ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DYNAMIC MODEL-BASED PATH PLANNING OPTIMIZATION AND CONTROL FOR USV IN INLAND WATERWAYS

M.Sc. THESIS

Ferhan BÜYÜKÇOLAK

Department of Mechatronics Engineering

Mechatronics Engineering Programme

JULY 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

DYNAMIC MODEL-BASED PATH PLANNING OPTIMIZATION AND CONTROL FOR USV IN INLAND WATERWAYS

M.Sc. THESIS

Ferhan BÜYÜKÇOLAK
(518201013)

Department of Mechatronics Engineering

Mechatronics Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Gökhan Tansel TAYYAR

JULY 2023

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

İÇ SU YOLLARINDA KULLANILAN İDA’LAR İÇİN DİNAMİK MODEL TABANLI YOL PLANLAMA OPTİMİZASYONU VE KONTROLÜ

YÜKSEK LİSANS TEZİ

Ferhan BÜYÜKÇOLAK
(518201013)

Mekatronik Mühendisliği Anabilim Dalı

Mekatronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Gökhan Tansel TAYYAR

TEMMUZ 2023

Ferhan BÜYÜKÇOLAK, a M.Sc. student of ITU Graduate School student ID 518201013 successfully defended the thesis entitled "DYNAMIC MODEL-BASED PATH PLANNING OPTIMIZATION AND CONTROL FOR USV IN INLAND WATERWAYS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Assoc. Prof. Dr. Gökhan Tansel TAYYAR** ....
Istanbul Technical University

Jury Members : **Asst. Prof. Dr. Melek ERTOGAN** ....
Istanbul Technical University

Prof. Dr. Şeniz ERTUĞRUL ....
Izmir University of Economics

....

Date of Submission : **26 May 2023**
Date of Defense : **10 July 2023**

FOREWORD

I express my gratitude to my beloved parents for their unwavering support and encouragement throughout my academic journey. I would also like to extend my appreciation to my thesis supervisor, Assoc. Prof. Dr. Gökhan Tansel TAYYAR and Asst. Prof. Dr. Melek ERTOGAN, whose guidance and instruction were instrumental in the completion of my thesis.

July 2023 Ferhan BÜYÜKÇOLAK

TABLE OF CONTENTS

Page

ABBREVIATIONS

BVP	: Boundary Value Problem
COG	: Center of Gravity
CFD	: Computational Fluid Dynamics
DOF	: Degree of Freedom
ECEF	: Earth Center Earth Fix
EKF	: Extended Kalman Filter
GPS	: Global Positining System
GNSS	: Global Navigation Satellite Systems
GNC	: Guidance Navigation Control
IMU	: Inertial Measurement System
LS	: Least Square
LOS	: Line of Sight
LQR	: Linear Quadratic Regulator
MPC	: Model Predictive Control
NED	: North East Down
NLP	: Nonlinear Programming
NMPC	: Nonlinear Model Predictive Control
RPM	: Revolution Per Second
RRT	: Rapidly-exploring Random Trees
SI	: System Identification
OCP	: Optimal Control Problem
ODE	: Ordinary Differential Equations
USV	: Unmanned Surface Vehicle
QP	: Quadratic Programming

SYMBOLS

x_{los}	: X location of los point
y_{los}	: Y location of los point
x_g	: Location of center of gravity on x axis
y_g	: Location of center of gravity on y axis
z_g	: Location of center of gravity on z axis
x_b	: Location of body frame origin point on x axis
y_b	: Location of body frame origin point on y axis
z_b	: Location of body frame origin point on z axis
O_n	: Location of NED frame origin point on x axis
O_b	: Location of NED frame origin point on y axis
u	: Surge speed
v	: Sway speed
w	: Heave speed
p	: Roll angular rate
q	: Pitch angular rate
r	: Yaw angular rate
ψ	: Yaw angle
θ	: Pitch angle
ϕ	: Roll angle
X	: Acting force on X axis
Y	: Acting force on Y axis
N	: Acting Moment on N rotation
τ_{wind}	: Wind force
τ_{wave}	: Wave force
τ_u	: Thruster force
M_A	: Added mass matrix
M_{RB}	: Rigid mass matrix
C_A	: Coriolis added matrix
C_{RB}	: Coriolis rigid matrix
D_{NL}	: Nonlinear damping matrix
R	: Rotation matrix
F_{port}	: Port side propeller thrust
F_{std}	: Starboard side propeller thrust
m	: vehicle mass
I_g	: Moment of inertia
q_{init}	: start point
q_{near}	: The nearest neigbouring vertex to q_rand
q_{rand}	: A randomly sampled state
q_{free}	: Obstacle free space
Δh	: Sampling time
y_p	: Distance between two thruster

LIST OF TABLES

LIST OF FIGURES

DYNAMIC MODEL-BASED PATH PLANNING OPTIMIZATION AND CONTROL FOR USV IN INLAND WATERWAYS

SUMMARY

The aim of this thesis study is to apply the Guidance, Navigation, and Control (GNC) system for autonomous marine technology to a marine vehicle in the form of a catamaran with a differential drive system. The Model Predictive Controller (MPC), which is a model-based control approach, is used for controlling the system. Within the scope of the study, a simulator including the dynamic model of Otter USV, a vehicle owned by Maritime Robotic AS company, has been utilized.

Firstly, within the scope of the study, the system model of the used vehicle has been defined, and the components of the vehicle's mathematical model have been described. In order to use it in the model-based controller, a system identification approach has been employed to determine the coefficients of the vehicle's mathematical model. For system identification, various maneuvers were performed on the vehicle's simulator model, and navigation data was collected. Using this collected navigation data, the values of the parameters of the vehicle's dynamic model were determined through the non-linear least square method.

Secondly, the development of the guidance system for the vehicle has been carried out. A system has been developed that can reach the target point while avoiding collisions in an environment with static obstacles, taking into account all input and differential constraints of the vessel model. For global path planning, a method called Kinodynamic RRT has been developed, which plans the path by considering the dynamics of the vehicle. In addition, utilizing the optimal control problem approach, an optimization-baed path planning has been performed. To enable the vehicle to follow the generated overall path, the implementation of a path following algorithm called Line of Sight (LOS) has been applied to this system.

In this study, since the focus was on the realization of vehicle trajectory tracking in narrow waterways, different approaches were taken in the calculation of the lookahead distance in this method to improve the performance of the classical LOS method.

The following of the generated path by the guidance system of the vehicle and the following of the reference state values generated for path following have been performed using a nonlinear model predictive control (NMPC) system. To find the optimum values of the parameters that affect the performance of the controller, the performance of the controller has been tested in different scenarios, and the most suitable values have been determined. In order to create this model-based controller and to have a suitable software architecture for real-time requirements, a fast solution method was needed, so the algorithm was developed in the CasADi optimization framework.

To observe the performance difference between the developed controller and conventional control methods, a suitable PID controller has been developed for the Otter USV. The performance of these two controllers has been compared in scenarios with and without disturbances.

İÇ SU YOLLARINDA KULLANILAN İDA’LAR İÇİN DİNAMİK MODEL TABANLI YOL PLANLAMA OPTİMİZASYONU VE KONTROLÜ

ÖZET

Otonom gemiler, denizcilik sektöründe önemli bir teknolojik ilerlemedir. İnsan müdahalesine gerek duymadan otomatik olarak faaliyetlerini gerçekleştirebilen bu gemiler, sensörler, yapay zeka ve otomasyon teknolojileriyle donatılmıştır. Enerji verimliliği, iş güvenliği ve çevresel sürdürülebilirlik gibi avantajlarıyla otonom gemiler, denizcilik endüstrisinde büyük potansiyele sahiptir. Otonom gemiler, verimliliği artırırken aynı zamanda insan hatalarını azaltarak maliyetleri düşürebilir. Gemi otonomisi, iç su yollarında önemli bir rol oynamaktadır. İç su yolları genellikle yoğun trafik ve dar manevra alanları içerir, bu nedenle otonom gemiler bu alanlarda daha etkili ve güvenli bir şekilde hareket edebilir. Otonom gemiler, insan hatası riskini azaltarak iç su yollarında çarpışma ve kaza olasılığını önemli ölçüde azaltabilir ve bu da daha verimli ve güvenli bir denizcilik trafiği sağlar. Ayrıca, otonom gemiler iç su yollarında daha düşük enerji tüketimi ve çevresel etki ile daha sürdürülebilir bir taşımacılık sağlayarak çevresel açıdan da önemli avantajlar sunar. Bu yüzden dar iç su yollarına uygun rota oluşturulması ve be bu rotanın takip edilmesi üzerine yapılacak çalışmalar önem taşımaktadır. BU tezde de bu alanlarda çalışmalar gerçekleştirilmiştir.

Bu tez çalışmasının amacı deniz otonom teknolojisi için güdüm, navigasyon, ve kontrol (GNK) sisteminin, bir katamaran formunda difransiyel itki sistemine sahip bir deniz aracına iç su yollarında engellerden kaçınarak, araç dinamiğine uygun rota oluşturulması ve bu rotanın takip edilemesini sağlayan denetleyicilerin geliştirilmesinin yapıldığı çalışmaları içermektedir. Sisteminin kontrol edilmesinde ise model tabanlı bir yaklaşım olan model öngörülü denetleyici (MPC) ve PID denetleyicisi kullanılarak istenilen rotayı takip etme performansları bozucuların olduğu veya olmadığı ortamlarda karşılaştırması yapılmıştır. Bu çalışma kapsamında oluşturulan algoritmaların başarımlarını gözlemlemek için Maritime Robotic AS şirketinin aracı olan Otter USV’nin dinamik modelini içeren simulator kullanılmıştır.

İlk olarak, aracın dinamik ve kinematik modelinin yapısı anlatılmıştır. Oluşturulan sistemin dinamik denkleminden yararlanılarak aracın sistem modelinin çıkarılması yapılmıştır ve aracın matematiksel modelinin farklı serbestlikteki hidrodinamik denklemleri gösterilmiş ve hangi seyir testleriyle, hangi denklemlerin parametrelerinin kestirilebileceği anlatılmıştır. Sistem modelinin belirlenebilmesi için aracın benzetim modelinde, çeşitli manevralar yaptırılarak, seyir verileri toplanmıştır. Elde edilen bu seyir verileri kullanılarak lineer olmayan en küçük kareler yöntemi aracılığıyla, aracın dinamik modelinin parametrelerinin değerleri tesbit edilmiştir.

İkinci olarak, aracın güdüm sisteminin geliştirilmesi yapılmıştır. Bir başlangıç durumunda, gemi modelinin tüm giriş ve sistem kısıtlamalarını dikkate alarak, statik

engellerin olduğu ortamda çarpışmadan kaçınarak hedef noktaya ulaşan genel rota planlama algoritmaları geliştirilmiştir. Bu çalışma kapsamında iki farklı rota planlama algoritması geliştirilmiştir. İlk Gelenesel yöntemlerden biri olan Kinematik RRT olarak adlandırılan, aracın dinamik modelini dikkate alarak genel rotayı planlayan algoritmadır. Bu yöntem çevreye rastgele noktalarak atayarak ve bu noktalar arasında aracın takip edebileceği bağlantı yolları oluştururak hedeflenen noktaya kadar genel bir rota oluşturmaktadır. Bu yaklaşım çalışma prensibinde rastgelilikten kaynaklı belirsizlikler içerdiğinden, günüzmüzde kullanılmaya başlanan optimizasyon tabanlı bir genel rota planlayıcı yapısı kurulmuştur. Optimizasyon tabanlı rota planlama yöntemi, esasında bir optimizasyon problemi olduğundan dolayı algoritması içersinde bir maliyet fonksiyonu barındırmaktadır. Bu maliyet fonksiyonunun yapısı içersine optimize edilemesi istenen, minimum zaman, mininmum yakıt tüketimi, minimum yol alma gibi kısıtlar eklenerek, maliyet fonksiyonun çözümü gerçekleştirilir. Böylece istenilen kısıtlara göre veya optimize edilmesi istenilen durumlara göre genel rota planlaması yapılmaktadır. Bu tez çalışmasında da bu yapı oluşturularak, belirlenenen hız kısıtlarına göre genel rota planlaması yapılmıştır. Böylece içersinde bir çok engelin olduğu iç su yollarında, kısıtlı bir alanma fazla manevra gerektiren rotalar planlanmak istedentiğinde belirtilen kısıtlar veya optimize edilmesi istenelin durumlar belirtilerek, dar alanlarada hareket edebilmeye uygun rotalar oluşturulabilmekter. Optimizasyon tabanlı rota oluşturulmasında, çözüm aracı olarak casadi optimazyon bilgisayar programı kullanılmıştır. Programın açık kaynak kodlu olması, benzerli ile karşılaştırıldığında, çok düşük hesaplama zamanına sahip olmasından dolayı bu casadi optimizasyon programı olarak seçilmiştir. Sonrasında, bu programla genel rota planlayıcının yapısı anlatılmış ve oluşturulan rotalara örnekler verilmiştir.

Navigasyon kısımında, Aracın belirlenen bir rotayı takip edebilmesi için denizcilik sektöründe genel olarak kullanılan görüş-hattı algoritmasının matematiksel yapısı nasıl oluşturulduğu anlatımıştır ve rota takibinde performansı etkileyen parametreler anlatılmıştır. Kullanılan bu rota takip algoritması daha çok açık denizlerde, manevra isterinin fazla olmadığı durumlarda başarılı sonuçlar vermektedir. Bu çalışma kapsamında dar su yollarına uygun olarak bu algoritmada değişiklikler yaparak, bu rota takip yönteminin dar su yollarına uygun hale getirilmiştir. Bu iki algoritmanın performanslarının karşılaştırılmasıda çeşitli seneryolar üzerinden test edilmiştir. Buna ek olarak, görüş-hattı algoritmasının avantajlarından ve dezavantajlarına da değinilmiştir.

Aracın güdüm sistemi tarafından oluşturulan rota ve rotanın takibi için üretilen referans durum değerlerinin takibinin yapılması linear olmayan model öngörü kontrol sistemi ile yapılmıştır. Model tabanlı bu kontrolcü yönteminin oluşturulması ve gerçek zamanlı sistemler için de uygun yazılım mimarisinde olması için, hızlı bir çözüm yöntemi gerektiğinden, algoritmanın oluşturulması tıpkı rota planamada olduğu gibi Casadi programıyla yapılmıştır. Casadi optimizasyon aracında bu model öngörülü denetleyici yapının nasıl oluşturulacağı bu kısımda anlatılmıştır. MPC ile pid algoritmasının rota takip performansları test edileceğinden, bu iki kontrolcünün tasarım parametreleri optimize edilmye çalışılmıştır. İlk olarak MPC'nin tasarımı gerçekleştirilmiştir. MPC içerinde üç adet tasarım parametresi içermektedir ve bunlar denetleyici ufku değeri, sistemin durumlarının ağırlık değerleri ve sistemin giriş değerlerinin ağırlık değerleridir. Bu parametrelerin uygun değerlenin bulunabilmesi için farklı seneryolarda MPC belirtilen tasarım parametreleri değiştirilerek, rota takip edebilme performansına bakılarak, uygun parametreler belirlenmiştir. Bu yaklaşıma benzer olarak hız ve açısal konum PID denetleyicinin katsayı değerleride belirlenmiştir.

Otonom sistemleri için olmazsa olmaz yapı olan güdüm,navigasyon ve denetleyici yapısı ilgili böl-mlerde bu çalışmda anlatıldıktan sonra oluşturulan bu sistemin performansı benzetim kıstımında test edilmiştir. Aracın güdüm kısmında geliştirilen rota planlama algoritmaları, navigasyon kısmında geliştirilen rota takip algoritmalarıyla, denetleyici tasarımı kısmında geliştirilen MPC ve PID denetleyicilerinin oluşturulmasıyla yapısı tamamlanan güdüm-navigasyon-denetleyici sisteminin performansları farklı seneryolar altında test edilmiştir. Bu seneryolar oluşturulurken, iç su yollarında karşılaşıbilecek cevresel yapı oluştutulmaya çalışılmıştır. Oluşturulan seneryolara çevresel bozucular da eklenerek, MPC sistemi ile pid- görüş hattı rota takip algoritmasının, dar su yollarında, belirlenen rotayı takip edebilme kapasiteleri karşılaştırılmıştır. Kıllanılan simulasyon içerisnde sadece akıntı çerveresel bozucu etkisinin modellemesi yapıldığından, bozucu olarak farklı açılarda, değişen akıntı hızlarında sisteme bozucu eklenerek denetleyici ve rota takip algoritmalarının performansı test edilmiştir.

1. INTRODUCTION

The maritime industry is undergoing a transformative phase with the emergence of autonomous vessels, which have garnered significant attention due to their potential to revolutionize maritime operations. Autonomous vessels offer a multitude of advantages, ranging from the mitigation of human errors to the optimization of efficiency and cost-effectiveness. By reducing reliance on human operators, the inherent risks associated with human factors can be substantially minimized, thereby ensuring enhanced safety within the maritime environment. Numerous marine accidents have occurred throughout history, and research indicates that the majority of them are caused by human error [1]. These accidents also result in significant economic losses. Furthermore, autonomous ships have the capability to optimize energy consumption, streamline navigation paths, and facilitate remote operations, thereby resulting in improved fuel efficiency and reduced operational expenses. Especially in inland waters, the efficiency of ship navigation can be increased with the vehicles brought by autonomous technology.

Automation is used in many different applications with the goal of enhancing safety while simultaneously lowering costs. Automation encompasses a wide range of activities, from relatively straightforward tasks like robotic application to much more complicated ones like self-governing, autonomous vehicles and vessels. In the maritime sector, one of the earliest application of the usage of control systems was the use of (PID) Proportional-Integral-Derivative controllers in the 1920s for designing automatic ship steering. This is one of the oldest examples of control systems ever used. The implementation of controller approaches came after these controllers and continues to nowadays, along with other forms of controller development such as LQR (Linear Quadratic Regulator), backstepping. At present, research on autonomous ships has gained momentum, and there is currently a significant amount of research being done in this area. The ability of machines to find solutions to problems that they confront is an essential component of autonomous systems. In point of fact, considering that human error is responsible for the vast majority of ship collisions and groundings [2] therefore, research into the maritime robotics of GNC (Guidance Navigation Control) systems, which make up the primary infrastructure of autonomous systems, has obtained a lot of importance [3].

The organizations known as classification societies are responsible for creating and enforcing technical standards for ship design, building, and survey as well as for conducting on-board surveys and inspections. For the purpose of conducting required ship surveys and certification activities, flag nations may delegate authority to classification bodies [4]. With the acceleration of advancements in autonomous ship technology, class organizations are conducting efforts to establish specific standards in this field. Because it is a necessary condition to prevent the concept confusion that may arise in technology and to establish a common framework. DNV GL, ClassNk, BUREAU VERITAS and ABS have published about autonomous ships and autonomous operations guidance

1.1 Guidance, Navigation and Control

With manned systems, the guidance, navigation, and control system is carried out by the human. In autonomous systems, however, these functions are carried out by algorithms. The general structure of GNC is shown in figure 1.1. Within the scope of this thesis, studies on guidance and control sections were carried out.

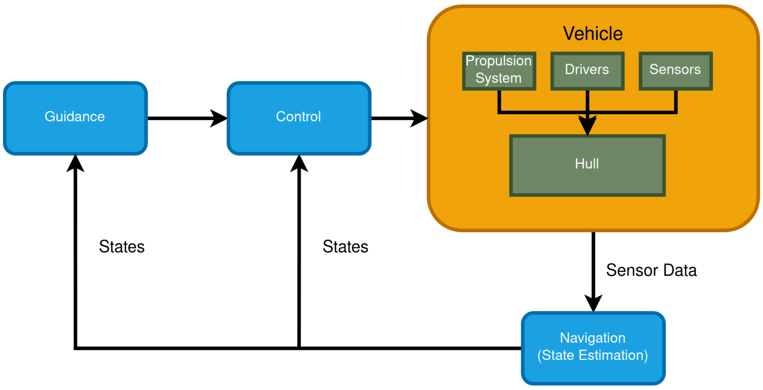


Figure 1.1 : A simplified illustration of the GNC system.

Guidance: The mission is to provide a guidance system for a marine vehicle that, utilizing motion sensors and data from external sources, can compute the vessel's reference location, velocity, and acceleration, and then deliver this information to the navigation system as well as the operator. For the purpose of determining the optimal trajectory or path, the guidance system makes use of advanced optimization techniques. These techniques include optimization of fuel consumption, navigation with the shortest possible time, weather routing, collision avoidance, formation control, and synchronization [5].

Navigation: Navigation involves determining a marine vehicle's position, course, and distance traveled. This is achieved through the use of a global navigation satellite system (GNSS), an inertial measurement unit (IMU) and motion sensors.

Control: The process of identifying the appropriate control forces and moments that need to be delivered by the craft in order to achieve a control objective is what we mean when we talk about motion control. Some examples of control objectives include following predetermined paths, controlling trajectory, and controlling maneuvering. Heading control, to manage the proper movement of a marine vehicle, and speed control, to enable it to travel at the desired speed, are essentially the vehicle states that need to be controlled. It can be considered the primary objective of controller design to track these two states, heading and speed, with appropriate RPM input signals. In this study, We can categorize the work done in this study under two main headings: The determination of an appropriate path from a given or current position of the vehicle to a desired target point is referred to as path planning. On the other hand, the act of following the generated path is known as path following. During this tracking process, efforts are made for the vehicle to avoid obstacles present in the environment. Collision avoidance is carried out according to the rules known as Convention on the International Regulations for Preventing Collisions at Sea (COLREG) that are established to prevent collisions at sea. These rules govern the actions and behaviors of vessels to ensure safe navigation and collision avoidance. For path planning, MPC and kinodynamic RRT are used, while for path following, two different methods have been used in this study. The first method involves improving the classical Line-of-Sight (LOS) method to be more useful in narrow waterways and developing a different algorithm for the lookahead distance value dependent on error to achieve better path following on the desired path in narrow areas. Secondly an MPC-based controller is developed for following desired path, which can work in environments with environmental disturbances. In addition, in working conditions where disturbance forces are high, the MPC-PID controller approach has been developed. When we look at the marine craft systems that we will be inspecting for control, we can classify them into two different groups: systems operating in open seas and systems operating in narrow waterways. Due to the difference in working environment, the constraints we will add to the solution of the MPC controller will vary. In open sea operations, a solution will be sought that takes into account wave and wind effects, as well as fuel and time optimization. However, in narrow waterways, it may not be necessary to consider wave effects. In this study, since we are working on path planning in a limited area where obstacles are close to each other, we just take current speed disturbance into account in our calculations.

1.2 Guidance

Guidance is done by path planning. Path planning is the name given to the process of generating a path for autonomous sea, air, and land systems from a predetermined starting point to a target point in free space environment or while avoiding static or dynamic obstacles in the environment. The path is designed to reach the target point while avoiding obstacles along the way. During the determination of this path, objective functions such as minimum fuel consumption, minimum travel time, or a path suitable for the vehicle dynamics can be added to the calculation processes, and path optimizations can be made according to the requirements. There are different path planning methods with different algorithmic structures in the literature. It is devided to two main catagories as traditional and Optimization-based algorithms. Traditional algorithms can be divided into four different groups: graph-search, sampling-based, interpolating curve, and reaction-based algorithms [6]. Graph-search based algorithms such as A* and Dijkstra have a high computational cost in large maps [7]. Reaction-based algorithms are used in local planning. A path generated solely using the optimization-based algorithm which structure is shown figure 1.2

may has a high computational cost for dynamic environment. Due to these reason, the kinodynamic RRT algorithm, which has relatively low computational cost and is widely used in the industry [6], has been used as the path planning algorithm in this study.

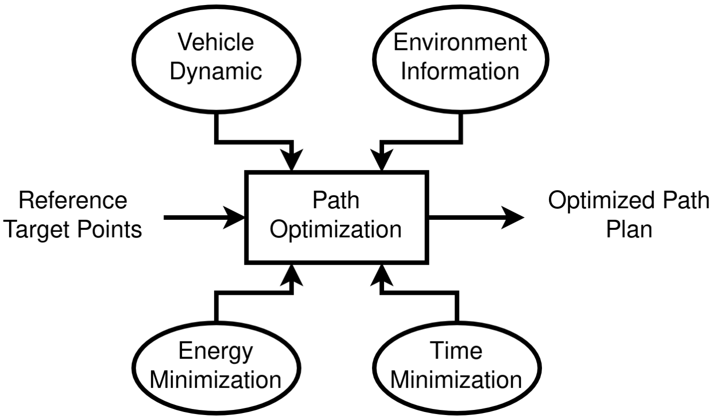


Figure 1.2 : Optimized path plan block shame.

1.3 Navigation

Navigation is general name of path following concept in marine field. Path following algorithms are used to guide a vehicle along a predetermined path or trajectory while avoiding environmental obstacles. The algorithm calculates the difference between the current vehicle position and the intended path, using the desired path as a reference. The goal is to make an Unmanned Surface Vehicle (USV) keep up with a specified path, mainly to meet the constraints in space, and the requirement for time is not very strict [8]. In this study, the LOS path following algorithm was used to follow the path created to reach the destination, and the desired path was followed using different control methods. There are cases where the classical LOS algorithm cannot work on paths that require sudden changes. To solve this problem, some modifications have been made to the operation of the LOS algorithm. This is explained in detail in the path following section.

1.4 Vehicle Specification

The Otter USV is developed by Maritime Robotics AS, Norway that is shown in figure 1.3 has two propulsion systems As physical feature, it is a catamaran type boat and has 200 cm in length and 105 cm in width and has differential drive propulsion system with an approximate thrust of 120 N. In this study, the mathematical model developed by Fossen has been utilized for this vehicle.

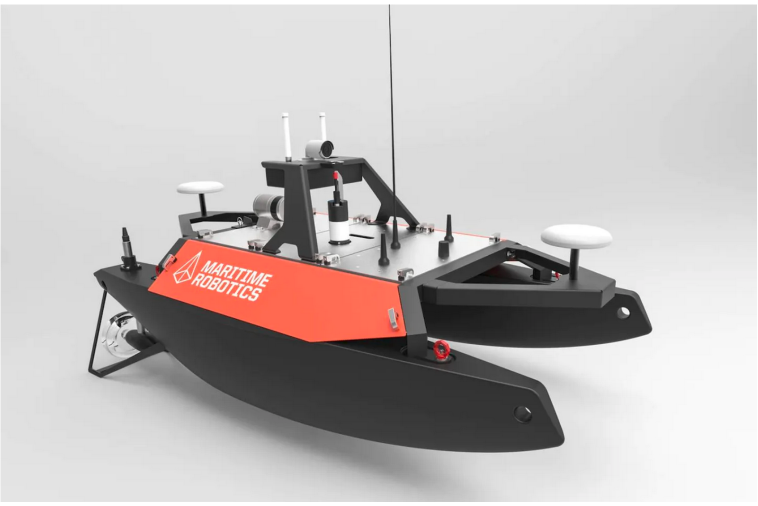


Figure 1.3 : Otter USV.

1.5 Outline

This thesis is divided into seven different chapters. Chapter 2, In the second chapter, information is presented on the dynamic and kinematic behavior of the vehicle, which is the system that is going to be managed, as well as information regarding generic reference systems. Chapter 3, provides information about path planning. The LOS working logic, also known as the path following algorithm, is the primary subject under study in Chapter 4. Chapter 5 explain the theoretical background of MPC, which is the controller method, which is the main subject of this thesis. Chapter 6 shows controller design and implementation to the marine vehicle. In Chapter 7 controller performance of MPC and PID compared in different scenarios . In Chapter 8, the thesis has been summarized in general, and future work has been discussed.

2. MARINE VESSELS DYNAMIC MODEL

The modeling of marine vessels is based on the forces and moments that are applied to their bodies. Maneuvering and seakeeping are two theories that describe the dynamics of marine vessels. While the maneuvering theory defines the motion of a vehicle in a no-wave water condition in surge, sway, and yaw, seakeeping theory describes hydrodynamic forces caused by waves at constant speed and direction. The study of dynamics can be divided into two parts: kinematics, which treats only geometrical aspects of motion, and kinetics, which is the analysis of the forces causing the motion [9].

Controlling the vehicle in the x-y plane is sufficient to carry out the tasks of guidance and navigation, as shown in Figure 1.1. These can be achieved by controlling the vehicle's surge, sway, and yaw axes. Roll, pitch, and yaw movements are accepted as zero.

2.1 Coordinate Frames

The position, rotation, and velocities of a displacement vessel are fully defined by six degrees of freedom (DOF), which are modeled as a single rigid body. Rigid-body dynamics serve as the foundation for current models used in speed prediction, time-domain simulation, and motion controller design. In order to characterize the motions of a ship and assign them specific names, a body-fixed reference frame b is established with its origin at O_b. The principle axis of the vessel determines the direction of the orthogonal axes x_b , y_b , and z_b within b. The movements of the ship, as depicted in the body frame, can be converted to coordinate frames with a fixed center. The commonly used coordinate frames are North-East-Down (NED) and Earth Centered Earth Fixed (ECEF). The illustration of coordinate frames are shown in figure 2.1.

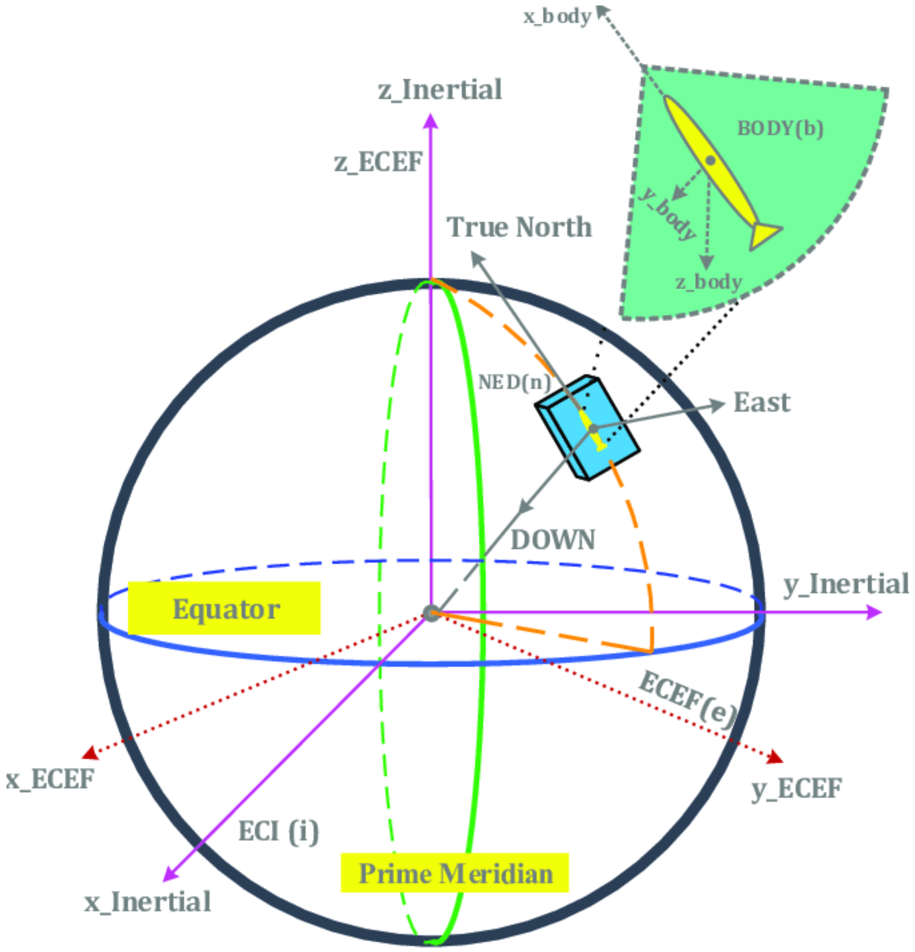


Figure 2.1 : NED and ECEF coordinate frames illustration [10].

As demonstrated in the body frame, the positive direction of x_b is towards the bow, the positive direction of y_b is towards the portside, and the positive direction of z_b is pointing upwards. In shipping, the velocities along the x_b, y_b, and z_b axes are known as surge, sway, and heave velocity, respectively, and are represented by u, v, and w. The rotational velocities about the x_b, y_b, and z_b axes are known as pitch, roll, and yaw, respectively, and are denoted by p, q, and r.

The linear velocities (u, v, w) are considered positive along the respective positive directions of the body-fixed axes. The angular velocities of pitch, roll, and yaw (p, q, r) are positive in accordance with the right-hand rule applied to the axes of b. To capture the position and orientation of the vessel in space, an earth-fixed reference frame n is utilized with O_n as its origin. The coordinates x, y, and z capture the position of

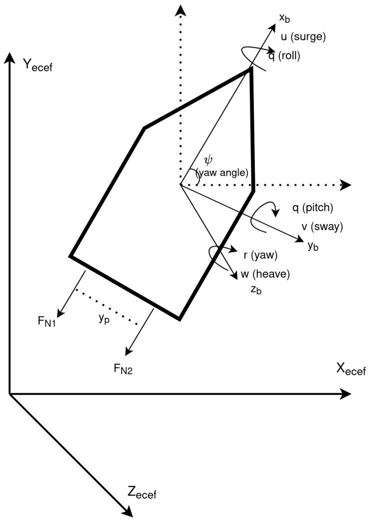


Figure 2.2 : Manoeuvering Coordinate System.

O_n, while the vessel's orientation is described by the Euler angles (ϕ, θ, ψ). General notation of underactuated differential drive USV is shown figure 2.2.

2.2 Rigid Body Dynamics

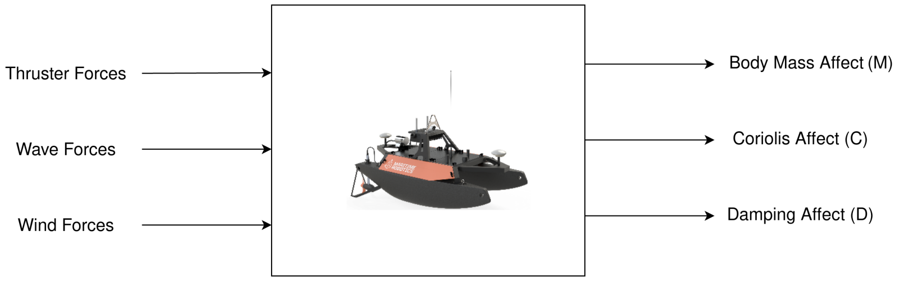


Figure 2.3 : Dynamic model of vehicle

In this thesis, 3 DOF dynamic of vehicle is used for controller design. The general structure of the components that make up the dynamic equations is shown in figure 2.3. The motion equations for a rigid body with respect to the body-fixed reference frame are determined using the Newton-Euler equation. This frame's origin does not need to coincide with the vessel's center of gravity (COG). If the position of the center of gravity (COG) in b is denoted by $r_g = [x_g, y_g]$, then the rigid body dynamics are given by

Table 2.1 : The notation of SNAME for marine vessels.

DOF	Forces and moment	Lin. and ang. vel.	Positions and euler ang.
1	X	u	x
2	Y	v	y
3	Z	w	z
4	K	p	ϕ
5	M	q	θ
6	N	r	ψ

$$m[\dot{u} - vr - x_g r^2 - y_g \dot{r}] = X \tag{2.1}$$

$$m[\dot{v} - ur - y_g r^2 - x_g \dot{r}] = Y \tag{2.2}$$

$$I_z \dot{r} + m[(\dot{v} + ur) - y_g(\dot{u} - vr)] = N \tag{2.3}$$

The notation that is constituted by SNAME (1950) [11] is show table 2.1

X, Y and N are forces acting on the vessel [9]. The forces are hydrostatic, hydrodynamic, environment, and actuator forces, and these are shown as τ_{hs}, τ_{hyd}, $\tau_{wind} + \tau_{wave}$ and τ_u respectively.

The rigid-body kinetics (equation 2.1-3) can be expressed in a vectorial setting as [12]

$$\mathbf{M}_{RB}\dot{v} + \mathbf{C}_{RB}(v)v = \tau \tag{2.4}$$

$\mathbf{M}_{RB}$ is the rigid-body matrix, $\mathbf{C}_{RB}(v)$ is inertia matrix that include coriolis and centripetal forces.

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \quad C_{RB}(v) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v)) & -mu & 0 \end{bmatrix} \tag{2.5}$$

The velocity vector v contains vessel velocities in the body-fixed frame

$$\vec{v} = \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{2.6}$$

Body-fixed movements are transferred to the earth-centered coordinate system for the display of ship movements in the earth-centered coordinate system. Earth-fixed coordinates are displayed in the η vector.

$$\vec{\eta} = \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} \tag{2.7}$$

$R(\psi)$ rotation matrix is used to transform earth-fixed frame movements from body-fixed frame. this conversion general equation.

$$\vec{\eta} = \mathbf{R}(\psi)\vec{v} = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{2.8}$$

2.3 Added Mass

The added mass is referred to as an additional mass of fluid caused by the acceleration of the vehicle body in the fluid. This has a nonlinear physical effect on the dynamic behavior of vehicles. Therefore, when creating models of marine vehicles, the effect is taken into account. The effect varies for the vehicle at different speeds and takes on different values, but for this study, fixed values were taken for different speeds. The effect is observed on the mass $\mathbf{M}_A$ term and the coriolis $\mathbf{C}_A$ term of the vehicle. This term is shown below.

$$M_A = \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{2.9}$$

$$C_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & -X_{\dot{u}}u & 0 \end{bmatrix} \tag{2.10}$$

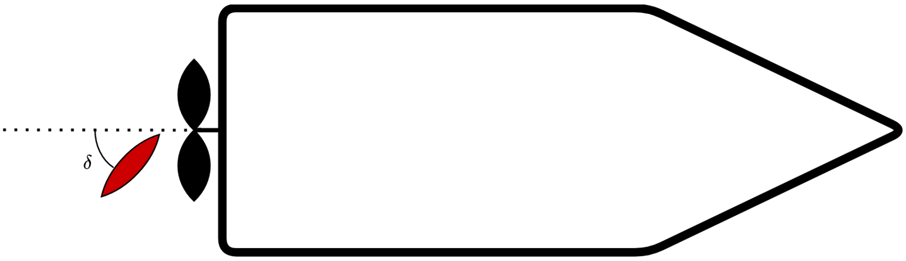


Figure 2.4 : Single rudder and single propeller system.

2.4 Thruster Allocation

There are different thrust allocation models available depending on the structure of a ship's propulsion system. For example, the modeling of a two-propeller propulsion system would differ from that of a single-propeller and rudder propulsion system.

2.4.1 Single rudder and single propeller model

The Nomoto yaw model is a method that can be used for thrust allocation modeling of ships' propulsion systems consisting of a single propeller and a rudder and shown in figure 2.4. The Nomoto yaw model is a mathematical model used to calculate the required rudder angle to turn a ship's head. This model explains the relationship between the ship's movement and the rudder angle and is used to predict the performance of the ship during turning motion.

$$T\dot{r} + r = K\delta \tag{2.11}$$

In equation, K represents tuning parameter, T is time coefficient δ is the actual rudder angle and r is yaw velocity. [13].

2.4.2 Dual rudder and dual propeller model

In this propulsion system model, the sum of the forces exerted on the ship's hull by the propeller and rudder determines the absolute force and moment acting on the ship's hull. it is shown in figure 2.5.

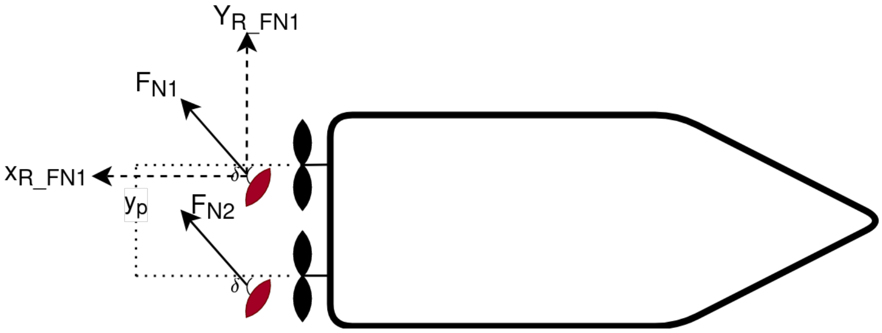


Figure 2.5 : Dual rudder and dual propeller system.

$$
\begin{aligned}
X &= X_H + X_P + X_R \\
Y &= Y_H + Y_P + Y_R \\
N &= N_H + N_P + N_R
\end{aligned}
\tag{2.12}
$$

X, Y, N show total force on the vehicle body on axis. X_H,Y_H and N_H represent hull forces on axis; X_P, Y_P, and N_P show total propellers on axis and X_R, Y_R, and N_R show rudder for on axis.

Rudder force:

In twin rudder, model force model is shown;

$$
\begin{gathered}
X_R = 1(1 - t_R)(F_{N1} + F_{N2}) \sin\delta \\
Y_R = (1 - t_R)(F_{N1} + F_{N2}) \cos\delta \\
N_R = [y_p(1 - t_R)(F_{N1} - F_{N2}) \sin\delta - (X_R + \alpha_H X_H)(F_{N1} + F_{N2}) \cos\delta]
\end{gathered}
\tag{2.13}
$$

t_R,α_H and X_H are the hull interaction coefficients, F_{N1} and F_{N2} are port and starboard rudder forces.

propeller forces are expressed at blow;

$$
\begin{gathered}
X_P = (1 - F_{N1})(T_1 + F_{N2}) \\
Y_P = 0
\end{gathered}
\tag{2.14}
$$

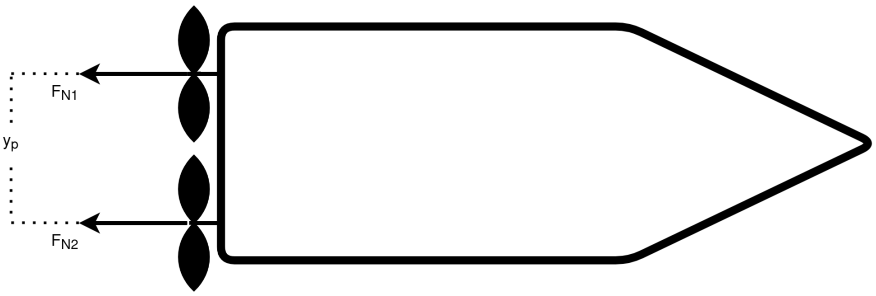


Figure 2.6 : Dual propeller without rudders.

$$N_P = (1 - t_p)(F_{N2} - F_{N1})(y_P/2)$$

where t_p is a thrust deduction, y_P is differences distance between propellers. T_1 and T_2 are the propeller thrusts of port and starboard, respectively.

Consequently, rudder and propeller force summation gives the total force and moment values of the dual propeller and dual rudder propulsion systems.

2.4.3 Dual propeller model without rudders

The Otter USV used in this study has two thrusters. Otter USV has a catamaran structure and therefore two thrusters at the stern. F_1 and F_2 forces act on x axes.

$$M_p = (F_{N1} - F_{N2})y_p/2 \tag{2.15}$$

$$T_p = F_{N1} + F_{N2} \tag{2.16}$$

Because of thrust differences show up moment force on body. y_p is length between two thrusters is shown in figure 2.6.

2.5 Hydrodynamic Coefficients

Hydrodynamic parameters are critical for the design and operation of ships and marine structures, and there are various methods for determining them. Towing tank tests, computational fluid dynamics (CFD) and system identification are the main methods for this.

The conventional method for determining hydrodynamic parameters involves conducting towing tank tests, which entail fixing a scale model onto a towing carriage and subjecting it to surge and sway motion while measuring the corresponding hydrodynamic and damping forces. Several tests, including straight-line, rotating-arm, and planar-motion-mechanisms tests, are performed to isolate multiple parameters for identification per test, thereby facilitating their numerical identification. However, identifying a comprehensive maneuvering model requires a substantial number of towing tank tests, resulting in an expensive and time-consuming identification process. As such, alternative methods like computational fluid dynamics (CFD) simulations and model-scale experiments in large wave basins have emerged as more cost-effective and efficient methods for determining hydrodynamic parameters. Nonetheless, these methods require thorough calibration and validation to ensure accurate results. In particular, physical experiments in wave basins are limited by the available wave conditions, which may not adequately represent the full range of operating conditions for the actual vessel.

SI is a mathematical field that leverages statistical techniques to construct models of dynamic systems using output measurements from the system. The methodology of system identification involves estimating the parameters of a system through the minimization of the error discrepancy between the system's output values and the model's output values. This is achieved by utilizing statistical methods to build dynamic system models that accurately represent the behavior of the actual system.

In the existing literature, several parameter estimation algorithms have been employed to estimate vessel parameters using output measurements from the system. Extended Kalman Filter (EKF) [14], Least Squares (LS) are two of the popular methods [15] for parameter determination of ship.

Hydrodynamically, these forces arise due to the interaction of the boat's hull with the water when the boat is in motion. [16] and [17] use the Taylor series that is utilized to represent the hydrodynamic reaction forces in a non-linear form.

$$X_{hyd} = X_{\dot{u}}\dot{u} + X_{u}uX_{|u|u}|u|u$$

$$Y_{hyd} = Y_{\dot{r}}\dot{r} + Y_{\dot{v}}\dot{v}Y_v v + Y_r r + Y_{|v|v}|v|v + Y_{|v|r}|v|r + Y_{|r|v}|r|v + Y_{|r|r}|r|r \tag{2.17}$$

$$N_{hyd} = N_{\dot{r}}\dot{r} + N_{\dot{v}}v + N_v v N_r r N_{|v|v}|v|v + N_{|v|r}|v|r + N_{|r|v}|r|v + N_{|r|r}|r|r$$

$X_{\dot{u}}$, $Y_{\dot{r}}$, $Y_{\dot{v}}$, $N_{\dot{r}}$, and $N_{\dot{v}}$ are showed added mass. X_u, Y_v, Y_r, N_r, and N_v represent linear effect of damping force and term has absolute ($|u|u$) etc. shows quadratic term of damping.

2.6 Dynamic Model of Vehicle

The general from of dynamic model of marine vehicle in three DOF.

$$\dot{\eta} = R(\psi)v \tag{2.18}$$

$$M\dot{v} + C(v)v + D(v)v = \tau_{total} \tag{2.19}$$

Where $\eta = [x, y, \psi]$ and $\nu = [u, v, r]$ and Rotation matrix is

$$\mathbf{R}(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

Inertia matrix of vessel is

$$M = M_{RB} + M_A = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mg_g \\ 0 & mx_g & I_z \end{bmatrix} + \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix} \tag{2.21}$$

Coriolis and centripetal matrix

$$C(v) = C_{RB} + C_A = \begin{bmatrix} 0 & 0 & -m(x_g r + v)) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & -X_{\dot{u}}u & 0 \end{bmatrix} \tag{2.22}$$

Damping matrix consist of D_L and D_{NL}

$$D(v) = D_L + D_{NL}(v) \tag{2.23}$$

$$D_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{2.24}$$

$$D_{NL} = \begin{bmatrix} -X_{|u|u}|u| - X_{uu}u^2 & 0 & 0 \\ 0 & -Y_{|v|v}|v| - Y_{|r|v}|r| & -Y_{|v|r}|v| - Y_{|r|r}|r| \\ 0 & -N_{|v|v}|v| - N_{|r|v}|r| & -N_{|v|r}|v| - N_{|r|r}|r| \end{bmatrix} \tag{2.25}$$

2.7 System Identification

Unlike the rotating vehicle model in the otter simulator, we will investigate the derivation of our own mathematical model based on the cruise data of the vehicle. The reason for doing this is that having the same model in the MPC model as the rotating model in the vehicle simulator may lead to questioning the accuracy of our algorithms and may also provide misleading information about the proper functioning of the controller. Therefore, the mathematical model that will work in the MPC must be the one derived by us, so that we can ensure the correct performance of the controller. System identification methods have been used to create this model.

In the system identification section, it has been explained which type of maneuver is performed and which parameters are estimated from these data to find the dynamic equation of the vehicle.

The task of developing mathematical models of dynamical systems based on observable data from the system is the subject of the field of study known as system identification [18]. In other words, it is the process of determining the mathematical correlations that exist between the signals that are input into a system and the signals that are produced from the system. The method is used in various fields such as control engineering, signal processing, robotics, and aerospace engineering for different purposes, like simulation, control, prediction, and optimization. According to the structure of the mathematical equation of the system that is known, the systems are called white-box, gray-box, and black-box. In the gray box, even when the inner workings of a system are not completely understood, a model is developed based on system knowledge and experimental data. Yet, this model has a number of undetermined free parameters that may be calculated through system identification

Table 2.2 : System dynamic equation parameters.

Category	Parameters
physical Futures	m, I_g
surge movement	X_u, $X_{u\|u\|}$,$X_{\dot{u}}$
yaw movement	N_r, $N_{r\|r\|}$,$Y_{r\|v\|}$,$Y_{r\|r\|}$,$N_{r\|v\|}$,$N_{r\|r\|}$

[19]. The Black-box has previous model is not available. The vast majority of system identification methods are of this kind.

2.8 SI Method for USV

While identifying the system, we can calculate the general equation of the system equation by dividing the general dynamic equation of USV into sub-equations. Parameters can be calculated by separating the ship's movements into surge, sway, and yaw movements into sub-equations. The system's general equation of motion is given by equation 2.25 [9]. The following table 2.2 categorizes the sub-equation-categorized hydrodynamic coefficients of the dynamic equation.

$$M\dot{v}+C(v)v+D(v)v+g(\eta)=\tau_{thrusts}+\tau_{wind}+\tau_{wave} \tag{2.26}$$

The coefficients of equation 2.26's dynamic equation can be categorized as given in the table 2.2.

2.8.1 Identification of hydrodynamic surge parameters

To derive the differential equation describing the linear speed, it is first necessary to generate a data set of linear speeds corresponding to these rpm values by applying identical rpm signals at various values to the two engines of the vehicle. In figure 2.7. Input signals are shown. With this movement, the effect of the yaw and sway movements of the surge speed movement is eliminated, and the coefficients of the surge speed equation are calculated. Parameter estimation of the surge speed differential equation is calculated nonlinear least squares method in the Matlab Optimization Toolbox module. The setting of the nonlinear least squares method algorithm is adjusted to 60 iteration loops for each calculation, and the estimator parameter

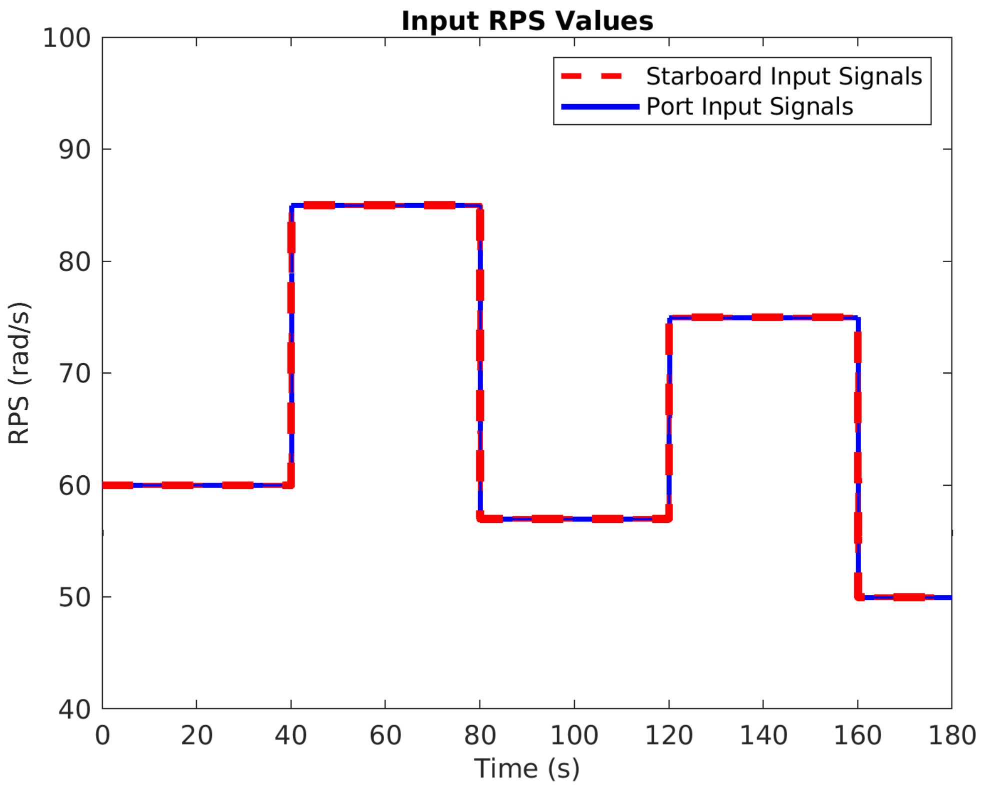


Figure 2.7 : Input signals for surge speed dataset

tolerance is set to $10e^{-6}$.

$$\dot{u} = \frac{X_u|u| + X_{u|u|}u|u|}{(m - X_{\dot{u}})} \tag{2.27}$$

The surge acceleration equation is written as equation 2.27. To estimate the coefficients of the surge speed equation, the input and output data of the system are given to the nonlinear least squares approximation, and the coefficients are calculated. The calculated hydrodynamic coefficients of the surge speed equation are shown in Table 2.3. The comparison of the surge speed equation derived by the SI method and the surge speed corresponding to the input RPM of the dynamic model of the Otter USV is shown below. The similarity between two speed values is %99.93. Otter USV surge velocity and surge velocity of SI model responses are shown in figure 2.8.

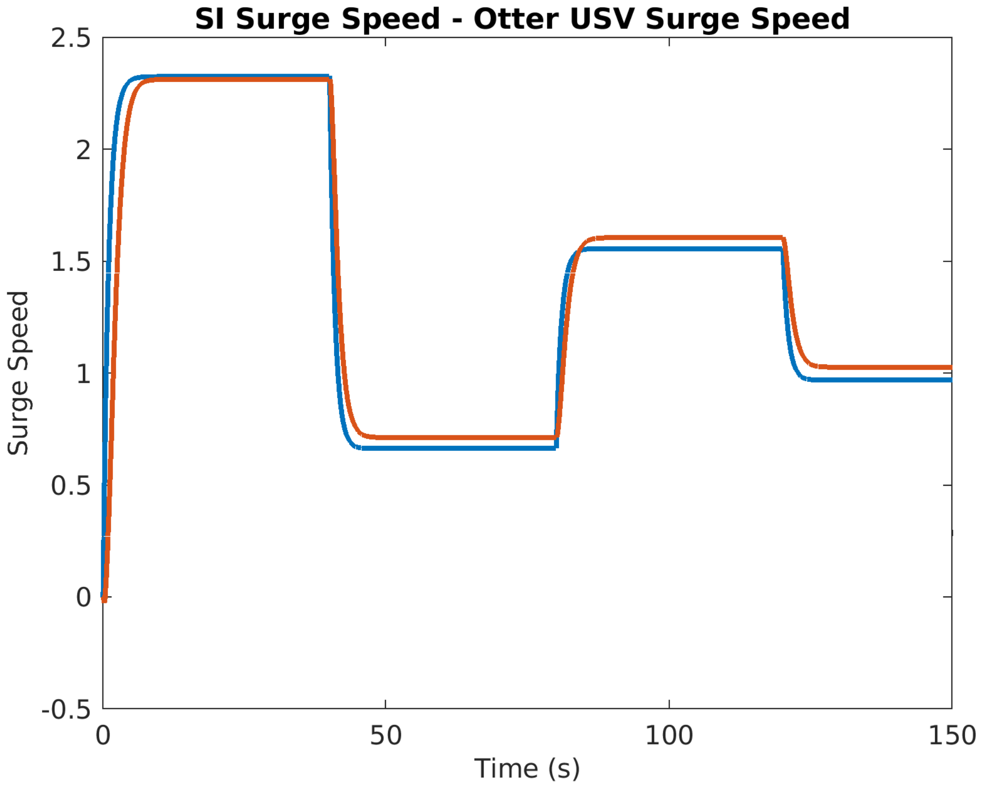


Figure 2.8 : The surge experiments with system output and model output with estimated surge parameters.

2.8.2 Identification of hydrodynamic sway and yaw parameters

We are unable to distinguish yaw movement from other ship movements, such as surge speed. Hence, yaw and sway movements are coupled. The consequences of sway speed are observed in yaw motion. In order to model the yaw movement, a dataset is created with inputs to the vehicle's engines at different rpm values. The equation 2.28

Table 2.3 : System dynamic equation parameters.

Parameters	Value
$X_{\dot{u}}$	12.51
X_u	-36.1331
X_{uu}	1.5572

Table 2.4 : System dynamic equation parameters.

Parameters	Value	Parameters	Value
$N_{\dot{r}}$	11.0643	Y_v	-0.2192
N_v	0.0017	Y_r	1.3684
N_r	-3.9154	$Y_{\lvert v\rvert v}$	-0.0422
$N_{\lvert v\rvert v}$	1.995	$Y_{\lvert v\rvert r}$	-3.5344
$N_{\lvert v\rvert r}$	-18.1398	$N_{\lvert r\rvert v}$	2.3178
$N_{\lvert r\rvert v}$	-5.5441	$N_{\lvert r\rvert r}$	-23.7569
$N_{\lvert r\rvert r}$	-18.5398	$N_{\dot{v}}$	-21.1184

shows the force-angular velocity dataset and the equation describing its coefficients.

$$\dot{r} = \frac{N_v v + N_r r + N_{|v|v}|v|v + N_{|v|r}|r|v + N_{|r|v}|v|r + N_{|r|r}|r|r}{(I_g - N_{\dot{r}})} \tag{2.28}$$

As the dynamic equation of the system is known, the gray-box approach for system identification is determined by identifying the coefficients of the system equation. In order to determine the computed parameters of the equations, the nonlinear least squares approach was used to estimate the parameters. A comparison of the mathematical model of the vehicle obtained by system identification with the actual system behavior is shown in figure 2.9 and the similarity rate is %98.76.

Consequently, the ship hydrodynamic coefficients were calculated with the nonlinear least squares method using the dataset of the linear motion for the surge speed and the zigzag motion for the yaw, which were created by using the Otter simulator. This estimated mathematical model will be used as the model of the system in MPC.

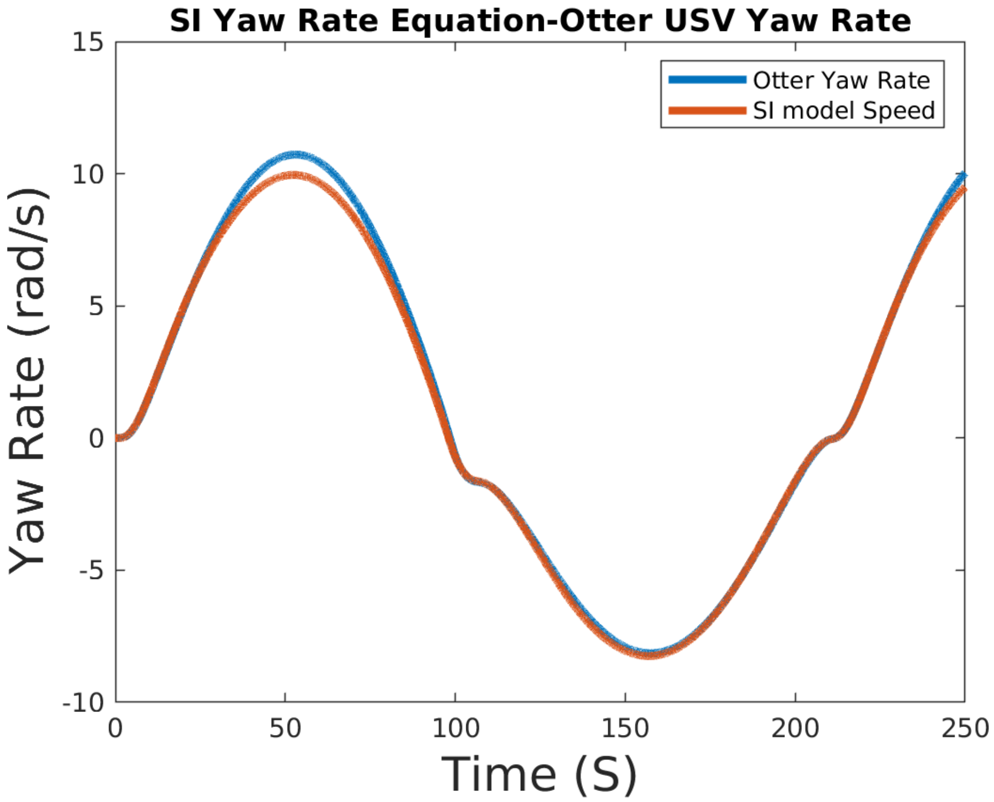


Figure 2.9 : The Heading experiments with system output and model output with estimated yaw parameters.

3. GUIDANCE

3.1 Introduction

The guidance is the part where the development of general path planning algorithms is explained and two different approaches for finding paths are examined. The first approach is a sample-based approach called the RRT path planning algorithm is explained. It is described how the constraint that takes into account the dynamics of the vehicle works on this algorithm and where optimizations can be made to increase the calculation speed. Secondly, optimization-based path planning uses to explain how a general path can be determined from the starting point of the vehicle to a target point and explain how to modify a cost function that can be modified according to the desired constraints.

Path planning is the process of determining the optimal path for a vehicle or a robotic system to take to reach its destination while avoiding obstacles and ensuring safety. Utilizing algorithms and sensors to analyze the surrounding environment and determine the vehicle's trajectory. Path planning is a crucial component of autonomous navigation because it enables unmanned vehicles to operate safely and efficiently in dynamic, complex environments [20]. Commonly employed optimization criteria for path and trajectory include minimization of path length, time, and energy consumption, as well as safety or risk measurements. In addition, path planning is typically defined within a purely geometric space, whereas trajectory planning, or trajectory generation, involves geometric trajectories with temporal properties, for instance to integrate dynamics [21]. Path planning algorithms used in autonomous vehicles or robotic systems share the same basic concept, but there are differences arising from the various constraints imposed by the operating environment and the systems for which the path is being planned.

During the general path planning for marine vessels, path calculations can be made taking into account various situations such as energy optimization, time optimization.

It is possible to calculate this using smart algorithms, such as MPC, by taking into account various constraints while also considering the dynamic model of the vehicle. However, these calculations may have very high computation costs or may need long calculation time.

Although this thesis focuses on using the RRT algorithm to create a path for a vehicle to reach a desired point in an environment with static obstacles, taking into account the dynamic behavior of the vehicle, an example of path planning that is suitable for the behavior of the Otter USV is also created in this section, utilizing the optimization-based path planners developed within the scope of the thesis.

3.2 Sampling-Based Planning

The most common method for differentially constrained planning in high dimensions is sampling-based planning. Planners, based on sampling, investigate the free state space via discretization. The connectivity of a free state is determined by randomly selecting states known as vertices and joining them to edges using a steering method. A collision detection function verifies that vertices and edges are collision-free before adding them to a graph. It is shown in figure 4.3. Many sample-based planner methods have been developed at present, and one of the most important examples is the RRT method. RRT searches high-dimensional spaces efficiently by constructing a tree of randomly generated feasible trajectories.

The algorithm grows a tree iteratively from an initial configuration (typically the starting point) to the target configuration (typically the destination). At each iteration, the algorithm generates a random configuration in the search space and locates the configuration in the tree that is the closest match. It then attempts to connect the new configuration to the configuration closest to it in the tree by extending it along a feasible path. This procedure is repeated until the desired configuration is achieved or a predetermined maximum number of iterations has been attained. However, this random point sampling-based approach for finding a path to the target point may not generate suitable paths for systems that need to follow the trajectory. The generated paths may not be feasible or safe for the system to follow, especially in the presence of obstacles or constraints. To generate paths that are suitable for vehicle dynamics, there is a method called the kinodynamic RRT method that takes into account the system's dynamics and generates paths accordingly. This approach incorporates the system's dynamics into the algorithm to ensure that the generated paths are feasible and safe for the system to follow, taking into account the vehicle's motion constraints and limitations.

3.2.1 Kinodynamic RRT

[22] provides a summary of the kinodynamic RRT algorithm as a sampling-based motion planning algorithm. Which is especially advantageous for motion planning tasks involving robots or vehicles with non-holonomic constraints or other dynamic constraints that affect their motion. The Kinodynamic RRT algorithm extends the fundamental RRT algorithm by incorporating the robot's or vehicle's dynamics during planning. Specifically, kinodynamic RRT incorporates into the planning process the robot's or vehicle's kinematic and dynamic constraints, such as maximal acceleration, velocity, and turning radius. This enables kinodynamic RRT to generate feasible and optimal trajectories through complex environments, even when obstacles are present.

3.2.1.1 Algorithm

Analyzing how the algorithm works. A path is initiated at a q_{init} start point, and then the algorithm randomly samples the configuration space to generate a new configuration q_{rand}. Then employs a local planner to find the closest node, call as q_{near} in the existing tree. Next, using the steering method that is visuliaze in figure 3.3, find a q_{new} node that is a suitable point according to vehicle dynamics and constraints. In the developed algorithm, the possible vehicle trajectory is calculated for the $Deltat$ interval with certain moment value increments in the specified moment values. The navigation method step is part of the procedure for generating feasible paths for a vehicle. This step takes into consideration the vehicle's dynamics when designing these paths. Predefined random input values are input into the vehicle's dynamic equation based on the vehicle's dynamic constraints. Using a cost function that evaluates the minimum distance between possible q'_{new} points and the goal point, the results are evaluated, and the most effective path is chosen. This stage ensures that the

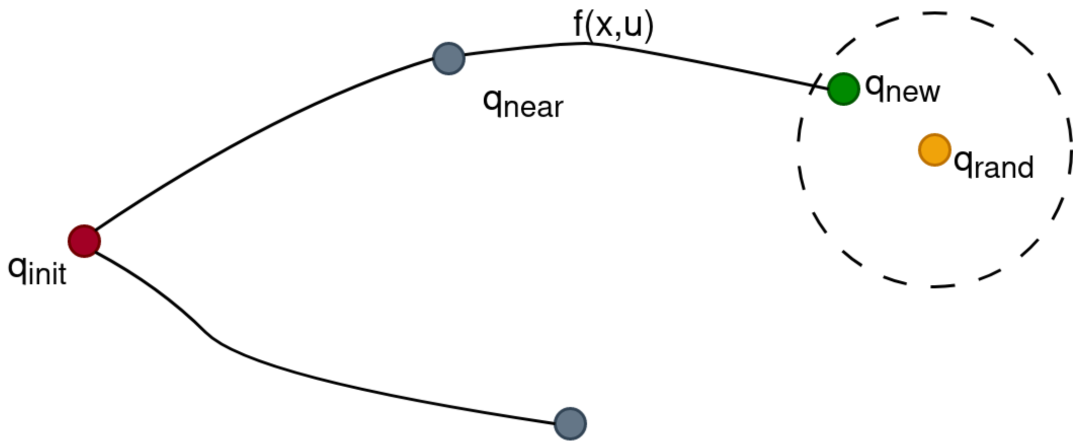


Figure 3.1 : Kinodynamic RRT illustration

generated paths are secure, feasible, and optimized according to the intended objectives by incorporating the vehicle's dynamic constraints. Next, it checks for collisions, and adds the new node to the tree if it is collision-free. The algorithm continues to explore the configuration space until a feasible path from the start to the goal configuration is found. A visualization of creating tree branch operations for each loop is shown in figure 3.1.

The addition of generating a suitable path with dynamic equations within the RRT algorithm increases the computational cost. To reduce this cost in the steering method stage, paths that comply with these constraints were calculated by giving the system a previously determined forward force and predefined yaw moment list values as input. Figure 3.4 shows the suitable movement behavior of the vehicle. When looking at the literature, the RRT* method is used to obtain the optimal path path. However, this approach includes a computational cost that increases the computational load. In this study, we attempted to optimize the path with low computational cost by adding a term to the cost function of the RRT algorithm that penalizes the distance between the target point and the current node. The entire flowchart of the described algorithm is shown in Figure 3.2.

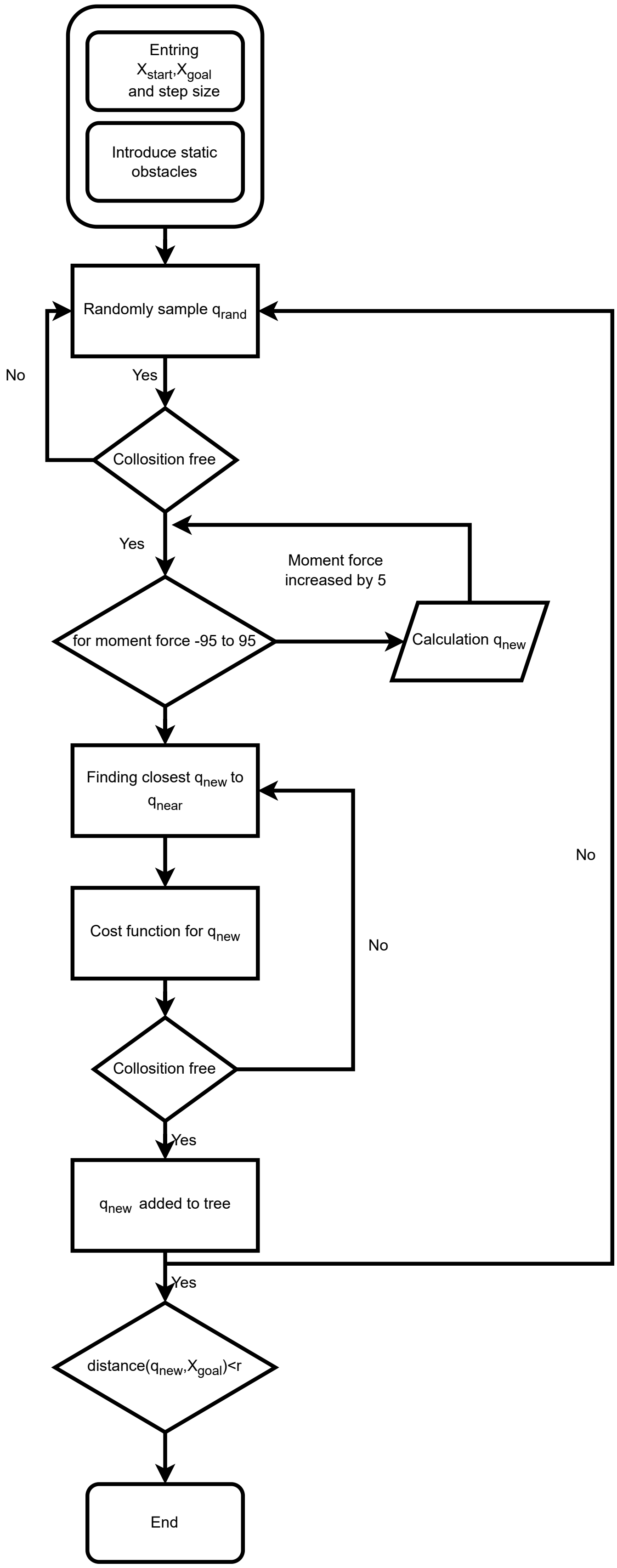

Entring
X_{start},X_{goal}
and step size
Introduce static obstacles
Randomly sample q_{rand}
No
Yes
Collosition free
Yes
Moment force increased by 5
for moment force -95 to 95
Calculation q_{new}
Finding closest q_{new} to q_{near}
No
Cost function for q_{new}
No
Collosition free
Yes
q_{new} added to tree
Yes
distance(q_{new},X_{goal})<r
End

Figure 3.2 : Kinodynamic RRT algorithm flow chart

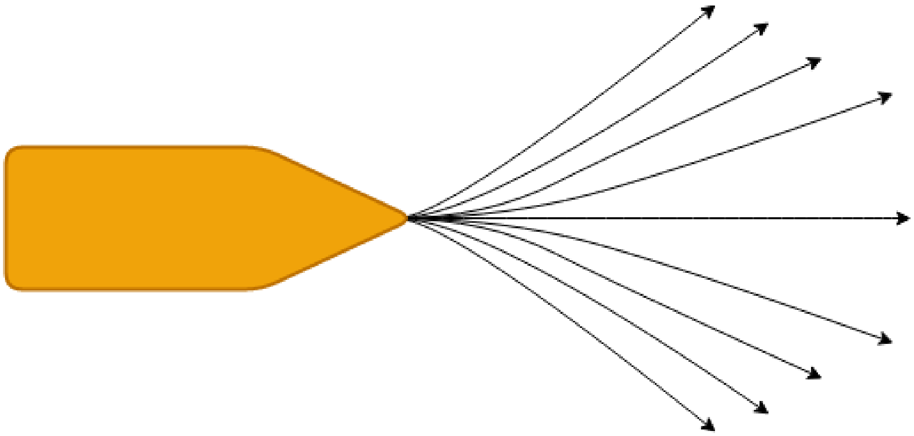
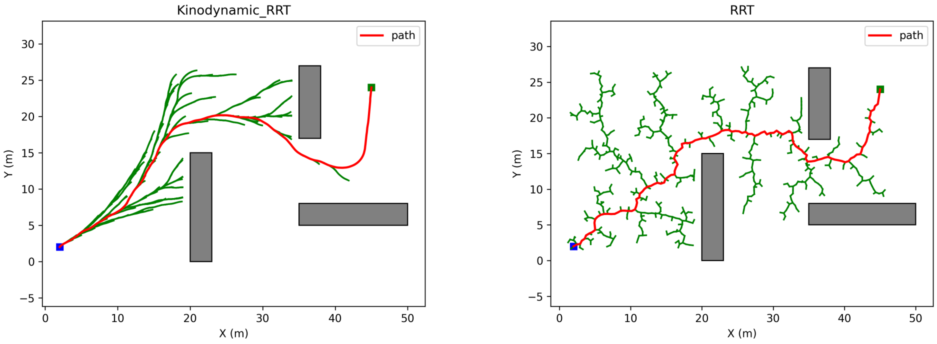


Figure 3.3 : Steering method possible vehicle maneuver



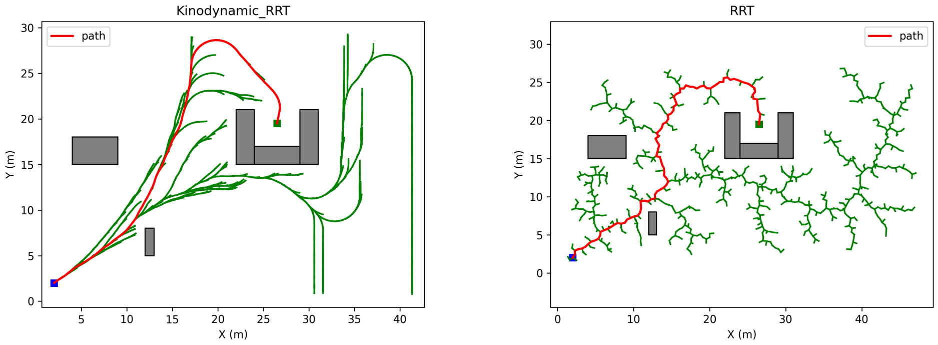
(a) Kinodynamic RRT

(b) RRT

Figure 3.4 : Kinodynamic RRT - RRT path planning

The path generation patterns between the method that takes into account the dynamic equation and the normal method in the path planning method used are shown in figure 3.4-a When a blue point represents the start point, the green point represents the goal point. In the method where the dynamics are taken into account, as seen in the generated path, it has a structure that a marine vehicle can follow. However, in the method where the normal algorithm is used, it is very difficult for the vehicle to follow the generated path, or it is not appropriate to follow such a path.

In Figure 3.5, a scenario environment suitable for generating a path to the specified point within a port structure is created in an environment with two static obstacles.



(a) kinodynamic RRT

(b) RRT

Figure 3.5 : Kinodynamic RRT - RRT docking planning

When we look at the path generation performances of the two methods, it is clearly seen that the kinodynamic RRT method creates a path that the vehicle can follow.

3.2.1.2 Performance comparison of rapidly-exploring random tree (RRT)

When designing path generation algorithms, it is important that they exhibit successful performance on the real system as well. Therefore, it is desired that the consumption of resources such as CPU, RAM, etc. by the created algorithms be as low as possible. In this section, the performance of the path planning algorithm that generates a path using RRT considering the vehicle dynamics used in this study was examined. Using the scenarios shown in Figures 3.4-5, the parameters affecting the performance of kinodynamic RRT were changed, and the CPU, RAM, and computational cost values were calculated. The step length, which determines how far the q_{rand} value should be placed from the tree in the algorithm, and the sampling time Δt values of the Euler equation, which the vehicle dynamics operates on, are parameters that affect the performance of the algorithm. Because the RRT algorithm is random-based, the resulting paths and computation times may vary under identical conditions. Even though Kinodynamic RRT generates paths that are compatible with vehicle dynamics, these paths do not guarantee to be the most optimal in terms of factors such as time or energy consumption. It can be used to generate fast solutions in scenarios with high congestion, such as inland waterways. However, determining the general path with various constraints and achieving smoother path is possible through optimization-based algorithms.

3.3 Optimization-Based Path Planning

Planning a path for autonomous systems in dynamic environments is a crucial task yet a challenging one due to the limitations imposed by the system's dynamics and the presence of static or dynamic obstacles. To overcome this problem, trajectory optimization-based approaches have gained popularity nowadays, and optimal control problem is one of these approaches. In systems where disturbances are frequent, such as the sea, and where vehicle control is relatively difficult due to system dynamics, successful results can be obtained with this approach. The cost function of the problem to be optimized is determined by taking into account the factors to be considered (time, distance, or energy optimization) for the path planning of the vehicle, and adding these terms as components to the function. By doing so, the algorithm can produce solutions that are tailored to the desired situations. Additionally, this approach can generate paths that the system can follow by updating the generated path according to the current situation in environments where there are many dynamic obstacles. This allows the system to adapt to the current situation and generate feasible paths that can be followed in real-time. The cost function specified in equation 3.1 of OCP tries to solve for the optimal path. As seen here, constraints can be introduced into the optimization by adding terms such as environmental disturbances (τ_{dist}) and minimum energy consumption (σ_{energy}) to the function. Alternatively, other terms to be optimized can also be added to this function to achieve a solution that satisfies the desired criteria. The block shape of the path generation optimization problem is shown in figure 3.6.

$$\min \Sigma_{k=0}^{n-1}(|x|_2^{Q1} + |\Delta u|_2^R + |\tau_{length}|_2^{Q2} + |\sigma_{energy}|_2^{Q3} + |\sigma_{time}|_3^{Q3}) \tag{3.1}$$

Path planning using the optimization-based algorithm created based on the vehicle dynamic model obtained in Section 3 was performed for a scenario, and the results are presented in figure 3.7. The cost function of the algorithm created for path planning includes terms for the states of the system and the increment of the input signal.

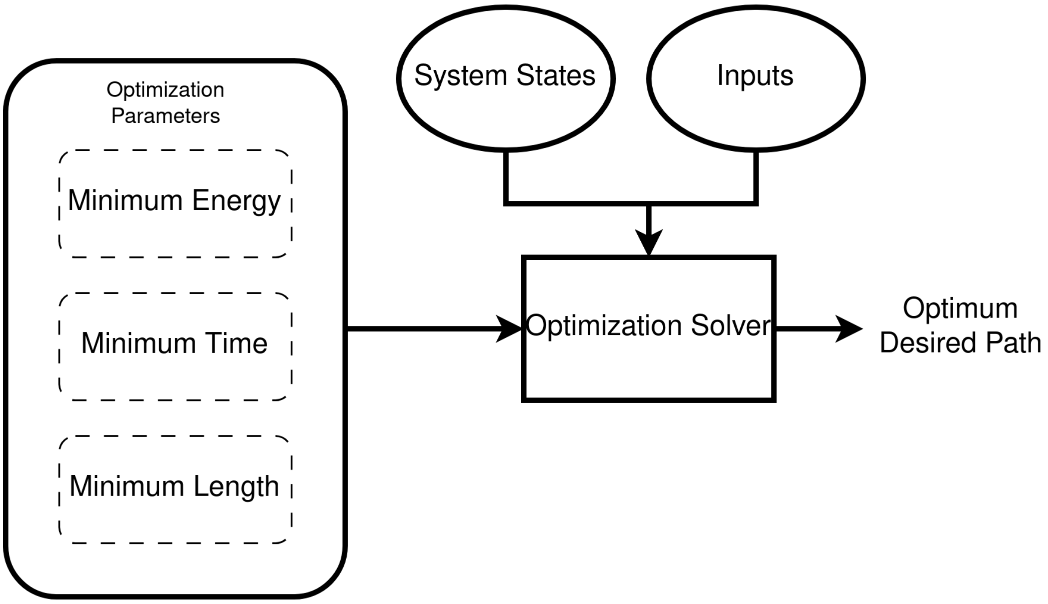


Figure 3.6 : Optimization problem for path generation

System constraints such as maximum surge speed and angular velocity have also been added to obtain a solution. By adding such constraints and optimization terms, general path planning can be performed online. When solving the algorithm, determining the prediction horizon is an important issue for optimizing the computational cost, which is a topic that can be researched in the future and will not be covered in the scope of this thesis.

In the optimization-based path planning method, due to considering vehicle dynamics, we have the ability to influence the curvature of the generated path by changing the limit value of vehicle's maximum cruising and yaw speed. For example, a path with a high curvature value can be created at low cruising speeds, while increasing the cruising speed can make this path flatter. Thanks to this approach, it is possible to determine the desired path structure in inland waterways and environments with many obstacles. Paths are planned using optimization-based path planner according to 1,2 and 3 m/s cruise speeds is shown in figure 3.7-9. As mentioned above, the path structure generated by the algorithm has varied according to the changing state limits, as maximum speed is increased, generated path structure has become more straight and direct.

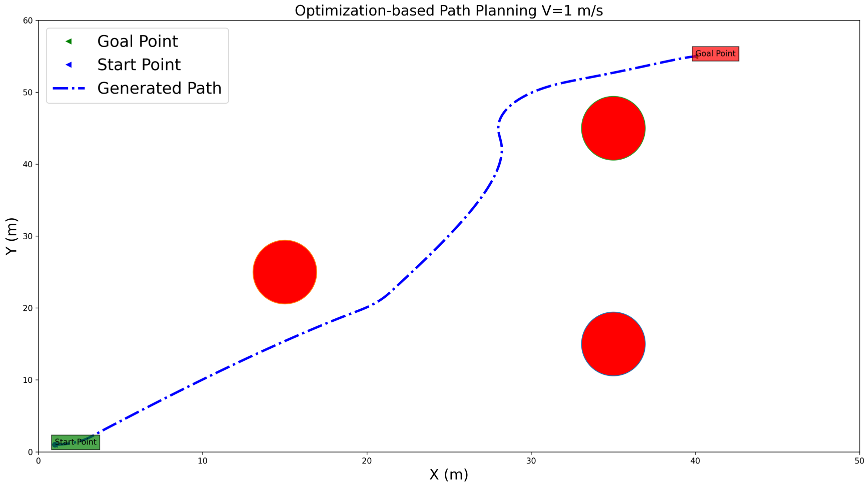


Figure 3.7 : Path planning created using optimization-based path planning for Otter USV at 1 m/s speed

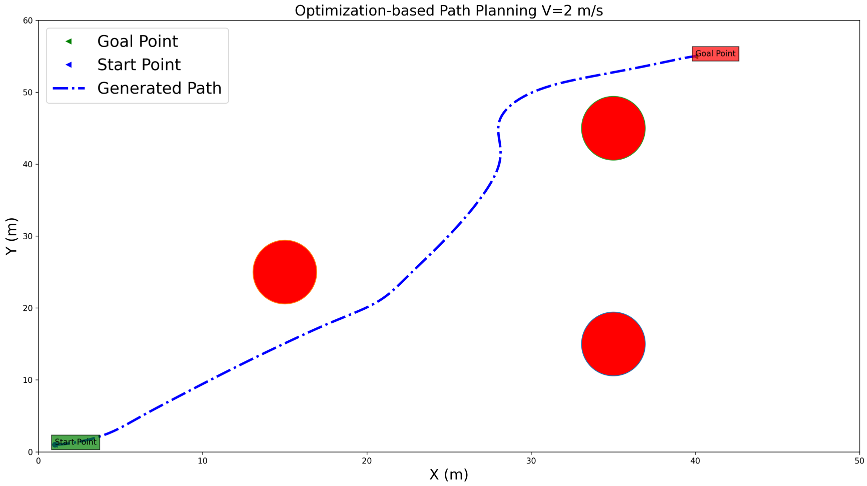


Figure 3.8 : Path planning created using optimization-based path planning for Otter USV at 2 m/s speed

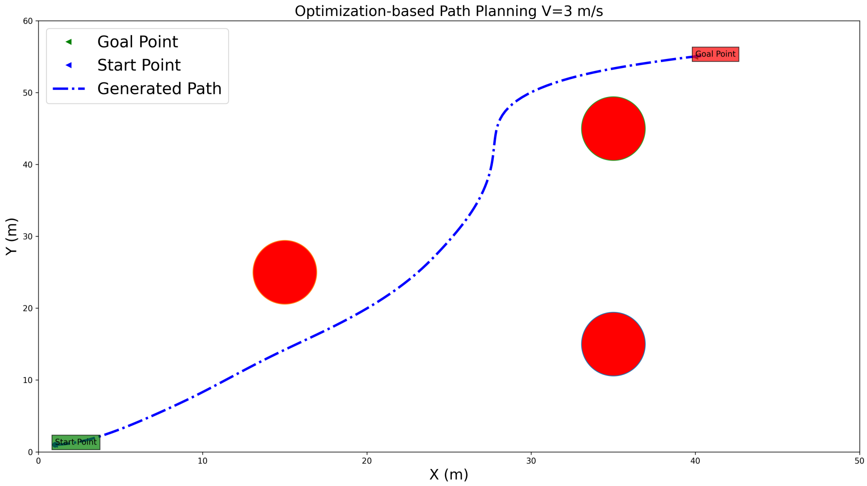


Figure 3.9 : Path planning created using optimization-based path planning for Otter USV at 3 m/s speed

When we look at the optimization problem of the path created in equation 3.1, when there are obstacles, to limit the solution space, a minimum approach distance (r) should be determined between the center of the obstacle and the vehicle, and this constraint should be added as an inequality constraint (equation 3.5) to the optimization problem at equations 3.1-7.

$$\min \Sigma_{k=0}^{n-1}(|x|_2^Q + |\Delta u|_2^R) \tag{3.2}$$

subject to

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{3.3}$$

$$\boldsymbol{x}_t \in \mathscr{X}, \boldsymbol{u}_t \in \mathscr{U} \tag{3.4}$$

$$g(\boldsymbol{x}_t, \boldsymbol{u}_t) - \boldsymbol{r} \leq 0 \tag{3.5}$$

$$h(\boldsymbol{x}_t, \boldsymbol{u}_t) = 0 \tag{3.6}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(t) \tag{3.7}$$

3.4 Optimization Tool CasADi

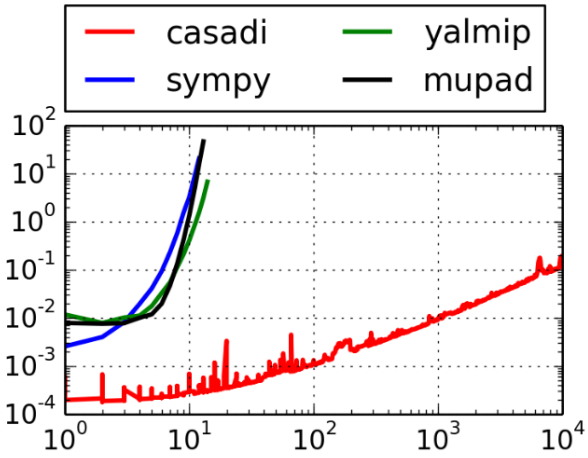


Figure 3.10 : Performance of optimization tools [23]

CasADi is an open-source software for numerical optimization, offering an alternative to conventional algebraic modeling languages such as AMPL, Pyomo, and JUMP. The main purpose of CasADi is to give a low-level interface to the user for quick and high efficient implementation of algorithms for nonlinear numerical optimization and formulate nonlinear programming problems (NLP) or optimal control problems (OCP) [23]. Performance of solvers according to states number is shown in figure 3.10.

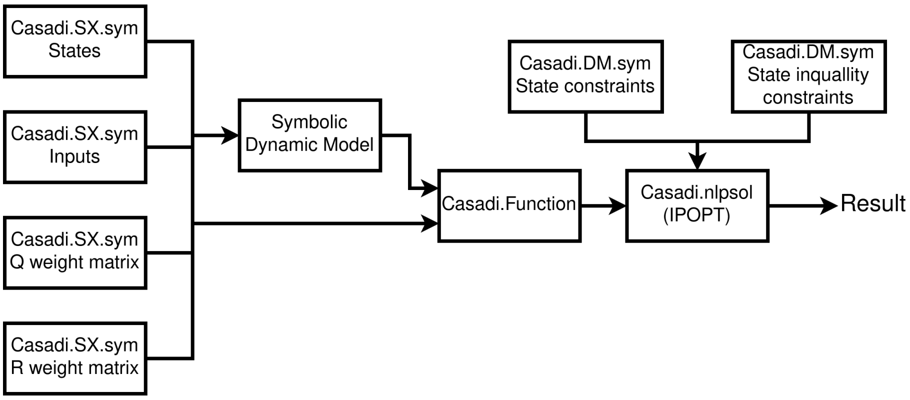


Figure 3.11 : MPC model in the CasADi interface

In this thesis, the Casadi program has been used for the optimization processes of the optimization-based path planning and MPC algorithms to be used within the scope of this thesis. General strucute of the optimizator is shown in figure 3.11.

CasADi is significantly quicker than other symbolic toolboxes that deal with symbolic expressions, such as sympy, yalmip, and mupad, even for a high number of variables, as demonstrated in figure 3.8. X axis show states number and Y axis represents computation time. The creation of an MPC model in the CasADi environment, as shown in figure 3.9, involves initially creating symbolic values using the SX structure for states, inputs, and their weight values. The SX data type is used to represent matrices whose elements are symbolic expressions constructed from a series of unary and binary operations [23]. Once the symbolic values are created, the dynamic model of the system is constructed using these values and function object is created for using in solver. In this study, the IPOPT solver has been used in CasADi for solving optimization problems. Finally, the equality and inequality constraints of the solver are provided in the DM (Dense Matrix) structure. DM is very similar to SX, but with the difference that the nonzero elements are numerical values and not symbolic expressions. DM is mainly used for storing matrices in CasADi and as inputs and outputs of functions [23]. CasADi solves OCPs by reformulating the OCP into NLP issues and using various approaches, such as those stated above, such as indirect (optimize then discretize) or direct (discretize then optimize) methods, such as single or multiple shooting. The user must develop his own OCP solver by accurately describing the constraints, parameters, and optimization variables; however, CasADi provides various high-level function blocks that make it simpler to define the problem in the syntax required by the NLP solver [23]. The block diagram depicting how the solution of an NLP (Nonlinear Programming) problem is achieved in the CasADi environment is shown in figure 3.12.

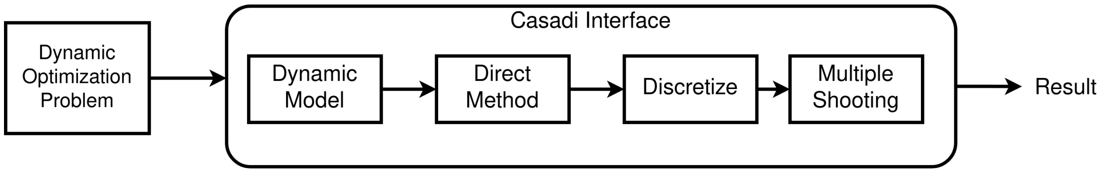


Figure 3.12 : OCP in the CasADi interface

4. NAVIGATION

4.1 Introduction

In mobile robot systems, these are the systems that continually compute the reference position, speed, and acceleration values required to reach the desired point. The system is the system that continuously computes the reference (desired) position, velocity, and acceleration of a vehicle. These data are usually provided to the human operator and the navigation system. In many cases, advanced optimization techniques are used to compute the optimal trajectory or path for the marine craft to follow [5]. Guidance system has several concepts, that are trajectory tracking, path following to guide a vehicle desired reference signals to reach determined final points, or following a predefined path. In the path following part, the advantages and disadvantages of the LOS path following algorithm used for tracking a generated path will be discussed. Furthermore, "The study on the variation of lookahead distance based on error for the improved performance of the LOS algorithm in narrow waterways, mentioned in Section 1, is described in this section.

4.2 Path Following

Path-following is one of the common control situations in the control literature, and it refers to following a preset path regardless of time, i.e., without placing limits on the temporal propagation along the path. [24]

Additionally, historically, path following scenarios have been quite common in maritime control literature. As stated before, the purpose of motion control is to converge on and follow a preset geometric path without any explicit time limitations linked with the path's propagation. [25] In marine robotic systems, there are some path following algorithms that are extensively used, such as pure pursuit, constant bearing, and line of sight.

As a LOS path following algorithm suitable for marine vessel motion behavior and having a successful performance in path following, it is one of the most preferred algorithms.

4.2.1 Path following objective

Control approach for a vehicle to go along a predetermined path that minimizes cross-track error (y_e) between the vehicle actual position (x,y) and the point that on a curve (x_θ, y_θ) is generated between two successive waypoints can be shown at equations 4.1-3. An illustration of a path following an objective is shown figure 4.1.

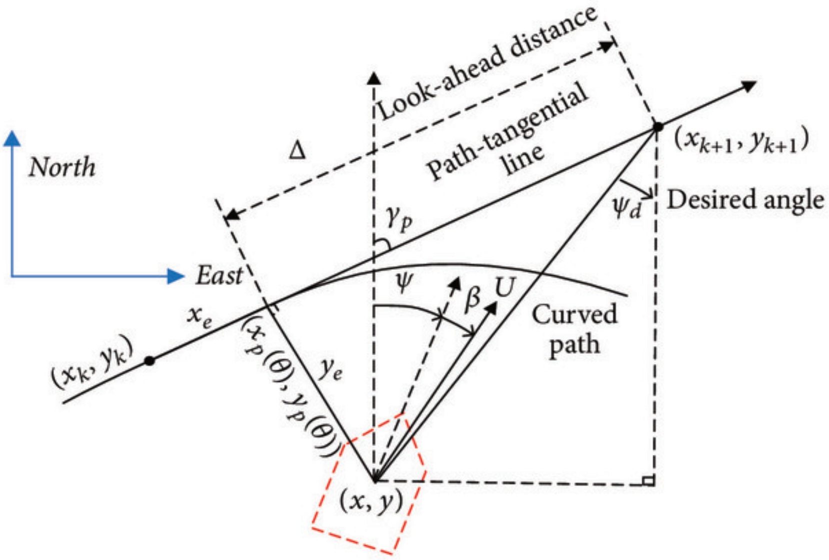


Figure 4.1 : Line of Sight path following illustration [25].

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^T(\gamma_p) \begin{bmatrix} x - x_p(\theta) \\ y - y_p(\theta) \end{bmatrix} \tag{4.1}$$

$$\mathbf{R} = (\gamma_p) \begin{bmatrix} \cos(\gamma_p) & -\sin(\gamma_p) \\ \sin(\gamma_p) & \cos(\gamma_p) \end{bmatrix} \tag{4.2}$$

$$\begin{aligned} x_e &= (x - x_p(\theta)\cos(\gamma_p) + (y - y_p(\theta))\sin(\gamma_p) \\ y_e &= -(x - x_p(\theta)\sin(\gamma_p) + (y - y_p(\theta))\cos(\gamma_p) \end{aligned} \tag{4.3}$$

where γ_p is the horizontal path-tangential angle:

$$\gamma_p = atan2(y_{y+1} - y_k, x_{x+1} - x_k) \tag{4.4}$$

In a path-following scenario, the along-track error x_e does not need to be minimized. Therefore, the control objective for curved paths;

$$\lim_{t \to \infty} y_e(t) = 0 \tag{4.5}$$

4.2.2 Cross-track error

Cross-track error is the minimum distance between vehicle current position and the closest point that is on path.

$$(x_p, y_p, \gamma_p) = argmin\sqrt{(x_i - x)^2 + (y_i - y)^2} \tag{4.6}$$

$P = [(x_1, y_1, \gamma_1), (x_2, y_2, \gamma_2), \cdots, (x_n, y_n, \gamma_n)]$ show the path that vehicle follows. the cross-track error can be shows

$$y_e = -(x - x(\theta) \sin(\gamma_p) + (y - y(\theta)) \cos(\gamma_p) \tag{4.7}$$

4.2.3 Line of sight (LOS)

The Line of Sight (LOS) path following algorithm is a navigation algorithm commonly used in marine applications to control the path of a vessel. The algorithm involves using sensors such as GPS, compasses, and inertial measurement units (IMUs) to determine the current position, heading, and velocity of the vessel, as well as the desired path to follow.

LOS attempts to approach the point by determining a point at a distance of Δh (where $\Delta h > 0$) from the point where the ship is projected on the path as the target point p at a linear distance from the vehicle. This allows the LOS vector to come closer to the point $(x_{los}; y_{los})$. The look-ahead distance is the design parameter that has the most significant impact on how well the vehicle is capable of following its path. When the Δ value is set low, it is possible to see overshoots when following the path; conversely, if the Δ value is kept high, it will settle more smoothly on the path, however takes a relatively long time to settle on the path. LOS that has a linear look-ahead distance brings some fault in complex path. such path has small radius value. Keeping the LOS point on the curve at all times also helps to solve this problem. Determining the curve length between a certain look-ahead value, an LOS point is assigned on the curve, which stays on the curve for the distance determined on the curve from the current position of the vehicle. This improves the performance of path following in places where sudden turns are required.

4.2.4 Advantages of LOS

The LOS algorithm is particularly useful in open sea environments where there are no fixed objects or obstacles to avoid. It is also commonly used in situations where the desired path is constantly changing, such as when navigating through channels or in inlet waterways. The algorithm can be combined with other navigation algorithms, such as path planning algorithms, to create more sophisticated navigation systems for marine applications. This algorithm enables more accurate and precise movement control for maritime vessels, particularly in tough environments.

- Improved accuracy:The LOS algorithm allows the vessel to follow a precise path, reducing the risk of deviation from the intended course. This is particularly important in narrow waterways, crowded harbors, and other challenging environments,Better control: The direct line of sight between the vessel and the target point allows for better control of the vessel's speed and heading, enabling the vessel to respond quickly to changes in the environment and avoid potential hazards.

- Increased safety: The LOS algorithm provides greater situational awareness for the vessel's crew, allowing them to identify potential hazards and respond quickly to avoid collisions or other dangerous situations. Additionally, for an autonomous marine vessel are similar to those for a manned vessel, but there are some additional benefits specific to autonomous operation. The LOS algorithm allows the autonomous vessel to navigate more independently, reducing the need for human intervention and increasing the vessel's autonomy.

- Reduced risk of collisions: The LOS algorithm provides the autonomous vessel with a direct line of sight to the target point, allowing it to detect and avoid potential hazards more effectively, which reduces the risk of collisions. Improved efficiency: The LOS algorithm can optimize the autonomous vessel's path to reduce fuel consumption, travel time, and other operating costs.

4.2.5 Disadvantages of LOS

The line of sight (LOS) path following approach provides a number of advantages, but there are also possible disadvantages to consider.

- Susceptible to disturbances: The LOS approach can be subject to disturbances such as wind, waves, and currents, which might deflect the vessel off its intended course. This can be especially difficult during stormy weather, when the vessel may struggle to maintain a constant LOS.

- Challenging in complex environments: In complicated maritime environments where various obstructions, currents, and other variables might impact navigation, the LOS approach may be less successful. In such circumstances, alternative path-following techniques, such as model predictive control or artificial potential fields, may be more appropriate.

- Suddenly chancing direction: The LOS path following algorithm may fail in conditions when the vessel is needed to quickly change course, such as in restricted areas. This is because the algorithm relies on keeping a direct line of sight between the vessel and the target point, and rapid changes in direction might cause the target point to drift out of view or it can be assigned to an unsuitable point for the path to be followed. This can lead to navigational mistakes and cause the ship to wander from its intended course.

To avoid this situation, in this thesis, the classical LOS algorithm was slightly modified by adjusting the assigned LOS point to always remain on the path curve. With this approach, the trackability of the path has been ensured at curve points with small turning radiuses.

4.2.6 Mathematical approach

The mathematical approach of the Line of Sight (LOS) path following algorithm in marine applications involves calculating the LOS angle between the vessel's current heading and the desired path and then using this angle to adjust the vessel's rudder to steer towards the desired path. The flow chart of the algorithm is shown in figure 4.2.

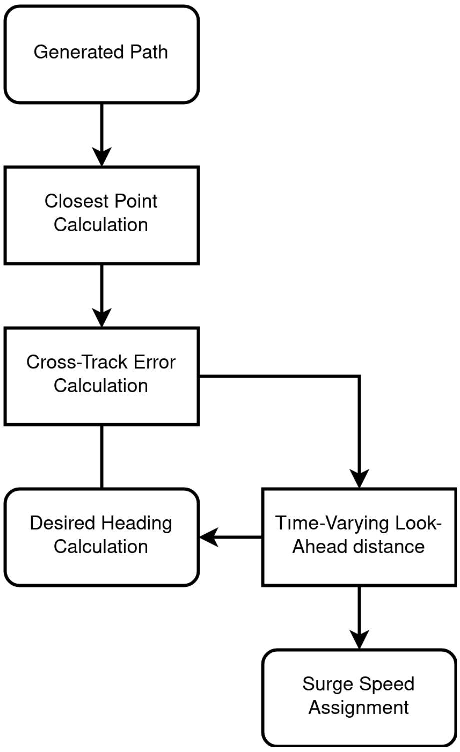

Generated Path
Closest Point Calculation
Cross-Track Error Calculation
Desired Heading Calculation
Tıme-Varying Look-Ahead distance
Surge Speed Assignment

Figure 4.2 : Flow chart of Line of Sight algorithm

The LOS guidance law;

$$\chi_d = \gamma_d + \arctan(\frac{-y_e}{\Delta}) \tag{4.8}$$

χ_d is desired course angle of vessel that follows desired path. In the presence of external disturbances, or during turns, the heading angle ψ_d and the course angle χ_d are not aligned anymore and are related in the following way: [5]

$$\chi_d = \psi_d + \beta \tag{4.9}$$

(4.9) and (4.10) equations are used, desired heading angle is

$$\chi_d = \gamma_p + \arctan(\frac{-y_e}{\Delta}) - \beta \tag{4.10}$$

β can be calculated by $atan2(v,u)$

Time-Varying Look-Ahead Distance

For increasing path following performance, look-ahead distance can be parameterazied according to cross-track error.

$$\Delta(y_e) = (\Delta_{max} - \Delta_{min}) \exp^{-K_\Delta y_e^2} + \Delta_{min} \tag{4.11}$$

$\Delta_{max}, \Delta_{min}$ are maximum and minimum allowed look-ahead values, K_Δ is a coefficient for the covergence ratio. As the vehicle is moving in the opposite direction of the path that is wanted, a relatively low value for Δ is used, and when the value of y_e goes down, a relatively high value for Δ is used for prevent overshooting. This principle is straightforward and easy to understand. It can be deduced from this equation that Δ_{min}, Δ_{max} and K_Δ are important parameters that affect the path following performance of vehicle.

Speed assignment Reference speed assignment can be formulated as follows

$$U_d = max(U_{max}(1 - \frac{|y_e|}{y_{max}} - \frac{|\chi_c|}{\chi_{max}} + U_{min}, U_{max})) \tag{4.12}$$

U_d is desired speed, U_{max} and U_{min} are maximum and minimum allowed speeds limits χ_c is error in course angle

This speed assignment makes it easier for the USV to return to the intended path by reducing the speed when the vehicle deviates from its path, reducing the turning radius at lower speed. When the speed reaches its lowest, which is denoted by the equation $U = U_{min}$, the USV will attain its minimal turning radius ρ_{min}, which confirms that any produced path may be followed [27].

4.2.7 Adaptive LOS for Inland Water

The regular LOS approach provides successful results in environments where the radius of the curve is high and maneuvers are performed in wide areas. However, in environments where maneuvering in narrow spaces is required, it becomes challenging to stay on the desired path when following path using LOS algorithm. In order to enhance the path following performance in such environments, a different approach was introduced for calculating the lookahead distance that is dependent on error. When the vehicle is on the path, the lookahead distance is kept lower, thereby increasing the ability to track the path in areas where the radius of the curve is low.

$$\Delta(y_e) = K.y_e + \Delta_{min} \tag{4.13}$$

With this method developed, tests were conducted on two scenarios that could be encountered in narrow waterways to compare the performance of classic LOS. In the first scenario that is shown figure 4.3 and control signals are in figure 4.4, circular path following in a narrow area was requested, while in the second scenario that is shown in figure 4.5 and control signals are in figure 4.6, following a path with zigzag movements was required. The algorithm developed for inland waterways has shown better performance in both scenarios.

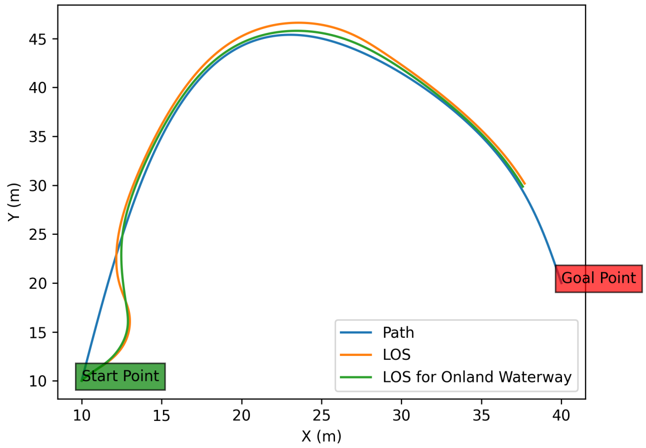


Figure 4.3 : LOS algorithms compasion

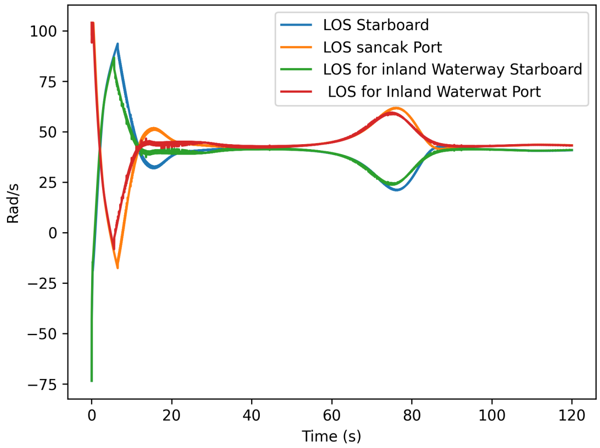


Figure 4.4 : Following of circular path input signals

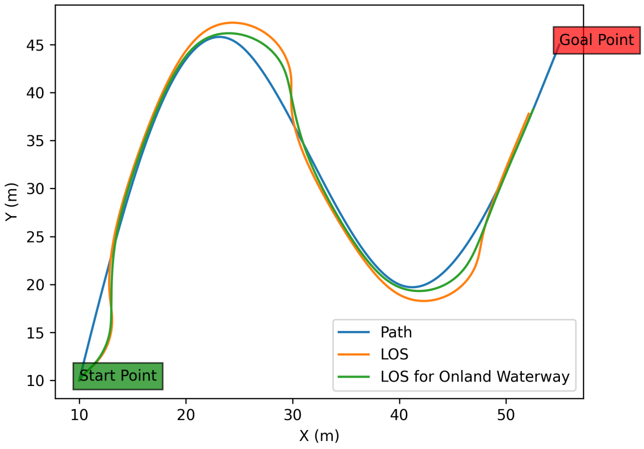


Figure 4.5 : LOS algorithms compasion

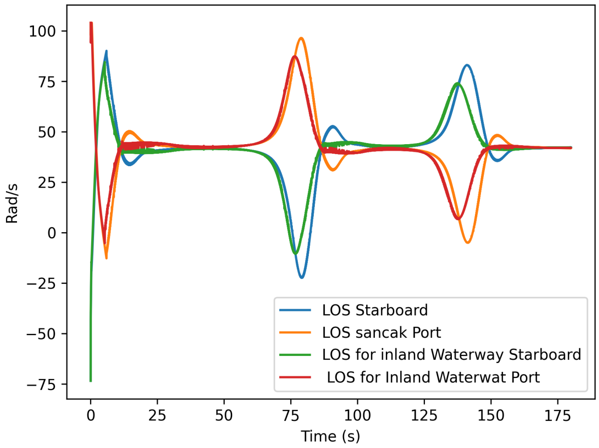


Figure 4.6 : Following of zigzag path input signals

5. MODEL PREDICTIVE CONTROL

5.1 Introduction

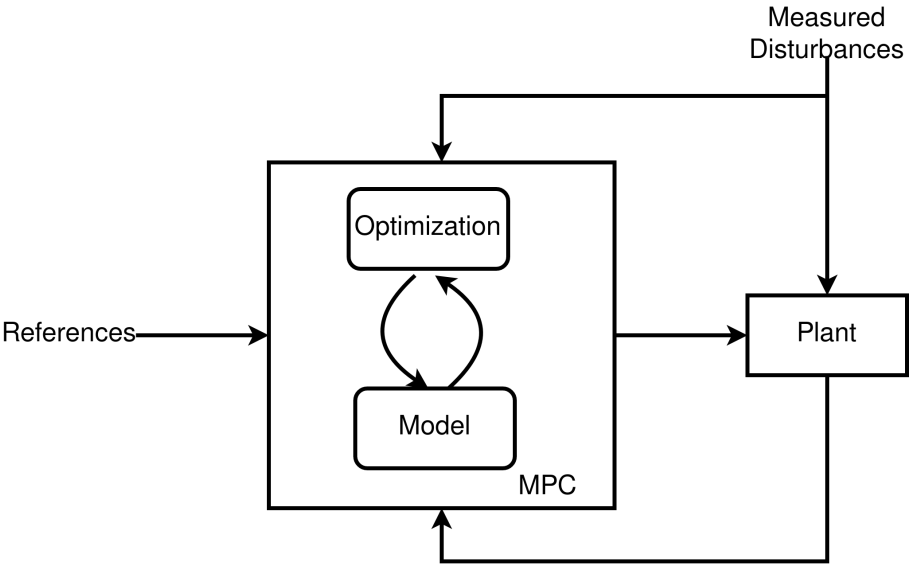


Figure 5.1 : MPC block diagram

Model Predictive Control (MPC) is a type of advanced control strategy that uses a model of the system being controlled to predict future behavior and calculate an optimal control action. The general structure of MPC is shown figure 5.1. The control action is chosen to minimize a cost function while satisfying constraints on the system's input and state variables. MPC has gained popularity in recent years due to its ability to handle complex systems with constraints and its ability to handle disturbances and uncertainties.

One of the main advantages of MPC is its ability to handle systems with constraints. The use of a predictive model allows the controller to anticipate the future behavior of the system and avoid violating constraints. For example, in a chemical process, MPC can be used to regulate temperature, pressure, and flow rate while ensuring that constraints on reaction rate, product purity, and equipment limits are not violated [28].

When the controlled system in MPC is nonlinear, the system is attempted to be controlled using the nonlinear model predictive control method. The general control algorithm of MPC remains exactly the same, but the system model used in the optimization part is nonlinear. In this study, the system to be used is nonlinear, and the nonlinear MPC model is described in Section 5.2.

MPC is a form of advanced control technique that calculates the best control action based on a predictive model of the system being controlled. Depending on the application, the predictive model is used to anticipate the future behavior of the system over a specified time horizon, often a few seconds to a few minutes.

The MPC controller computes an optimum control action that minimizes a cost function while meeting limitations on the system's input and state variables, based on the expected future behavior. Typically, the cost function is a weighted sum of the deviations between the projected future behavior and the intended future behavior, as well as the violations of restrictions.

Using numerical optimization methods such as quadratic programming (QP), the optimal control action is derived. The optimization issue includes minimization of the cost function subject to input and state variable restrictions. The restrictions guarantee that the control action does not exceed physical limits or operational needs [29].

The optimal control action is executed on the controlled system, and the procedure is repeated at each time step in order to continually update the optimal control action based on the system's current state.

Adventages: MPC is a predictive control method that anticipates the future conditions of the vehicle and plans its control actions accordingly, as opposed to reactive controllers. The MPC algorithm can deal with non-linear and complex vehicle dynamics, such as tire force models and actuator models, enabling for precise and accurate trajectory tracking. Another effective aspect of MPC is that it can accommodate multiple constraints. Even fuel efficiency and energy consumption can be factored into MPC optimization, resulting in greater efficiency than other controllers.

Disadventages: Yes, there are some disadvantages to utilizing MPC. Computational complexity is one of the most serious drawbacks. MPC necessitates extensive computation to generate optimal control actions. This can make real-time implementation in embedded systems more difficult. In order to apply the MPC path following model developed within the scope of this thesis to embedded systems, the system model used to reduce the transaction cost has been tried to be simplified. MPC is dependent on precise models for system control. This can be challenging if you do not understand the underlying system or if it is difficult to model.

5.2 Mathematical Approach of NMPC

Because this thesis focuses on the study of a nonlinear model (the otter vehicle model), the NMPC method was selected as the method of choice for controlling the model. Publications by [30], [31], and [32] are excellent sources for the theory behind NMPC, the framing of problems, and prospective applications of this theory. Nonlinear Model Predictive Control (NMPC) is a control strategy that uses a mathematical model of the system to predict future behavior and optimize control actions. Ilustration of the algoritm is shown in figure 5.2. MPC consists of several fundamental components, which are shown in equations 5.1-6.

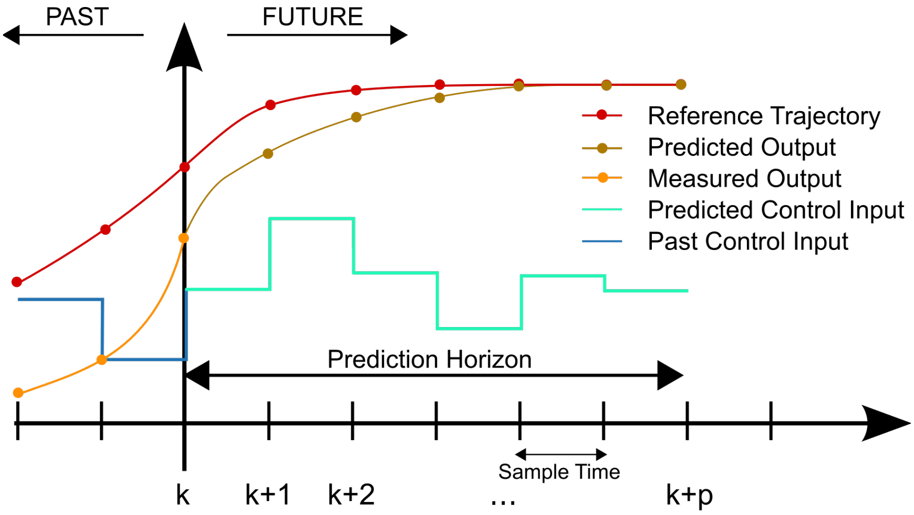


Figure 5.2 : Model Predictive Controller time horizon [32]

NMPC uses some fundamental terms when creating mathematical models. and they are shown in the basic mathematical formulation of the NMPC algorithm.

$$\min_{\boldsymbol{u}} J(\boldsymbol{x}, \boldsymbol{u}) \tag{5.1}$$

subject to

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{5.2}$$

$$\boldsymbol{x}_t \in \mathscr{X}, \boldsymbol{u}_t \in \mathscr{U} \tag{5.3}$$

$$g(\boldsymbol{x}_t, \boldsymbol{u}_t) \leq 0 \tag{5.4}$$

$$h(\boldsymbol{x}_t, \boldsymbol{u}_t) = 0 \tag{5.5}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(t) \tag{5.6}$$

where $\boldsymbol{x}_k$ is the **states** of the system at time k, $\boldsymbol{u}_k$ is **the control input** applied at time t, $J(\boldsymbol{x}, \boldsymbol{u})$ is **the cost function**, $f(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is the nonlinear system model, $g(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and $h(\boldsymbol{x}_k, \boldsymbol{u}_k)$ are **the inequality and equality constraints**, respectively. The initial state $\boldsymbol{x}_0$ is given by the current state of the system.

The optimization problem seeks to minimize the cost function over a finite **time horizon** $k+p$, subject to the system dynamics and the constraints on the states and inputs. The solution of the optimization problem provides a sequence of optimal control inputs $\boldsymbol{u}_k$ that should be applied to the system over the next time steps. However, only the first control input $\boldsymbol{u}_0$ is actually applied to the system, and the optimization problem is solved again at the next time step to generate a new sequence of optimal control inputs based on the updated state of the system.

States: The parameters that characterize the behavior of a systems at a given time.
Control input: The control parameters that are determined through MPC computation.
System model: The system's motion is mathematically modeled by either a kinematics or dynamics model.
Constraints: The maximum and minimum values for the states and control inputs.
Cost Function: The mathematical expression representing the control system's

objective. There are various optimization models that can be utilized in this area. In this study, Quadratic programming has been employed.

Prediction Horizon: The number of future samples over which MPC predicts the outputs and control inputs.

5.2.1 Quadratic programming

Quadratic programming (QP) is a widely used optimization technique in nonlinear model predictive control (NMPC) to solve the optimization problem. In NMPC, the objective is to find an optimal control input sequence that minimizes a cost function subject to system constraints over a finite prediction horizon while considering the nonlinearity of the system dynamics.

The general form of the NMPC optimization problem can be expressed as a QP problem, where the Hessian matrix and the gradient vector are approximated at each time step and the constraints are transformed into a linear form. The QP problem can be formulated as follows;

$$\underset{x,u}{\text{minimize}} \quad J(x,u) = \frac{1}{2}|\mathbf{y} - \mathbf{y}^r|Qy^2 + \frac{1}{2}|\mathbf{u} - \mathbf{u}^r|Ru^2 \tag{5.7}$$

$$\text{subject to} \quad \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_{\mathbf{k}} + \mathbf{B}\mathbf{u}_k \qquad \mathbf{x}_k \in \mathscr{X},;\mathbf{u}_k \in \mathscr{U} \tag{5.8}$$

where $\mathbf{x}_k$ and $\mathbf{u}_k$ are the state and control input vectors at time step k, $\mathbf{y}$ is the output vector, $\mathbf{y}^r$ and $\mathbf{u}^r$ are the reference trajectories, Q_y and R_u are positive semi-definite weighting matrices, $\mathbf{f}$ is the system dynamics function, and $\mathscr{X}$ and $\mathscr{U}$ are the state and control input constraint sets.

The objective function $J(x,u)$ represents the sum of the quadratic cost functions that penalize deviations from the reference trajectories. The constraints in the QP problem ensure that the system dynamics are satisfied and that the state and control input vectors are within their respective constraint sets. [34]

QP solvers can be used to solve the QP problem iteratively. The solution provides the optimal control input sequence $\mathbf{u}_{(k:k+N-1)}$, where N is the prediction horizon. The first element of the optimal control input sequence, $\mathbf{u}_{(k)}$, is applied to the system, and the optimization problem is solved again at the next time step.

In conclusion, QP is a powerful optimization technique used extensively in NMPC to solve the optimization problem by formulating it as a QP problem, where the Hessian matrix and the gradient vector are approximated, and the constraints are transformed into a linear form. QP-based NMPC has been successfully applied to various applications, demonstrating its effectiveness in controlling nonlinear systems while satisfying constraints.

5.3 Optimal Control Problem:

When it comes to finding optimal solutions to control problems, there are three different approaches, dynamic programming, indirect and direct approaches

Dynamic programming: Dynamic programming uses the principle of optimality of subarcs to compute recursively a feedback control for all times t and x_0 , leading to the Hamilton-Jacobi-Bellman (HJB) equation. Methods to numerically compute solution approximations exist, but are limited to small state dimensions.

Indirect method: Indirect approaches derive a boundary value problem (BVP) from ordinary differential equations (ODE) using the optimality requirements of an infinite problem. The BVP is often solved numerically using shooting techniques or collocation. The two significant drawbacks are that the underlying differential equations are frequently difficult to solve because of their severe nonlinearity and instability, and that the control structure is susceptible to modification.

Direct method Direct techniques convert the original infinite optimal control problem into a finite-dimensional nonlinear programming issue (NLP). This NLP is then solved by variations of state-of-the-art numerical optimization methods; hence, "first discretize, then optimize" is a common approach schematic. One of the most significant advantages of direct methods over indirect ones is their ability to readily handle inequality constraints, such as the inequality path limitations in the preceding formulation. [35] Direct methods consist of sequential and simultaneous approaches.

When sequential one use single shooting for solving optimal contol problem (OCP), simultaneous approaches use multiple shooting.

5.3.1 Direct optimal control

Finite dimensional nonlinear programming problem (NLP) at below;

$$\min_{\boldsymbol{u}} J(\boldsymbol{u}) \tag{5.9}$$

subject to

$$g(\boldsymbol{u}) \leq 0 \tag{5.10}$$

$$h(\boldsymbol{u}) = 0 \tag{5.11}$$

While $\boldsymbol{u}$ show optimization degrees of freedom, $\boldsymbol{J}$ differentiable function and $\boldsymbol{g}$,$\boldsymbol{h}$ are vector. Direct methods for solving optimal control problems typically involve parameterizing the control trajectory, but they vary in how they handle the state trajectory, with some being classified as sequential and others as simultaneous that mentioned in the previous section.

5.3.1.1 Single shooting

The single-shoot approach that is shown in figure 5.3 is an iterative method in which the optimization problem is handled by repeatedly revising the control settings until a satisfying solution is reached. During each iteration, the simulation procedure is utilized to solve the system dynamics, and the resulting trajectory is utilized to compute the cost function. An optimization procedure that minimizes the cost function is then used to update the control parameters.

Initial guess $u^{(0)}$, time horizon T, convergence tolerance ε, Optimal control u^*, state variables x. Discretize the time horizon $[0,T]$ into N sub-intervals of equal length $\Delta t = T/N$. Introduce a set of state variables x_i for each sub-interval i.Define the system dynamics $f(t,x,u)$ that describes the evolution of the state variables over time. Define the cost functional $J(u,x)$ as a function of the control variables and the state variables. Set up the optimization problem as follows:

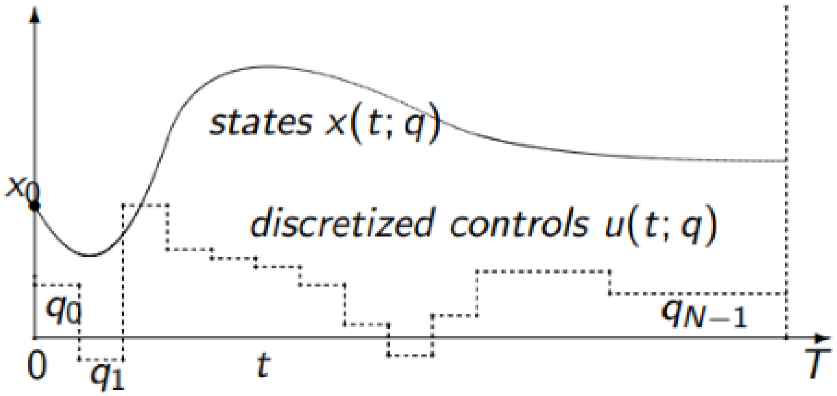


Figure 5.3 : Single shoot method [35]

$$minimize \quad J(u,x) \tag{5.12}$$

$$x_{i+1} = f(t_{i+1,x_i,u_i}) \tag{5.13}$$

$$for \quad i = 0,1,....,N-1 \tag{5.14}$$

$$x_0 = x_{init} \quad x_N = x_{final} \tag{5.15}$$

where x_{init} and x_{final} are the initial and final values of the state variables.

Solve the optimization problem using an optimization algorithm, such as a nonlinear programming solver.

Obtain the optimal control variables and state variables.

If the convergence criteria are not satisfied, update the initial guess $u^{(0)}$ and repeat steps 1-7 until convergence is achieved. [36]

5.3.1.2 Multiple shooting

The fundamental concept underlying the multiple shooting approach that is ilustrated in figure 5.4 is to treat the state variables at the endpoints of each subinterval as optimization variables. Hence, the state variables can be individually updated at each sub-interval, yielding a set of coupled nonlinear equations. Using a nonlinear programming solver, the optimization problem can then be resolved.

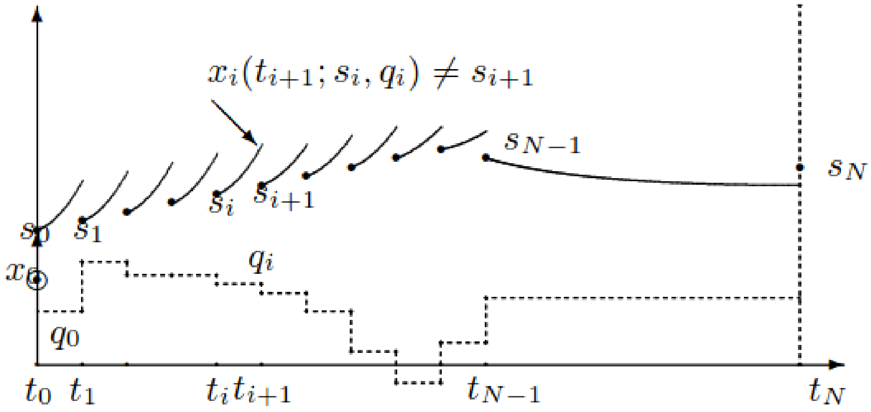


Figure 5.4 : Multiple shoot method [35]

we again discretize the controls piecewise on a coarse grid

$$\boldsymbol{u(t)} = \boldsymbol{q_i} \quad \boldsymbol{t} \in [\boldsymbol{t_i}, \boldsymbol{t}_{i+1}] \tag{5.16}$$

where the distance between shots might be as great as in single firing. Second, we solve the ODE on each interval $[t_i, t_{i+1}]$ separately, beginning with an arbitrary initial value $\boldsymbol{s_i}$.

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{f}(\boldsymbol{x_i}, \boldsymbol{q_i})) \in [\boldsymbol{t_i}, \boldsymbol{t}_{i+1}] \quad \boldsymbol{x_i(t_i)} = \boldsymbol{s_i} \tag{5.17}$$

By numerically solving these initial value issues, we obtain trajectory pieces $x_i(t; s_i, q_i)$, where the additional parameters after the semicolon identify the interval's beginning values and controls. In addition to the decoupled ODE solution, the integrals are also numerically computed.

$$\boldsymbol{l_i}(\boldsymbol{s_i}, \boldsymbol{q_i}) := \int_{t_i}^{t_{i+1}} \boldsymbol{L}(\boldsymbol{x_i}(\boldsymbol{t_i}; \boldsymbol{s_i}, \boldsymbol{q_i}), \boldsymbol{q_i})\boldsymbol{dt} \tag{5.18}$$

To confine the artificial degrees of freedom si to physically relevant values, continuity requirements $s_{i+1} = x_i(t_{i+1}; s_i, q_i)$ are imposed. Hence, we arrive at the following NLP formulation, which is entirely equal to the single shot NLP, but includes the additional variable si and has a block sparse structure [35].

$$\underset{s,q}{\text{minimize}} \quad J(x,u) = \sum_{i=0}^{N-1} \boldsymbol{l_i}(\boldsymbol{s_i},\boldsymbol{q_i}) + E(\boldsymbol{s}_N) \tag{5.19}$$

subject to

$$s_0 - x_o = 0 \tag{5.20}$$

$$s_{i+1} - x - i(t_{i+1}; s_i, q_i) = 0 \tag{5.21}$$

$$h(s_i, q_i) \geq \tag{5.22}$$

$$r(s_N) = 0 \tag{5.23}$$

6. CONTROLLER

6.1 Introduction

In this chapter, the development of the design of the MPC and PID controllers required for following a path created with the path planning algorithm will be explained. The states required for path control and the block diagram to be created for their control will be explained. In order to determine the optimum values of the weight parameters that affect the MPC algorithm, various trials have been conducted in different scenarios to calculate the state and input weight values. Lastly, the same process has been carried out to determine the PID coefficients. Various trials have been conducted to calculate the optimal values of the PID coefficients by evaluating the system's response in different scenarios. These trials help in tuning the PID controller to achieve the desired control performance.

6.2 Path Following Objective

To achieve proper path following, it is vital to ensure that the boat's control continually performs well and can follow simple references. This can be accomplished by modifying the course and speed controls to execute benchmark operations, like as turning and speed, in order to get a boat that is accurate and responsive. The following of the path using the MPC controller algorithm we described in in figure 6.1. In order to successfully following the specified path, the states that we need to control are x,y,heading,surge and yaw angle rate. By using the MPC to track the reference values that generated from desired path, successful path following performance is demonstrated.

The controller optimally controls the system according to the changing targets, allowing the vehicle to move on the determined path. Figure 6.1 represents the general structure of the GNC system necessary for path following.

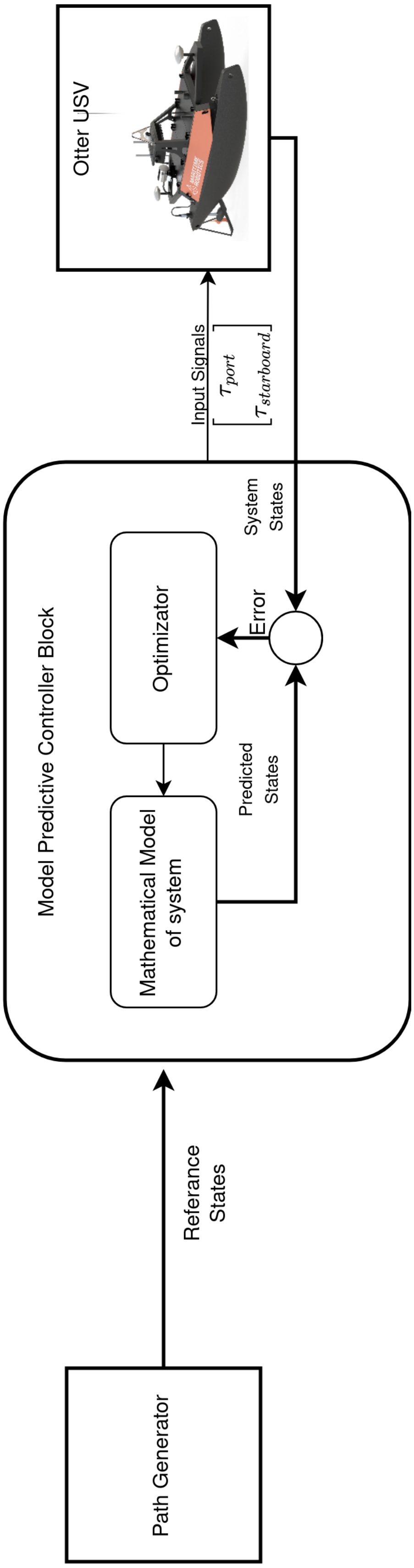

Path Generator
Referance States
Model Predictive Controller Block
Mathematical Model of system
Optimizator
Predicted States
Error
System States
Input Signals
$\begin{bmatrix} \tau_{port} \\ \tau_{starboard} \end{bmatrix}$
Otter USV

Figure 6.1 : MPC path following system

The path to be followed is generated in the Path generator block (section 3) based on the given goal point that provided to the system. To follow this created path, the values of the states that will be entered as references to the controller block in the path following block (section 4) are determined. In the controller block, calculations are made until the time horizon determined with the help of the system model and the optimizer according to the references given. Differences between the real system and the system model are feedback to the optimizer as an error.

6.2.1 MPC controller design

MPC consist of several sub parts. These are cost function, system model, opimization algorithm and constraints. These topics have been discussed in ongoing subsections in this part.

6.2.1.1 Cost function

For path following, the MPC must stabilize system states relative to the reference value, and has to follow with minimum error in different environmental conditions. About the construction of the cost function for the optimum control input issue, the safety and comfort factors of driving are taken into account. The cost term that minimizes the error in following the reference trajectory, which is constrained by physical and environmental restrictions, is intended to maximize safety. The second goal is utilized to provide feedback output information and incremental output information to the model in order to improve controller performance [37]. The cost term of MPC can be represent at equation 6.1. The first term of the penalize the states term using Q state weight matrix, second term adjust input signal with R input weight matrix.

$$\min \Sigma_{k=0}^{n-1}(|\zeta_{t+k,t} - \zeta_{ref_{t+k,t}}|_2^Q + |\Delta u|_2^R) \tag{6.1}$$

$$\zeta_{t+k,t} = (x_{t+k}, y_{t+k}, \psi_{t+k}, u_{t+k}, r_{t+k}) \tag{6.2}$$

$$\zeta_{ref_{t+k,t}=(x_{ref},y_{ref},\psi_{ref},u_{ref},r_{ref})}(6.3)$$

$$\Delta u = \Delta\tau_{port}, \Delta\tau_{starboard} \tag{6.4}$$

6.2.1.2 Constraints

In order for the solution of the cost function to be applicable by the controlled system, we need to impose some constraints on the solution function. These constraints are dynamic system constraints, the maximum and minimum state values that the vehicle can reach, along with the possible input signal values that can be provided. The limits of the system we are controlling are maximum 3 m/s and minimum -1 m/s surge velocity for the state u, and -0.35-0.35 rad/s for the angular velocity r. The range of the input signal is between -95 N and 120 N.

$$\tau_{port_{min}} \leq \tau_{port} \leq \tau_{port_{max}}) \tag{6.5}$$

$$\tau_{starboard_{min}} \leq \tau_{starboard} \leq \tau_{starboard_{max}}) \tag{6.6}$$

$$u_{min} \leq u \leq u_{max} \tag{6.7}$$

$$r_{min} \leq r \leq r_{max} \tag{6.8}$$

The optimization control problem, which was created from the cost function and constraint, has been solved using the IPOPT solver.

6.2.1.3 Disturbance rejection

In the MPC prediction phase, predictions are made up to the prediction horizon by utilizing the state and input values of the system model to forecast the future state of the system. If environmental forces such as wind, waves, currents, etc. that affect the system are incorrectly or incompletely defined in the mathematical model of the system, errors occur in the solution when these disturbances affect the system. Because differences occur between the values calculated by the system and the real system due to incorrect calculation of the disruptive effects, the difference between the actual values and the predicted values increases as the prediction horizon increases. In this case, it directly affects the solution performance of the MPC. To prevent this, the difference between the predicted states of the MPC and the actual states of the system is fed back to the system as feedback at the end of each calculation cycle. The visualization of this is shown under the MPC block in figure 6.1.

6.2.1.4 Optimization parameters

MPC has many parameters that provide flexibility in controlling a system and optimize control performance. These are prediction horizon, weight value of states, rate of change weight of input and weight of input. By adjusting these parameters, the desired solution is attempted to be optimized.

Prediction Horizon Parameters: The prediction horizon sets the range of system evolution during which the control improvements are implemented. It is an essential MPC tuning parameter. In path following scenarios, creating a long prediction to go on the desired path increases the controller performance, however it is not always true. As you increase the prediction horizon, optimization performance may reduces or can not solved properly. Therefor adjusting of optimizing this length is important. Because there is a trade off between control performance and computation cost. In this study, the control horizon was determined as 20 for this study. The performance comparison according to the changing control horizon is shown in the figure 6.2 and control signals are shown in figure 6.3 and control signals behavior of it is shown figure 6.3. To follow a predetermined path, it is sufficient to control the positions and surge speed state. When we examine the behavior of these states based on the varying horizon values; firstly, when we look at the speed behavior for the prediction horizons 10, 20 and 30; the prediction horizon 10 causes an overshoot, while the 20 and 30 show almost the same behavior. On examining the rotation values of the propellers, the input signals for the 10 prediction horizon showed more aggressive behavior, while the input signals for the 20 and 30 prediction horizons were smooth.

The behavior of the heading state according to the changing prediction horizon values is shown in the figure 6.4 and control signals is shown figure 6.5. While the control performance shows almost the same behavior, the input signals for the prediction horizon 10 are more oscillating.

Table 6.1 : MPC calculation time.

Prediction Horizon	Calculation Time (s)
10	0.008
20	0.011
30	0.016

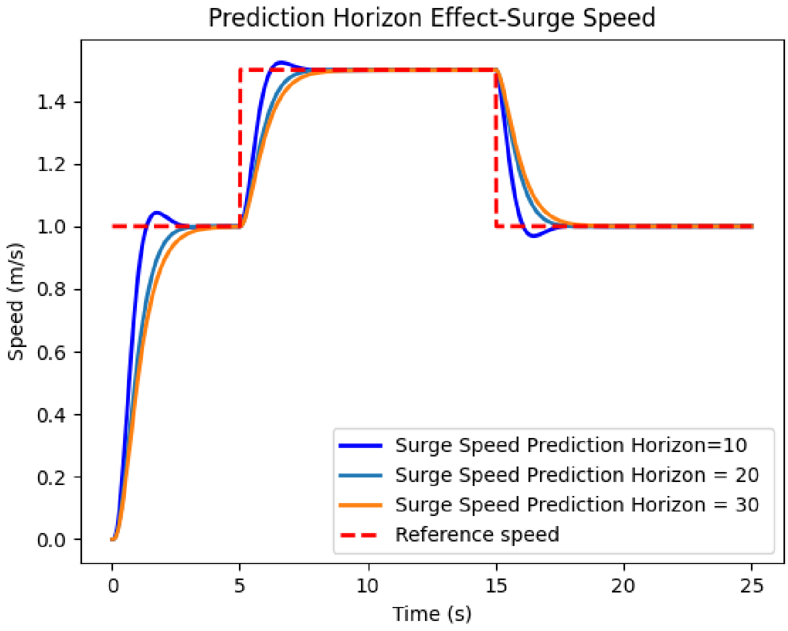


Figure 6.2 : Surge speed state behavior according to changing prediction horizon values

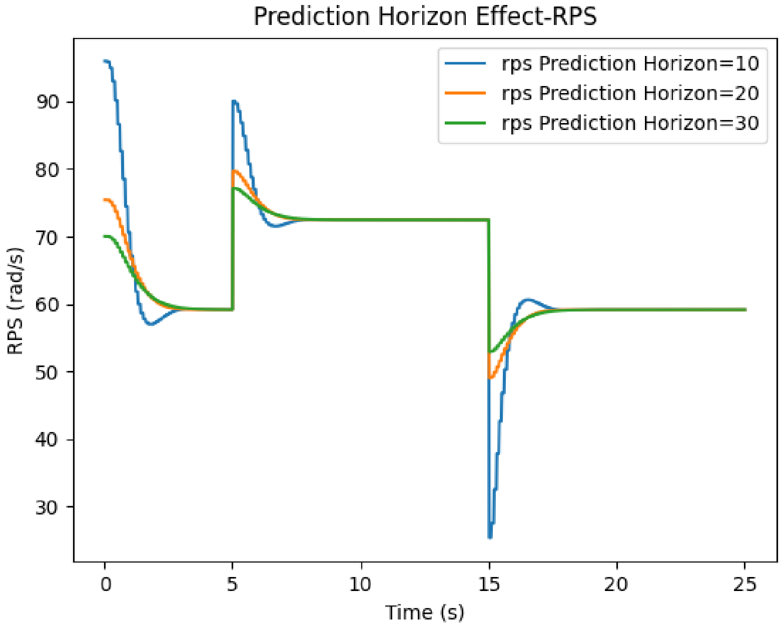


Figure 6.3 : Rps input values according to changing prediction horizon values

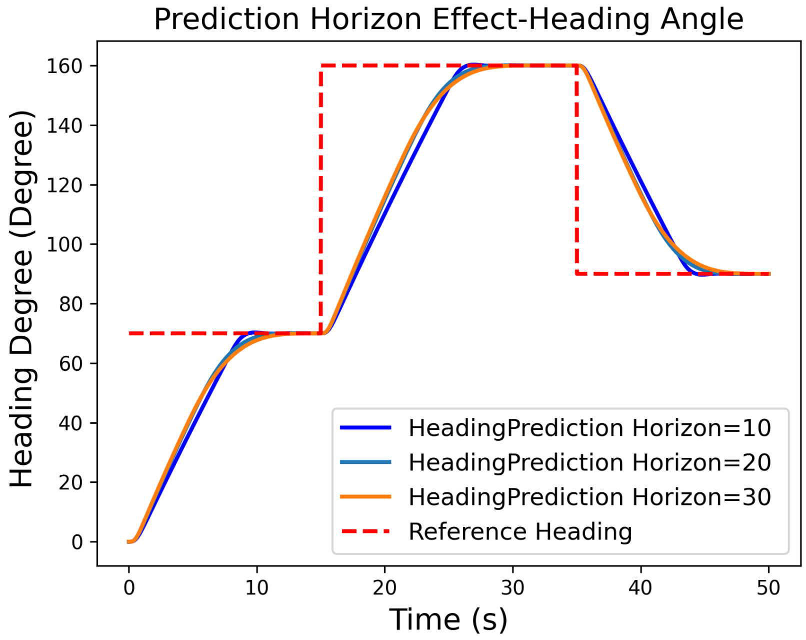


Figure 6.4 : Heading angle state behavior according to changing prediction horizon values

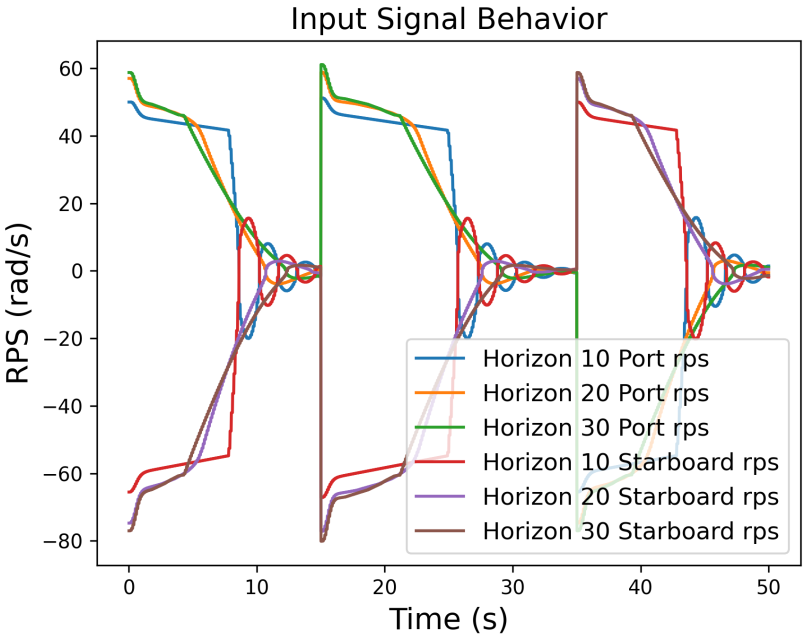


Figure 6.5 : Rps input values according to changing prediction horizon values for Heading angle state

Weight Matrix: The tuning parameters Q, R, and Δu weight matrix are the factors that affect the performance of the cost function (equation 6.1) within the MPC algorithm. These matrix can be utilized to weight the significance of states, control signals, and changing rates of control, respectively. The components of the weighting matrix Q penalize controlled outputs that deviate from their reference trajectories. Their relative values show the significance of the regulated outputs. The large-valued Q-matrix prioritizes following reference state values and penalizes states that differ significantly from the target value. R and Δu have a significant impact on the magnitudes and rates of change of the controller outputs. The relative values of Δu's components represent the relative costs of the changed inputs. Penalizing the change rate provides a more robust controller but causes the controller to be slower. This penalty of rate of change can be calculated in equation 6.9.

$$\Sigma_{k=1}^{n-1}|u_{t+k+1,t} - u_{t+k,t}|_2^R \tag{6.9}$$

Setting a small penalty or no penalty provides a more aggressive controller that is less robust Adjusting the weights of these weight matrices directly affects the controller's performance. For example, when the position is controlled, selecting the weight value of the relevant weight matrix too low will cause the control signals to be generated too low, unable to track the reference position. Otherwise, the system will act quickly and cause oscillation. First of all, in order to observe how the weights of the Q and R matrix, which affect the situations, affect the path following, the effect on the scenarios in which we follow the path which includes different paths, was examined. Four different scenarios are setted. While scenarios 1 and 2 show the effect of weight matrix of the cost function for a straight line path, scenarios 3 and 4 show the effect of weight values for a complex path.

Scenario-1, weight of position states affect for path following: To observe the effect of the weight matrix on the heading state, the values of the position states were tested in three different weight matrices. The weights of the states are selected as 1,10,100 respectively. In this scenario, the weight value affecting the linear velocity effect is kept constant.

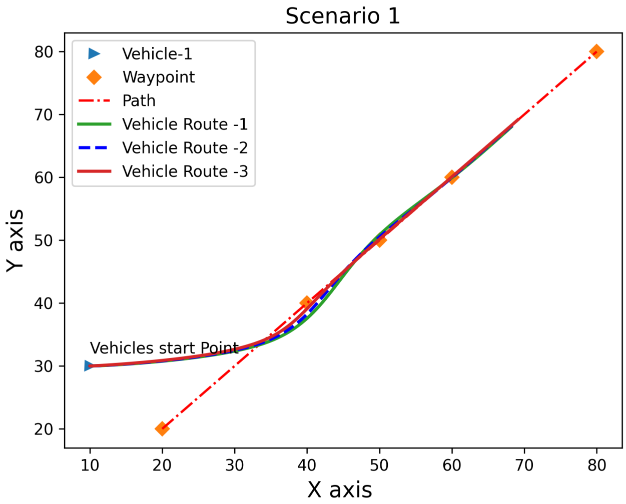


Figure 6.6 : Scenario-1: Path following performance according to different Q matrix weight values.

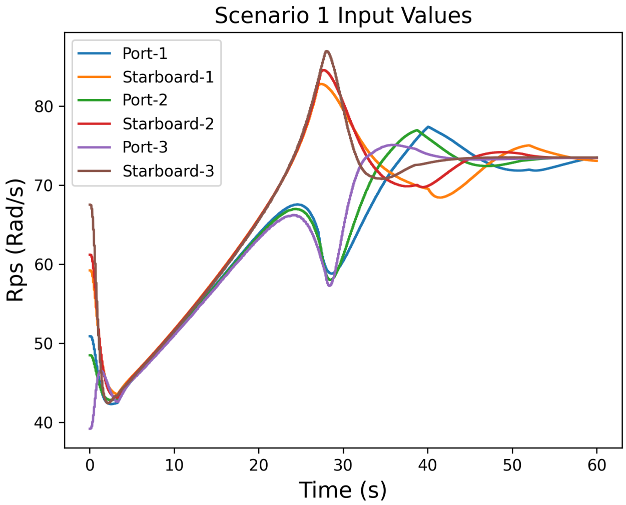


Figure 6.7 : Scenario-1: Input signal behavior according to different Q matrix weight values.

It has been observed that in the path following scenario where disruptors do not affect and a linear path is followed, changing the weights affecting the position states does not have much effect for this scenario. However, slight differences are observed in the input signals. Path following performance is shown figure 6.6 and control signals is in figure 6.7.

Scenario-2, weight of input affect for path following: In this scenario, To observe the effect of input weight, MPC has been executed with different input weights. the Q matrix values are assigned the same values [1,0,5] and the R matrices that affect the inputs are set as $R_1 = 1, R_2 = 0.1$ and $R_3 = 0.001$. Same input values are used for port and starboard inputs. The values of the input weight matrix had a greater effect on the path following performance than the state weight matrix. It is seen that the response of the input signals becomes more aggressive as the R values decrease. If it is reduced too much, oscillations can be seen in the input signals. However, since the path followed is a simple straight line, a performance analysis was made on a more difficult path in scenario-3 to observe this path following performance that is shown in figure 6.8 and control signals shown in figure 6.9.

Scenario-3 In scenario 3, the performance of MPC has been observed on a more challenging path using the weight matrix values from scenario 2, and compared based on changing R matrix values. In a scenario where following a road that includes turning maneuvers, it was observed that the most optimal input weight matrix value for generating the required input signals to follow the desired path was 0.001. The result is shown figure 6.10 and control signals are shown in figure 6.11.

Scenario-4 In this scenario, performances were observed based on varying position states weight matrix values, using the input weight matrix values determined as optimal in scenario 3. The matrix values are in the form of 5, 50, and 500, and it is observed that the performance of the controller increases as the weight value increases. The optimal values for state positions matrix value is 500 and input weight matrix value is 0.001. Results are shown in figure 6.12 and signals of this is shown in figure 6.13.

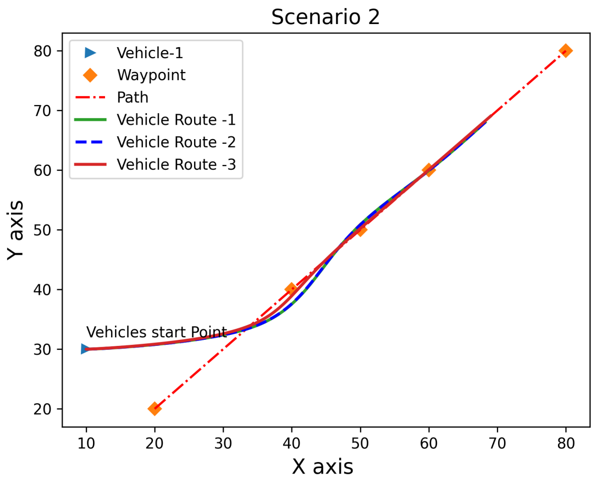


Figure 6.8 : Scenario-2: Path following performance according to different R matrix weight values.

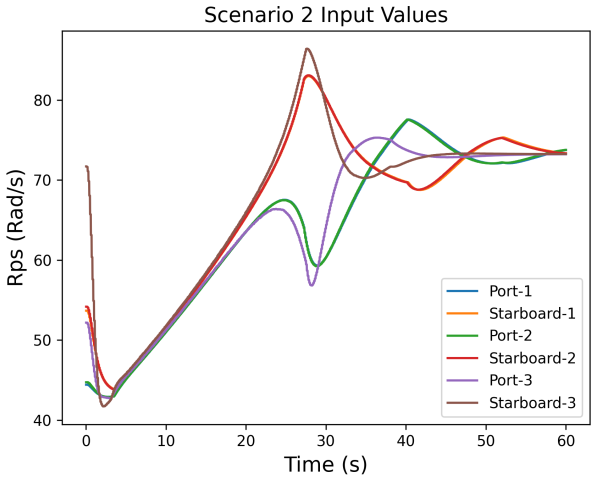


Figure 6.9 : Scenario-2: Input signal behavior according to different R matrix weight values.

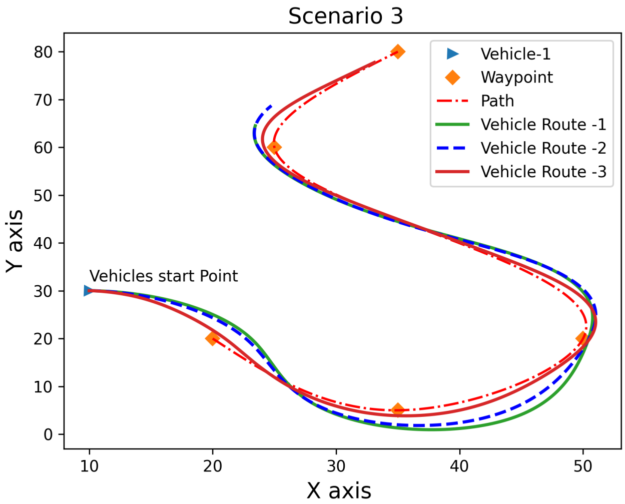


Figure 6.10 : Scenario-3: Path following performance according to different R matrix weight values for curved path.

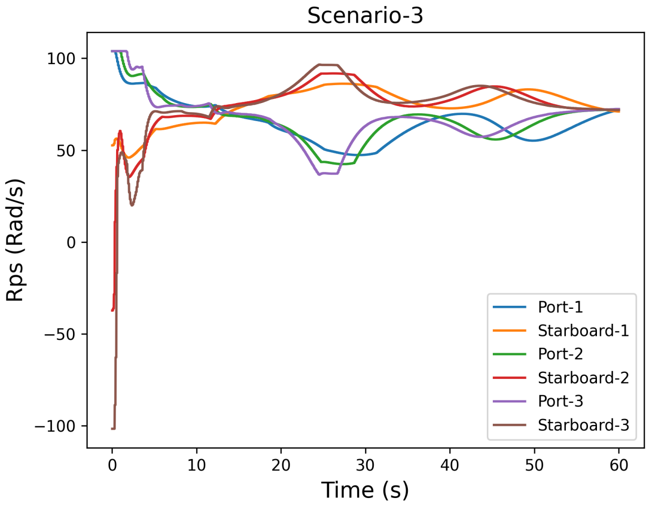


Figure 6.11 : Scenario-3: Input signal behavior according to different R matrix weight values.

6.2.2 Performance under disturbance

When there are environmental disturbances that are difficult to model or are not taken into account at all, it directly impacts the performance of the MPC negatively, because optimization in MPC work is done through the mathematical model of the system to be managed. The disturbances that cannot be calculated or calculated correctly cause MPC to make incorrect predictions in the prediction section. In order to minimize the error between the real state of the vehicles and the states predicted in the prediction part, the differences between these two states are given as feedback to the MPC block.

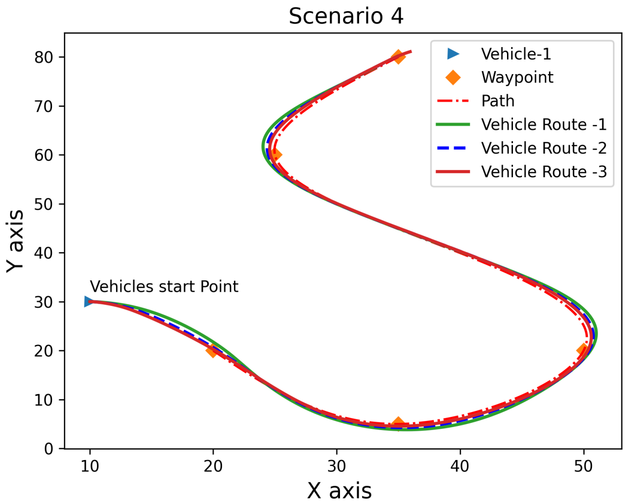


Figure 6.12 : Scenario-4: Path following performance according to different position states weight values for curved path.

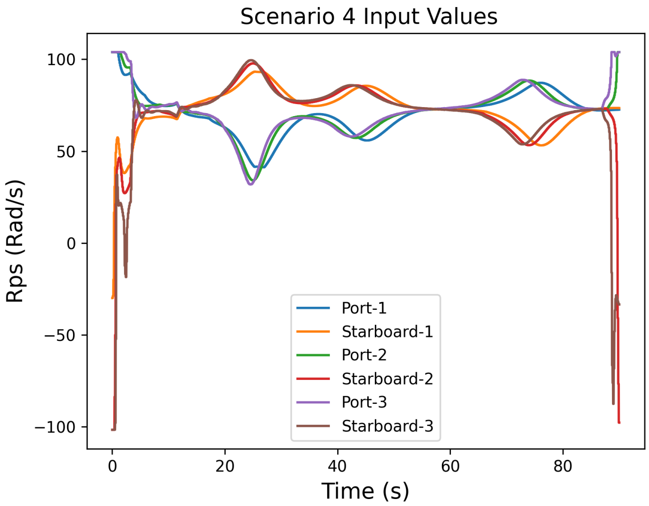


Figure 6.13 : Scenario-4: Input signal behavior according to different position states weight values.

This method is called as output feedback [38]. Adding a bias correction, $b(k+j)$, to the forecast is a common strategy. The definition of the corrected prediction, $\widetilde{y}(k+j)$, is [39]

$$\widetilde{y}(k+j) \cong \hat{y}(k+j) + b(k+j) \tag{6.10}$$

$\hat{y}(k+j)$ refers uncorrected prediction. The last measurement, $y(k)$, and the matching predicted value, $\hat{y}(k)$, are often used to define the bias correction:

$$b(k+j) = y(k) - \hat{y}(k) \tag{6.11}$$

This structure can be seen in figure 6.1.

6.2.2.1 Current disturbance

We examined the path following scenarios that include current disturbances acting on the vehicle from different directions and magnitudes in order to understand the effect of the disturbance effect on the control states. In this section, we observed the controller performances of the current forces acting at different current speeds and directions, according to different W matrix values, by disrupting the environment in which the vehicle operates.

0.2 m/s and 0 degree current: As stated previously, state following performance is determined by the Q weight matrix's interest values in the MPC cost function. If the states weight value is increased, it is tried to converge more aggressively to the desired state, while at lower weight values, this behavior occurs more slowly. While speed is being tracked in an environment where there are disturbances, the disturbance effect acts as a distraction from the desired state of the system. Its behavior is shown in figure 6.14 Therefore, this is achieved by increasing the state weight matrix value so that the system can follow the desired state. However, this input signal causes aggression in their behavior that is shown in figure 6.15.

First, the performance of following the effect of the current, which has a speed of 0.2 m/s and moves in the direction of 0 degrees, at different $Q_1 = 1$, $Q_2 = 10$ and $Q_3 = 500$ values, are shown in the figure. When we look at the controller performance of the system, the performance of the matrix with the same value as the Q weight matrix used in the disrupting environment remained low. In the environment where the current, 0.2 m/s current velocity and an accelerating effect from behind the vehicle, the difference in speed between the reference speed and the vehicle was observed as 0.1 m/s. The speed error in Q_2 and Q_3 values was 0.05 and 0.02 m/s, respectively. In the input signal behavior, the larger Q matrix value produced more aggressive input signals.

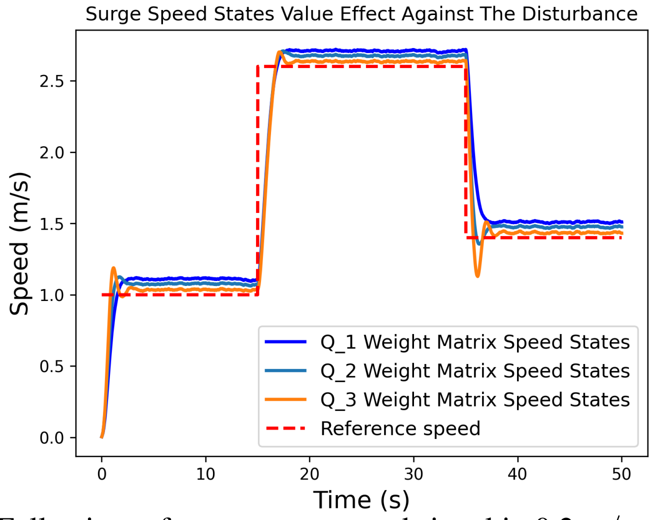


Figure 6.14 : Following reference surge speed signal in 0.2 m/s current speed and 0 degree current direction disturbance.

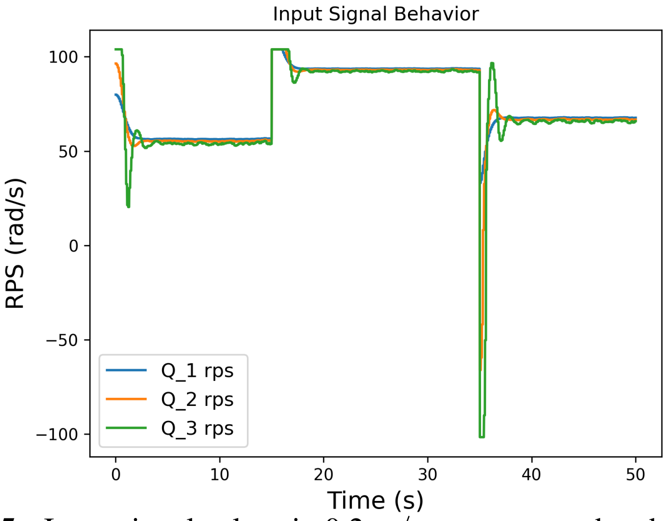


Figure 6.15 : Input signal values in 0.2 m/s current speed and 0 degree current direction disturbance.

0.5 m/s and 90 degree current:

When the vehicle speed was zero, the heading position control was tried to be made under the influence of the current speed affecting the vehicle at a speed of 0.5 m/s and towards the 90 degree direction. For this scenario, the performance of heading position control was compared on three different Q matrix values. $Q_1 = 1$,$Q_2 = 50$ and $Q_3 = 500$ matrix are used. Performance of controller is shown in figure 6.16 and control signals behavior is in figure 6.17

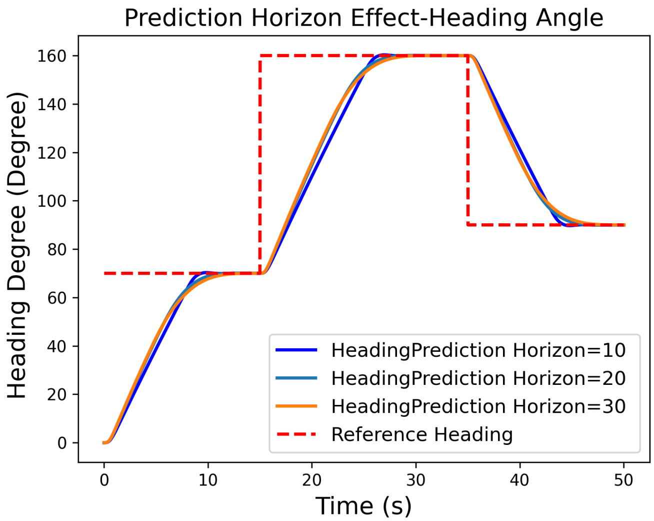


Figure 6.16 : Following reference heading signal in 0.5 m/s current speed and 90 degree current direction disturbance.

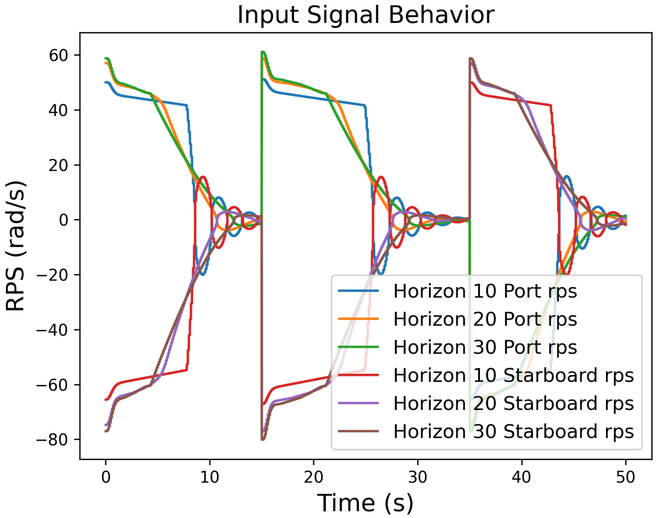


Figure 6.17 : Input signal values in 0.5 m/s current speed and 90 degree current direction disturbance.

Since position control is studied in this scenario, there is no error such as speed control, which differs depending on the Q matrix values that vary between the reference signal and the actual vehicle speed. There may be a slowdown in reaching the reference value of the system.

When examining the performance of MPC under disturbances, it is observed that a high penalty should be applied to the error value to prevent deviations from the desired reference state values due to the disturbance. It is observed that assigning high weight values to both the speed and heading state variables leads to better performance in a disturbed environment compared to models with lower weight values.

Controller performance under current disturbances In this scenario, the path following performance of the vehicle has been observed under the influence of flow forces from different directions. There are flow forces acting on the vehicle at speeds of 0.2, 0.3, and 0.4 m/s from 0, 60, and 125 degrees, respectively. Looking at the results, it has been observed that a different performance was exhibited in vehicle path following compared to the performance in an environment without disturbances. Due to the disruptive effects, the vehicle exhibited a lower performance in path following.As the weight matrix value affecting the heading decreases, it is observed that the vehicle's path following performance decreases. When the vehicle has Q has 10 for speed, 500 for heading and R = 0.001 values in an environment with disturbances, its performance is at an acceptable level. It is observed that the ability to maintain course against disturbances is better with respect to the other heading weight values. Performance of controller is shown in figure 6.18.

As a result, due to the absence of the flow velocity model in the system modeled with SI, a decrease in the performance of the controller has been observed in environments influenced by the flow. As the effect of the disturbance force increases, performance will also decrease.

6.3 Feedforward PID

Feed-forward augmentation is a prediction technique that estimates the output from a proportional-integral-derivative (PID) control algorithm without waiting for the PID algorithm to respond [40]. Block diagram of the controller is shown in figure 6.19 For feedforward part of the controller, the system input values required to reach the reference value of the request must be determined. For this, the input RPM values required for the desired linear speed of the speed controller of the vehicle were graphed. A 2nd degree curve was fitted to the generated graph using the curve fitting method. According to the curve, required RPM valeu can obtain for desired surge speed. Then, the error that occurs after the given F term to reach the desired reference value is minimized by adding P and I to the controller. The mathematical representation of this is as stated in the equations (6.12-14).

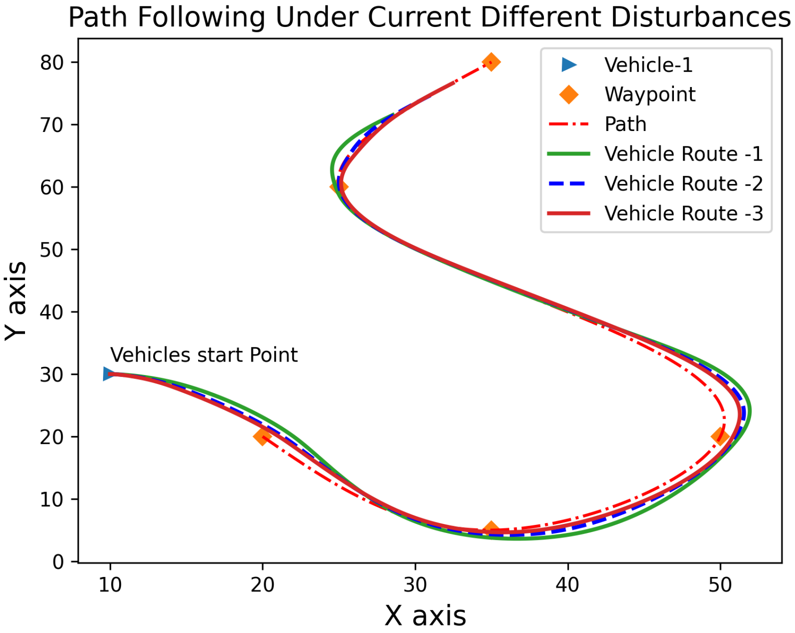


Figure 6.18 : Following path under current disturbances.

$$u = K_p e_u + K_i \int e_u + K_f \tag{6.12}$$

where;

$$e_u = u_{ref} - u \tag{6.13}$$

$$K_f = 4.295u_{ref}^3 - 25.1u_{ref}^2 + 66.66u_{ref} + 11.51 \tag{6.14}$$

Table 6.2 : Surge speed FPID controller coefficients.

Parameters	Value
K_p	60
K_i	8

For the generation of the reference signal for the vehicle's linear speed, the signals were limited to prevent abrupt reference signal variations. This limit is determined according to the maximum acceleration value of the vehicle. In figure 6.20, the performance of the surge speed controller is shown.

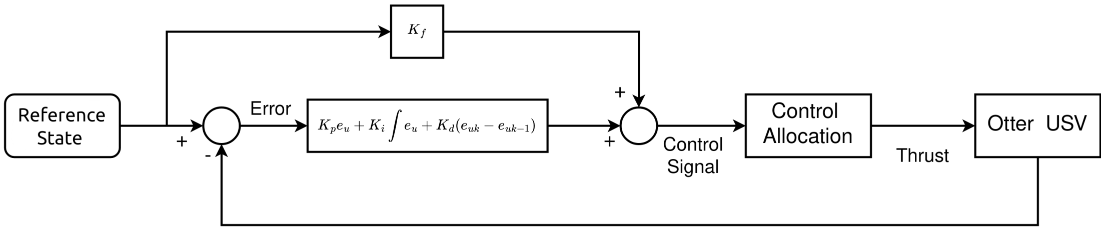


Figure 6.19 : FPID surge speed controller block diagram

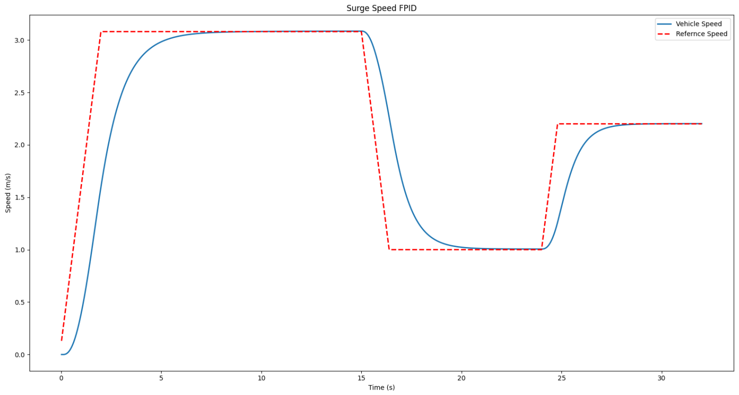


Figure 6.20 : Surge speed FPID controller.

6.4 Heading PID controller

PID controller design is used for the Heading controller that is shown in figure 6.21. The mathematical explanation is shown in the equations 6.15-16.

$$\psi = K_p e_\psi + K_i \int e_\psi + K_d(e_{\psi k} - e_{\psi k-1}) \tag{6.15}$$

where;

$$e_\psi = \psi_{ref} - \psi \tag{6.16}$$

In order to prevent oscillation due to derivative effect in the sudden change of reference in the control signal, the reference signal change rate of the system is restricted. Rate

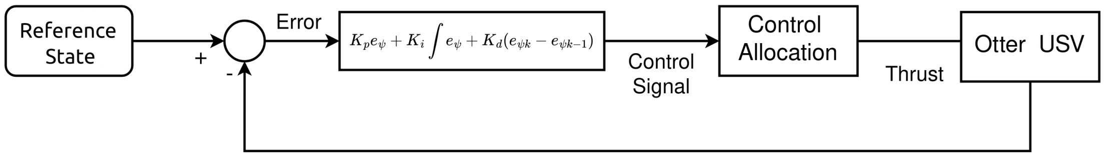


Figure 6.21 : PID heading controller block diagram

Table 6.3 : Heading PID controller coefficients.

Parameters	Value
K_p	3.5
K_i	0.001
K_d	30

of change was determined according to the maximum rotation angle ratio. The heading PID control performance is shown in figure 6.22.

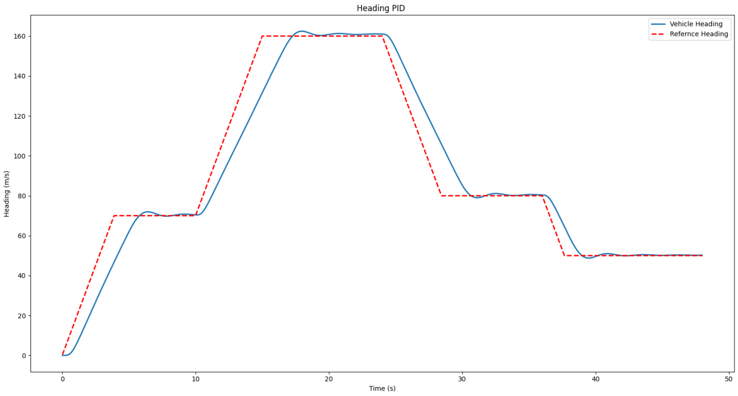


Figure 6.22 : Heading PID controller.

6.5 Control allocation

In normal operation of the vehicle, neither a signal will be produced only for surge speed nor the heading pid controller work only for reference heading position control. In a scenario where signals from these two controllers are used together, the vehicle will try to reach the desired heading reference while traveling at a surge speed. Therefore, a structure should be designed in which the signals of these two controllers are allocated. The allocation structure used in this tool is as shown in the equation.

$$U_{portsignal} = U_{surgespeedsignal} + U_{headingsignal} \tag{6.17}$$

$$U_{starboardsignal} = U_{surgespeedsignal} - U_{headingsignal} \tag{6.18}$$

7. SIMULATION

7.1 Introduction

In the simulation section, the design of the surge speed FPID and heading PID controllers for the Otter vehicle is explained to compare the performance of the path following controller developed using MPC with the PID controller. Later on, the performance of the controller under the influence of disruptive forces was demonstrated. Using Kinodynamic RRT algorithm, path planning was performed for environments with static obstacles and port docking scenarios, and the performance of the controllers in cases where the path was affected or unaffected by disturbances was demonstrated.

7.2 Development Environment

The software developed within the scope of this thesis has been developed and runs on the Ubuntu 20.04 operating system using the Python programming language. Also, the simulation environment in which the study was carried out was made using the dynamic model created by Fossen in the Python programming language for Otter USV.

7.3 Path Following Objective

To achieve proper path following, it is vital to ensure that the vessel's control continually performs well and can follow simple references. This can be accomplished by modifying the course and speed controls to execute benchmark operations, like turning and accelerating, in order to get a boat that is accurate and responsive. In this study, two different controller designs were made. The first is the feedforward PID controller, and the other is the Model Predict Control (MPC) design, which is the main scope of the thesis. The performance of MPC according to changing scenarios will be compared with FPID in the next section.

7.4 Path Following Control

In order for the LOS path following algorithm, whose theory is explained in Part 4, to follow a path generated from start point to goal point, the surge speed value that varies based on the cross-track error value and on the route and the reference heading angle created to direct the vehicle to the virtual target point placed at a distance of a determined ship length on the path are generated. Generated surge speed and heading angle are used as targets for PID controller. The controller optimally controls the system according to the changing targets, allowing the vehicle to move on the determined path. Figure 7.1 in chapter 7 represents the general structure of the GNC system necessary for route tracking.

The path to be followed is generated in the Path generator block (section 3) based on the given goal point that provided to the system. To follow this created path, the values of the states that will be entered as references to the controller block in the path following block (section 4) are determined. In the controller block, calculations are made until the time horizon determined with the help of the system model and the optimizer according to the references given. Differences between the real system and the system model are feedback to the optimizer as an error.

7.5 Static Obstacle Environment Scanerios

In this section, environmental conditions similar to narrow waterways were created to test the performance of controllers in scenario paths where there may or may not be an effect of environmental disturbances.

7.5.1 Static obstacle environment without disturbance

Simulations were conducted on following the path generated by kinodynamic RRT to reach a target point from a starting point in an environment with static obstacles, using LOS guidance. Although kinodynamic RRT considers vehicle dynamics in the path planning process, oscillations may occur in the generated path due to the random node creation step in the algorithm. The position data of this path is passed through an average window filter to smooth out the path, resulting in a more suitable path to

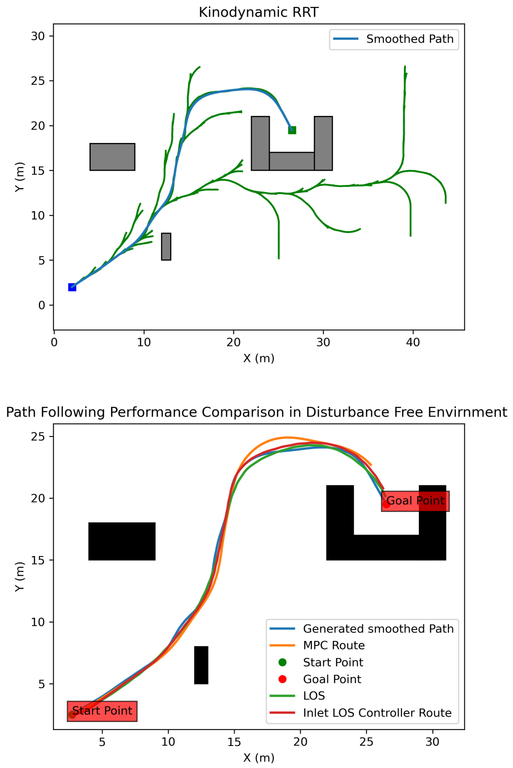


Figure 7.1 : Path following performance in disturbance free environment.

follow. As shown in figure 7.1, the path displayed in blue represents the path after the smoothing process. In figure 7.2, the performance of the NMPC-based path following and FPID controller path following with the LOS guidance algorithms is demonstrated in an environment without disturbances. Although there is not a significant difference between the performance of the two controllers, it is observed that path following algorithms show similar performance under disturbance free environment.

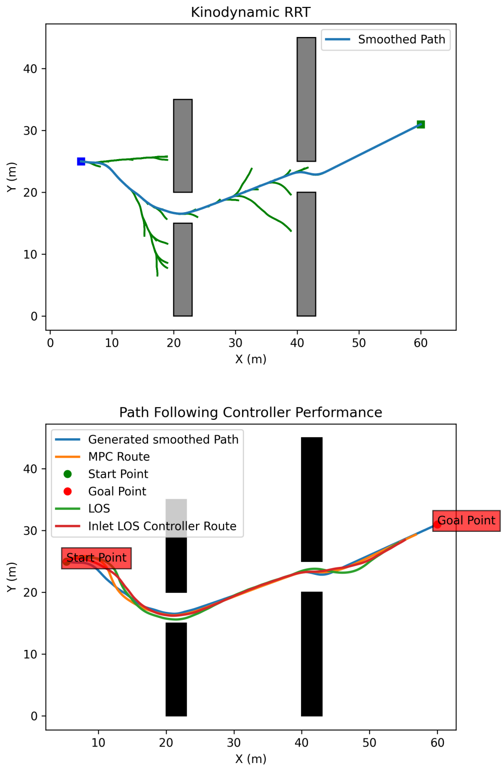


Figure 7.2 : Path following performance in disturbance free environment.

7.5.2 Static obstacle environment with disturbance

When the performance of the controllers was tested in different environments with a flow disturbance, the results are shown in figures 7.3-4. In this scenario where the disturbance is present, the flow was affected by the disturbance to 90 degrees at different times with a speed of 0.3 m/s. As seen in the graphs, path following performance of the MPC is better than that of the PID controller for LOS algorithms, so in an environment with disturbances, the MPC path following algorithm has shown better performance compared to the normal LOS and Inland water LOS algorithms.

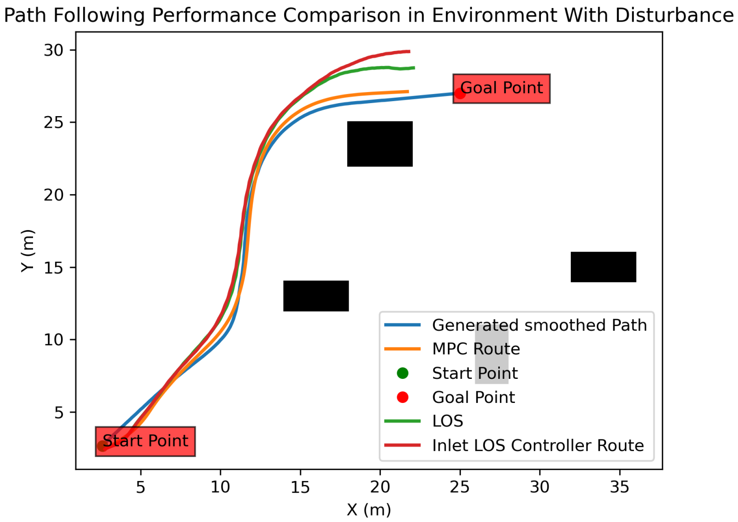


Figure 7.3 : Controller performance for disturbance scenario

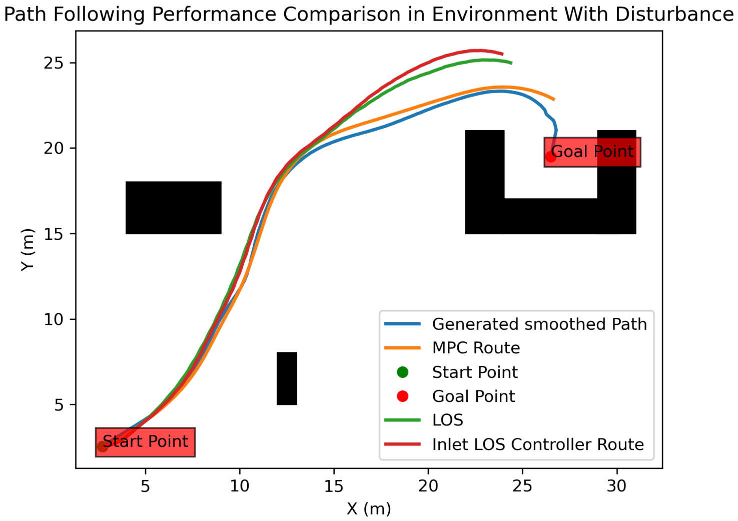


Figure 7.4 : Controller performance for disturbance scenario

8. CONCLUSIONS AND FUTURE WORK

The objective of this thesis was to design and implement guidance, navigation, and control systems that enable autonomous point-to-point sailing of the Otter USV simulation model. A path planning study was carried out for the guidance block of the GNC concept, from a starting point determined by the kinodynamic RRT, which is a sample-based method that takes vehicle dynamics into account, to a target point. In addition, due to the inherent nature of the RRT algorithm, the generated path is not always optimal, and both the time required for path generation and the actual path itself can be different each time. When a more stable structure is desired for path planning, optimization-based path planners can be beneficial. The structure of the path generated by intervening in the dynamic constraints of the optimization problem can be altered. For example, by changing the vehicle's cruising speed, the structure of the path generated by the algorithm in an environment with constraints can be modified. This also provides diversity for path created for different scenarios in inland waterways. Further development of this approach is left for future research for this path generation approach.

In the pursuit of following the generated path, we can say that the structure of the path directly affects the performance of path following algorithms. In environments with tight spaces, heavy traffic, or high maneuvering requirements such as inland waterways, it can be challenging to track the generated path. Classical LOS algorithms may not perform well under such conditions. Therefore, in this study, we improved the performance of path following by modifying the structure of the LOS algorithm, enhancing the performance of path following particularly in inland waterways.

The performance of the path following using MPC and PID with LOS has been compared in different scenarios during the following of the generated path. To compare the performance of the MPC model developed for Otter, both a suitable PID controller for the vehicle has been designed, and cruise data has been generated in the Otter vehicle simulator to create a vehicle model using a nonlinear least squares method and a system identification approach for the system model running within the MPC algorithm. To improve the controller performance of the MPC, the effects of parameters affecting optimization have been compared in different scenarios, and the appropriate parameters for the system have been determined. The performance of the created controllers has been tested in various scenarios, where disturbances are present, to evaluate their performance in environments affected by disturbances.

In path following part,

When we look at the future study topics regarding path planning and following, optimization-based algorithms have begun to demonstrate their success in this field. OCP, one of these algorithms, provides acceptable outputs in complex environments where many constraints need to be taken into account. The cost function, which is at the core of such optimization-based algorithms, provides flexibility in creating problem constraints. For example, by defining the situations we want to take into account and adding them to the structure of the cost function, solutions suitable for this situation can be produced Thus, optimizations can be made in areas such as energy, time, and distance, and a general path planning that is suitable for vehicle dynamics can be created. Furthermore, it is necessary to work on creating a general path by providing the vehicle with curve sections that it can follow using differential geometric approaches, utilizing information about the vehicle's turning radius at predetermined RPM values as a subject that needs to be addressed in general path planning.

REFERENCES

[1] N. Berg, J. Storgård, and J. Lappalainen, “The impact of ship crews on maritime safety,” *Publications of the Centre for Maritime Studies, University of Turku A*, vol. 64, pp. 1–48, 2013.

[2] C. Chauvin, S. Lardjane, G. Morel, J.-P. Clostermann, and B. Langard, “Human and organisational factors in maritime accidents: Analysis of collisions at sea using the hfacs,” *Accident Analysis & Prevention*, vol. 59, pp. 26–37, 2013.

[3] R. Ghaemi, S. Oh, and J. Sun, “Path following of a model ship using model predictive control with experimental verification,” in *Proceedings of the 2010 American control conference*, pp. 5236–5241, IEEE, 2010.

[4] May 2023.

[5] A. M. Lekkas, “Guidance and path-planning systems for autonomous vehicles,” 2014.

[6] C. Zhou, B. Huang, and P. Fränti, “A review of motion planning algorithms for intelligent robots,” *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.

[7] A. K. Guruji, H. Agarwal, and D. Parsediya, “Time-efficient a* algorithm for robot path planning,” *Procedia Technology*, vol. 23, pp. 144–149, 2016.

[8] M. Xu, Y. Wang, Z. Han, and X. Jia, “Unmanned surface vehicle path following based on path parameter description,” in *Global Oceans 2020: Singapore–US Gulf Coast*, pp. 1–6, IEEE, 2020.

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

[10] N. Shaukat, A. Ali, M. Javed Iqbal, M. Moinuddin, and P. Otero, “Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter,” *Sensors*, vol. 21, no. 4, p. 1149, 2021.

[11] T. SNAME, “Nomenclature for treating the motion of a submerged body through a fluid,” *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin*, no. 1950, pp. 1–5, 1950.

[12] T. I. Fossen, *Nonlinear modelling and control of underwater vehicles*. Universitetet i Trondheim (Norway), 1991.

[13] J. Lan, M. Zheng, X. Chu, and S. Ding, "Parameter prediction of the non-linear nomoto model for different ship loading conditions using support vector regression," *Journal of Marine Science and Engineering*, vol. 11, no. 5, p. 903, 2023.

[14] T. I. Fossen, S. I. Sagatun, and A. J. Sørensen, "Identification of dynamically positioned ships," *Control Engineering Practice*, vol. 4, no. 3, pp. 369–376, 1996.

[15] R. Skjetne, Ø. Smogeli, and T. I. Fossen, "Modeling, identification, and adaptive maneuvering of cybership ii: A complete design with experiments," *IFAC Proceedings Volumes*, vol. 37, no. 10, pp. 203–208, 2004.

[16] K. Fedyaevsky and G. Sobolev, "Control and stability in ship design," 1964.

[17] N. H. Norrbin, "Theory and observation on the use of a mathematical model for ship maneuvering in deep and confined water," in *Proc. 8th Symposium on naval Hydrodynamics*, 1977.

[18] L. Ljung, *System identification.* Springer, 1998.

[19] H. A. Nielsen, H. Madsen, *et al.*, *Predicting the heat consumption in district heating systems using meteorological forecasts.* Citeseer, 2000.

[20] Ü. Öztürk, M. Akdağ, and T. Ayabakan, "A review of path planning algorithms in maritime autonomous surface ships: Navigation safety perspective," *Ocean Engineering*, vol. 251, p. 111010, 2022.

[21] A. Vagale, R. Oucheikh, R. T. Bye, O. L. Osen, and T. I. Fossen, "Path planning and collision avoidance for autonomous surface vehicles i: a review," *Journal of Marine Science and Technology*, pp. 1–15, 2021.

[22] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[23] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[24] A. M. Lekkas and T. I. Fossen, "Integral los path following for curved paths based on a monotone cubic hermite spline parametrization," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2287–2301, 2014.

[25] M. Breivik, "Topics in guided motion control of marine vehicles," 2010.

[26] D. Mu, G. Wang, Y. Fan, X. Sun, and B. Qiu, "Adaptive los path following for a podded propulsion unmanned surface vehicle with uncertainty of model and actuator saturation," *Applied sciences*, vol. 7, no. 12, p. 1232, 2017.

[27] J. H. Lenes, "Autonomous online path planning and path-following control for complete coverage maneuvering of a usv," Master's thesis, NTNU, 2019.

[28] E. F. Camacho and C. B. Alba, *Model predictive control.* Springer science & business media, 2013.

[29] J. Maciejowski, *Predictive Control with Constraints.* England.: Prentice Hall, 2002.

[30] F. Allgöwer and A. Zheng, *Nonlinear model predictive control*, vol. 26. Birkhäuser, 2012.

[31] M. Diehl, *Real-time optimization for large scale nonlinear processes.* PhD thesis, 2001.

[32] D. Mayne, "Nonlinear model predictive control: Challenges and opportunities," *Nonlinear model predictive control*, pp. 23–44, 2000.

[33] euraad (https://math.stackexchange.com/users/443999/euraad), "Is there any rule of thumb when it comes to selecting control/predict horizon for mpc?." Mathematics Stack Exchange. URL:https://math.stackexchange.com/q/3052921 (version: 2018-12-26).

[34] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*, vol. 2. Nob Hill Publishing Madison, WI, 2017.

[35] M. Diehl, H. G. Bock, H. Diedam, and P.-B. Wieber, "Fast direct multiple shooting algorithms for optimal robot control," *Fast motions in biomechanics and robotics: optimization and feedback control*, pp. 65–93, 2006.

[36] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes.* SIAM, 2010.

[37] G. Reddy, "Hierarchical model predictive control for trajectory generation and tracking in highly automated vehicles," 2016.

[38] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[39] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle III, *Process dynamics and control.* John Wiley & Sons, 2016.

[40] P. Nachtwey, "Feed forwards augment PID control." `https://www.controleng.com/articles/feed-forwards-augment-pid-control/`, Mar. 2015. Accessed: 2023-3-18.

CURRICULUM VITAE

Ferhan BÜYÜKÇOLAK:

EDUCATION:

- **B.Sc.:** 2018, Istanbul Technical University, Naval Architecture and Ocean Engineering Faculty, Shipbuilding and Ocean Engineering

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2020- Present TÜBITAK BILGEM - Naval Platform Navigation Systems Department