DEGREE PROJECT IN ELECTRICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2018

Autonomous Docking and Navigation of ships using Model Predictive Control

MAO-WEI NILSSON

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Autonomous Docking and Navigation of ships using Model Predictive Control

MAO-WEI NILSSON

Degree Programme in Electrical Engineering
Date: July 6, 2018
Supervisor: Michael Lundh
Examiner: Jonas Mårtenson
Swedish title: Autonom navigering och tilläggning vid kaj av skepp med användning av Model Predictive Controller
School of Electrical Engineering and Computer Science

Abstract

Autonomous shipping is a coming field, where it will be important to operate a ship without manual intervention. Although there are many issues yet to be solved, not the least the legal ones, it would be interesting to investigate functions that already now would be possible to use in today's ship operation. One such field is autonomous navigation in narrow areas. The purpose of this study is to implement a motion control system to navigate marine vessel autonomously, and a Guidance, Navigation and Control system (GNC) is implemented for docking and navigating vessels. Voronoi diagram is used for generating a waypoint list for waypoint tracking. MPC with integral action is applied to control the vessel for reducing model mismatches and constant disturbance from current and wind. We performed the GNC system for South Harbour of Helsinki, and shown that the vessel is navigated and docked at port. Moreover, we studied the effects of disturbance to keep the controller stabilized and suggested an upper limit for the disturbance.

Sammanfattning

Autonomt framförande av skepp är ett växande fält där det kommer att bli viktigt att manövrera ett fartyg utan manuell ingrepp. Även om det fortfarande finns många problem kvar att lösa, inte minst de juridiska, vore det intressant att undersöka funktioner som redan nu skulle kunna användas i dagens fartygsoperationer. Ett sådant fält är autonom navigering i trånga områden. Syftet med denna studie är att implementera ett rörelsekontroll system för autonom navigering av marina fordon och ett vägledning-, navigations- och kontrollsystem för angöring vid kaj för fartyg. Voronoidiagram används för att generera en lista med koordinater att följa. MPC med integralverkan används för att styra fartyget för att minska inverkan av modellfel och konstanta störning från strömmar och vind. Vi gjorde simuleringar för Södra Hamnen i Helsingfors och visade att fartyget navigeras genom hamnen och angör kajen. Dessutom studerade vi effekterna av störningar för att hålla systemet stabiliserat och föreslog en övre gräns för störningenar från strömmar.

Contents

Chapter 1

Introduction and Background

1.1 Review of Guidance, Navigation and Control system for a marine vessel

1.1.1 Introduction

The first attempt to navigate a marine vessel autonomously was in the 1920s. The US navy battleship New Mexico was steered along the ship's course automatically by controlling of the angle of the rudder [24]. Although Nicolas Minorsky has shown the possibility of future autonomous ship control, his earlier work remains to be unknown for control engineers [3].

A more mature control system, the Primary Guidance, Navigation and Control (PGNC) system, was applied into Apollo guidance and control functions by US NASA [1]. The spacecraft was guided according to its well-predefined course automatically before it reached orbit. During this phase, the astronaut's position was monitored and was compared with the reference course. An updated information was sent back to an on-board command module system (CMS), which is the brain of the PGNC system, in order to control the craft's propulsion system.

Later on, the Guidance, Navigation and Control system (GNC), was commonly applied in motion control of vehicles, spacecraft, aircraft, auto-mobiles and underwater vehicles [10][33][29][14]. In a Mars pinpoint landing project, the GNC system was implemented for guiding a spacecraft to visit accessible sites on Mars and helping to place the vehicle as close as possible to pre-positioned sites [33]. In another

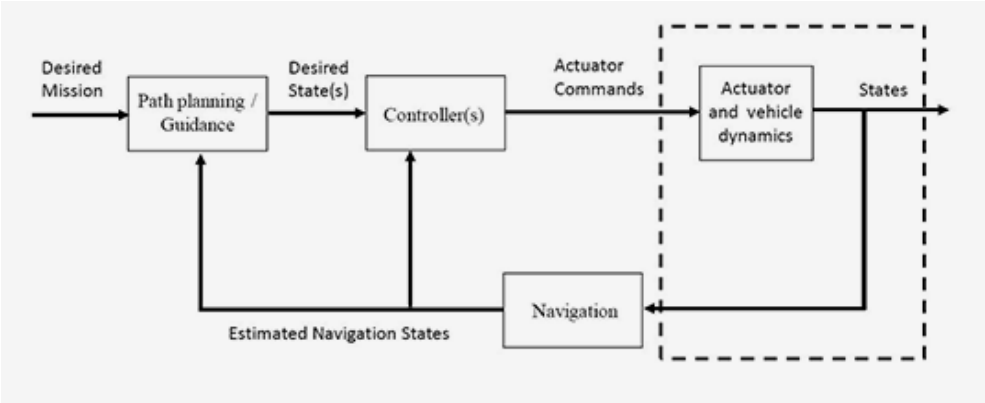

Figure 1.1: Guidance, Navigation and Control system. [10]

Mars Science Laboratory (MSL), the GNC system was applied for reducing the landing area by considering the whole landing process as several phases and decreasing velocity for those phases [29]. In these studies, the GNC system is generally presented as shown Fig. 1.1, which contains a guidance block, controllers, a plant for vehicle dynamics, and a navigation block. The guidance block is for creating trajectory. The controller is for bringing the vehicles to follow the trajectory. The navigation block is for providing more accurate measurement data. In this section, we are going to study the background of each of the blocks.

1.1.2 Guidance System

The guidance system connects controller and navigation parts, according to Fig. 1.1. The inputs of the guidance system usually are the sensor readings such as motion data, weather data, and the outputs of the guidance system usually are the desired position, velocity or accelerations which will be the inputs to the controller for the control system. The guidance system aims to provide a path or trajectory which fulfills some specific requirements, such as minimum time and fuel optimization. There are various guidance law or the strategies used for the guidance system, and most common strategies are target tracking, trajectory tracking and path following [10]. The strategy of target tracking is applied when there is no information about the path or we aim to track a moving object. Given the northeast position of an object

by $p_t^n = [N_t, E_t]^T$, the control objective can be presented as:

$$\lim_{t\to\infty}[p^n(t) - p_t^n(t)] = 0$$

where t is time, $p_t^n(t)$ represents the target path at time t, and $p^n(t)$ is the actual path at time t.

Another strategy, called trajectory tracking, aims to track a predesigned trajectory and to minimise the error between the actual trajectory and the reference trajectory. The control objective can be presented as follows:

$$e(t) := \begin{bmatrix} N(t) - N_d(t) \\ E(t) - E_d(t) \\ \psi(t) - \psi_d(t) \end{bmatrix}$$

where $N(t), E(t)$ and $\psi(t)$ are the actual north, east position and yaw at time t. $N_d(t), E_d(t)$ and $\psi_d(t)$ are the desired north, east position and yaw at time t. The path for such strategy can be generated with simulations or using optimization method to minimise the error.

The path following strategy is similar to the trajectory tracking strategy, but the path is designed so that it does not need to be time dependent.

The path for such strategy can be generated by using line-of-sight (LOS) method, which will, for example, provide a steering law, according to the difference between the actual position and the reference position, as mentioned below and in Fig. 1.2:

$$\psi_d(t) = atan2(y_{los} - y(t), x_{los} - x(t))$$

Another way to generated path or trajectory is by using constrained optimization techniques. In general, it can be formulated as follows:

$$\begin{aligned} & J = min \;\; f(x) \\ & \text{s.t.} \\ & Ax \leq B \\ & h_j(x) \leq 0 \\ & x_{i,min} \leq x_i \leq x_{i,max} \qquad (i = 1, ..., n_x) \end{aligned}$$

where $f(x)$ is the objective function, x is a state vector which could be position and yaw, and h_j is non-linear constraints.

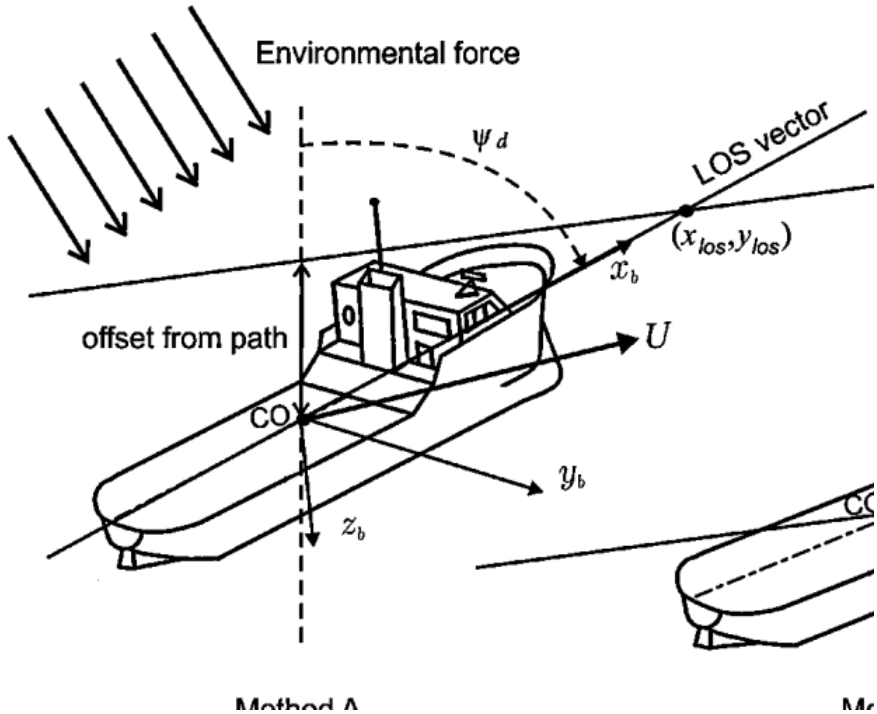

Figure 1.2: Line-Of-Sight method is used for controlling the steering of the vessel by comparing the LOS position and the actual position [10]

These methods used for path following, such as LOS, as well as target tracking that aims to minimizing the distance between the target and vessels. They can be represented as predator and prey, and they are easier to implement. However, disturbances, such as wind and current, might cause the measurement of vessel state become inaccurate, because the vessel is tracking an observed target with the speed and position from previous sampling time. On the other hand, optimization methods for path following or trajectory following might require more computing power. Moreover, not only a better accuracy, but also constraints such as minimum time, fuel optimization, should be taken into account.

The traditional approaches for path planning can be divided into cell decomposition, the roadmap method and potential fields [4]. The cell decomposition is a strategy for keeping on decomposing a map once obstacles appear [19], see Fig. 1.5. Vertices that are separated out will be used for generating waypoints to reach the goal. The roadmap method focuses on creating a path through finding a path in free space [4], and this requires knowledge about the geometry of the environment. Voronoi diagram is studied by Bhattacharya et al. [4] for finding a path for vessel navigation along South American coastline, see Fig. 1.3, and the diagram is generated by a set of points which are the edges of obstacles. The advantage of the Voronoi diagram method is that the maximum clearance path can be generated. However, the path is not necessarily shortest when the maximum clearance path is chosen [4]. For example, in Fig. 1.4, a path in the middle of two polygons is generated, however it is not the shortest path. The potential field method presents potential from obstacles, and high potential will be provided when a vehicle is close to an obstacle.

1.1.3 Navigation System

The states of the vessel should be measured by sensors, such as an inertial measurement unit (IMU), compass or accelerometer, and the navigation system provides measurements to the control system, see Fig.1.1.

The data from sensors often contain noise, thus the measurement data should not be used directly as inputs to the control system. Many navigation system contains technology, that can filter the noise of the data, such as wave filtering. A common technique is to use a low-pass

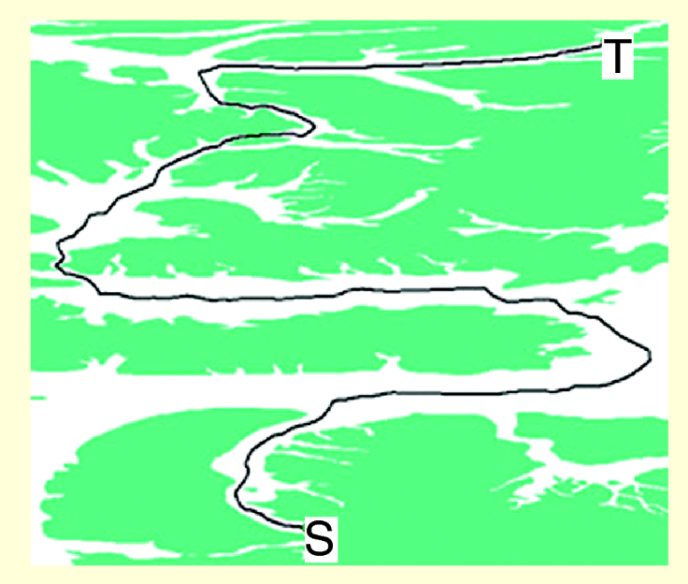

Figure 1.3: A path is generated with Voronoi diagram along the South American coastline [19]. S is start point. T is target. Green presents obstacles. Black line is the path generated using Voronoi diagram.

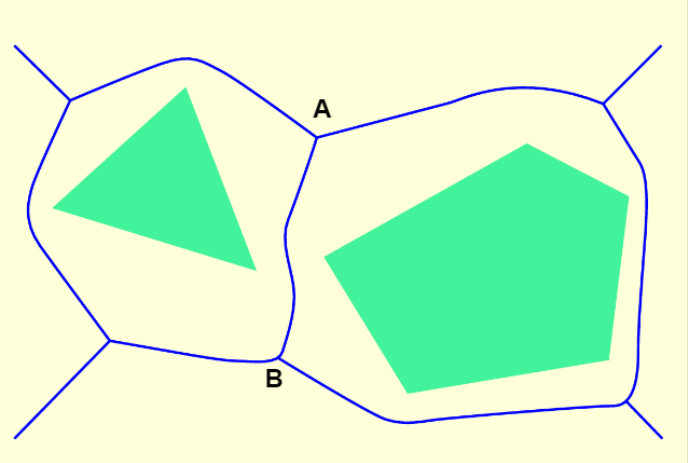

Figure 1.4: Voronoi diagram [19]. From A to B, the shortest path should be a straight line. By using Voronoi diagram, blue paths are generated. Green polygons are obstacles.

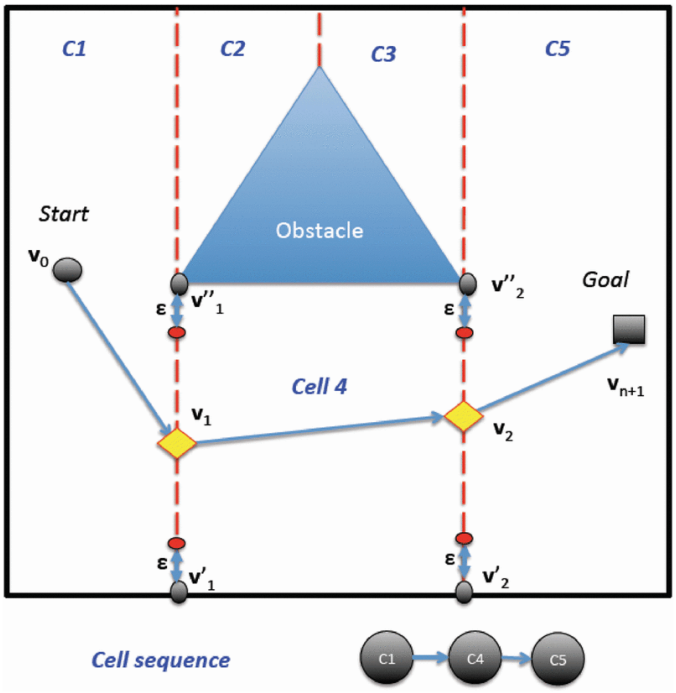

Figure 1.5: The cell decomposition [19].

filter for filtering the wave, assuming that most of the waves are in very high frequency, but we are interested in the actual force on the vessel instead of the wave forces [10]. Another common technique is to use Luenberger Observer to reconstruct the actual states given the measurement of control inputs and outputs of a dynamic system, assuming the noise is white noise. The other common technique is to use the Kalman Filter to iteratively estimate the states of a dynamic system. However, all these techniques require to have an understanding of the dynamic system, and they might suffer the curse of dimensionality when we get high degrees of freedom from the dynamic system [10].

Finding an observer for the navigation system is still a challenging topic. A Kalman Filter was used in [28], and multi-sensor data fusion was employed in the study in order to estimate wave disturbance. In [25], a Luenberger observer was investigated for estimating states of the motion model for marine vessel, so that the model states converged to the states of the plant.

1.1.4 Control System

The control system receives the data from the navigation system and control the vessel so that the vessel can perform as desired performance, see Fig.1.1. There are standard criteria to fulfil when we work on the control system, and those criteria are regarded to stability and maneuverability. In the following section, we are going to discuss the aims and techniques for the control system.

When we design a motion control system, the main purpose is to control the stability and the maneuverability [10]. When we say that a vessel is stable, we mean that the vessel should be able to return back to an equilibrium point after it is disturbed. When we say that a vessel is maneuverable, we mean that a vessel can perform the action given by with a specific maneuver.

When we consider about the stability of a motion control for marine vessel, we focus on the closed loop analysis regards to three characteristics of stability: straight-line, directional and positional motion stability [10]. The vessel is straight-line stable if a vessel that is moving in a straight line can still follow the straight line after a disturbance is applied to it. The similar criteria works for directional and positional motion stability, and we check if the vessel is moving in the same direction and position respectively.

When we consider about maneuverability of a motion control for marine vessel, we focus on if the vessel can move in a specific desired path, i.e. path following control, trajectory, i.e. trajectory-tracking control or if the vessel can reach a specific position, i.e. waypoint-tracking control [10]. When we have path following or trajectory-tracking control, experimental tests could include such as moving a vessel in a circle or in a zigzag path.

When we choose controllers for the motion control, we often take stability and maneuverability into consideration, see [25][36][26][32][11]. The successful results from implementing controller, such as Model Predict Control (MPC), bring interests in way point tracking and path-following control system for an autonomous craft. In [36], the MPC control was used for bringing a spacecraft into orbit, by setting a switch algorithm for using different MPC constraints between docking phase and rendezvous phase. Path following algorithm was used to calculate the cost, and only simple desired paths, such as circle and line, were designed for the simulation, see [25] and [26]. Waypoint tracking

method was used in [9], and MPC was shown to be able to reduce the error between positions for a target, which was an underwater robot, and current state for it. Other methods, such as back-stepping, were also introduced to control an autonomous ship, see [32] and [11]. In the study, the authors presented not only a 3DOF ship model with disturbance from current, wave and wind, but also used system identification to find the parameters for the ship model. Finally, the authors applied these parameters of the designed ship model and successfully demonstrated a simulation of maneuvering of selected ship around a simple path with acceptable accuracy.

Not only the idea of maneuvering a craft, but also the idea of designing a controller so that the craft can be parked or docked at a specific position exit in many applications. In [36], two switched constraints, one has constraints for maneuvering and another has constraints for docking, were applied to a simulated spacecraft. Due to the dimension of the designed spacecraft, orientation was not major roles in designing constraints and objectives to be considered in these spacecraft applications. Instead of using one control strategy, it was proposed in [17] that switching control strategy, between a linear feedforward-feedback and the MPC, can bring the vehicle into a desired parking spot. Instead of using waypoint tracking, path following method was investigated in [34], in order to park a car. However, because the path was pre-defined, only one case, reverse parking, was studied. A multi-rotor micro air vehicle was tested and shown to be able to park on a QR code marked location with PID controller in [5], however wind disturbance was not investigated and specific environment was required, with the QR code in a clear view.

1.1.5 Summary of the GNC system

In this section, we introduced a Guidance, Navigation and Control (GNC) system and we discussed techniques for each of the blocks for the system. For the guidance system, we mentioned about different techniques about generating path. For the navigation system, we mentioned about measuring data of marine motion from different sensors and discussed about obtaining accurate data is still challenging. For the control system, we talked about the main purposes for motion control which are stability and maneuverability control, and we introduced different closed loop analysis that we can use.

1.2 Motivation

To the best knowledge of the author, there are no previous studies about navigating marine vessel with Model Predictive Control, although it is proposed by Fossen [10]. To the best of our knowledge, docking marine vessel has not been reported in previous studies, although there are already studies about docking other crafts such as spacecrafts [36]. Although there are long history of using GNC system, to our knowledge, no existing empirical research addresses GNC with marine vessel for a docking problem.

1.3 Objective

The thesis work is proposed to design an MPC based on a model of the ship that controls a simulated ship from one state (position, angle and velocity) to another state. To be specific, it should simulate a case where a ship is controlled from a given position to its docking position. The GNC system should be implemented for such task. Possible objectives to consider are time, minimal fuel cost, and passenger comfort. The center of gravity of the ship should reach the desired state (presented with 3DOF, surge, sway and yaw) with a tolerance of 5 meters in distance and 5 degree. The influence of disturbances of current, wind, and measurement noise should also be evaluated.

1.4 Outline

The following sections are system modeling, solution, results and discussion. In the system model section, we are going to introduce how to present the nonlinear ship plant. In the solution section, each block of the GNC system will be discussed separately. In the results section, we are going to show how MPC works for navigation problem. We are also going to show the performance of GNC implementation for docking. In this GNC system, a Voronoi diagram is chosen for the guidance system, and MPC is chosen for the control system.

Chapter 2

System Modeling

2.1 Introduction

The modeling of marine craft is based on forces and moments applied on the vessels. There are two theories for describing such dynamics and they are maneuvering theory and sea-keeping theory. For maneuvering theory, it is assumed that there is no wave excitation on the vessels. The maneuvering theory is used to describe the motions of vessels in surge, sway and yaw. For the other theory, sea-keeping theory, it is assumed that there is wave excitation on the vessels and the vessels should have constant course and speed, such as zero speed. The sea-keeping theory is for describing hydrodynamic force from the sea with varying wave excitation.

In the book from Fossen [10], a marine craft model is presented, and this model applies for both the maneuvering theory and the sea-keeping theory. In order to derive the motion model for marine craft, the dynamics of the marine craft need to be studied and the study of the dynamics can be divided into two aspects: kinematics, which is based on geometrical aspects of motion, and kinetics, which is based on forces that caused the motion. In the following section, the marine craft model is going to be derived and discussed based on these two aspects.

2.2 Kinematics

One way to study the dynamics of marine craft is by considering the craft as a point and studying the position and orientation of the point. The studies on kinematics are used to describe a set of reference frames and thus a set of coordinate systems of the reference frames, and then find out the relations between acceleration, velocity and position of the point in the system [10]. By choosing the centre of a marine craft as the point, we can study the motion of the point in 6 degrees of freedom, as shown in Figure. 2.1. In general, we have information about the position of the centre of gravity along the northern, eastern and vertical axis of the earth. Then we need to transform this position from the reference frame, so called North, East and Down (NED) $\{n\}$ frame, into the position refer to the body of vessel, so called in body parallel frame $\{b\}$. This transformation can be derived with rotation matrix based on the Euler's rotation theorem, as shown in (2.1) and (2.2).

$$\dot{p}^n_{b/n} = R^n_b(\Theta_{nb})v^b_{b/n} \tag{2.1}$$

$$\dot{\Theta}_{nb} = T_\Theta(\Theta_{nb})\omega^b_{b/n} \tag{2.2}$$

where $p^n_{b/n} = [N\ E\ D]^T$ presents position for the NED coordinate system, $\{n\}$, $\Theta_{nb} = [\phi\ \theta\ \psi]^T$ presents Euler angles between the coordinates systems $\{n\}$ and $\{b\}$, and $v^b_{b/n} = [u, v, w]^T$ and $\omega^b_{b/n} = [p, q, r]^T$ present the linear and angular velocity vectors, shown in Fig. 2.1, in the body fixed reference frame $\{b\}$. Euler angles are the angles caused by rotation of the coordinate system. Furthermore, we have two rotation matrices, $R^n_b(\Theta_{nb})$ and $T_\Theta(\Theta_{nb})$, which are matrices derived by rotation of the coordinate system. We can get the formula for these two rotation matrix by rotating the point along three axis $[x, y, z]$ with angles of ϕ, θ, ψ and multiple three rotation matrices derived from these rotations as follows:

$$R^n_b = R_{z,\psi}R_{y,\theta}R_{x,\phi} \tag{2.3}$$

After multiplication, we get:

$$R^n_b = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi c\phi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{2.4}$$

$$s \equiv sin(.) \qquad c \equiv cos(.)$$

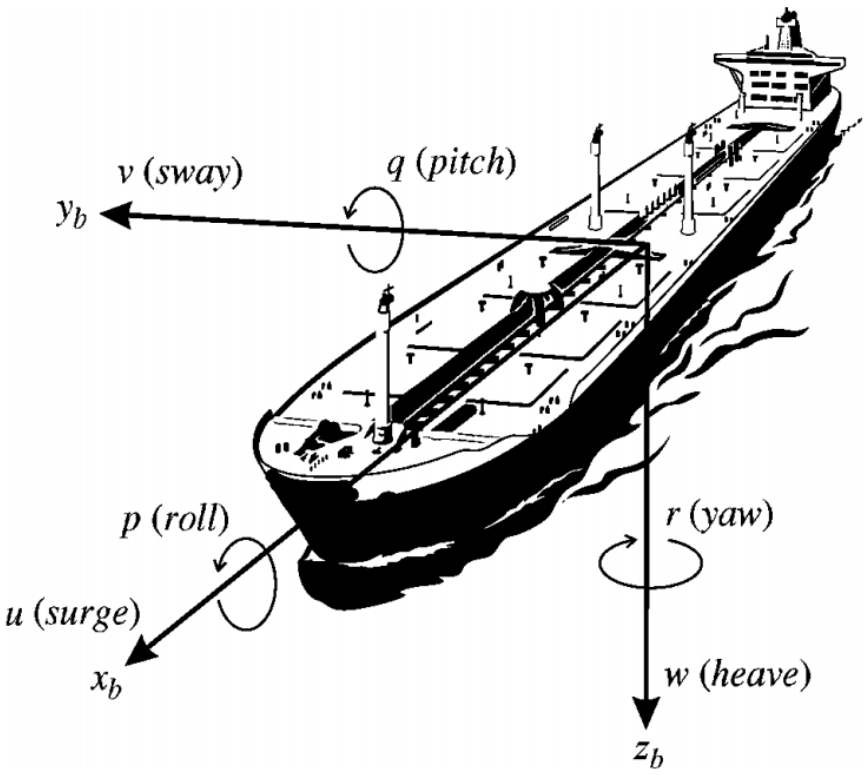

Figure 2.1: The 6DOF velocities presented in the body-fixed reference frame

By using (2.4) and the formula $r^b = R_b^a r^a$ for transforming from reference $\{a\}$ to reference $\{b\}$, we can transform the velocity in vessel parallel frame to NED frame, given (2.1) as follows:

$$\begin{aligned}
\dot{N} &= u\,cos(\psi)cos(\theta) + v[cos(\psi)sin(\theta)sin(\phi) - sin(\psi)cos(\phi)] + \\
&\quad w[sin(\psi)sin(\phi) + cos(\psi)cos(\phi)sin(\theta)] \\
\dot{E} &= u\,sin(\psi)cos(\theta) + v[cos(\psi)cos(\theta) + sin(\phi)sin(\theta)sin(\psi)] + \\
&\quad w[sin(\theta)sin(\psi)cos(\phi) - cos(\psi)sin(\phi)] \\
\dot{D} &= -u\,sin(\theta) + vcos(\phi)sin(\theta) + wcos(\theta)cos(\phi)
\end{aligned} \tag{2.5}$$

where N, E and D are position of center of gravity along the northern, eastern and vertical axis of earth, and $[u,\ v,\ w]$ linear velocities in the body parallel frame, and they are corresponding to surge, sway, heave in Fig. 2.1 respectively.

The rotation matrix for angular velocity is derived in [10], which yields the Euler angle equations in the following:

$$\begin{aligned}
\dot{\phi} &= p + qsin(\phi)tan(\theta) + rcos(\phi)tan(\theta) \\
\dot{\theta} &= qcos(\phi) - rsin(\phi) \\
\dot{\psi} &= q\frac{sin(\phi)}{cos(\theta)} + r\frac{cos(\phi)}{cos(\theta)}, \theta \neq \pm 90^\circ
\end{aligned} \tag{2.6}$$

(2.1) and (2.2) can be re-organized into vector form for presenting 6DOF kinematics, and thus we get the vessel trajectory formula, as shown in (2.7).

$$\begin{aligned}
\dot{\eta} &= J_b^n(\eta)\nu \\
\dot{\eta} &= \begin{bmatrix} \dot{r}_{b/n}^n \\ \dot{\Theta}_{nb} \end{bmatrix} = \begin{bmatrix} R_b^n(\Theta_{nb}) & 0 \\ 0 & T_b^{-1}(\Theta_{nb}) \end{bmatrix} \begin{bmatrix} v_{bn}^b \\ \omega_{bn}^b \end{bmatrix}
\end{aligned} \tag{2.7}$$

where $\eta = [N, E, D, \phi, \theta, \psi]^T$ is a position vector. $\nu = [u, v, w, p, q, r]^T$ is a linear-angular velocity vector. Vessel position in $\{n\}$ yields $r_{nb}^n = [N, E, D]^T$, vessel linear velocity in $\{b\}$ yields $v_{nb}^b = [u, v, w]^T$, vessel angular velocity in $\{b\}$ yields $\omega_{bn} = [p, q, r]^T$, and Euler angles that take $\{n\}$ into $\{b\}$ yields $\Theta_{nb} = [\phi, \theta, \psi]^T$.

2.3 Kinetics

In order to derive the motion of marine craft, we need to study the motion and forces on the vessel. According to [10], we can derive the kinetics by Euler's First and Second Axioms, which are described as followed:

$$\begin{aligned} &\frac{^{i}d}{dt}\vec{p}_g = \vec{f}_g \qquad \vec{p}_g = m\vec{v}_{g/i} \qquad \text{First Axiom} \\ &\frac{^{i}d}{dt}\vec{h}_g = \vec{m}_g \qquad \vec{h}_g = I_g\vec{\omega}_{b/i} \qquad \text{Second Axiom} \end{aligned} \tag{2.8}$$

where $\vec{f}_g$ and $\vec{m}_g$ are forces and moments on the center of gravity of the vessel, $\vec{v}_{g/i}$ and $\vec{\omega}_{g/i}$ are velocity and angular velocity of $\{b\}$ with respect to $\{i\}$.

Given the coordinate transform from $\{i\}$ to $\{b\}$ in Fig. 2.2, translation motion of a point on the vessel rigid body about CG can be presented as follows:

$$\vec{r}_{g/i} = \vec{r}_{b/i} + \vec{r}_g \tag{2.9}$$

where $\vec{r}_g$ is a vector from CO to CG, and $\vec{r}_{b/i}$ and $\vec{r}_{g/i}$ are vectors for CO and CG respect to the reference frame $\{i\}$. Furthermore, we need to apply Euler's equation, which is defined as follows:

$$\frac{^{i}d}{dt}\vec{a} = \frac{^{b}d}{dt}\vec{a} + \vec{\omega}_{b/i} \times \vec{a} \tag{2.10}$$

where $\frac{^{i}d}{dt}$ and $\frac{^{b}d}{dt}$ represents time differentiation in $\{i\}$ and $\{b\}$, respectively. The first term on the right side of the equation is a derivation term, while the second term on the right side of equation is a correction term for presenting the changes of the angular moment during reference frame transformation.

By applying (2.10) to (2.9), we get:

$$\vec{v}_{g/n} = \vec{v}_{b/n} + (\frac{^{b}d}{dt}\vec{r}_g + \vec{\omega}_{b/n} \times \vec{r}_g) \tag{2.11}$$

For rigid body, we have $\frac{^{b}d}{dt}\vec{r}_g = \vec{0}$, thus we have:

$$\vec{r}_{g/n} = \vec{v}_{b/n} + \vec{\omega}_{b/n} \times \vec{r}_g \tag{2.12}$$

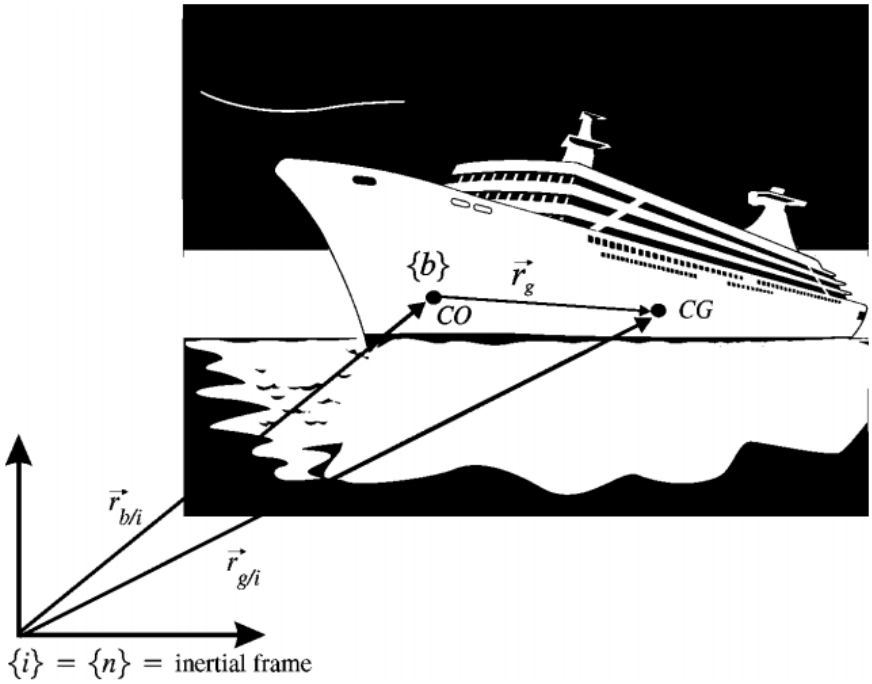

Figure 2.2: The coordinate transform from origins of the vessel, CO, and center of gravity of the vessel, CG.

By using (2.8) and (2.10), it yields that:

$$\begin{aligned}\vec{f}_g &= \frac{^i d}{dt}(m\vec{v}_{g/i}) \\ &= \frac{^i d}{dt}(m\vec{v}_{g/n}) \\ &= \frac{^b d}{dt}(m\vec{v}_{g/n}) + m\vec{\omega}_{g/n} \times \vec{v}_{g/n} \\ &= m(\dot{\vec{v}}_{g/n} + \vec{\omega}_{b/n} \times \vec{v}_{g/n})\end{aligned} \tag{2.13}$$

If we define $S(\gamma)a := \gamma \times a$, we will get translation motion in CG as follows:

$$m[\dot{v}^b_{g/n} + S(\omega^b_{b/n})v^b_{g/n}] = f^b_g \tag{2.14}$$

where $S(\gamma)$ is a multiplication matrix for cross product, and it has the following form:

$$S(\gamma) = \begin{bmatrix} 0 & -\gamma_3 & \gamma_2 \\ \gamma_3 & 0 & -\gamma_1 \\ -\gamma_2 & \gamma_1 & 0 \end{bmatrix} \tag{2.15}$$

We require also rotational motion about CG, and this is based on Second Axioms as mentioned in (2.8). Assume the reference $\{i\}$ is the same with the reference frame $\{n\}$, it yields that:

$$\begin{aligned}\vec{m}_g &= \frac{^i d}{dt}(I_g\vec{\omega}_{b/i}) \\ &= \frac{^i d}{dt}(I_g\vec{\omega}_{b/n}) \\ &= \frac{^b d}{dt}(I_g\vec{\omega}_{b/n}) + \vec{\omega}_{b/n} \times (I_g\vec{\omega}_{b/n}) \\ &= I_g\dot{\vec{\omega}}_{b/n} - (I_g\vec{\omega}_{b/n}) \times \vec{\omega}_{b/n}\end{aligned} \tag{2.16}$$

We apply the same definition in (2.15), we get:

$$I_g\dot{\omega}^b_{b/n} - S(I_g\omega^b_{b/n})\omega^b_{b/n} = m^b_g \tag{2.17}$$

where inertia I_g is defined as:

$$Ig := \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} \tag{2.18}$$

where I_x, I_y and I_z are the moments of inertia along x_b, y_b and z_b in $\{b\}$. I_{xy}, I_{yx},I_{xz}, I_{zx},I_{yz} and I_{zy} are defined in [10]. We can combine (2.17) and (2.14), and we can get vector form for motion about CG as follows:

$$\begin{aligned} M_{RB}^{CG}\begin{bmatrix}\dot{v}_{g/n}^b\\ \dot{\omega}_{b/n}^b\end{bmatrix} + C_{RB}^{CG}\begin{bmatrix}v_{g/n}^b\\ \omega_{b/n}^b\end{bmatrix} &= \begin{bmatrix}f_g^b\\ m_g^b\end{bmatrix} \\ M_{RB}^{CG} &= \begin{bmatrix}mI & 0\\ 0 & I_g\end{bmatrix} \\ C_{RB}^{CG} &= \begin{bmatrix}mS(\omega_{b/n}^b) & 0\\ 0 & -S(I_g\omega_{b/n}^b)\end{bmatrix} \end{aligned} \tag{2.19}$$

Since we often consider origin of the vessel, CO, which is often assumed to be the centre of gravity, as the point for investing motion dynamics, we need to get motion model for CO instead of CG. In order to do this transformation, we need to derive a transformation matrix. According to [10], we can find the transformation matrix by using rotation matrix given by Euler equation, (2.10). Recall (2.12), we have:

$$\begin{bmatrix}v_{g/n}^b\\ \omega_{b/n}^b\end{bmatrix} = \begin{bmatrix}I & S^T(r_g^b)\\ 0 & I\end{bmatrix}\begin{bmatrix}v_{b/n}^b\\ \omega_{b/n}^b\end{bmatrix} = H(r_g^b)\begin{bmatrix}v_{b/n}^b\\ \omega_{b/n}^b\end{bmatrix} \tag{2.20}$$

By inserting (2.20) into (2.19) and multiplying $H(r_g^b)^T$ on each term, we get:

$$H^T(r_g^b)M_{RB}^{CG}H(r_g^b)\begin{bmatrix}\dot{v}_{b/n}^b\\ \dot{\omega}_{b/n}^b\end{bmatrix} + H^T(r_g^b)C_{RB}^{CG}H(r_g^b)\begin{bmatrix}v_{b/n}^b\\ \omega_{b/n}^b\end{bmatrix} = H^T(r_g^b)\begin{bmatrix}f_g^b\\ m_g^b\end{bmatrix}$$

or

$$\begin{aligned} M_{RB}^{CO}\begin{bmatrix}\dot{v}_{b/n}^b\\ \dot{\omega}_{b/n}^b\end{bmatrix} + C_{RB}^{CO}\begin{bmatrix}v_{b/n}^b\\ \omega_{b/n}^b\end{bmatrix} &= H^T(r_g^b)\begin{bmatrix}f_g^b\\ m_g^b\end{bmatrix} \\ M_{RB}^{CO} &= \begin{bmatrix}mI & -mS(r_g^b)\\ mS(r_g^b) & I_g - mS^2(r_g^b)\end{bmatrix} \\ C_{RB}^{CO} &= \begin{bmatrix}mS(\omega_{b/n}^b) & -mS(\omega_{b/n}^b)S(r_g^b)\\ mS(r_g^b)S(\omega_{b/n}^b) & -S((I_g - mS^2(r_g^b))\omega_{b/n}^b)\end{bmatrix} \end{aligned} \tag{2.21}$$

By deriving (2.21), and applying the rule, $S(\gamma)a := \gamma \times a$, we will get the following:

$$\begin{aligned} &m[\dot{v}_{b/n}^b + \dot{\omega}_{b/n}^b \times r_g^b + \omega_{b/n}^b \times v_{b/n}^b + \omega_{b/n}^b \times (\omega_{b/n}^b \times r_g^b)] = f_b^b \\ &I_b\dot{\omega}_{b/n}^b + \omega_{b/n}^b \times I_b\omega_{b/n}^b + mr_g^b \times (\dot{v}_{b/n}^b + \omega_{b/n}^b \times v_{b/n}^b) = m_b^b \end{aligned} \tag{2.22}$$

We introduce the vectors according to [10]:

$$\begin{aligned}
&f_b^b = [X, Y, Z]^T \text{force through } O_b \text{ in } \{b\} \\
&m_b^b = [K, M, N]^T \text{moment through } O_b \text{ in } \{b\} \\
&v_{b/n}^b = [u, v, w]^T \text{linear velocity of } O_b \text{ relative to } O_n \text{ in } \{b\} \\
&m_{b/n)}^b = [p, q, r]^T \text{angular velocity of } O_b \text{ relative to } O_n \text{ in } \{b\} \\
&r_g^b = [x_g, y_g, z_g]^T \text{vector from CO, which is } O_b \text{ here, to CG}
\end{aligned} \tag{2.23}$$

By applying definition of inertia, together with the vectors above, we get 6DOF kinetics motion model as follows:

$$\begin{aligned}
&m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X \\
&m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y \\
&m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = Z \\
&I_x\dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} \\
&+ m[y_g(\dot{w} - yq + vp) - z_g(\dot{v} - wp + ur)] = K \\
&I_y\dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} \\
&+ m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \\
&I_z\dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} \\
&+ m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N
\end{aligned} \tag{2.24}$$

or

$$M_{RB}\dot{v} + C_{RB}(v)v = \tau_{RB}$$

According to [10], the vector, τ_{RB}, represents propulsion forces and moments, and those include hydrodynamic forces, hydrostatic forces, forces from wind and from wave, as follows:

$$\tau_{RB} = \tau_{hyd} + \tau_{hs} + \tau_{wind} + \tau_{wave} + \tau \tag{2.25}$$

where hydrodynamics forces, τ_{hyd}, and hydrostatic forces, τ_{hs}, are defined as:

$$\begin{aligned}
\tau_{hyd} &= -M_A\dot{v} - C_A(v_r)v_r - D(v_r)v_r \\
\tau_{hs} &= -g(\eta) - g_0
\end{aligned} \tag{2.26}$$

where M_A is added mass due to inertia of surrounding fluid, C_A is hydrodynamic Coriolis -Centripetal matrix due to rotation of rigid body, $D(v_r)$ is a damping matrix caused by the vessel moving through

an ideal fluid. We notice that both M_A and C_A are constant matrices depend on the craft, and these two matrix can be approximated by system identification. The hydrostatic forces include gravitational force (buoyancy force) and moments, $g(\eta)$, and pre-trimming forces, g_0, which uses as a ballast to stabilize a vessel. The relative velocities, denoted as v_r, are defined as the difference between vessel velocities and current speed, $v_r = v - r_c$.

2.4 Maneuvering Models for ships (3DOF)

As we mentioned in the previous section, the maneuvering model contains two parts: kinematics and kinetics ship model, by combining (2.7) and (2.24) [31]. For 3DOF model, we only consider the states linear velocities for surge and sway, $[u, v]$, and angular velocity for yaw, r, and we consider only the output for position for north and east, $[N, E]$, and angle for yaw, ψ. That is to say, we assume the motion in heave, roll and pitch are negligible with $w = p = q = 0$. According the study of kinematics before, we can present 6DOF kinematics with the help of rotation matrix, and we assume that $\phi = \theta = \dot{\phi} = \dot{\theta} = 0$ for 3DOF model. If we approximate with $cos(.)$ and $sin(.)$ for small angle and consider only the state for $[u, v, r]^T$, we can rewrite (2.7), and get the 3DOF rotation matrix as follows:

$$R(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

and

$$\dot{\eta} = R(\psi)\nu = R(\psi)[u, v, r]^T$$

According to the study of kinetics, we can present 6DOF kinetics as (2.24), and we consider only the first, second and sixth rows of (2.24) based on our assumption. We consider also hydrodynamic force as mentioned in (2.25) and (2.26), and rewrite (2.24) into the following:

$$\begin{aligned} &(M_{RB} + M_A)\dot{v} + C_{RB}(v)v + (C_A(v_r) + D(v_r))v_r = \tau + \tau_{wind} + \tau_{wave} \\ &\text{or} \\ &M\dot{v} + C_{RB}(v)v + N(v_r)v_r = \tau + \tau_{wind} + \tau_{wave} \end{aligned} \tag{2.28}$$

where v_r is relative velocity which is different between vessel's velocity and current velocity. If we rewrite (2.24) into state space form, and assume the moments of inertia for $I_{xy} = I_{xz} = I_{yz} = 0$, we can get a simplified M_{RB} and C_{RB} as follows:

$$\begin{aligned} M_{RB} &= \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \\ C_{RB} &= \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \end{aligned} \tag{2.29}$$

where m is mass, x_g is length from o_b to CG expressed in $\{b\}$, and $[u, v, r]^T$ are linear and angular velocity of $\{b\}$ relative to $\{b\}$ expressed in $\{b\}$. According to [10], the other term, M_A, C_A and D, are constant matrices and can be approximated by curve fitting. For low speed situations, with surge speed below 4 knots, we can neglect hydrodynamic Coriolis-Centripetal effect and replace the term of C_{RB} and C_A with a biased term, which result to a 3DOF model:

$$\begin{aligned} &\dot{\eta} = R(\psi)\nu \\ &M\dot{\nu} + D\nu = \tau + \tau_{wind} + \tau_{wave} + R(\psi)^T b \end{aligned} \tag{2.30}$$

where b is a constant biased term, which presents disturbance such as current. ν presents speed in surge, sway, and yaw direction. τ, τ_{wave}, and τ_{wind} present the forces from ship rigid body, wave and wind respectively. D is hydrodynamic damping term.

2.5 Thruster allocation

Except the ship model, the dynamic of thruster and propeller should also be taken into consideration. However, the mechanics of such system is complex and need to be simplified. A simplified model is proposed in [10], and force from ship rigid body can be approximated as follows:

$$\dot{\tau} = A_{thr}(\tau_{com} - \tau) \tag{2.31}$$

where τ_{com} is commended thrust, which is a vector with dimension of 3, and A_{thr} can be presented to $diag[\frac{1}{T_{surge}}, \frac{1}{T_{sway}}, \frac{1}{T_{yaw}}]$ with time constants in surge, sway and yaw. Let the commended thrust, τ_{com}, as input, u, we can approximate the thruster dynamic, $\dot{\tau}$, with:

$$\begin{bmatrix} T_{surge} & 0 & 0 \\ 0 & T_{sway} & 0 \\ 0 & 0 & T_{yaw} \end{bmatrix} \dot{\tau} + \tau = u \tag{2.32}$$

$$u = sat(u_{low}, u_0, u_{high}) = \begin{cases} u_{low}, & \text{if } u < u_{low} \\ u_{high}, & \text{if } u > u_{high} \\ u_0, & \text{otherwise} \end{cases} \tag{2.33}$$

where u is force and moment, X, Y and N from (2.23), and it is saturated due to physical limitation.

2.6 Linearisation

In order to stay with a quadratic optimisation problem in the MPC, we would like to linearize model in controller. According to [10], (2.7) can be simplified to:

$$\dot{\eta}_p \approx \nu \tag{2.34}$$

where $\dot{\eta}_p := P^T(\phi)\eta$, and $P(\phi) = \begin{bmatrix} R(\phi) & 0_{2\times2} \\ 0_{2\times2} & I_{2\times2} \end{bmatrix}$ in 3DOF model. With this rotation for η, with NED coordinate, at each time step, we get η_p, with vessel parallel coordinate.

We can also notice that:

$$\begin{aligned} &\dot{\eta}_p = \dot{P}^T(\phi)\eta + P^T(\phi)\dot{\eta} = \\ &\dot{P}^T(\phi)P(\phi)\eta_p + P^T(\phi)P(\phi)\nu = rS\eta_p + \nu \end{aligned} \tag{2.35}$$

where $r = \dot{\phi}$ and for low speed application $r \approx 0$. This leads to a linear system, given by (2.30) and (2.34). The resulting state space model is as follows:

$$\dot{x} = Ax + Bu \tag{2.36}$$

where $x = [\eta_p, \nu]^T$, $u = \tau$, $A = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}D \end{bmatrix}$ and $B = \begin{bmatrix} 0 \\ M^{-1} \end{bmatrix}$. We assume that forces from wind and wave are omitted here.

2.7 Summary of Model for Control

We introduced a 3 DOF nonlinear ship plant model as shown in (2.30) and we will use this nonlinear model for simulating the plant model. We introduced also a linearized model as shown in (2.36) for the control dynamics. We would like to include the thruster allocation in the state space model for the control dynamics, so we get a state space model with 9 states for marine craft, assuming that we have a vessel with low speed. The states of model are a combination of position vector, velocity vector, and forces on thrusters. The state space model is presented as follows:

$$\dot{\boldsymbol{x}}_{9\times1} = \begin{bmatrix} \dot{\eta}_{p3\times1} \\ \dot{\nu}_{3\times1} \\ \dot{\tau}_{3\times1} \end{bmatrix} = \begin{bmatrix} 0_{3\times1} & I_{3\times1} & 0_{3\times1} \\ 0_{3\times1} & -\frac{D}{M} & \frac{1}{M} \\ 0_{3\times1} & 0_{3\times1} & A_{thr} \end{bmatrix} \begin{bmatrix} \eta_p \\ \nu \\ \tau \end{bmatrix} + \begin{bmatrix} 0_{3\times1} \\ I_{3\times1} \\ -A_{thr} \end{bmatrix} u \qquad (2.37)$$

where position vector, η_p, is $[x_p, y_p, \psi_p]$, velocity vector, ν, is $[u, v, r]$, and thruster force vector, τ, is $[\tau_1, \tau_2, \tau_3]$.

2.8 Modelling of Obstacles

When we consider about docking a vessel, one of the topics is obstacle avoidance, and those obstacles can be the harbour, island and other vessels. For example, islands and narrow paths are both needed to be taken into account when a vessel is docked at the port of Helsinki, which is the top European passenger ports [35]. In Fig. 2.3, one of the largest harbour at the port of Helsinki, the South Harbour, has a complicate geographic feature, thus it increases the difficulties for autonomous docking.

In general, there are mainly two types of obstacles, static and dynamic obstacles [15]. Static obstacles can be predicted before the path is planned, while dynamic obstacles will appear during the motion. For a low speed moving vessel, we assume that we have previous knowledge of the environment and thus treat the obstacles as static obstacles. Moreover, there are many different strategies for obstacle avoidance, such as treating obstacles as a node in a grid [15] and [6], or treating obstacles as boundaries [20]. Since we assume that obstacles are static, treating obstacles as boundaries will give less complicate calculation and constraints for our controller.

Treating obstacles as boundaries will provide less overload modelling, however, in most of the cases, boundaries are not symmetrical. As shown in Fig. 2.3, boundaries are not parallel and the narrowness of the path is changing. In order to present the obstacles with one strategy, we can divide the path into several parts, as shown in Fig. 2.4. In Fig. 2.4, we present the same map as south harbour, and we paint the black part as the forbidden area. We divide harbour into three phases and we use red lines to present the boundaries as static obstacle.

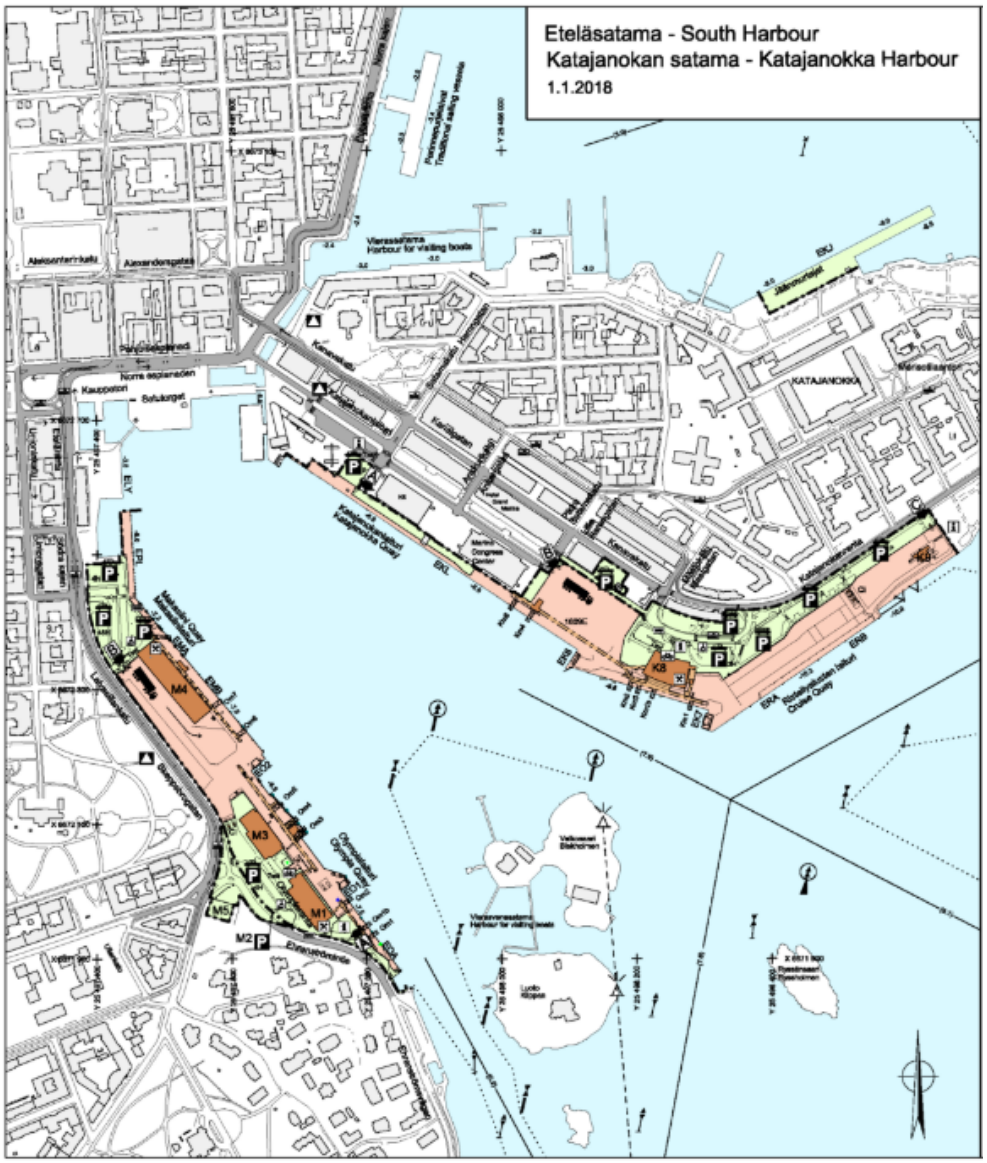

Figure 2.3: South Harbour Technical Map [35]

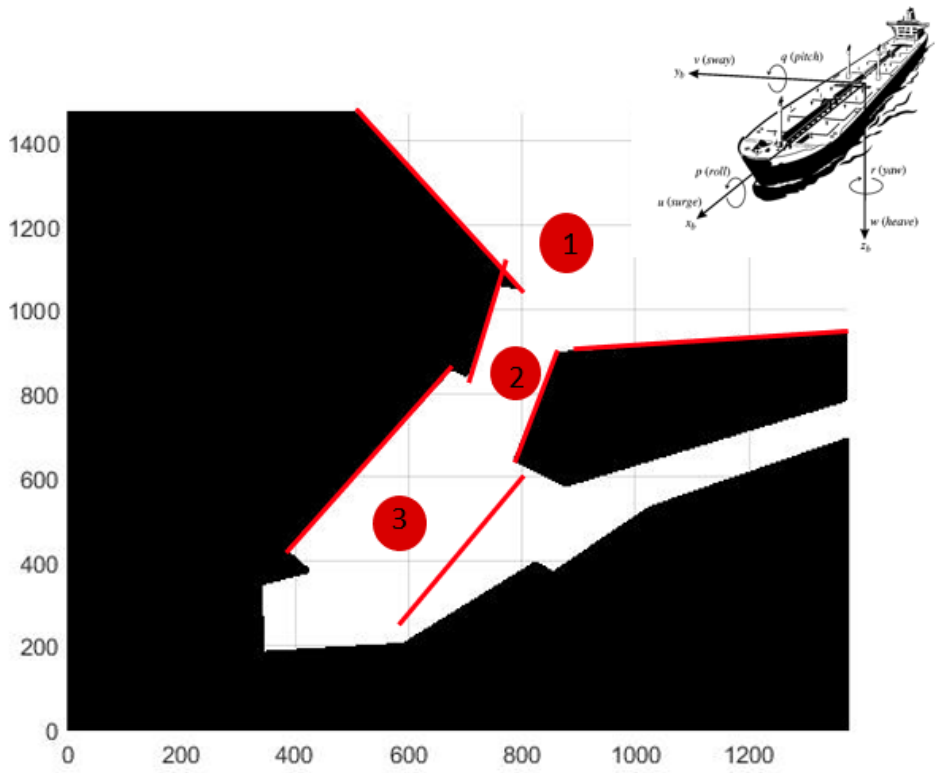

Figure 2.4: Divide South Harbour into Three parts for docking a vessel

Chapter 3

Solutions

3.1 Introduction

We have previously introduced a general closed loop implementation for Guidance, Navigation and Control (GNC) for a marine vessel, and a modified GNC. In this section, we will provide a detailed implementation for docking vessels as shown in Fig. 3.1.

We choose waypoint tracking and Voronoi diagram for the guidance system. We choose Model Predictive Control for the control system. We assume that we can measure the states of the vessel without measurement error, and we choose to have a frame transform and state generator for the navigation system for modelling purposes. In the following section, we are going to explain about our modified GNC system.

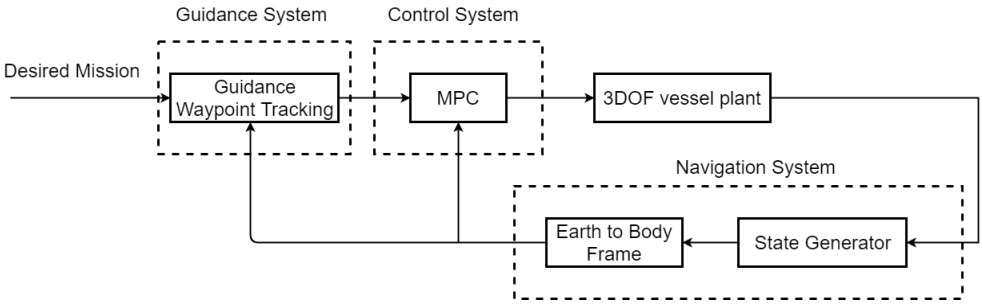

Figure 3.1: The Modified Guidance, Navigation and Control system (GNC)

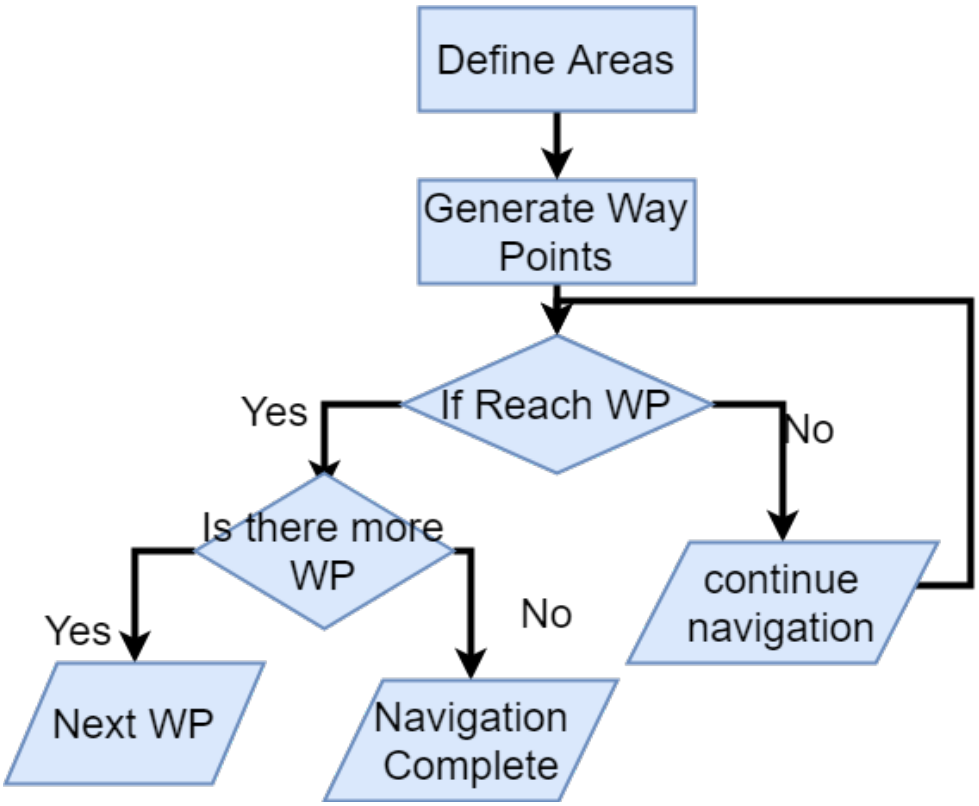

Figure 3.2: Waypoint tracking algorithem.

3.2 The Guidance System

Since there are narrow paths and several non-symmetric static obstacles when a vessel is close to a harbour as discussed in Chapter 2, the guidance system should be modified according to a specific harbour geometric.

It might be difficult to design a path or trajectory in this case, because the path should depend on the specific location of the vessel. Instead, designing several desired position in order to navigate the vessel to a final docking position should be more appreciated.

For waypoint tracking, we will have a list of waypoints, and these waypoints should be updated and become reference points for the controller. Therefore, we need an algorithm for updating such reference/waypoints, as shown in Fig.3.2.

The list of waypoints should be generated automatically according to the geometric map. We choose to generate such list by using Voronoi

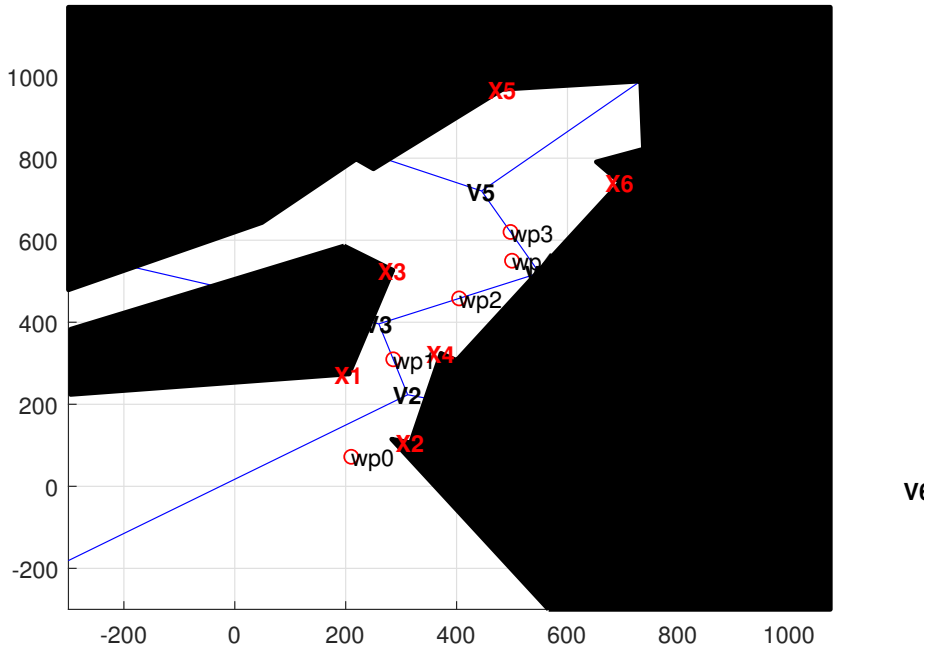

Figure 3.3: Voronoi diagram for harbour in Fig. 2.3. Red markers from x1 to x6 are a set of points for inputs to voronoi diagram. Blue lines are edges for the voronoi diagrams. The markers, v1 to v6, are vertices of the voronoi diagrams. The markers, wp1 to wp3, are calculated as middle of vertices. The markers, wp0 and wp4, are added for control. The desired position is wp4.

diagram, which is used for finding waypoints that has a maximum distance to bounds [2], as shown in Fig. 3.3. Waypoints are the middle points of two vertices of Voronoi diagram. For example, one waypoint is:

$$P = \frac{v_2 + v_3}{2} \tag{3.1}$$

where P is the middle point for vertices 2 and 3, and it is a waypoint.

Two more points are appended to the list of waypoints. We defined two extra bounds before the vessel is driven into the narrow path, because we want to align the vessel towards to the narrow path. An extra waypoint is generated based on these two bounds. Moreover, the vessel needs to reach a desired docking position also and we would like to

have less control efforts, so we append the designed point to the end of waypoints.

The yaw is also required for control inputs, and we choose the slope of one bound that is closest to each waypoint. For example, in Fig. 3.3, yaw for wp1 is the slope of edges with x1 and x3.

3.3 The Control System with Model Predictive Control

3.3.1 Introduction of MPC

For controlling a dynamic system, a control engineer will look for a strategy to find a sequence of control inputs for the system so that the system will behave as it supposed to. There are many types of controllers for finding such strategy, and the choices of controllers could depend on a specific problem. One type of problems is to find an optimized sequence of control inputs to the system, such that the plant can reach a specific reference within finite time and also many constraints should be fulfilled.

To solve such optimization problem, we can formulate and minimise a cost function, which is the difference between the current state of the system and the specific reference. This strategy is called reference tracking, and it can be formulated in the following way:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1}((z_k - r_k)^T Q_1 (z_k - r_k) + u_k^T Q_2 u_k)) + (z_N - r_N)^T Q_f (z_N - r_N) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad i = 1, ..., m \\ & g_j^T x = h_i \quad j = 1, ..., p \end{aligned} \tag{3.2}$$

where k is time index. N is total steps for control sequence. z_k is the state at sample time k. r_k is reference value at sample time k. u_k is the control strategy for at time k. Q_1, Q_2 and Q_f are weights which are used for penalizing.

Sometimes, the problem presented before will require states or control inputs to fulfil some constraints. For example, for marine control, there will be constraints on actuators. Thus, inequality constraints, $f_i(x)$, and linear equality constraints, g_j, are also introduced here.

There are constraints on inputs and states. They could be implemented as hard or soft constraints. Input constraints can be set for limiting the absolute value of the control input, u, or limiting the rate of control inputs, Δu. State constraints are often used for limiting the state of the plant. Hard/soft constraints are another kind of concept where hard constraints are always needed to be fulfilled, while soft constraints may be violated. Moreover, soft constraints are often used to increase the feasibility of an optimization problem.

Instead of solving such optimization problem for infinite time, one way to solve it is by considering a finite time control strategy. This strategy will bring the output of the system at the time, $t + N$, to the reference, $r(t + N)$ from output at the current time, t. Such strategy is called model predictive control (MPC). The future response of the plant is predicted with a discrete-time linear system with state-space representation as follows:

$$x(k+1) = Ax(k) + B(k) \tag{3.3}$$

where $x(k)$ is state and $u(k)$ is input at sample time k. We introduce now the predicted state and input for the next N sample as follows:

$$\boldsymbol{u}(k) = \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \tag{3.4}$$

and

$$\boldsymbol{x}(k) = \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N|k) \end{bmatrix} \tag{3.5}$$

where the notation, $u(k+i|k)$, represents the input at sample time, $k+i$, which is predicted at time, k. The same rules work for $x(k+i|k)$. If we are interested in minimizing state and input for N samples after the sampling time, k, we can formulate the cost function as follows:

$$\min \quad \sum_{k=0}^{N-1} (x(k+i+1|k)^T Q x(k+i+1|k) + u(k+i|k)^T R u(k+i|k)) \tag{3.6}$$

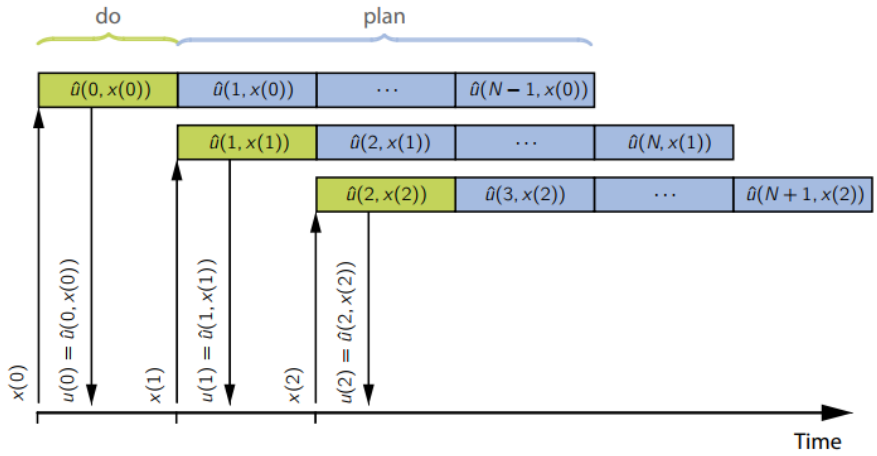

Figure 3.4: Receding Horizon Principle: we use a fixed prediction horizon for getting optimal control sequence. We use only the first from the control sequence. We repeat this process for same fixed prediction horizon every time.

As the aim now is to find an optimized control input sequence, $\boldsymbol{u}^*(k)$, which can minimize the cost function. The main idea of receding horizon is to take only the first control from the sequence of control inputs which is solved by optimization problem. Then we repeat the same strategy for next time period from the time, $k + 1$, and we repeat again. Each time, same length of the input sequence, which is called prediction horizon, is calculated with numerical optimization. The receding horizon principle is shown in Fig. 3.4.

MPC can handle multiple inputs, and rely on some previous knowledge of the dynamics model by using the state space model. Moreover, the numerical optimization problems allow considering the constraints. Thus, MPC has been increasingly applied in industries due to the advantages of being able to do multivariable control and handle constraints.

3.3.2 Prediction and optimization

In the previous sections, we have introduced a typical cost function through (3.2) and motivate the benefits of using MPC. We are going to talk about how MPC predicts and how the cost function is minimized as an optimization problem in this section.

The cost function in (3.6) includes N samples of estimated state over time, and each of the state should be presented based on the known state space model. In order to present the predicted state in (3.4) and (3.5) and assume that we observe our estimated state with $\hat{x}(k|k) = x(k) = y(k)$, we can compute the prediction from (3.3) iteratively as follows:

$$\begin{aligned}
&\hat{x}(k+1|k) = Ax(k) + B\hat{u}(k|k) \\
&\hat{x}(k+2|k) = Ax(k+1) + B\hat{u}(k+1|k) \\
&\quad = A^2x(k) + AB\hat{u}(k|k) + B\hat{u}(k+1|k) \\
&\vdots \\
&\hat{x}(k+H_p|k) = A\hat{x}(k+H_p-1|k) + B\hat{u}(k+H_p-1|k) \\
&\quad = A^{H_p}x(k) + A^{H_p-1}B\hat{u}(k|k) + \cdots + B\hat{u}(k+H_p-1|k)
\end{aligned} \tag{3.7}$$

where H_p is receding horizon, $\hat{x}$ and $\hat{u}$ are chosen to be estimated values of state and input because we assume that observer is not perfect.

In order to present the system from (3.7) in a simplified way, we introduce matrix presentation by:

$$\begin{aligned}
&x(k+i|k) = A^i x(k) + C_i \boldsymbol{u}(k), i = 0, \cdots, N \\
&\text{or} \\
&\boldsymbol{x}(k) = Mx(k) + C\boldsymbol{u}(k) \\
&\text{where } M = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} \text{ and } C = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix}
\end{aligned} \tag{3.8}$$

We can rewrite the cost function in (3.6) by:

$$\begin{aligned}
&J(k) = \boldsymbol{u}^T(k)H\boldsymbol{u}(k) + 2x^TF^T\boldsymbol{u}(k) + x^T(k)Gx(k) \\
&H = C^T\tilde{Q}C + \tilde{R} \\
&F = C^T\tilde{Q}M \\
&G = M^T\tilde{Q}M + Q
\end{aligned} \tag{3.9}$$

where R and Q are penalties.

After that we have rewritten the cost function, we are interested in finding the optimal control inputs, and we can divide this optimization problem into two cases: unconstrained optimization and constrained optimization. For unconstrained optimization, we can find

the optimal by setting the gradient of the cost function equals to zero, if H in (3.9) is non-singular. For constrained optimization, we can find the optimum by quadratic optimization.

3.3.3 Feasibility and stability of MPC

With standard MPC, we might face the situation with no feasibility and no stability guarantee. This means that there is no guarantee to have solution to minimize the cost function, which means it is infeasible, or trajectory diverges, i.e. unstable. Therefore, we need to take feasibility and stability of MPC into consideration when we design MPC.

Instead of having the cost function as (3.6). We usually include a terminal set and rewrite the problem as a finite-horizon LQR problem as follows [18]:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=0}^{N-1} q(\hat{x}_{t+k}, \hat{u}_{t+k}) + q_f(\hat{x}_N) \\ \text{subject to} \quad & \hat{x}_{t+k+1} = A\hat{x}_{t+k} + B\hat{u}_{t+k} \\ & \hat{x}_t = x_t \end{aligned} \tag{3.10}$$

where the notation, $\hat{x}_t$, is the estimated state at time t, while $\hat{u}_t$ is the estimated control action at time t. We use q to present quadratic equation as follows:

$$\begin{aligned} q(x, u) &= x^T Q_1 x + u^T Q_2 u \\ q_f(x) &= x^T Q_f x \end{aligned} \tag{3.11}$$

We say a system is feasible when we can find a set of initial states x for MPC with N such that the states fulfill the constraints and there exists a feasible solution $\{u_0, u_1, \cdots, u_{N-1}\}$ for (3.3).

If our system from a state from our planned problem has a feasible solution, and also the system from every future state for every future time will have feasible solutions, we can call it recursive feasibility. In order to prove if the MPC problem is recursive feasible from all states, we can prove that the terminal set X_f is control invariant [16].

3.3.4 MPC with Integral Action

Since we are using an augmented model for our MPC, there will be a mismatch between the real plant and the model used for the controller.

This will result in static errors [22], which implies that the output does not reach desired reference state even the states stabilized. One of the common solutions for such problem is using integral action [7][27], and this strategy is going to be presented in this section.

In general, integral action to MPC means that we will introduce extra states which are the rate of states instead of states, together with the outputs as our new states, and we use these new states for the MPC. This strategy will argument inputs as $u_k = u_{k-1} + \Delta u_k$, where $\Delta u_k = \Delta u_k^*$, which should be the optimal solution from MPC [23][7].

Instead of using the state space model from (3.3), a real plant model can have unknown disturbance as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + w(k) \\ y(k+1) &= Cx(k+1) + v(k+1) \end{aligned} \tag{3.12}$$

where x and u are system states and input respectively. The measured output is y_k. A, B and C are system matrices. The disturbance v and w might be unknown, and we want to control independent on these unknown disturbances.

If we take one sample times ahead using (3.12), we can get:

$$\begin{aligned} x(k) &= Ax(k-1) + B_u u(k-1) + w(k-1) \\ y(k) &= Cx(k) + v(k) \end{aligned} \tag{3.13}$$

If we subtract (3.12) with (3.13), we get:

$$\begin{aligned} &x(k+1) - x(k) = A(x(k) - x(k-1)) + B_u(u(k) - u(k-1)) \\ &+ (w(k) - w(k-1)) \\ &y(k+1) - y(k) = C(x(k) - x(k-1) + (v(k) - v(k-1) \end{aligned} \tag{3.14}$$

Because x, u, d, and w are assumed to be time invariant, we can introduce the integration as follows:

$$\begin{aligned} \Delta x(k) &= x(k) - x(k-1) \\ \Delta u(k) &= u(k) - u(k-1) \\ \Delta w(k) &= w(k) - w(k-1) \end{aligned} \tag{3.15}$$

Now we apply the general idea of integral action by treating $\begin{bmatrix}\Delta x(k+1)\\ y(k+1)\end{bmatrix}$ as new state, so the extended state-space model is defined as:

$$\begin{bmatrix}\Delta x(k+1)\\ y(k+1)\end{bmatrix} = \begin{bmatrix}A & 0\\ CA & I\end{bmatrix}\begin{bmatrix}\Delta x(k)\\ y(k)\end{bmatrix} + \begin{bmatrix}B_u\\ CB_u\end{bmatrix}\Delta u(k) + \begin{bmatrix}I\\ C\end{bmatrix}\Delta w(k) \tag{3.16}$$

We can rewrite it to:

$$\begin{aligned}\tilde{x}(k+1) &= \tilde{A}\tilde{x}(k) + \tilde{B}_u\Delta u(k) + \Delta\tilde{w}(k)\\ y(k) &= \begin{bmatrix}0 & I\end{bmatrix}\begin{bmatrix}\Delta x(k)\\ y(k)\end{bmatrix} = \begin{bmatrix}0 & I\end{bmatrix}\tilde{x}(k) = \tilde{C}\tilde{x}(k)\end{aligned} \tag{3.17}$$

We apply predicted model as presented in (3.7) and (3.5), we can get prediction of output for each receding horizon for MPC and thus we can find optimal solutions [22] [16] [7]. Moreover, since the output from MPC is Δu, we should apply the relationship mentioned in (3.14) and convert to the integrated u. We introduce a vector $U(k)$ as present and future values of the manipulated outputs over the N receding horizon with:

$$U(k) = \begin{bmatrix}u(k)\\ \vdots\\ u(k+N-1)\end{bmatrix} \tag{3.18}$$

If we apply the relationship according to (3.14), we will have:

$$\begin{aligned}U(k) &= I_L\Delta U(k) + I_M u(k-1)\\ I_L &= \begin{bmatrix}I & 0 & \cdots & 0\\ I & I & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ I & I & I & I\end{bmatrix} \text{ and } I_M = \begin{bmatrix}I\\ I\\ \vdots\\ I\end{bmatrix}\end{aligned} \tag{3.19}$$

Since we only apply the first control inputs, we will have:

$$u(k) = \begin{bmatrix}I & 0 & \cdots & 0\end{bmatrix}\Delta U(k) + u(k-1) \tag{3.20}$$

We can see that we just need to add the first control input rate, Δu, to the previous control input to the real plant.

We might have constraints in our modelling and we want to transfer such constraints for a quadratic programming problem for computation simplicity. However, since we received control input rate, Δu, instead of the control input, u, we need to reformulate the constraints. Assume our constraints are:

$$\begin{aligned} \Delta u_{min} \leq \Delta u \leq \Delta u_{max} \\ u_{min} \leq u \leq u_{max} \end{aligned} \tag{3.21}$$

According to [7], we can rewrite equation above to:

$$\begin{bmatrix} I \\ -I \\ I_L \\ -I_L \end{bmatrix} \Delta u \leq \begin{bmatrix} \Delta u_{max} \\ -\Delta u_{min} \\ u_{max} - I_M u(k-1) \\ -u_{min} + I_M u(k-1) \end{bmatrix} \tag{3.22}$$

Thus we have new formulation that will allow us to set constraints on both control input rates, Δu, and control input, u.

In this section, we motivate the MPC with integral action, and introduce the strategy to transform our original state space model to an extended state space model. We propose also the way to construct constraints for such strategy.

3.3.5 MPC formulation for 3DOF ship model

We sampled the state-space model in (2.37) with zero-order hold and constructed an extended state-space model by using MPC with integral action. Moreover, we are interested in a reference tracking problem, so we rewrite the MPC from (3.2) to following formulation:

$$
\begin{aligned}
&\min \sum_{k=0}^{N-1}((\tilde{C}x(k) - r(k))^T Q_1(\tilde{C}x(k) - r(k)) + \Delta u(k)^T Q_2 \Delta u(k)) \\
&+ (\tilde{C}x_N - r_N)^T Q_f(\tilde{C}x_N - r_N) + \epsilon^T S\epsilon \\
&\text{s.t. } u_{min} - \epsilon \leq \begin{bmatrix} x_{16} \\ x_{17} \\ x_{18} \end{bmatrix} \leq u_{max} + \epsilon \\
&\Delta u_{min} \leq \Delta u \leq \Delta u_{max} \\
&\text{position} \cap \Theta = \emptyset \\
&x(k+1) = \tilde{A}x(k) + \tilde{B}\Delta u(k) \\
&\epsilon >= 0
\end{aligned}
\tag{3.23}
$$

where the vessel state at time k is $x(k)$, which is an extended state in the form of $\begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}$ according to (3.16), and this extended state contains the state rate and output state. Soft constraint is applied in order to increase feasibility when we want to limit violations of states or inputs. Soft-constraint MPC control is nominally stabilizing and Lipschitz continuous, and the method can be used for minimizing the size of violations [37][30]. It is a state with dimension of 18×1, which is $[\Delta\eta\ \Delta\nu\ \Delta\tau\ \eta\ \nu\ \tau]$. The input to the cost function is the control input rate, Δu. We have constant matrix, $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$, according to (3.16), with:

$$
\begin{aligned}
\tilde{A} &= \begin{bmatrix} A & 0 \\ CA & I \end{bmatrix} \\
\tilde{B} &= \begin{bmatrix} B \\ CB \end{bmatrix} \\
\tilde{C} &= \begin{bmatrix} 0 & I \end{bmatrix}
\end{aligned}
\tag{3.24}
$$

where the constant matrix is A, B and C, given by (2.37).

The reference state is for time k is $r(k)$. We choose $r(k) = r_N$ and it is a constant matrix, which has a dimension of 18×1. Since we include the thruster force τ into the states, we choose the limits of the actuators between $[u_{min}\ u_{max}]$ for input directly on the state. We want to limit the input rate for the thruster, and we choose an another bound, between $[\Delta u_{min}\ \Delta u_{max}]$.

3.4 The Navigation System

We have mentioned in the previous chapter that the navigation system is used for providing reference states of the control system. We have a nonlinear plant model, so we will need to find the optimal control input from a nonlinear MPC. However, a nonlinear MPC will give computational burden, and we would like to use a linear MPC. If we try to use MPC with a linearized model directly, i.e. around one operating point, we might not be able to find minima for the cost function over time due to the fact that we have a high nonlinear plant model. Instead, we would like to have a linearized plant model for all time or linearized around different operating points over time, (2.34) provides a possibility as our solution. In this section, we are going to explain this strategy.

3.4.1 Frame Transform

Frame transform is applied for both velocity transform and position transform between the vessel parallel frame and the earth frame. Vessel parallel frame is the frame considering the ship orientation as the origin orientation. In Fig. 3.5, for a vessel from the same position, the coordinate from the vessel parallel frame is, and always is, [0 0 0], while the coordinate for the earth fixed frame is [2 2 45], if we present a position of a vessel with [surge sway yaw].

According to [10], the ship plant model can be approximated as a linear model, given η_p in (2.34) is the position of the vessel in vessel parallel frame. However, the ship plant model always provides the state from the earth fixed frame, see Fig. 3.5. We need to use coordinate transform between vessel parallel frame and earth fixed frame for the closed loop.

Two types of transform functions are used in our studies, the transform function with both rotation and translation matrix, and the transform function with only rotation matrix. Given a position of a vessel in the vessel parallel frame as [0 0 0], we would like to know how this vessel can move to a specific reference point. The position of the specific reference point should be also presented in the vessel parallel frame. In order to present the new position for the reference, we use both translation and rotation matrix as follows:

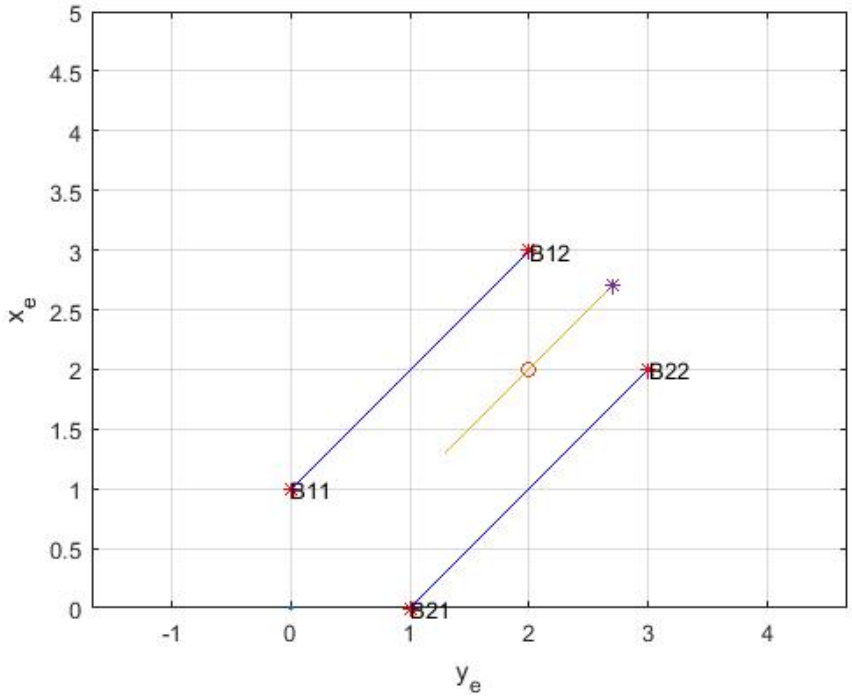

Figure 3.5: Same ship position will give different coordinate in vessel parallel frame and earth fixed frame. For vessel parallel frame, the ship position is [0 0 0], while the ship position is [2 2 45] for the earth fixed frame. Note: the vessel is presented at the yellow line, while the blue lines are two designed bounds. The sign * presents the bow of the vessel, and the other side is the aft of the vessel. The sign o presents the centre of gravity of the vessel.

$$z_b = tf_1(z_e, o_b, \psi) = R(\psi)(z_e - o_b)$$
and
$$R(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

where ψ is the Euler angle, z_b is the position of the vessel in the vessel parallel frame, z_e is the position of the vessel in earth frame, and o_b is the position of the origin for the vessel parallel frame in earth fixed frame.

If we have already a velocity from a vessel parallel frame, and we would like to get this velocity represented in a new vessel parallel frame, we use only translation matrix. This calculation is performed when we would like to find new velocity for the next sample time with MPC. Then the transform function here is:

$$v_{new} = tf_2(v, \psi) = R(\psi)v \tag{3.26}$$

where v is the vector presented in the original frame, and v_{new} is the vector presented in the new frame for the next sample time.

3.4.2 States Generation

We get 9 states, $[x\ y\ \psi\ u\ v\ r\ \tau_1\ \tau_2\ \tau_3]$, from the ship plant model, and we need to build a states generator that prepare 18 states for the MPC with integral action. Since all 9 states from the plant model are in earth fixed frame and MPC based on using states in vessel parallel frame, we need to transform all states into the vessel parallel frame and calculate 18 states from 9 states in the following way:

- $\eta_b(k)$ = [0 0 0]
- $\Delta\eta_e(k) = \eta_e(k) - \eta_e(k-1)$
- $\Delta\nu_e(k) = \nu_e(k) - \nu_e(k-1)$
- $\Delta\tau_e(k) = \tau_e(k) - \tau_e(k-1)$
- $\Delta\eta_b(k) = tf_2(\Delta\eta_e, \psi(k))$
- $\Delta\nu_b(k) = tf_2(\Delta\nu_e, \psi(k))$

- $\Delta\tau_b(k) = \tau_e(k)$
- $\eta_b(k)$ = [0 0 0]
- $\nu_b(k) = tf_2(\nu_e(k), \psi(k))$
- $\tau_b(k) = \tau_e(k)$
- $x = \begin{bmatrix}\Delta\eta_b & \Delta\nu_b & \Delta\tau_b & \eta_b & \nu_b & \tau_b\end{bmatrix}$

3.5 Software and Toolbox

We performed our studies on MATLAB. We used the toolbox YALMIP for building the control system [21].

YALMIP is a toolbox for modelling and solving optimization problems in MATLAB. YALMIP is developed to models semidefinite programing (SDP), and it provides solvers for optimization problems [21]. It supports linear programming (LP), quadratic programming (QP), mixed integer programming, etc. Almost around 20 solvers are supported for solving those problems. It is easy to use YALMIP as a toolbox in MATLAB, and it is easy to implement YALMIP with Simulink. Quadratic programming solver is chosen in our studies.

3.6 Summary

In previous sections, we talked about how we construct our solutions according to the Guidance, Navigation and Control system. In the guidance system, we generated waypoints. In the navigation system, we transformed coordinate systems and generated states for the control system. In the control system, we implemented MPC with integral action. The completed implementation can be presented in Fig.3.1.

Chapter 4

Simulation Results

4.1 The Ship Model

We implemented the 3DOF ship model from Chapter 2 by combining the ship model from (2.28) with thruster allocation dynamics. We simulated this ship model as shown in Fig. 4.1, in order to observe the dynamics of open loop behavior and the response time of each of the degrees of freedom. Given three inputs as commands to the thruster, (2.31), we analysis the models by varying one of the inputs with a constant and keep other two inputs zero.

Given the upper bound values for the thruster as $[4\times10^6$ N 10^6 N 4×10^8 Nm], we select our nonlinear ship plant with three different step inputs with $[4 \times 10^6$ N 0 0], $[0\ 10^6$ N 0], $[0\ 0\ 4 \times 10^8$ Nm] respectively, as shown in Fig. 4.1.

Three tests show that the yaw rate, r, has a shorter response time than the velocity on surge and sway direction.

4.2 Linearized Plant Model without Integral Action

The first part of the study is to approximate a state-space model for MPC, and use a linearized model as an approximation of the plant model. We use MPC toolbox and we choose 3DOF from (2.30) without actuator dynamics from (2.31). We tested same reference values for without integral action and analysis the inputs, trajectory and states, as shown in Fig. 4.2. Given the MPC formulation in (3.2), we choose

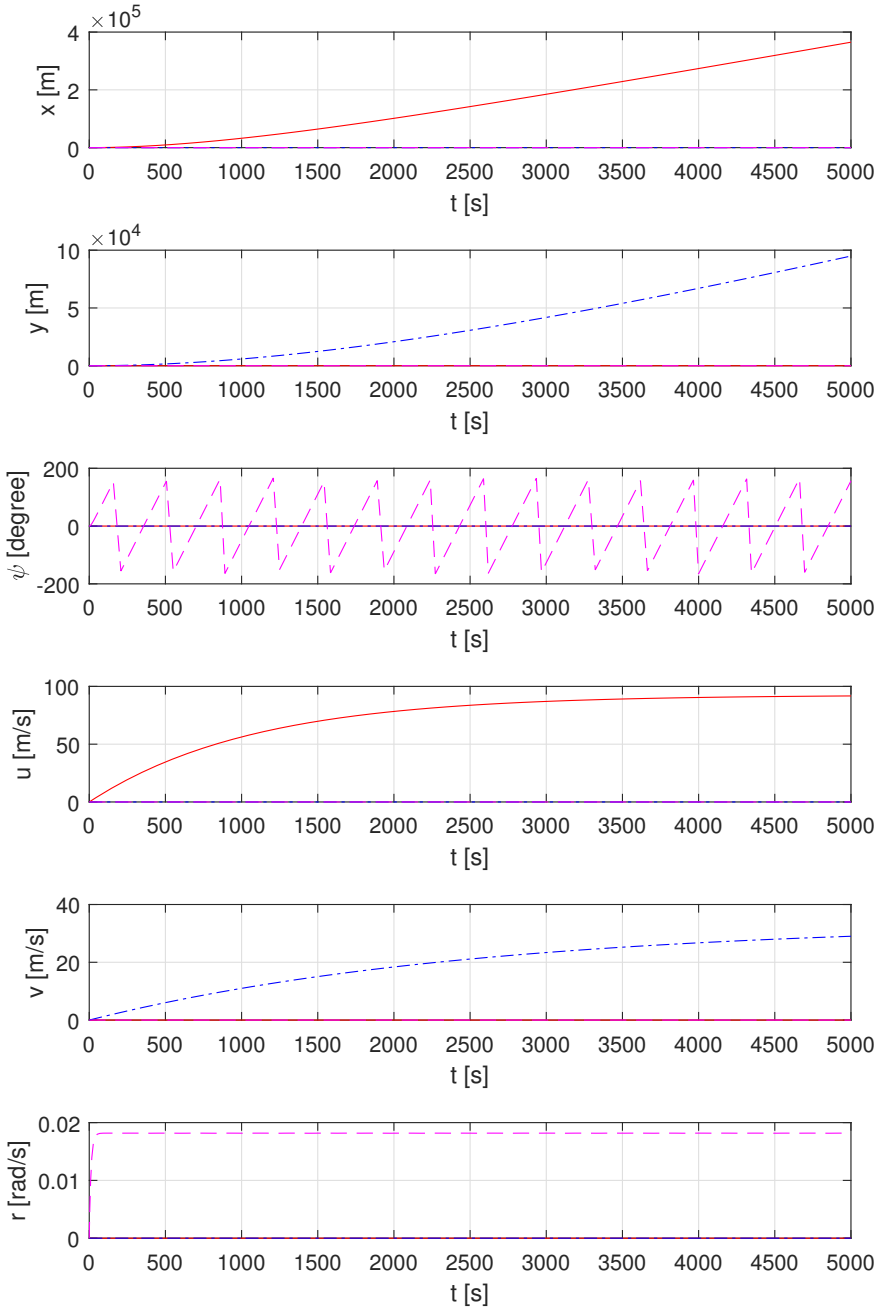

Figure 4.1: States over 5000 seconds. Solid red line presents first test with [1e6 0 0]. Dash dot line presents second test with [0 1e6 0]. Dash line presents third test with [0 0 1e8].

the weights for input and weights for terminal set are zero and the weights for states as follows:

$$Q_1 = diag([10^6\ 10^6\ 10^6\ 1\ 1\ 1]) \tag{4.1}$$

Given the reference input $[50$ m 50 m 50 deg$]$, the results of MPC are shown in Fig. 4.2 and Fig. 4.4. Sample time is chosen to be 10 s for all simulations. Although the inputs stabilized as shown in Fig. 4.3, only yaw reaches the reference values while there are static errors for surge and sway due to mismatch of the system. The surge position only reaches 25m even that the final position for surge should be 50m. Moreover, a large overshoot for surge is obtained with 82 percent, while overshoot for sway is 3.9 percent and for yaw is 0.2 percent. The system is fast enough, and rise time for surge is 14.3 seconds, for sway is 66.1 seconds and for yaw is 66.8 seconds.

4.3 Performance with Integral Action

By using MPC with integral action, we aim to minimize the mismatching between the state space model for MPC and the nonlinear model for the plant, and handle constant disturbance from current and wind.

Instead of using the state-space model for in (2.36), we chose a state-space model with thruster dynamics from (2.37) and included integral action according to (3.17).

$$\begin{aligned} Q_f &= diag([1\ 1\ 10^{10}\ 0\ 0\ 0\ 10^8\ 10^8\ 10^8]) \\ Q_2 &= I_{3\times 3} \\ Q_1 &= 0.05 Q_f \end{aligned} \tag{4.2}$$

We pick weights for terminal set much larger than for states because we are interested in reaching the final states. We choose weights for inputs much larger than other states because changing 1 N of force affects less for a large ship. We choose a large weight for yaw because we would like to change the angle faster than the position.

This MPC with integral action bring the ship into desired position, as shown in Fig.4.6 and Fig.4.7. The plot for states is in Fig. 4.5.

As shown in Fig. 4.6, the system stabilized and converges to the reference. The system does not have much overshoot. There is 0.47

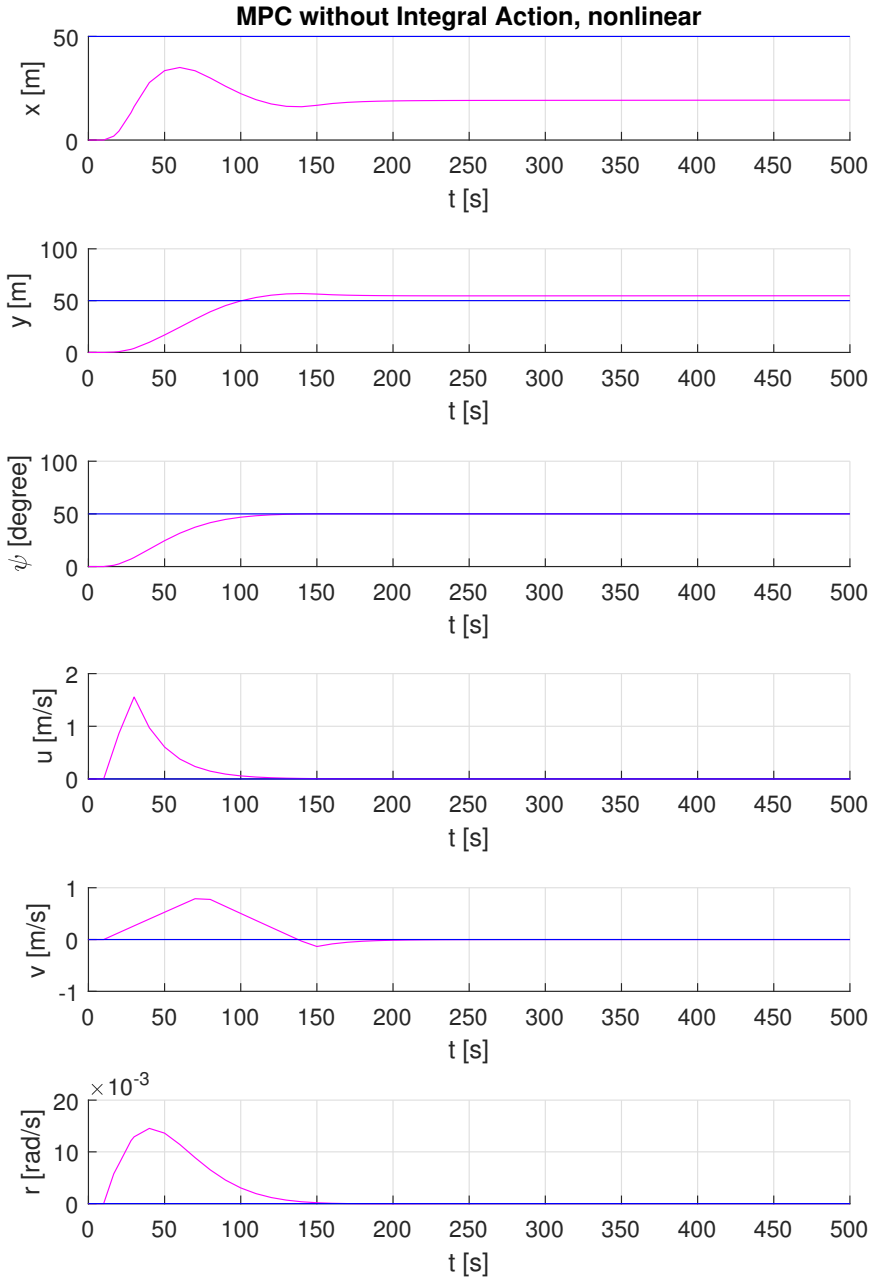

Figure 4.2: States for MPC without integral action. Reference is set to [50 m 50 m 50 deg], and we get static error. Pink solid line presents the system performance, given the reference [50 m 50 m 50 deg]. Blue solid line presents the reference value. The system results in a static error for surge and sway.

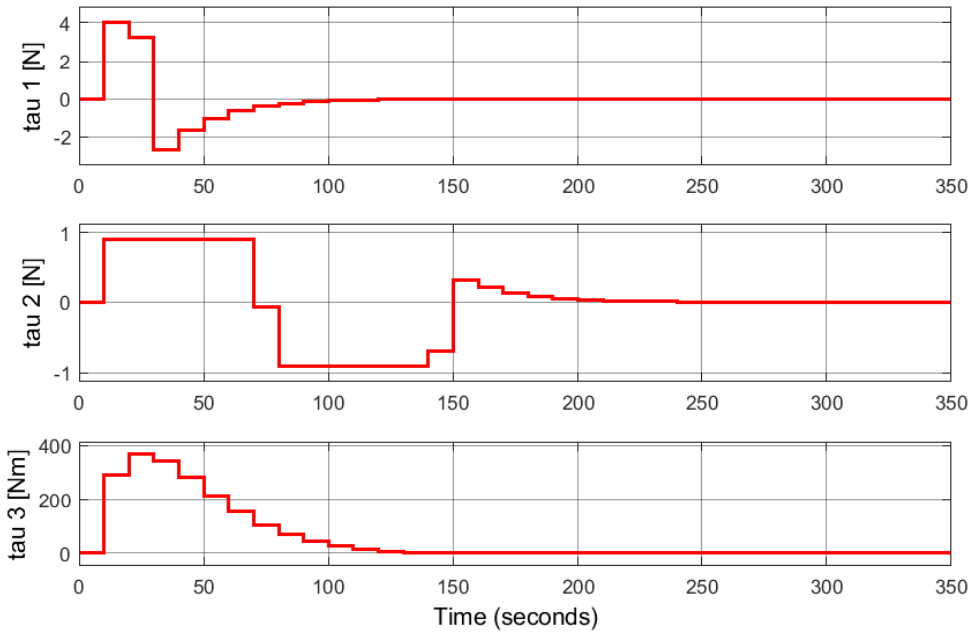

Figure 4.3: Inputs for MPC without Integral Action with Reference [50 m 50 m 50 deg]. Inputs stabilized. From top to bottom, the inputs value is τ_1, τ_2, τ_3. x axis is time for 350 seconds. y axis is inputs value.

percent overshoot for the surge, 0.01 percent overshoot for the sway, and 0 for the yaw. But the rise time is longer for MPC with integral action. The rise time for surge is 222.8 seconds, for sway is 235.5 seconds. However, the rise time for yaw is 814.4 seconds, and it takes long time to settle. This could be improved by updating the weights for MPC.

4.4 Waypoint tracking

Waypoint tracking is chosen for navigating the vessel, and this strategy is popular for autonomous vehicle, see [12] [8] and [13]. By using Voronoi diagram, we get a list of waypoints as shown in Fig. 3.3. A higher level control algorithm is used to check if the vessel is reached within the convex area, defined as follows:

$$\begin{aligned} &\sqrt{(x - x_{di})^2 + (y - y_{di})^2} < tol_{1i} \\ &abs(\psi - \psi_{di}) < tol_{2i} \end{aligned} \tag{4.3}$$

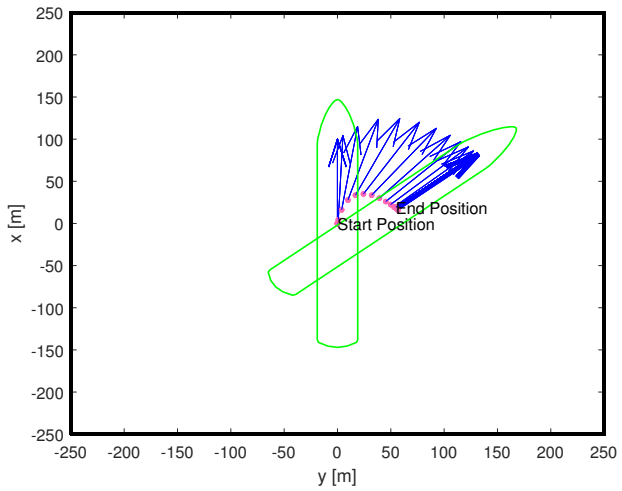

Figure 4.4: Trajectory for MPC without Integral Action with Reference [50 m 50 m 50 deg]. Trajectory is not planned well, given a curve towards to the desire position. It reaches a position close to the reference position but it does not reach the reference. Initial position is at [0 m 0 m 0 deg]. Green shape is an approximated ship plot. Red is the trajectory.

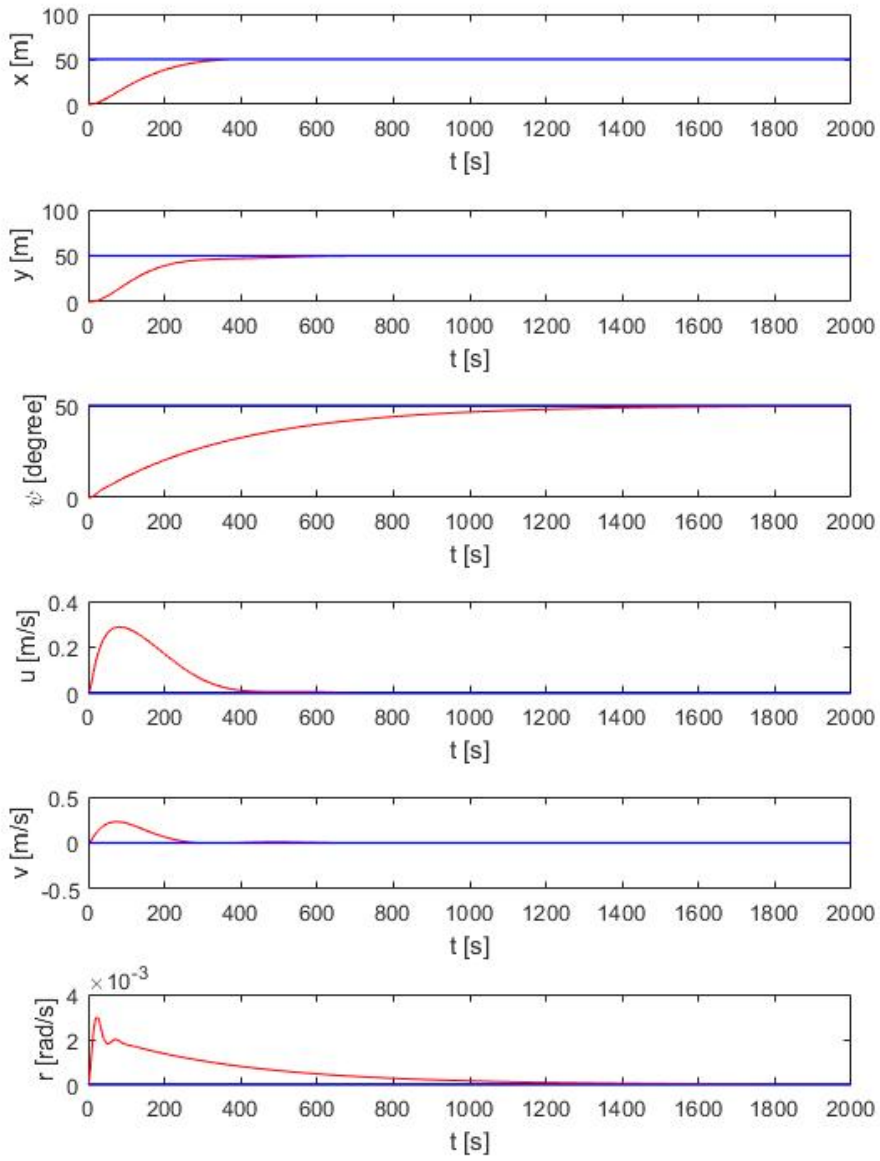

Figure 4.5: States for MPC with integral action. Reference is set to [50 m 50 m 50 deg], and system converges to the reference. Blue solid line is reference. Red solid line is trajectory.

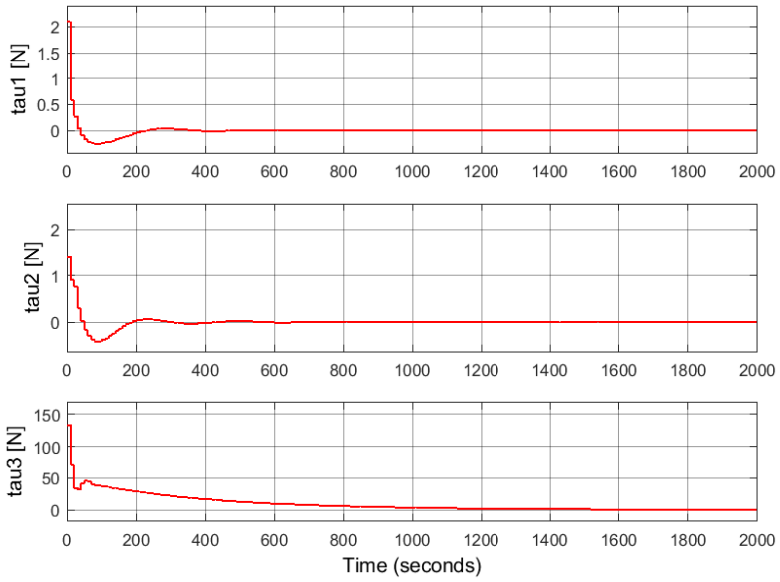

Figure 4.6: Inputs for MPC with integral action with Reference [50 m 50 m 50 deg]. Inputs stabilized. From top to bottom, the inputs value is τ_1, τ_2, τ_3. x axis is time for 2000 seconds. y axis is inputs value.

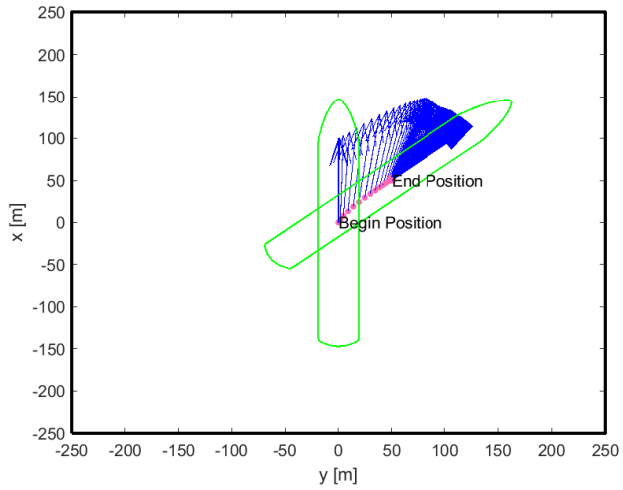

Figure 4.7: Ship trajectory for MPC with integral action. with Reference [50 m 50 m 50 deg]. Trajectory is planned well, given directly towards to the desired position. It reaches the reference position. Initial position is at [0 m 0 m 0 deg]. Green shape is an approximated ship plot. Red is the trajectory.

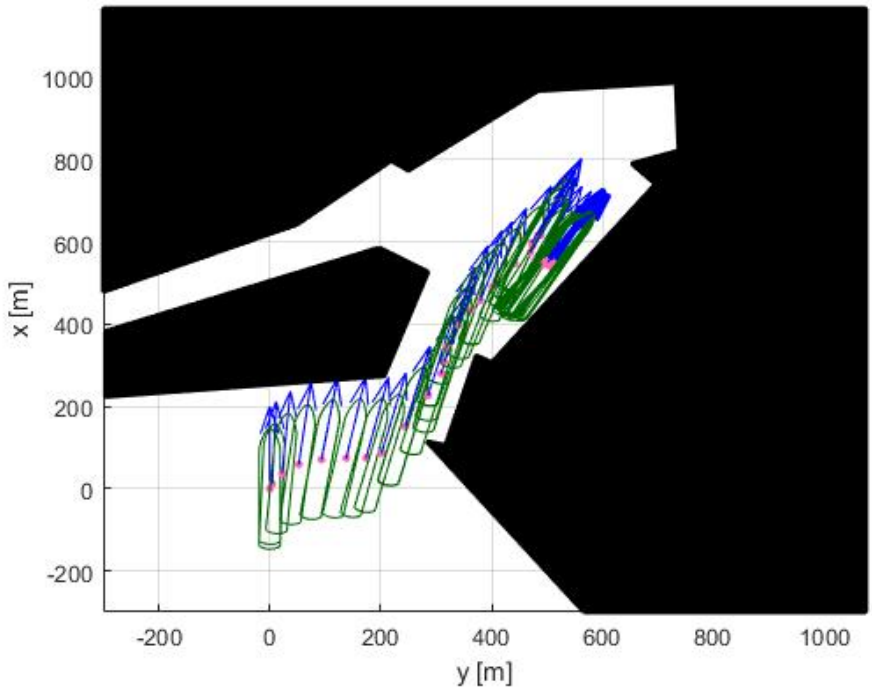

Figure 4.8: Performance for using MPC with integral action in Guidance Navigation and Control system and use waypoint tracking. Red is trajectory for the vessel when presenting the centre of gravity of the vessel. Green path is trajectory for the vessel when presenting the vessel with shape.

where x, y and ψ are surge sway and yaw in earth frame, x_d, y_d and ψ_d are desired surge, sway and yaw. The notation tol_{1i} and tol_{2i} are introduced for presenting the distance that is allowed for being away from the i desired position. Tolerance areas are chosen such that the size of allowed areas decreases when the vessel is closer to the desired position.

We use the waypoints from Fig. 3.3, and we get the performance from the GNC system in Fig. 4.8. Receding horizon, N, is chosen to be 20. Bounds for thruster forces, tau, is chosen to be $[\pm 4$ MN $\pm$ 1 MN $\pm$ 400 MNm$]$. Sample time is chosen to be 10 s. Weights for terminal set is mentioned at (4.2). Weight for the soft constraints, S, is set to be 10^{10}.

The GNC manage to bring the vessel to dock without having any side of vessel hit the bounds by using Voroni diagram and MPC with integral action. We can see five peaks in Fig. 4.9, due to switching to

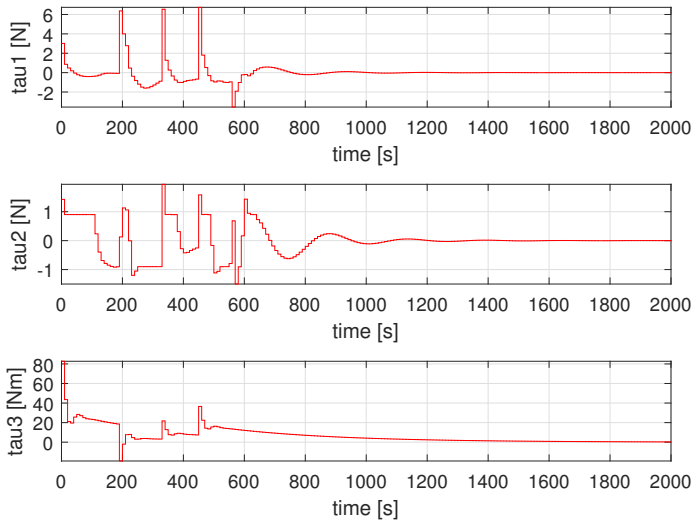

Figure 4.9: Inputs for using MPC with integral action in Guidance Navigation and Control system and use waypoint tracking. Inputs stabilized. From top to bottom, the inputs value is τ_1, τ_2, τ_3. x axis is time for 2000 seconds. y axis is inputs value.

another waypoint will require extra forces. Soft constraints minimize violations for states, but the limits of states are violated during the beginning when it starts to switched MPC controller. This is due to delay of sampling, which could be solved by saturation for thruster dynamics. In Fig. 4.10, surge, x, and sway, y, are stabilized before switching to another waypoint. However, the settling time for yaw, ψ, is long, although yaw reaches the reference value at the end of the simulation. The simulation shows also the vessel reached the final area set, see (4.3) around 900 seconds.

4.5 Disturbance Test

There will be disturbance such as wind and current, when a vessel is navigated to the port. We assume that the disturbance is unknown constant or slow varying measurement noise, and we would like to simulate such disturbance and analysis the performance of our system. Since the MPC with integral action can be used for dealing with unknown constant or slowly varying noise [7], we model such dis-

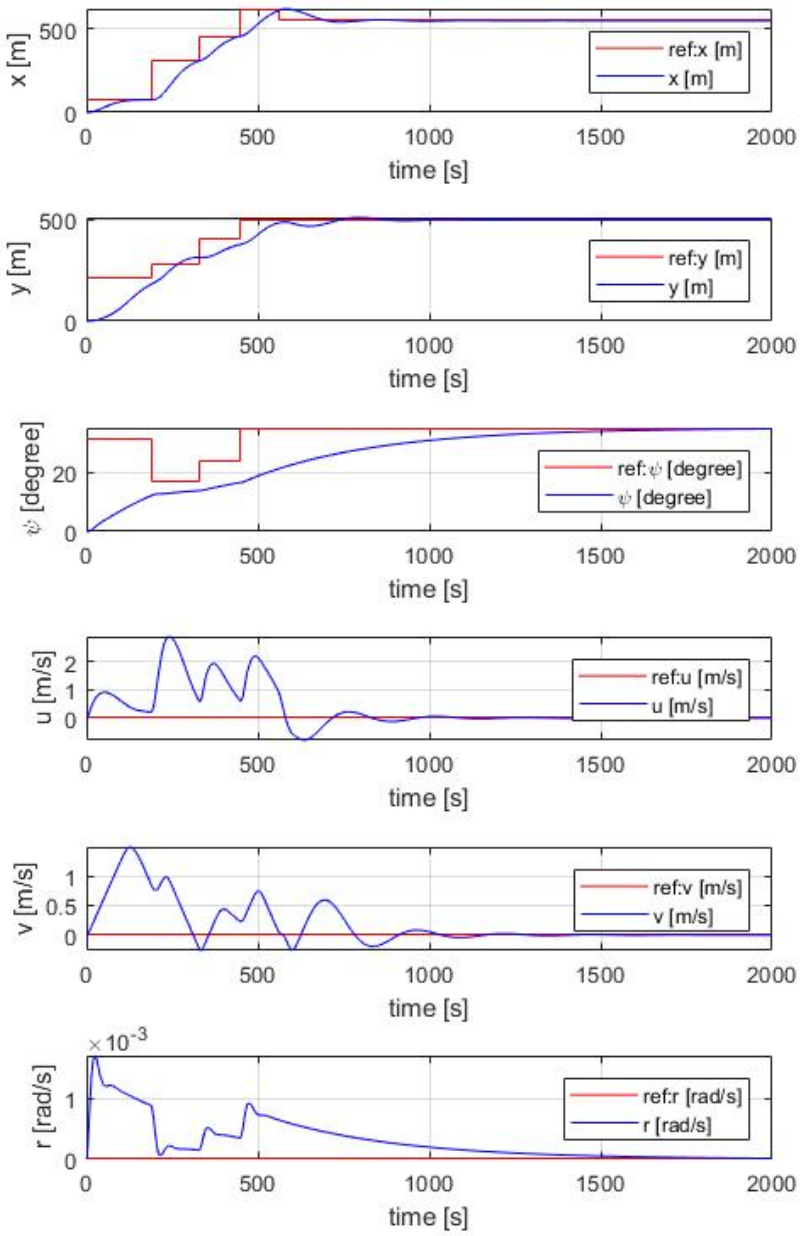

Figure 4.10: States for MPC without integral action. Red solid line is reference. Blue solid line is actual trajectory.

turbance as a constant bias vector as in (2.30). We are interested in studying how the size of bias vectors and desired position affects the performance, chosen preceding horizon and weight as same as previous studies.

By selecting different bias vector, we get the simulation results as shown in Fig. 4.11, Fig. 4.12, Fig. 4.13, Fig. 4.14. The results show that there is an upper bound of disturbance for having a performance such that the vessel can reach reference value, and this upper bound is $b = [10^6\ 10^5\ 10^5]$. When the disturbance is over the bound, inputs will start to change very large and the states cannot be stabilized towards to the reference value.

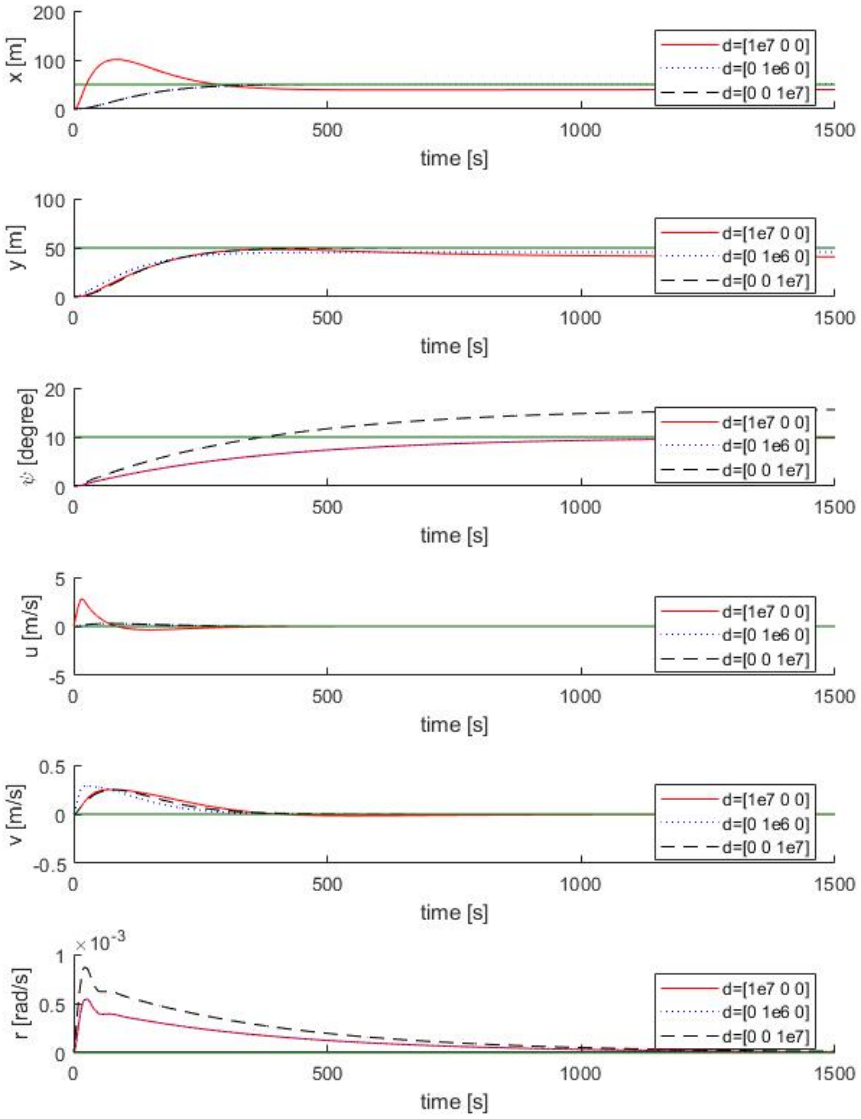

Figure 4.11: States for three disturbance test for (2.30) with $b = [10^7\ 0\ 0]$, $b = [0\ 10^6\ 0]$ and $b = [0\ 0\ 10^7]$ respectively. Green solid line is the reference value with $[50$ m 50 m 10 deg $]$.

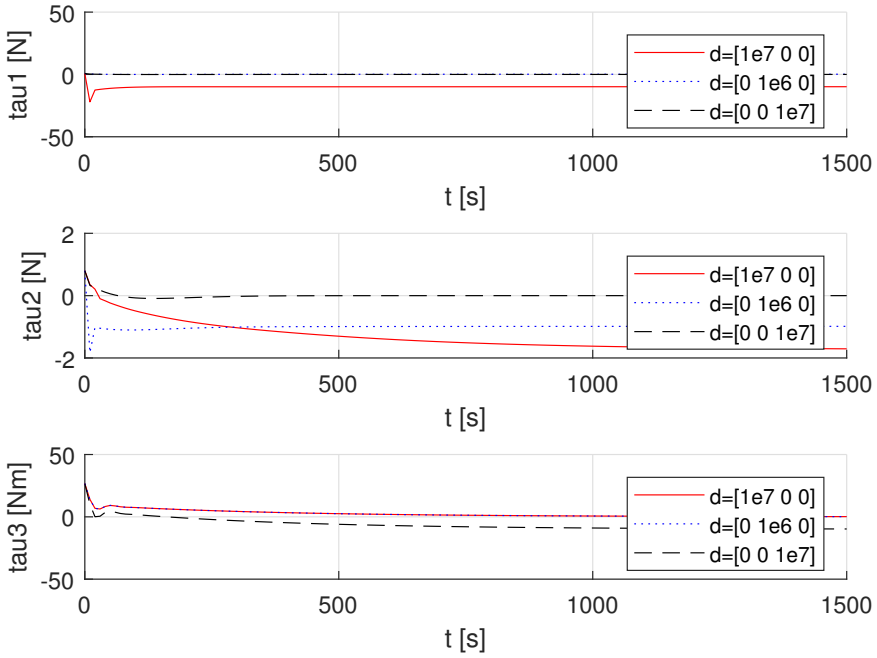

Figure 4.12: Inputs for three disturbance test for (2.30) with $b = [10^7\ 0\ 0]$, $b = [0\ 10^6\ 0]$ and $b = [0\ 0\ 10^7]$ respectively.

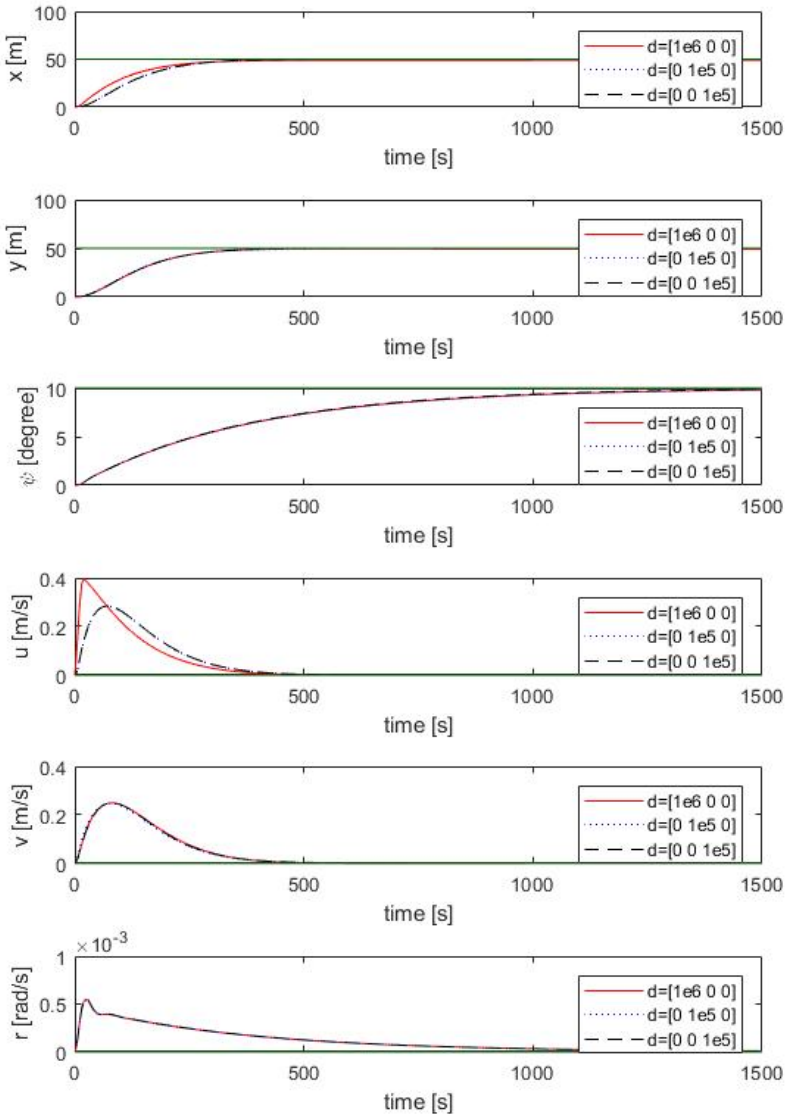

Figure 4.13: States for disturbance that is under the upper bound, and disturbance is applied to all 3 DOF with $b = [10^6\ 10^5\ 10^5]$. Green solid line is the reference value with $[50$ m 50 m 10 deg $]$.

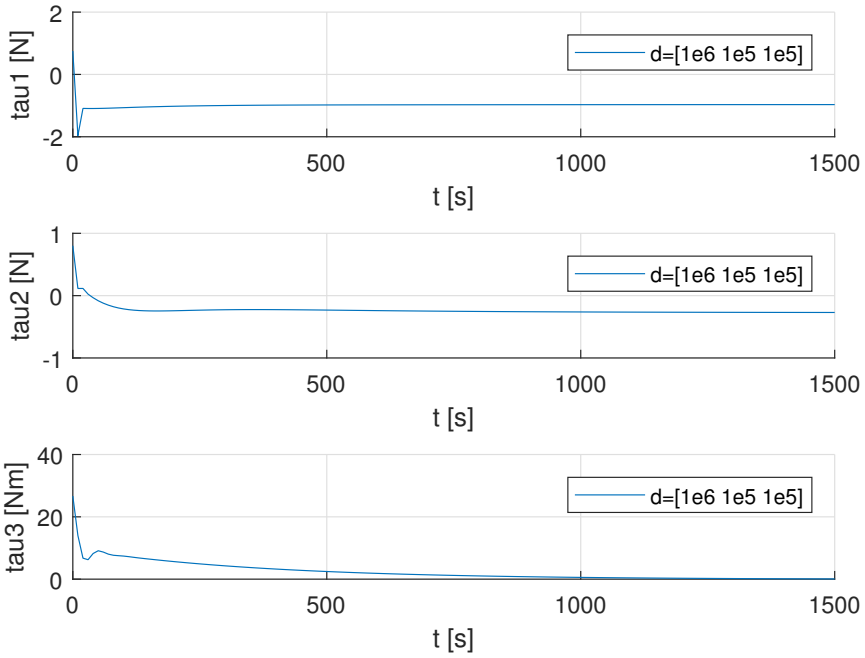

Figure 4.14: Input for disturbance that is under the upper bound, and disturbance is applied to all 3 DOF with $b = [10^6\ 10^5\ 10^5]$.

Chapter 5

Conclusion and Future Work

The main contribution of the thesis is that we implemented the GNC system for docking and navigating ships. Voronoi diagram is used for generating a waypoint list for waypoint tracking. MPC with integral action is applied to control the vessel for reducing model disturbance and constant disturbance from current and wind. The main focus of our thesis is on the controller for the navigation problem. We compared MPC with and without integral action, and shown that the static error can be minimized by using MPC with integral action. To test our methods, we performed the GNC system for South Harbour of Helsikin, and shown the vessel is navigated and docked at port. Moreover, we studied the effects of disturbance to keep the controller stabilized and suggested an upper limit for the disturbance.

Future work is discussed about the improvements of the controller, designs of path planning, and modelling of thruster dynamics. Although the guidance system is already able to generate waypoints, the MPC should be also improved with constraint for the environment for real-time object avoidance by including extra constraints. Future studies may also include studies on minimizing time for docking and minimizing fuel for docking due to the fact that the Voronoi diagram does not guarantee an optimal path. The future studies may include constraints for object avoidance for MPC instead of relying on the guidance system. One possible solution is to include different constraints for each phase as shown in Fig. 2.4. About the tolerance for switching MPC, if a larger tolerance for accessing a waypoint is chosen, a neighbour waypoint might be mistaken as a current waypoint instead. If a smaller tolerance is chosen, MPC will have a slower performance

because there is a higher requirement to be considered as reaching a waypoint. We have shown that the switching MPC is feasible with simulation, due to the fact that the time between each switch is so long in order to stabilize the motion. A possible study could be the recursive feasibility by studying the feasibility set over time. We used the thruster dynamics according to [10], and the future study could be applied to study the true thruster dynamics.

Bibliography

[1] National Aeronautics and Space Administration. "Apollo Operations Handbook Block II Spacecraft". In: *Apollo Flight Journal* 1 (2017).

[2] Randal W Beard et al. "Coordinated target assignment and intercept for unmanned air vehicles". In: *IEEE transactions on robotics and automation* 18.6 (2002), pp. 911–922.

[3] Stuart Bennett. "Nicholas Minorsky and the automatic steering of ships". In: *IEEE Control Systems Magazine* 4.4 (1984), pp. 10–15.

[4] Priyadarshi Bhattacharya and Marina L Gavrilova. "Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path". In: *IEEE Robotics & Automation Magazine* 15.2 (2008).

[5] Alexandre Borowczyk et al. "Autonomous landing of a multirotor micro air vehicle on a high velocity ground vehicle". In: *arXiv preprint arXiv:1611.07329* (2016).

[6] S Campbell, Wasif Naeem, and George W Irwin. "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres". In: *Annual Reviews in Control* 36.2 (2012), pp. 267–283.

[7] David Di Ruscio. "Model predictive control with integral action: A simple MPC algorithm". In: (2013).

[8] KD Do and J Pan. "Global waypoint tracking control of underactuated ships under relaxed assumptions". In: *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*. Vol. 2. IEEE. 2003, pp. 1244–1249.

[9] Daniel C Fernández and Geoffrey A Hollinger. "Model predictive control for underwater robots in ocean waves". In: *IEEE Robotics and Automation Letters* 2.1 (2017), pp. 88–95.

[10] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[11] Thor I Fossen and Jann Peter Strand. "Nonlinear passive weather optimal positioning control (WOPC) system for ships and rigs: experimental results". In: *Automatica* 37.5 (2001), pp. 701–715.

[12] Jenhwa Guo. "A waypoint-tracking controller for a biomimetic autonomous underwater vehicle". In: *Ocean engineering* 33.17-18 (2006), pp. 2369–2380.

[13] Dong-in Han et al. "Development of Unmanned Aerial Vehicle (UAV) system with waypoint tracking and vision-based reconnaissance". In: *International Journal of Control, Automation and Systems* 8.5 (2010), pp. 1091–1099.

[14] Rui Huang, Yong Liu, and J Jim Zhu. "Guidance, Navigation, and Control System Design for Tripropeller Vertical-Take-Off-and-Landing Unmanned Air Vehicle". In: *Journal of aircraft* 46.6 (2009), pp. 1837–1856.

[15] AT Ismail, Alaa Sheta, and Mohammed Al-Weshah. "A mobile robot path planning using genetic algorithm in static environment". In: *Journal of Computer Science* 4.4 (2008), pp. 341–344.

[16] Mayne David Q. James B. Rawlings. *Model predictive control: theory and design*. Nob Hill Pub, Llc, 2009.

[17] E Jelavic, J Gonzales, and F Borrelli. "Autonomous Drift Parking using a Switched Control Strategy with Onboard Sensors". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 3714–3719.

[18] Mikael Johansson. *Notes on receding horizon and model predictive control*. Oct. 2017.

[19] Marius Kloetzer, Cristian Mahulea, and Ramon Gonzalez. "Optimizing cell decomposition path planning for mobile robots using different metrics". In: *System Theory, Control and Computing (ICSTCC), 2015 19th International Conference on*. IEEE. 2015, pp. 565–570.

[20] Jacoby Larson, Michael Bruch, and John Ebken. "Autonomous navigation and obstacle avoidance for unmanned surface vehicles". In: *Unmanned Systems Technology VIII*. Vol. 6230. International Society for Optics and Photonics. 2006, p. 623007.

[21] J. Löfberg. "YALMIP : A Toolbox for Modeling and Optimization in MATLAB". In: *In Proceedings of the CACSD Conference*. Taipei, Taiwan, 2004.

[22] Jan Maciejowski. *Predictive control: with constraints*. Pearson Education, 2002.

[23] David Q Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[24] Nicolas Minorsky. "Directional stability of automatically steered bodies". In: *Journal of ASNE* 42.2 (1922), pp. 280–309.

[25] Signe Moe et al. "Path following of underactuated marine surface vessels in the presence of unknown ocean currents". In: *American Control Conference (ACC), 2014*. IEEE. 2014, pp. 3856–3861.

[26] So-Ryeok Oh and Jing Sun. "Path following of underactuated marine surface vessels using line-of-sight based model predictive control". In: *Ocean Engineering* 37.2-3 (2010), pp. 289–295.

[27] Gabriele Pannocchia and James B Rawlings. "Disturbance models for offset-free model-predictive control". In: *AIChE journal* 49.2 (2003), pp. 426–437.

[28] Jeffrey S Riedel and Anthony J Healey. "Shallow water station keeping of AUVs using multi-sensor fusion for wave disturbance prediction and compensation". In: *OCEANS'98 Conference Proceedings*. Vol. 2. IEEE. 1998, pp. 1064–1068.

[29] Miguel San Martin et al. "In-flight experience of the mars science laboratory guidance, navigation, and control system for entry, descent, and landing". In: *CEAS Space Journal* 7.2 (2015), pp. 119–142.

[30] Pierre OM Scokaert and James B Rawlings. "Feasibility issues in linear model predictive control". In: *AIChE Journal* 45.8 (1999), pp. 1649–1659.

[31] Roger Skjetne, Øyvind Smogeli, and Thor I. Fossen. "Modeling, identification, and adaptive maneuvering of CyberShip II: A complete design with experiments". In: *IFAC Proceedings Volumes* 37.10 (2004). IFAC Conference on Computer Applications in Marine Systems - CAMS 2004, Ancona, Italy, 7-9 July 2004, pp. 203–208. ISSN: 1474-6670. DOI: https://doi.org/10.

1016/S1474-6670(17)31732-9. URL: http://www.sciencedirect.com/science/article/pii/S1474667017317329.

[32] Roger Skjetne, Øyvind Smogeli, and Thor I Fossen. "Modeling, identification, and adaptive maneuvering of Cybership II: A complete design with experiments". In: *IFAC Proceedings Volumes* 37.10 (2004), pp. 203–208.

[33] Bradley A Steinfeldt et al. "Guidance, navigation, and control system performance trades for Mars pinpoint landing". In: *Journal of Spacecraft and Rockets* 47.1 (2010), pp. 188–198.

[34] Tsutomu Tashiro. "Vehicle steering control with MPC for target trajectory tracking of autonomous reverse parking". In: *Control Applications (CCA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 247–251.

[35] *The Port of Helsinki takes the top spot among European passenger ports*. Available at http://www.portofhelsinki.fi (2018/05/12).

[36] Avishai Weiss et al. "Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies". In: *IEEE Transactions on Control Systems Technology* 23.4 (2015), pp. 1638–1647.

[37] Melanie N Zeilinger, Manfred Morari, and Colin N Jones. "Soft constrained model predictive control with robust stability guarantees". In: *IEEE Transactions on Automatic Control* 59.5 (2014), pp. 1190–1202.

Appendix A

Tuning weights

We studied on different weights for terminal set, Q_f, and on different weights for soft constraints, S, with Fig. A.1 and Fig. A.2. We set:

$$Q_{f1} = diag([1\ 1\ 10^{10}\ 10^8\ 10^8\ 10^8\ 10^8\ 10^8\ 10^8])$$
$$Q_{f2} = diag([1\ 1\ 10^{10}\ 0\ 0\ 0\ 10^8\ 10^8\ 10^8])$$

The results show that the vessel reached each waypoint ealier with Q_{f2}. A MPC controller without soft constraints is marked with S_1. A MPC controller with soft constraints with $S = 10^{10}$ is marked with S_2. The results show that the violations of inputs are minimized with higher weights of soft constraints.

We studied on different convex area for checking if the vessel reaches each waypoint, see (4.3). Given five waypoints from the voronoi diagram in Figure. 3.3, we set tolerance for $tol_{11}tol_{12}...tol_{15}tol_{21}tol_{22}...tol_{25}$ from (4.3) as follows:

$$cv_1 = \begin{bmatrix}35 & 30 & 25 & 20 & 25 & 25 & 25 & 25\end{bmatrix}'$$
$$cv_2 = \begin{bmatrix}30 & 25 & 20 & 15 & 25 & 25 & 25 & 25\end{bmatrix}$$
$$cv_3 = \begin{bmatrix}35 & 30 & 25 & 20 & 15 & 15 & 15 & 15\end{bmatrix}$$

States and Inputs for these three convex area settings are shown in Fig. A.3 and Fig. A.4 respectively. The simulated vessel trajectory is shown in Fig. A.5, Fig. A.6 and Fig. A.7 by choosing settings from (A).

Results above show that the setting for tolerance cv_1 is fastest, see Fig. A.3 and Fig. A.4, however the vessel body is closer to the bound according to Fig. A.5, Fig. A.6 and Fig. A.7. Therefore, there is a trade-off between tolerance and safety of navigation.

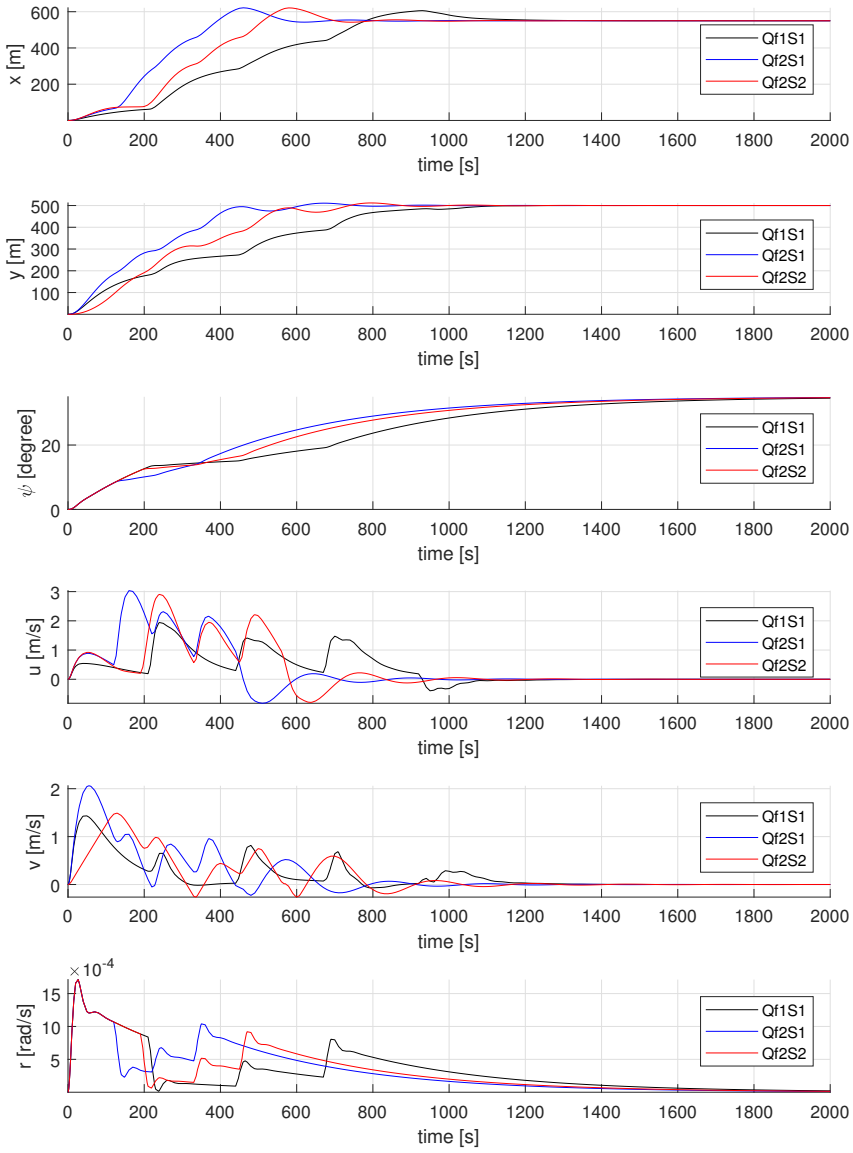

Figure A.1: States for choosing Q_{f1}, Q_{f2}, S_1 and S_2.

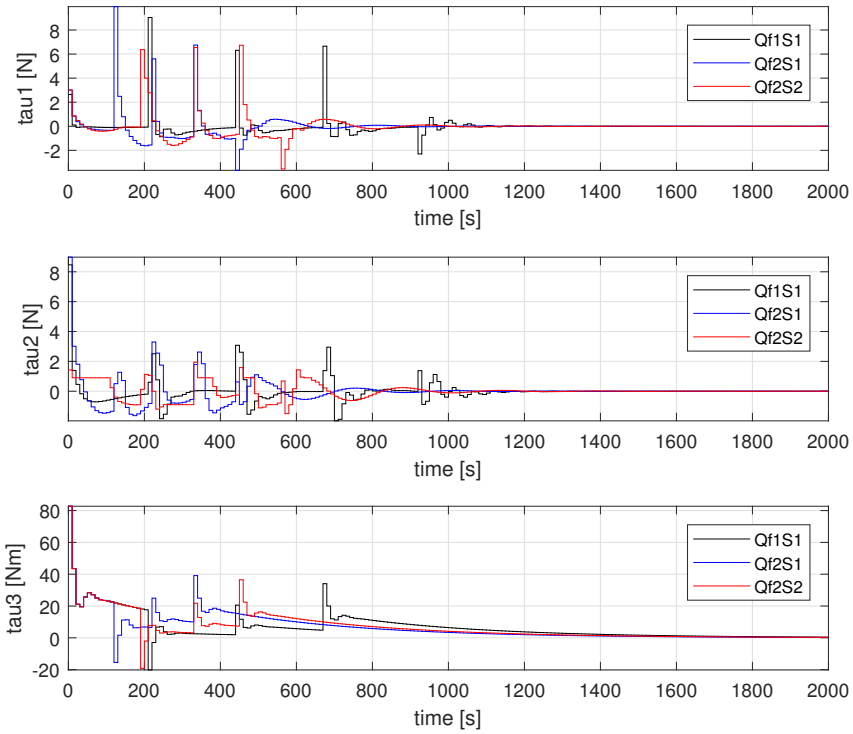

Figure A.2: Inputs for choosing Q_{f1}, Q_{f2}, S_1 and S_2.

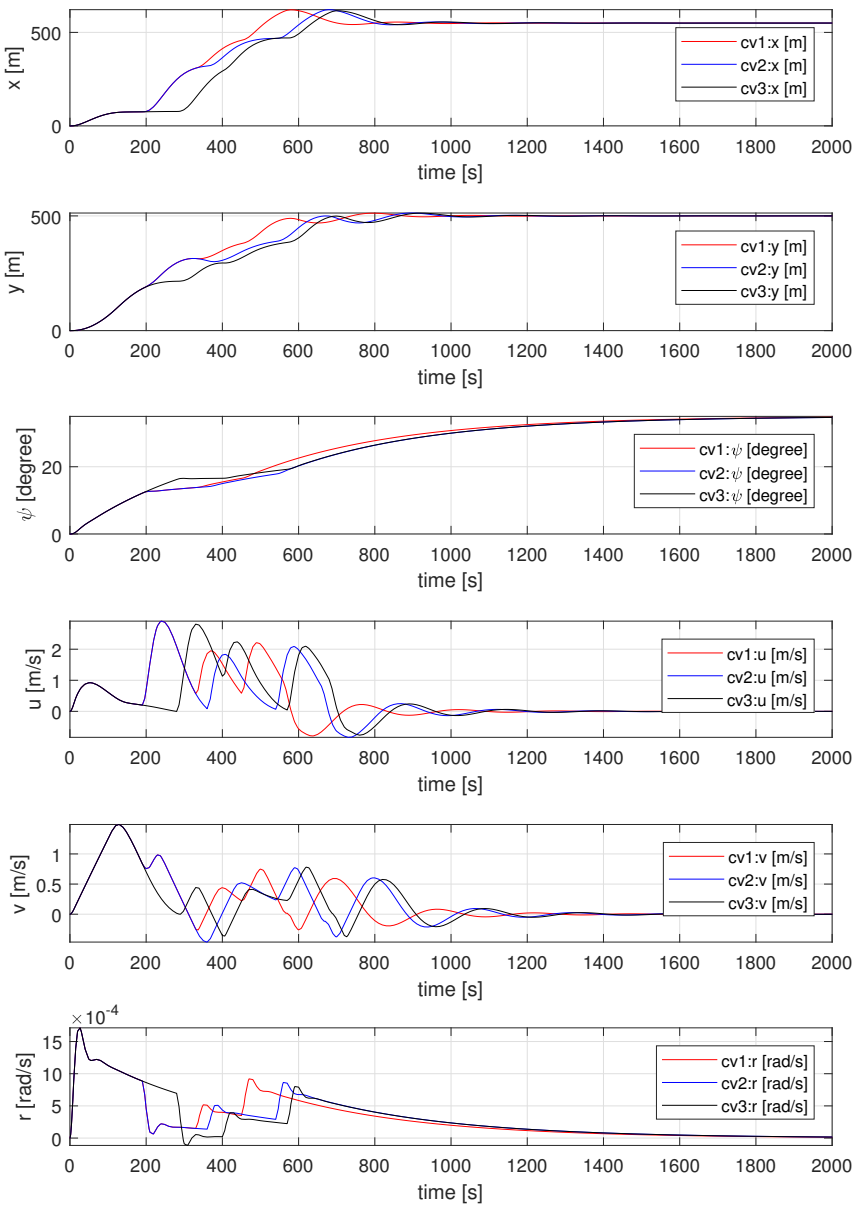

Figure A.3: States for choosing cv_1, cv_2 and cv_3.

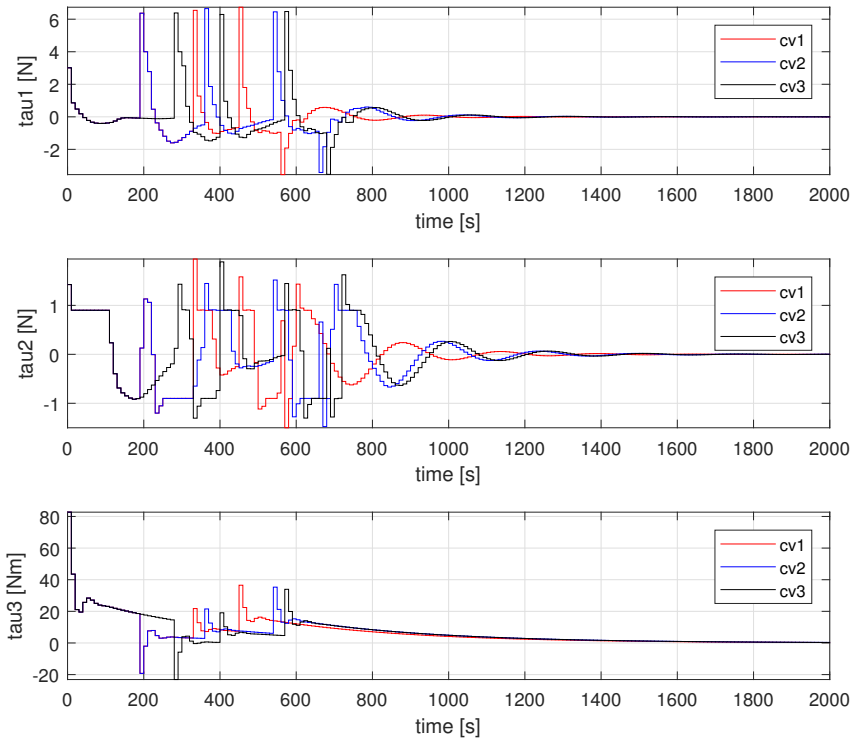

Figure A.4: Inputs for choosing cv_1, cv_2 and cv_3.

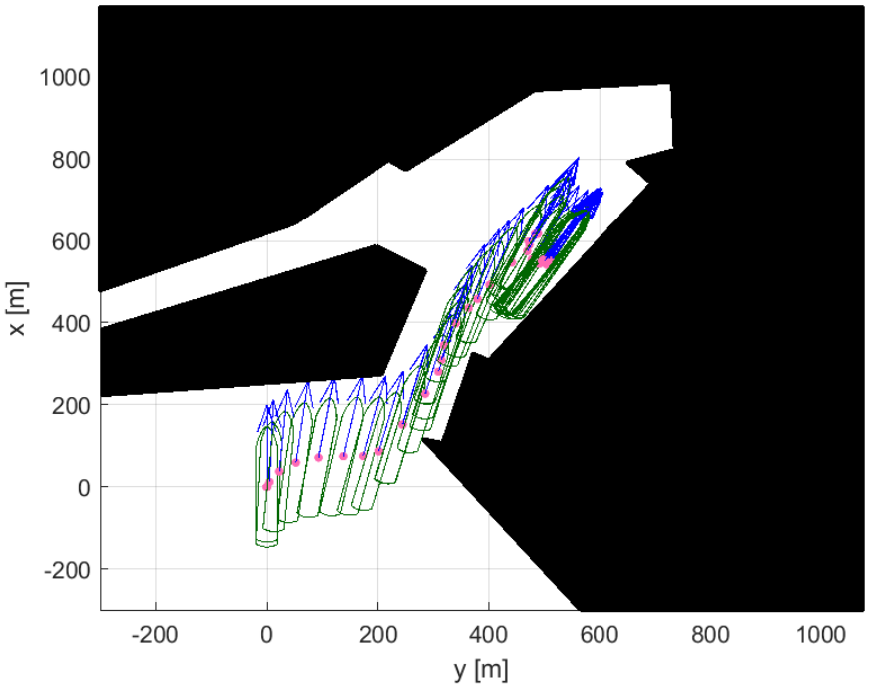

Figure A.5: Vessel simulated path for choosing cv_1.

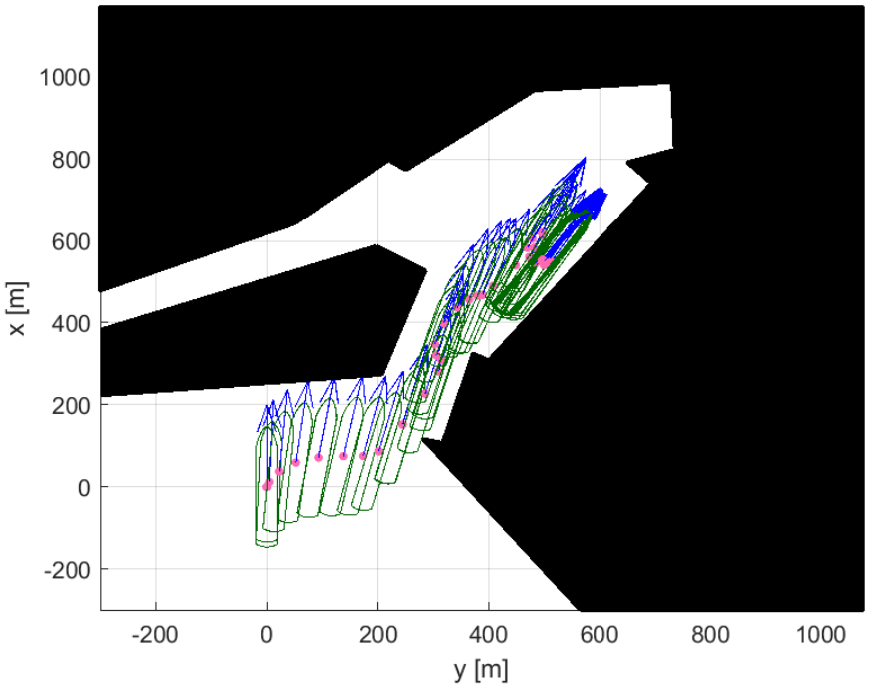

Figure A.6: Vessel simulated path for choosing cv_2.

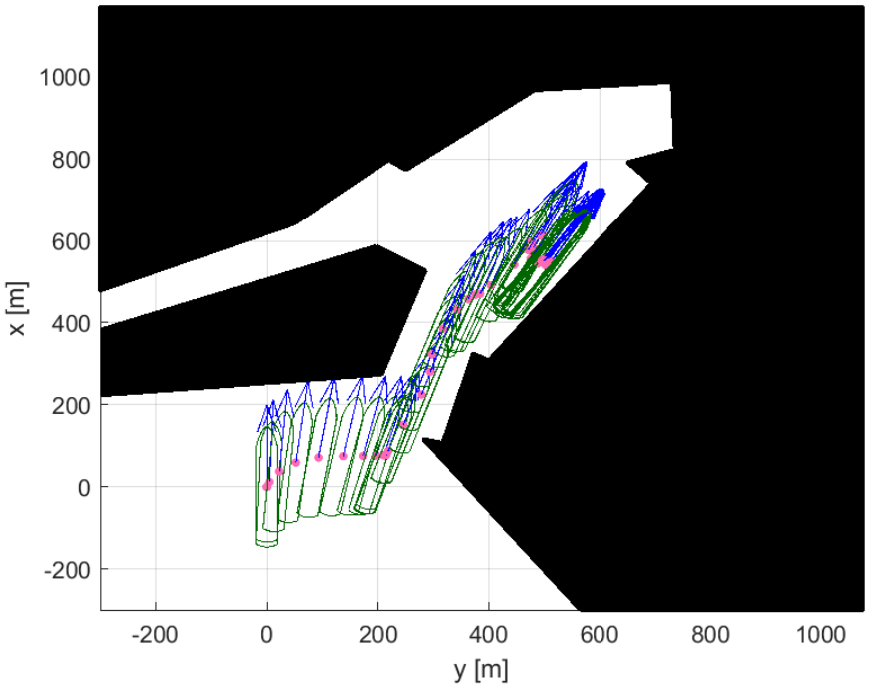

Figure A.7: Vessel simulated path for choosing cv_3.

TRITA TRITA-EECS-EX-2018:401
ISSN 1653-5146