore, the need for new types of operation, such as cable laying and positioning mooring, meant the expectations of DP expanded beyond mere position keeping. It is hard to imagine the developments within dynamic positioning in the last 50 years without the increasing importance of the offshore oil and gas industry.

The importance of DP in a Norwegian context is hard to overestimate, the readers of the technical magazine *Teknisk Ukeblad* [11], elected dynamic positioning as the single most impressive engineering feat in Norway since the end of the second world war.

## 1.6 More recent developments

While the original dynamic positioning problem has been solved, it is still an area of active development as there are many novel ways dynamic positioning can be used to facilitate new tasks. An example of this is rocket launch platforms on the sea surface. However, the details and internal workings of such a system are often proprietary knowledge.

Some recent work has been done with respect to the low-frequency Balchen model. In [12] the Balchen model was linearized and controlled using LQ optimal control and MPC. The Balchen model was identified as a linear state-space model (using the sub-space method known as DSR) and then used in LQ optimal control. This is a particularly attractive approach, as the control method can be based on the identified model, only requiring recorded data over time. The only assumption here is that the dynamics of the vessel can be reasonably well explained by a linear model. In [13] the Balchen low-frequency model was studied in more detail; it was discovered that the Balchen model could be simplified to a linear three double integrator model. In the thesis [14] experiments with PID controllers showed, that while being a computationally efficient method, the performance of PID controllers were highly dependent on the weather conditions. Thus, one set of parameters might work well under calm weather conditions, but struggle under more harsh weather conditions, and vice versa.

Another interesting approach to system identification was written about by Fossen et al. [15]. Here, a supply model is identified by subjecting the vessel to predefined maneuvers to secure persistently excited data. An extended Kalman filter is used to estimate the model parameters.

The supply model from [15] will be the main model used for control purposes in this thesis. Two other models, OSV and Balchen, will be used in the context of system identification and LQ optimal control. PID controllers will not be investigated further since it seems from previous work that the tuning parameters are highly weather-dependent, suggesting that the model-based approach is the most promising avenue to pursue.

There are some interesting papers on the topic of thruster allocation. The mathematical complexity and computational load of the thruster allocation algorithms depend on the thruster configuration. Some thrusters have two degrees of freedom, not only can they adjust the force output but also decide on the direction of said force. Typical examples of this are azimuth thrusters (which can rotate along a vertical axis). But also rudders in combination with main propellers can direct forces along different angles. Thruster allocation algorithms for nonrotatable thrusters are described in [16][2]. An algorithm for rotatable thrusters is described in [17]. An interesting approach was described in [18] where neural networks were used for thruster allocation; however, using this black-box approach, there is no guarantee that the constraints will hold.

# 2 Theory

This chapter discusses some of the theory behind dynamic positioning, different control strategies, system identification, path-following, trajectory tracking, and thruster allocation. An environmentally friendly DP control strategy named green DP will also be explained. Three different vessel models will be discussed. Two of the models, supply and OSV, stem from the MSS library by Fossen, the last is the vessel model described by Balchen in his paper "A Dynamic Positioning System Based on Kalman Filtering and Optimal Control" [7].

## 2.1 Kinematics and coordinate frames

Any rigid-object in the real world has a position and attitude. This amounts to a total of six degrees of freedom; surge, sway, heave, roll, pitch, and yaw. These are defined with respect to the BODY coordinate frame and are depicted in Figure 2.1.

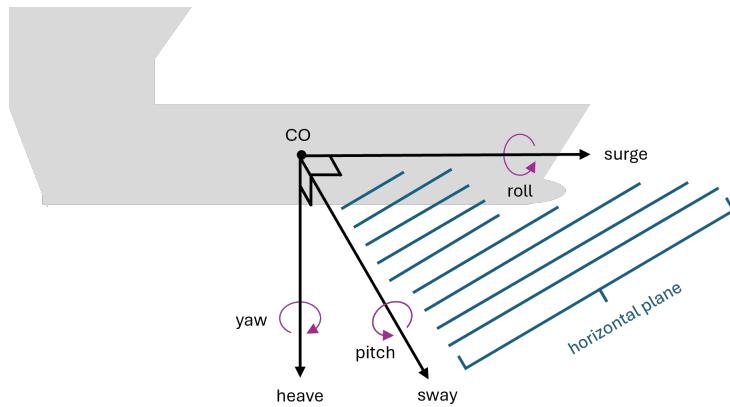


Figure 2.1: Depicts the BODY coordinate frame. The surge, sway and heave direction is fixed with respect to the ship. Surge represents longitudinal motion, sway represents transverse (sideways) motion, and heave represents vertical motion. Roll is the rotational angle around the surge axis, pitch is the rotational angle around the sway axis, and yaw is the angle around the heave axis. The yaw is especially important since it represents the heading of the marine craft. CO refers to Coordinate Origin, the zero position  $(0,0,0)$  of the vessel.

Depending on the type of vessel in question and the intended use of the model; models can be simplified using fewer degrees of freedom. Some examples are given below [2]:

- **High fidelity ship model (6-DoF):** If a high-accuracy model is required, it might be necessary to use all degrees of freedom.
- **Simplified ship model (3-DoF):** This type of model typically uses the sway and surge position together with the yaw angle. It assumes that the ship moves in a horizontal plane spanned by the surge and sway axes. The yaw angle represents the ship's heading in this plane. This model is typically used in dynamic positioning.
- **Simplified ship model with roll stabilization (4-DoF):** This model includes the roll angle. Roll movement can induce sea sickness in crew, damage cargo, or, in the worst case: contribute to capsizing the vessel. Ship stabilizers can be installed on a ship (a form of fins) to reduce roll movement.
- **Ship autopilot (1-DoF):** A one-dimensional model that aims to control the vessel heading (yaw angle).
- **Other 1-DoF models:** Speed controllers (surge) and roll damping.

For the purpose of dynamic position we will here consider three degrees of freedom, which includes the surge and sway axes together with the yaw angle.

To better understand how movement in the BODY frame is related to some geographic frame, it is necessary to define a second frame named NED<sup>1</sup>. The NED frame can be considered as a fixed frame relative to some point on the surface of the Earth (some fixed latitude and longitude). This gives a good approximation of the ship position in the context of dynamic positioning since the curvature of the Earth will be insignificant when movement is limited to a small area. However, for vessels traveling long distances, say cargo vessels, the curvature of the Earth must be taken into account.

The relation between these two frames is shown in Figure 2.2. Both are right-handed coordinate systems. The coordinates of the BODY frame will be labeled  $(x^b, y^b, z^b)$  where  $x^b$  represents the longitudinal axis,  $y^b$  is the transversal axis, and  $z^b$  is the normal axis (pointing into the paper). Likewise, the NED coordinates are labeled  $(x^n, y^n, z^n)$  where  $x^n$  represents an axis pointing north,  $y^n$  is an axis pointing east, and  $z^n$  points in the same direction as  $z^b$ .

It should be stressed that the NED coordinate system can be treated as inertial reference frame since the angular velocity of earth is small<sup>2</sup>. This implies that Newton's laws of motion apply (with a good approximation) to this frame. The body frame is non-inertial, as it is fixed to a vessel that is free to accelerate or rotate [19].

One key insight is that the velocities in surge, sway, and yaw can be directly converted to velocities in the NED coordinate system. This kind of operation can be performed

---

<sup>1</sup>NED: North East Down.

<sup>2</sup>The Earth's angular velocity is  $7.2921 \times 10^{-5}$  rad/s.

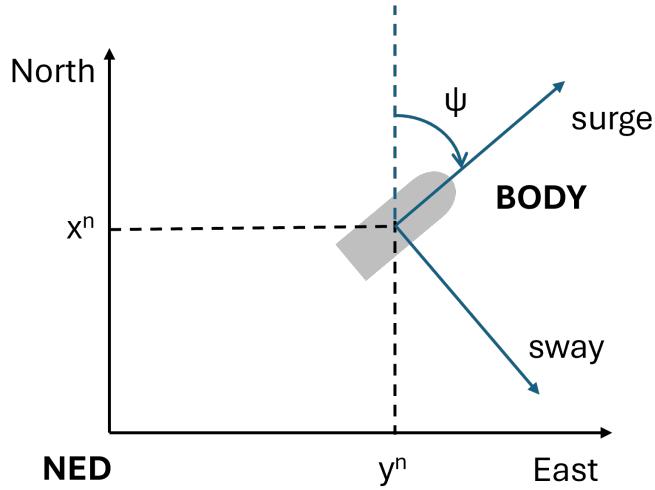


Figure 2.2: Shows the relation between the geographical NED frame and the BODY frame. The NED frame is fixed to the surface of the earth while the BODY frame is fixed to the vessel. The surge axis pointing in the longitudinal direction and the sway axis pointing in the transversal direction. Both frames have a third axis pointing into the paper, but not shown here.

through the use of a rotation matrix. This matrix can be deduced from Euler's rotation theorem (see [2] for details). The rotation matrix is defined in equation (2.1).

$$R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

where  $\psi$  represents the heading angle, that is, the angle between the north axis of the NED frame and the surge axis of the BODY frame of the vessel (as shown in Figure 2.2). The kinematics of motion for a 3-DoF ship can then be described by equation (2.2).

$$\dot{\eta} = R(\psi)v \quad (2.2)$$

where  $v = [u, v, r]^T$  which represents the velocity in surge ( $u$ ), velocity in sway ( $v$ ), and angular velocity ( $r = \dot{\psi}$ ). All of these quantities are related to the BODY frame. Here  $\eta = [x^n, y^n, \psi]^T$  is a vector that contains the position in the north ( $x^n$ ) and east ( $y^n$ ) direction with respect to a NED frame, together with the vessel heading ( $\psi$ ). The rotation matrix transforms the velocity components of the BODY frame to the corresponding velocity components in the NED frame. From this perspective, it is clear that (2.2) is simply stating that the derivative of the north/east position is equal to the velocity in the north/east direction ( $\dot{x}^n = v_x^n$  and  $\dot{y}^n = v_y^n$ ), with an analogous interpretation for the yaw angle ( $\dot{\psi} = r$ ).

The notation here will follow the SNAME standard as outlined in [2] and [20]. For easy reference, the variable notation scheme is given in Table 2.1.

Table 2.1: SNAME standard for variable naming. SNAME is an acronym for Society of Naval Architects and Marine Engineers.

Description	Variable name	Description	Variable name
Force in surge	$X$	Angular velocity in roll	$p$
Force in sway	$Y$	Angular velocity in pitch	$q$
Force in heave	$Z$	Angular velocity in yaw	$r$
Moment in roll	$K$	Position north	$x^n$
Moment in pitch	$M$	Position east	$y^n$
Moment in yaw	$N$	Vertical position	$z^n$
Velocity in surge	$u$	Roll angle	$\phi$
Velocity in sway	$v$	Pitch angle	$\theta$
Velocity in heave	$w$	Yaw angle	$\psi$

### 2.1.1 Modeling

It is important to understand that the dynamics of the vessel is best understood from a BODY frame perspective. The vessel will generate forces and moment due to thrusters, but will also be subjected to a set of external disturbances due to wind, waves, and current.

A challenge is that not all forces are known. It is common to use one or more wind sensors (anemometer) on a vessel, but less common to measure waves and current. Wind, wave, and current models can be complicated, and a deep dive into these issues is beyond the scope of this text.

A general ship model derived by Fossen [2] is shown in (2.3) and (2.4).

$$\dot{\eta} = J_\Theta(\eta)v \quad (2.3)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) + g_0 = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \quad (2.4)$$

The first equation (2.3) is responsible for computing the position and attitude of the vessel in a NED frame from the linear and angular velocities defined in the BODY frame. This is the same process as shown in (2.2). In equation (2.4)  $M$  is the inertia term,  $C(v)$  is due to the Coriolis effect<sup>3</sup>,  $D(v)$  is caused by hydrodynamic damping,  $g(\eta)$  represents buoyancy and gravitational forces, and  $g_0$  is a term associated with ballast tanks.  $\tau$  are forces/torques generated by propellers, tunnel thrusters, rudders, and azimuths. This is a general 6-DoF freedom model. While this model will not be used in the thesis, it does illustrate the complexity and non-linearity inherent in simulating high-fidelity models. Here, the OSV and supply models (both by Fossen) will be used together with the Balchen model. All of these models will be described in more detail later.

---

<sup>3</sup>The Coriolis force is a pseudo force that occurs due to Earth's surface being a non-inertial frame.

## 2.1.2 Determining position

Setpoints must be defined with respect to the NED coordinate system. If the setpoint is defined with respect to surge and sway (in the BODY coordinate system), the final position would depend on the dynamics of the ship and the controller tuning. This is because the yaw angle changes over time; thus the algebraic relation between the north/east axis and the surge/sway axis will vary. This property is clear from the rotation matrix due to its heading ( $\psi$ ) dependency. An example of this is given in the textblock below.

**Example:** Imagine a ship pointing directly northward. It is then supplied with a setpoint that is 20 meters in surge, 10 meters in sway, and with a 90 degree heading change. One possibility is that the ship moves 20 meters north and 10 meters east followed by a heading change of 90 degrees. Another possibility is that the ship first performs the attitude change, moves 20 meters east, then 10 meters south. Thus, we end up in two very different positions. Normally, the heading would not only change at the start or the beginning of the operation, but occur continuously. Depending on how the heading angle changes over time in conjunction with the surge and sway movement will determine the final position of the vessel.

This example is illustrated in Figure 2.3. Under the assumption of absence of overshoot,

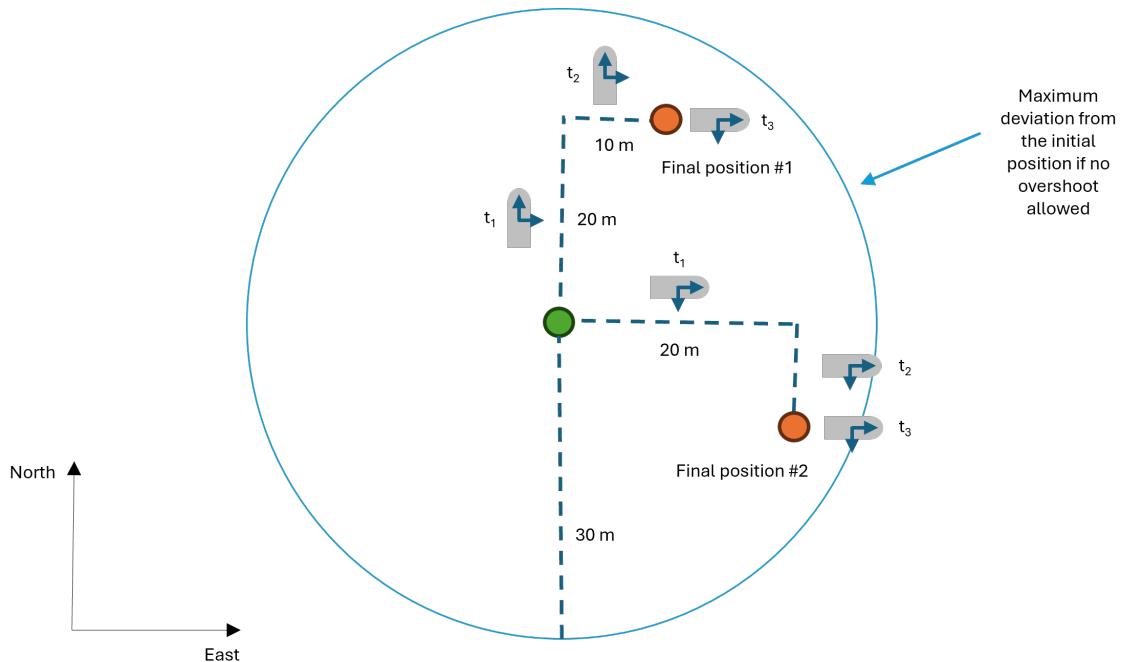


Figure 2.3: Depicts how the final position in a north-east coordinate system can vary for a particular setpoint given in BODY coordinates, depending on the dynamics of the control action. The body setpoint can end up placing the ship in any position within the circle.

the final final destination with respect to the initial position (origin) is within a circle of radius  $r = |x_{\text{setpoint}}^b| + |y_{\text{setpoint}}^b|$ . The longest distance is obtained when both the surge and sway setpoint is used to move in the same direction (straight line).

It can be compared to a car that drives in an empty parking lot. The surge setpoint will be the length traveled (measured by the odometer). Obviously, the final destination will depend on how the steering wheel is turned during driving.

The solution to this problem is to define the setpoint in the NED reference frame. That way, the dynamics of the ship will not interfere with the final position; this is defined relative to a fixed spot on the Earth's surface defined by some latitude and longitude. Alternatively, a new setpoint can be calculated in the BODY frame at each timestep based on the original setpoint in the NED frame.

## 2.2 Wind, waves and current

A ship is subjected to external disturbances. These disturbances are related to the wind, waves, and current. They introduce forces and moments that will influence the vessel and hence must be dealt with by the control system. If some or all of these external disturbances are unknown, an integral action can be introduced to estimate the effect of these disturbances.

The forces and moment due to environmental disturbances are functions of the vessel parameters and the attitude (heading) of the vessel, but the weather phenomena themselves are independent of the ship. For example, the wind velocity should not depend on the vessel heading, but the wind forces and moment acting on the vessel depend on the wind angle relative to the vessel heading and the lateral and frontal projection of the vessel. It follows from this that the weather phenomena themselves should be modeled in a geographic frame such as the NED frame.

What follows is a short discussion of wind, waves, and current. No DP vessel can guarantee position-keeping under any weather conditions, but the DP capability plot offers theoretical limits on the disturbances the vessel can withstand without loss of position. Knowing such limits is important when operating in areas where loss of position can cause damage to other surrounding structures or loss of life.

In the remainder of this thesis it will be assumed that the forces and moments resulting from wind and waves are known. However, the current is unknown to the controller and must be estimated.

## Wind forces

Wind speed and angle can be measured using a wind sensor, also known as a wind vane anemometer.<sup>4</sup> Knowing the wind speed and angle of attack, it is possible to estimate the forces in surge/sway and the moment in yaw due to wind. This thesis will use the wind formulas described in Fossen [2]. Here, the wind force is described by equation (2.5).

$$\tau_{\text{wind}} = \frac{1}{2} \rho_a V_{rw}^2 \begin{bmatrix} -c_x \cos(\gamma_{rw}) A_F \\ c_y \sin(\gamma_{rw}) A_L \\ c_n \sin(2\gamma_{rw}) A_L L_{oa} \end{bmatrix} \quad (2.5)$$

This is the wind forces with respect to the BODY coordinate frame  $\tau_{\text{wind}} = [X_{\text{wind}}, Y_{\text{wind}}, N_{\text{wind}}]^T$ . The air density is given by  $\rho_a$  (temperature dependent).  $V_{rw}$  is the relative wind velocity. Here,  $c_x$ ,  $c_y$ , and  $c_n$  are the wind coefficients with respect to surge, sway, and yaw. Typical values for wind coefficients are shown in Table 2.2. Determining wind coefficient on any given vessel is not trivial, and there exist many methods for identifying these parameters, but it is outside the scope of this thesis. Here we will use the average of the coefficients suggested in Table 2.2. A more in-depth discussion of wind coefficients for different types of vessels is given in [2].  $A_F$  and  $A_L$  represent the frontal

Table 2.2: Typical values of wind coefficients based on experimental experience. These intervals are suggested by Fossen [2], but it is noted these values should be used with care.

Coefficient	Force/Moment	Direction	Min. value	Max. value
$c_x$	Force	Surge	0.50	0.90
$c_y$	Force	Sway	0.70	0.95
$c_n$	Moment	Yaw	0.05	0.20

and lateral projection areas of the vessel (that is, the area above the water level of the vessel subjected to wind force in surge and sway direction).  $L_{oa}$  is known as length overall, representing the maximum length from aft to bow of the hull along the waterline. The frontal projection, lateral projection, and overall length are shown in Figure 2.4. Lastly, the  $\gamma_{rw}$  is the wind angle of attack relative to the bow. This is shown in Figure 2.5. In order to calculate the wind forces and moment, it is necessary to know the wind direction and velocity at each time step. It will be assumed that the wind velocity and angle of attack can be simulated using a random walk algorithm. Discrete gaussian random walk can be simulated using equations (2.6) and (2.7).

$$V_{w,k+1} = V_{w,k} + \eta_{1,k} \quad (2.6)$$

$$\beta_{w,k+1} = \beta_{w,k} + \eta_{2,k} \quad (2.7)$$

---

<sup>4</sup>Wind is commonly classified according to the Beaufort number, spanning from 0 (calm weather - wind speed of 0-1 knots) to 12 (hurricane - wind speed of more than 65 knots).

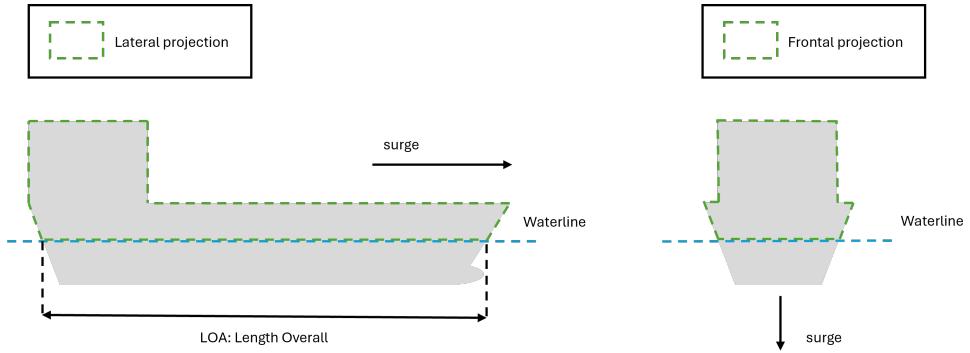


Figure 2.4: Depicts the lateral and frontal projection of the ship above the waterline. This is the area affected by wind. It also shows the meaning of length overall: The maximum distance between the aft and bow part of the hull along the waterline.

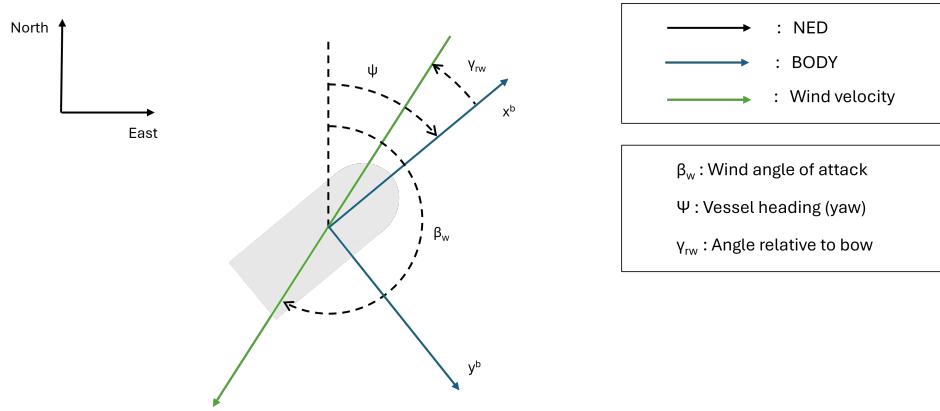


Figure 2.5: Depicts different angles of interest when calculating wind forces.

$V_w$  is here the wind velocity,  $\beta_w$  is the wind angle of attack, and  $\eta_i$  is zero mean Gaussian noise. The decomposed wind velocity in surge and sway can be calculated using the formula (2.8) and (2.9). Note how the heading of the vessel ( $\psi$ ) and wind direction ( $\beta_w$ ) are included in the equations.

$$u_w = V_w \cos(\beta_w - \psi) \quad (2.8)$$

$$v_w = V_w \sin(\beta_w - \psi) \quad (2.9)$$

It is now straightforward to calculate the relative wind velocities, as shown in (2.10) and (2.11).

$$u_{rw} = u - u_w \quad (2.10)$$

$$v_{rw} = v - v_w \quad (2.11)$$

From this the relative wind velocity ( $V_{rw}$ ) and relative angle of attack ( $\gamma_{rw}$ ) with respect to the bow can be calculated as shown in (2.12) and (2.13) respectively.

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \quad (2.12)$$

$$\gamma_{rw} = -\text{atan2}(v_{rw}, u_{rw}) \quad (2.13)$$

From these equations it is clear that vessel with zero velocity will be subjected to wind force given a non-zero relative wind velocity. Likewise, a vessel moving in the same direction and speed as the wind velocity will not experience a wind force.

## Waves

This section is based on information from [2]. Wave-induced forces can be divided into two categories. The first-order wave forces causing zero mean oscillatory motion and the second-order wave forces responsible for a slowly varying net wave drift force. Only second-order wave forces should be compensated for by the DP system. The high-frequency first-order wave forces can be removed using a low-pass filter. This is related to the concept of *wave filtering*, i.e. separating the first-order wave forces from the slowly varying second-order drift force.

In this text, only second-order (low-frequency) wave forces will be considered. Calculating the wave forces is not trivial and is dependent on complicated wave spectra. Since this text is primarily concerned with control, for simplicity it will be assumed that the wave forces and the yaw moment can be simulated using a random walk algorithm in the NED frame. These forces are then converted to the BODY frame using the rotation matrix. This force transformation is given in the following equation:

$$\tau_{\text{wave}}^b = R^T(\psi) \tau_{\text{wave}}^n \quad (2.14)$$

Here,  $\tau_{\text{wave}}^b$  and  $\tau_{\text{wave}}^n$  are the wave forces and torque in BODY and NED coordinates, respectively. It should be noted that the inverse of the rotation matrix can be obtained by transposing the original matrix ( $R^{-1}(\psi) = R^T(\psi)$ ).

## Current

The ocean current is simply the movement of water. There are three main drivers for the water current [2], [21];

- **Wind** - Wind acts on the surface of the sea, causing water to move.
- **Density variation** - The density variation causes water current. The density variation itself is due to salinity and temperature differences in the water.

- **Tides** - Currents driven by the gravity of the moon. The fall and rise of seawater induce sea current.

Just as in the case of waves, the current will be modeled as a random walk process for forces in north and east directions in the NED coordinate system. Then it is transformed into the BODY coordinate system using the transposed rotation matrix,  $R^T(\psi)$ .

## 2.3 Supply model

In a paper published by Thor Inge Fossen et al. [15] a linear state space model was identified for a supply ship. The model parameters were estimated using the data obtained from a real supply ship performing predefined maneuvers (to improve parameter convergence). An extended Kalman filter is used to calculate the unknown parameters. The identified model took the state space form shown in equation (2.15).

$$\dot{\mathbf{v}} = A\mathbf{v} + B\tau \quad (2.15)$$

Here,  $A$  and  $B$  are linear, time-invariant matrices associated with the ship dynamics in the BODY frame.  $\tau$  represents the forces and moment input from the thrusters (and external disturbances). The state space vector includes the linear and angular velocities in the BODY frame, i.e.  $\mathbf{v} = [u, v, r]$ . The inertial mass (including the hydrodynamic added mass),  $M$ , and the linear hydrodynamic damping,  $D$ , are related to the state transition and input matrix as follows:  $A = -M^{-1}D$  and  $B = M^{-1}$ . However, to obtain the motion in a fixed NED frame, it is necessary to extend the state vector to include the north ( $x^n$ ) and east ( $y^n$ ) positions together with the heading angle ( $\psi$ ). This is done using the rotation matrix described in equation (2.1). The results in a non-linear state space model of system order 6. The nonlinearity is due to the sine and cosine functions in the rotation matrix.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\eta} \\ \dot{\mathbf{v}} \end{bmatrix} = \begin{bmatrix} \dot{x}^n \\ \dot{y}^n \\ \dot{\psi} \\ \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{3 \times 3} & R(\psi) \\ 0_{3 \times 3} & A \end{bmatrix}}_{\tilde{A}(\psi)} \mathbf{x} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{\tilde{B}} \tau \quad (2.16)$$

Here,  $\eta = [x^n, y^n, \psi]$  represents the position in a north-east plane and vessel orientation (heading). Each matrix entry for  $\tilde{A}$  and  $\tilde{B}$  is shown in equation (2.17).

$$\tilde{A} = \begin{bmatrix} 0 & 0 & 0 & \cos \psi & -\sin \psi & 0 \\ 0 & 0 & 0 & \sin \psi & \cos \psi & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_{11} & a_{12} & a_{13} \\ 0 & 0 & 0 & a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 & a_{31} & a_{32} & a_{33} \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \quad (2.17)$$

Here,  $a_{ij}$  and  $b_{ij}$  are the values identified by Fossen, which describe the dynamics of the vessel. The values for  $a_{ij}$  are listed below:

$$\begin{array}{lll} a_{11} = -0.0114 & a_{12} = 0 & a_{13} = 0 \\ a_{21} = 0 & a_{22} = -0.0225 & a_{23} = -0.0819 \\ a_{31} = 0 & a_{32} = 0 & a_{33} = -0.0871 \end{array} \quad (2.18)$$

The entries in the input matrix  $b_{ij}$  are shown below:

$$\begin{array}{lll} b_{11} = 0.1478 \cdot 10^{-6} & b_{12} = 0 & b_{13} = 0 \\ b_{21} = 0 & b_{22} = 0.0902 \cdot 10^{-6} & b_{23} = 0.0007 \cdot 10^{-6} \\ b_{31} = 0 & b_{32} = 0.0007 \cdot 10^{-6} & b_{33} = 0.0002 \cdot 10^{-6} \end{array} \quad (2.19)$$

Some interesting observations are listed below.

- There are three integrators associated with the calculation of position and heading. The lower right  $3 \times 3$  quadrant represents the dynamics associated with hydrodynamic damping. The negative values of the entries indicate that the velocity and rotation would slow down over time when not subjected to external forces.
- If total surge and sway movement is of interest, the rotation matrix can be replaced by an identity matrix. The governing equations would then take the following form:

$$\dot{x} = \begin{bmatrix} x^b \\ y^b \\ \psi \\ \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} = \underbrace{\begin{bmatrix} 0_{3 \times 3} & I \\ 0_{3 \times 3} & A \end{bmatrix}}_{\tilde{A}} x + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{\tilde{B}} \tau \quad (2.20)$$

- The small values of entries in the input matrix reflect the fact that large forces and torque are required to move supply vessels that can have thousands of metric tons in mass.

### 2.3.1 Eigenvalues of state transition matrix

It is interesting to note that different values of  $\psi$  in the rotation matrix ( $R(\psi)$ ) do not affect the eigenvalues of the state transition matrix, as will be shown here.

It is well known that the eigenvalues,  $\lambda$ , can be calculated from  $\det(A - \lambda I) = 0$ . This is similar to the roots of the characteristic equation. The calculation of this determinant is

shown below.

$$\det(A - \lambda I) = \begin{vmatrix} -\lambda & 0 & 0 & \cos \psi & -\sin \psi & 0 \\ 0 & -\lambda & 0 & \sin \psi & \cos \psi & 0 \\ 0 & 0 & -\lambda & 0 & 0 & 1 \\ 0 & 0 & 0 & q_{11} - \lambda & a_{12} & a_{13} \\ 0 & 0 & 0 & a_{21} & a_{22} - \lambda & a_{23} \\ 0 & 0 & 0 & a_{31} & a_{32} & a_{33} - \lambda \end{vmatrix} \quad (2.21)$$

$$\det(A - \lambda I) = (-\lambda)(-\lambda)(-\lambda)[(a_{11} - \lambda)((a_{22} - \lambda)(a_{33} - \lambda) - a_{23}a_{32})] \quad (2.22)$$

$$- a_{21}(a_{12}(a_{33} - \lambda) - a_{13}a_{32}) \quad (2.23)$$

$$+ a_{31}(a_{12}a_{23} - a_{13}(a_{22} - \lambda))] \quad (2.24)$$

This results in the following characteristic equation:

$$p(\lambda) = -\lambda^6 + \alpha_5\lambda^5 + \alpha_4\lambda^4 + \alpha_3\lambda^3 + \alpha_2\lambda^2 + a_1\lambda^1 + a_0 \quad (2.25)$$

Where:

$$\alpha_0 = 0$$

$$\alpha_1 = 1$$

$$\alpha_2 = 0$$

$$\alpha_3 = a_{11}a_{22}a_{33} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31}$$

$$\alpha_4 = a_{12}a_{21} - a_{11}a_{22} - a_{11}a_{33} + a_{13}a_{31} - a_{22}a_{33} + a_{23}a_{32}$$

$$\alpha_5 = a_{11} + a_{22} + a_{33}$$

Since the characteristic equation is independent of the heading angle of the ship, it is clear that the eigenvalues will be constant. This is natural since the vessel model represents BODY dynamics. The eigenvalues are:

$$\lambda = \begin{bmatrix} -0.0114 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ -0.0871 \\ -0.0225 \end{bmatrix} \quad (2.26)$$

The three zero eigenvalues are due to the integration process occurring when velocities in surge, sway, and yaw are integrated to obtain the north position, east position, and yaw angle through the use of the rotation matrix.

## 2.4 DP-model

For DP application Fossen [2] suggests the model defined in equations (2.27), (2.28) and (2.29).

$$\dot{\eta} = R(t)v \quad (2.27)$$

$$M\dot{v} + Dv = R^T(t)b + \tau + \tau_{wind} + \tau_{wave} \quad (2.28)$$

$$\dot{b} = 0 \quad (2.29)$$

where

$$\eta = [x^n, y^n, \psi] \quad (2.30)$$

$$v = [u, v, r] \quad (2.31)$$

Here,  $R(\psi)$  has been replaced by  $R(t)$  since it is assumed that the heading angle of the vessel can be accurately measured using a gyrocompass. This assumption was suggested by Fossen in [2] and will be applied in all simulations in this text. The term  $b$ , is a three-dimensional vector representing the forces and moment of the unknown current disturbance. This term causes integral action; hence it really represents all dynamics not accounted for by the model and all unknown disturbances, including current.  $\tau$  is a vector of the net propulsive forces and torque generated by the thrusters ( $[X_{thr}, Y_{thr}, N_{thr}]$ ). Here,  $M$  represents the inertia matrix ( $M_{RB}$ ) and the added mass ( $M_A$ ).

$$M = M_{RB} + M_A \quad (2.32)$$

$D$  is the damping matrix. Nonlinear damping is not included in this model due to the close-to-zero-speed assumption. In [2], it is shown that the linear damping dominates at low velocities. If this assumption is violated, i.e. the vessel starts moving at a significant speed, the validity of the DP-model will be questionable. Including a non-linear damping term would increase the complexity of the model and make the control process more difficult.

As shown by Fossen [2], the above model can be rewritten into the following state-space form:

$$\dot{x} = \begin{bmatrix} 0_{3 \times 3} & R(t) & 0_{3 \times 3} \\ 0_{3 \times 3} & -M^{-1}D & M^{-1}R^T(t) \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} x + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} \tau + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} (\tau_{wind} + \tau_{wave}) \quad (2.33)$$

$$y = [I_{3 \times 3} \ 0_{3 \times 3} \ 0_{3 \times 3}] x + \epsilon \quad (2.34)$$

Here,  $\epsilon$  represents noise and  $x = [\eta^T, v^T, b^T]^T$ . This is the same model structure as the supply model, but includes an integral term named  $b$ .

This model will be applied to the Kalman filter to estimate position, heading, velocities, and disturbance term ( $b$ ). The input will then be the position (latitude and longitude)

and the heading of the vessel. These values are typically obtained using GPS and gyro-compass.

The model will also be used in model predictive control (MPC). The challenge then is to estimate the future horizon that depends on the rotation matrix  $R(t)$ . Although knowledge of this heading value is known with high precision at the current time, the future value is not known and must be estimated. Some methods for doing so will be discussed later.

## 2.5 LQ optimal control with integral action

This section looks into linear quadratic (LQ) optimal control. This is a method that optimizes the control input with respect to some cost function,  $J$ . A special form of linear quadratic control will be used that depends on past output ( $y_{k-1}$ ) and state deviation ( $\Delta x$ ) to calculate the change in control input ( $\Delta u$ ). This is known as the LQ optimal control on deviation form and was developed by Dr. David Di Ruscio.

### 2.5.1 LQ optimal control theory

The discussion below is based on the LQ optimal control theory as described by Ruscio in [22] and [23]. The LQ optimal control problem can be applied to the state-space model described in equations (2.35) and (2.36).

$$x_{k+1} = Ax_k + Bu_k + v_k \quad (2.35)$$

$$y_k = Cx_k + w_k \quad (2.36)$$

Here  $v$  and  $w$  are unknown disturbances,  $A$  is the  $(n \times n)$  state transition matrix,  $B$  is the  $(n \times r)$  input matrix, and  $C$  is the  $(m \times n)$  output matrix. Here,  $n$  is the system order,  $r$  is the number of inputs, and  $m$  is the number of outputs. The goal is to minimize a cost function,  $J$ . The cost function considered here will take the form given in equation (2.37).

$$\min_u J = \min_u \frac{1}{2} \left[ \sum_{k=i}^{\infty} ((y_k - r)^T Q (y_k - r) + \Delta u_k^T P \Delta u_k) \right] \quad (2.37)$$

Here,  $Q$  and  $P$  are weighting matrices that determine the controller behavior. One of the challenges with the state-space model in equations (2.35) and (2.36), is the presence of unknown disturbances,  $v_k$  and  $w_k$ . Ideally, we want to remove these. In the case of dynamic positioning these unknown disturbances could represent contributions from wind, waves, and current. Assuming that process noise,  $v_k$  and measurement noise,  $w_k$

are slowly varying, the state space model given in equations (2.35) and (2.36) can be rewritten to deviation form<sup>5</sup>.

$$\Delta x_{k+1} = A\Delta x + B\Delta u_k + \underbrace{v_{k+1} - v_k}_{\approx 0} = A\Delta x_k + B\Delta u_k \quad (2.38)$$

$$y_k = y_{k-1} + C\Delta x_k + \underbrace{w_k - w_{k-1}}_{\approx 0} = y_{k-1} + C\Delta x_k \quad (2.39)$$

Here, the state deviation is defined as  $\Delta x_{k+1} = x_{k+1} - x_k$ . Due to the assumption of slowly varying disturbance terms  $v_k$  and  $w_k$  it follows that the difference between two consecutive disturbance terms will be close to zero. Removal of these terms adds integral action to the control algorithm. Now, an augmented state space model can be made as shown below:

$$\underbrace{\begin{bmatrix} \Delta x_{k+1} \\ y_k - r \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} A & 0_{n \times m} \\ C & I_{m \times m} \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} B \\ 0_{m \times r} \end{bmatrix}}_{\tilde{B}} \Delta u_k \quad (2.40)$$

$$\underbrace{y_k - r}_{\tilde{y}} = \underbrace{\begin{bmatrix} C & I_{m \times m} \end{bmatrix}}_{\tilde{D}} \underbrace{\begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix}}_{\tilde{x}} \quad (2.41)$$

In this equation, the term  $y_{k-1} - r$  refers to the difference between the previous output and the corresponding setpoint. From [22] and [23] it can be shown that the controller gain can be calculated from equation (2.42).

$$\Delta u_k = G\tilde{x}_k = \begin{bmatrix} G_1 & G_2 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} = G_1\Delta x_k + G_2(y_{k-1} - r) \quad (2.42)$$

We note that the first term  $G_1\Delta x_k$  represents state feedback and the second term  $G_2(y_{k-1} - r)$  is the output feedback. In real systems, we usually do not have perfect information about the current state, and hence it must be estimated. In this text a Kalman filter will be used for this purpose. If the filter estimate is denoted  $\hat{\Delta x}_k = \hat{x}_k - \hat{x}_{k-1}$  then the equation to calculate  $\Delta u_k$  becomes as in equation (2.43).

$$\Delta u_k = G_1\hat{\Delta x}_k + G_2(y_{k-1} - r) \quad (2.43)$$

The control input can be derived from the above equation by keeping track of the previous control input as shown in equation (2.44).

$$u_k = \Delta u_k + u_{k-1} = G_1\hat{\Delta x}_k + G_2(y_{k-1} - r) + u_{k-1} \quad (2.44)$$

With knowledge of the previous control input ( $u_{k-1}$ ) and output ( $y_{k-1}$ ) in combination with the estimated state deviation ( $\hat{\Delta x}_k$ ), the current control input ( $u_k$ ) can be obtained.

---

<sup>5</sup>There is also another implicit assumption made here when applied to the supply model, since the transition matrix can vary, it is assumed that for short timesteps  $A(t_i) \approx A(t_{i+1})$ .

This formulation of a combined linear quadratic regulator in combination with Kalman filter is known under the name Linear Quadratic Gaussian (LQG) controller. Notice that the filter and controller can be developed independently. The filter would still function even under open-loop control.

## 2.6 Kalman Filter

The discrete Kalman filter is applied to the DP model in equations (2.33) and (2.34). Using this model, the Kalman filter is responsible for estimating the north-east position, heading, linear velocities in surge and sway, and angular velocity in yaw (heading rate of change). It also estimates state  $b$ , representing unknown disturbances and modeling error. The inputs to the filter are the measurements from the GPS, gyrocompass, and external wind and wave forces. Any bias in wind or wave measurements will be compensated for by the integral term  $b$ .

The Kalman filter computes the optimal state estimate in the sense that it minimizes the covariance of the estimation error. Assuming all modeling assumptions are valid, the Kalman filter promises that  $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$  will be minimized. Here,  $x_k$  represents the true state value and  $\hat{x}_k$  is the optimal state estimate from the Kalman filter.

The filter can be written on an apriori-aposteriori form. The apriori estimate is the uncorrected state value obtained from using the mathematical model. However, the aposteriori estimate corrects the apriori estimate by taking into account the actual measurements obtained. The apriori state for the timestep  $k$  is given by equation (2.48). Notice how it depends on the previous input ( $u_{k-1}$ ) and optimal estimate ( $\hat{x}_{k-1}$ ). The second step (equation (2.49)) calculates the estimated output based on the apriori estimate of the states. Finally, in equation (2.50) the optimal state estimate is obtained using the current observed measurements to correct the state estimate. The Kalman gain,  $K$ , is a measure of how much the new measurements are to be trusted compared to the results from the

process model<sup>6</sup>.

$$\bar{x}_k = A\hat{x}_{k-1} + Bu_{k-1} \quad (2.48)$$

$$\bar{y}_k = C\bar{x}_k \quad (2.49)$$

$$\hat{x}_k = \bar{x}_k + K(y_k - \bar{y}_k) \quad (2.50)$$

The state transition matrix and input matrix is obtained by discretizing Fossen's DP model (see equation (2.33)). Since it will be assumed that only GPS and gyrocompass measurement are used in estimating the states, the output matrix, takes the following form:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.51)$$

The Kalman gain itself is obtained from solving a Riccati equation. The details can be found in a paper by Ruscio [24].

The input will here consist of thruster forces and external disturbances  $u = \tau + \tau_{\text{wind}} + \tau_{\text{wave}}$ .

A simplified approach to how a Kalman filter works is shown in Figure 2.6. Note how the correction step is essential to reduce the uncertainty. If no measurements were available, the uncertainty would only grow over time.

Note that if no position measurements were available, the system would be operating in dead-reckoning mode. Even if other non-position measurement were available, such as linear and angular velocity measurements, the system would still fail to estimate position accurately over time. This is a known problem when performing dead-reckoning, a fallback mode in cases where the position reference system(s) fails. This is typically not a problem for surface vessels which typically operate with more than one position reference system,

---

<sup>6</sup>The trust with respect to measurement and process model is defined by specifying the entries of the covariance matrices of measurement noise and process noise. It is assumed that the true state and measurement can be described by equation (2.45) and (2.46) respectively.

$$x_{k+1} = Ax_k + Bu_k + v_k \quad (2.45)$$

$$y_k = Cx_k + w_k \quad (2.46)$$

Where  $v_k$  and  $w_k$  represent white noise processes that are uncorrelated.

$$E \left[ \begin{bmatrix} v_k \\ w_k \end{bmatrix} \begin{bmatrix} v_k \\ w_k \end{bmatrix}^T \right] = \begin{bmatrix} V & 0 \\ 0 & W \end{bmatrix} \quad (2.47)$$

Here,  $V$  and  $W$  are considered to be diagonal matrices representing process and measurement noise.  $V$  and  $W$  are typical tuning parameters used to obtain good filter performance. These covariance matrices are used when calculating the Kalman gain matrix.

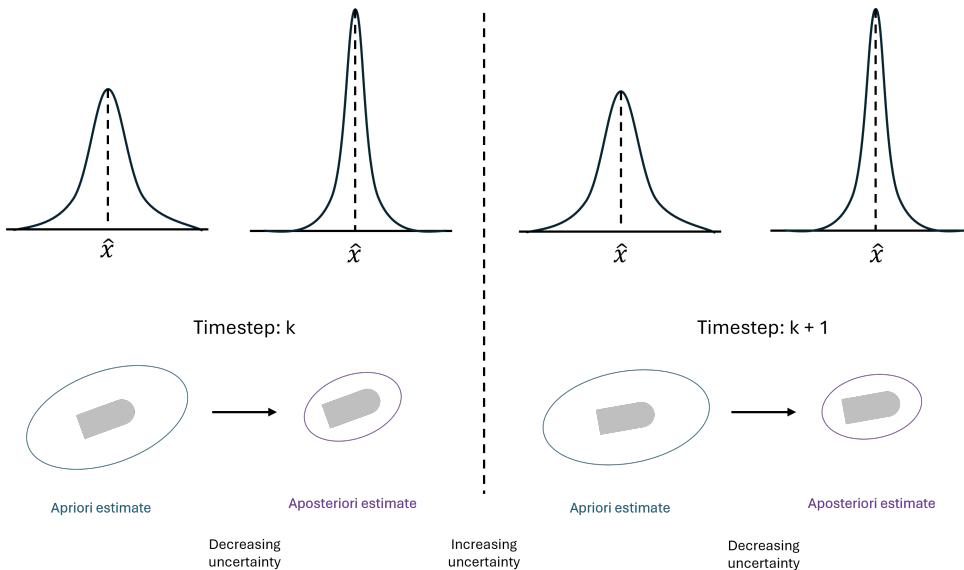


Figure 2.6: Simplified example of how the Kalman filter works with respect to a seafaring vessel. The top part of the figure depicts the gaussian curve for the north position. Notice that the uncertainty of the position estimate increases as the apriori estimate is calculated and decreases when the optimal estimate is computed (measurement correction). The bottom part of the figure represents the vessel, the ellipses represents uncertainty of the vessel position (the line represents one standard deviation).

but for submarines (or underwater vehicles in general) this can be more challenging, relying on other systems than GPS.

It is important to note that the state transition matrix will depend on the heading angle due to the conversion of BODY coordinates to NED coordinates, which implies that  $A = A(\psi)$ . This is solved in a similar manner to the controller, assuming that the measurements from the gyrocompass are close to exact, hence  $A(\psi) = A(t)$ . Since the model is in continuous form, it must first be discretized before it is used in the Kalman filter. The dependence on the heading angle means that the optimal Kalman gain  $K$  must be recomputed when the vessel heading changes.

When the Kalman filter is applied to the models identified by system identification, the state-space model used is given in equations (2.131) and (2.132).

## 2.7 Vessel models

Different models will be described in this section. The supply model is the main vessel model applied in this thesis, but the OSV and the Balchen model are used to test system identification.

## 2.7.1 Supply model

The simplest model is the supply model suggested by Fossen [15]. This was discussed earlier in Section 2.3. It is the vessel model applied when testing different control algorithms, path-following, trajectory tracking, green DP, and thruster allocation.

This model is available in the **MSS**<sup>7</sup> MATLAB toolbox.

## 2.7.2 Offshore Support Vessel (OSV) model

The OSV model can be formulated as shown in equations (2.52) and (2.53).

$$\dot{\eta} = R(\psi)v \quad (2.52)$$

$$\dot{v} = \begin{bmatrix} rv_c \\ -ru_c \\ 0 \end{bmatrix} + M^{-1}(\tau + \tau_{\text{wind}} + \tau_{\text{wave}} - C(v_r)v_r - Dv_r - D_n(v_r)v_r) \quad (2.53)$$

Here,  $v_r$  is the relative velocity with respect to the ocean current. Thus,  $v_r = v - [u_c, v_c, 0]^T$ . The heading rotation is denoted by  $r$ . The surge and sway velocity are named  $u_c$  and  $v_c$ . In this equation the matrices  $D$  and  $D_n$  are the linear and non-linear hydrodynamic damping matrices, respectively. The term  $C(v_r)$  is the Coriolis-centripetal matrix. The exact details of how these matrices are defined are given in the textbook by Fossen [2]. It gives a detailed description of the physics involved. The important point here is that the model is non-linear.

In the **MSS** toolbox the function interface to this model is defined with respect to thruster inputs, not the net forces and moment acting on the vessel. The input to the model is in rotations per second (RPS) for each thruster. Furthermore, there are four thrusters: 2 tunnel thrusters in the bow and two azimuths in the stern. The thruster configuration is shown in Figure 2.7.

## 2.7.3 Balchen vessel model

The seminal paper by Balchen [7] outlines a dynamic positioning system based on linear quadratic optimal control and Kalman filtering for state estimation. The paper includes a mathematical model to describe the dynamics of vessels. This is a low-frequency model<sup>8</sup>, i.e. concerned with forces caused by external disturbances (wind and current),

---

<sup>7</sup>MSS (Marine System Simulator) is a toolbox developed by Thor Inge Fossen. The supply model and the OSV model are both defined in this toolbox.

<sup>8</sup>Balchen also suggest a high-frequency model that represents oscillatory motion (typically due to waves) by modeling surge, sway and yaw rate as harmonic oscillators including a gaussian noise terms.

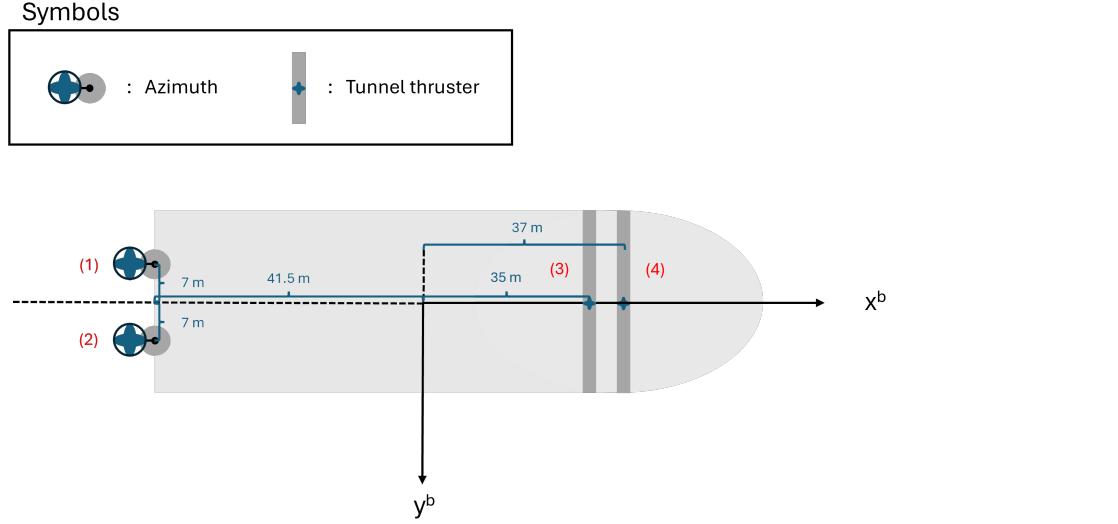


Figure 2.7: Thruster configuration of OSV model. Here (1) and (2) represents two stern azimuth thruster, (3) and (4) depicts two bow tunnel thrusters.

and propulsive forces generated by thrusters. The model equations are given in (2.54), (2.55), (2.56), (2.57), (2.58) and (2.59).

$$\dot{x}^b = u \quad (2.54)$$

$$\dot{y}^b = v \quad (2.55)$$

$$\dot{\psi} = r \quad (2.56)$$

$$\dot{u} = -\frac{d_1}{m_1}|u - u_c|(u - u_c) + \frac{1}{m_1}(u_1 + v_1) + \eta_1 \quad (2.57)$$

$$\dot{v} = -\frac{d_2}{m_2}|v - v_c|(v - v_c) + \frac{1}{m_2}(u_2 + v_2) + \eta_2 \quad (2.58)$$

$$\dot{r} = -\frac{d_3}{m_3}|r|r - \frac{d_4}{m_3}|v - v_c|(v - v_c) + \frac{1}{m_3}(u_3 + v_3 + n_c) + \eta_3 \quad (2.59)$$

Here,  $\psi$  is the vessel heading,  $r$  is the corresponding angular velocity. The vessel velocities in surge and sway are denoted by  $u$  and  $v$ . The movement in surge and sway direction is given by  $x^b$  and  $y^b$ . The thruster and wind forces/moments are given by  $u_i$  and  $v_i$  ( $i = 1$ : surge,  $i = 2$ : sway, and  $i = 3$ : yaw). The variables  $u_c$ ,  $v_c$  and  $n_c$  represents the current velocity in surge, current velocity in sway and current moment in yaw, respectively. Gaussian noise is represented by  $\eta_i$ . The parameter values used for the Balchen model are available in the Appendix.

To obtain the position in a NED frame it is necessary to add two equations relating the north-east position to the BODY velocity in surge and sway. The equations for this

conversion follow directly from the rotation matrix. The kinematic equations are given in (2.60) and (2.61). These equations are added to the other Balchen equations when simulating vessel dynamics.

$$\dot{x}^n = \cos(\psi)u - \sin(\psi)v \quad (2.60)$$

$$\dot{y}^n = \sin(\psi)u + \cos(\psi)v \quad (2.61)$$

## 2.7.4 Simulating vessel models

The vessel models proposed here have been simulated using the Runge-Kutta 4th order method (see [25] for more details). This is a common numerical method; Given some function  $\frac{dx}{dt} = f(t, x)$  and starting value  $x_0$ , then the values of  $x$  can be obtained iteratively using equations (2.62), (2.63), (2.64), (2.65) and (2.66).

$$c_1 = \Delta t f(t_k, x_k) \quad (2.62)$$

$$c_2 = \Delta t f\left(t_k + \frac{\Delta t}{2}, x_k + \frac{c_1}{2}\right) \quad (2.63)$$

$$c_3 = \Delta t f\left(t_k + \frac{\Delta t}{2}, x_k + \frac{c_2}{2}\right) \quad (2.64)$$

$$c_4 = \Delta t f(t_k + \Delta t, x_k + c_3) \quad (2.65)$$

$$x_{k+1} = x_k + \frac{c_1}{6} + \frac{c_2}{3} + \frac{c_3}{3} + \frac{c_4}{6} \quad (2.66)$$

The error in calculating a new discrete value of  $x(t_k)$  is related to the size of the timestep,  $\Delta t$ . This integration is applied to simulate the different vessel models and in the optimization function in nonlinear model predictive control.

## 2.7.5 Discretizing continuous state space models and precalculate matrices

To make the model linear, it is assumed that  $R(t_k)$  is known at the current timestep. However, this implies that  $A(t_k)$  depends on time and, therefore, must be recalculated every time step.

The original supply model discussed here is continuous. For control purposes, this model must first be turned into a discrete state-space model for use in an optimal linear quadratic controller or model predictive controller.

Generally, if the continuous linear state space model takes the form  $\dot{x}(t) = A_c x(t) + B_c u(t)$  and  $y(t) = C_c x(t) + D_c u(t)$ , then the discretized model  $x_{k+1} = A_d x_k + B_d u_k$  and  $y_k = C_d x_k +$

$D_d u_k$  can be obtained (assuming zero-order hold) [22] from equations (2.67), (2.68), (2.69) and (2.70).

$$A_d = e^{A_c \Delta t} \quad (2.67)$$

$$B_d = \int_0^{\Delta t} e^{A_c \tau} B_c d\tau \quad (2.68)$$

$$C_d = C_c \quad (2.69)$$

$$D_d = 0 \quad (2.70)$$

Here  $A_d$ ,  $B_d$ ,  $C_d$  and  $D_d$ <sup>9</sup> denotes the discrete state transition matrix, input matrix, output matrix, and input-to-output matrix. Likewise  $A_c$ ,  $B_c$ ,  $C_c$  and  $D_d$  represent the continuous version of these matrices. Since  $A_c$  is not constant, but a function of vessel heading, then  $A_d$  and  $B_d$  will also depend on the heading angle.

Another approach to solving for discrete matrices is to use the forward Euler approach. This is shown below:

$$\dot{x} = A_c x + B_c u \quad (2.73)$$

$$\frac{1}{\Delta t} (x_{k+1} - x_k) = A_c x_k + B_c u_k \quad (2.74)$$

$$x_{k+1} = x_k + \Delta t A_c x_k + \Delta t B_c u_k \quad (2.75)$$

$$x_{k+1} = \underbrace{(I + \Delta t A_c)}_{A_d} x_k + \underbrace{\Delta t B_c}_{B_d} u_k \quad (2.76)$$

$$x_{k+1} = A_d x_k + B_d u_k \quad (2.77)$$

Thus,  $A_d = I + \Delta t A_c$  and  $B_d = \Delta t B_c$ . The measurement equation remains unchanged ( $C_d = C_c$ ) since the relation between the output and state vector is purely algebraic. This method is much more computationally efficient than the first one presented. Here we only require basic matrix algebra, while the exact method requires integration and calculation of matrix exponentials. In this thesis, the exact discretizations of state-space matrices with zero-order hold have been used (by means of the `c2d()` function in MATLAB). However, it should be mentioned that the forward Euler approach has been tested and found to be a viable approach.

It should be noted that the Kalman gain also depends on the state transition matrix and is therefore dependent on the heading angle ( $\psi$ ).

---

<sup>9</sup>It should be noted that it is assumed here that there is no displacement in time between the input and output. If such displacement were present equations (2.69) and (2.70) must be modified as shown below:

$$C_d = C_c e^{A_c \hat{\Delta t}} \quad (2.71)$$

$$D_d = C_c \int_0^{\hat{\Delta t}} e^{A_c \tau} \tau \quad (2.72)$$

Here  $\hat{\Delta t}$  is the displacement between the input and output signals.

Since discretization is a computationally expensive task, it is possible to pre-calculate the heading-dependent matrices used in the control simulations. These matrices can then easily be retrieved upon request using a lookup-table. This is possible since the sine and cosine functions in  $R(\psi)$  are periodic functions (with a period of  $2\pi$  or  $360^\circ$ ). This table must be defined for some reasonable resolution of angles. This resolution will, of course, be a source of error, which is the downside of applying this method.

The values that need storing are:

- $A_d(\psi)$ : State transition matrix, needed for calculating the apriori estimate in the Kalman filter.
- $B_d(\psi)$ : Input matrix, needed for calculating the apriori estimate in the Kalman filter.
- $G(\psi)$ : LQ optimal control gain.
- $K(\psi)$ : Kalman gain.

The dictionary structure in MATLAB was chosen to store values since the dictionary structure is based on hash table technology. Hash tables are known to be very efficient in retrieving values based on key-value pairs. It can also be implemented as a simple array by keeping track of the relation between the indices and heading angles.

The pre-calculated matrix approach is depicted in Figure 2.8.

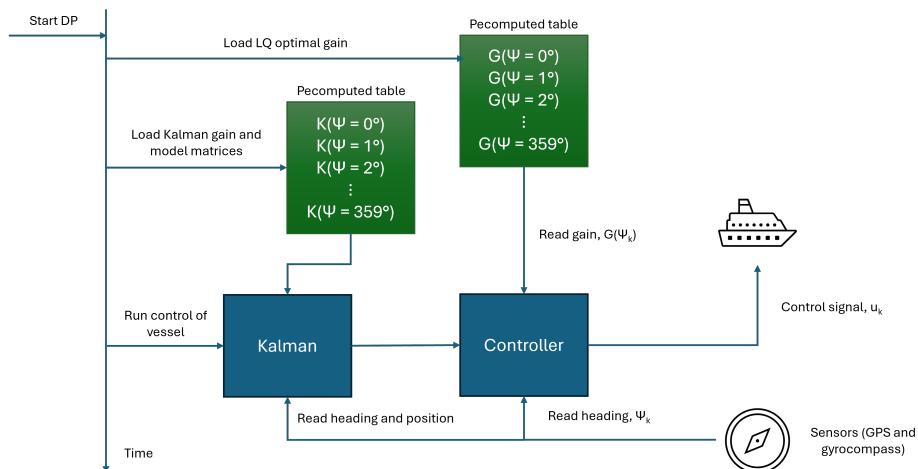


Figure 2.8: Shows the process of pre-calculating heading dependent matrices. Here the Kalman gain,  $K(\psi)$ , and the controller gain,  $G(\psi)$ , are depicted as tables. However, the Kalman filter will also make use of precalculated tables for  $A_d(\psi)$  and  $B_d(\psi)$  in the apriopri-aposteriori formulation, but this is not depicted here for simplicity.

## 2.8 MPC

This section discusses model predictive control in the context of dynamic positioning. Three different algorithms will be described, but the only difference between them is how  $A(t_k)$  and  $B(t_k)$  are estimated in the prediction horizon. Unlike the LQ optimal control simulations (on deviation form), here we must include the  $b$ -term ("current" estimate) to get integral action in the controller.

### 2.8.1 MPC algorithm

A model predictive control algorithm can be obtained by using the DP model by Fossen, see equations (2.33) and (2.34). The states are estimated using the Kalman filter. The MPC is based on minimizing an objective function with respect to a future time horizon obtained through model-based estimation. The objective function,  $J$ , to be minimized is given in equation (2.78).

$$\min_u J = \min_u \left[ \frac{1}{2} \sum_{k=1}^N (e_k^T Q_k e_k + \Delta u_{k-1}^T P_{k-1} \Delta u_{k-1}) \right] \quad (2.78)$$

Here, it will be assumed that  $Q_k$  and  $P_{k-1}$  are constant matrices for the entire prediction horizon, that is,  $Q_1, Q_2, \dots, Q_N = Q$  and  $P_0, P_1, \dots, P_{N-1} = P$ . Here, the length of the prediction horizon is defined by  $N$ , that is, the horizon will span a period of  $N\Delta t$  seconds. The error is calculated as the difference between setpoint and the output value, i.e.  $e_k = r_k - y_k$ . It should be noted that  $\Delta u$  is used in the objective function, not  $u$  directly; the reason for this is that the aim is to reach the reference position, but a bias occurs if there is a competition between minimizing the position error and thruster usage. At some point the setpoint error can become so low that the net thruster forces start to dominate the objective function. This is only a problem if external forces are present. Lastly, the size of the bias will depend on the tuning parameters  $Q$  and  $P$ .

Future values of  $y_k$  are obtained from the state-space model. The model-based controller makes use of the state and measurement equation. These are repeated below for ease of reference.

$$\dot{x} = \begin{bmatrix} 0_{3 \times 3} & R(t) & 0_{3 \times 3} \\ 0_{3 \times 3} & -M^{-1}D & M^{-1}R^T(t) \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} x + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} \tau + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} (\tau_{wind} + \tau_{wave}) \quad (2.79)$$

$$y = [I_{3 \times 3} \ 0_{3 \times 3} \ 0_{3 \times 3}] x \quad (2.80)$$

Here,  $x = [\eta^T, v^T, b^T]^T$ . The goal is to minimize equation (2.78) while also complying with the mathematical model defined in equations (2.79) and (2.80).

The governing principle of a model predictive controller is to solve successive optimization problems to obtain the optimal input to the process being controlled. This means finding the optimal input  $u^*$  for some future horizon (consisting of  $N$  timesteps into the future). The initial value of these control inputs  $u_0^*$  at time  $t_k$  is then applied to the process before calculating the next optimal control input for the next prediction horizon starting at time  $t_{k+1}$ . These optimizations are optimal in the sense that both the optimization criterium (see equation (2.78)) is minimized and the mathematical model (see equations (2.79) and (2.80) are satisfied). A simplified example is shown in Figure 2.9 for a system with one input and one output. Using the previous optimization values as an initial guess in the

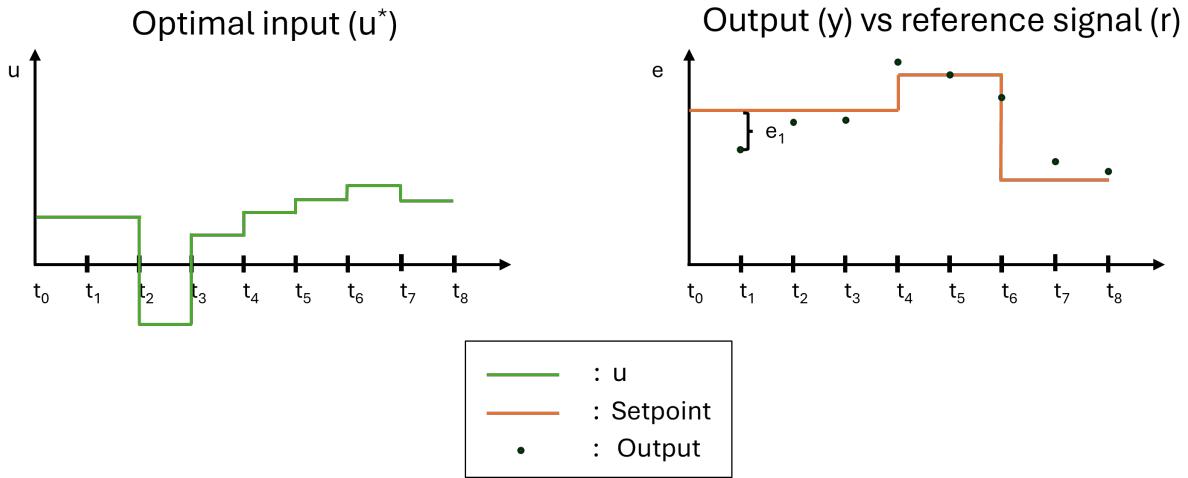


Figure 2.9: An optimal set of future inputs to the system are calculated in model predictive control. These are optimal with respect to the objective function and obeys the underlying mathematical model. The horizon in this example is 8 samples.

new optimization routine is known as warm starting and can reduce the time it takes for the solver to find a new optimal solution.

To solve the optimization problem, the easiest approach is to rewrite the equations in standard form, since this is the form used by MATLAB's `quadprog()` optimization function. The standard form (for quadratic programming) is shown in equation (2.81).

$$\min_z J = \min_z \left[ \frac{1}{2} z^T H z + c^T z \right] \quad (2.81)$$

Subject to:

$$A_{\text{eq}} z = b_{\text{eq}} \quad (2.82)$$

$$A_{\text{in}} z \leq b_{\text{in}} \quad (2.83)$$

$$z_L \leq z \leq z_U \quad (2.84)$$

The original problem formulation can be reorganized to fit in the standard form by clever choice of variables in the vector  $z$ . The following derivation mirrors the lecture notes by

Sharma [26] with a slight modification (adding the  $\Delta u$  term to the vector  $z$ ). It is chosen to contain the original state variables ( $x_k$ ), the errors ( $e_k$ ), references ( $r_k$ ), inputs ( $u_k$ ), and input differences ( $\Delta u$ ). Here, it will be assumed that  $z$  takes the form described below.

$$z = [u^T, x^T, e^T, y^T, \Delta u^T]^T \quad (2.85)$$

$$z = [u_0^T, u_1^T, \dots, u_{N-1}^T, x_1^T, x_2^T, \dots, x_N^T, e_1^T, e_2^T, \dots, e_N^T, \\ y_1^T, y_2^T, \dots, y_N^T, \Delta u_0^T, \Delta u_1^T, \dots, \Delta u_{N-1}^T]^T \quad (2.86)$$

The vectors  $u_i$ ,  $x_i$ ,  $e_i$ ,  $y_i$  and  $\Delta u_i$  are column vectors that contain the relevant values at step  $i$  in the prediction horizon. An example is  $u_i = [u_1(t_i), u_2(t_i), u_3(t_i)]^T$ , representing the force in surge, sway, and moment in yaw at time  $t_i$ .

The standard form equation (see (2.81)) can be rewritten to the following form:

$$J = \frac{1}{2} \begin{bmatrix} u \\ x \\ e \\ y \\ \Delta u \end{bmatrix}^T \begin{bmatrix} H_{11} & 0 & 0 & 0 & 0 \\ 0 & H_{22} & 0 & 0 & 0 \\ 0 & 0 & H_{33} & 0 & 0 \\ 0 & 0 & 0 & H_{44} & 0 \\ 0 & 0 & 0 & 0 & H_{55} \end{bmatrix} \begin{bmatrix} u \\ x \\ e \\ y \\ \Delta u \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}^T \begin{bmatrix} u \\ x \\ e \\ y \\ \Delta u \end{bmatrix} \quad (2.87)$$

The original objective function (2.78) can be expanded as shown in equation (2.88).

$$J = \frac{1}{2} (e_1^T Q e_1 + e_2^T Q e_2 + \dots + e_N^T Q e_N + \Delta u_0^T P \Delta u_0 + \Delta u_1^T P \Delta u_1 + \dots + \Delta_{N-1}^T P \Delta u_{N-1}) \quad (2.88)$$

Comparing equations (2.87) and (2.88) it becomes clear that  $H_{11} = 0_{Nr \times Nr}$ ,  $H_{22} = 0_{Nn \times Nn}$  and  $H_{44} = 0_{Nm \times Nm}$ . Here  $n$  is the number of state variables,  $r$  is the number of inputs and  $m$  is the number of outputs. The block entries of  $H_{33}$  and  $H_{55}$  are given in equation (2.89).

$$H_{33} = \begin{bmatrix} P & 0 & \dots & 0 \\ 0 & P & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P \end{bmatrix} \quad H_{55} = \begin{bmatrix} Q & 0 & \dots & 0 \\ 0 & Q & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & Q \end{bmatrix} \quad (2.89)$$

Since all the terms in equation (2.88) are quadratic terms with respect to  $e_k$  and  $u_k$  it follows that all the coefficients of the linear term in equation (2.87) are zero vectors, i.e.  $c_i = 0 \quad \forall i \in \{1, 2, 3, 4, 5\}$ .

The next step is to include the state space model into the equality constraints in (2.82).

These equality constraints can be rewritten as:

$$\underbrace{\begin{bmatrix} A_{e1u} & A_{e1x} & A_{e1e} & A_{e1y} & A_{e1\Delta u} \\ A_{e2u} & A_{e2x} & A_{e2e} & A_{e2y} & A_{e2\Delta u} \\ A_{e3u} & A_{e3x} & A_{e3e} & A_{e3y} & A_{e3\Delta u} \\ A_{e4u} & A_{e4x} & A_{e4e} & A_{e4y} & A_{e4\Delta u} \end{bmatrix}}_{A_{eq}} \begin{bmatrix} u \\ x \\ e \\ y \\ \Delta u \end{bmatrix} = \underbrace{\begin{bmatrix} b_{e1} \\ b_{e2} \\ b_{e3} \\ b_{e4} \\ b_{eq} \end{bmatrix}}_{b_{eq}} \quad (2.90)$$

**First block row.** This row contains the calculations related to the state equation for the entire prediction horizon.

$$x_1 - B(t_0)u_0 = A(t_0)x_0 + B(t_0)(\tau_{wind} + \tau_{wave}) \quad (2.91)$$

$$x_2 - A(t_1)x_1 - B(t_1)u_1 = B(t_1)(\tau_{wind} + \tau_{wave}) \quad (2.92)$$

$$x_3 - A(t_2)x_2 - B(t_2)u_2 = B(t_2)(\tau_{wind} + \tau_{wave}) \quad (2.93)$$

⋮

$$x_N - A(t_{N-1})x_{N-1} - B(t_{N-1})u_{N-1} = B(t_{N-1})(\tau_{wind} + \tau_{wave}) \quad (2.94)$$

It should be stressed that  $A(t_k)$  and  $B(t_k)$  is only known for  $t_0$  (current time when starting to calculate the future horizon). We need some strategy to estimate  $A(t_k)$  and  $B(t_k)$   $\forall k \in \{1, 2, \dots, N-1\}$ . This will be discussed later. Furthermore, the future of external disturbances, in this case wind and wave forces, is not known. The solution to this is to use the current disturbance values for the entire prediction horizon. Although these are subject to change, unless the horizon spans a long time interval, this should be a reasonable approximation. Recall that a slowly changing disturbance assumption was used in the derivation of the LQ optimal controller on deviation form.

From this it follows that:

$$A_{e1u} = \begin{bmatrix} -B(t_0) & 0 & \cdots & 0 \\ 0 & -B(t_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -B(t_{N-1}) \end{bmatrix} \quad A_{e1x} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -A(t_1) & I & 0 & \cdots & 0 \\ 0 & -A(t_2) & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}$$

$$A_{e1e} = \mathbf{0}_{Nn \times Nm}$$

$$A_{e1y} = \mathbf{0}_{Nn \times Nm}$$

$$A_{e1\Delta u} = \mathbf{0}_{Nn \times Nr} \quad b_{e1} = \begin{bmatrix} A(t_0)x_0 + B(t_0)(\tau_{wind} + \tau_{wave}) \\ B(t_1)(\tau_{wind} + \tau_{wave}) \\ B(t_2)(\tau_{wind} + \tau_{wave}) \\ \vdots \\ B(t_{N-1})(\tau_{wind} + \tau_{wave}) \end{bmatrix}$$

**Second block row.** This block row is related to the measurement equation.

$$y_1 - Cx_1 = 0 \quad (2.95)$$

$$y_2 - Cx_2 = 0 \quad (2.96)$$

⋮

$$y_N - Cx_N = 0 \quad (2.97)$$

Thus, the matrices in the second block row become:

$$A_{e2u} = \mathbf{0}_{Nm \times Nr} \quad A_{e2x} = \begin{bmatrix} -C & 0 & \cdots & 0 \\ 0 & -C & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -C \end{bmatrix}$$

$$A_{e2e} = \mathbf{0}_{Nm \times Nm} \quad A_{e2y} = \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix}$$

$$A_{e2\Delta u} = \mathbf{0}_{Nm \times Nr} \quad b_{e2} = \mathbf{0}_{Nm \times 1}$$

**Third block row.** In this block row, the equality constraint is responsible for ensuring that  $e_k = r_k - y_k$ .

$$e_1 + y_1 = r_1 \quad (2.98)$$

$$e_2 + y_2 = r_2 \quad (2.99)$$

⋮

$$e_N + y_N = r_N \quad (2.100)$$

The resulting block entries then become:

$$A_{e3u} = \mathbf{0}_{Nm \times Nr}$$

$$A_{e3x} = \mathbf{0}_{Nm \times Nn}$$

$$A_{e3e} = \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix}$$

$$A_{e3y} = \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix}$$

$$A_{e3\Delta u} = \mathbf{0}_{Nm \times Nr}$$

$$b_{e3} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_N \end{bmatrix}$$

Here  $r_k$  is a column vector of setpoints valid at time  $t_k$ .

**Fourth block row.** This last block row represents equality constraints associated with  $\Delta u$ :

$$u_0 - \Delta u_0 = u_{-1} \quad (2.101)$$

$$u_1 - \Delta u_1 - u_0 = 0 \quad (2.102)$$

$$u_2 - \Delta u_2 - u_1 = 0 \quad (2.103)$$

$$\vdots$$

$$u_{N-1} - \Delta u_{N-1} - u_{N-2} = 0 \quad (2.104)$$

The resulting blocks then become:

$$A_{e4u} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -I & I & 0 & \cdots & 0 \\ 0 & -I & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix} \quad A_{e4x} = \mathbf{0}_{Nr \times Nn}$$

$$A_{e4e} = \mathbf{0}_{Nr \times Nm}$$

$$A_{e4y} = \mathbf{0}_{Nr \times Nm}$$

$$b_{e4\Delta u} = \begin{bmatrix} -I & 0 & \cdots & 0 \\ 0 & -I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -I \end{bmatrix} \quad b_{e4} = \begin{bmatrix} u_{-1} \\ \mathbf{0}_{(N-1)r \times 1} \end{bmatrix}$$

All these matrices can be summarized in a single matrix using the kronecker product<sup>10</sup> as shown in equation (2.106). Two sub-matrices are marked in red because they will take a different form when  $A$  and  $B$  are allowed to vary over time<sup>11</sup>.

$$A_{\text{eq}} = \begin{bmatrix} -I_N \otimes B & I_{Nn} - (I_{N,-1} \otimes A) & 0_{Nn \times Nm} & 0_{Nn \times Nm} & 0_{Nn \times Nr} \\ 0_{Nm \times Nr} & -I_N \otimes C & 0_{Nm \times Nm} & I_{Nm} & 0_{Nm \times Nr} \\ 0_{Nm \times Nr} & 0_{Nm \times Nn} & I_{Nm} & I_{Nm} & 0_{Nm \times Nr} \\ I_{Nr} - (I_{N,-1} \otimes I_r) & 0_{Nr \times Nn} & 0_{Nr \times Nm} & 0_{Nr \times Nm} & -I_{Nr} \end{bmatrix} \quad (2.106)$$

The  $b_{\text{eq}}$  term takes the form in (2.107).

$$b_{\text{eq}} = \begin{bmatrix} A(t_0)x_0 + B(t_0)(\tau_{\text{wind}} + \tau_{\text{wave}}) \\ B(t_1)(\tau_{\text{wind}} + \tau_{\text{wave}}) \\ B(t_2)(\tau_{\text{wind}} + \tau_{\text{wave}}) \\ \vdots \\ B(t_{N-1})(\tau_{\text{wind}} + \tau_{\text{wave}}) \\ \mathbf{0}_{Nm \times 1} \\ r_1 \\ r_2 \\ \vdots \\ r_N \\ u_{-1} \\ \mathbf{0}_{(N-1)r \times 1} \end{bmatrix} \quad (2.107)$$

The term  $I_{N,-1}$  is an identity matrix that has been modified by pushing the diagonal entries down one step to the adjacent row.

---

<sup>10</sup>Given two matrices  $A$  and  $B$ , the kronecker product,  $A \otimes B$ , can be written as in equation (2.105). If  $A$  is an  $n \times m$  matrix and  $B$  is an  $r \times s$  matrix then the kronecker product  $A \otimes B$  will be a  $nr \times ms$  matrix.

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{bmatrix} \quad (2.105)$$

<sup>11</sup>When  $A$  and  $B$  vary over time, then the red marked matrices in (2.106) takes the following form:

$$A_{e1u} = \begin{bmatrix} -B(t_0) & 0 & \cdots & 0 \\ 0 & -B(t_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -B(t_{N-1}) \end{bmatrix} \quad A_{e1x} = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ -A(t_1) & I & 0 & \cdots & 0 \\ 0 & -A(t_2) & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I \end{bmatrix}$$

These matrices were found earlier when considering the first block row in the equality constraint matrix,  $A_{\text{eq}}$ .

$$I_{N,-1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad (2.108)$$

The inequality constraint matrices in equation (2.83) and (2.84) are all set to zero matrices. However, these could be used to set limitations on the maximum allowed thruster forces and moment.

## 2.8.2 MPC strategies

Three strategies will be considered when estimating  $A(t_k)$  and  $B(t_k)$  in the future prediction horizon.

- **Constant heading:** Assume that the initial transition matrix remains unchanged for the entire prediction horizon, that is, if  $t_0$  is the current time step, then  $A(t_k) = A(t_0)$  and  $B(t_k) = B(t_0) \forall k \in \{1, 2, \dots, N-1\}$ . Another way to phrase this assumption is that the heading angle remains constant for the entire prediction horizon.
- **Constant rotation:** Since the heading rate is known, it can be assumed that the angular velocity remains constant for the entire prediction horizon. The estimated vessel heading used in the rotation matrix  $R(\psi)$  can then be calculated from the heading estimate  $\hat{\psi}(t_k)$  in equation (2.109).

$$\hat{\psi}(t_k) = \dot{\psi}(t_0)(t_k - t_0) + \psi(t_0) \quad (2.109)$$

Thus,

$$R(t_k) = R(\hat{\psi}(t_k)) \quad (2.110)$$

Knowing the rotation matrix,  $R(t_k)$  it is straightforward to calculate  $A(t_k)$  and  $B(t_k)$ . This method is an improvement on the constant heading approach discussed earlier.

- **Linear Parameter Varying (LPV) approach:** We assume the current time is  $t_0$ . To calculate future values of the state transition matrix and input matrix ( $A(t_k)$  and  $B(t_k)$  for  $k > 0$ ), the previously calculated sets of control inputs  $u^*$  (from the optimization) together with the estimated present state  $\hat{x}_0$  can be used to propagate the state vector into the future. Knowing that the vessel heading is part of the state vector, it is then trivial to calculate the values of  $A$  and  $B$  in the prediction horizon.

The algorithm can be described as follows:

1. Set the  $\tilde{x}_0$  equal to the current estimate of the state vector from the Kalman filter.

2. Shift the previous input vector. If the previous timestep used  $u^*(t_{-1}) = [u_0^*, u_1^*, \dots, u_{N-1}^*]$  then we adapt this array to the current time step  $t_0$ , that is,  $\tilde{u}^*(t_0) = [u_1^*, u_2^*, \dots, u_{N-1}^*, u_{N-1}^*]$ .
3. Predict the next state from the process model using the first entry in  $\tilde{u}^*(t_0)$ . That is,  $\tilde{x}_1 = A(t_0)\tilde{x}_0 + B(t_0)u_1^* + B(t_0)(\tau_{\text{wind}} + \tau_{\text{wave}})$ . Since  $\tilde{x}_1$  now contains the heading,  $\tilde{\psi}_1$ , it is possible to calculate  $A(t_1)$  and  $B(t_1)$ . We can now obtain  $\tilde{x}_2$  from the process model:  $\tilde{x}_2 = A(t_1)\tilde{x}_1 + B(t_1)u_2^* + B(t_1)(\tau_{\text{wind}} + \tau_{\text{wave}})$ . This can be repeated for the entire prediction horizon. Wind and wave forces are assumed to be constant for the entire prediction horizon, and equal to the known values at time  $t_0$ .
4. Since the  $A$  and  $B$  matrices are now known for the entire prediction horizon, the MPC can be solved following the steps in Section 2.8.1.
5. Go back to step 1 and repeat the procedure for the next timestep.

This method was inspired by an online control theory course in MPC [27] based on a master's thesis [28]. This method is fairly universal and can be applied to non-linear systems. The course used the method to control the movement of a car in a horizontal plane (analogously to a 3-DoF vessel model). But the method can be extended to deal with drones and other aircraft operating in 6-DoF.

Not surprisingly, of all the three methods outlined above, the LPV approach is the most time consuming considering it requires the largest number of calculation steps.

## 2.9 NMPC

This chapter discusses how to implement a non-linear model predictive control strategy to perform dynamic positioning. This is a first step toward developing a green DP.

### 2.9.1 NMPC algorithm

This approach uses the DP model discussed in Section 2.4. The DP model is repeated here for easy reference (see equations (2.111) and (2.112)).

$$\dot{x} = \begin{bmatrix} 0_{3 \times 3} & R(t) & 0_{3 \times 3} \\ 0_{3 \times 3} & -M^{-1}D & M^{-1}R^T(t) \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} x + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} \tau + \begin{bmatrix} 0_{3 \times 3} \\ M^{-1} \\ 0_{3 \times 3} \end{bmatrix} (\tau_{\text{wind}} + \tau_{\text{wave}}) \quad (2.111)$$

$$y = [I_{3 \times 3} \ 0_{3 \times 3} \ 0_{3 \times 3}] x + \varepsilon \quad (2.112)$$

Just like in the previous example of model predictive control, the cost function chosen here will take the same quadratic form. It is repeated here for convenience in equation (2.113).

$$\min_u J = \min_u \left[ \frac{1}{2} \sum_{k=1}^N (e_k^T Q e_k + \Delta u_{k-1}^T P \Delta u_{k-1}) \right] \quad (2.113)$$

A nonlinear optimizer, `fmincon()` in MATLAB, will be used to calculate the best choice of  $\Delta u$  that minimizes the cost function above. For any choice of  $\Delta u$  the cost function is calculated by integrating the DP model from the current time to the end of the prediction horizon ( $N$  steps).

This is a very costly procedure with respect to computing time. The previous linear formulations are much more efficient from a computational perspective. The `fmincon()` function will make successive calls to the integrating function - here using Runge-Kutta 4'th order - for different values of inputs.

## 2.9.2 Grouping strategy for inputs

Since NMPC is a computationally intensive method, the computational burden can be lessened by applying some restrictions on the control input. The obvious approach is to limit the horizon length, but this can also degrade performance, and small horizons leads to instability. Another approach is to apply grouping. This is a strategy that divides the input into sets of adjacent inputs that are forced to use the same value. Say, the horizon consists of 12 samples, that is,  $\{u_0, u_1, u_2, \dots, u_{11}\}$ , then it can be divided into three groups of size 2, 4, and 6. Then each set of inputs is forced to use the same input value ( $\{u_0, u_1\}$ ,  $\{u_2, u_3, u_4, u_5\}$  and  $\{u_6, u_7, u_8, u_9, u_{10}, u_{11}\}$ ). This restriction can be imposed by applying the following equality constraints:

$$\begin{array}{lll} u_0 = u_1 & u_2 = u_3 & u_6 = u_7 \\ u_3 = u_4 & u_7 = u_8 & u_8 = u_9 \\ u_4 = u_5 & & u_9 = u_{10} \\ & & u_{10} = u_{11} \end{array}$$

It might appear counterintuitive that applying more restrictions on the control problem would lessen the computational load; however, there is a trade-off between the number of optimization variables and the number of constraints. In this case, the number of effective optimization variables decreases, causing the time required to find a solution to decrease. Adding optimization variables will typically require more computing power than adding constraints.

## 2.10 Applying system identification to identify ship dynamics

This chapter will discuss the application of system identification to identify vessel models for control purposes. Three linear time-invariant state-space models will be identified from the following vessel simulations:

- Supply model
- OSV model
- Balchen model

### 2.10.1 Data generation and system identification

The ship process model will be subjected to an input signal (net force and moment from thrusters, or, in the case of the OSV model, the RPS signal to the thrusters). The vessel velocities in surge ( $u$ ) and sway ( $v$ ) together with the rotational velocity ( $r$ ) are stored, representing the outputs of interest ( $y$ ). Knowing both the output  $y$  and the input  $u$ , it is possible to identify a linear state-space model that relates these quantities using system identification methods like DSR<sup>12</sup> and PEM<sup>13</sup>.

The input signal applied here is a pseudo random binary signal. A MATLAB function named `prbs1()` is used for this purpose, developed by Dr. David Di Ruscio. The PRBS signal is generated using three parameters:

- $N$ : The number of samples generated.
- $T_{\min}$ : Minimal interval size where the signal remains constant.
- $T_{\max}$ : Maximum interval size where the signal remains constant.

The signal is generated by calculating consecutive intervals with the signal altering between  $+1$  and  $-1$ . Each interval length,  $t_i$  (in this context referring to the number of samples), is determined from the discrete uniform distribution shown in Equation (2.114)

$$P(T = t) = \begin{cases} \frac{1}{T_{\max}-T_{\min}+1} & \text{for } T_{\min} \leq t \leq T_{\max} \\ 0 & \text{for } t < T_{\min} \text{ or } t > T_{\max} \end{cases} \quad (2.114)$$

Here,  $t$  belongs to the set of natural numbers ( $t \in \mathbb{N}$ ). In other words, every integer between  $T_{\min}$  and  $T_{\max}$  is equally likely to be chosen as the next interval length when generating the signal.

An example output from running PRBS is depicted in Figure 2.10.

---

<sup>12</sup>DSR: Combined Deterministic and Stochastic System Identification and Realization.

<sup>13</sup>PEM: Prediction Error Method.

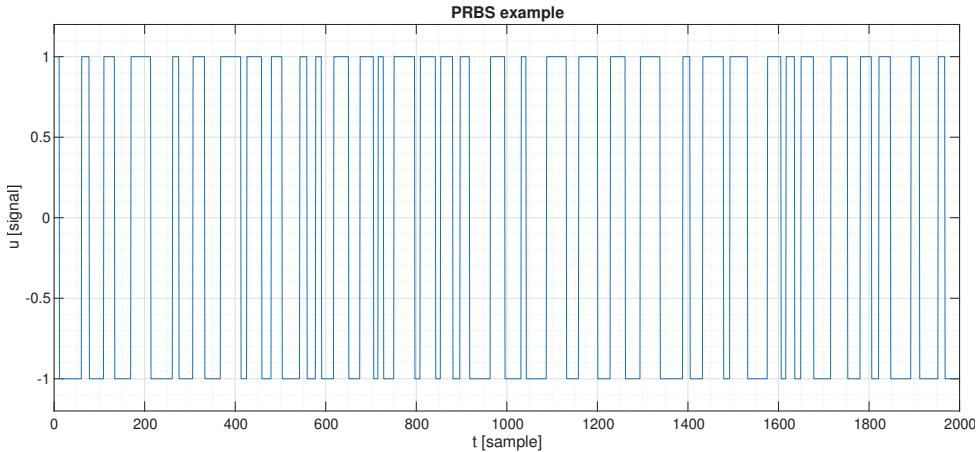


Figure 2.10: Depicts a pseudo random binary signal using the following parameters:  $N = 2000$ ,  $T_{min} = 10$  and  $T_{max} = 50$ .

The PRBS signal is applied to the inputs of the vessel models. The signal is engineered to excite the system in such a way that the information collected ( $y, u$ ) is sufficiently rich to capture the dynamics of the system. This is necessary for performing successful system identification.

The system identification methods discussed here identifies a linear time-invariant state space model on the form shown in equations (2.115) and (2.116).

$$x_{k+1} = Ax_k + Bu_k \quad (2.115)$$

$$y_k = Cx_k + Du_k \quad (2.116)$$

Here  $A$  is the state transition matrix,  $B$  is the input matrix,  $C$  is the output matrix, and  $D$  is the direct input to output matrix. It will be assumed that  $D$  is zero (this is equivalent to setting the structure parameter  $g$  to zero in the context of DSR). The entire identification process is depicted in Figure 2.11.

Since a pseudo random binary signal is used to generate the inputs to the vessel simulations, the simulations are open-loop.

### 2.10.2 Combined Deterministic and Stochastic System Identification and Realization (DSR)

The DSR method identifies a linear state space model based on a collection of inputs ( $u$ ) and outputs ( $y$ ). This method is a subspace method based on a series of matrix manipulations. No optimization routine is needed to identify the model parameters. This makes it a vastly more computationally efficient method than the classical prediction error

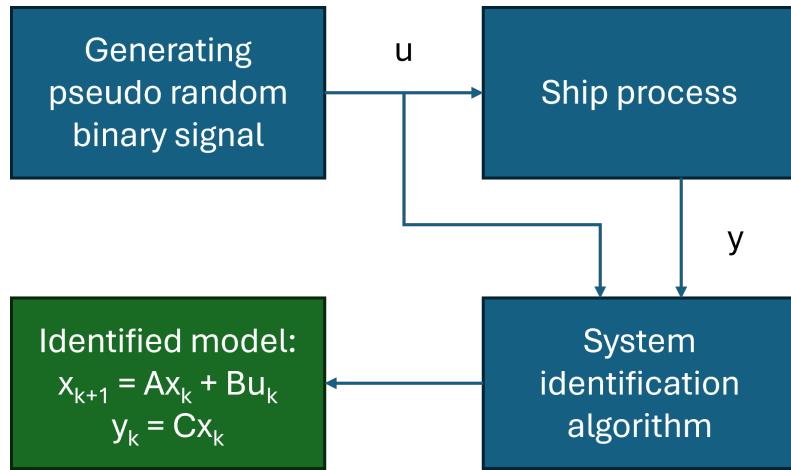


Figure 2.11: Depicts the process of system identification.

methods that rely on optimization. The DSR algorithm was invented and implemented by Dr. David Di Ruscio [29] [30]. There is also an extended version of the same algorithm named DSR\_e that works well on data generated from both open-loop and closed-loop simulations.

### 2.10.3 Prediction Error Method (PEM)

The prediction error method used here is the inbuilt `pem()` function in MATLAB (part of the system identification toolbox). This method is not a pure prediction error method, in the sense that it first finds an initial guess using the N4SID method, which is another subspace method (like DSR and DSR\_e).

### 2.10.4 Novel approach to identifying model for control

The approach outlined here is to determine the BODY dynamics of the vessel. This is inspired by the DP model defined in equations (2.33) and (2.34). The kinematic relationship between BODY velocities and position is known. The goal is to identify a state-space model where the outputs are velocities in surge, sway, and yaw. The inputs are the net thruster forces/moment. We want to identify the matrices  $A$ ,  $B$  and  $C$  in the state-space model defined in equations (2.117) and (2.118).

$$x_{k+1} = Ax_k + Bu_k \quad (2.117)$$

$$y_k = Cx_k \quad (2.118)$$

The output will be denoted  $y_k = v_k = [u_k, v_k, r]$ .

As before, we can assume that  $R(t)$  is known at the current time. The kinematic equation that relates the BODY velocities to NED position is well known from before;  $\dot{\eta} = R(t)v$ . From this we are left with three matrix equations: The kinematic equation (2.119), the process model model (2.120) and the output model (2.121).

$$\dot{\eta} = R(t)v \quad (2.119)$$

$$x_{k+1} = Ax_k + Bu_k \quad (2.120)$$

$$y_k = v_k = Cx_k \quad (2.121)$$

First we need to discretize the kinematic equation (2.119) using the forward Euler method. This results in equation (2.122).

$$\eta_{k+1} = \Delta t R(t_k) v_k + \eta_k \quad (2.122)$$

Here,  $\Delta t$  is the time step in the simulation. Equation (2.121) is modified as shown in (2.123).

$$v_{k+1} = Cx_{k+1} = C(Ax_k + Bu_k) = CAx_k + CBu_k \quad (2.123)$$

This allows for the process and output equations to be defined:

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ x_{k+1} \\ v_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & 0_{3 \times 3} & \Delta t R(t_k) \\ 0_{3 \times 3} & A & 0_{3 \times 3} \\ 0_{3 \times 3} & CA & 0_{3 \times 3} \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \end{bmatrix}}_{\tilde{B}} u_k \quad (2.124)$$

$$y_k = \eta_k = \underbrace{\begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}}_{\tilde{C}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \end{bmatrix}}_{\tilde{x}} = \tilde{C} \tilde{x} \quad (2.125)$$

The matrix  $\tilde{A}$  can be recalculated at each time step based on the known value (due to the precise gyrocompass) of the heading angle, and thus the rotation matrix is known ( $R(t_k)$ ). This formulation can be used directly in the LQ optimal controller on deviation form, since this model structure includes integral action. However, for use in the finite-horizon model predictive control strategies (MPC and NMPC), it is necessary to include integral action in the state-space model to estimate disturbances and compensate for modeling errors.

It is straightforward to include external forces due to weather conditions. The value of  $u_k$  can be considered to include the net force on the ship, since the net force acting on the vessel, whether it comes from external disturbances or thrusters, will affect the vessel in the same way. Thus,  $u_k = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}$ , where  $\tau$  represents the thruster forces. Then equation (2.124) can be rewritten in the form given in equation (2.126).

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ x_{k+1} \\ v_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & 0_{3 \times 3} & \Delta t R(t_k) \\ 0_{3 \times 3} & A & 0_{3 \times 3} \\ 0_{3 \times 3} & CA & 0_{3 \times 3} \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \end{bmatrix}}_{\tilde{B}} \tau + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \end{bmatrix}}_{F} (\tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}) \quad (2.126)$$

If it is assumed that current is a phenomenon occurring in the NED frame (as in the original DP model by Fossen), then the current forces need to be converted to the BODY frame, i.e.  $\tau_{\text{curr}} = R^T(t_k)b_k$ , where  $b_k$  represents the current force in the NED frame. The second and third block column can then be written as in (2.128) and (2.130).

$$x_{k+1} = Ax_k + B\tau + B(\tau_{\text{wind}} + \tau_{\text{wave}} + R^T(t_k)b_k) \quad (2.127)$$

$$= Ax_k + B\tau + B(\tau_{\text{wind}} + \tau_{\text{wave}}) + BR^T(t_k)b_k \quad (2.128)$$

$$v_{k+1} = CAx_k + CB\tau + CB(\tau_{\text{wind}} + \tau_{\text{wave}} + R^T(t_k)b_k) \quad (2.129)$$

$$= CAx_k + CB\tau + CB(\tau_{\text{wind}} + \tau_{\text{wave}}) + CBR^T(t_k)b_k \quad (2.130)$$

If we treat  $b_k$  as an integral term (since the current is unknown), then equation (2.126) can be modified to obtain equations (2.131) and (2.132). The dynamics of  $b$  is defined as  $\dot{b} = 0$ . To obtain the discrete version, simple forward Euler can be applied, hence:  $\frac{b_{k+1} - b_k}{\Delta t} = 0$ . This results in  $b_{k+1} = b_k$ . The change in this parameter is due to noise, and hence it is necessary to adjust a relatively high noise variance to these terms to obtain good estimates within a reasonable time.

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ x_{k+1} \\ v_{k+1} \\ b_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & 0_{3 \times 3} & \Delta t R(t_k) & 0_{3 \times 3} \\ 0_{3 \times 3} & A & 0_{3 \times 3} & BR^T(t_k) \\ 0_{3 \times 3} & CA & 0_{3 \times 3} & CBR^T(t_k) \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \\ b_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \\ 0_{3 \times 3} \end{bmatrix}}_{\tilde{B}} \tau + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \\ 0_{3 \times 3} \end{bmatrix}}_{\tilde{F}} (\tau_{\text{wind}} + \tau_{\text{wave}}) \quad (2.131)$$

$$y_k = \eta_k = \underbrace{\begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}}_{\tilde{C}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \\ b_k \end{bmatrix}}_{\tilde{x}} = \tilde{C}\tilde{x} \quad (2.132)$$

Here,  $b_k$  will compensate for model error, parameter uncertainty, and current (if unknown). This can also be viewed as diagnostic tool in the sense that unreasonable force values (far beyond what is believed to be due to current) indicate troubles with the model.

If we compare equations (2.34) and (2.33) with the new model: equations (2.131) and (2.132) it is clear that these models share some characteristics, but there is an extra block row in the new model since both the identified process model and the output model must be integrated into the state transition matrix.

### 2.10.5 Identifying supply model

The input to the supply model in **MSS** is thruster force/moment in surge, sway, and yaw. The input data is generated using PRBS. It is assumed that there are no external disturbances present, that is,  $u = \tau_{\text{thr}}$ . If wind, waves, or current were present, then the input  $u$  must be modified to include these forces:  $u = \tau_{\text{thr}} + \tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}$ . The output is the velocities in surge, sway, and yaw:  $v = [u, v, r]^T$ .

### 2.10.6 Identifying OSV model

The thruster configuration of the OSV model was depicted in Figure 2.7.

The vessel has a total of four thrusters, but 6 degrees of freedom considering that the azimuth thrusters can rotate around a vertical axis. The **MSS** toolbox offers a function interface to this model; `osv()` in MATLAB. However, the input here is the rotational speed in RPS (Rotations Per Second) to individual thrusters. This is not the common force and moment input used earlier. The new system identification approach described in Section 2.10.4 can still be used, but the interpretation of the  $b_k$  integrator term has a less obvious interpretation, even though it will still give integral action. In the OSV model, the force is proportional to the square of the rotational speed. Thus, the input fed to the system identification method should be  $u = u_{\text{RPS}}|u_{\text{RPS}}|$ .

The challenge here is that the net force is now related to the individual thruster forces and the azimuth angles ( $\alpha$ ). The problem with azimuth thrusters is that the input matrix,  $B$ , will struggle to describe the dynamics, since the forces and moment acting on the ship due to thrusters are related to the input by the following equation:

$$\tau = T(\alpha)Ku \quad (2.133)$$

Here,  $T$  is a thruster configuration matrix (depends on azimuth angles  $\alpha$ ), and  $K$  is the force coefficient matrix. A more detailed description of these terms can be found in Section 2.13. The matrix  $T$  is nonlinear since it contains sine and cosine terms of the azimuth

angles  $\alpha$ . This known nonlinearity needs to be baked into a linear input matrix,  $B$ , during system identification. This is the very reason why we would normally split the controller into two parts; one being the the controller that calculates the net forces and moment needed to maintain the setpoint, the other being the thruster allocation algorithm that realizes the demand by allocating individual thruster forces and angles.

To simplify matters, the azimuth angles are set to fixed angles, effectively acting as main propellers. For simplicity, only the tunnel thruster closest to the bow is used. In this case, the thruster configuration matrix becomes fixed as shown in equation (2.134).

$$\tau = T(\alpha_0)Ku = \underbrace{TK}_{\text{const}} u \quad (2.134)$$

Here,  $\alpha_0$  refers to fixed azimuth angles. The new vessel will then have the thruster configuration depicted in Figure 2.12.

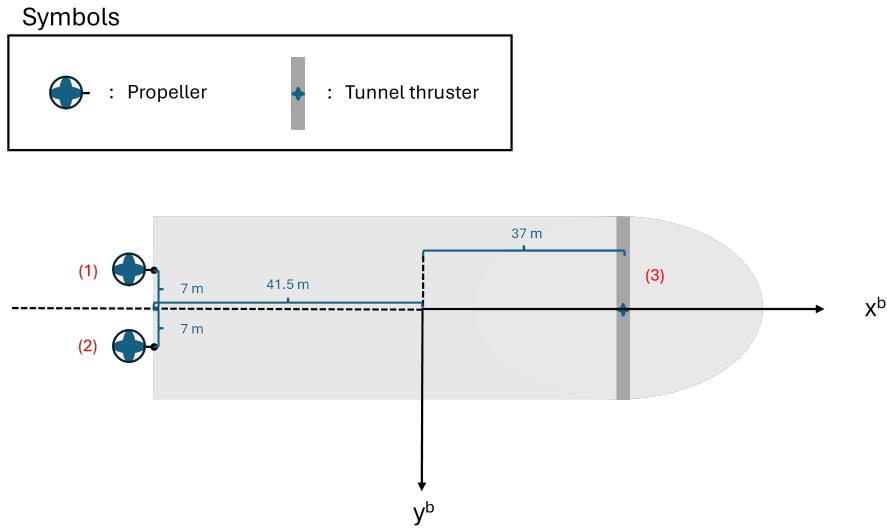


Figure 2.12: Depicts the modified thruster configuration for the OSV model. Here (1) and (2) are azimuth thrusters acting as port and starboard propellers (due to fixed angles), respectively. The third thruster (3) is a bow tunnel thruster.

Earlier, it was noted that the input to the SI algorithm could be defined as  $u = u_{RPS}|u_{RPS}|$  since it is proportional to force. After calculating the optimal input values, the vessel model needs the input in the  $u_{RPS}$  format. This is a straightforward transformation:  $u_{RPS} = \text{sgn}(u)\sqrt{|u|}$ , where  $u$  is the input calculated by the controller. This entire control structure is depicted in Figure 2.13.

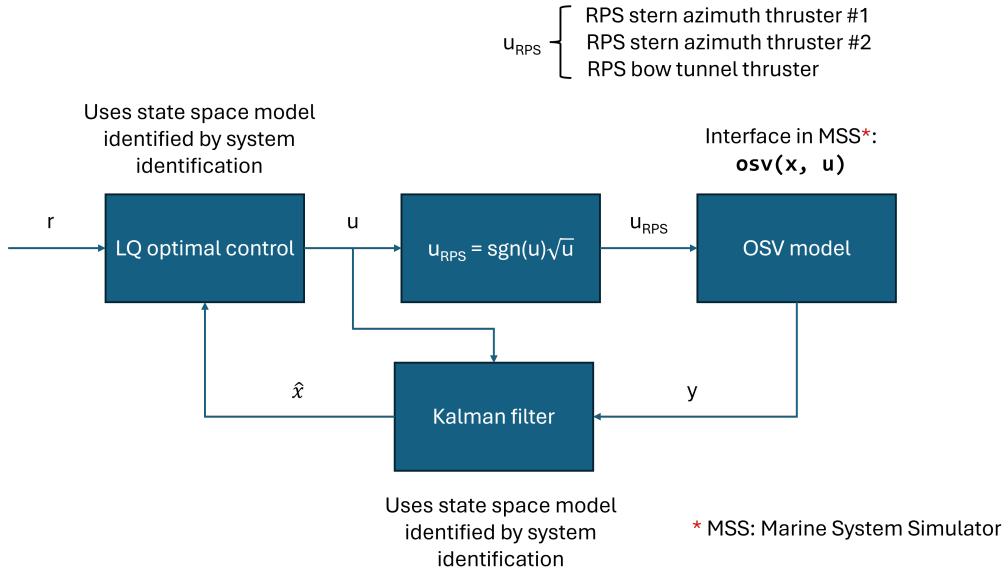


Figure 2.13: Depicts data flow in controlled OSV process. Note the transformation between what the controller demands  $u^*$  in force input and what the vessel function uses  $u_{RPS}$ .

## 2.10.7 Identifying Balchen model

The Balchen model is not part of the MSS software package and was coded from scratch. The current and wind velocity are assumed to be zero during the simulation. Since the model uses the deviation between vessel velocity and current velocity to estimate forces, it implies that there will be some current forces present due to vessel movement. This is a situation where the net forces are not only due to thrusters. This could be corrected, but the vessel movement is fairly limited and it is interesting to see if the controller still manages setpoint tracking.

## 2.10.8 Reducing model to three double integrators

A simple example of a double integrator is a point-mass subject to a force along some x-axis. The governing equations then follows directly from Newtons equations:

$$\dot{x} = v \quad (2.135)$$

$$m\dot{v} = F \quad (2.136)$$

This can clearly be written on state-space form:

$$\begin{bmatrix} \dot{x} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ F \\ m \end{bmatrix} \quad (2.137)$$

This continuous model has two eigenvalues in the state transition matrix, both zeros. This is the property associated with double integrating system. If discretized, the eigenvalues would become equal to 1.

If we were to extend this to cover three dimensions, movement along x-axis, y-axis and z-axis<sup>14</sup>, the number of eigenvalues equal to zero would become 6 in the continuous case and the number of eigenvalues equal to 1 would become 6 in the discrete case. Note that these eigenvalues lie on the border between the stable and unstable regions. This is a system with three double integrators. In the case of the point-mass system the future position of the vessel will be decided on by the net forces alone, without damping. The supply model is not a three double integrator system due to the hydrodynamic damping that is described in the lower right block matrix in the state transition matrix (see equation 2.16). If this damping was removed, the system would be a three integrator system.

Just like in the earlier system identification discussion, we will assume that the model takes the form in equation 2.126. This model is repeated here for convenience:

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ x_{k+1} \\ v_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & 0_{3 \times 3} & \Delta t R(t_k) \\ 0_{3 \times 3} & A & 0_{3 \times 3} \\ 0_{3 \times 3} & CA & 0_{3 \times 3} \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ x_k \\ v_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \end{bmatrix}}_{\tilde{B}} \tau + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \\ CB \end{bmatrix}}_{F} (\tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}) \quad (2.139)$$

Three strategies will be discussed that turn this model into a three double integrator system:

- **Strategy 1:**  $A$  and  $C$  are identity matrices. In this case, the model equation can be simplified as  $v_{k+1} = Av_k + Bu_k$ . The new equation becomes.

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ v_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & \Delta t R(t_k) \\ 0_{3 \times 3} & I \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ v_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{\tilde{B}} \tau + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{F} (\tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}) \quad (2.140)$$

This is clearly a three double integrator system since the eigenvalues are  $(1, 1, 1, 1, 1, 1)$ . This follows from the fact that  $\tilde{A}$  is an upper triangular matrix with ones on the diagonal. In this case,  $B$  alone must be estimated.

---

<sup>14</sup>The three dimensional point-mass system would take the following form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{v}_z \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ v_x \\ v_y \\ v_z \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{F_x}{m} \\ \frac{F_y}{m} \\ \frac{F_z}{m} \end{bmatrix} \quad (2.138)$$

- **Strategy 2:**  $A$  is an upper triangular matrix and  $C$  is the identity matrix. Then the model becomes:

$$\underbrace{\begin{bmatrix} \eta_{k+1} \\ v_{k+1} \end{bmatrix}}_{\tilde{x}_{k+1}} = \underbrace{\begin{bmatrix} I & \Delta t R(t_k) \\ 0_{3 \times 3} & A \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} \eta_k \\ v_k \end{bmatrix}}_{\tilde{x}_k} + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{\tilde{B}} \tau + \underbrace{\begin{bmatrix} 0_{3 \times 3} \\ B \end{bmatrix}}_{F} (\tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{curr}}) \quad (2.141)$$

Where,

$$A = \begin{bmatrix} 1 & a_{12} & a_{13} \\ 0 & 1 & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (2.142)$$

As before the state transition clearly has six eigenvalues all being 1 (due to the upper diagonal nature of the matrix). Here, the entries of the  $B$  matrix and three entries of the  $A$  matrix ( $a_{12}$ ,  $a_{13}$  and  $a_{23}$ ) must be estimated.

- **Strategy 3:** The final version forces  $A$  to be the identity matrix, but the entries of  $B$  and  $C$  must be estimated.

The process of identifying the matrix entries is performed by an optimization function. Every unknown matrix entry is part of the optimization variable fed to `fmincon()` in MATLAB. The goal is to minimize the difference between the output values obtained from the vessel model and the corresponding output estimates from the candidate discrete state-space model (where several matrix entries are included in the optimization variable). The objective function is chosen to be quadratic and is depicted in (2.143).

$$\min_u J = \min_u \left[ \frac{1}{2} \sum_{k=1}^N (e_k^T Q e_k) \right] \quad (2.143)$$

Here  $e_k = y_k - \hat{y}_k$  where  $y_k$  is the output of the vessel model and  $\hat{y}_k$  is the estimate of the state space model.  $Q$  is a weighting matrix. The outputs are the velocities in surge, sway, and yaw. The inputs are net forces and moment acting on the vessel in the BODY plane. Depending on strategy, the number of matrix entries to identify are 9, 12 or 18. This is a prediction error method. It had to be written from scratch as MATLAB has no inbuilt PEM function that supports only partial identification of the state-space matrices.

## 2.11 Tracking

This section contains some simple examples of path-following and trajectory tracking. The methods discussed here apply to relatively small distances. This is because DP systems are primarily concerned with position-keeping, not for moving long distances. If setpoints would be far apart, the thrusters would aggressively try to cover the distance, in doing so causing significant fuel consumption and wear and tear to the thrusters. In such cases a simple autopilot is probably preferred.

### 2.11.1 Simple path following

Here we discuss a simple algorithm that uses waypoints for simple path following. Waypoints define the intended route of the ship. The ship is then given the task to visit each waypoint in a specified order. Path-following differs from trajectory tracking in that the time is not considered during entry to the next waypoint. These waypoints are specified by three variables (or setpoints): latitude, longitude, and vessel heading. Waypoints can be defined as a set of triplets:  $W = \{(x_1^n, y_1^n, \psi_1), (x_2^n, y_2^n, \psi_2), \dots, (x_N^n, y_N^n, \psi_N)\}$ . We define some parameter  $d$  that represents the distance between the ship position and the active waypoint that triggers a change to the next waypoint (assuming also that the heading is less than  $w$  radians from the heading setpoint). Figure 2.14 depicts the principle of this simple path following strategy.

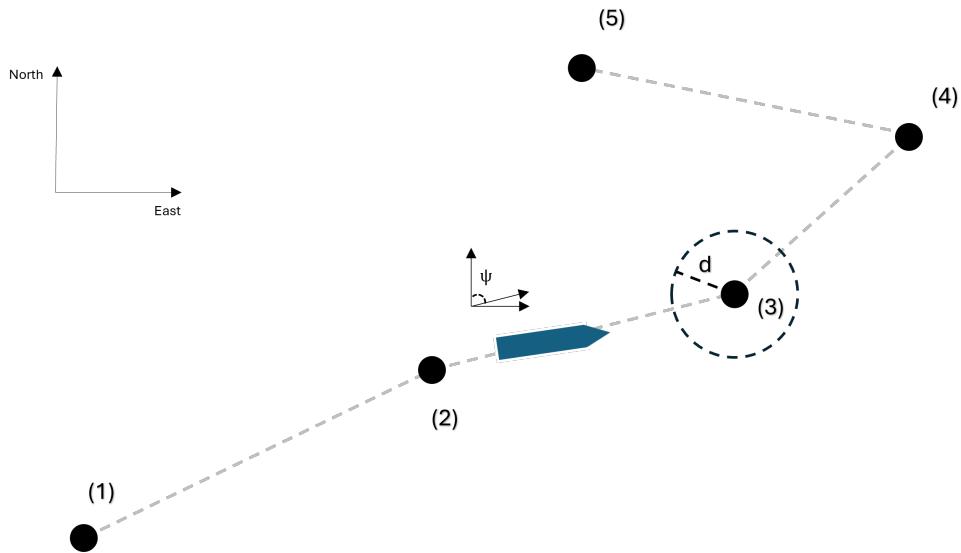


Figure 2.14: Depiction of path following with five waypoints. The vessel turns to the next waypoint when it is within some specified distance  $d$  to the active waypoint and the vessel heading is less than  $w$  radians from the heading setpoint.

### 2.11.2 Trajectory tracking

Another approach is to implement a form of trajectory tracking. This method was inspired by [27]. The operator defines a set of waypoints, just like the path following algorithm (adding velocities at each setpoint for smoothness and including time of arrival), but by using cubic spline interpolation between these waypoints an array of setpoints for each timestep can be obtained.

Each waypoint is associated with five parameters: time ( $t_k$ ), north position  $x_k^n$ , east position  $y_k^n$ , north velocity ( $\dot{x}_k^n = v_{x,k}$ ) and east velocity ( $\dot{y}_k^n = v_{y,k}$ ). The vessel heading setpoint

can be chosen as the angle between the previous and the currently active position set-points. When the current time  $t$  is  $t_k \leq t < t_{k+1}$  a polynomial can be defined for the north position as given in (2.144).

$$x^n = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (2.144)$$

The corresponding north velocity is found from the derivative of this function, see equation (2.145).

$$\dot{x}^n = a_1 + 2a_2 t + 3a_3 t^2 \quad (2.145)$$

A similar equation can be found for the east direction. To find the coefficients of the polynomial it is important to realize that both the endpoints have known position and velocity. The matrix relation shown in equation (2.146) must be true.

$$\underbrace{\begin{bmatrix} 1 & t_k & t_k^2 & t_k^3 \\ 1 & t_{k+1} & t_{k+1}^2 & t_{k+1}^3 \\ 0 & 1 & 2t_k & 3t_k^2 \\ 0 & 1 & 2t_{k+1} & 3t_{k+1}^2 \end{bmatrix}}_{X_k} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}}_a = \underbrace{\begin{bmatrix} x_k^n \\ x_{k+1}^n \\ v_{x,k} \\ v_{x,k+1} \end{bmatrix}}_{w_{x,k}} \quad (2.146)$$

From this it is clear that the unknown  $a_i$  coefficients can be determined by simple matrix manipulation, as shown in equation (2.147).

$$a = X_k^{-1} w_{x,k} \quad (2.147)$$

The same logic can be applied to the east direction; see equations (2.148) and (2.149).

$$\underbrace{\begin{bmatrix} 1 & t_k & t_k^2 & t_k^3 \\ 1 & t_{k+1} & t_{k+1}^2 & t_{k+1}^3 \\ 0 & 1 & 2t_k & 3t_k^2 \\ 0 & 1 & 2t_{k+1} & 3t_{k+1}^2 \end{bmatrix}}_{Y_k} \underbrace{\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}}_b = \underbrace{\begin{bmatrix} y_k \\ y_{k+1} \\ v_{y,k} \\ v_{y,k+1} \end{bmatrix}}_{w_{y,k}} \quad (2.148)$$

$$b = Y_k^{-1} w_{y,k} \quad (2.149)$$

The heading setpoint is calculated as the angle between the setpoints. The trajectory of setpoints are then calculated from equation 2.144. For  $N$  waypoints it is necessary to solve  $2(N - 1)$  matrix equations for finding all the necessary polynomial coefficients.

## 2.12 Green DP

Green DP is a function offered by Kongsberg Maritime that aims to reduce fuel consumption (and reduce the environmental footprint) by allowing the vessel to operate within

a predefined area rather than a specific setpoint. According to [5], this can reduce fuel consumption by as much as 20 percent. It is also noted that the technology utilizes non-linear model predictive control. Although the author has no knowledge of the underlying cost function, a simple candidate cost function is used here. The area of operation defined here will be a circle around some point in north/east coordinates.

The cost function considered here is shown in equation (2.150).

$$\min_u J = \min_u \left[ \frac{1}{2} \sum_{k=1}^N (C(x_k) + u_{k-1}^T P u_{k-1}) \right] \quad (2.150)$$

Here,  $C(x_k)$  is some function that depends on the state vector. In this case, the operational area will be considered a circle, hence the cost term  $C(x_k)$  should strongly penalize movement outside the circle, but less for movement within the circle. Since it is often beneficial to use smooth functions in optimization processes, the cost term  $C(d_k)$  is defined as shown in equation (2.151).

$$C(d_k) = \begin{cases} W_i d_k & \text{if } d_k \leq d_0 \\ W_i d_k + W_o (d_k - d_o)^2 & \text{if } d_k > d_0 \end{cases} \quad (2.151)$$

Here,  $d_k$  is the Euclidean distance between the position of the ship and the center of the circular area of operation and  $d_o$  is the radius of the circle. The distance  $d_k$  can be calculated directly from the state vector  $x_k$ .  $W_i$  and  $W_o$  are weights. Note that this is a continuously differentiable function ( $C^1$ ), implying that the function and its first derivative are both continuous. The main idea is to punish movement outside the circle with radius  $d_o$  by setting  $W_o >> W_i$ . The cost function calculations are visually depicted in Figure 2.15. The second term in the cost function (see 2.150) is necessary to penalize thruster usage, which, of course, is the intention behind this algorithm. Without this condition, the optimization routine is free to generate any force/moment in surge/sway and yaw as long as the vessel is within the operational area.

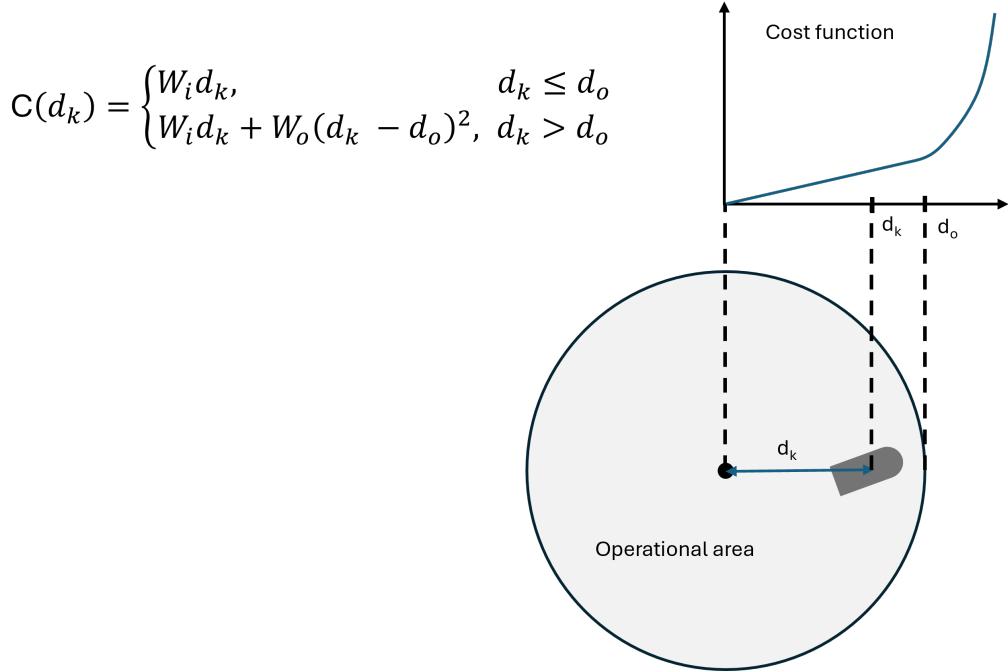


Figure 2.15: Figure depicts how the cost function is calculated when vessel is within and outside the operational area.

## 2.13 Thruster allocation

This chapter discusses the various types of thrusters, the principles of thruster allocation, and some algorithms that calculate the thruster setpoints. Both rotatable and non-rotatable thrusters will be considered.

### 2.13.1 Thruster types

With the exception of the OSV simulation, all the controllers used calculated the net force and moment required from thrusters. In the real world, the vessel in question must realize the net force/moment by adjusting the setpoints to the thrusters. This is the process known as thruster allocation: What force should every thruster generate and at what angle (in case of azimuths) to realize the demand from the controller.

There are typically two types of thrusters used in DP, the fixed pitch (FP) and the controllable pitch (CP) thrusters [2]. Pitch in this context refers to the angle of the propeller blades. The fixed pitch thruster works by varying the speed, rotations per minute (RPM), of the propeller. The force can then be modeled as in equation (2.152).

$$F(n) = K_n |n| \quad (2.152)$$

Here  $K$  represents the thruster force coefficient and  $n$  is the thruster speed in RPM. The pitch remains constant while the rotational speed of the propeller can change. Notice that the square of the rotation is proportional to the force generated by the propeller. Likewise, the force formula for the controllable pitch thruster takes the form depicted in equation (2.153).

$$F(n, p) = K(n)(p - p_0)|p - p_0| \quad (2.153)$$

Here  $p_0$  represents pitch offset. The thruster is assumed to operate on some fixed rotational speed and the force is generated by adjusting the angles of the propeller blades. It is interesting to note that the controllable pitch thruster is subject to constant rotation, and hence even when generating no thrust, it will still consume power.

There are different categories of thrusters, some of them are listed below:

- **Main propeller(s):** These are the thrusters placed at the very stern of the ship. They are the primary propulsive system and generate a strong thrust in surge. Rudders can be installed to change vessel heading.
- **Tunnel thruster:** Generates force in sway direction and moment in yaw. A tunnel is cut through the hull with a propeller that forces water through the tunnel, creating thrust.
- **Azimuth thruster:** Azimuth thrusters are propellers that can rotate, thus they are able to not only generate thrust, but also decide on the direction of that thrust.

A depiction of different thrusters on a vessel is depicted in Figure 2.16.

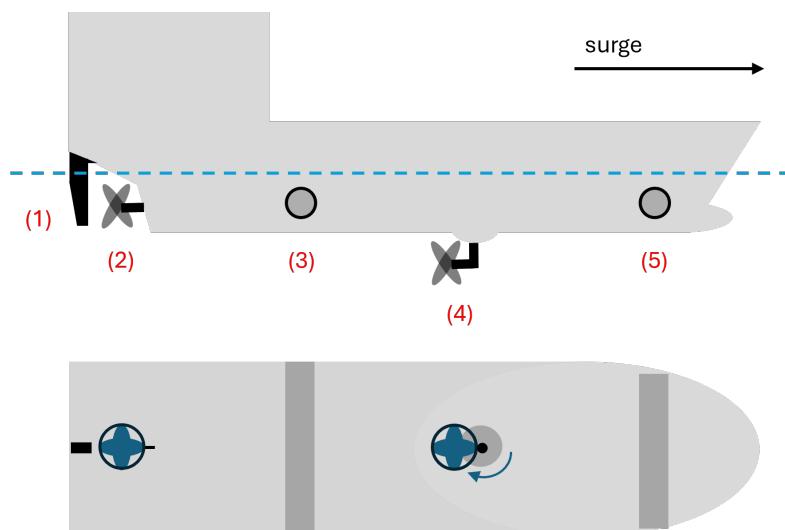


Figure 2.16: Depiction of different thrusters. Here (1) marks the rudder, (2) is the main propeller, (3) is the stern tunnel thruster, (4) is the azimuth thruster, and (5) is the bow tunnel thruster.

Note that thrusters do not immediately adapt to new setpoint changes. Such operations are naturally associated with some time delay (time constants). But, in the following, we will assume that there is no time delay between thrusters receiving the setpoint and realizing the corresponding force output.

### 2.13.2 Thruster configuration

An example of a thruster configuration is depicted in Figure 2.17.

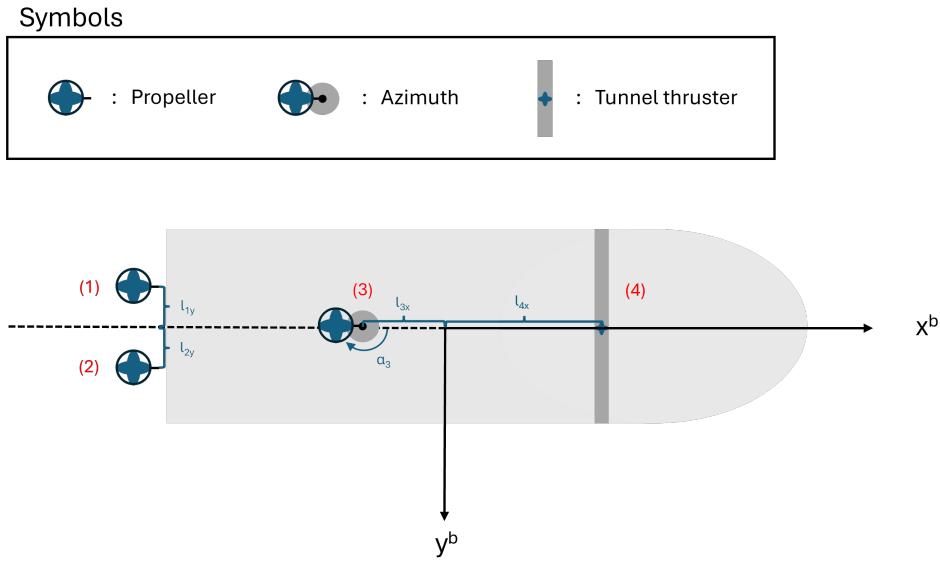


Figure 2.17: Depicts an example of a thruster configuration for a ship having both rotatable and non-rotatable thrusters. Here (1) and (2) marks the port and starboard main propellers respectively. An azimuth thruster is marked by (3). Finally, a bow tunnel thruster is placed at point (4).

The body coordinate system should be placed in the center of gravity of the vessel. The force generated by each thruster can be defined by a vector  $f$ . This vector is related to the individual control input to each thruster through equation (2.154).

$$f = Ku \quad (2.154)$$

Here,  $u$  is the control input to each thruster and  $K$  is the force coefficient matrix. The interpretation of  $u$  can vary; for example, it can be rotations per minute, pitch-angle, or a voltage signal. The net thruster forces can be obtained using a thruster configuration matrix  $T$ . This is shown in equation (2.155).

$$\tau = TKu \quad (2.155)$$

Here,  $\tau$  is the net force calculated by the control algorithm. The thruster configuration matrix,  $T$ , computes the net force and moment from the individual thruster force contributions. For the ship depicted in Figure 2.17 the thruster configuration matrix takes the form shown in equation (2.156).

$$\underbrace{\begin{bmatrix} X \\ Y \\ N \\ \tau \end{bmatrix}}_{\boldsymbol{\tau}} = \underbrace{\begin{bmatrix} 1 & 1 & \cos(\alpha) & 0 \\ 0 & 0 & \sin(\alpha) & 1 \\ l_{1y} & -l_{2y} & -l_{3x}\sin(\alpha) & l_{4x} \end{bmatrix}}_T \underbrace{\begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 \\ 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & K_4 \end{bmatrix}}_K \underbrace{\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}}_u \quad (2.156)$$

Each column in the thruster configuration matrix,  $T$ , represents one thruster, and the column entries explains how the thruster contributes to the net force/moment. The distances  $l_i$  are here considered to be positive. Note that the thruster configuration matrix contains the azimuth angle ( $\alpha$ ) as a parameter. The optimal solution of thruster allocation must solve not only the force setpoints on each thruster, but also the angles of the azimuths. If rudder angles were included, a similar complication would occur. Solving problems with azimuth angles significantly complicates the problem. Three solution methods will be discussed here. The first two only apply for ships with non-rotatable thrusters. The third allows for azimuths and rudders to be included.

### 2.13.3 Thruster allocation: explicit solution

This method is described in [2]. It solves for optimal input to each thruster given a non-rotatable thruster configuration. An example of such a vessel is depicted in Figure 2.18.

The optimization problem can be defined as in equations (2.157) and (2.158).

$$J = \min_f f^T W f \quad (2.157)$$

Subject to:

$$\tau - Tf = 0 \quad (2.158)$$

Here  $W$  is a weight matrix of the individual thruster forces. By changing  $W$  different thruster can be prioritized. The system could be set up to favor thrusters that require less power usage or that are more robust and capable of dealing with rapid speed/pitch changes. It is straightforward to demonstrate<sup>15</sup> that this leads to an expression on the

<sup>15</sup>Define the Lagrangian as follows:

$$L = \underbrace{f^T W f}_{\text{Cost function}} + \lambda^T \underbrace{(\tau - Tf)}_{\text{Constraint}}. \quad (2.159)$$

### Symbols

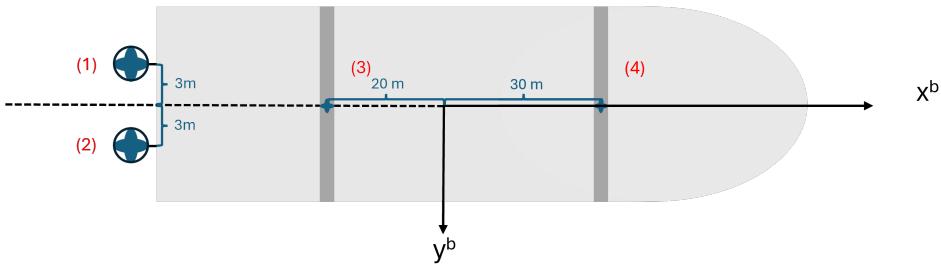
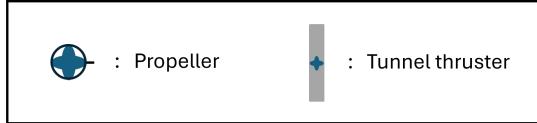


Figure 2.18: Depicts an example of a thruster configuration for a ship. Here (1) and (2) represents the port and starboard main propellers respectively. Two tunnel thrusters in bow and stern are depicted at (4) and (3).

form shown in equation (2.163).

$$u = K^{-1} \tilde{T} \tau \quad (2.163)$$

Where  $\tilde{T}$  is defined in equation (2.164).

$$\tilde{T} = W^{-1} T^T (TW^{-1} T^T)^{-1} \quad (2.164)$$

The force coefficient matrix,  $K$ , is always invertible due to the diagonal entries<sup>16</sup>.

---

The partial derivatives of  $L$  with respect to  $f$  and  $\lambda$  is then set equal to zero:

$$\frac{\partial L}{\partial f} = 2Wf - T^T \lambda = 0 \quad \frac{\partial L}{\partial \lambda} = \tau - Tf = 0 \quad (2.160)$$

From these two expression the following relation between the  $f$  and  $\tau$  is obtained:

$$f = \underbrace{W^{-1} T^{-1} (TW^{-1} T^T)^{-1}}_{\tilde{T}} \tau \quad (2.161)$$

Since  $f = Ku$ :

$$u = K^{-1} \tilde{T} \tau \quad (2.162)$$

<sup>16</sup>A square diagonal matrix with non-zero entries is always invertible. This is shown below:

$$W = \begin{bmatrix} w_{11} & 0 & \cdots & 0 \\ 0 & w_{22} & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & w_{nn} \end{bmatrix} \quad W^{-1} = \begin{bmatrix} \frac{1}{w_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{w_{22}} & \cdots & 0 \\ \vdots & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \frac{1}{w_{nn}} \end{bmatrix} \quad (2.165)$$

Thus, the problem has an explicit solution that can be easily solved with minor computational load. Note that the formula for  $\tilde{T}$  becomes even simpler if the weighting matrix is set equal to the identity matrix ( $W = I$ ), i.e.  $\tilde{T} = W^{-1}T^T(TW^{-1}T^T)^{-1} = T^T(TT^T)^{-1}$ .

It should be stressed that this method can handle azimuth thrusters, but only under the condition that the azimuth thrusters have fixed angles. Enforcing this limitation would be a suboptimal approach as it would limit the thruster allocation capabilities of the vessel.

#### 2.13.4 Thruster allocation: quadratic programming approach

This method is based on [16] and [2]. This is a slightly more advanced method using a quadratic programming approach. The following optimization problem is suggested: (2.169).

$$J = \min_{(f,s,\tilde{f})} [f^T W f + s^T Q s + \beta \tilde{f}] \quad (2.166)$$

Subject to:

$$T f = \tau + s \quad (2.167)$$

$$f_{\min} \leq f \leq f_{\max} \quad (2.168)$$

$$-\tilde{f} \leq f_1, f_2, \dots, f_r \leq \tilde{f} \quad (2.169)$$

Here  $f$  is force applied to each thruster (a total of  $r$  thrusters). The force vector is related to the input through  $f = Ku$ . In other words, from knowing the value of  $f$ , it is straightforward to calculate the inputs to the individual thrusters, as seen in equation (2.170).

$$u = K^{-1} f \quad (2.170)$$

The optimization variable  $s$  is a slack variable. This punishes mismatch between the force and moment required by the controller and the actual force and moment delivered by the thrusters according to the allocation algorithm. The  $\tilde{f}$  term attempts to avoid any one thruster having a disproportionately large output compared to the other thrusters. Maximum and minimum thruster output can be defined by specifying  $f_{\max}$  and  $f_{\min}$ . Finally,  $W$  and  $Q$  are weighting matrices, and  $\beta$  is a scalar weighting term.

The optimization problem can be reformulated as QP problem on standard form as shown

---

Clearly  $WW^{-1} = W^{-1}W = I$ .

below.

$$J = \min_z [z^T \Phi z + z^T R p] \quad (2.171)$$

Subject to:

$$A_1 z = C_1 p \quad (2.172)$$

$$A_2 z \leq C_2 p \quad (2.173)$$

Here  $z = [f^T, s^T, \tilde{f}]^T$  and  $p = [\tau^T, f_{\min}^T, f_{\max}^T, \beta]^T$ , From this the following matrices are derived:

$$\Phi = \begin{bmatrix} W & 0_{r \times n} & 0_{r \times 1} \\ 0_{n \times r} & Q & 0_{n \times 1} \\ 0_{1 \times r} & 0_{1 \times n} & 0 \end{bmatrix} \quad R = \begin{bmatrix} 0_{(r+n) \times (n+2r)} & 0_{(r+n) \times 1} \\ 0_{1 \times (n+2r)} & 1 \end{bmatrix} \quad (2.174)$$

$$A_1 = \begin{bmatrix} T - I_{n \times n} & 0_{n \times 1} \end{bmatrix} \quad A_2 = \begin{bmatrix} -I_{r \times r} & 0_{r \times n} & 0_{r \times 1} \\ I_{r \times r} & 0_{r \times n} & 0_{r \times 1} \\ -I_{r \times r} & 0_{r \times n} & -1_{r \times 1} \\ I_{r \times r} & 0_{r \times n} & -1_{r \times 1} \end{bmatrix} \quad (2.175)$$

$$C_1 = \begin{bmatrix} I_{n \times n} & 0_{n \times (2r+1)} \end{bmatrix} \quad C_2 = \begin{bmatrix} I_{r \times n} & -I_{r \times r} & 0_{r \times r} & 0_{r \times 1} \\ 0_{r \times n} & 0_{r \times r} & I_{r \times r} & 0_{r \times 1} \\ 0_{r \times n} & 0_{r \times r} & 0_{r \times r} & 0_{r \times 1} \\ 0_{r \times n} & 0_{r \times r} & 0_{r \times r} & 0_{r \times 1} \end{bmatrix} \quad (2.176)$$

### 2.13.5 Thruster allocation: Nonlinear optimization approach

A thruster allocation method that allows for azimuth thrusters is suggested by [17] and [2]. The non-linear optimization problem is given in equations (2.177), (2.178), (2.179), (2.180), and (2.181)

$$J = \min_{f, \alpha, s} \left[ \sum_i^r P_i |f_i|^{\frac{3}{2}} + s^T Q s + (\alpha - \alpha_0)^T \Omega (\alpha - \alpha_0) + \frac{\rho}{\varepsilon + \det(T(\alpha) W^{-1} T^T(\alpha))} \right] \quad (2.177)$$

Subject to:

$$T(\alpha) f = \tau + s \quad (2.178)$$

$$f_{\min} \leq f \leq f_{\max} \quad (2.179)$$

$$\alpha_{\min} \leq \alpha \leq \alpha_{\max} \quad (2.180)$$

$$\Delta \alpha_{\min} \leq \alpha - \alpha_0 \leq \Delta \alpha_{\max} \quad (2.181)$$

Note in particular how the thruster configuration matrix now depends on the the azimuth angles ( $\alpha$ ). The objective function (2.177) consists of four terms:

- $\sum_i^r P_i |f_i|^{\frac{3}{2}}$ : This represents the thruster power usage of a vessel with  $r$  thrusters. Here it is assumed that costs related to power consumption is proportional to the sum of each thruster force raised to the power of  $\frac{3}{2}$ . This is a necessary condition to limit the generation of thruster forces. It has been observed through simulations that not including this term in the optimization routine could lead to solutions where the azimuths negated each other. A pair of azimuth thrusters could create thrust in the direction of the other, creating an overall zero net force, but each thruster individually generates considerable thrust.  $P_i$  are weights associated with each thruster ( $r$  thrusters in total).
- $s^T Q s$ : Penalty term with respect to the slack variable  $s$ . The slack variables represent the deviation between the requested forces and moment from the control system and the solution from the thruster allocation algorithm, i.e.  $T(\alpha)f = \tau + s$ . The  $Q$  term is a weighting matrix.
- $(\alpha - \alpha_0)^T \Omega (\alpha - \alpha_0)$ : Large deviation between the new  $(\alpha)$  and previous  $(\alpha_0)$  azimuth thruster angles is penalized.  $\Omega$  is a weighting matrix.
- $\frac{\rho}{\epsilon + \det(T(\alpha)W^{-1}T^T(\alpha))}$ : This equation is easy to understand by looking back at the fundamental force relation between thrusters and the corresponding net forces and moment acting on the ship:

$$\tau = W^{-1} T^T(\alpha) (T(\alpha) W^{-1} T^T(\alpha))^{-1} \quad (2.182)$$

For this relation to exist both both  $W$  and  $T(\alpha)W^{-1}T^T(\alpha)$  must be non-singular.  $W$  is a diagonal weighting matrix with non-zero entries along the diagonal, and hence always invertible. However, to avoid  $T(\alpha)W^{-1}T^T(\alpha)$  becoming non-invertible the cost function punishes situations where the values of  $\alpha$  make the term close to non-invertible. The value  $\rho$  is a measure of how strong this penalty is relative to other terms in the cost function. A small value  $\epsilon$  is added to the denominator to ensure that zero division never occurs in the case of a zero determinant.

The equality and inequality constraints are similar to the previous algorithms. The only thing to note is that they include limitations on the absolute value of the thruster angles and the angle deviation between the timesteps. These limits are either due to mechanical limits on the azimuths or to reduce wear and tear on the equipment.

## 2.14 MISC

An error measure commonly used is the Integrated Absolute Error, IAE. It is a measure of how well the control algorithm manages to reduce the error between the setpoint and

actual output. The discretized version of this metric is shown in equation (2.183).

$$\text{IAE} = \sum_{k=1}^n |e_k| \Delta t \quad (2.183)$$

Here,  $e_k = r_k - y_k$  where  $r_k$  is the setpoint and  $y_k$  is the output variable being controlled at time  $t_k$ . The simulation time step is denoted by  $\Delta t$  and the total number of samples is equal to  $n$ .

Another common measure for how well the system manages to track the setpoint is the Root Mean Squared Error (RMSE) metric. This metric is shown in equation (2.184).

$$\text{RMSE} = \sqrt{\frac{\sum_{k=1}^n (r_k - y_k)^2}{n}} \quad (2.184)$$

We define the Total Value (TV) metric for the inputs calculated by the controllers. Following the SNAME notation, the metric takes different forms depending on whether surge forces ( $X$ ), sway forces ( $Y$ ) or yaw momentum ( $N$ ) are considered.

$$\text{TV(surge)} = \sum_{i=1}^n |X_i| \Delta t \quad (2.185)$$

$$\text{TV(sway)} = \sum_{i=1}^n |Y_i| \Delta t \quad (2.186)$$

$$\text{TV(yaw)} = \sum_{i=1}^n |N_i| \Delta t \quad (2.187)$$

These are important metrics to evaluate the aggressiveness of the controllers.

For individual thruster forces, we can define a total force metric, Integrated Absolute value of (thruster) Force (IAF).

$$\text{IAF}_i = \sum_{k=1}^n |f_{i,k}| \Delta t \quad (2.188)$$

This is an important metric for estimating the force generation during thruster allocation. The index  $i$  refers to the numbering of thrusters.

### 2.14.1 Monte Carlo simulations

Monte Carlo simulations are performed in several places in this document. Each simulation is run a total of 100 times. In some cases this can be quite time consuming, especially in the case of the green DP algorithm or non-linear model predictive control.

The environmental disturbances and setpoints are created using pseudo-random numbers generated by the Mersenne Twister<sup>17</sup> (or `rand('twister', seed)` function in MATLAB).

The environmental forces are generated using the random walk algorithm. In every simulation, the vessel starts in the zero position (origin) and is then subjected to two setpoint changes (occurring at time  $t = 100$  and  $t = 600$ ). The vessel is moved each time in some random direction (which also becomes the new heading setpoint). The new setpoint is always moved 10 meters from the previous setpoint. All simulations are run for a total time of 1000 seconds with a timestep ( $\Delta t$ ) of 1 second.

In the case of green DP, only the environmental disturbances are varied between the simulations, but the setpoint or border remains fixed. The main point of interest here is not the setpoint error, but thruster usage (or fuel consumption) during the simulation.

In the case of thruster allocation, both the environmental forces and the setpoints are changed. However, the focus here is on the effectiveness of the allocation algorithms measured in thruster force usage.

---

<sup>17</sup>According to the original paper [31], the Mersenne twister is Pseudo-Random Number Generator (PRNG) with an astonishing period of  $2^{219937} - 1$ . Comparing this to the estimated number of atoms in the observable universe (around  $10^{80}$ ) suggests that this period should be fine for every conceivable application. The Mersenne twister strikes a nice balance between generating well-behaving random numbers at a computationally low cost.

# 3 Results

This section contains results from running different controller algorithms (LQ optimal control, MPC and NMPC), system identification, tracking strategies, green DP and thruster allocation.

## 3.1 LQ Optimal Control

This section looks at LQ optimal control applied to the supply model. The simulations were run with and without external disturbances (wind, waves, and current). Then the computational effect of pre-storing certain heading-dependent matrices (like the controller gain) is discussed.

### 3.1.1 Simulating with and without disturbances

This section looks into LQ optimal control (in deviation form) applied to the supply model where the true vessel heading is assumed to be known from the gyrocompass. The control structure is depicted in Figure 3.1.

All simulations here start at the origin with the vessel heading set pointing towards the north (0 degrees). Then at the 100 second mark the setpoint changes to 10 meters north and 5 meters east. The vessel heading is changed to 26 degrees. After 600 seconds, the setpoint is changed to -5 meters north and -5 meters east. The vessel heading is set to 225 degrees. The entire simulation lasts for a 1000 seconds. The timestep is set to  $\Delta t = 1.0$  during the simulation.

For the simulation subjected to external disturbances, the wind, current, and wave disturbances are depicted in Figures 3.2, 3.3, and 3.4, respectively. The current figure shows the current force in the north and east directions, and also includes the estimates from the Kalman filter. It takes about 4 minutes before the estimates are close to the true value. Of course, this depends on the Kalman filter tuning, but nonetheless, it indicates that the time for reaching good current estimates takes several minutes. This is important for model-based algorithms, such as model predictive control, which use these estimates to calculate the future horizon. The LQ optimal control algorithm used here is in deviation

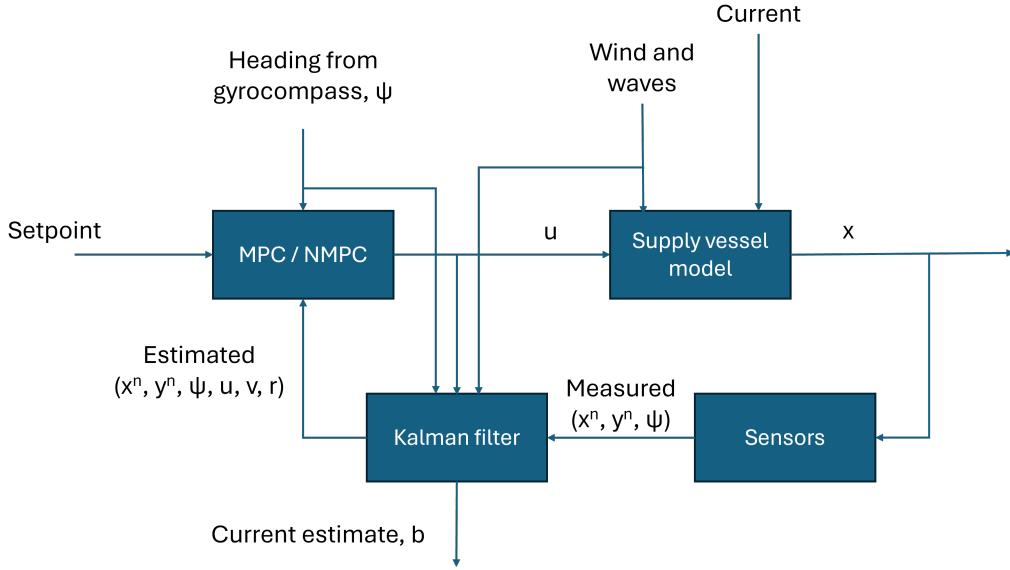


Figure 3.1: Plot depicts the control structure for linear quadratic optimal control when applied to the supply model. Notice that the Kalman filter estimates the current estimate, but it is not used in the controller.

form (removing slowly varying disturbances) and hence is not directly affected by these current estimates.

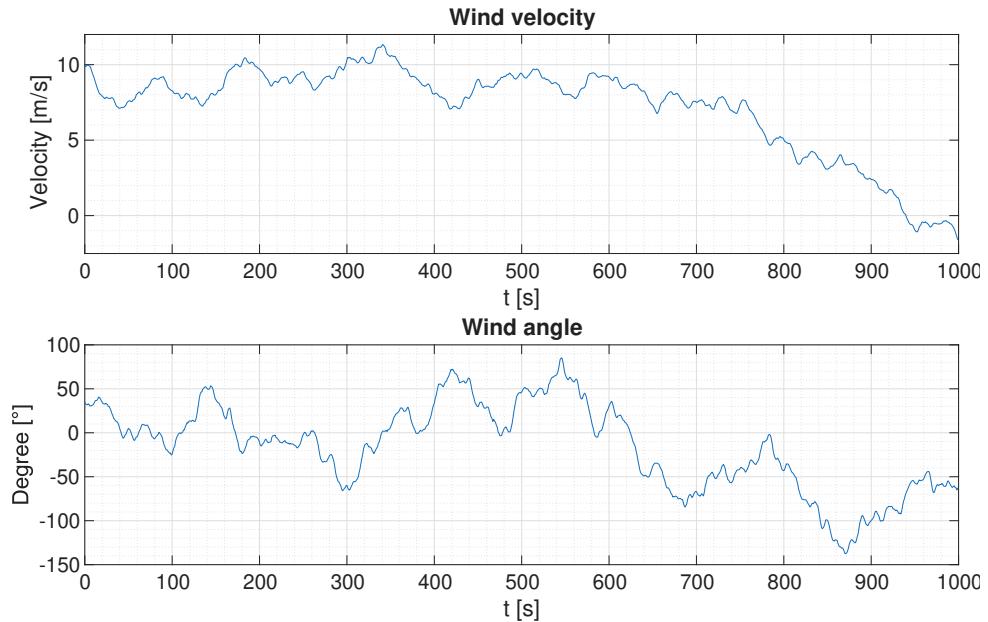


Figure 3.2: Plot depicts the wind disturbance used when simulating dynamic positioning using linear quadratic optimal control. It depicts the wind speed and angle of attack.

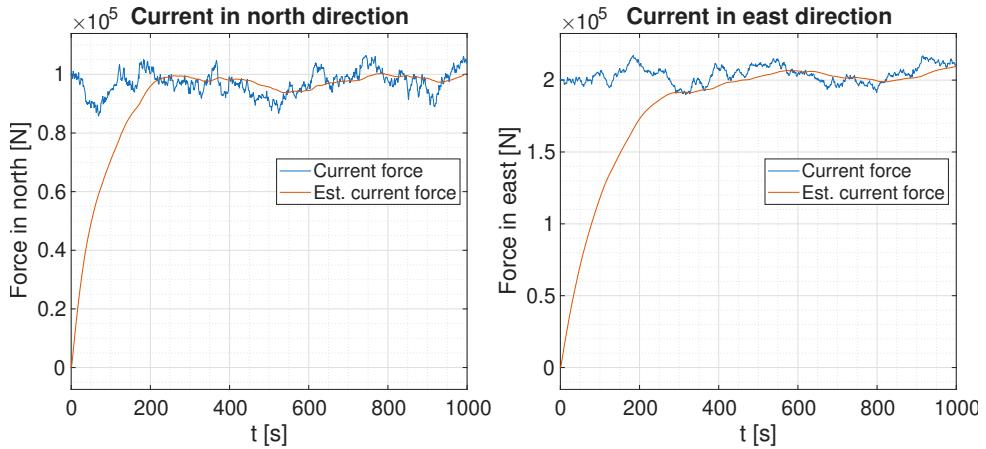


Figure 3.3: Plot shows current forces in north and east direction. It also includes the force estimates from the Kalman filter. Note, it takes about 4 minutes before the estimates reach close to the true values.

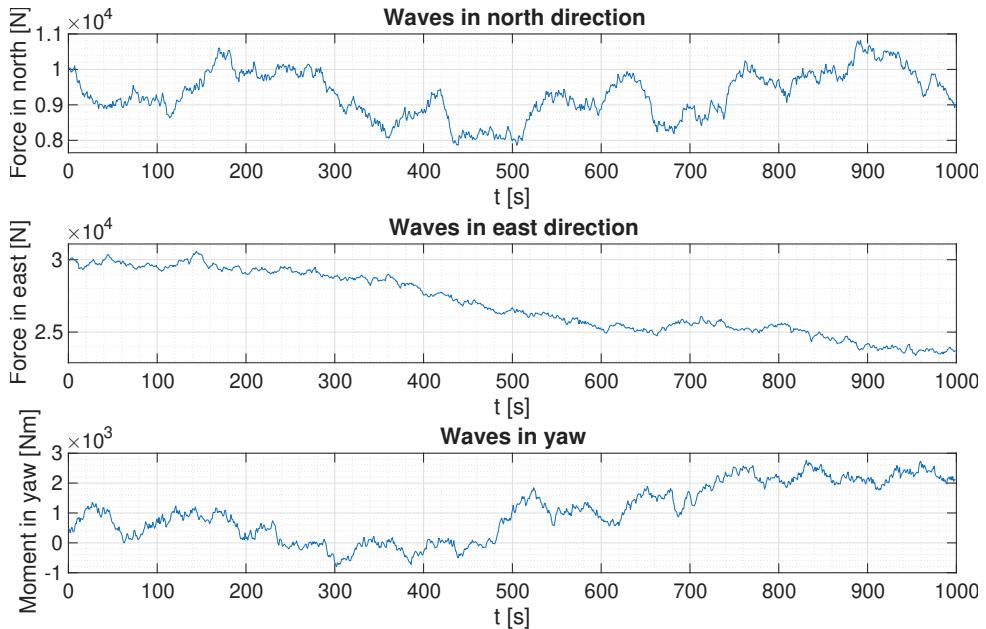


Figure 3.4: Plot depicts wave forces in north and east direction, and moment in yaw.

The path plots for both simulations are shown in Figure 3.5. Setpoint tracking and inputs are depicted in Figure 3.6 and 3.7, respectively. It is clear that the controller manages to settle to the new setpoints. It is not a very aggressive tuning, as the vessel heading takes some time to reach its intended angle. There are many considerations to take into account when tuning a vessel:

- Fuel consumption

- Wear and tear on the thrusters
- Safety - A quickly accelerating vessel can cause discomfort for the crew and at worst cause dangerous situations.
- Aggressive tuning can cause significant overshoot, potentially creating dangerous situations if other vessels or sea structures are in the vicinity of the maneuver.

The forces/moment calculated by the LQ optimal controller are close to zero during most of the non-disturbance simulation run. This makes sense since the vessel will not need any net force or moment after settling at its target position. This is not the case when introducing wind, waves, and current, as the ship will constantly counteract these disturbances to stay at the setpoint.

The change in the Kalman filter gain is shown in Figure 3.8. The Kalman gain matrix entries change during the period where the vessel adjusts to new setpoints. This is expected due to the heading dependency of the model matrices. The heading changes when the vessel approaches a new setpoint and, after reaching the setpoint, the heading remains constant.

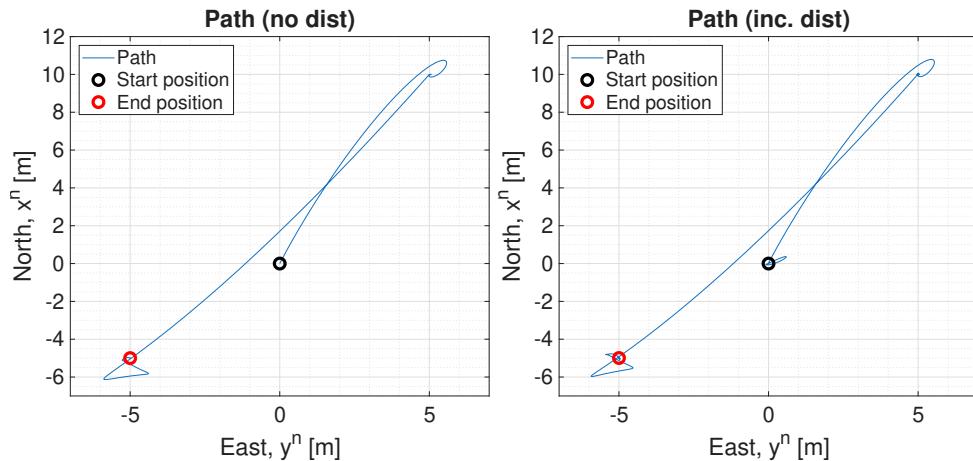


Figure 3.5: Depicts the ship movement in the NED geographic coordinate frame. The ship starts in the origin pointing north, then is subjected to a setpoint change at  $t = 100$  to position  $[10, 5, 26^\circ]$  and  $t = 600$  to position  $[-5, -5, 225^\circ]$ . The plot to the left is from a simulation without external disturbances, while the plot to the right includes such disturbances.

The integrated absolute error (IAE) and the total value (TV) are summarized in Table 3.1. Not surprisingly, the simulation without disturbance has the lowest IAE value. The biggest difference between the two simulations is the amount of force and moment required to maintain the setpoint. The simulation that includes disturbances requires considerably more thruster usage.

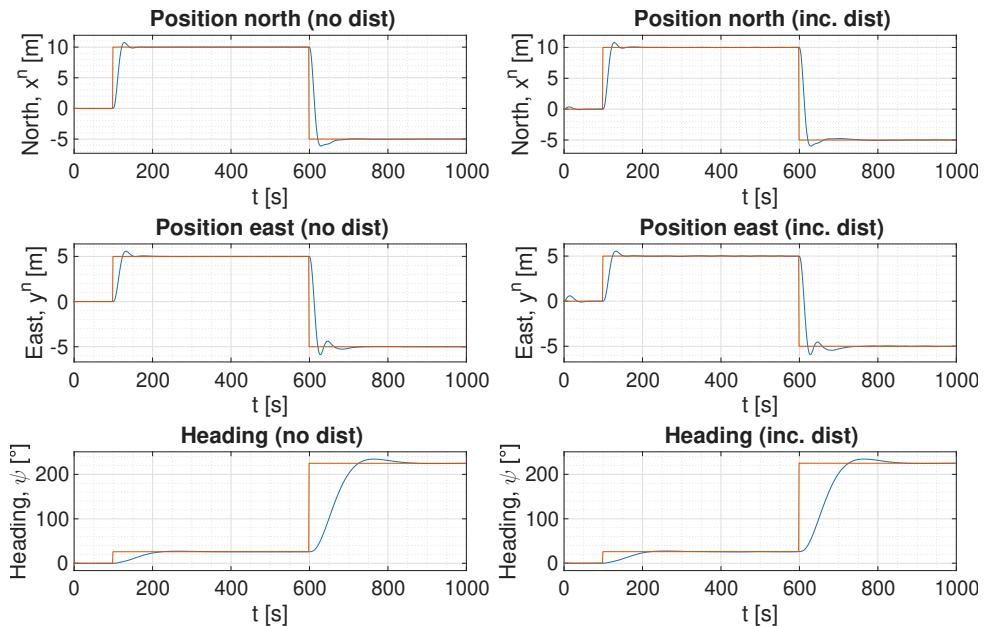


Figure 3.6: Depicts setpoint tracking for the LQ optimal control simulations with and without external disturbances.

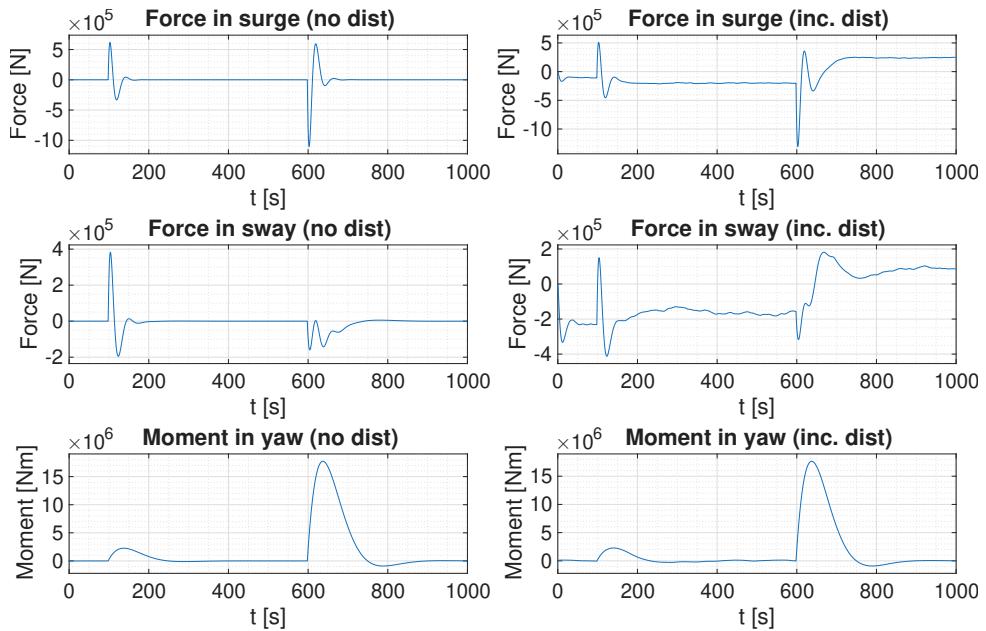


Figure 3.7: Inputs computed by LQ optimal control algorithm. These inputs represent net force in surge/sway and moment in yaw.

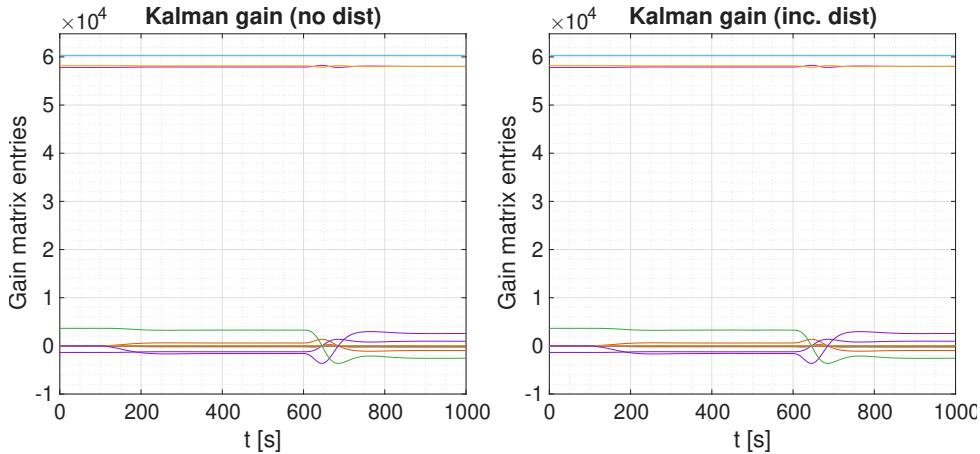


Figure 3.8: Depicts the individual entries in the Kalman filter matrix. Notice that the entries change when the vessel adapt to a new setpoint. This is due to the heading dependency ( $\psi$ ) of the model matrices.

Table 3.1: Comparison of integrated absolute error (IAE) and total value (TV) for LQ optimal control simulations with and without disturbances.

Description	Disturbance	No disturbance	Difference
IAE - North position	383.75	370.77	12.98
IAE - East position	273.95	248.14	25.81
IAE - Heading	263.80	263.11	0.69
TV - North position	$2.11 \cdot 10^8$	$0.29 \cdot 10^8$	$1.82 \cdot 10^8$
TV - East position	$1.47 \cdot 10^8$	$0.16 \cdot 10^8$	$1.31 \cdot 10^8$
TV - Heading	$16.99 \cdot 10^8$	$16.71 \cdot 10^8$	$0.28 \cdot 10^8$

### 3.1.2 Monte Carlo simulation

To obtain an improved estimate of the linear quadratic optimal controller performance, Monte Carlo simulations were run with varying disturbances and setpoint changes as discussed in Section 2.14.1. The RMSE and IAE values for individual simulations are shown in Figure 3.9. The mean values of RMSE, IAE and TV are shown in Table 3.2.

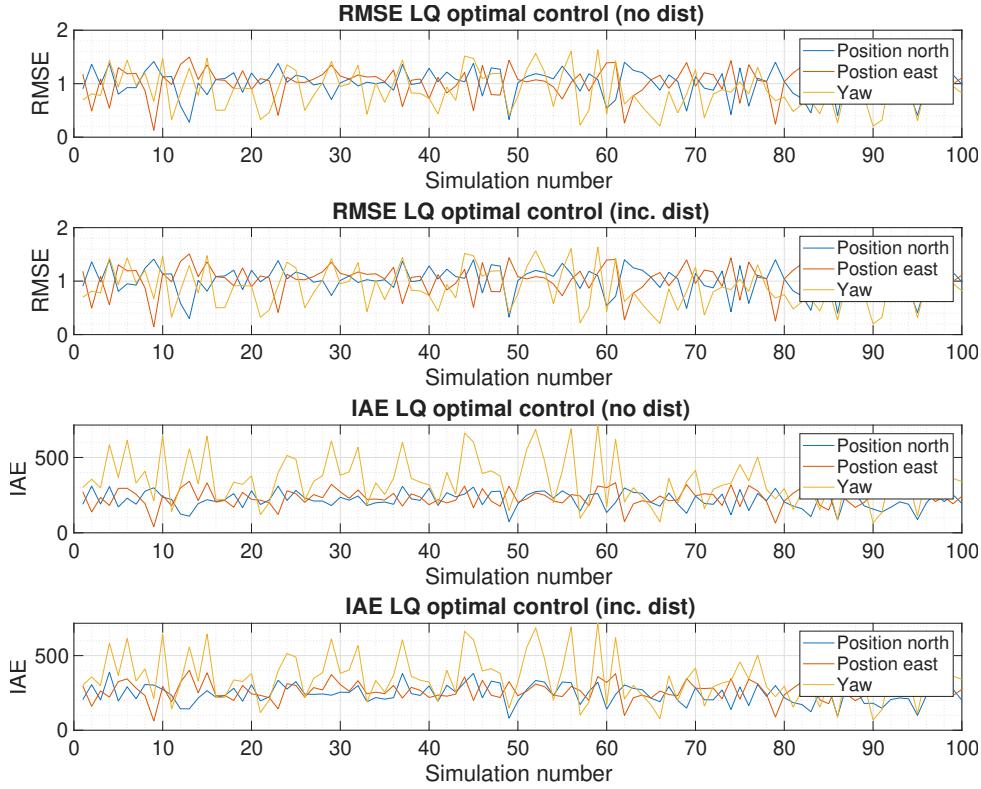


Figure 3.9: Monte Carlo simulations of supply model with linear quadratic optimal controller. Depicts RMSE and IAE values for every simulation with and without external disturbances (wind, waves and current).

Table 3.2: Mean values of RMSE, IAE and TV from Monte Carlo simulations of LQ optimal control applied to the supply model.

Metric	Description	Disturbance	No disturbance
RMSE	North position	1.01	1.01
	East position	1.05	1.05
	Heading	0.88	0.88
IAE	North position	242.58	217.88
	East position	264.22	234.34
	Heading	347.10	345.38
<b>Total</b>	-	853.90	797.50
TV	North position	$1.65 \cdot 10^8$	$0.13 \cdot 10^8$
	East position	$1.78 \cdot 10^8$	$0.25 \cdot 10^8$
	Heading	$22.60 \cdot 10^8$	$21.94 \cdot 10^8$
<b>Total</b>	-	$2.60 \cdot 10^9$	$2.23 \cdot 10^9$

### 3.1.3 Computational improvement when prestoring matrices

In order to avoid calculating matrices like the controller gain and Kalman gain at each new timestep, it is possible to calculate these matrices in advance for some specified precision of the heading angle (remember that these matrices depend on the rotational matrix which is periodic, with period  $2\pi$ ). This approach was outlined in Section 2.7.5.

In this text, the heading-dependent matrices are calculated for every heading angle with a resolution of 0.1 degrees. In other words, in the case of the controller gain, the following matrices will be computed (or loaded)  $[G(0^\circ), G(0.1^\circ), G(0.2^\circ) \dots G(359.8^\circ), G(359.9^\circ)]$  before the simulation starts. During simulation, at any time  $t_k$ , the value of  $\psi$  is rounded to the closest value defined in the precalculated matrix array, and the corresponding gain is used in the simulation.

To obtain a reasonable estimate of the time difference running each mode, 100 simulations were run for each scenario<sup>1</sup> (prestored and dynamically calculated matrices). The results are shown in Figure 3.10.

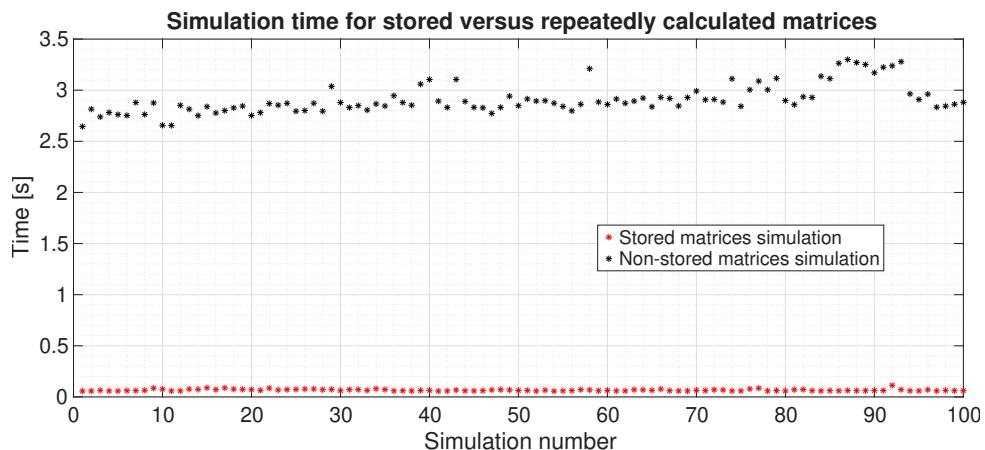


Figure 3.10: A hundred 100 simulation were performed for each mode: using pre-stored matrices or dynamically calculated matrices. The elapsed time for each simulation is depicted in the plot.

The average simulation time was 2.9103 seconds for the simulations that calculated the heading-dependent matrices at each time step. Using precalculated matrices reduced the average simulation time to 0.066694. The simulation using precalculated matrices

---

<sup>1</sup>The simulations were run on a computer with the following specifications:

- **CPU:** Intel Core i7-14700F
- **GPU:** GAINWARD RTX 5070 Ti PHOENIX V1 16GB
- **RAM:** Kingston Fury Beast RGB 32GB 6000
- **Motherboard:** MSI B760 MAG TOMAHAWK WIFI D5

runs 43.6 times faster than the alternative simulation. The differences in performance between these two simulations are minuscule. This can be seen by comparing the setpoint tracking between the two simulations as shown in Figure 3.11. The same environmental disturbances used previously are reused here (see Figures 3.2, 3.3 and 3.4).

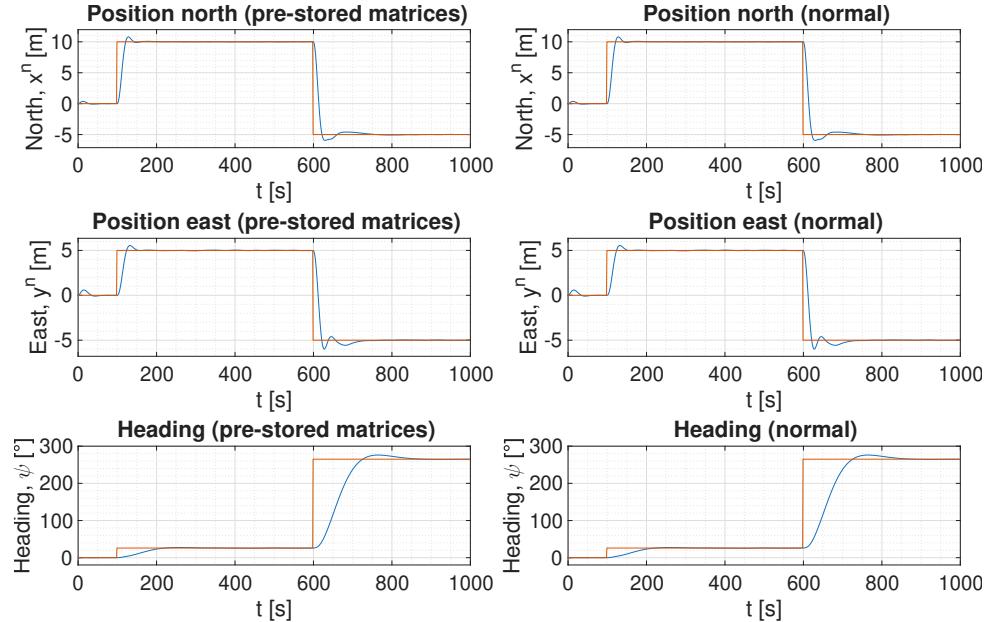


Figure 3.11: Comparison of simulation with or without prestored matrices when running LQ optimal control on the supply model. Visually there is no clear difference in performance. Both are subjected to the same environmental disturbances.

There is only a small change in the integrated absolute error as depicted in Table 3.3.

Table 3.3: Summary of using pre-stored heading dependent matrices versus dynamically computing them at every timestep. Here LQ optimal control has been applied on the supply model to generate these results. The elapsed time is the average of running the same simulation a 100 times.

Description	Dynamic	Prestored	Difference
Average elapsed time [s]	2.9103	0.066694	2.8436
IAE - North position	397.2414	397.2486	-0.0072
IAE - East position	276.4216	276.4206	0.0010
IAE - Heading	310.6607	310.6607	0.0001

## 3.2 MPC

This section will investigate three different formulations of the model predictive controller as discussed in Section 2.8. The performance of these simulations will be compared.

In all simulations, the vessel starts at the origin and points northwards. At time  $t = 100$  the vessel moves to position  $[10, 5, 26^\circ]$  and at  $t = 600$  to position  $[-5, -5, 225^\circ]$ . The simulations are run for a total of 1000 seconds. The timestep is  $\Delta t = 1.0$ . The prediction horizon covers a period of 20 samples (20 seconds).

The control structure for MPC is shown in Figure 3.12. This is the same structure used for the NMPC.

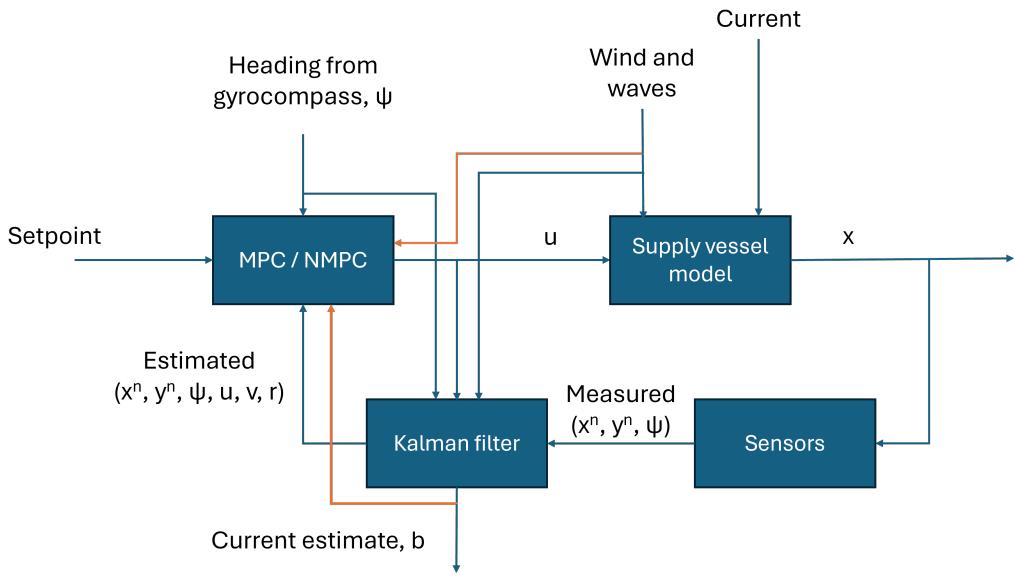


Figure 3.12: Plot depicts the control structure for the MPC/NMPC controller when applied to the supply model. Notice that the Kalman filter estimates the current estimate which is then fed to the controller. Wind and wave disturbances are also sent to the controller. The orange lines mark the difference between the structure used for MPC and the one used for LQ optimal control.

### 3.2.1 Comparison of MPC formulations

In the following, we will compare three different MPC formulations. They are subjected to the same environmental disturbances and setpoint changes. The wind, current and waves used in the simulations are shown in Figures 3.13, 3.14 and 3.15, respectively.

The corresponding path, setpoint-tracking, and input plots are depicted in Figures 3.16, 3.17 and 3.18.

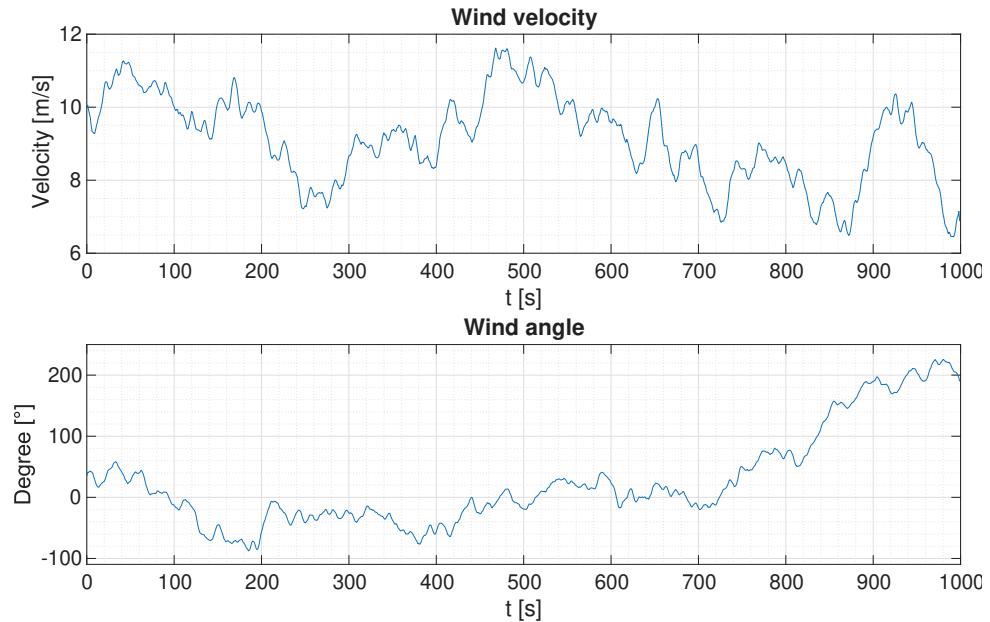


Figure 3.13: Plot depicts the wind disturbance used when simulating dynamic positioning using MPC controller on supply model. It depicts the wind speed and angle.

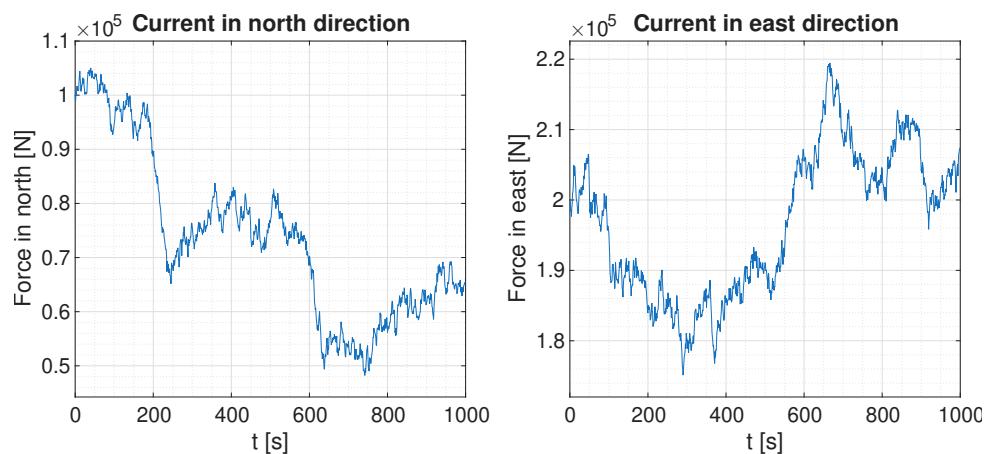


Figure 3.14: Plot shows current forces in north and east direction when simulating dynamic positioning using MPC controller on supply model.

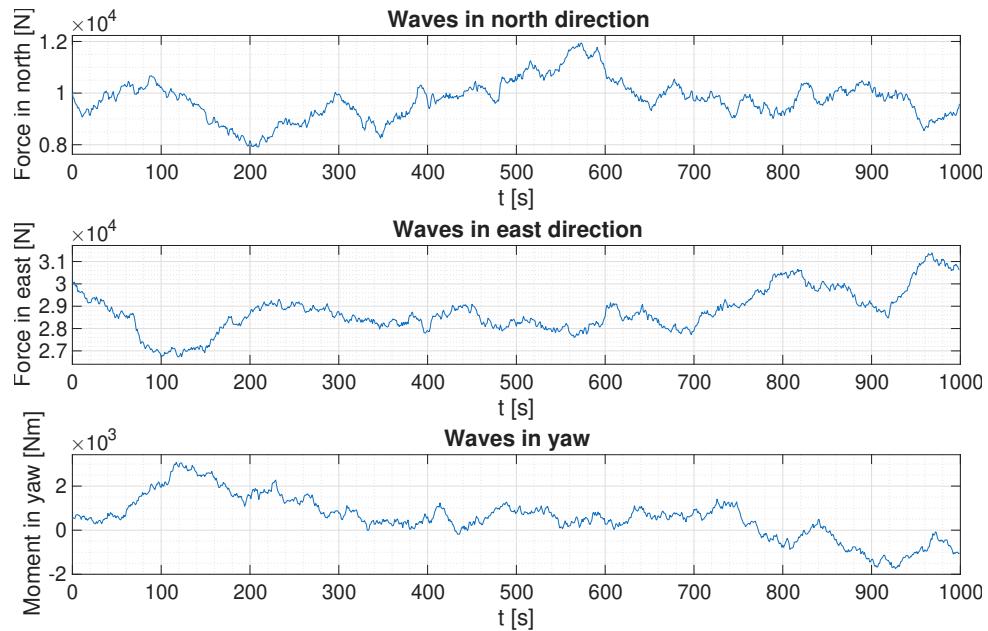


Figure 3.15: Plot depicts wave forces in north and east direction, and moment in yaw. Used when simulating dynamic positioning using MPC controller on supply model.

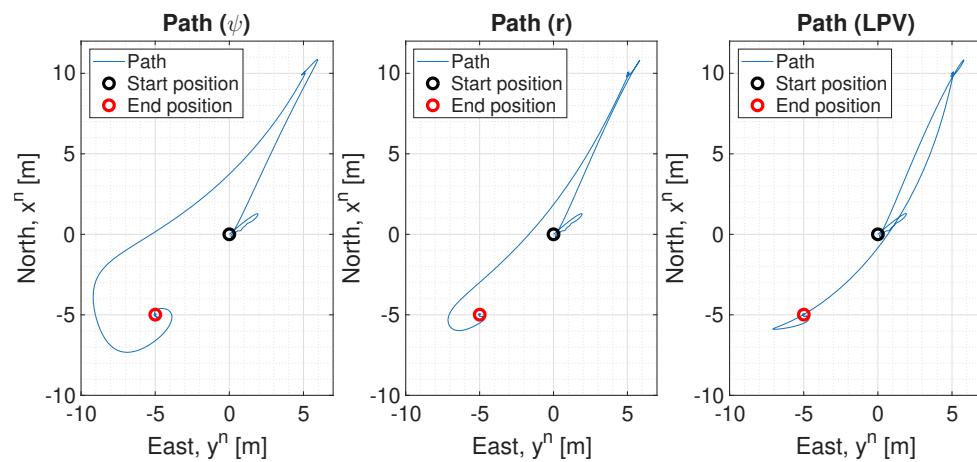


Figure 3.16: Depicts the ship movement in the NED geographic coordinate frame. The ship starts in the origin pointing north, then is subjected to a setpoint change at  $t = 100$  to position  $[10, 5, 26^\circ]$  and at  $t = 600$  to position  $[-5, -5, 225^\circ]$ .

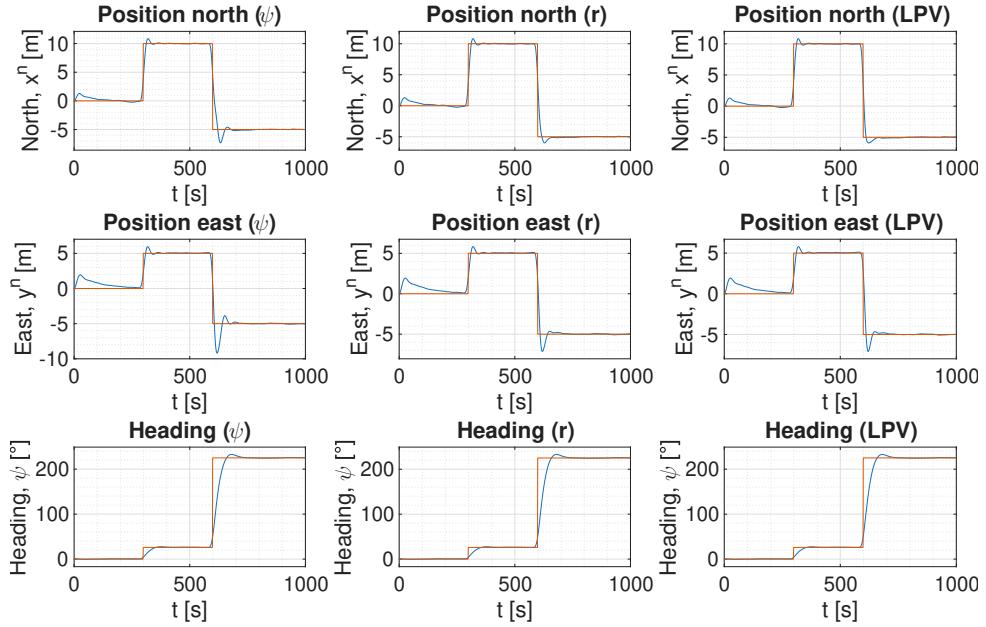


Figure 3.17: Depicts setpoint tracking for the different MPC formulations subjected to the same external disturbances. Here the orange line represents the setpoint, the blue line is the actual value.

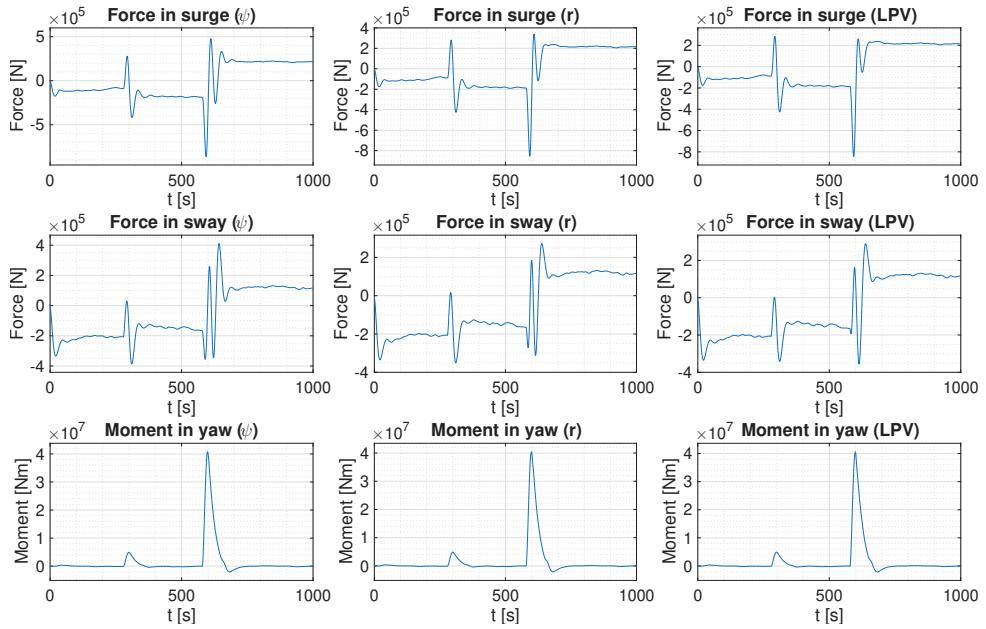


Figure 3.18: Inputs computed by the different MPC formulations. These inputs represent net force in surge/sway and moment in yaw.

The integrated absolute error and total value (TV) for each simulation are shown in Table 3.4. Since all of the different MPC formulations use the same weights, these models are directly comparable in performance. The model assuming constant heading through the prediction horizon is showing the worst performance with respect to the IAE metric. This is also clear from the path plots. This is not surprising, since we know that the vessel will experience heading changes, either due to environmental disturbances or new heading setpoints. The constant rotation formulation and LPV perform considerably better. The LPV MPC and the constant rotation formulation have a similar performance when looking at the integrated absolute error, but it should be noted that the LPV has a much higher computational cost. When it comes to thruster force usage, there are no notable differences between the models.

Table 3.4: Summary of simulation results for different version of model predictive control formulations.

Description	Const $\psi$	Const r	LPV
IAE - North position	352.45	296.21	282.82
IAE - East position	399.75	347.07	350.55
IAE - Heading	93.28	93.08	93.03
TV - North position	$1.85 \cdot 10^8$	$1.81 \cdot 10^8$	$1.80 \cdot 10^8$
TV - East position	$1.65 \cdot 10^8$	$1.61 \cdot 10^8$	$1.61 \cdot 10^8$
TV - Heading	$17.24 \cdot 10^8$	$17.24 \cdot 10^8$	$17.20 \cdot 10^8$

It is interesting to note that there is an initial offset during the start of the MPC run. It can be shown that this does not occur when running without environmental disturbances. This is simply due to the fact that the Kalman filter estimates the  $b$  term (integrator) starting from an initial value of zero. Figure 3.19 shows the previous simulation using the constant rotation MPC formulation. It is clear that the setpoint error reduces as the Kalman filter closes in on the true value of  $b$ . This makes sense since this term is involved in the model equations in the MPC formulation. If it starts from a value close to the true value, the setpoint error is significantly reduced in the early stages of the simulation, as can be seen in Figure 3.20.

So what does this mean in practice? If the Kalman filter runs in the background when performing other tasks than dynamic positioning, there should already be a reasonable estimate of  $b$  when the vessel enters DP mode. However, if for some reason this history was lacking when engaging in DP mode, the vessel would experience some initial struggle to maintain the setpoint. This can be partially offset by more aggressive tuning.

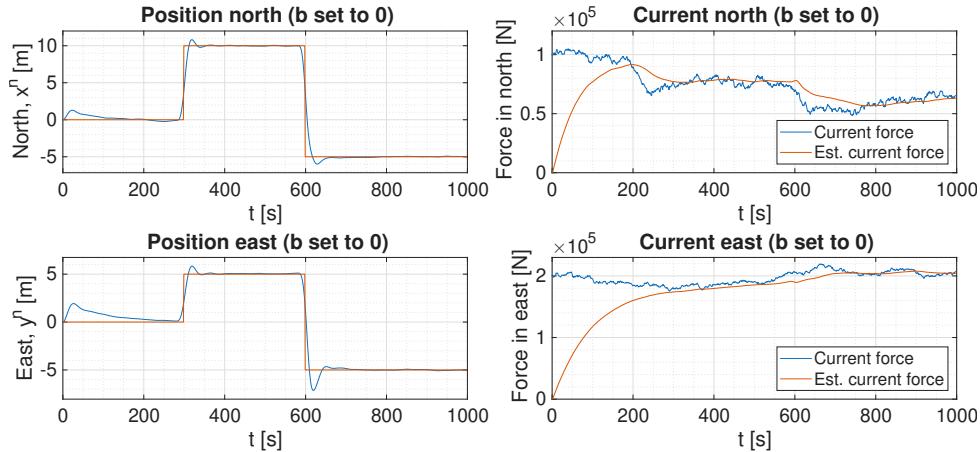


Figure 3.19: Setpoint tracking using the constant rotation formulation of MPC. The initial  $b$  estimate in the Kalman filter is set to zero ( $[0, 0, 0]$ ). It is clear that as the  $b$  estimate is improved, the setpoint tracking improves.

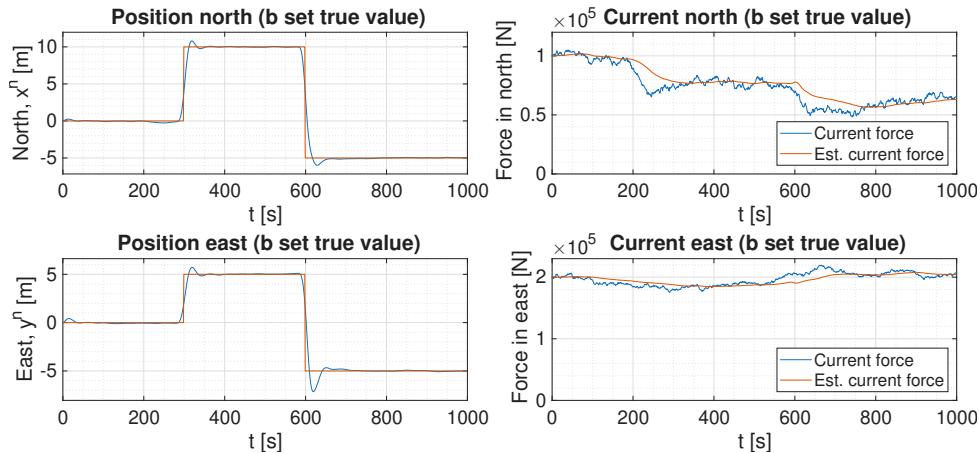


Figure 3.20: Setpoint tracking using the constant rotation formulation of MPC. The initial  $b$  estimate in the Kalman filter is set to  $([1 \cdot 10^5, 2 \cdot 10^5, 0])$ . These initial values clearly improves the early performance of the simulation, compared to starting with zero values for  $b$ .

### 3.2.2 Monte Carlo simulation

To understand how the different variants of MPC controllers compare, Monte Carlo simulations were run with varying disturbances and setpoint changes as discussed in Section 2.14.1. The RMSE and IAE values for individual simulations are depicted in Figure 3.21. The mean values of RMSE, IAE and TV are shown in Table 3.5. These results mirror what was observed earlier. The constant heading formulation performs the worst with respect to IAE, while the constant rotation formulation and LPV MPC perform about the same. No significant difference in total value (TV) is observed.

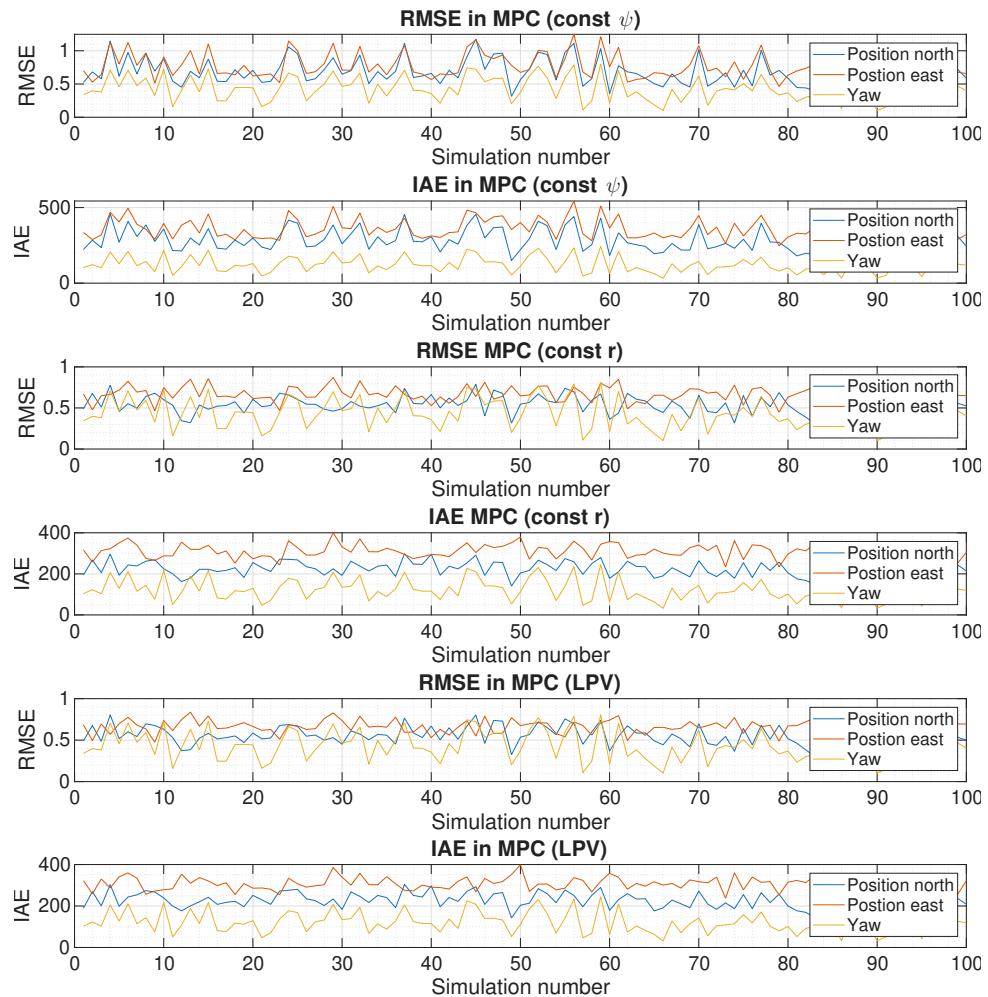


Figure 3.21: Shows results from Monte Carlo simulations of the supply model subjected to different MPC controllers. Depicts RMSE and IAE for every simulation.

Table 3.5: Table depicts the mean values of RMSE, IAE and TV from the Monte Carlo simulation using different MPC formulations.

Metric	Description	Constant heading	Constant rotation	LPV
RMSE	North position	0.68	0.54	0.55
	East position	0.77	0.68	0.67
	Heading	0.43	0.43	0.43
IAE	North position	283.44	222.87	225.72
	East position	358.88	312.20	308.99
	Heading	120.54	121.06	120.41
<b>Total</b>	-	762.86	656.13	655.13
TV	North position	$1.66 \cdot 10^8$	$1.64 \cdot 10^8$	$1.64 \cdot 10^8$
	East position	$1.83 \cdot 10^8$	$1.78 \cdot 10^8$	$1.78 \cdot 10^8$
	Heading	$22.63 \cdot 10^8$	$22.52 \cdot 10^8$	$22.62 \cdot 10^8$
<b>Total</b>	-	$2.61 \cdot 10^9$	$2.59 \cdot 10^9$	$2.60 \cdot 10^9$

### 3.3 NMPC

This section will show the results of running non-linear model predictive control on the supply model. The algorithm was outlined in Section 2.9.1. The horizon length is set to 20 samples, which span a total of 20 seconds into the future (the timestep is  $\Delta t = 1.0$ ). To reduce the computational burden, a grouping strategy was applied; the horizon was subdivided into three groups of length 2, 8 and 10. This implies that,

$$u_0 = u_1 \quad (3.1)$$

$$u_2 = u_3 = u_4 = u_5 = u_6 = u_7 = u_8 = u_9 \quad (3.2)$$

$$u_{10} = u_{11} = u_{12} = u_{13} = u_{14} = u_{15} = u_{16} = u_{17} = u_{18} = u_{19} \quad (3.3)$$

The vessel starts in  $\eta = [0, 0, 0]^\top$ . At time  $t = 100$  seconds the setpoint is changed to  $\eta = [-5, 10, -26]^\top$  and another change occurs at time  $t = 500$ ,  $\eta = [5, -5, 135]^\top$ . The entire simulation lasts for a total of 800 seconds.

The vessel is subjected to external forces as depicted in Figures 3.22, 3.23 and 3.24.

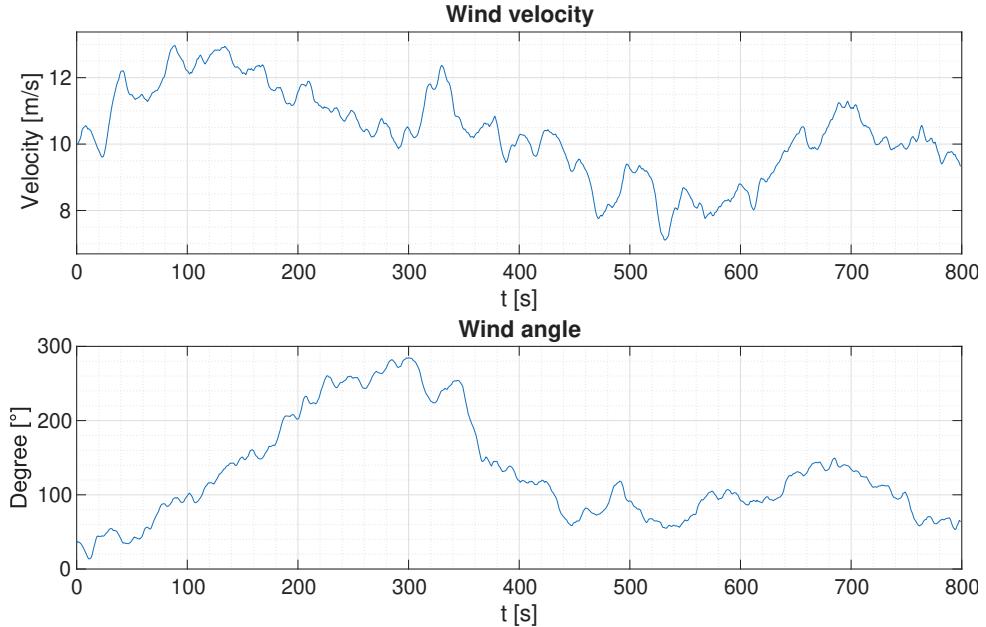


Figure 3.22: Plot depicts the wind disturbance applied when simulating the supply model using an NMPC controller.

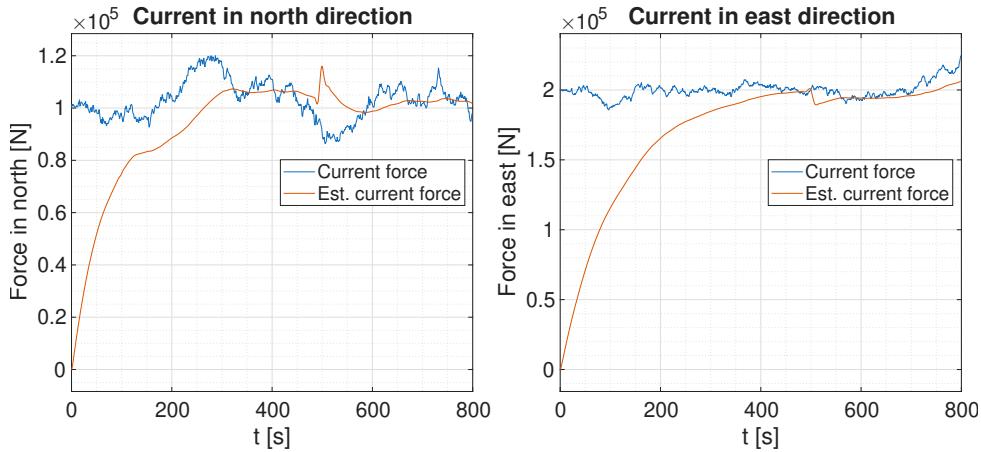


Figure 3.23: Plot shows current forces in north and east direction applied during the NMPC simulation. It also includes the current force estimates from the Kalman filter.

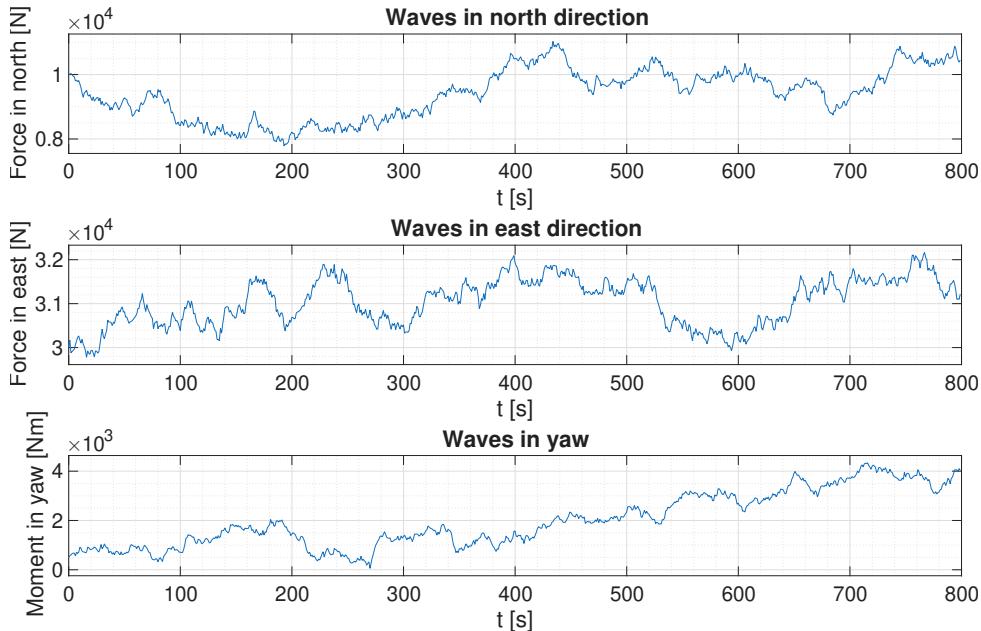


Figure 3.24: Plot depicts wave forces in north and east direction applied during the NMPC simulation. It also shows the moment in yaw generated due to waves.

The path, setpoint tracking, and inputs plots are shown in Figures 3.25, 3.26 and 3.27, respectively.

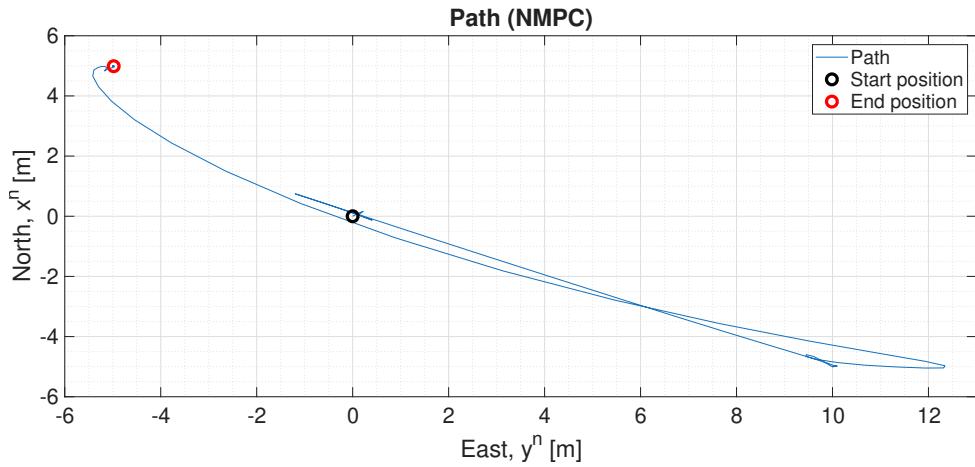


Figure 3.25: Depicts the ship movement in the NED coordinate frame. There are two setpoint changes occurring at time  $t = 100$  moving to position  $[-5, 10, -26^\circ]$  and at time  $t = 500$  moving to position  $[5, -5, 135^\circ]$ .

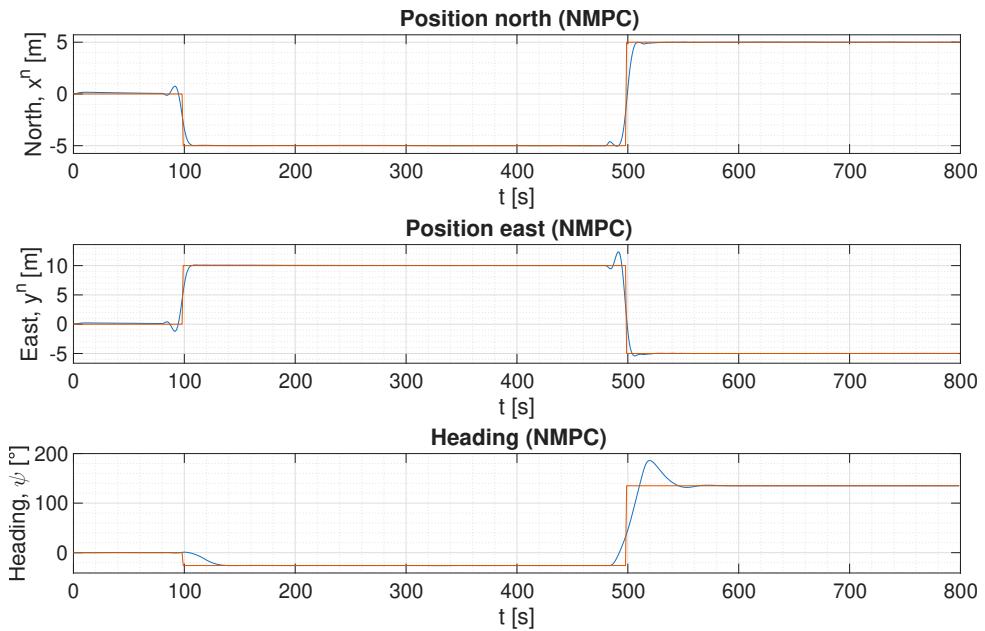


Figure 3.26: Depicts setpoint tracking when using NMPC to control supply vessel.

The integrated absolute error values and the total value are summarized in Table 3.6. We note that the IAE values are small in this simulation, and looking at the setpoint tracking plot it is clear that the tuning used in this case is more aggressive than in the MPC and LQ optimal control. The initial early struggle in setpoint tracking due to incorrect  $b$  value have much less effect due to this tuning.

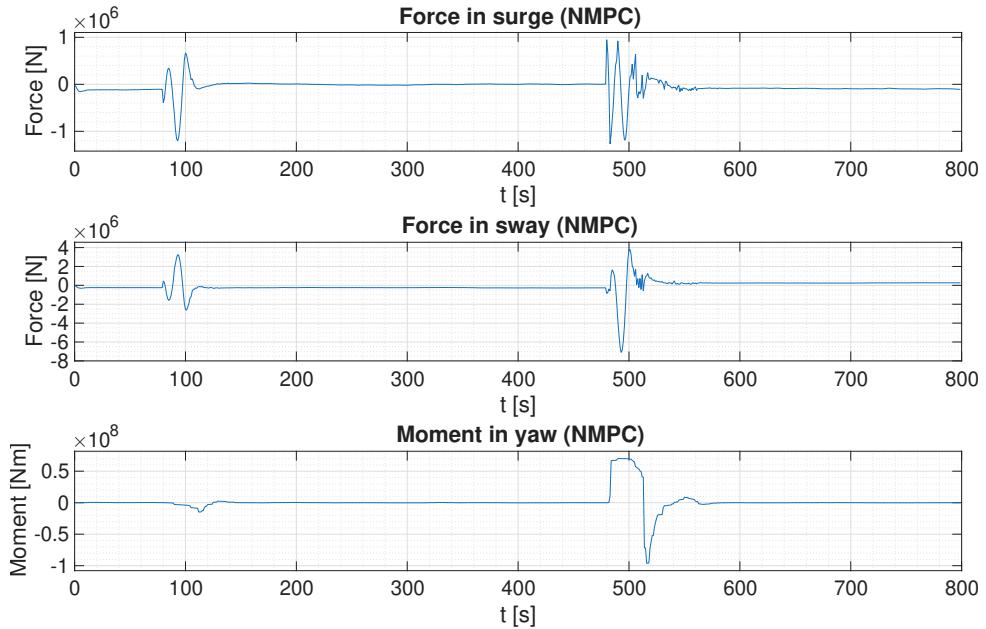


Figure 3.27: Shows the net forces and moment calculated by the NMPC.

Table 3.6: Summary of simulation results for NMPC.

Description	NMPC
IAE - North position	62.02
IAE - East position	100.09
IAE - Heading	43.98
TV - North position	$0.69 \cdot 10^8$
TV - East position	$3.09 \cdot 10^8$
TV - Heading	$32.82 \cdot 10^8$

### 3.3.1 Monte Carlo simulation

Monte Carlo simulations were run with varying disturbances and setpoint changes as discussed in Section 2.14.1. The RMSE and IAE values for individual simulations are depicted in Figure 3.28. The mean values of RMSE, IAE and TV are shown in Table 3.7. When comparing these values with the corresponding values obtained from LQ optimal control (see Table 3.2) and MPC (see Table 3.5), it is clear that this more aggressive tuning results in better IAE values, but at the cost of more force and moment generation. It is questionable whether this improvement in IAE is worth the added cost of fuel and potential wear and tear on the thrusters.

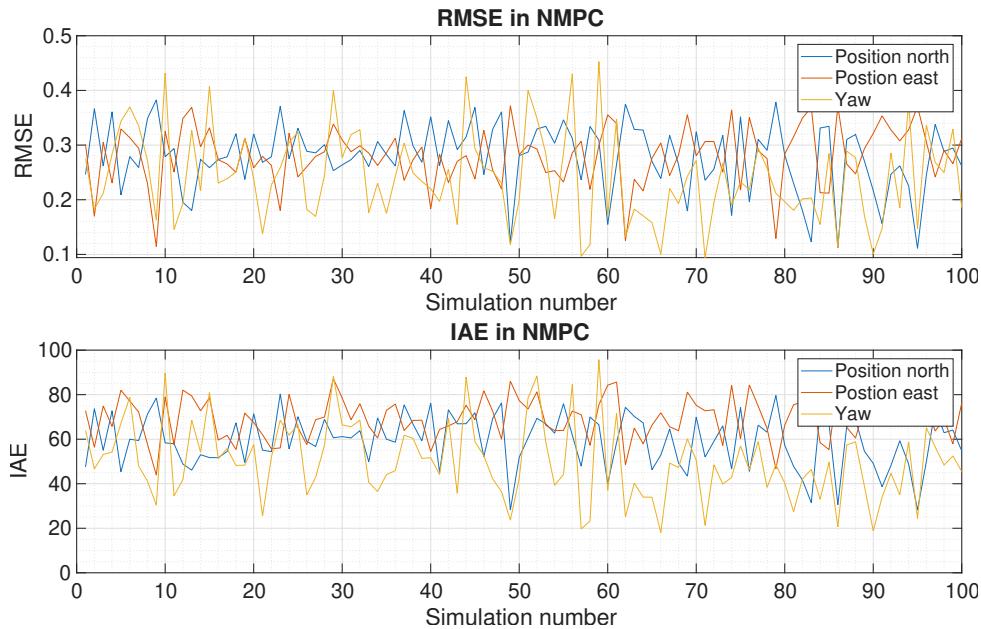


Figure 3.28: Depiction of individual RMSE and IAE values when running Monte Carlo simulations of the supply model using a non-linear model predictive controller (NMPC).

Table 3.7: Table depicts the mean values of RMSE, IAE and TV from the Monte Carlo simulations.

Description	NMPC
RMSE - North position	0.28
RMSE - East position	0.28
RMSE - Heading	0.24
IAE - North position	59.24
IAE - East position	69.62
IAE - Heading	49.98
<b>Total</b>	<b>178.83</b>
TV - North position	$1.89 \cdot 10^8$
TV - East position	$2.31 \cdot 10^8$
TV - Heading	$36.62 \cdot 10^8$
<b>Total</b>	<b><math>4.08 \cdot 10^9</math></b>

## 3.4 System Identification

This section applies system identification to identify the BODY dynamics. The model is then extended into the geographical NED frame by applying the known relation between vessel heading and north-east position. The details are outlined in Section 2.10.4. The vessel setpoints are tracked by applying linear quadratic optimal control. Both the controller and Kalman filter use the new identified model when controlling the original vessel model. A simple method for model reduction will also be discussed, reducing the system to a three double integrator system.

### 3.4.1 Identifying vessel models

When generating data for system identification, the models are run for half an hour with a timestep of 0.3 seconds. The velocities in surge, sway, and yaw are recorded at each timestep together with the inputs (generated using PRBS).

All vessel control simulations are run for 1000 seconds with a timestep,  $\Delta t$ , set to 0.3 seconds. Changes in the setpoint occur at times  $t = 100$  and  $t = 600$ . The new setpoints are  $[10, 5, 30^\circ]$  and  $[-5, -5, 45^\circ]$ . Note that the time step,  $\Delta t$ , must correspond to the same time step used when identifying the models. This is because the identified discrete matrices are a function of the time step.

When simulating, no external disturbances will be applied. The reason for this is that the model interfaces to OSV in the MSS library only accepts current velocity and thruster RPS as input. We want to run all the models under the same conditions (both with regard to the setpoints and external disturbances).

Each of the vessel models (supply, OSV, and Balchen model) will be identified and discussed in this section. The identified mathematical models (see equation 2.126) will be tested with an LQ optimal controller on deviation form. The Kalman filter will use equation 2.131 which includes an integrator ( $b$  term) to estimate unknown disturbances and modeling errors.

These models were identified under conditions with very limited environmental disturbances; in other words, the dominant forces acting on the models are the forces generated from thrusters. If there are significant environmental disturbances, this information should be fed into the input  $u$ . In other words,  $u = \tau_{\text{thr}} + \tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{current}}$ . Identifying a mathematical model is much easier under conditions where the force and moment contribution from environmental conditions are insignificant. This is a limitation of the method outlined here.

The models are not directly comparable. They represent different vessels with different dynamics. The aim of this section is to demonstrate that it is possible to control the position and attitude of a vessel based on data collection and system identification.

### 3.4.2 Supply model simulation

The identified mathematical model for the supply model (using DSR) is shown in equations (3.4) and (3.5), respectively<sup>2</sup>. The supply model interface in the MSS library, the function `supply(x, tau)`, allows us to specify the net forces and moment (`tau`). This model can then easily be simulated without external disturbances.

$$\tilde{A} = \left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \cos \psi & -\Delta t \sin \psi & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \sin \psi & \Delta t \cos \psi & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \\ \hline 0 & 0 & 0 & 0.9964 & 0.0008 & 0.0046 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0008 & 0.9925 & -0.0244 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0002 & -0.0007 & 0.9752 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -0.5678 & -0.1046 & -0.0017 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.1044 & 0.5674 & -0.0227 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0057 & 0.0218 & 0.5765 & 0 & 0 & 0 \end{array} \right] \quad (3.4)$$

$$\tilde{B} = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline -0.7516 & -0.0840 & -0.0006 \\ -0.1384 & 0.4562 & 0.0035 \\ -0.0022 & -0.0146 & 0.0010 \\ \hline 0.4427 & 0.0000 & 0.0000 \\ 0.0000 & 0.2698 & 0.0021 \\ 0.0000 & 0.0020 & 0.0007 \end{array} \right] \cdot 10^{-7} \quad (3.5)$$

The supply model simulation is depicted in Figures 3.29 and 3.30, showing the setpoint tracking and inputs.

To verify that the  $b$  term in the integrator is able to estimate the current when the supply vessel is subjected to disturbances, a simulation was run using the identified DSR model

---

<sup>2</sup>The entries marked red in the matrix identifies the part of the state transition matrix found using system identification, likewise the blue entries represent the corresponding values for the input matrix. This convention will be used every time these matrices are shown. Only DSR matrices will be depicted here, but simulation results from all identification methods will be summarized in a table at the end of this section (see Table 3.8).

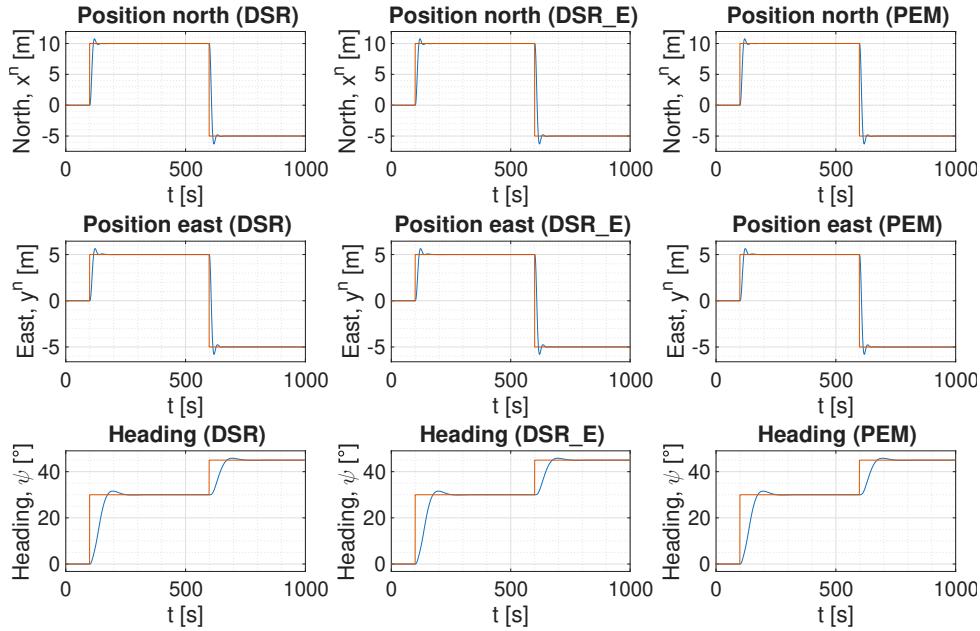


Figure 3.29: Shows setpoint tracking using the mathematical model generated from system identification of the supply model. Here the LQ optimal controller is used. Depicts the results for different SI methods.

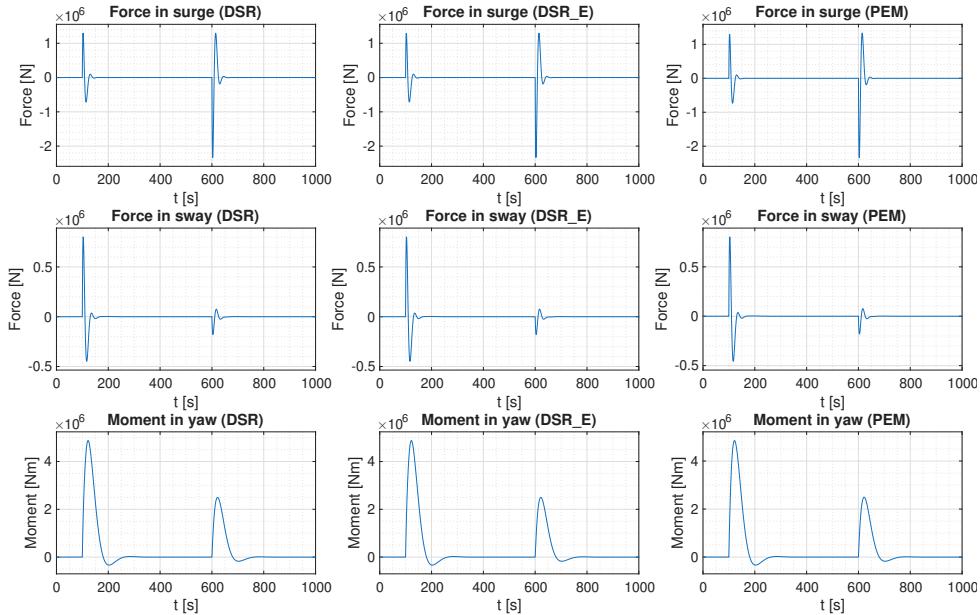


Figure 3.30: Inputs computed by the LQ optimal control algorithm. These inputs are total forces in surge and sway, and moment in yaw.

together with external disturbances. The DP current estimate (*b*) is depicted in Figure 3.31.

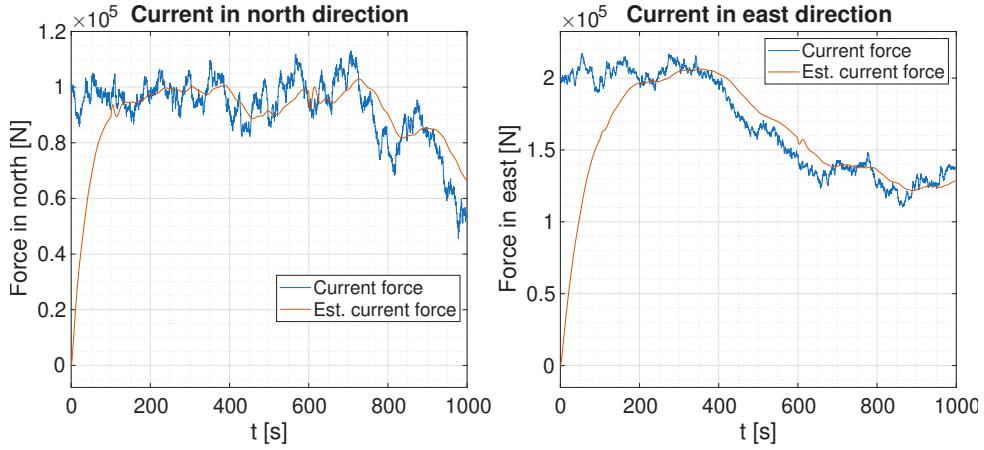


Figure 3.31: Shows the current estimate from the Kalman filter. The mathematical model used in both the LQ optimal controller and in the Kalman filter is based on DSR system identification of the supply model. Note that the estimate nicely follows the real current.

### 3.4.3 OSV model simulation

The identified mathematical model for OSV is given in (3.6) and (3.7). The interface to the OSV function in MSS, `osv(x, ui, Vc, betaVc)` takes current velocity ( $V_c$ ) and angle ( $\beta V_c$ ) as input parameters. These two current values are set to zero when collecting data for system identification, but since the vessel is moving during this process, there will be a small current force (since this force depends on the relative velocity between the vessel and the current). Hence, thrusters alone will not be the only source of forces. Of course, the mathematical model does not need to perfectly represent the system in question, just be good enough to capture the dynamics needed for control purposes. Note that the inputs to this model are the RPS signals to the thrusters, not generalized forces and moment. It is known (from the OSV code documentation) that the individual thruster forces are proportional to the square of the RPS signal, i.e.  $f \propto u_{RPS}|u_{RPS}|$ .

$$\tilde{A} = \left[ \begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \cos \psi & -\Delta t \sin \psi & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \sin \psi & \Delta t \cos \psi & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \\ \hline 0 & 0 & 0 & 0.9985 & -0.0004 & -0.1980 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 0.9944 & 0.9660 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 0.0001 & 0.8312 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -0.5773 & 0.0094 & 0.8312 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0094 & -0.5772 & -0.0497 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 0.0001 & 0.3075 & 0 & 0 & 0 \end{array} \right] \quad (3.6)$$

$$\tilde{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0.0065 & -0.1633 & -0.1405 \\ -0.1424 & 0.0107 & -0.0054 \\ 0.0088 & -0.0018 & 0.0016 \\ -0.0062 & 0.0947 & 0.0810 \\ 0.0880 & -0.0058 & 0.0054 \\ 0.0033 & -0.0007 & 0.0006 \end{bmatrix} \cdot 10^{-5} \quad (3.7)$$

The results of simulations performed on the identified OSV model using the LQ optimal controller are depicted in Figures 3.32 and 3.33. These figures show setpoint tracking and inputs. The net forces have been calculated using the relation  $\tau = Tf$ .

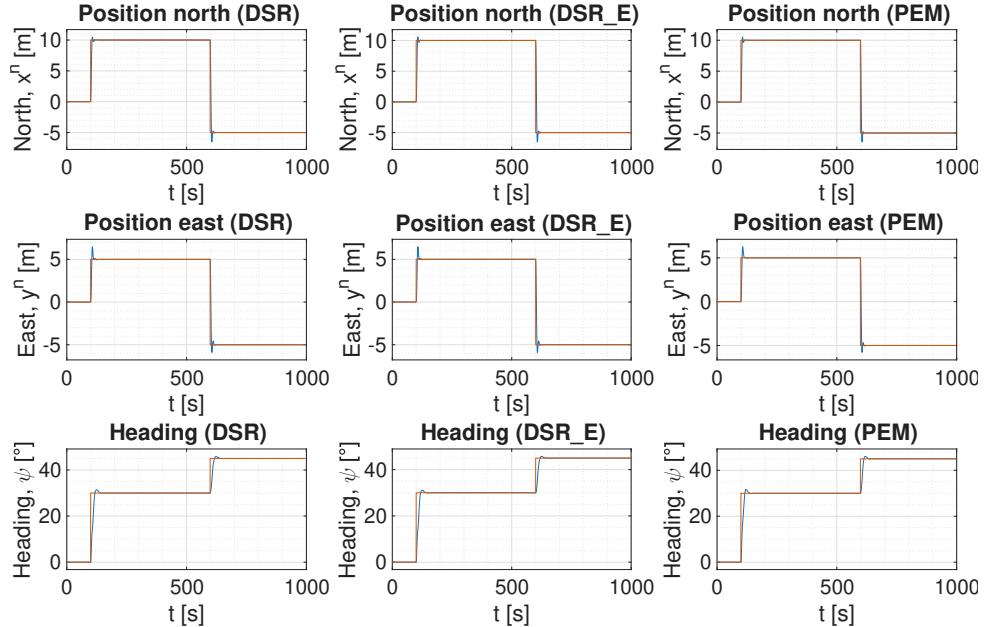


Figure 3.32: Shows setpoint tracking using the mathematical model generated from the system identification of the OSV model. Depicts the results for different SI methods.

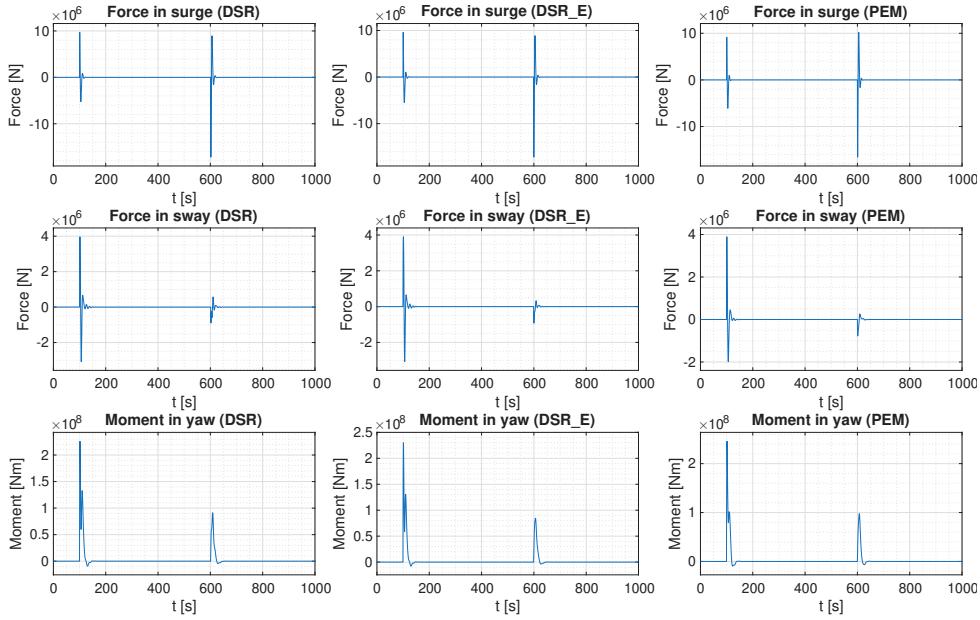


Figure 3.33: Shows inputs to the OSV model. The inputs, computed by the LQ optimal control algorithm, have been converted to forces in surge and sway, and moment in yaw.

This particular tuning seems very aggressive. The reason for this is that the input variable was chosen as  $u = |u_{RPS}|u_{RPS}$ . This value  $u$  is related to the generalized forces through the following relation:

$$\begin{aligned} \tau = TKu &= \underbrace{\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 37 & -7 & 7 \end{bmatrix}}_T \underbrace{\begin{bmatrix} 15.3061 & 0 & 0 \\ 0 & 18.6667 & 0 \\ 0 & 0 & 16.3750 \end{bmatrix}}_K u \\ &= \underbrace{\begin{bmatrix} 0 & 18.6667 & 16.3750 \\ 15.3061 & 0 & 0 \\ 566.3265 & -130.6667 & 114.6250 \end{bmatrix}}_R u \end{aligned} \quad (3.8)$$

This indicates that if there is no change in tuning parameters in the LQ optimal controllers for the vessel models run here (as is the case), the OSV controller is likely to be the most aggressive since one unit of input value will here represent many Newtons of force (or Newton-meters of moment). In this case, the matrix  $R$  is invertible, but this is not generally the case depending on the number of thrusters. Here,  $\tau$  could be used as input directly in the system identification algorithm. However, this is generally not the case. The link between thrusters output and controller demand usually have to be bridged by a thruster allocation algorithm.

### 3.4.4 Balchen model simulation

The identified model matrices (DSR) for Balchen are shown in equations (3.9) and (3.10). From the model equations (see Section 2.7.3) it is clear that the wind can easily be removed, however, the current force acting on the vessel is due to the relative velocity of the current. If we assume the current to be zero, it will still be a current force when the vessel moves (analogously to the discussion of OSV). It is possible to run the simulation so that the current speed always matches the vessel speed, but this is not something that can be done in real-world applications and will not be pursued here.

$$\tilde{A} = \left[ \begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \cos \psi & -\Delta t \sin \psi & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \Delta t \sin \psi & \Delta t \cos \psi & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \Delta t \\ \hline 0 & 0 & 0 & 1.0000 & 0.0000 & 0.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 1.0000 & 0.0000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000 & 0.0000 & 1.0000 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & -0.5754 & 0.0480 & -0.0009 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0480 & 0.5753 & -0.0029 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.0007 & 0.0029 & 0.5773 & 0 & 0 & 0 \end{array} \right] \quad (3.9)$$

$$\tilde{B} = \left[ \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline -0.1295 & 0.0011 & 0.0000 \\ 0.0108 & 0.0129 & 0.0000 \\ -0.0002 & -0.0001 & 0.0000 \\ \hline 0.0750 & 0.0000 & 0.0000 \\ 0.0000 & 0.0075 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 \end{array} \right] \cdot 10^{-6} \quad (3.10)$$

The Balchen model simulation is shown in Figures 3.34 and 3.35, depicting the setpoint tracking and inputs.

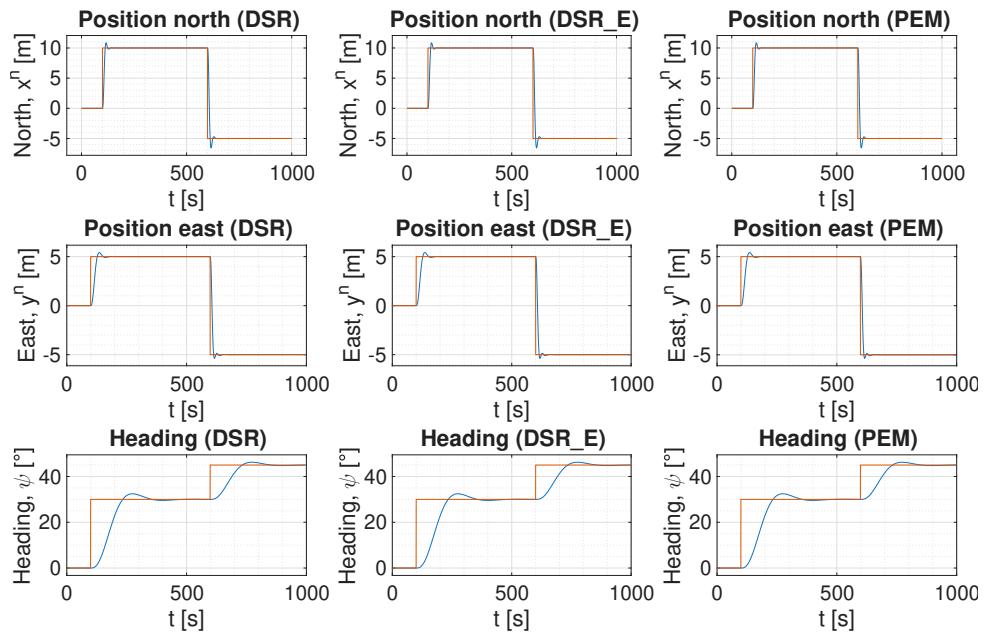


Figure 3.34: Shows setpoint tracking using the mathematical model generated from system identification of the Balchen model. Depicts the results for different SI methods.

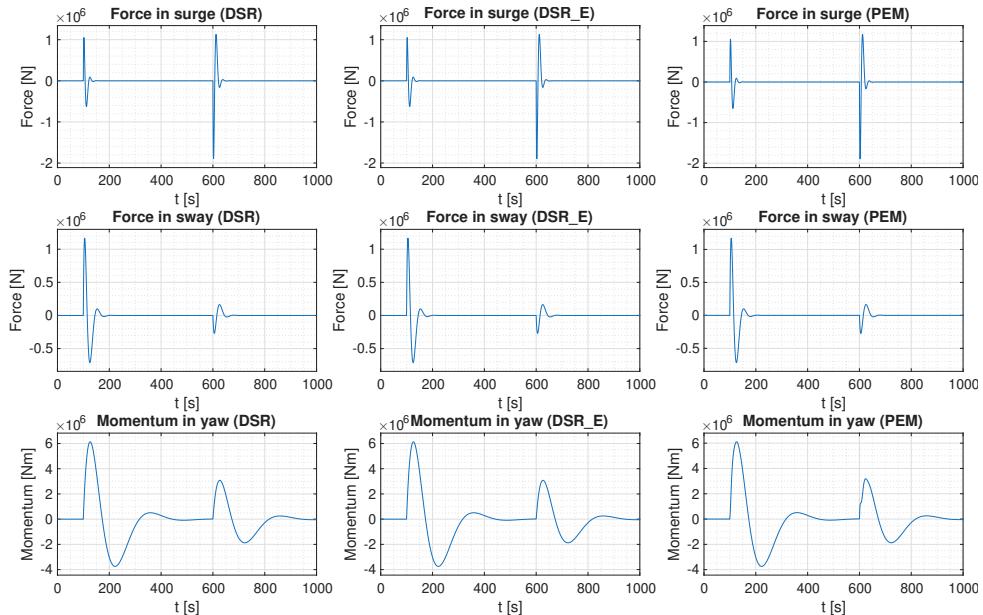


Figure 3.35: Inputs computed by the LQ optimal control algorithm. These inputs are total forces in surge and sway, and moment in yaw.

### 3.4.5 Comparison of SI models

The path plot for all the simulations are depicted in 3.36. From this it appears that the Balchen model has the most trouble reaching new setpoints. This is also reflected by the IAE metric in the table below.

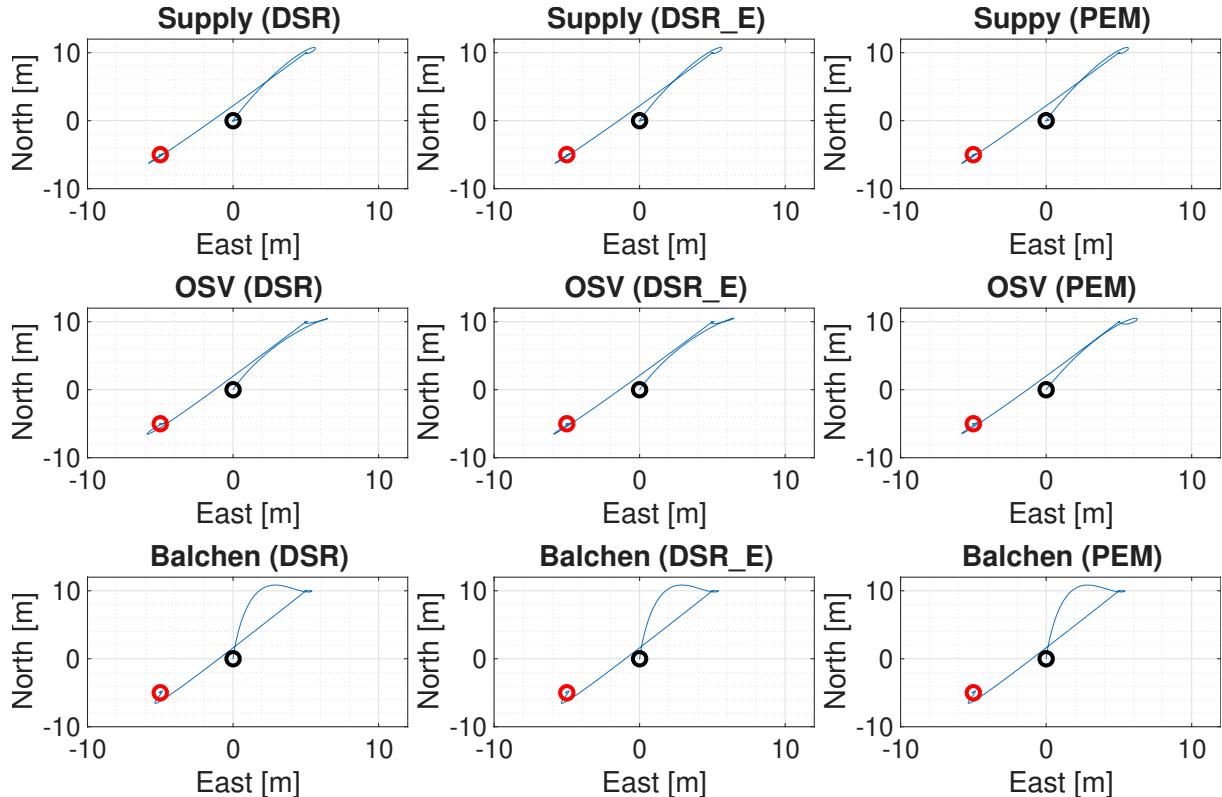


Figure 3.36: Path plot for all vessels and identification methods. Black circles mark the starting point and the red circles mark the end position.

The integrated absolute error for each simulation is given in Table 3.8. We note that the OSV has the best integrated absolute error values. This is likely due to the choice of input variables in the system identification process, which effectively makes the tuning of the system more aggressive. It should also be noted that all these vessel models have their own dynamics and will react differently to the input from the controller. The total values (TV) for the different simulations are given in Table 3.9. These values confirm that the OSV model is the most aggressive.

Only model matrices generated by DSR were shown in this report, but the identified models have different matrix entries for the same vessel models depending on the identification method. However, judging from the path plot and the IAE table, it is clear that they generate models that are close in performance. The difference in matrix entries is not really indicative of actual model difference, since any state space model can be

Table 3.8: Summary of simulation results. It depicts the integrated absolute error (IAE) for different vessel models and system identification methods.

<b>SI</b>	<b>Description</b>	$\Delta t$	<b>Supply</b>	<b>OSV</b>	<b>Balchen</b>
DSR	IAE - North position	0.3s	239.57	84.43	204.32
	IAE - East position	0.3s	156.32	59.08	168.54
	IAE - Heading	0.3s	32.05	7.11	65.33
DSR_e	IAE - North position	0.3s	239.57	84.96	204.32
	IAE - East position	0.3s	156.32	58.47	168.54
	IAE - Heading	0.3s	32.05	7.03	65.33
PEM	IAE - North position	0.3s	238.93	83.29	203.42
	IAE - East position	0.3s	156.00	56.88	168.19
	IAE - Heading	0.3s	32.05	6.43	65.34

Table 3.9: Summary of simulation results. It depicts the TV for the different vessel models and system identification methods.

<b>SI</b>	<b>Description</b>	$\Delta t$	<b>Supply</b>	<b>OSV</b>	<b>Balchen</b>
DSR	TV - North position	0.3s	$0.42 \cdot 10^8$	$0.37 \cdot 10^9$	$0.30 \cdot 10^8$
	TV - East position	0.3s	$0.13 \cdot 10^8$	$0.11 \cdot 10^9$	$0.31 \cdot 10^8$
	TV - Heading	0.3s	$3.45 \cdot 10^8$	$10.19 \cdot 10^9$	$9.55 \cdot 10^8$
DSR_e	TV - North position	0.3s	$0.42 \cdot 10^8$	$0.37 \cdot 10^9$	$0.30 \cdot 10^8$
	TV - East position	0.3s	$0.13 \cdot 10^8$	$0.11 \cdot 10^9$	$0.31 \cdot 10^8$
	TV - Heading	0.3s	$3.45 \cdot 10^8$	$10.00 \cdot 10^9$	$9.55 \cdot 10^8$
PEM	TV - North position	0.3s	$0.43 \cdot 10^8$	$0.38 \cdot 10^9$	$0.30 \cdot 10^8$
	TV - East position	0.3s	$0.13 \cdot 10^8$	$0.09 \cdot 10^9$	$0.31 \cdot 10^8$
	TV - Heading	0.3s	$3.45 \cdot 10^8$	$10.45 \cdot 10^9$	$9.55 \cdot 10^8$

transformed into an infinite number of other realizations of the model matrices ( $A$ ,  $B$ ,  $C$  and  $D$ ) while preserving the input-output relation (due to similarity transformation). However, the eigenvalues will not change. The eigenvalues for the different models are shown in Table 3.10.

Table 3.10: Eigenvalues of state transition matrix obtained from system identification of the supply, OSV and Balchen model.

Model	SI method	Eigenvalues								
		0	0	0	1	1	1	0.9966	0.9742	0.9933
Supply	DSR	0	0	0	1	1	1	0.9966	0.9742	0.9933
	DSR_e	0	0	0	1	1	1	0.9966	0.9742	0.9933
	PEM	0	0	0	1	1	1	0.9966	0.9742	0.9933
OSV	DSR	0	0	0	1	1	1	0.9985	0.8304	0.9951
	DSR_e	0	0	0	1	1	1	0.9988	0.8226	0.9948
	PEM	0	0	0	1	1	1	0.9979	0.7365	0.9952
Balchen	DSR	0	0	0	1	1	1	1	1	1
	DSR_e	0	0	0	1	1	1	1	1	1
	PEM	0	0	0	1	1	1	1	1	1

All three identification methods appear to identify very similar models in the case of the supply model. This is a linear model in the BODY dynamics, and hence it is not unexpected that all three system identification methods would perform well. The OSV model is non-linear and results in differing eigenvalues. However, the Balchen model is effectively a three double integrator model according to the eigenvalues identified. Since both the supply model and OSV have eigenvalues close to 1, it would be interesting to investigate whether these models could be reduced to three double-integrator models. This is exactly the topic of the next section.

### 3.4.6 Using double integrator model

The double integrator models were discussed in Section 2.10.8. Data from the supply model were collected for a period of 50 seconds with a timestep of 0.1 seconds. This data was used to identify three different reduced models. Three strategies were outlined in the theory chapter. The first strategy sets the transition matrix and the output matrix equal to the identity matrix (strategy 1). The only parameters that need to be identified are the entries in the input matrix,  $B$ . Figure 3.37 compares the data from the simulated supply model with the output from the new reduced state-space model. The same plots for strategy 2 and 3 are depicted in Figures 3.38 and 3.39, respectively.

Using LQ optimal control (on deviation form) with these new reduced models together with the supply model shows that only two of the models are stable. Only identifying the input matrix (assuming that  $A$  and  $C$  are identity matrices) results in unstable behavior. This seems reasonable since Figure 3.37 exhibits the worst performance compared to the other two model approaches. External disturbances were not used during these simulations. The simulations were run for 1400 seconds with a timestep of 0.1 seconds.

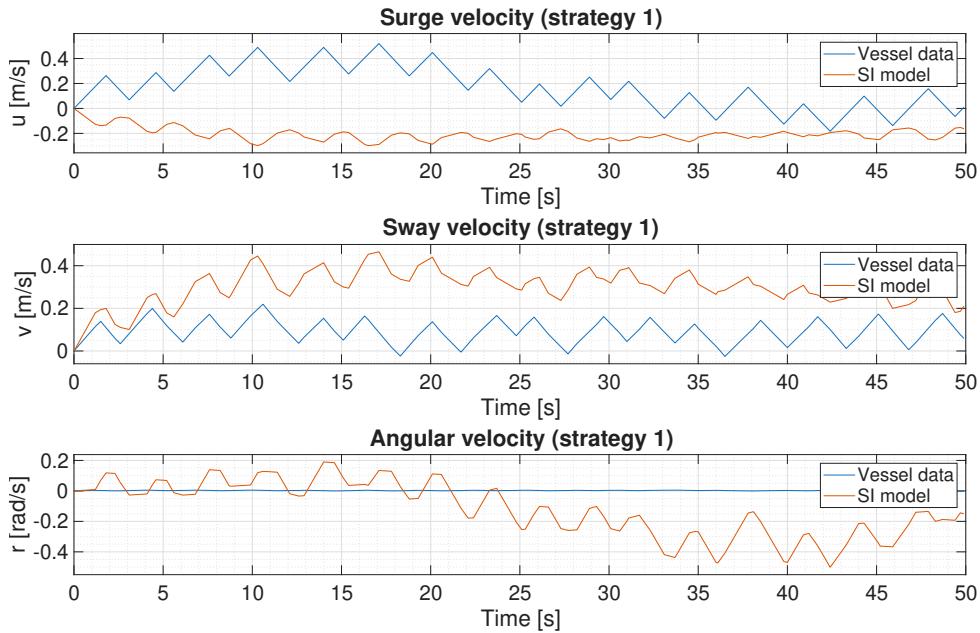


Figure 3.37: Compares the supply model output versus integrator model where the state transition matrix ( $A$ ) and output matrix ( $C$ ) are both identity matrices. The input matrix ( $B$ ) is estimated.

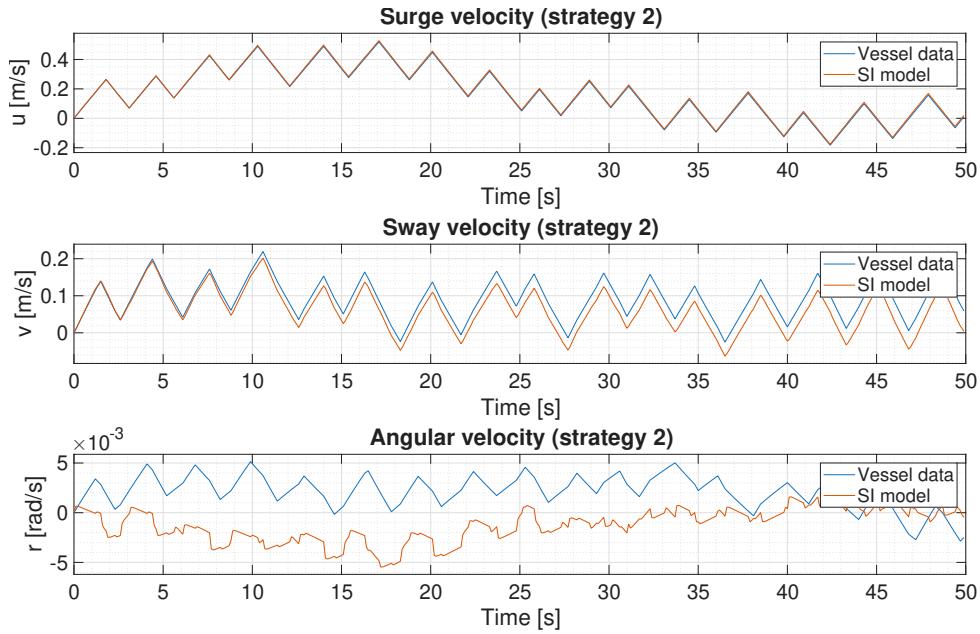


Figure 3.38: Compares the supply model output versus integrator model where the output matrix ( $C$ ) is an identity matrix. The input matrix ( $B$ ) and upper triangular entries of the state transition matrix ( $A$ ) are estimated.

The setpoint tracking and force inputs for the second strategy (identifying  $B$  and upper triangular entries of  $A$ ) are shown in Figures 3.40 and 3.41. The third approach (identifying

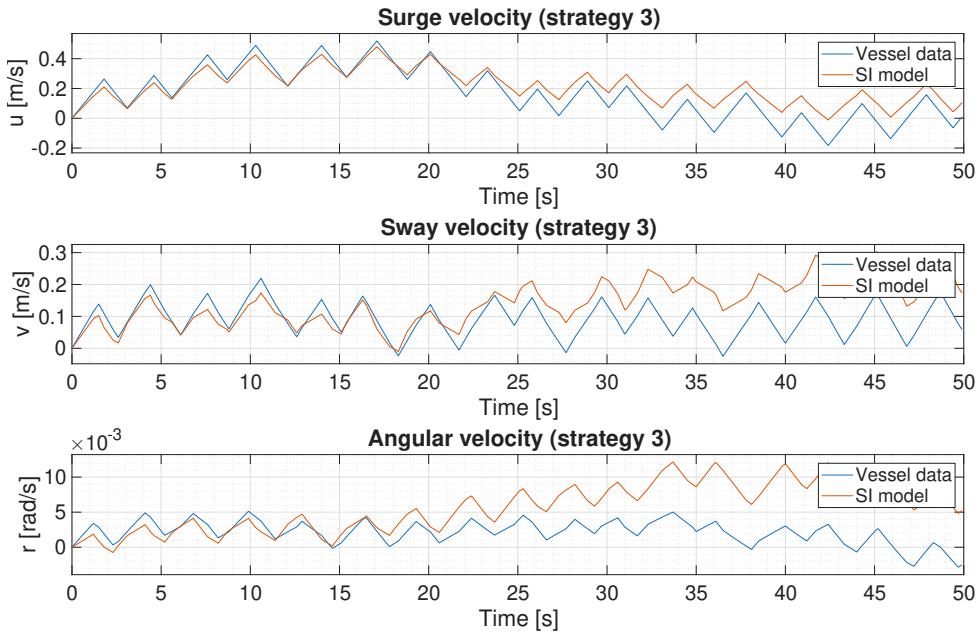


Figure 3.39: Compares the supply model output versus integrator model where the state transition matrix ( $A$ ) is an identity matrix. The input matrix ( $B$ ) and output matrix ( $C$ ) are estimated.

$B$  and  $C$  with  $A$  set to the identity matrix) is shown in Figures 3.42 and 3.43.

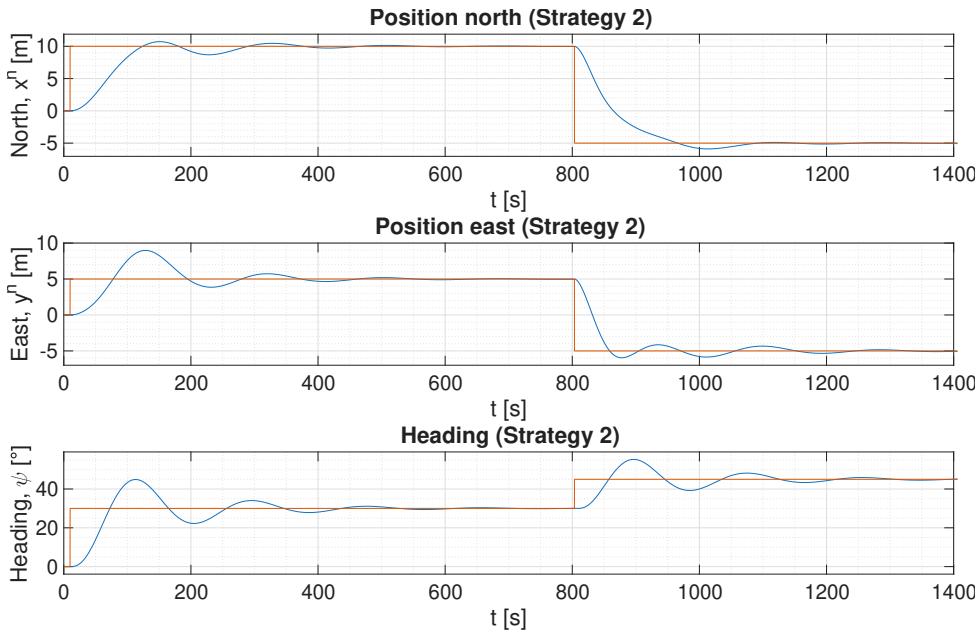


Figure 3.40: Shows setpoint tracking using mathematical model calculated using the second strategy for estimating a three double integrator model. Here the output matrix ( $C$ ) is an identity matrix. The input matrix ( $B$ ) and upper triangular entries of the state transition matrix ( $A$ ) are estimated.

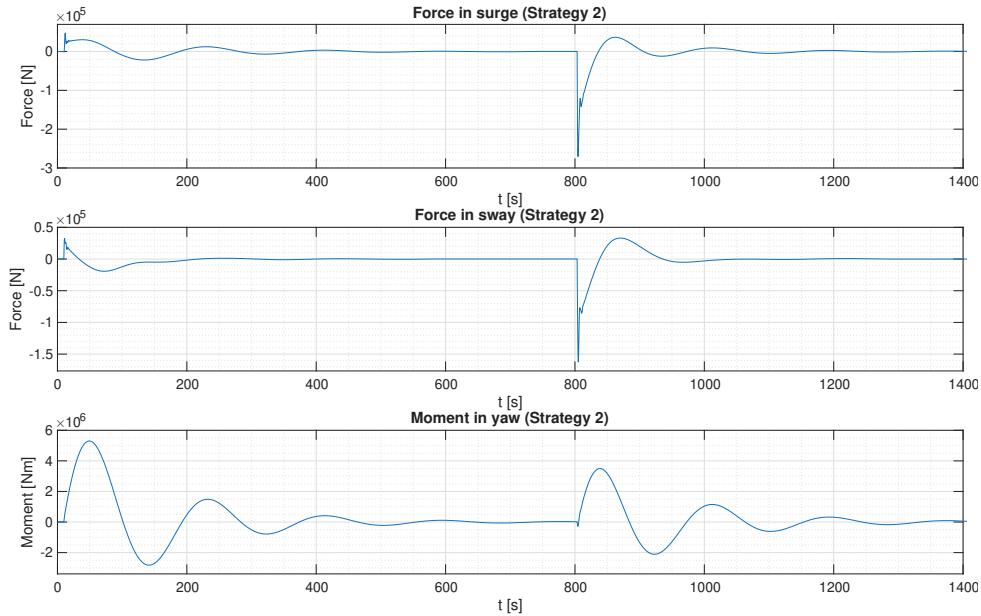


Figure 3.41: Inputs computed by LQ optimal control algorithm. Based on the second strategy for estimating model matrices, resulting in a three double integrators model.

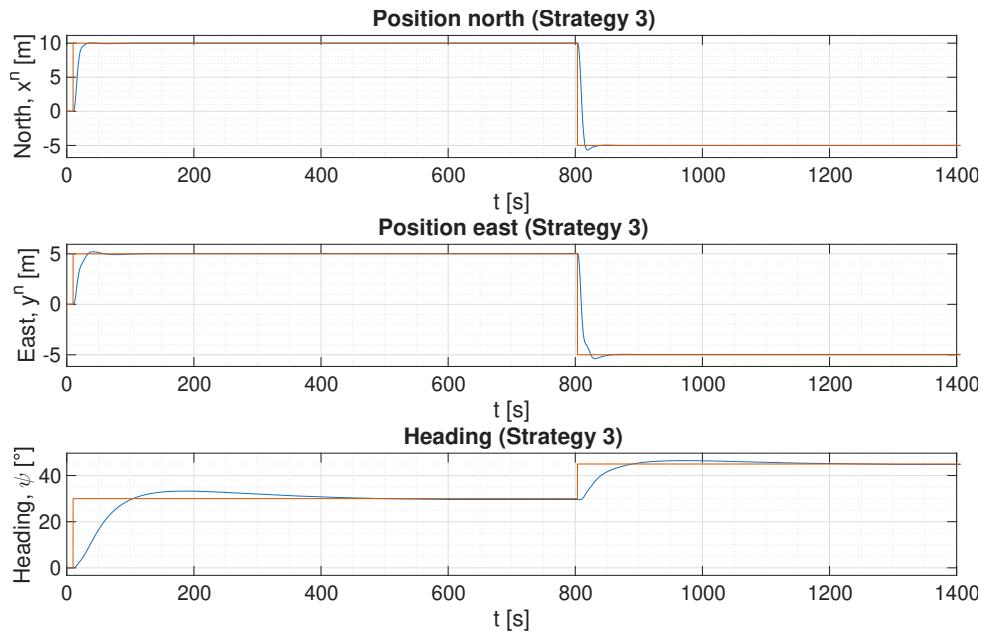


Figure 3.42: Shows setpoint tracking using mathematical model generated from the third strategy for estimating a three double integrator model. Here the state transition matrix ( $A$ ) is an identity matrix. The input matrix ( $B$ ) and output matrix ( $C$ ) are estimated.

The IAE and TV values are given in Table 3.11. From this it is clear that the model based on the third strategy has the smallest error (using the IAE metric), but it also has

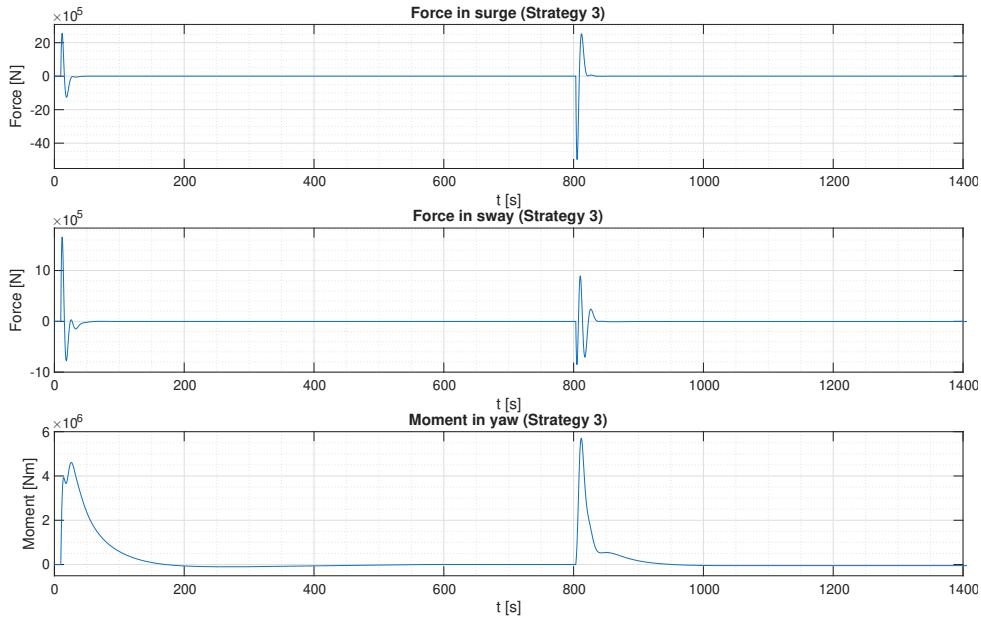


Figure 3.43: Inputs computed by LQ optimal control algorithm. Based on the third strategy for estimating model matrices, resulting in a three double integrators model.

a much higher total value in surge and sway than the other model.

Table 3.11: Summary of simulation results when using the three double integrator models for controlling the supply vessel.

Description	Strategy 2	Strategy 3
IAE - North position	1666.52	180.07
IAE - East position	1115.53	138.83
IAE - Heading	83.36	48.21
TV - North position	$0.10 \cdot 10^8$	$0.50 \cdot 10^8$
TV - East position	$0.06 \cdot 10^8$	$0.23 \cdot 10^8$
TV - Heading	$10.70 \cdot 10^8$	$3.71 \cdot 10^8$

## 3.5 Tracking

This section will look into path-following and trajectory tracking. Any of the control strategies discussed earlier can be used for this purpose, but here we will look into LQ optimal control and LPV MPC.

For all simulations, the wind, current, and wave disturbances are as shown in Figures 3.44, 3.45 and 3.46.

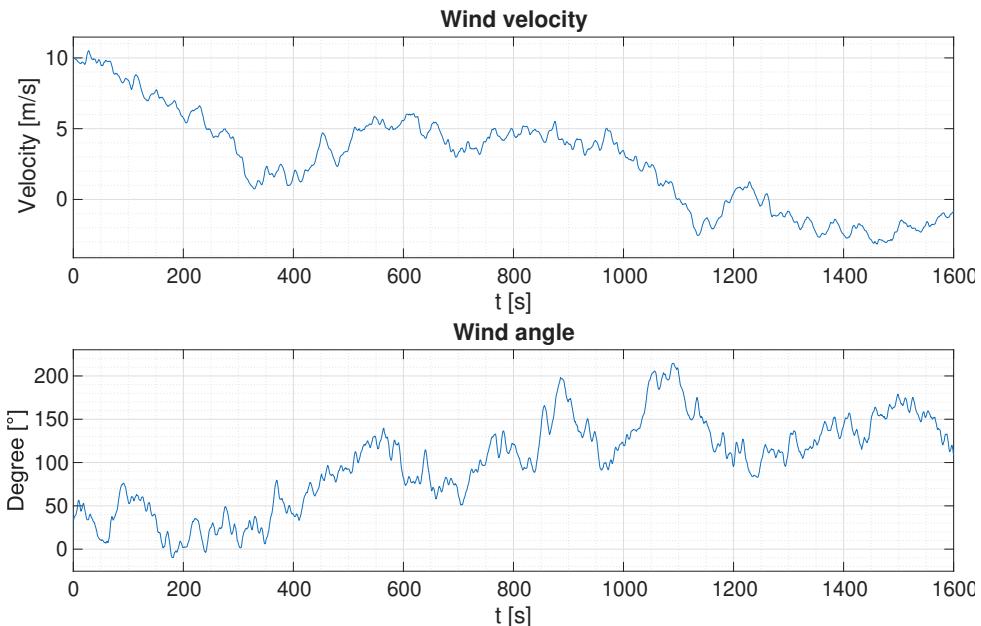


Figure 3.44: Plot depicts the wind disturbance applied when simulating path-following and trajectory tracking.

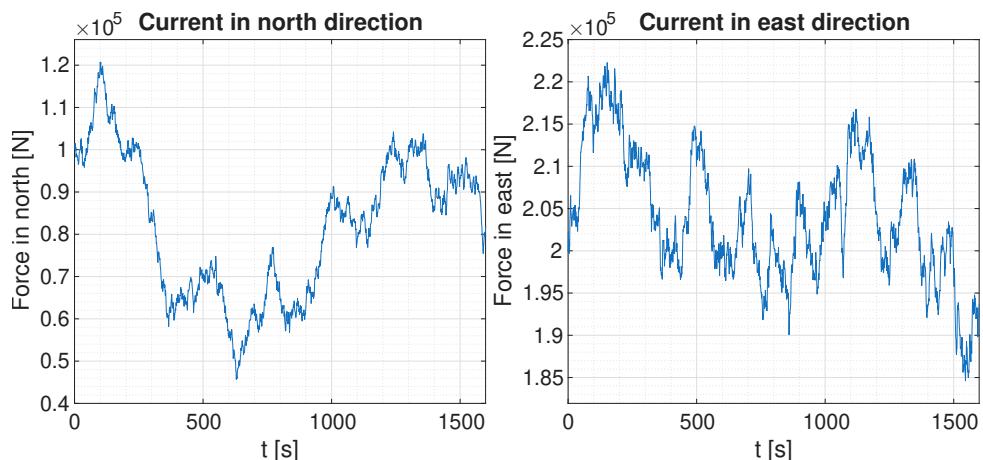


Figure 3.45: Plot shows current forces in north and east direction applied when simulating path-following and trajectory tracking.

### 3.5.1 Simple path following

The path of the vessel when using simple path-following is shown in 3.47. Setpoint tracking and inputs are depicted in 3.48 and 3.49, respectively. For both controllers, it is clear that the vessel heading is slow in changing, causing the vessel to wait at the active waypoint until the heading is within the required distance from the setpoint; then the vessel will turn to the next waypoint.

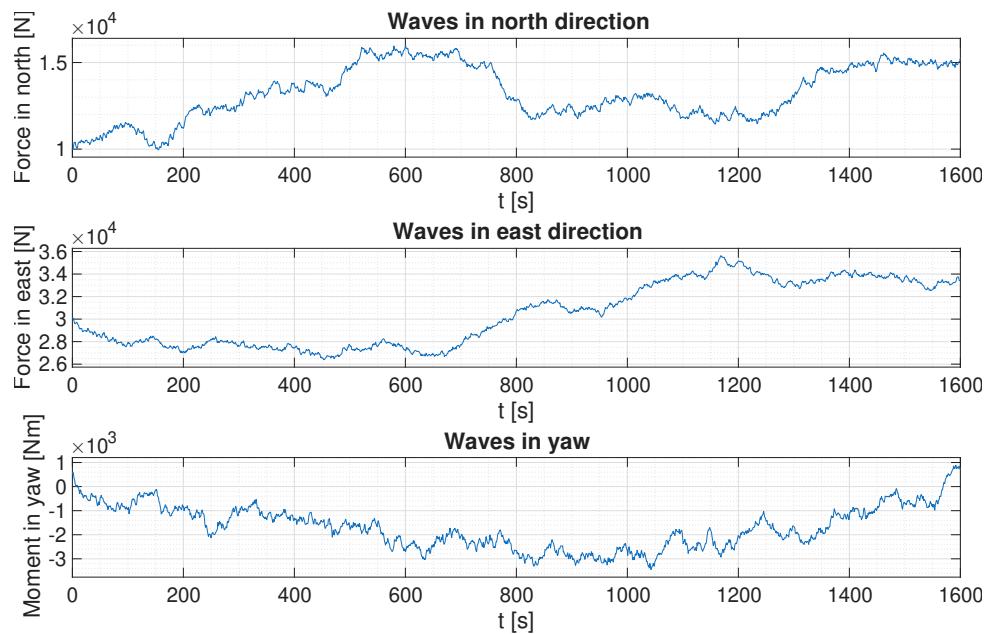


Figure 3.46: Plot depicts wave forces in north and east direction, and yaw moment as applied when simulating path-following and trajectory tracking.

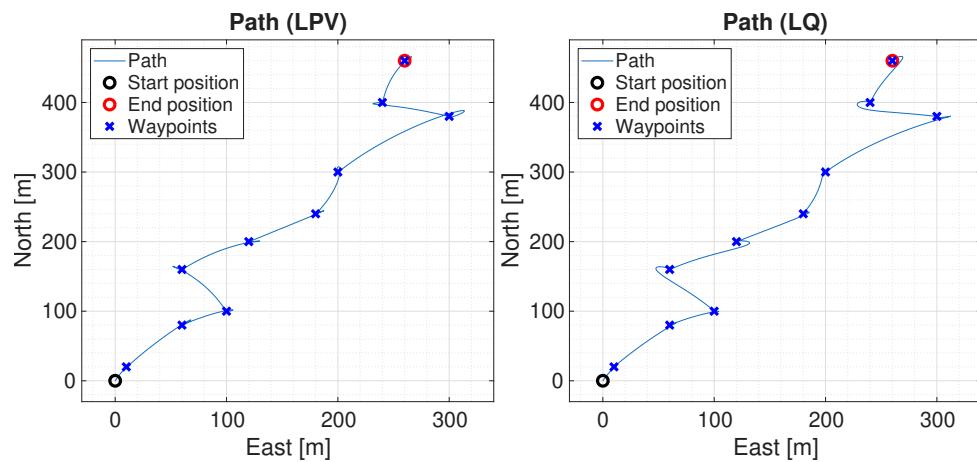


Figure 3.47: This plot depicts the waypoints and actual ship movement in north-east plane when using simple path following. It shows the results when running LQ optimal control and LPV MPC as control strategies.

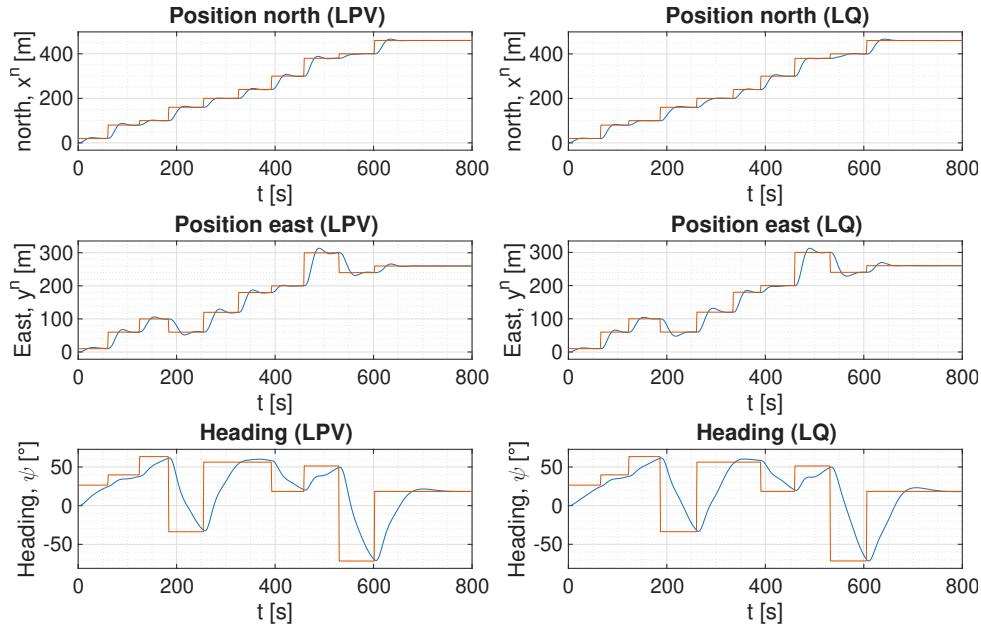


Figure 3.48: Shows setpoint tracking when using simple path following. The control problem is solved using both LQ optimal control and LPV MPC. Here the blue line marks the actual vessel data and the orange line is the setpoint.

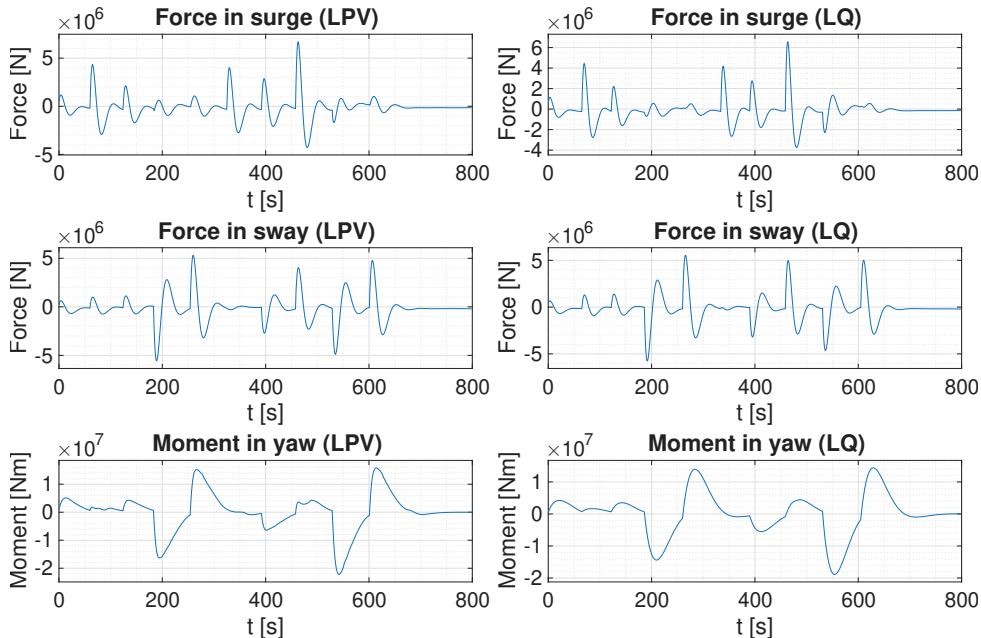


Figure 3.49: Inputs computed by LQ optimal control algorithm and LPV MPC when running simple path following. These inputs are net thruster forces in surge and sway, and moment in yaw.

### 3.5.2 More advanced tracking strategy

When using cubic spline interpolation for trajectory generation, the results are depicted in Figures 3.50, 3.51 and 3.52.

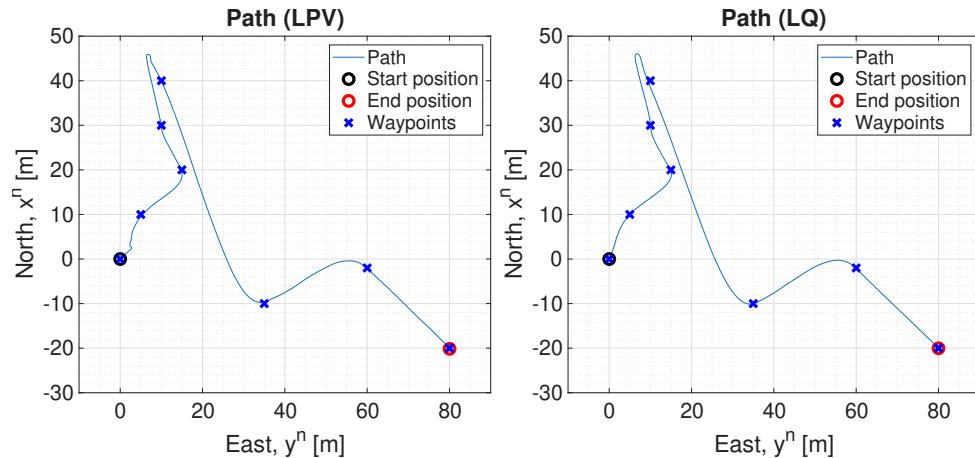


Figure 3.50: This plot depicts the vessel movement when generating setpoints from cubic spline interpolation in the north-east plane. It shows the results when running LQ optimal control and LPV MPC.

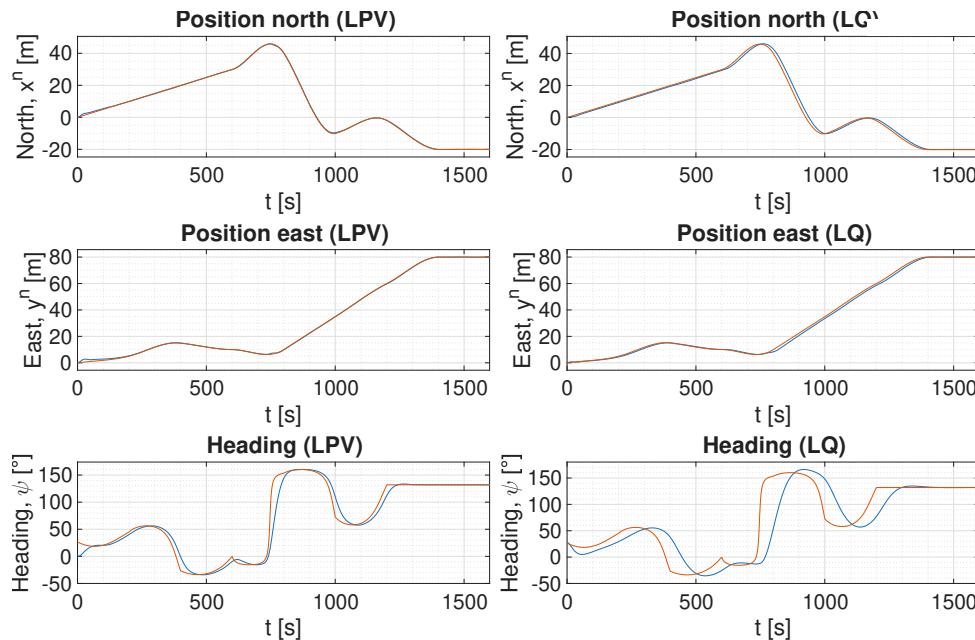


Figure 3.51: Shows setpoint tracking when using tracking strategy based on cubic spline interpolation. The control problem is solved using both LQ optimal control and LPV MPC. Here blue line marks the actual vessel data and the orange line is the setpoint.

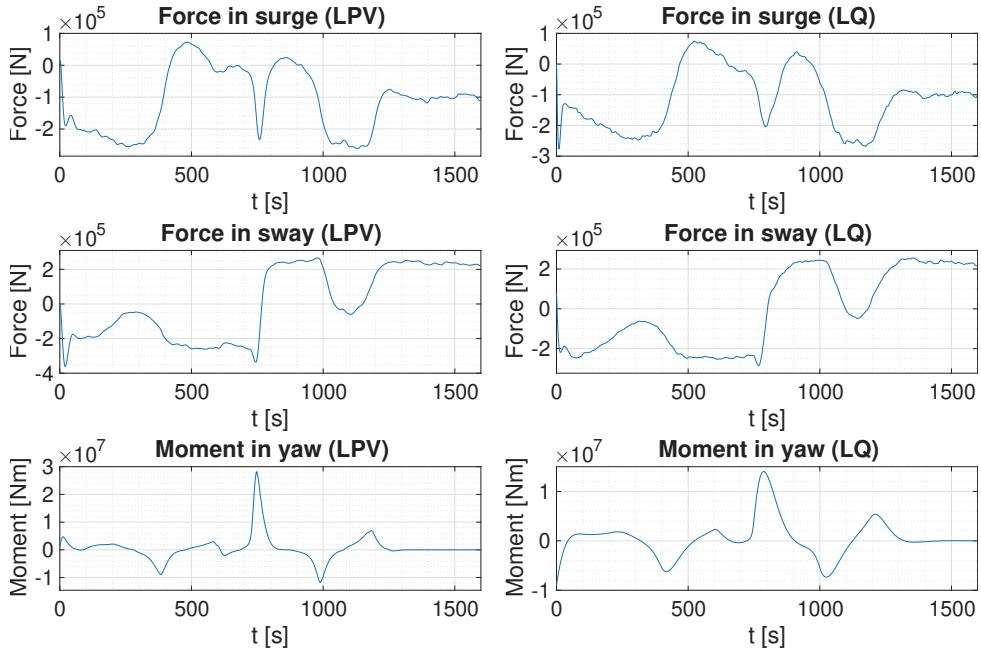


Figure 3.52: Inputs computed by LQ optimal control algorithm and LPV MPC when using cubic spline interpolation for setpoint calculation. These inputs are net forces in surge, sway, and moment in yaw.

### 3.5.3 Summary

The results from the simulations are summarized in Table 3.12.

In the case of simple path following strategy, both methods manage to reach the setpoints and have an integrated absolute error of the same order. However, it is worth noting that the applied inputs (forces and moment) experience dramatic spikes each time one waypoint (setpoint) is replaced with another. This is of course not ideal behavior, as it induces wear and tear on the thrusters and can cause increased fuel usage. It should also be stressed that the tuning weights of the control algorithm might need re-tuning for good performance if the distances between consecutive waypoints are considerably larger than what is usually used when performing setpoint changes under normal DP operation.

In the case of the more advanced tracking algorithm, the LPV MPC is clearly preferable compared to the LQ optimal control approach. The setpoints are continuously changing, but always in close proximity to the previous setpoint. The LQ optimal control concerns itself with the current value of the setpoint, but the MPC uses knowledge of the future setpoint horizon to better adapt to setpoint tracking. This might partially explain the improved performance of LPV MPC compared to LQ optimal control.

The nature of the simple path-following approach is that large errors are observed each time a new waypoint becomes active. This explains why the IAE values are much higher

Table 3.12: Summary of tracking results using IAE and TV metric. The simple strategy refers to the usage of waypoints. The more advanced strategy refers to the use of cubic spline interpolation to calculate setpoints.

Strategy	Description	LPV	LQ
Simple	IAE - North position	6896.28	7165.81
	IAE - East position	7663.58	7565.06
	IAE - Heading	302.65	347.10
Advanced	IAE - North position	295.96	1585.33
	IAE - East position	325.58	1229.37
	IAE - Heading	183.82	578.14
Simple	TV - North position	$5.61 \cdot 10^8$	$5.20 \cdot 10^8$
	TV - East position	$6.64 \cdot 10^8$	$6.92 \cdot 10^8$
	TV - Heading	$35.86 \cdot 10^8$	$37.29 \cdot 10^8$
Advanced	TV - North position	$1.96 \cdot 10^8$	$2.04 \cdot 10^8$
	TV - East position	$3.04 \cdot 10^8$	$2.95 \cdot 10^8$
	TV - Heading	$36.71 \cdot 10^8$	$37.88 \cdot 10^8$

than the corresponding cubic spline algorithm.

Notice that the total values (TV) are not very different for the two controllers.

## 3.6 Green DP

Green dynamic positioning is concerned with reducing the environmental footprint by minimizing fuel consumption. A version has been implemented using a specialized non-linear cost function. This was discussed in Section 2.12. The green DP controller will be applied to the supply model.

### 3.6.1 Comparison of green DP versus ordinary DP control when using reduction factor

To test out whether the effect of green DP could reduce fuel consumption, a scenario was considered where the wave and current forces trend toward a mean zero value. This was obtained by introducing a reduction factor  $\alpha$  such that the random walk for wave drift and current forces could be modified:

$$\tau_{c,k+1} = \alpha \tau_{c,k} + \varepsilon_{c,k} \quad (3.11)$$

$$\tau_{w,k+1} = \alpha \tau_{w,k} + \varepsilon_{w,k} \quad (3.12)$$

Here  $\tau_c$  and  $\tau_w$  represents the current and wave forces, while  $\varepsilon$  is zero mean gaussian noise. Using  $0 < \alpha < 1$  ensures the environmental forces do not deviate too much from zero. In the simulations the wind model remains unchanged. Example realization of using this method is depicted in Figure 3.53.

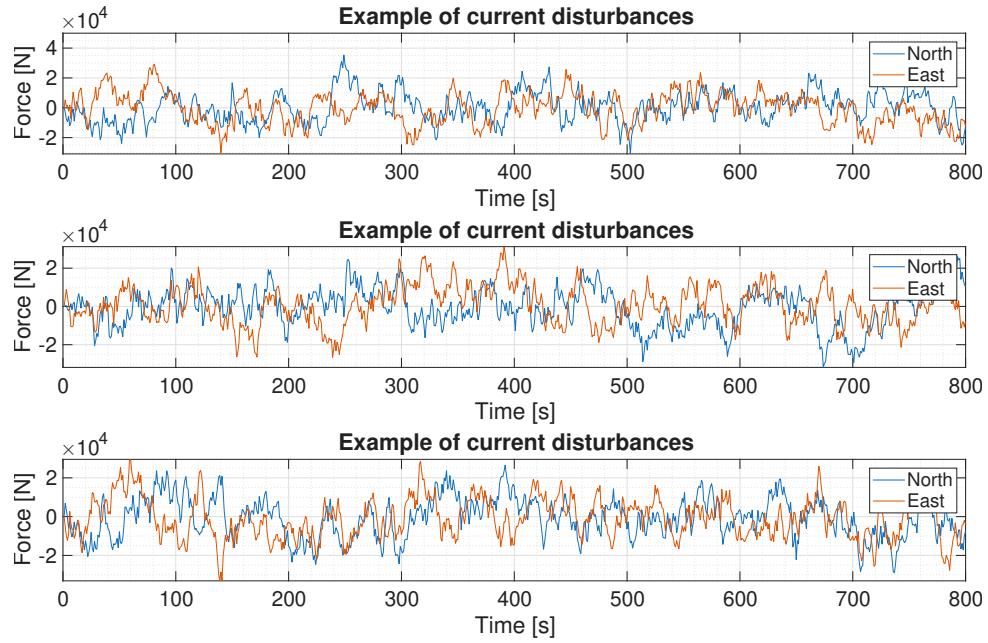


Figure 3.53: Plot shows three examples of current realization using random walk with reduction factor. Note how the forces are changing sign frequently.

It is not obvious that there will be any advantage in operating in green DP under circumstances where environmental conditions are pushing the ship to operate at the border edge of the green DP limit compared to staying at a certain setpoint. However, if the environmental disturbances are small and frequently changes direction, then the vessel is allowed to drift in different directions, as long as the border is not crossed. The normal DP, however, will punish any deviation immediately. These are the conditions that are expected to be favorable for the green DP approach.

The metric for fuel consumption considered here will be the total value (TV) in surge, sway, and yaw. The integrated absolute error makes little sense as a performance measure in this case, as the vessel should be allowed to drift.

To investigate the fuel saving potential of the environmental mode, 100 randomly generated realizations of environmental disturbances were created. Then the green mode was compared against normal DP mode. Non-linear MPC were used in both cases to make the algorithms as similar as possible. Each simulation lasted for a total of 800 seconds.

The results of these Monte Carlo simulations are depicted in Figure 3.54. The aggregated results are shown in Table 3.13.

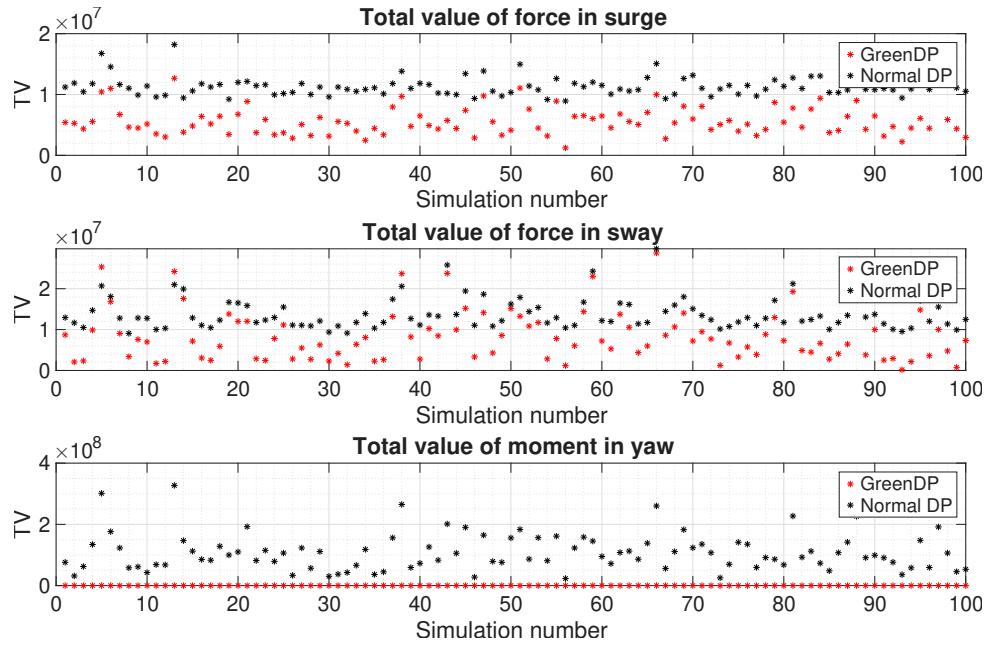


Figure 3.54: The plot shows the total value (TV) for forces and moment from running 100 simulations (Monte Carlo) in both green DP mode and normal DP mode under different environmental conditions. Clearly the simulations suggests that the green DP mode is preferable to normal DP when it comes total value.

Table 3.13: Summary of Monte Carlo simulation with respect to green DP when using reduction factor in the calculation of randomly generated waves and current. The mean total value (TV) is depicted for both the green DP and the normal DP.

Description	Green DP	Normal DP	Reduction (%)
TV surge force	$0.56 \times 10^7$	$1.13 \times 10^7$	50.2%
TV sway force	$0.83 \times 10^7$	$1.38 \times 10^7$	39.9%
TV yaw moment	$0.02 \times 10^7$	$10.80 \times 10^7$	99.9%

Some realizations from the Monte Carlo simulations are depicted in Figure 3.55.

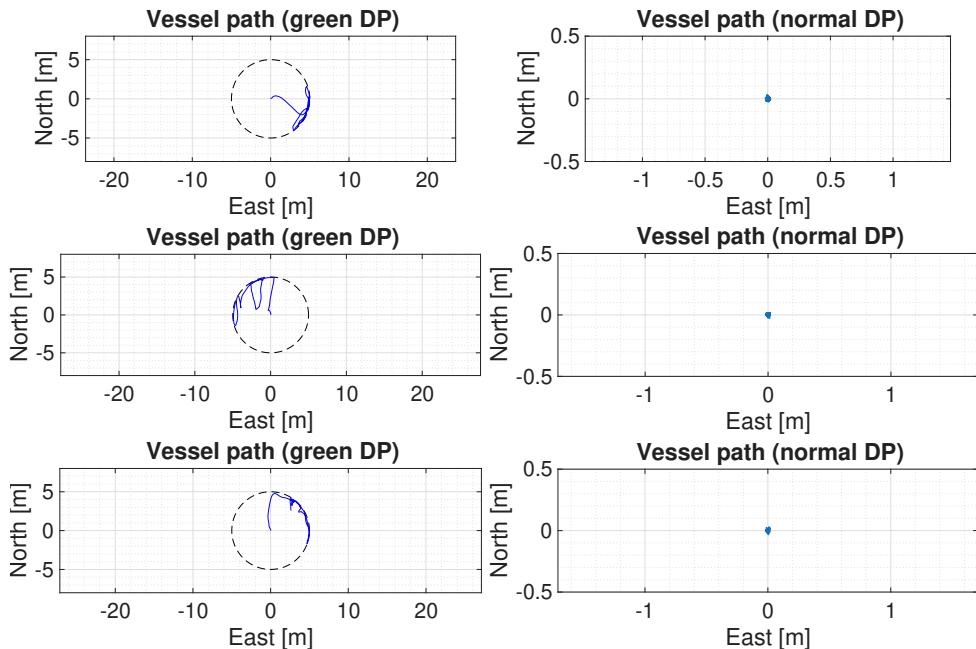


Figure 3.55: Example realizations of vessel movement running green DP mode and normal DP mode.

### 3.6.2 Comparison of green DP versus ordinary DP control when using no reduction factor

A second approach is to run the same Monte Carlo approach as above but with no reduction factor,  $\alpha = 1$  (allow the wave and current forces deviate significantly from zero). It seems likely that such conditions would be less favorable. The effect of doing so is shown in Figure 3.56. Some example path plots are shown in Figure 3.57. From these snapshots, we can see that the vessel spends time along the limiting border. This implies that it is likely constantly fighting environmental forces to avoid crossing the border. Thus, a situation occurs where thrusters are used aggressively to avoid being punished by the cost function.

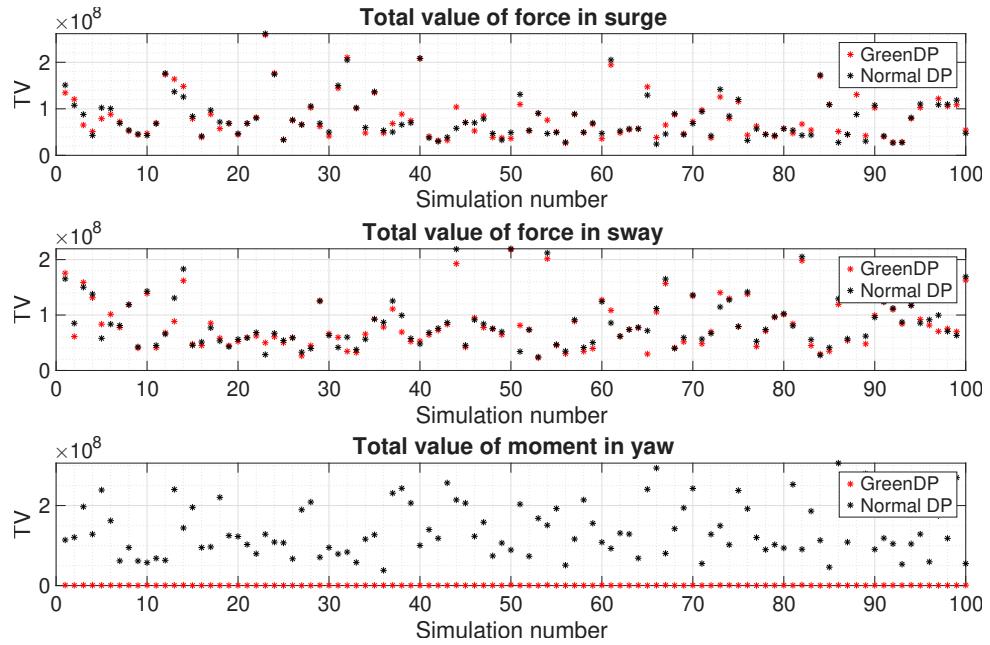


Figure 3.56: Monte Carlo simulations running 100 simulations in green DP mode and normal DP under different environmental conditions. The reduction factor is removed ( $\alpha = 1$ ). There is no obvious difference in the total value (except for moment in yaw since heading is not adjusted in green DP).

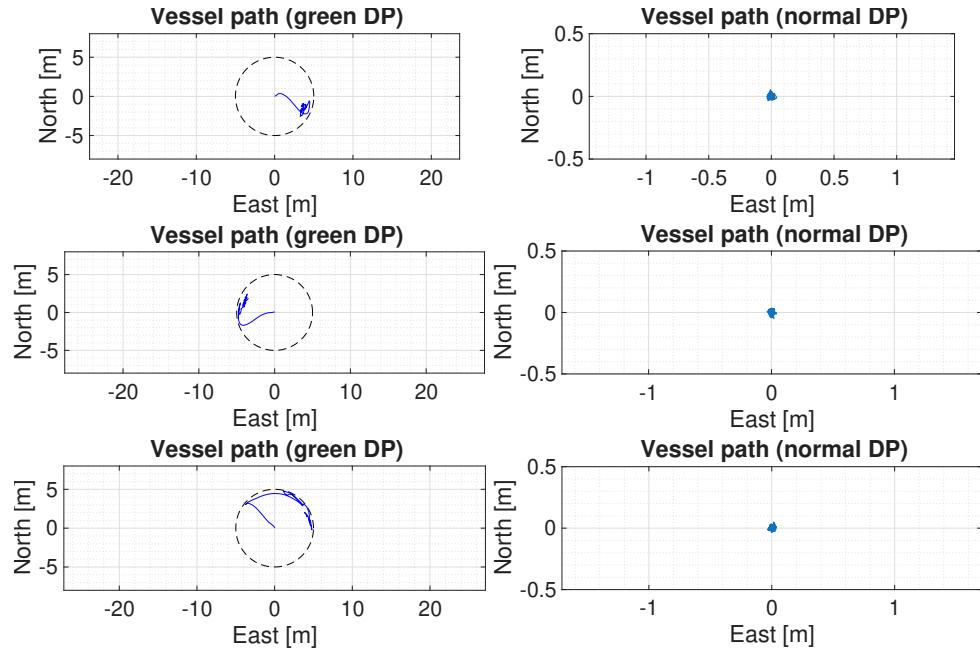


Figure 3.57: Example realizations of vessel movement running green DP mode and normal DP mode with the reduction factor removed.

The aggregated results are shown in Table 3.14.

Table 3.14: Summary of running Monte Carlo simulations. The mean total values are shown. It compares the simulations running on green DP controller with the normal DP controller. Here the reduction factor has been removed ( $\alpha = 1$ ).

Description	GreenDP	Normal DP	Percentage
TV surge force	$8.15 \times 10^7$	$8.04 \times 10^7$	-1.3%
TV sway force	$8.36 \times 10^7$	$8.63 \times 10^7$	3.2%
TV yaw moment	$0.05 \times 10^7$	$13.78 \times 10^7$	99.6%

One question is whether the green DP algorithm can be made to work better under these more challenging conditions using a different cost function or different tuning parameters. In all simulations, the cost function parameters  $W_i$  and  $W_o$  (used in equation (2.151)) were set to  $10^2$  and  $10^{14}$  respectively. The large value of  $W_o$  forces the vessel to never cross the circular border. Since running Monte Carlo is a prohibitively expensive operation computationally, only a limited variation of parameters were explored.

## 3.7 Thruster allocation

The thruster allocation section will be split into two parts: First a single simulation will be discussed with respect to three different thruster configurations. This will give some understanding of the mechanics of thruster allocation and the algorithms involved. The second part will use Monte Carlo simulation to verify that the results obtained in the original simulations are true in general.

In both cases LQ optimal control will be applied to the supply model.

### 3.7.1 Thruster allocation applied to single scenario

The control algorithm in DP will normally compute the net forces and yaw moment needed to move the ship to the desired position. This is not information that can be directly translated into individual thruster forces. The process of distributing forces and, in some cases, force angles (in the case of azimuth thrusters) is known as thruster allocation. The complexity of this process depends on the number of thrusters and the thruster types involved. Rotatable thrusters and rudders greatly complicate this process. We will look into three algorithms for thruster allocation. These are discussed in Section 2.13.

First, a single simulation will be run to build some understanding on the topic. The vessel will start at the origin, point directly north, and will undergo two setpoint changes at

times  $t = 100$  and  $t = 600$ . The entire simulation will last for 1000 seconds. The setpoints change as follows:  $[0, 0, 0^\circ] \rightarrow [10, 5, 30^\circ] \rightarrow [0, -5, 45^\circ]$ .

The wind, current, and wave disturbances used in all simulations are depicted in Figures 3.58, 3.59 and 3.60 respectively.

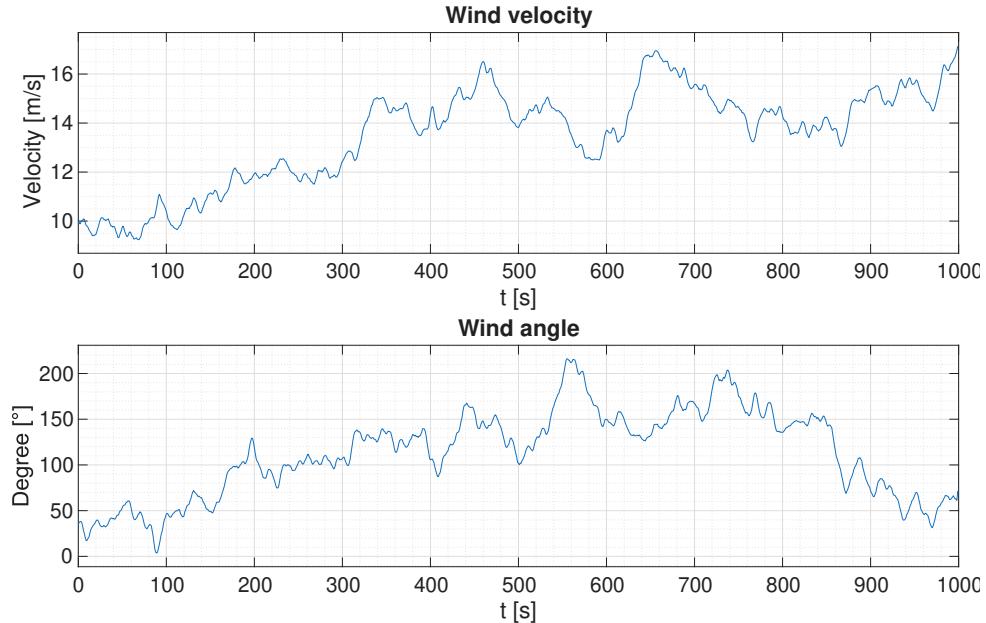


Figure 3.58: Plot depicts the wind disturbances applied during thruster allocation simulations.

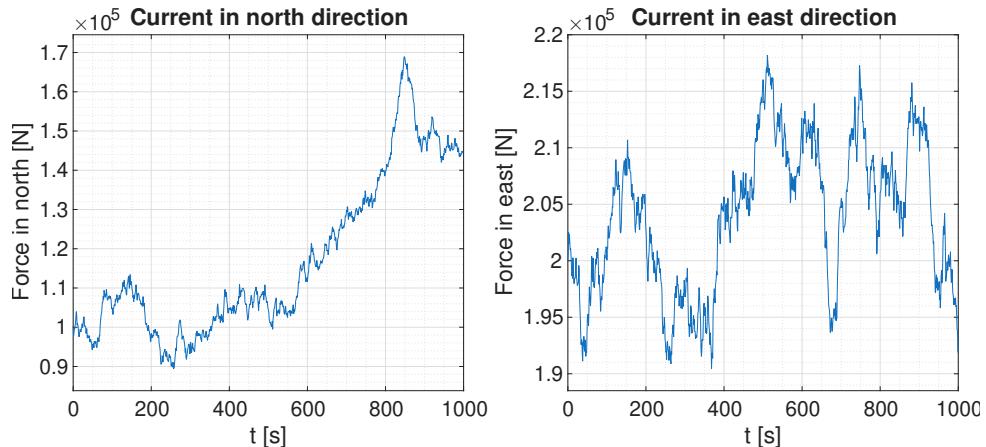


Figure 3.59: Plot shows current forces in north and east direction applied during during thruster allocation simulations.

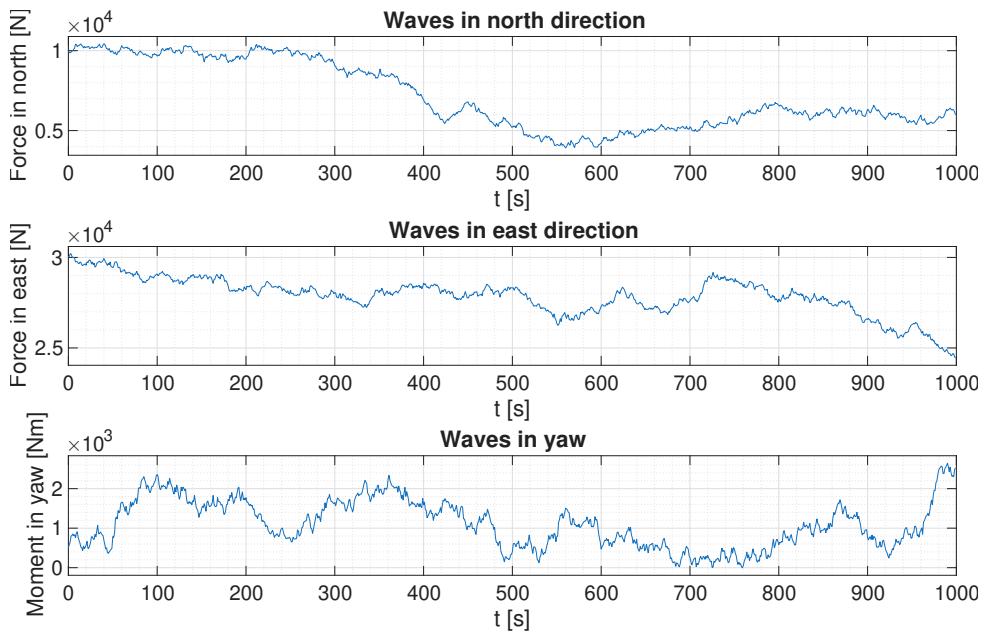


Figure 3.60: Plot depicts wave forces in north and east direction as applied during during thruster allocation simulations.

Three ship configurations will be considered here. The first two vessels will make use of two main propellers located in port and starboard positions. The first vessel will contain a bow tunnel thruster, while the second will contain a tunnel thruster in both the bow and stern position. These ships are depicted in Figure 3.61. These two vessels are used when employing explicit and quadratic programming approaches to thruster allocation. A third vessel type, using two azimuths in combination with main propellers, will be used when applying the nonlinear programming algorithm for thruster allocation. This vessel is depicted in Figure 3.62.

### Symbols

 : Propeller

 : Tunnel thruster

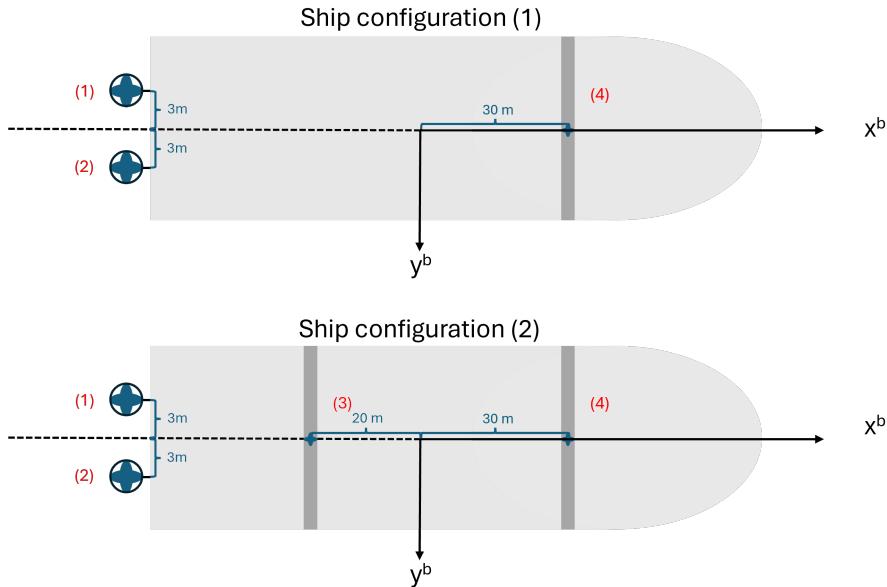


Figure 3.61: Nonrotatable thruster configuration for two ships investigated when performing thruster allocation. The first ship configuration (1) represents a ship with two main propellers and one bow thruster. The second ship configuration (2) is similar to the previous configuration, but includes a stern thruster. The red numbers mark port (1) and starboard (2) main propellers, stern (3) and bow (4) tunnel thruster.

### Symbols

 : Propeller

 : Azimuth

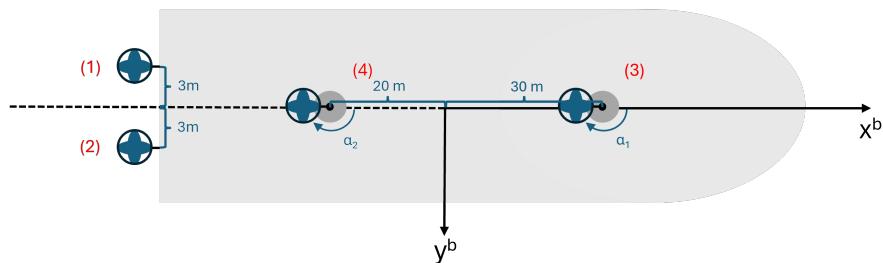


Figure 3.62: Rotatable thruster configuration for a vessel investigated when using the non-linear programming approach in solving the thruster allocation problem. Propulsive unit (1) and (2) represent the port and starboard main propellers, while unit (3) and (4) are the bow and aft azimuth thrusters (rotatable).

Notice that the thruster allocation algorithm will not change vessel performance unless the vessel fails to realize the power requested from the control algorithm (not the case here). The vessel model used is the supply model by Fossen. The LQ optimal controller discussed earlier is applied here. The results are shown in Figures 3.63, 3.64 and 3.65, depicting the setpoint tracking, inputs, and path plot.

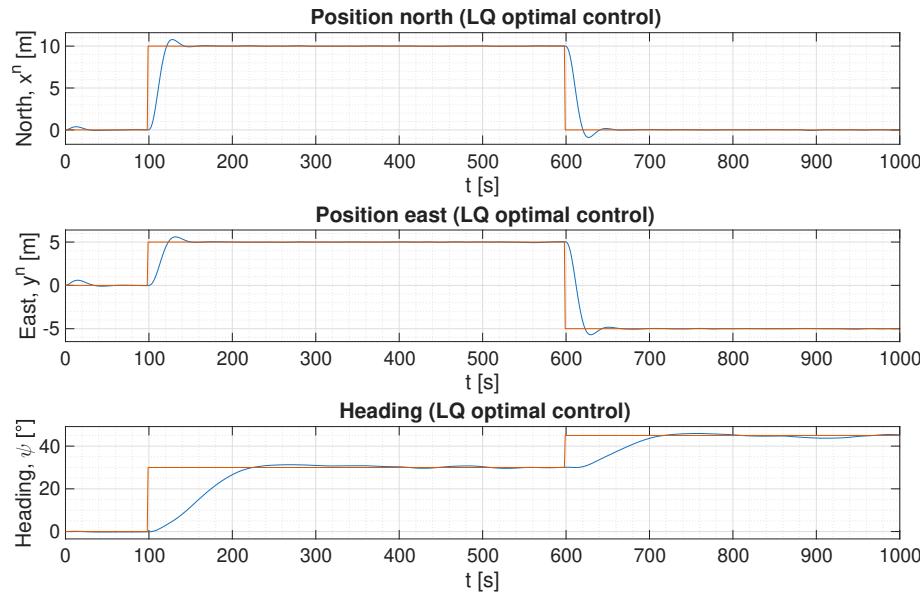


Figure 3.63: Shows setpoint tracking when using LQ optimal control on the supply model.

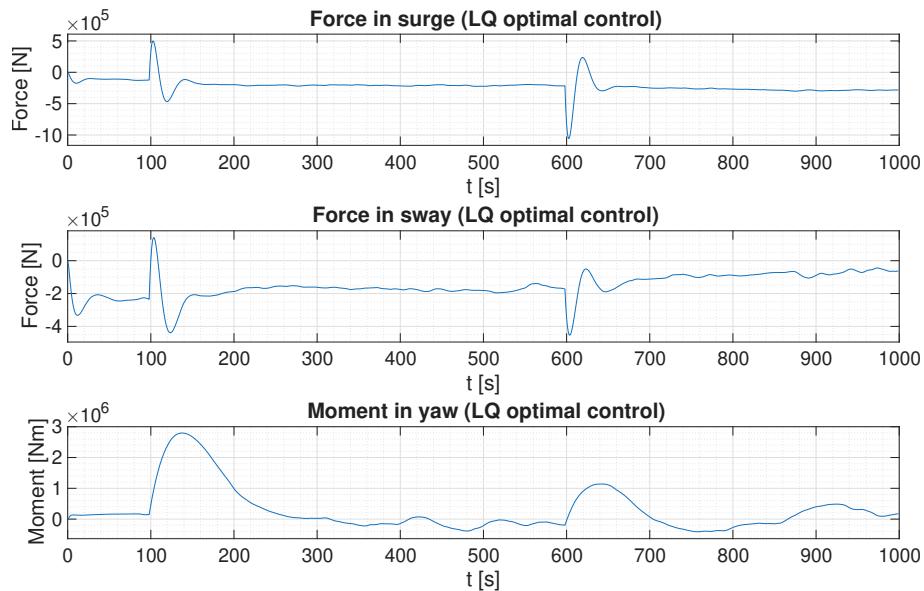


Figure 3.64: Depicts net thruster forces and moment computed by LQ optimal control algorithm.

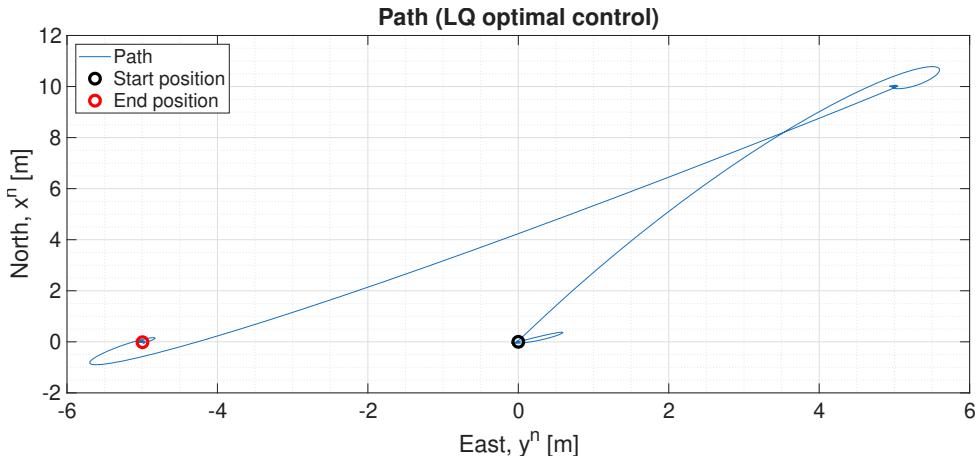


Figure 3.65: Path plot for simulation when using LQ optimal control on the supply model.

An interesting question here is how well the three different thruster configurations perform with respect to individual thruster force generation (this is related to fuel consumption and general wear and tear on equipment). The net force in surge and sway is the sum of individual thruster forces:

$$F_{su} = \sum_{i=1}^n f_{su,i} \quad F_{sw} = \sum_{i=1}^n f_{sw,i} \quad \tau_\psi = \sum_{i=1}^n \tau_i \quad (3.13)$$

Here  $n$  is the number of thrusters,  $F_{su}$  is the net force in surge,  $F_{sw}$  is the net force in sway,  $f_i$  is the thruster force generated by thruster  $i$ ,  $\tau_\psi$  is the net moment in yaw, and  $\tau_i$  represents the torque generated by thruster  $i$ . Since it is the sum of forces that generate the net force, it is possible for some of the thruster forces to point in opposite directions (partly cancel each-other out). In other words, the sum of forces in surge might be less than the sum of the absolute value of individual thruster forces in surge, or stated mathematically:  $F_{su} = \sum_{i=1}^n f_{i,su} \leq \sum_{i=1}^n |f_{i,su}|$ . This, of course, is also the case for forces in sway and moments in yaw. This is the reason why different thruster configurations can have different total values (TV).

The effect of forces and torque is illustrated in Figure 3.66 for a simple thruster setup. In summary; the sum of forces on a rigid body will move the vessel toward the position setpoint, but these forces will also generate torques that need to be balanced if the vessel is to reach the heading setpoint.

The thruster configuration matrix for thruster configuration (1) and (2) are depicted in equations (3.14) and (3.15), respectively.

$$T_{1 \text{ tunnel}} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 3 & -3 & 30 \end{bmatrix} \quad (3.14)$$

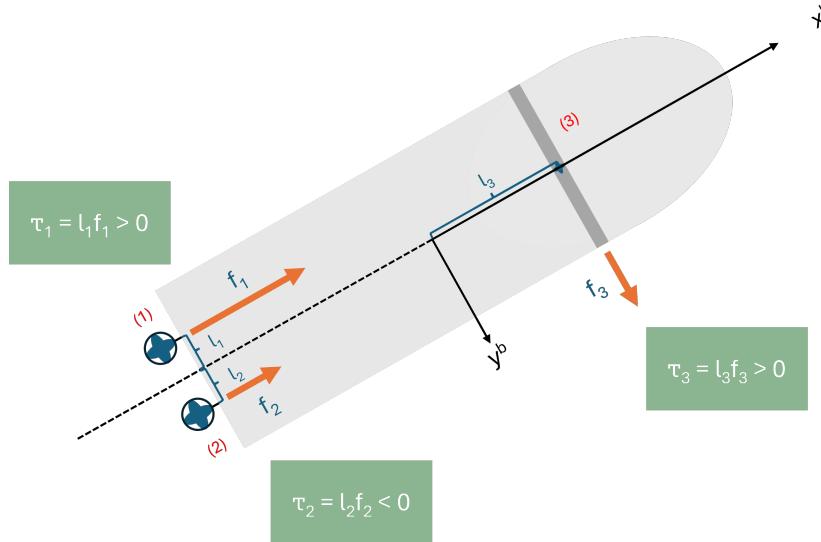


Figure 3.66: The figure depicts the effect of different thruster forces with respect to the torque. Note how the forces must be allocated in such a way that both the position and heading setpoint can be maintained. This allocation of individual thruster forces that realize the net forces and torque requested by the control algorithm is the general problem solved by the thruster allocation algorithm.

$$T_2 \text{ tunnels} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 3 & -3 & -20 & 30 \end{bmatrix} \quad (3.15)$$

### Explicit solution obtained using Lagrangian multipliers

This is based on the method described in Section 2.13.3 describing an explicit equation for calculating the thruster allocation obtained using Lagrangian multipliers. Here, the weighting matrix is set equal to the identity matrix, thus no thruster is preferential in realizing the total force required from the LQ optimal control algorithm.

The realization of thruster forces for configuration (1) is shown in Figure 3.67. The realization of thruster forces for configuration (2) is shown in Figure 3.68.

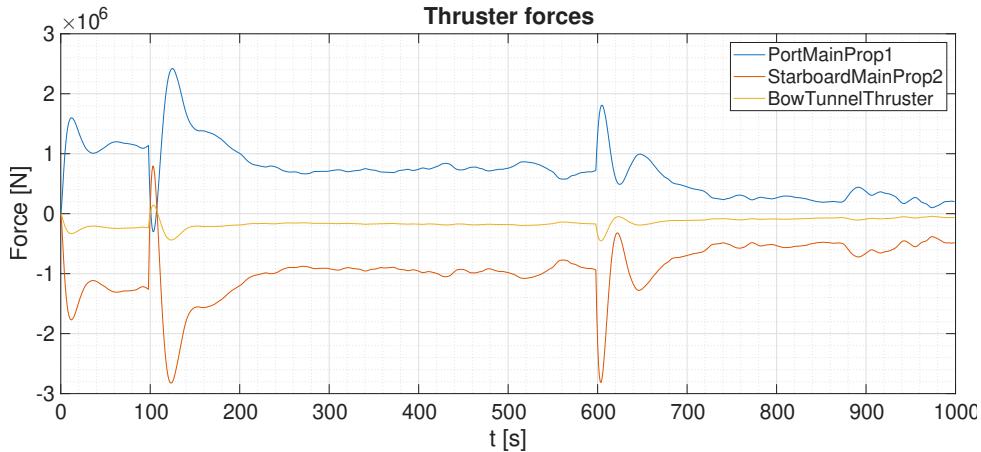


Figure 3.67: Thruster forces from thruster allocation algorithm for nonrotatable thruster configuration (1). It is using the explicit thruster allocation method.

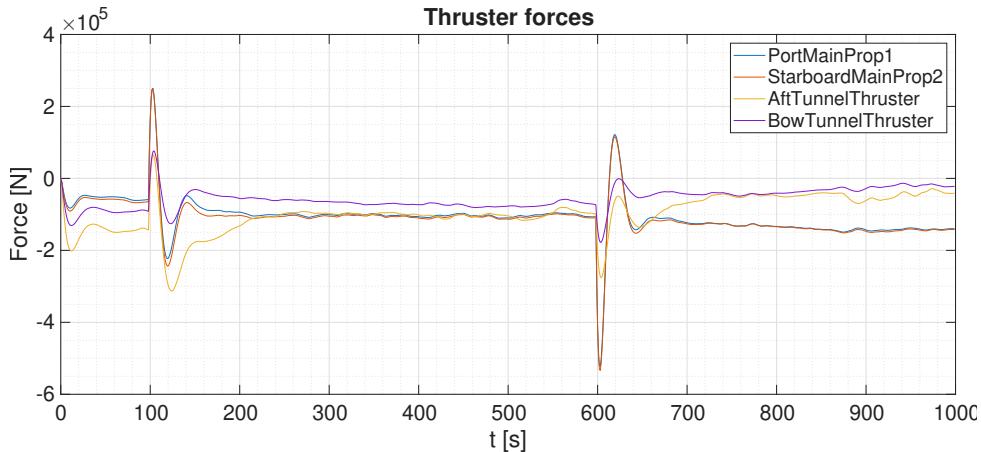


Figure 3.68: Thruster forces from thruster allocation algorithm for nonrotatable thruster configuration (2). It is using the explicit thruster allocation method.

By mere visual inspection, it is clear that the vessel configuration with thrusters in bow and stern is preferable. This seems natural as there is another degree of freedom and that the stern thruster can help balance the moment generated from the bow thruster.

This becomes even more clear when looking at snapshots of the vessel forces at three time instances:

- At time  $t = 110$  which is shortly after the setpoint change occurs at time  $t = 100$ .
- At time  $t = 610$  which is shortly after the second setpoint change occurs at time  $t = 600$ .
- At time  $t = 950$  where the vessel has settled on the setpoint and uses thruster forces to counter the environmental disturbances.

The snapshot of forces are depicted in Figures 3.69 and 3.70.

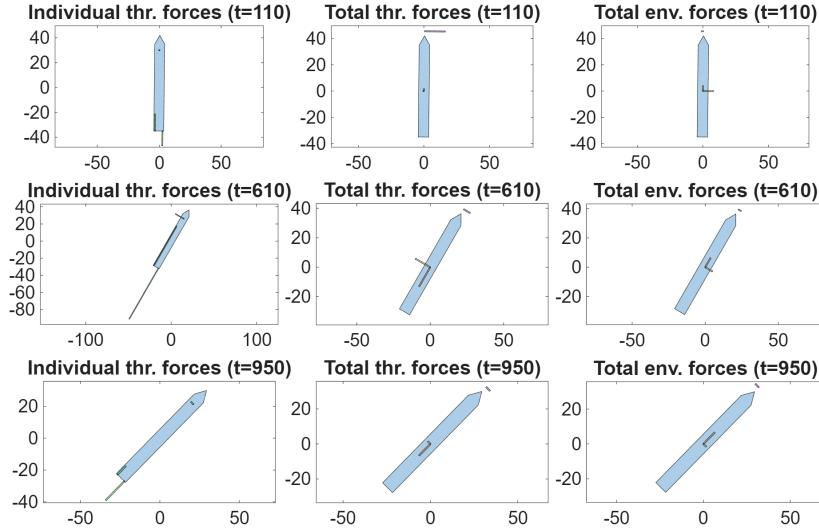


Figure 3.69: Shows snapshot of forces at time  $t = 110$ ,  $t = 610$  and  $t = 950$ . The thruster allocation algorithm used here is the explicit solution derived from Lagrange multipliers. The vessel model used has two main propellers and one bow tunnel thruster (ship configuration 1).

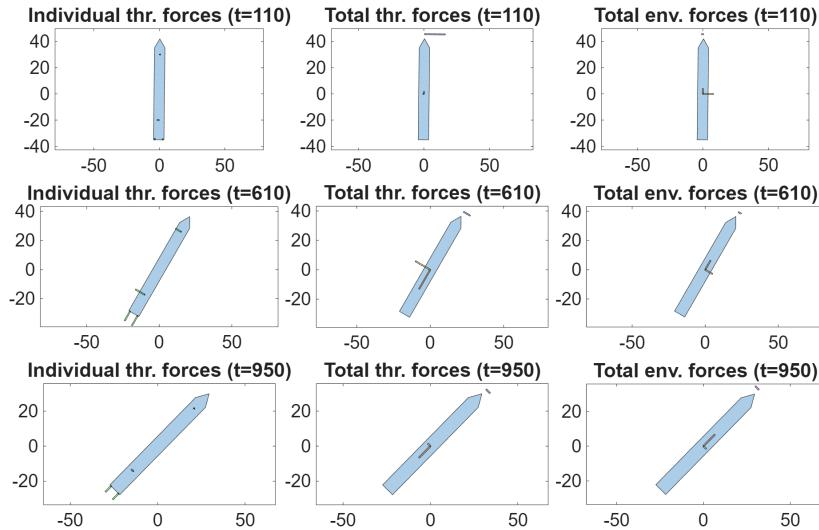


Figure 3.70: Shows snapshot of forces at time  $t = 110$ ,  $t = 610$  and  $t = 950$ . The thruster allocation algorithm used here is the explicit solution derived from Lagrange multipliers. The vessel model used has two main propellers and two tunnel thrusters in bow and stern (ship configuration 2).

It is worth noting that the vessel uses much less force when two tunnel thrusters are available. This is especially clear by comparing the thruster forces displayed in row 2 column 1 in the two snapshots (see Figures 3.69 and 3.70). It is also important to note that the thrusters for the one-tunnel configuration use more power even when in resting position, that is, there is no setpoint change transition occurring (see the pictures associated with  $t = 950$ ).

### QP problem when using nonrotatable thrusters

We will here consider the second nonrotatable thruster configuration discussed in the previous section. With this new allocation algorithm, it is possible to put limitations on the thruster force output.

We will here consider the case where the starboard main propeller is damaged or is for some other reason limited to only produce a maximum force of  $1.0 \times 10^5$  Newton. As can be seen in Figure 3.68, this propeller produced a higher thrust than this limit during the previous allocation using the explicit algorithm.

The new solution using the quadratic programming algorithm demands thruster forces as depicted in Figure 3.71. Note how the limitation is obeyed and how it changes the thruster force distribution. This distribution is expected to have a higher fuel cost, as the previously optimal solution is no longer valid due to the new thruster limitation. Snapshots of the force distribution are shown in Figure 3.72.

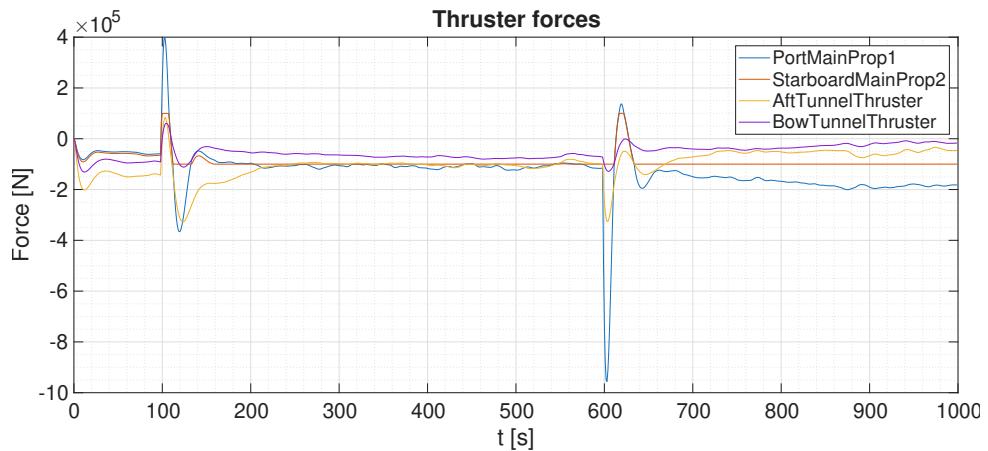


Figure 3.71: Thruster forces from thruster allocation algorithm for ship configuration (2). The quadratic programming algorithm is used for calculating thruster forces. Here the starboard propeller has been limited to produce a maximum of  $1.0 \times 10^5$  Newton.

Even though the starboard propeller is subject to a force limitation, we see no large jumps in thruster forces in the snapshots. It is clear from the individual thruster force plot that the forces have been redistributed.

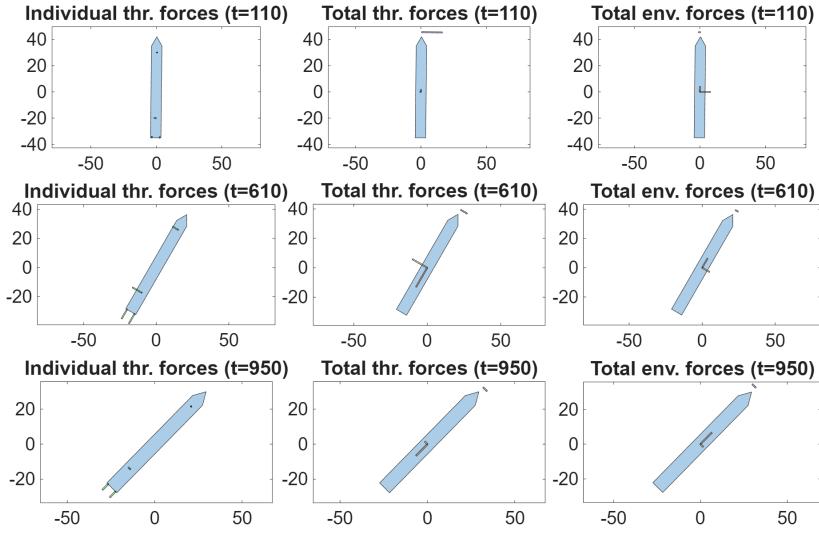


Figure 3.72: Shows snapshot of forces at time  $t = 110$ ,  $t = 610$  and  $t = 950$ . The thruster allocation algorithm used here is based on quadratic programming. The vessel model used has two main propellers and two tunnel thrusters in bow and stern (ship configuration 2). Here the starboard propeller has been limited to produce a maximum of  $1.0 \times 10^5$  Newton

### Non-linear optimization for thruster allocation with azimuths

This algorithm uses the ship configuration depicted in Figure 3.62. Such algorithms are needed when thrusters are given rotational degrees of freedom. This method is the most fuel efficient of all the algorithms and vessel configurations considered here, but it comes at a price. The algorithm is considerably more computationally expensive than the others; furthermore, there are several tuning parameters, increasing the complexity.

The thruster allocation matrix is given in (3.16).

$$T(\alpha) = \begin{bmatrix} 1 & 1 & \cos \alpha_1 & \cos \alpha_2 \\ 0 & 0 & \sin \alpha_1 & \sin \alpha_2 \\ 3 & -3 & 30 \sin \alpha_1 & -20 \sin \alpha_2 \end{bmatrix} \quad (3.16)$$

Thruster forces and snapshots are given in Figures 3.73 and 3.74, respectively.

We observe no dramatic thruster force jumps. Both force and azimuth angle changes tend to occur around the time of the setpoint changes.

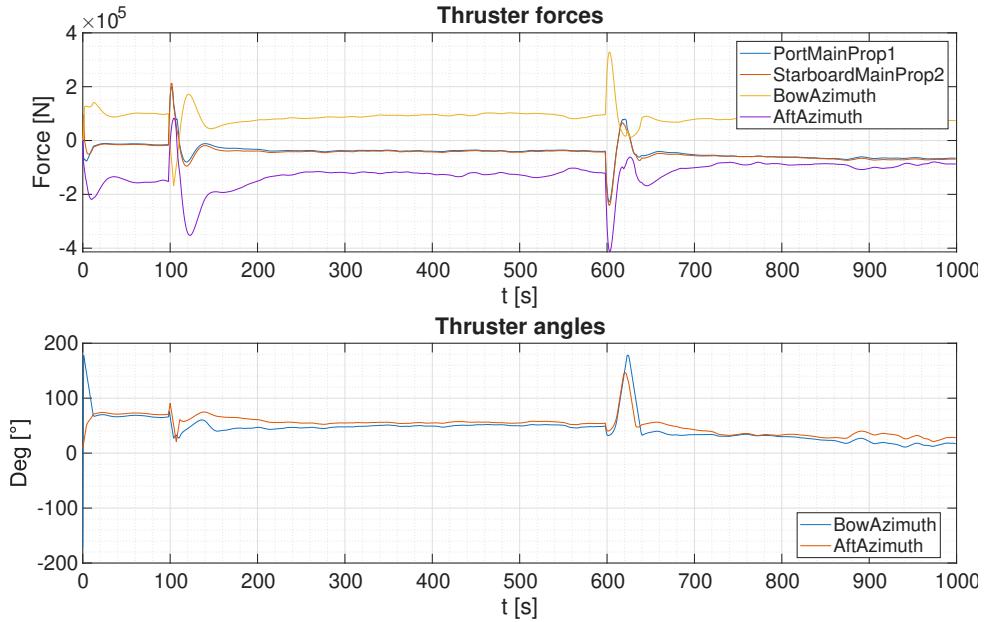


Figure 3.73: Thruster forces and angles from thruster allocation algorithm for thruster configuration with two main propellers and two azimuths located in bow and stern. A nonlinear optimization algorithm is used for obtaining individual thruster forces and angles.

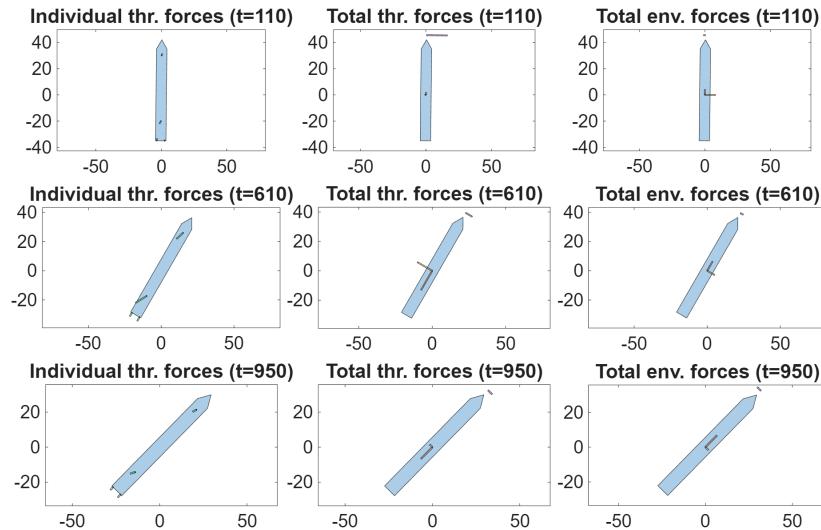


Figure 3.74: Shows snapshot of forces at time  $t = 110$ ,  $t = 610$  and  $t = 950$ . The thruster allocation algorithm used here is a non-linear optimization algorithm (due to azimuth non-linearity). The vessel model used has two main propellers and two azimuths in bow and stern.

### 3.7.2 Summary

The results of using different thruster configurations and allocation algorithms are summarized in Table 3.15. It is clear from this that increasing the number of thrusters

Table 3.15: Summary of result from single simulation run. L1/L2 refers to the explicit solution with one or two tunnel thrusters. QP1/QP2 has the same interpretation but refers to the quadratic programming approach. Azimuth refers to the non-linear optimization algorithm that allows for rotatable thrusters.

Description	L1	L2	QP1	QP2	Azimuth
IAF thr. 1	$7.12 \times 10^8$	$1.12 \times 10^8$	$7.12 \times 10^8$	$1.34 \times 10^8$	$0.46 \times 10^8$
IAF thr. 2	$9.36 \times 10^8$	$1.17 \times 10^8$	$9.36 \times 10^8$	$0.95 \times 10^8$	$0.50 \times 10^8$
IAF thr. 3	$1.56 \times 10^8$	$0.99 \times 10^8$	$1.56 \times 10^8$	$1.01 \times 10^8$	$0.87 \times 10^8$
IAF thr. 4	-	$0.57 \times 10^9$	-	$0.55 \times 10^8$	$1.26 \times 10^8$
Total	$1.80 \times 10^9$	$3.85 \times 10^8$	$1.80 \times 10^9$	$3.85 \times 10^8$	$3.09 \times 10^8$

improves the fuel efficiency of the vessel. In short, the more degrees of freedom available for force generation, the better the vessel performs. The vessel with the highest degree of freedom is the azimuth configuration. Not only can it generate four force vectors, but it can also adjust the angle of two of the force vectors for improved performance.

### 3.7.3 Monte Carlo simulation

To verify that certain thruster configurations are more expensive than others with respect to fuel consumption, several simulations were run with different setpoints and under varying weather conditions. This was explained in Section 2.14.1.

The results from the Monte Carlo simulation with respect to the integrated absolute value of thruster force (IAF), are shown in Figure 3.75. The results are also summarized in Table 3.16.

These results confirm what was discovered earlier; the thruster configuration is important in reducing fuel usage. It is important to invest in a proper thruster setup when building a vessel, as it can dramatically reduce fuel consumption during DP operations. This might be less important for ships performing long-distance movement (for example, merchant ships) as such operations rely mostly on main propellers.

Slack variables are also available for methods using the quadratic and non-linear programming approach. These are measures of failure to adhere to requested force and torque values from the controller. The individual slack variables (force in surge, force in sway and

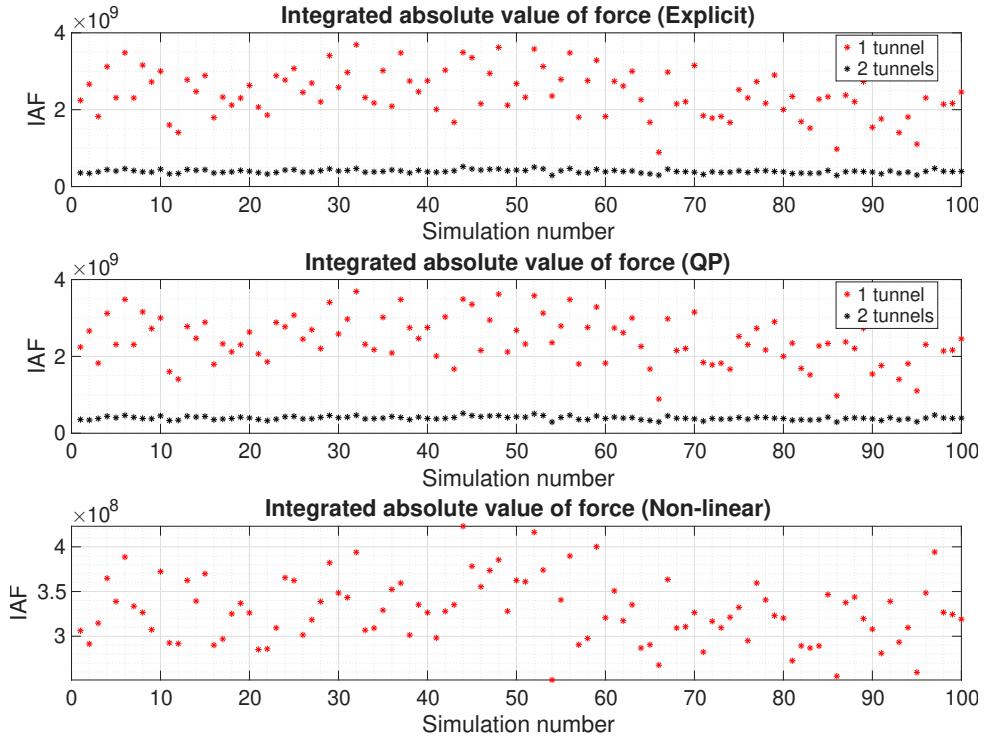


Figure 3.75: Results from Monte Carlo simulation when applied to thruster allocation. **Top plot:** depicts the forces when applying the explicit solution to the thruster allocation problem. It shows the total integrated absolute force for both the vessel with one and two tunnel thrusters. **Middle plot:** depicts the same, but running the quadratic programming approach with thruster limitations. The thruster force limitations are set very large in the vessel model with one tunnel thruster, hence not affecting the results (note that the allocation gets results equal to the explicit solution). In the two tunnel model the starboard main propeller is limited, which do affect the simulation, but it appears from the plot that the effect is small, the algorithm manages to compensate for the defective starboard main propeller using the three remaining propulsive units. **Bottom plot:** shows the integrated absolute force values for the azimuth ship configuration utilizing the non-linear thruster allocation algorithm.

torque in yaw) are computed at each timestep. For a single simulation run the individual slack variables can be summarized using the metric defined in equation 3.17.

$$s_{i,\text{tot}} = \sum_{k=1}^n |s_{i,k}| \Delta t. \quad (3.17)$$

Here  $\Delta t$  is the simulation time step,  $s_{i,k}$  representing the slack variable  $i$  (surge, sway, or yaw) at time  $t_k$ . This mirror the definition of the total value (TV) metric used earlier.

The mean of these values from the Monte Carlo simulations are given in Table 3.17.

It is interesting to note that the two-tunnel thruster configuration has significantly lower slack variable values in sway than the one-tunnel configuration. This makes sense since

Table 3.16: Summary of Monte Carlo simulation of thruster allocation algorithms and different thruster setups. L1/L2 refers to the explicit solution with one or two tunnel thrusters. QP1/QP2 has the same interpretation but refers to the quadratic programming approach. Azimuth refers to the non-linear optimization algorithm that allows for rotatable thrusters, using the azimuth ship configuration.

Description	L1	L2	QP1	QP2	Azimuth
IAF thr. 1	$1.12 \times 10^9$	$0.81 \times 10^8$	$1.12 \times 10^9$	$0.92 \times 10^8$	$0.30 \times 10^8$
IAF thr. 2	$1.15 \times 10^9$	$0.85 \times 10^8$	$1.15 \times 10^9$	$0.74 \times 10^8$	$0.32 \times 10^8$
IAF thr. 3	$0.18 \times 10^9$	$1.35 \times 10^8$	$0.18 \times 10^9$	$1.35 \times 10^8$	$1.13 \times 10^8$
IAF thr. 4	-	$0.94 \times 10^9$	-	$0.94 \times 10^8$	$1.53 \times 10^8$
Total	$2.44 \times 10^9$	$3.94 \times 10^8$	$2.44 \times 10^9$	$3.95 \times 10^8$	$3.28 \times 10^8$

Table 3.17: Summary of mean slack variables obtained during Monte Carlo simulations of different thruster allocation algorithms. QP1/QP2 refers to the quadratic programming approach with one or two tunnel thrusters. Azimuth refers to the non-linear optimization algorithm applied to the vessel configuration with two azimuth thrusters and two main propellers.

Description	QP1	QP2	Azimuth
Slack surge	0.82	0.94	12.09
Slack sway	114.75	0.96	19.45
Slack yaw 3	3.77	0.02	0.53
Total slack	119.34	1.91	32.07

this configuration contains a second tunnel thruster that generates force in the sway direction. The difference in surge is small, but this is not surprising since both models have two main propellers generating forces in the surge direction. Lastly, it should be noted that the yaw slack variable also improves, this is because the second tunnel thruster also contributes to the torque.

It should be stressed that these are very small values in general. For supply ships, forces of a hundred Newtons or less are hardly noticeable, and in most cases much smaller than the environmental forces affecting the vessel.

# 4 Discussion

This chapter summarizes all the major results with relevant remarks. It is split into five sections: control algorithms, system identification, path following/trajectory tracking, green DP, and thruster allocation.

## 4.1 Control algorithms

The results from the different control algorithms are summarized in Table 4.1. Looking at the integrated absolute error (IAE) metric, it appears that the MPC controller using the constant rotation formulation and LPV MPC approach perform the best among the MPC algorithms. These models are run on the same tuning parameters; thus, their performance should be directly comparable. From this, the obvious choice is either the constant-rotation formulation or LPV MPC, given they both have similar IAE and TV values. However, also computational efficiency should be taken into account as shown in Table 4.2. From this perspective, the constant-rotation formulation seems like the most attractive solution.

LQ performs slightly worse with respect to IAE, but this is a different controller with different tuning parameters and cannot be directly compared. The NMPC has by far the highest total value (TV) compared to the other controllers, but also the smallest IAE value. Comparison of NMPC with the other methods suggests that reducing the setpoint error quickly comes at the cost of higher thruster demand. This can also be seen by re-tuning the other controllers to more aggressive tuning.

It should be noted that there are no correct tuning, but some of the considerations that need to be taken into account are the following:

- Wear and tear on the thrusters and the mechanical limits of the thrusters.
- Fuel consumption.
- Safety of the ship crew. Rapid motion can create dangerous situations onboard the vessel and also to surrounding structures.
- Time to reach the setpoint and ability to remain close to the setpoint.

Table 4.1: Comparison of RMSE, IAE and TV values from all the different control strategies: LQ optimal control (deviation form), MPC and NMPC. All these values were created using Monte Carlo simulations and are copied from the relevant tables in the result chapter. Environmental disturbances were present in all simulations.

Algorithm	Description	RMSE	IAE	TV
LQ optimal control	North position	1.01	242.58	$1.65 \cdot 10^8$
	East position	1.05	264.22	$1.78 \cdot 10^8$
	Heading angle	0.88	347.10	$22.60 \cdot 10^8$
	<b>Total</b>	-	853.90	$2.60 \cdot 10^9$
MPC constant heading	North position	0.68	283.44	$1.66 \cdot 10^8$
	East position	0.77	358.88	$1.83 \cdot 10^8$
	Heading angle	0.43	120.54	$22.63 \cdot 10^8$
	<b>Total</b>	-	762.86	$2.61 \cdot 10^9$
MPC constant rotation	North position	0.54	222.87	$1.64 \cdot 10^8$
	East position	0.68	312.20	$1.78 \cdot 10^8$
	Heading angle	0.43	121.06	$22.52 \cdot 10^8$
	<b>Total</b>	-	656.13	$2.59 \cdot 10^9$
LPV MPC	North position	0.55	225.72	$1.64 \cdot 10^8$
	East position	0.67	308.99	$1.78 \cdot 10^8$
	Heading angle	0.43	120.41	$22.62 \cdot 10^8$
	<b>Total</b>	-	655.13	$2.60 \cdot 10^9$
Non-linear MPC	North position	0.28	59.24	$1.89 \cdot 10^8$
	East position	0.28	69.62	$2.31 \cdot 10^8$
	Heading angle	0.24	49.98	$36.62 \cdot 10^8$
	<b>Total</b>	-	178.83	$4.08 \cdot 10^9$

One advantage with MPC and NMPC is that they allow for putting limits on the maximum and minimum forces and moment allowed. Thus, it is possible to avoid situations where the thruster allocation algorithm fails to realize the controller demand because the thrusters have reached their force limit.

When it comes to controller tuning, it was the author's experience that LQ optimal control (on deviation form) was the easiest to tune. The MPC formulations and the NMPC required much more trial and error before landing on parameters that worked well.

Due to the computational load of NMPC, this is best avoided, unless special operations like green DP are needed. With MPC and NMPC one also has to deal with an extra tuning parameter, the horizon length. The LQ optimal control algorithm is found to be effective in dealing with external disturbances, but the MPC and NMPC need wave and wind disturbances together with the current estimate. If, say, the wind sensors were to

Table 4.2: Shows the computational load associated with different controller algorithms.

Algorithm	Computational load
LQ optimal control	Low
MPC constant heading	Moderate
MPC constant rotation	Moderate
LPV MPC	Moderate+
Non-linear MPC	High

suddenly deliver biased data, this would need to be compensated for by the "current" estimate. This process can take some time.

The cost functions always used the  $\Delta u$  term instead of  $u$ . The reason for this is that the cost function on the form  $J = e_k^T Q e_k + u_{k-1}^T P u_{k-1}$  is essentially a competition between reducing the setpoint error and reducing force and moment generation,  $u = (X, Y, N)$ . If no external disturbances are present, the vessel position will converge to the setpoint. However, in the presence of wind, waves, and current, there will be a bias present since  $u_k$  must balance these disturbances. However, since the disturbances are slowly varying, the change in thruster forces will be small, thus  $\Delta u \approx \mathbf{0}$ . Then the cost function will force  $e_k \rightarrow \mathbf{0}$ . This has been observed in simulations.

## 4.2 System identification

Different methods of system identification were applied to three different vessel models: supply, OSV and Balchen model. It was found that the mathematical model identified for each vessel was sufficient for use in a linear quadratic optimal controller.

One limitation is the assumption that the net force is known during the data collection process. In most cases external forces from waves, wind, and current will be present, so this limits the usefulness of this approach<sup>1</sup>. It seems unlikely that these forces would all be known while collecting data in a real world situation.

System identification is a particularly interesting approach since it allows the user to estimate the parameters of the underlying model directly from the data. The more conventional approach is to calculate such parameters from first-principles modeling based on vessel data. A second approach is to include the estimation of physical parameters

---

<sup>1</sup>If forces and moment from external disturbances are known, the input to the SI algorithm can be calculated using the following formula:

$$u = \tau_{\text{thr}} + \tau_{\text{wind}} + \tau_{\text{wave}} + \tau_{\text{current}}$$

as part of the tuning process. With system identification, the engineer will only need to focus on tuning the controller and Kalman filter.

Both the DSR, DSR\_e and PEM methods resulted in very similar controller performance. The IAE and TV values were very close for the three system identification methods, as can be seen in Tables 3.8 and 3.9.

Note that both the Balchen model and the OSV model are non-linear. This shows that the system identification method can manage to identify a sufficiently good linear model that works well in a model-based controller. It was noted that some care had to go into how the inputs were defined. The OSV model used the squared RPS signal as input, resulting in a very aggressive tuning. The net forces and moment demanded by the controller became so high that it is likely that the vessel would fail to realize them.

An interesting possibility is the use of recursive system identification algorithms. The controller can then adjust live to vessel changes due to ballast tank load or cargo weight.

Identifying a three double integrator model based on data collection from the supply model was shown to be possible, but it was necessary to identify more than just the input matrix for control purposes. One of the challenges is that the optimization routines that identify the matrix entries are highly dependent on the starting points; hence, multiple starting points must be tried, and this requires some domain knowledge.

## 4.3 Path following and trajectory tracking

The path following and trajectory tracking algorithms were tested using the supply model with two different controllers (LPV MPC and LQ optimal control). Looking at the integrated absolute error metric (as given in Table 3.12) both controllers performed about the same for path following. However, there is a large difference when using the trajectory tracking algorithm. Here, LPV clearly dominates. The reason for this discrepancy is that the model predictive controller has knowledge about the future setpoint values, hence it is more able to adjust the velocity. Figure 3.51 shows that the LPV MPC follows the setpoints fairly accurately while the LQ algorithm lags a little behind.

## 4.4 Green DP

The green DP algorithm suggested in this text used a non-linear cost function. By tailoring the external disturbances (waves and current) to frequently change direction, it was observed through Monte Carlo simulations that some significant improvement in fuel consumption was possible (see Table 3.13). However, it should be stressed that, apart from the wind model, very simple models were used for external disturbances, based on random

walk and a reduction factor (see equations (3.11) and (3.12)). The total reduction in the net forces/moment in surge, sway, and yaw was significant. The yaw moment reduction is obvious, as there is no longer any restriction on heading angle, but also the surge and sway forces are reduced considerably. This indicates that a considerable fuel reduction is possible under the weather conditions described above.

When the weather conditions were changed to follow a pure random walk algorithm, allowing the net disturbance forces to drift more in one direction, the results showed small changes in total value between running in green DP mode and normal DP (See Table 3.14). In this case, the weather conditions are more likely to push the vessel toward the border, thus forcing the ship to fight the environmental forces to remain within the circular border that represents the green area of operation.

## 4.5 Thruster allocation

The thruster allocation algorithms were the final piece of the puzzle when solving dynamic positioning. The thruster configuration (the position of the tunnel thrusters, the main propellers, and the azimuth thrusters) was shown to be important. These propulsive units are responsible for achieving the net forces in surge, sway, and moment in yaw demanded by the controller.

Generally, it was found that an increase in the number of degrees of freedom in the thruster configuration resulted in reduced thruster usage. This can be seen from Table 3.16

If only fixed thrusters and propellers are present (no rudder or azimuths), the thruster allocation problem can be solved explicitly using the method described in Section 2.13.3. If thruster output limits are used, it can be solved using a quadratic programming approach described in Section 2.13.4. Finally, when rotatable propulsive units are introduced, such as azimuths, the problem becomes nonlinear. This is solved using the non-linear optimization algorithm outlined in Section 2.13.5.

The effect of running different algorithms and vessel setups is shown in Figure 4.1.

It should be noted that the setup with two tunnel thrusters (in bow and stern) generated far less force than the single bow tunnel thruster setup for both the explicit and QP algorithm. The placement of tunnel thrusters in bow and stern is important since the vessel can then better control the torque in yaw, as opposed to the possibility of placing both tunnels in bow or stern. The azimuth ship configuration (two main propellers and two azimuths) was the best thruster allocation design. Note that this solution has two extra degrees of freedom as the azimuth thrusters can change the force angle.

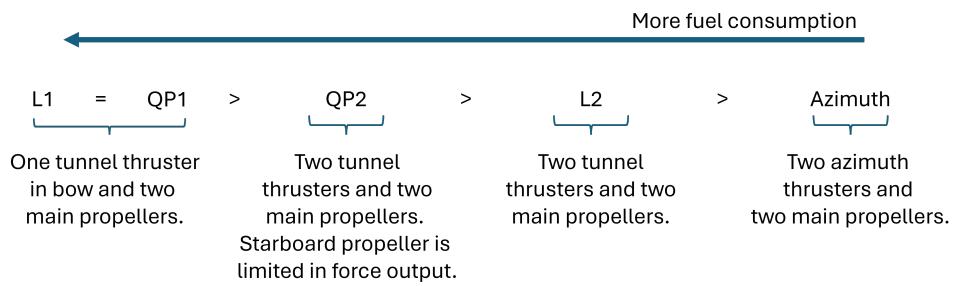


Figure 4.1: Depiction of fuel consumption usage for different algorithms and vessel setups based on results from Month Carlo simulation.  $L_1/L_2$  refers to the explicit solution with one or two tunnel thrusters.  $QP_1/QP_2$  has the same interpretation but refers to the quadratic programming approach. Azimuth refers to the non-linear optimization algorithm that allows for rotatable thrusters, applied to the azimuth ship configuration.

# 5 Conclusion

This thesis has studied different control strategies to maintain and change the position of a vessel under varying weather conditions. The control algorithms investigated were a linear quadratic optimal controller on deviation form, different formulations of model predictive controllers, and a nonlinear model predictive controller. The latter method was needed to implement a non-linear cost function for the less restrictive fuel-conserving green DP mode. There are advantages and disadvantages with each method with respect to complexity, computational load, and performance.

DP has been shown to be more than just position-keeping. The green DP approach showed that it is possible to reduce fuel consumption by allowing the vessel to drift within specified boundaries. The path following and trajectory tracking algorithms showed how the DP could be used to follow some path in the horizontal plane. The trajectory tracking algorithm was the preferred method since it gave smooth transitions.

System identification was performed on three different vessel models (two from the Marine System Simulator (**MSS**) and the third being the famous Balchen model). These identified models were then used in the model-based LQ optimal controller to control the original vessel models. The Balchen and OSV models are non-linear, but the controller still managed to follow the setpoints. It was also shown that it was possible to reduce the supply model to a three double integrator model using a prediction error method (and specifying entries to be identified in the model matrices).

Thruster allocation algorithms were also discussed. It was shown very clearly that the number of thrusters and degrees of freedom in the thruster configuration allowed for improved performance in the sense that fuel consumption and thruster usage were reduced. Reducing thruster usage helps to avoid wear and tear on the equipment, in turn saving both money and future downtime. The reduction in fuel consumption is beneficial both from an economic and environmental perspective.

In essence, this thesis has looked at all the major components of a dynamic positioning system from a control perspective. From determining the forces and moment needed for position-keeping to allocating thruster forces (and angles in the case of azimuth thrusters) to all available thrusters on the ship. The effects of environmental disturbances have been considered, though the models for waves and current have been greatly simplified.

### **5.0.1 Suggestions for future work**

Some suggestions for possible extensions of the work are presented here:

- Add more precise models for wave and current. The wave forces consists of first- and second-order wave forces. The second-order forces cause oscillatory motion and have been ignored in this thesis. Normally, the oscillatory motion is removed from the data sent to the controller (through the use of a *wave filter*), to reduce fuel consumption.
- In the thruster allocation section, azimuth thrusters were investigated. It is also possible to look into the effect of using rudders for the main propellers in the allocation algorithm. This should improve the allocation capabilities of the vessel.
- Investigate the failure cases of the vessel, i.e. when the vessel fails to maintain the position due to strong environmental disturbances. Quantify the net force from the environment that the vessel can handle before losing position (looking at different angles of attack with respect to the BODY coordinate system). This requires that the thruster allocation algorithm have in-built limits on the individual thruster force outputs. Look into worst- and best-case thruster failure scenarios: If a thruster is lost, determine which thruster will have the most significant and least significant effect on position maintenance.
- There exists a recursive version of the DSR algorithm. This can be used to continuously update model matrices in the model-based controllers. By changing the underlying vessel parameters in the simulated model, the system identification process should be able to adjust to these changes. Such parameter changes do occur in the real world due to changes in the ballast tank load and cargo weight.





# Bibliography

- [1] D. Sobel, *Longitude: The True Story of a Lone Genius Who Solved the Greatest Scientific Problem of His Time*, First edition. Bloomsbury USA, 2007, ISBN: 9780802715296.
- [2] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*, Second edition. John Wiley & Sons Inc, 2021, ISBN: 9781119575054.
- [3] D. Ruscio and C. Dalen, ‘Tuning pd and pid controllers for double integrating plus time delay systems,’ *Modeling, Identification and Control (MIC Journal)*, vol. 38, no. 2, pp. 95–110, 2017.
- [4] M. Breivik, S. Kvaal and P. Østby, ‘From eureka to k-pos: Dynamic positioning as a highly successful and important marine control technology,’ *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 313–323, 2015, 10th IFAC Conference on Manoeuvring and Control of Marine Craft MCMC 2015.
- [5] S. Kvaal and P. Østby, *The Jewel in the Crown - Kongsberg dynamic positioning systems 1975 - 2015*, First edition. Pax, 2015, ISBN: 9788253037882.
- [6] P. Ø. Stig Kvaal and M. Breivik, ‘Dp and the art of perfect positioning,’ *Modeling, Identification and Control (MIC Journal)*, vol. 43, no. 4, pp. 141–159, 2022.
- [7] E. M. Jens Glad Balchen Nils Albert Jenssen and S. Sælid, ‘A dynamic positioning system based on kalman filtering and optimal control,’ *Modeling, Identification and Control (MIC Journal)*, vol. 1, no. 3, pp. 135–163, 1980.
- [8] O. M. Faltinsen, *Sea Loads on ships and offshore structures*, First edition. Cambridge University Press, 1993, ISBN: 9780521458702.
- [9] Kystverket. ‘Radionavigasjon (dgps),’ Accessed: 23rd Jul. 2025. [Online]. Available: <https://www.kystverket.no/navigasjonstjenester/dgps---radionavigasjon/>.
- [10] Kystverket. ‘Kystverket avvikler dgps-tjenesten i norge,’ Accessed: 23rd Jul. 2025. [Online]. Available: <https://www.kystverket.no/nyheter/2024/kystverket-avvikler-dgps-tjenesten-i-norge/>.
- [11] R. R. Per Erlien Dalløkken and E. T. Lygre. ‘Norges største ingeniørbragd: Disse to fikk desidert flest stemmer,’ Accessed: 7th Oct. 2025. [Online]. Available: <https://www.tu.no/artikler/norges-storste-ingeniorbragd-disse-to-fikk-desidert-flest-stemmer/223409>.

- [12] N. M. Bargouth, ‘Dynamic positioning, system identification and control of marine vessels,’ Available at <https://openarchive.usn.no/usn-xmlui/handle/11250/3000363>, Master’s thesis, University of South-Eastern Norway (USN), 2022.
- [13] A. T. B. Nuruddin, ‘Dynamic positioning, system identification and control of marine vessels - based on balchen model,’ Available at <https://openarchive.usn.no/usn-xmlui/handle/11250/3134586>, Master’s thesis, University of South-Eastern Norway (USN), 2024.
- [14] M. Giraldo, ‘System identification and dynamic positioning of ships,’ Available at <https://openarchive.usn.no/usn-xmlui/handle/11250/3074567>, Master’s thesis, University of South-Eastern Norway (USN), 2023.
- [15] T. Fossen, S. Sagatun and A. Sørensen, ‘Identification of dynamically positioned ships,’ *Control Engineering Practice*, vol. 4, no. 3, pp. 369–376, 1996.
- [16] T. I. F. Tor Arne Johansen and P. Tøndel, ‘Efficient optimal constrained control allocation via multi-parametric programming,’ *Journal of Guidance Control and Dynamics*, vol. 28, no. 3, pp. 506–515, 2005.
- [17] T. I. F. Tor Arne Johansen and S. P. Berge, ‘Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming,’ *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 211–216, 2004.
- [18] T. I. F. Robert Skulstad Guoyuan Li and H. Zhang, ‘Constrained control allocation for dynamic ship positioning using deep neural network,’ *Ocean Engineering*, vol. 279, 2023.
- [19] J. R. Taylor, *Classical Mechanics*, First edition. University Science Books, 2005, ISBN: 9781891389221.
- [20] S. of Naval Architects, M. E. ( Technical and R. C. H. Subcommittee, *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid: Report of the American Towing Tank Conference*. Society of Naval Architects and Marine Engineers, 1950.
- [21] N. Oceanic and A. A. (NOAA). ‘What is a current?’ Accessed: 5th Oct. 2025. [Online]. Available: <https://oceanservice.noaa.gov/facts/current.html>.
- [22] D. D. Ruscio, *Optimal model based control - system analysis and design*, Available at: [http://www.davidr.no/iiav3017/syllabus/syllabus\\_advanced.html](http://www.davidr.no/iiav3017/syllabus/syllabus_advanced.html), Lecture notes, Sep. 2025.
- [23] D. D. Ruscio, *Discrete lq optimal control with integral action*, Available at: [http://www.davidr.no/iiav3017/syllabus/syllabus\\_advanced.html](http://www.davidr.no/iiav3017/syllabus/syllabus_advanced.html), Lecture notes.
- [24] D. D. Ruscio, *State estimation and the kalman filter*, Available at: [http://www.davidr.no/iia2217/pensum\\_2003.html](http://www.davidr.no/iia2217/pensum_2003.html), Lecture notes.
- [25] S. Linge and H. P. Langtangen, *Programming for Computations - Python*, Second edition. Springer, 2019, ISBN: 9783030168766.

- [26] R. Sharma, *Lecture notes for the course iia 4117: Model predictive control*, Available at: [http://www.davidr.no/iia2217/pensum\\_2003.html](http://www.davidr.no/iia2217/pensum_2003.html), Lecture notes, Oct. 2019.
- [27] M. Misin, *Applied control systems 2: Autonomous cars (360 tracking)*, Available at: <https://www.udemy.com/course/applied-control-systems-2-autonomous-cars-360-tracking/>, Online course, 2025.
- [28] M. Misin, ‘Model predictive controller of a uav using the lpv approach,’ Available at <https://upcommons.upc.edu/entities/publication/af5025b4-5181-40da-9dc0-17415e18975a>, Master’s thesis, Polytechnic University of Catalonia, 2020.
- [29] D. D. Ruscio, *Subspace system identification - theory and applications*, Available at: [http://www.davidr.no/iia2217/pensum\\_2003.html](http://www.davidr.no/iia2217/pensum_2003.html), Lecture notes, Sep. 2022.
- [30] D. D. Ruscio, ‘Closed and open loop subspace system identification of the kalman filter,’ *Modeling, Identification and Control (MIC Journal)*, vol. 30, no. 2, pp. 71–86, 2009.
- [31] M. Matsumoto and T. Nishimura, ‘Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator,’ *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, 1998.



# Appendix A

## Programming code

All the code developed during this project has been written in MATLAB<sup>1</sup>. This is an excellent language for dealing with mathematical simulation and control. As the name implies, it has many features for dealing with matrices in an intuitive and computationally efficient manner.

The entire body of code developed and used for this thesis has been published on GitHub and is publicly available as of the time of writing this thesis<sup>2</sup>. If for some reason it were to be made a private repository in the future, the author intends to give access to the interested reader upon request.

Some parts of the code depend on libraries written by other people as listed below:

- **Marine System Simulator (MSS):** A Library developed and maintained by Thor Inge Fossen. It contains a wide range of useful functions. This thesis uses the `osv()` and `supply()` functions from the `MSS` library.
- **Deterministic and Stochastic system identification and Realization (DSR):** A state of the art system identification toolbox written and developed by Dr. David Di Ruscio. Instead of applying optimization routines such as the classical Prediction Error Methods (PEM), it finds a solution by doing matrix manipulations.
- **Pseudo-Random Binary Signal (PRBS):** Uses a function developed by David Di Ruscio, named `prbs1()`. The function creates a rich excitation signal that helps generate good data for system identification purposes.

The general code structure for simulations is divided into two main files. It is the simulation file itself that contains the overall structure of the program, and there are different scenario files that can be run for each simulation. Splitting the simulation code and the

---

<sup>1</sup>MATLAB: MATrix LABoratory

<sup>2</sup>The GitHub project is available under: [https://github.com/GoblinTower/DP\\_master](https://github.com/GoblinTower/DP_master). It is necessary to download the `MSS` library since some of the function calls are defined within this toolbox. If the reader has any questions about the code or has found errors, feel free to contact the author.

configuration data allows for running many different scenarios without altering the simulation code itself. This makes for a cleaner code structure and makes it easy to save or load scenarios of interest. An example of this, for the thruster allocation simulations, is shown in Figure A.1.

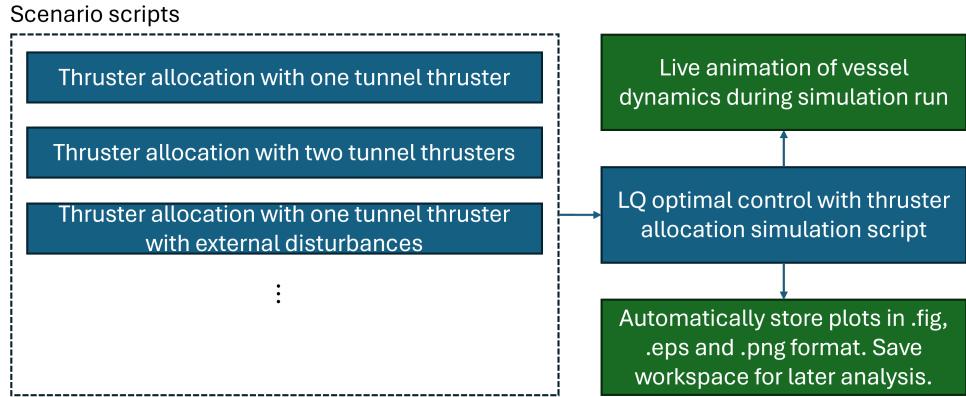


Figure A.1: Depicts general structure of how the different simulations are structured.

There are four classes in the code that display live data during simulation:

- **AnimateKalman:** Displays real and estimated position and vessel heading. Also displays the setpoint (latitude, longitude, and heading). The view is shown in Figure A.2.
- **AnimateForces:** Displays a visual representation of the net thruster forces and torque in the plot. The view is shown in Figure A.3.
- **AnimateThrusters:** Displays a visual representation of each individual thruster contribution. The view is displayed in Figure A.4.
- **AnimateCombined:** Includes information from both **AnimateForces** and **AnimateThrusters** together with external disturbances acting on the vessel and the total net forces and torque (disturbances + thrusters). This plot is shown in Figure A.5.

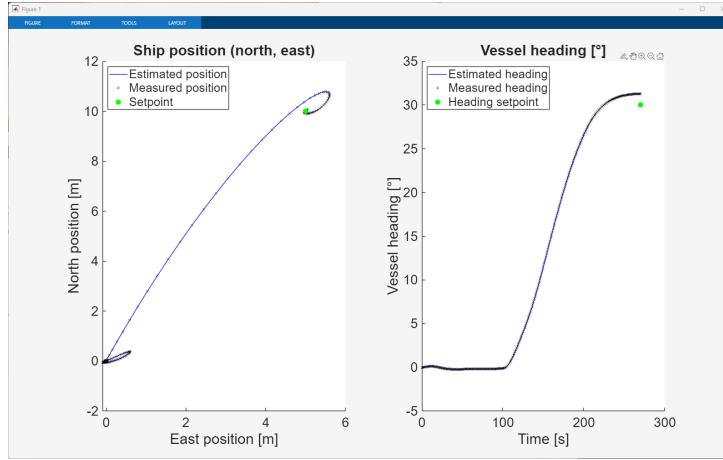


Figure A.2: **Left:** Depicts the position, both real and estimated. Also includes position setpoint. **Right:** Depicts the vessel heading, both real and estimated. Also includes heading setpoint.

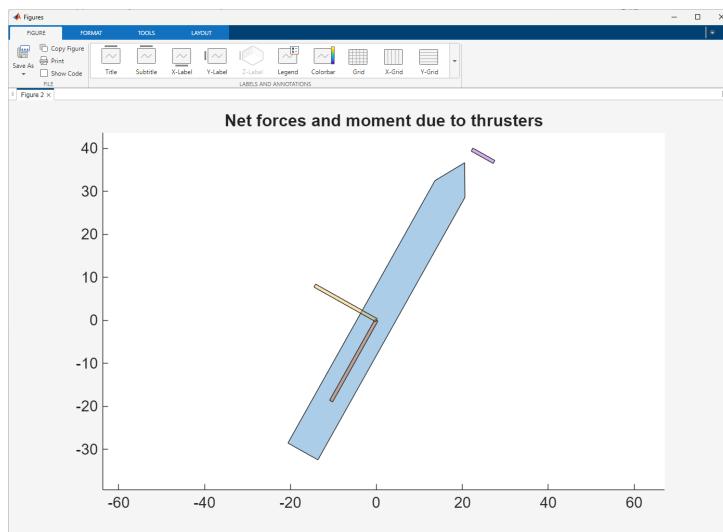


Figure A.3: Depicts net forces in surge and sway together with torque in yaw calculated by the controller algorithm. Forces are shown as scaled rectangles along the surge and sway axes. The torque is displayed as a scaled rectangle at the bow of the ship.

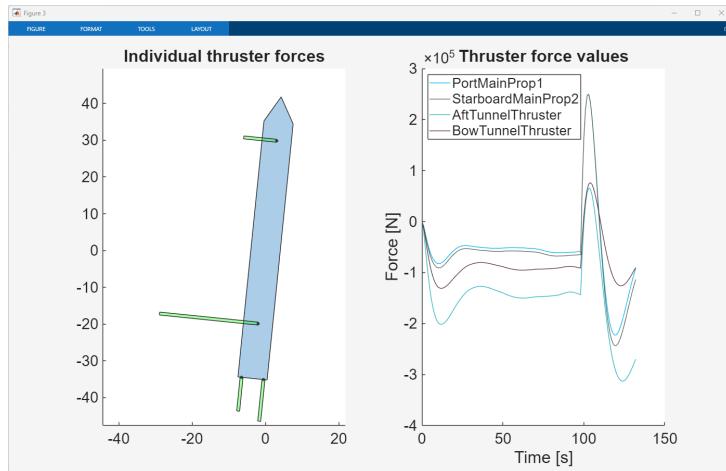


Figure A.4: **Left:** Depicts each individual scaled thruster force as a rectangle pointing in the direction of the force. **Right:** Depicts the individual thruster forces in a line plot.

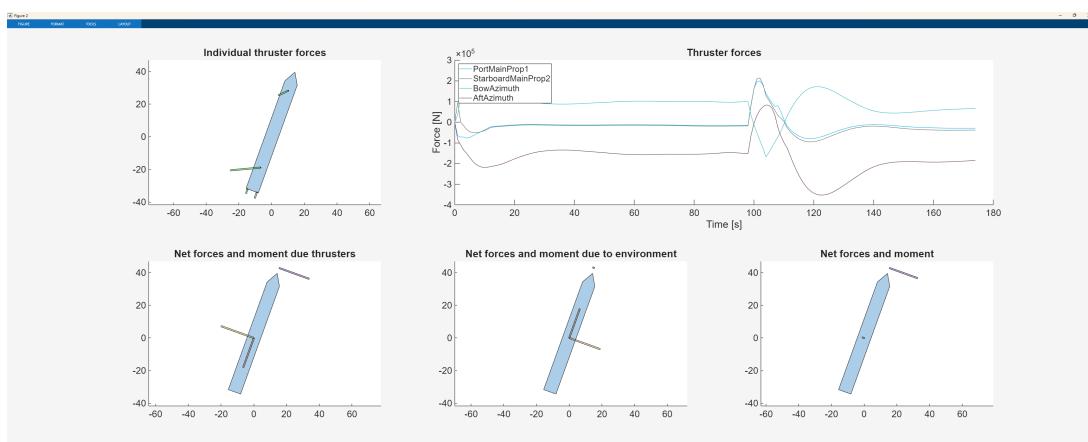


Figure A.5: **Top left:** Depicts each individual scaled thruster force as a rectangle pointing in the direction of the force. **Top right:** Depicts the individual thruster forces in a line plot. **Bottom left:** Net forces and torque in sway, surge, and yaw due to thrusters. **Bottom middle:** Net forces and torque in sway, surge, and yaw due to environmental forces. **Bottom right:** Total net forces and torque in sway, surge, and yaw.

# **Appendix B**

## **Video demonstrations**

Some simple videos have been generated to show how the simulations can be visualized live during simulation. The first video depicts setpoint tracking with regard to LQ optimal control with and without external disturbances as discussed in Section 3.1. The second video shows live thruster allocation during simulation for all the five configurations discussed in Section 3.7.

- LQ optimal control: <https://www.youtube.com/watch?v=ts05r7WjeDE>
- Thruster allocation: <https://www.youtube.com/watch?v=8j7m005wcr8>



# Appendix C

## Parameter values

Here follows a list of the most important parameter values used during the different simulations.

### C.1 Wind model

The wind model discussed in this thesis uses the parameter values given in Table C.1.

Table C.1: Summary of wind parameter values used in simulations.

Parameter	Description	Value	Unit
$\rho$	Air density at 10° C	1.237	$\frac{kg}{m^3}$
$L_{oa}$	Length overall	76.2	m
$A_f$	Frontal projected area	180.0	$m^2$
$A_l$	Lateral projected area	311.0	$m^2$
$C_x$	Wind coefficient (surge)	0.7	—
$C_y$	Wind coefficient (sway)	0.825	—
$C_n$	Wind coefficient (yaw)	0.125	—

### C.2 Balchen model

The Balchen model uses the parameters in Table C.2. These parameter values are based on work done in an earlier master's thesis [14].

Table C.2: Balchen model parameters and associated values.

Parameter	Description	Value	Unit
$d_1$	Drag coefficient (surge)	$5 \cdot 10^{-5}$	$\frac{N}{(m/s)^2}$
$d_2$	Drag coefficient (sway)	$21 \cdot 10^{-5}$	$\frac{N}{(m/s)^2}$
$d_3$	Moment coefficient (yaw)	$1.1 \cdot 10^{-10}$	$\frac{N.m}{(rad/s)^2}$
$d_4$	Moment coefficient (yaw)	$201 \cdot 10^{-15}$	$\frac{N.m}{(m/s)^2}$
$m_1$	Inertial coefficient (surge)	$4 \cdot 10^6$	$kg$
$m_2$	Inertial coefficient (sway)	$4 \cdot 10^7$	$kg$
$m_3$	Inertial coefficient (yaw)	$4.7 \cdot 10^{10}$	$kg.m^2$

### C.3 Kalman filter

The Kalman filter is always tuned using the process and measurement covariance matrices shown in equation (C.1).

$$W = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{12} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 10^{12} \end{bmatrix} \quad V = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix} \quad (C.1)$$

### C.4 LQ optimal control (MPC)

The LQ optimal control on deviation form uses the following weighting matrices shown in equation (C.2).

$$Q = \begin{bmatrix} 10^9 & 0 & 0 \\ 0 & 10^9 & 0 \\ 0 & 0 & 10^{11} \end{bmatrix} \quad P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (C.2)$$

## C.5 Model predictive control

The model predictive control (MPC) strategies use an horizon of 20 samples, spanning 20 seconds into the future ( $\Delta t = 1.0$ ). The weighting matrices are shown in equation (C.3).

$$Q = \begin{bmatrix} 10^5 & 0 & 0 \\ 0 & 10^5 & 0 \\ 0 & 0 & 10^7 \end{bmatrix} \quad P = \begin{bmatrix} 10^{-4} & 0 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix} \quad (\text{C.3})$$

## C.6 Nonlinear Model predictive control (NMPC)

The nonlinear model predictive control (NMPC) strategy uses an horizon of 20 samples that spans 20 seconds into the future ( $\Delta t = 1.0$ ). The weighting matrices are shown in equation (C.4).

$$Q = \begin{bmatrix} 5 \cdot 10^8 & 0 & 0 \\ 0 & 5 \cdot 10^8 & 0 \\ 0 & 0 & 10^9 \end{bmatrix} \quad P = \begin{bmatrix} 10^{-4} & 0 & 0 \\ 0 & 10^{-4} & 0 \\ 0 & 0 & 10^{-6} \end{bmatrix} \quad (\text{C.4})$$

## C.7 System identification

The weighting matrix in the three double integrator identification algorithm, see equation (2.143), is shown in equation (C.5).

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad (\text{C.5})$$

## C.8 Tracking

The simple path-following algorithm uses two parameters:

- The parameter  $d$ . It represents the Euclidean distance to the active waypoint that the vessel must be within to pass to the next waypoint. This is set to  $d = 5$ .
- The parameter  $w$ . This is the maximum heading angle difference between the vessel and the setpoint to allow the vessel to pass to the next waypoint. It is set to  $w = 2^\circ$ .

## C.9 Green DP

The parameters in the green DP cost function are set to  $W_i = 10^2$  and  $W_o = 10^{14}$ . The tuning parameters of the normal DP controller use the same values as the NMPC discussed above. The reduction factor is set to  $\alpha = 0.9$ .

## C.10 Thruster allocation

The different parameters are listed for each algorithm below:

- **Explicit solution:** This solution uses an identity weighting matrix ( $W = I$ ).
- **QP solution:** This solution uses the weights shown in equation (C.6).

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} 10^8 & 0 & 0 \\ 0 & 10^8 & 0 \\ 0 & 0 & 10^8 \end{bmatrix} \quad \beta = 1 \quad (\text{C.6})$$

The limits of the individual thruster forces are set to  $f_{\min} = -[10^8, 10^8, 10^{10}]^T$  and  $f_{\max} = [10^8, 10^8, 10^{10}]^T$  unless otherwise specified. This is for the configuration with only three thrusters, but the four thruster setup is simply a straightforward extension of this setup.

- **Nonlinear programming approach:** The weights are shown in equations (C.7) and (C.8).

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q = \begin{bmatrix} 10^8 & 0 & 0 \\ 0 & 10^8 & 0 \\ 0 & 0 & 10^8 \end{bmatrix} \quad (\text{C.7})$$

$$\Omega = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad P = [10^4 \quad 10^4 \quad 10^4 \quad 10^4]^T \quad (\text{C.8})$$

The values of  $\rho$  and  $\epsilon$  are set to  $10^3$  and  $10^{-6}$ , respectively. The limits on thruster forces are set to  $f_{\min} = -[10^8, 10^8, 10^{10}, 10^{10}]^T$  and  $f_{\max} = [10^8, 10^8, 10^{10}, 10^{10}]^T$ . The azimuths are allowed to rotate freely, but the rate of angle change is limited by  $10^\circ$  every timestep ( $\Delta t = 1.0$ ) in both directions.

## **Appendix D**

### **Signed agreement with USN supervisor concerning content of thesis**

Below is a signed agreement between the author of this thesis (Tommy Sneltvedt) and the assigned supervisor at USN (Dr. David Di Ruscio). The document was signed 31/01/2025. It describes the main tasks of the project.

The final delivery date of the project was set to 31/10/2025 at 14:00.

## FMH606 Master's Thesis

**Title:** Dynamic Positioning , system identification and control of marine vessels – based on a vessel model

**USN supervisor:** David Di Ruscio

**External partner:** None

**Task background:**

One of the first mathematical models used for Dynamic Positioning (DP) of ships was the [Balchen et al 1980 model](#). This model was used for building a DP system based on Kalman filtering and optimal control. It is of interest to look into existing models for DP and possibly make a modified and simplified model based on three double integrators in surge, sway and yaw (the three moving directions), and from this make a modified DP system. Different models exists so if time check out the [Fossen models](#).

**Task description:**

1. Perform a literature research about DP systems of ships.
2. Models for the use with Dynamic Positioning (DP) control of ships are published in e.g. the [MIC paper](#), the OSV and Supply models by [Fossen's Github](#) tool and the Balchen et al DP system. Implement possibly modified models for DP systems in MATLAB or similar. Give a comparison of such models.
3. Investigate if the DP models may be simplified, and e.g. described reasonably well with three linear Double Integrating Pluss Time Delay (DIPTD) models !
4. Use system identification, e.g. the DSR/DSR\_e algorithms to identify the model. Data may be generated from the existing models, in open or closed loop.
5. Perform simulation experiments.

**Student category:** IIA students

**Is the task suitable for online students (not present at the campus)?** Yes

**Practical arrangements:** None

**Supervision:**

As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

**Signatures:**

Supervisor (date and signature): 31.01.2025 

Student (write clearly in all capitalized letters): Tommy Snelthvedt

Student (date and signature):

31.01.2025 