NTNU
Norwegian University of
Science and Technology

Towards the Development of Autonomous Ferries

Glenn Ivan Bitar

Master of Science in Cybernetics and Robotics
Submission date: September 2017
Supervisor: Morten Breivik, ITK
Co-supervisor: Anastasios Lekkas, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Preface

This thesis concludes five years of education at Telemark University College, South Dakota School of Mines and Technology, and at the Norwegian University of Science and Technology. The experiences I have had during this period have been joyful, formative and humbling.

Thanks to my advisor Morten Breivik and to my co-advisor Anastasios Lekkas for invaluable guidance throughout the duration of my master's thesis. Thanks to my peers for our discussions and their encouragement. Thanks to my friends and family for their endless love and support. Thanks to the amazing person who proofread this thesis. The examiner will be grateful.

Abstract

Autonomous ships is at the moment a heavily researched topic in the maritime industry. Development to introduce autonomous ferries in the Norwegian fjords is under way. This thesis is a study of technical and formal challenges related to autonomous ferries. The thesis goes into topics such as industrial control systems for ships, path planning and collision avoidance algorithms, as well as automatic docking. Additionally, information and statistics regarding ferry activities in Norway are presented.

Various path planning algorithms are examined, and a distinction has been made between roadmap methods and optimization-based parametrized path generation. Additional research on the optimization-based methods is performed. Software for optimal control with the pseudospectral method is demonstrated. A nonlinear 3-DOF ship model is used to generate feasible time-optimal and energy-optimal paths. A guidance controller is implemented in a simulation to verify the feasibility of the paths when the ship is exposed to unknown disturbances.

A selection of collision avoidance algorithms are reviewed. This is related to sensors, sensor fusion and target tracking systems.

Finally, various methods for automatic docking are reviewed. This review consists of algorithms used with underwater and surface vessels in static environments.

Sammendrag

Autonome skip er nå et tema som blir forsket mye på i den maritime industrien. Utvikling for å introdusere autonome ferjer er i norske fjorder er i gang. Denne oppgaven er en studie i tekniske og formelle utfordringer relatert til autonome ferjer. Oppgaven går inn i temaer som industrielle styresystemer for skip, stiplanlegging, algoritmer for kollisjonsunngåelse samt automatisk anløp. I tillegg til dette er det samlet informasjon og statistikk om ferjeaktivitet i norske fjorder.

Forskjellige stiplanleggingsalgoritmer er gjennomgått, og det blir skilt mellom veivalgsmetoder (roadmap) og optimaliseringsmetoder som produserer parametriserte stier. Det er gått dypere inn i optimaliseringsmetodene, hvor programvare for optimal regulering ved hjelp av pseudospektral metode har blitt benyttet. Her har en ulineær skipsmodell i tre frihetsgrader blitt brukt for å generere tidsoptimale og energioptimale stier. En gaidingsregulator er implementert i en simulering, for å verifisere at de genererte stiene er mulig å følge med ukjente forstyrrelser.

En gjennomgang av et utvalg algoritmer for kollisjonsunngåelse er også presentert. Dette er satt i sammenheng med sensorer, sensorfusjon og følgesystemer for bevegelige hindringer.

Til sist blir forskjellige metoder for automatisk anløp gjennomgått. Her vises det til algoritmer brukt i undervanns- og overflatefartøy i statiske omgivelser.

Contents

List of Figures

List of Tables

Acronyms

Acronym	Description
ADCP	Acoustic Doppler current profiler
AHRS	Attitude and heading reference system
AIS	Automatic identification system
API	Application programming interface
ARPA	Automatic radar plotting aid
ASV	Autonomous surface vehicle
AUV	Autonomous underwater vehicle
COLREGS	International Regulations for Preventing Collisions at Sea
COTS	Commercial off-the-shelf
DGPS	Differential GPS
DOF	Degree of freedom
DP	Dynamic positioning
ECDIS	Electronic chart display and information system
ECEF	Earth-centered, earth-fixed
EHF	Extra high frequency
EMODnet	European Marine Observation and Data Network
GNC	Guidance, navigation and control
GNSS	Global navigation satellite system
GPS	Global Positioning System
HF	High frequency

Acronym	Description
HJB	Hamilton-Jacobi-Bellman
IMO	International Maritime Organization
IMU	Inertial measurement unit
INS	Inertial navigation system
LBL	Long baseline
LF	Low frequency
LG	Legendre-Gauss
LGL	Legendre-Gauss-Lobatto
LGR	Legendre-Gauss-Radau
LIDAR	Light detection and ranging
LOLO	Lift-on/lift-off
LOS	Line-of-sight
LQG	Linear-quadratic-Gaussian
LQR	Linear-quadratic regulator
MF	Medium frequency
MHT	Multiple hypothesis tests
MIMO	Multiple-input multiple-output
MMSI	Maritime Mobile Service Identity
MRU	Motion reference unit
MTS	Marine Technology Society
NED	North-east-down
NIS	Normalized innovation squared
NLP	Nonlinear program
NMEA	National Marine Electronics Association
NPRA	Statens vegvesen (Norwegian Public Roads Administration)
ODE	Ordinary differential equation
PCE	Passenger car equivalent
PDE	Partial differential equation
PID	Proportional-integral-derivative
PRM	Probabilistic roadmap
PS	Pseudospectral

Acronym	Description
QP	Quadratic programming
RK	Runge-Kutta
RORO	Roll-on/roll-off
ROV	Remotely operated vehicle
RRT	Rapidly-exploring random trees
RTK	Real-time kinematic
SBL	Short baseline
SISO	Single-input single-output
SL	Semi-Lagrangian
SSBL	Super-short baseline
UGAS	Uniformly globally asymptotically stable
ULES	Uniformly locally exponentially stable
USBL	Ultra-short baseline
iUSBL	Inverted ultra-short baseline
USV	Unmanned surface vehicle
UTC	Coordinated universal time
VHF	Very high frequency
VRU	Vertical reference unit

Nomenclature

Symbol	Description
x	North position in NED frame
y	East position in NED frame
z	Down position in NED frame
ϕ	Roll angle
θ	Pitch angle
ψ	Yaw angle
$\boldsymbol{\eta}$	Generalized position vector
$\boldsymbol{\Theta}$	Vector of Euler angles
u	Surge speed
v	Sway speed
w	Heave speed
u_r	Surge speed, relative to ocean current
v_r	Sway speed, relative to ocean current
p	Roll rate
q	Pitch rate
r	Yaw rate
U	Absolute ground speed $U = \sqrt{u^2 + v^2}$
$\boldsymbol{\nu}$	Generalized velocity vector
$\boldsymbol{\nu}_r$	Vehicle velocity in BODY, relative to ocean current
$\boldsymbol{\nu}_c$	Ocean current velocity vector
V_x	Current velocity in the north direction
V_y	Current velocity in the east direction
$\boldsymbol{v}$	Vector of linear velocities
$\boldsymbol{\omega}$	Vector of angular velocities
X	Force in surge direction
Y	Force in sway direction
Z	Force in heave direction

Symbol	Description
K	Moment around x axis
M	Moment around y axis
N	Moment around z axis
$\boldsymbol{\tau}$	Virtual force vector
$\boldsymbol{u}$	Control action
χ	Course angle
β	Sideslip angle
m	Ship mass
∇	Ship displacement
e	Cross-track error
s	Along-track error
$\boldsymbol{p}$	Vehicle position in NED frame
θ	Path parameter
$\mathcal{A}(\boldsymbol{\eta})$	Subset of $\mathcal{W}$ occupied by ship
$\mathcal{C}$	Configuration space
$\mathcal{U}$	Set of admissible controls for the vector $\boldsymbol{u}$
$\mathcal{W}$	Workspace
$(\cdot)_d$	Desired value
$\mathbf{S}(\cdot)$	Skew symmetric matrix

Chapter 1

Introduction

Autonomous shipping is a topic under heavy research by major players in the industry. Technology developer Kongsberg and fertilizer firm Yara are developing the to-be autonomous container ship *Yara Birkeland* to reduce load on roads in eastern Norway [1]. See Figure 1.1. The port of Amsterdam is the workplace for the *Roboat* program, which explores and tests the possibilities for autonomous shipping [2].

Figure 1.1: The to-be autonomous container ship *Yara Birkeland*. © 2017 Kongsberg.

This thesis is concerned with the challenges and possible solutions related to autonomous ferries. Comprehensive background research on industrial control systems, path planning algorithms, collision avoidance methods and automatic docking is the main contribution of the thesis. An overview of statistics and ferry

Figure 1.2: The battery-powered ferry *Ampere*. © 2015 DNV GL.

activities in Norway is also provided. In addition, an algorithm for pseudospectral optimal control is utilized for path planning under different cost functions and environmental forces.

This introduction contains background and motivation for the thesis, a detailed problem description, as well as a summary of the thesis' contribution to the field.

1.1 Background and motivation

To reduce the environmental impact of transportation, Norwegian governmental organizations and other institutions focus heavily on emission-free ferries [3]. Examples of pilot projects investigating emission-free solutions include the battery-powered ferry *Ampere*, operating the Oppedal–Lavik connection along road E39 in western Norway [4]. *Ampere* is powered by two 500 kWh batteries that are partially recharged at both sides of the 20 minute crossing. *Ampere* is depicted in Figure 1.2.

In addition to emission-free technologies, autonomy may further improve energy efficiency, making battery-powered ferries feasible with longer ferry connections. Rolls-Royce cooperates with the ferry operator Fjord1 to develop an auto-crossing system for ferries [5, 6]. The connection Anna–Lote is operated by battery-powered ferries implementing the auto-crossing system, which plans the ship's route and controls the speed and acceleration, adapted to weather conditions in the area. Rolls-Royce continues to develop the auto-crossing system to handle docking and collision avoidance.

Reducing the cost of ferry operations by increasing the level of autonomy may reveal opportunities to revitalize coastal areas. Small ferries with low operational

costs may act as "dynamic bridges" in rivers and fjords. The city of Trondheim in Norway is investigating the possibilities of replacing a planned bridge across the river *Nidelva* with a small passenger ferry [7]. At a tenth of the price of a bridge, the ferry will carry cyclists and pedestrians over a narrow part of the river.

Many technical challenges lie between today's situation and that of having completely autonomous ferries. Among these are path planning, collision avoidance, automatic docking, robustness of low-level control, and traffic management. Developing methods that solve these challenges with energy-optimality as a criterion may greatly reduce environmental impact from transportation. Research motivated by autonomous ferries is also applicable to autonomous shipping in general.

1.2 Description of the ferry scenario

To understand which challenges the development of autonomous ferries comprises, it is important to have a clear picture of the tasks a conventional ferry performs under regular conditions. The obvious objective of a ferry is to transport people and vehicles between locations in a safe, timely and efficient manner. A typical sequence of events looks like this:

1. Ferry arrives at dock and keeps itself in place using its thrusters.
2. Hatches and doors open which let the vehicles and passengers off the ferry.
3. Ferry personnel guides waiting vehicles and passengers on-board the ship.
4. Hatches and doors close, and ticketing is performed.
5. Ferry undocks from the current harbor and starts transiting to the next one.
6. During transit and docking/undocking ferry personnel takes care to follow the International Regulations for Preventing Collisions at Sea (COLREGS) to avoid any collision.

During all stages of ferry operation, this list of priorities should be followed:

1. Safety — The top priority for any shipping operation is to keep people, material, and the environment safe from harm. This means that during transit, the ferry must be able to detect and avoid both static and dynamic obstacles.
2. Area of operation — If the ferry needs to react to obstacles, it will still need to stay within a safe corridor that is out of danger from reefs and land.

3. Punctuality — A ferry operation is not only about getting from one place to another while staying safe. Being on time is crucial for road communication, as many people and businesses depend on this.

4. Green — While maintaining safety and punctuality, operating the ferry as energy optimal as possible reduces cost and environmental impact.

The weather affects several aspects of ferry operation. Currents, wind and waves affect the ferry motion, which has implications for energy use. Being able to understand and account for how the weather affects ferry motion may increase efficiency. Extremely bad weather and tall waves may also force ferry operations to suspend temporarily, affecting local traffic.

1.3 Problem description

The problem of "Towards the Development of Autonomous Ferries" is considered. This includes giving a comprehensive introduction to the topics relevant to the development of autonomous ferries. These topics are identified by looking at the tasks and challenges humans solve when performing ferry operations. Statistics and context should be obtained to motivate the task, and to further identify the challenges ahead. The work includes:

- Give an overview of the main types of ferry activities in Norway today.
- Give an overview of motion control systems used for ferries today.
- Give an overview of path planning algorithms suitable for ferries.
- Give an overview of automatic collision avoidance algorithms relevant for ferry operations and scenarios.
- Implement an optimization-based path planning algorithm on a suitable ship model.
- Present the main challenges associated with automatic docking of ferries.
- Give an assessment of the main challenges and possible solutions associated with developing autonomous ferries.

1.4 Contributions

The main contribution of this thesis is the background research performed on path planning, collision avoidance and automatic docking. In addition, the use of pseudospectral optimal control for path planning under different ocean current conditions is tested. This is performed with minimum-energy and minimum-time cost

functions, and the method is validated by simulation using a guidance controller. Additional contributions are a collection of statistics regarding ferry activity in Norway, and an overview of sensors available to use for collision avoidance and navigation.

1.5 Thesis outline

Information and statistics regarding ferry activities in Norway are found in Chapter 2. Following that, Chapter 3 contains basics on ship modeling, as well as information on low-level control and control allocation. Chapter 4 is an introduction to the different types of control systems found in industrial ships. Chapter 5 contains research about path planning algorithms, including roadmap and optimization-based algorithms. The pseudospectral approach is further explored in Chapter 6, where optimal control is used to find time- and energy-optimized paths for a 3-DOF nonlinear ship model. Feasibility is confirmed by the use of a guidance algorithm. Collision avoidance algorithms are explored in Chapter 7. A brief overview of automatic docking algorithms is presented in Chapter 8. Chapter 9 serves as a discussion of the research performed in this thesis, and summarizes the characteristics of autonomous ferries. Chapter 10 concludes the thesis and proposes further work.

Chapter 2

Ferry Activities in Norway

Figure 2.1: A panoramic view of the fjords Øyfjorden and Mefjorden in Senja, Norway. Simo Räsänen.

Being a coastal country with a large prevalence of mountains and fjords, transportation has always been a big challenge in Norway. Bridges are not feasible to build over large bodies of water, thus ferries are commonly used for transportation over fjords and along the coast. This chapter provides some statistics and general information about ferry operations in Norway, as well as a brief overview of the laws and regulations governing these operations.

2.1 Statistics

The number of ferry connections in Norway was 121 in 2012 [8]. This had increased to approximately 150 connections by 2017. Of these, 19 belong to the national roads (riksvei) and 102 to the county roads (fylkesvei). There were in total 1.5 million trips, which transported 21 million vehicles and 43 million people an average

Figure 2.2: Overview of 124 Norwegian ferry routes. This image is generated from a list of 135 connections from the four major companies, where 11 were not able to be located. The colors identify the companies: Red is Fjord1, blue Norled, green Torghatten and yellow Boreal. Made with `https://batchgeo.com/`.

of 8 kilometers each. In 2012, ferries produced 350 million passenger kilometers. This was 0.5% of the total domestic transportation, where private cars account for 78%, air transport 6.3% and buses 5.0% [9]. In 1997, Norway kept close to 10% of the world's car ferry fleet [10].

Transportation is the largest contributor to climate gas emissions in Norway, and domestic shipping (including ferries) is the third largest contributor within transportation, accounting for 2.8 million tons CO_2 equivalents in 2015 [11]. Along with offshore and fishing vessels, ferries are among the worst contributors within the domestic shipping section [12].

2.2 Ferry operators

Per 2017, there are four companies operating the vast majority of the ferry connections (135 of approximately 150). Torghatten with 47 connections, Norled (40), Fjord1 (35) and Boreal (12) account for approximately 99% of the revenue generated from ferry transport [13]. There are also seven smaller companies running various single connections, often on behalf of a larger company [14]. In addition, some municipalities have small ferry connections over rivers and lakes, with little or no statistics.

Figure 2.2 displays the ferry connections, colored by operator[1].

2.3 Designers and shipyards

The Norwegian ferries are designed specifically for their environment and role. Multi Maritime is a ship design company based in Førde, Norway. Among their designs are the car ferries *Bastø IV*, *V* and *VI*, *Tidefjord* and *Ullensvang*. The *Bastø* ferries are built at the Turkish companies Cemre Shipyard and Sefine Shipyard. *Tidefjord* and *Ullensvang* are built at the Norwegian shipyards Fiskerstrand and Ankerløkken, respectively.

Another ship design company is LMG Marin, based in Bergen, Norway. Among their designs are the car ferries *Stavangerfjord*, *Hardanger* and *Korsfjord*. The former is built at Norwegian Aker Yards, and the two latter are built at Remontowa in Poland.

2.4 Ferry types

The Norwegian ferries are divided into two main classes: speedboats which only carry passengers and no vehicles, and car ferries. These ships are either monohulls or catamarans. Figure 2.3a shows a catamaran speedboat, and Figure 2.3b shows a monohull car ferry. While some ferries have traditional propulsion systems with fixed propellers and rudders, the vast majority are nearly aft-bow symmetrical and have one azimuth thruster in each end [15, 16]. This configuration allows the ferry to cross and dock without turning around, and cars to drive easily in and out of the ferry without reversing. It also gives full actuation during docking. Even ferries with fixed propellers and rudders usually have either a tunnel thruster or a retractable azimuth bow thruster.

[1]The map is available for interactive viewing at `https://batchgeo.com/map/9a1b1d6420bf71875a0312f8cf72b170`.

(a) The speedboat *Kistefjell*, a catamaran. The boat operates the route Tromsø–Finnsnes–Harstad, and takes 250 passengers. © 2017 Skipsrevyen.

(b) The car ferry *Oppedal*, a monohull. The ferry operates the route Lavik–Oppedal, and takes 120 PCEs and 350 passengers. © 2017 Skipsrevyen.

Figure 2.3: Images of a catamaran and a monohull.

Table 2.1: Ships operating the Horten–Moss connection [18, 19].

Name	Year	Passengers	PCEs	Speed kn	Length m	Beam m	Draft m
Bastø I	1996	600	200	17	109.0	18.5	4.67
Bastø II	1996	600	200	17	109.0	18.5	4.67
Bastø III[2]	2005	600	212	18.5	116.2	19.5	4.85
Bastø IV	2016	600	200	18	142.9	21	-
Bastø V	2016	600	200	18	142.9	21	-
Bastø VI	2016	600	200	18	142.9	21	-

2.5 Selected routes

In this section, three selected routes will be presented, together with their passenger statistics and ferry types. Figure 2.2 shows a map of most of the ferry connections in Norway.

2.5.1 Horten–Moss

The route from Horten in Vestfold to Moss in Østfold is operated by Bastø Fosen, a Torghatten subsidiary, and is the most trafficked ferry connection in Norway. In 2016, it transported 1.74 million vehicles (including large trucks) and 3.35 million passengers (including drivers) [17].

Figure 2.4 shows a map of the ferry route and its location. The connection has transported cars since 1934 and is now a part of national road 19 [18]. The connection is approximately 11 km, takes 30 minutes to complete, and saves a car trip of 130 km and 2 hours. The fare is 178 NOK (p.t.) one way for a passenger car [19].

The Horten–Moss connection is operated by six ferries per 2017. Table 2.1 lists the active ferries.

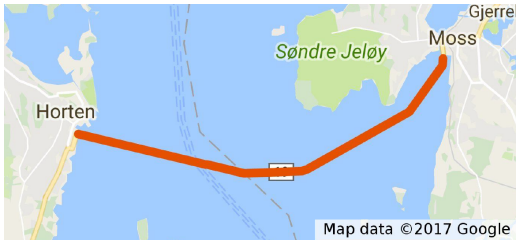
2.5.2 Hareid–Sulesund

The route from Hareid to Sulesund in Møre og Romsdal is operated by Norled. Figure 2.5b shows where the connection is located, and Figures 2.5c and 2.5d show detailed views. The connection is part of county road 19, and connects Hareid to Sula, which is close to Ålesund and Vigra airport.

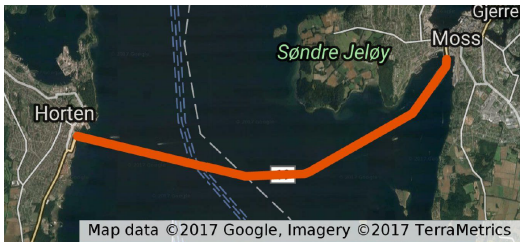
[2] *Bastø III* will serve as backup from early 2017, when Bastø VI is commissioned. Until then, Bastø VIII is the backup.

(a) *Bastø VI* entering Horten, for trial, newly delivered 2016-12-16. Ulf Larsen.

(b) Overview map of the Horten–Moss route.



(c) Map.



(d) Satellite image.

Figure 2.4: Horten–Moss.

The route completes in 20 minutes, is 8 km long, and saves a car trip of 180 km and almost 4 hours. The fare is 115 NOK (p.t.) for passenger cars [20]. It is served by three ships, seen in Table 2.2 [20]. *Ullensvang* is pictured in Figure 2.5a.

2.5.3 Fjone–Sundsodden (lake)

The Nisser lake in Telemark has Norway's only remaining cable ferry, which takes passengers, bikes, and cars across from Fjone to Sundsodden. The connection is part of county road 354 and is run by Nissedal municipality. The fare is 60 NOK (p.t.) for passenger cars [21]. It is 400 m long, and transports 5000 passenger car

Table 2.2: Ships operating the Hareid–Sulesund connection [18].

Name	Year	Passengers	PCEs	Speed kn	Length m	Beam m	Draft m
Ullensvang	1986	500	110	16	87.0	14.5	4.30
Tidefjord	2008	350	120	13	114.0	16.8	3.36
Høgsfjord	1992	299	76	14	84.7	15.5	3.36

(a) *Ullensvang*, a ferry operating the Hareid–Sulesund connection. CC BY: Harald Sætre / Wikipedia.

(b) Overview map of the Hareid–Sulesund route.

(c) Map.

(d) Satellite image.

Figure 2.5: Hareid–Sulesund.

Table 2.3: The cable ferry operating the Fjone–Sundsodden connection [18].

Name	Year	Passengers	PCEs	Speed kn	Length m	Beam m	Draft m
Fjoneferga *Nissen*	1976	15	3	5	15	-	-

equivalents (PCEs) per year, in addition to passengers and bicycles [22].

2.6 Regulations related to ferry operations

The main laws and regulations governing ferry operations in Norway are *Forskrift om transport med ferje* (Regulations for transport by ferry) [23], *Sjøveisreglene* (Rules of the road at sea) [24], *Sjøloven* (Maritime code) [25], *Forskrift om skipsførerens og rederiets plikter* (Regulations for shipmaster's and shipowner's duties) [26], and *Havne- og farvannsloven* (Harbor act) [27]. In addition, Statens vegvesen (Norwegian Public Roads Administration) (NPRA) has some manuals which describe best practices on how to design ferry ports [10, 28].

(a) *Nissen*, the cable ferry operating the Fjone–Sundsodden connection.

(b) Overview map of the Fjone–Sundsodden route.

(c) Map.

(d) Satellite image.

Figure 2.6: Fjone–Sundsodden.

2.6.1 Regulations for transport by ferry

The Regulations for transport by ferry considers the responsibilities of the driver of a vehicle on board a ferry, such as their duty to follow instructions given by signboards and ferry crew, and operate their vehicle properly during transit [23]. They also regulate preferential rights of emergency vehicles and other important officials to board the ferry.

There are few implications from this law on an autonomous ferry system. All of the items may be fulfilled by proper signposting and ticketing systems.

2.6.2 Rules of the road at sea

The Rules of the road at sea provide regulations for collision avoidance, rules for placement and operation of lanterns and signals, and special rules for Norwegian domestic waters [24]. These rules comply with the COLREGS [29]. Section 2.7 contains an introduction to these regulations.

The Rules of the road at sea have major implications on an autonomous ferry system, as any seafaring vessel has to behave as expected by other vessels, not to

mention that the highest priority of any vessel is to not collide with anything.

2.6.3 Maritime code

The Maritime code is a comprehensive set of laws governing shipping and transport by sea [25]. It also governs ship building, responsibilities, shipwrecks, salvaging, and disputes.

Chapter 6 of the Maritime code is about the shipmaster's duties to ensure seaworthiness, safe navigation, loading and handling of the ship in the case of distress. While the safety of these operations may be handled by well-constructed automatic systems, the legislation requires the presence of a shipmaster. This may be circumvented by exemptions for testing purposes, but for large-scale implementations, this requires a change in legislation.

2.6.4 Regulations for shipmaster's and shipowner's duties

These regulations are used when there is suspicion that an offense of a serious nature is committed on board a ship [26]. They give the shipmaster rights and duties to investigate, interrogate and arrest suspected offenders.

2.6.5 Harbor act

The Harbor act facilitates maneuverability and management of waters in accordance with the public interest, fisheries, and other industries [27]. It mostly gives the Ministry of Transport and Communications the right to regulate and delegate matters of traffic, hazards, navigation, and harbor facilities. This implies that an autonomous system must follow regulations given by local governments about navigation and use of harbors.

2.7 COLREGS

The International Regulations for Preventing Collisions at Sea (COLREGS) are a set of international regulations that all surface-going ocean vehicles must abide by to prevent collisions at sea [29]. The most recent version of the regulations were presented in 1972, and are maintained by the International Maritime Organization (IMO). The COLREGS consist of five parts:

Part A — General: Covers responsibilities and when the rules apply. *Rules 1–3.*

Part B — Steering and Sailing: Covers general conducts of vessels and special situations. *Rules 4–19.*

Section I — Conduct of vessels in any condition of visibility: General regulations regarding speed, radar use etc. *Rules 4–10.*

Section II — Conduct of vessels in sight of one another: The section most relevant to this thesis, regarding head-on, crossing and overtaking scenarios. *Rules 11–18.*

Section III — Conduct of vessels in restricted visibility: Contains a rule regarding operation in scenarios with restricted visibility. *Rule 19.*

Part C — Lights and Shapes: Defines the proper use and placement of lights, flags and lanterns. *Rules 20–31.*

Part D — Sound and Light Signals: Defines the proper use of sounds and light signals. *Rules 32–37.*

Part E — Exemptions: Specifies that some ships (from before the regulations were active) are exempted from some of the signal and light requirements.

To this thesis, the most relevant parts of the COLREGS appear in Part B, and inside that part, Section II provides the rules which handle collision avoidance in the situations where other ships are nearby. An introduction to the relevant rules, and its implications on autonomous operations is given here:

Rule 8 Action to avoid collision

(a) *Any action taken to avoid collision shall be taken in accordance with the Rules of this Part and shall, if the circumstances of the case admit, be positive, made in ample time and with due regard to the observance of good seamanship.*

(b) *Any alteration of course and/or speed to avoid collision shall, if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.*

These two items imply that any alteration of speed or course should be *positive* and clearly observable for other vessels. This may be interpreted as a constraint or priority for the motion control system.

Rule 9 Narrow channels

(a) *A vessel proceeding along the course of a narrow channel or fairway shall keep as near to the outer limit of the channel or fairway which lies on her starboard side as is safe and practicable.*

This implies that the path planning algorithm should optimally plan passages in narrow seaways to keep near the starboard side.

Rule 13 Overtaking

(a) *Notwithstanding anything contained in the Rules of Part B, Sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken.*

(b) *A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than* 22.5 *degrees abaft her beam, that is, in such a position with reference to the vessel she is overtaking, that at night she would be able to see only the stern light of that vessel but neither of her sidelights.*

(c) *When a vessel is in any doubt as to whether she is overtaking another, she shall assume that this is the case and act accordingly.*

(d) *Any subsequent alteration of the bearing between the two vessels shall not make the overtaking vessel a crossing vessel within the meaning of these Rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear.*

The implications of this rule are covered in Section 2.7.1.

Rule 14 Head-on situation

(a) *When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.*

(b) *Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she would see the mast head lights of the other in a line or nearly in a line and or both sidelights and by day she observes the corresponding aspect of the other vessel.*

(c) *When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.*

The implications of this rule are covered in Section 2.7.2.

Rule 15 Crossing situation

When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

The implications of this rule are covered in Section 2.7.3.

Rule 16 Action by give-way vessel

Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

This means that when one of the situations described in rules 13 to 2.9 is encountered, the give-way vessel should react early.

Rule 17 Action by stand-on vessel

(a) (i) *Where one of two vessels is to keep out of the way the other shall keep her course and speed.*

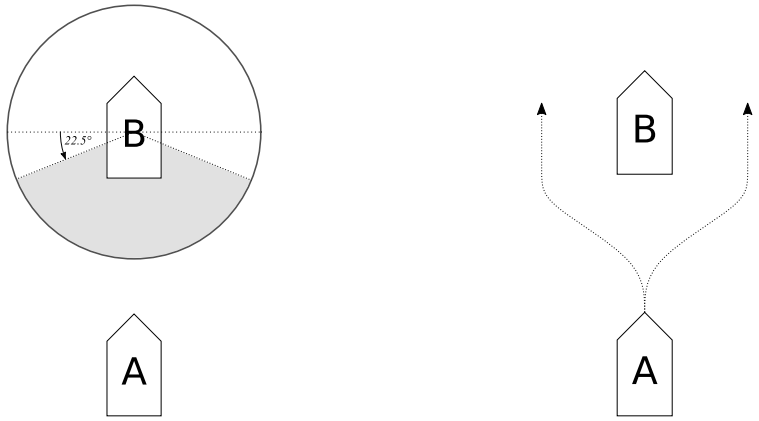
(ii) *The latter vessel may however take action to avoid collision by her manoeuvre alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules.*

(b) *When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid collision.*

(c) *A power-driven vessel which takes action in a crossing situation in accordance with sub-paragraph (a)(ii) of this Rule to avoid collision with another power-driven vessel shall, if the circumstances of the case admit, not alter course to port for a vessel on her own port side.*

(d) *This Rule does not relieve the give-way vessel of her obligation to keep out of the way.*

This means, that unless the give-way ship does not react to a situation, the stand-on vessel should do what it can to avoid collisions. For the collision avoidance system, this means that even if the ship has right of way, barriers must be implemented to avoid collisions caused by other vessels.



(a) Illustration of the overtaking situation. The gray segment of ship (B) is where an approaching ship (A) is considered to be overtaking.

(b) An overtaking ship may pass at either side, but has to stay out of the way, and may not initiate a crossing situation.

Figure 2.7: Overtaking situation.

2.7.1 Overtaking

Rule 13 deals with the overtaking situation. This rule says that the overtaking situation is apparent when the approaching vessel is within the 135° wide segment behind the ship being overtaken. This situation is illustrated in Figure 2.7a. The only guidance to how the overtaking vessel ((A) in this situation) should conduct itself, is to keep out of the way, and to not initiate a crossing situation. Figure 2.7b illustrates a possible solution.

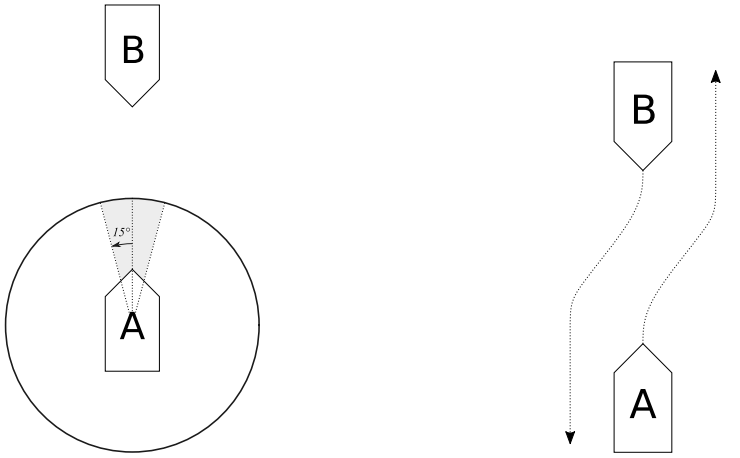
2.7.2 Head-on situation

The head-on situation is defined in Rule 14. The situation is said to exist when "a vessel sees the other ahead or nearly ahead and by night ..." Some publications set actual figures on this, and the figure used in [30, 31, 32, 33] is a 30° arc centered around the vessel's x axis (the heading).

Rule 14 says that both vehicles shall turn starboard to avoid a collision. The situation is illustrated in Figure 2.8a, and Figure 2.8b shows the solution.

2.7.3 Crossing situation

The only guidance Rule 15 gives when a crossing situation is apparent, is that "the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the



(a) Illustration of the head-on situation. The gray segment of ship (A) shows where an approaching ship gets within the head-on situation.

(b) In a head-on situation, each ship is to turn towards starboard, as per the COLREGS.

Figure 2.8: Head-on situation.

other vessel." Figure 2.9 shows this situation, and a possible solution.

2.7.4 Summary of the COLREGS

These regulations are made for ships controlled by humans, and must also be followed by autonomous ships, since an autonomous ship will not be alone in the sea. A ferry shuttling a fjord is likely to encounter leisure vessels, fishing boats, sailboats, tankers and cruise ships in addition to smaller obstacles such as kayakers or divers. To be able to operate in a public area an autonomous ferry needs to be able to prevent any dangerous situation, including those described by the COLREGS. Solutions to the most common situations such as overtaking, head-on and crossing must be implemented by a collision avoidance system. More information about such systems is found in Chapter 7.

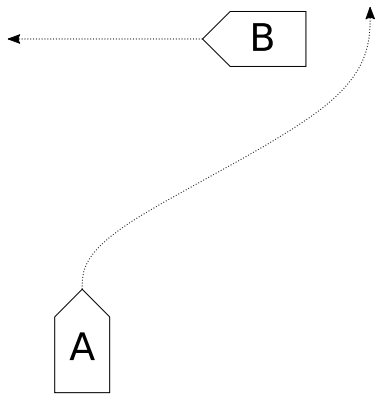


Figure 2.9: Crossing situation. Ship (A) gets the crossing vessel on its starboard side, and is required to steer out of its way — preferably behind it. Ship (B) has the other vessel on its port side, and continues ahead.

Chapter 3

Ship Modeling and Low-Level Control

This chapter contains theoretical background information that is necessary to understand in order to comprehend the rest of the thesis. A review of ship modeling is presented, including reference frames, kinematics and kinetic differential equations. Additionally, some examples of thruster configurations are introduced, along with their equations. The concepts of workspaces and configuration spaces are subsequently covered. At the end of the chapter, there is a section on low-level control of speed and heading, including a short review of control allocation algorithms.

3.1 Ship modeling

Mathematical models are used to describe how a physical system reacts to external forces, study stability properties under given operation conditions, and to design controllers that are able to move the system states to desired values. Ship models vary in complexity, and the required level of complexity depends on what is to be achieved with the model. Different models are used in seakeeping and maneuvering, and one may use varying number of degrees of freedom in either case. For the purposes of this thesis, where the motion control of surface vessels is studied, 3 degree of freedom (DOF) models (surge, sway and yaw) are considered sufficient.

A great resource on vessel modeling is Fossen's book [34], where most of the content of this chapter is based upon.

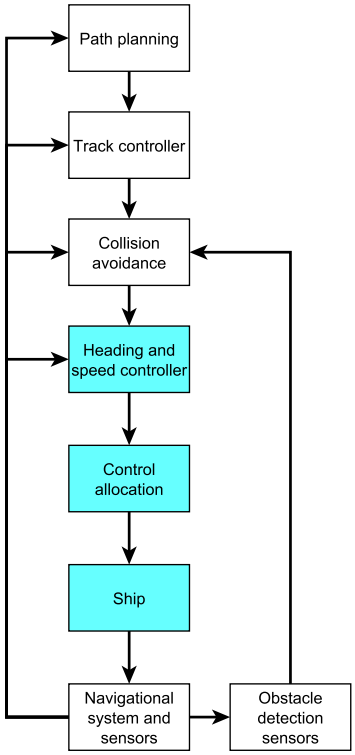


Figure 3.1: Block diagram of a motion control system. The highlighted blocks are relevant to the discussion of ship modeling and low-level control.

3.1.1 Displacement, semi-displacement and planing vessels

Faltinsen [35] and Fossen [34] classify ships in three groups, with respect to how much of the forces that keep the ship afloat come from buoyancy and hydrodynamics. The *Froude number* is used:

$$Fn = \frac{U_r}{\sqrt{gL}} \tag{3.1}$$

Here U_r is the maximum relative speed, L is the submerged length of the vessel, and g is the acceleration of gravity. The classes are divided as such:

Displacement vessels ($Fn < 0.4$): The buoyancy force dominates the hydrodynamic forces.

Semi-displacement vessel (0.4–$0.5 < Fn < 1.0$–1.2): The buoyancy force is not dominant.

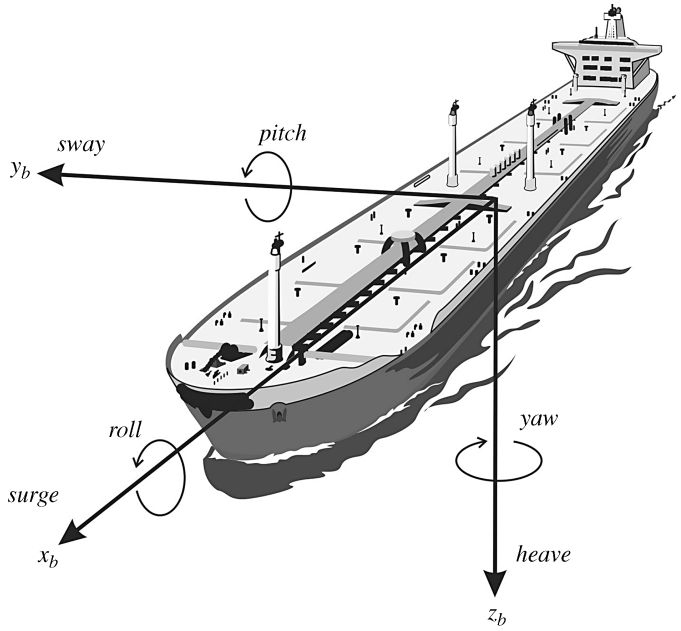


Figure 3.2: Illustration of motion in 6 DOF. From [34]. © 2011 John Wiley & Sons, Ltd.

Planing vessel ($Fn > 1.0$–1.2): The hydrodynamic forces dominate the buoyancy force.

All car ferries are classified as displacement vessels, and this will be the only form of vessel modeled and discussed in this thesis.

3.1.2 Notation, reference frames and transformations

Before going into the differential equations, it is necessary to be familiar with the notation and reference frames used.

Notation

Before going into reference frames and transformations, it is useful to have a clear sense of the notation used throughout the thesis. The notation will be the same as the one used in [34], and is listed in Table 3.1. Figure 3.2 illustrates the DOFs used in ship modeling.

It is useful to collect the forces, velocities and positions in these vectors: $\boldsymbol{\tau} = [X, Y, Z, K, M, N]^\top$, $\boldsymbol{\nu} = [u, v, w, p, q, r]^\top$, and $\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^\top$. The linear and angular velocities are useful to collect in the following vectors, respectively: $\boldsymbol{v} = [u, v, w]^\top$ and $\boldsymbol{\omega} = [p, q, r]^\top$.

Some important angles are introduced:

Table 3.1: Notation used for forces, positions and velocities. Retrieved from [34].

	Forces and moments	Linear and angular velocities	Positions and Euler angles
motions in the x direction (surge)	X	u	x
motions in the y direction (sway)	Y	v	y
motions in the z direction (heave)	Z	w	z
rotations about the x axis (roll)	K	p	ϕ
rotations about the y axis (pitch)	M	q	θ
rotations about the z axis (yaw)	N	r	ψ

Heading ψ The angle between the ship's longitudinal axis (from aft to fore) and true north.

Course χ The angle of the velocity vector of the ship (the actual direction of travel) relative to true north.

Sideslip β The angle between the ship's heading and velocity.

These angles are illustrated in Figure 3.3. From this description, we see that the angles satisfy

$$\chi = \psi + \beta \tag{3.2}$$

and

$$\beta = \sin^{-1}\left(\frac{v}{U}\right), \tag{3.3}$$

where $U = \sqrt{u^2 + v^2}$. In the presence of ocean currents, the sideslip angle can be extended to

$$\beta_r = \sin^{-1}\left(\frac{v_r}{U_r}\right), \tag{3.4}$$

where $U_r = \sqrt{(u - u_c)^2 + (v - v_c)^2}$.

The following notation is adapted for the velocities:

$\boldsymbol{v}^c_{a/b}$ = linear velocity of $\{a\}$ with respect to the frame $\{b\}$ expressed in $\{c\}$

$\boldsymbol{\omega}^c_{a/b}$ = angular velocity of $\{a\}$ with respect to $\{b\}$ expressed in $\{c\}$

When describing velocities and rotation matrices, the following notation is used:

$$\boldsymbol{v}^{\text{to}} = \mathbf{R}^{\text{to}}_{\text{from}} \boldsymbol{v}^{\text{from}} \tag{3.5}$$

Here the texts "to" and "from" describe the reference frames rotated to and from.

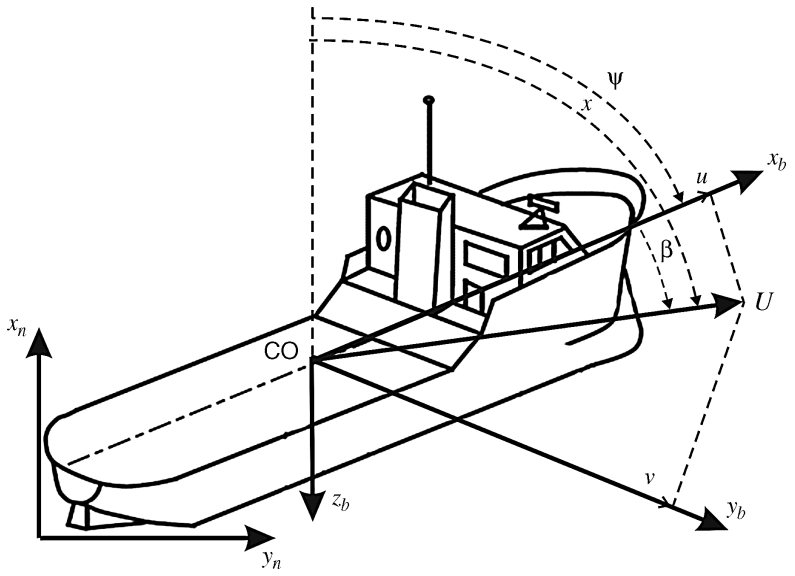


Figure 3.3: Illustration of the geometric relationship between heading, course and sideslip. From [34]. © 2011 John Wiley & Sons, Ltd.

Reference frames

The frames most important to this thesis are presented here:

NED The north-east-down (NED) coordinate system is "usually defined as the tangent plane on the surface of the Earth moving with the craft, but with axes pointing in different directions than the body-fixed axes of the craft [34]." The coordinate system is denoted $\{n\} = (x_n, y_n, z_n)$ with origin o_n. The x axis points towards true north, y points towards east, while the z axis points downwards, forming a right-handed coordinate system. This reference frame is sufficient to use when operating in a local area, and a flat earth is approximated. For the purposes encountered in this thesis, the frame may be considered *inertial*.

BODY The body-fixed reference frame $\{b\} = (x_b, y_b, z_b)$, o_b is fixed to the vessel. The x axis points forward, y points starboard, and z points down, forming a right-handed coordinate system. The origin is placed on a line from aft to fore on the ship, midships.

FLOW This reference frame is convenient to use when calculating hydrodynamic forces, e.g. drag. The FLOW reference frame's x axis is rotated from the BODY frame, and points into the relative flow. The y axis follows the rotation, while the z axis still points downwards.

Transformations

The transformations between these reference frames are needed to create the models that are used in the thesis. The transformations are used to describe velocities in the different frames, and to create the kinematic differential equations.

The transformation between the BODY and NED frames are described by the rotation matrix

$$\overbrace{\boldsymbol{v}_{b/n}^{n}}^{\substack{\text{velocity of BODY}\\ \text{relative to NED,}\\ \text{represented in NED}}} = \underbrace{\mathbf{R}_{b}^{n}(\boldsymbol{\Theta}_{nb})}_{\substack{\text{rotation matrix}\\ \text{from BODY to NED}}} \overbrace{\boldsymbol{v}_{b/n}^{b}}^{\substack{\text{velocity of BODY}\\ \text{relative to NED,}\\ \text{represented in BODY}}} . \tag{3.6}$$

Here the vector $\boldsymbol{\Theta}_{nb} = [\phi, \theta, \psi]^{\top}$ denotes the Euler angles (ZYX rotation order) that describe the ship attitude. The rotation matrix $\mathbf{R}_{b}^{n}$ is

$$\mathbf{R}_{b}^{n} = \mathbf{R}_{z,\psi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\phi} \tag{3.7}$$

where each of the three matrices on the right-hand side are principal rotation matrices.

The relationship of the relative FLOW and BODY frame is described by

$$\boldsymbol{v}^{\text{FLOW}} = \mathbf{R}_{z,-\beta}\boldsymbol{v}^{\text{BODY}}. \tag{3.8}$$

In 3 DOF, the kinematics (relationship between the body velocity and the position) are described by the differential equation

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{3.9}$$

Here the position vector is $\boldsymbol{\eta} = [x, y, \psi]^{\top}$, and the velocity vector is $\boldsymbol{\nu} = [u, v, r]^{\top}$. The transformation matrix $\mathbf{R}$ is

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

Ocean currents

In addition to the ground velocities, it is useful to define the velocities relative to the ocean currents. Ocean current velocity relative to ground may be described in the NED frame as

$$\boldsymbol{v}_{c}^{n} = \begin{bmatrix} V_x, V_y, 0 \end{bmatrix}^{\top}. \tag{3.11}$$

Here the current is assumed horizontal and irrotational, which is a reasonable assumption for our purposes. In the body frame the current velocity is expressed as

$$\boldsymbol{v}_c^b = [u_c, v_c, 0]^\top = \mathbf{R}^\top \boldsymbol{v}_c^n \tag{3.12}$$

Compunded into a 6-element vector we get

$$\boldsymbol{\nu}_c = [u_c, v_c, 0, 0, 0, 0]^\top \tag{3.13}$$

It is also useful to define the ship body velocity relative to ocean current:

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c = [u - u_c, v - v_c, w, p, q, r]^\top = [u_r, v_r, w, \ldots]^\top \tag{3.14}$$

3.1.3 Kinetics in 3 DOF

The kinetics of the 3 DOF model will have the following form:

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \tag{3.15}$$

where

$$\mathbf{M} = \mathbf{M}_A + \mathbf{M}_{RB} \tag{3.16}$$

$$\mathbf{C}(\boldsymbol{\nu}_r) = \mathbf{C}_A(\boldsymbol{\nu}_r) + \mathbf{C}_{RB}(\boldsymbol{\nu}_r). \tag{3.17}$$

The subscripts A and RB denote effects from *added mass* and *rigid body*, respectively (see [34, (3.57)]). The matrix $\mathbf{C}$ represents the Coriolis and centripetal forces, both from the rigid body and the added mass effects. The model (3.15) requires the matrix $\mathbf{C}$ to be constructed independent of the linear velocities. The hydrostatic forces from drag are represented by the term $\mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r$. The vector $\boldsymbol{\tau}$ is the virtual control vector, and consists of the actuated forces and moments from thrusters and rudders.

Center of gravity

The vector $\boldsymbol{r}_g = \begin{bmatrix} x_g, y_g, z_g \end{bmatrix}$ is the distance from the control origin o_b to the center of gravity. In the 3 DOF models, $z_g = 0$, and we assume that o_b is placed along the aft-fore longitudinal axis, such that we get starboard-port symmetry ($y_g = 0$).

Inertia matrix M

The rigid body inertia matrix $\mathbf{M}_{RB} = \mathbf{M}_{RB}^\top > 0$ has the form

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \tag{3.18}$$

in 3 DOF. The total ship mass is m, and x_g is the longitudinal distance of the body origin o_b to the center of gravity. The moment of inertia around the body z axis is I_z.

The added mass matrix $\mathbf{M}_A = \mathbf{M}_A^\top$ has the form

$$\mathbf{M}_A = -\begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & Y_{\dot{r}} & N_{\dot{r}} \end{bmatrix} \quad (N_{\dot{v}} = Y_{\dot{r}}). \tag{3.19}$$

These are the added mass derivatives. They are found by experiment or by advanced modeling[1].

Together they become the system inertia matrix $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$.

Inertia matrix $\mathbf{I}_g$ and $\mathbf{I}_b$

The inertia matrix $\mathbf{I}_g$ around the center of gravity has the following form:

$$\mathbf{I}_g = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{xy} & I_y & -I_{yz} \\ -I_{xz} & -I_{yz} & I_z \end{bmatrix}, \quad \mathbf{I}_g = \mathbf{I}_g^\top > 0. \tag{3.20}$$

All elements of this matrix are 0 in the 3 DOF case, except for I_z, which has the value

$$I_z = \int_V (x^2 + y^2)\rho_m \, \mathrm{d}V. \tag{3.21}$$

The inertia matrix when rotating about the BODY origin is

$$\mathbf{I}_b = \mathbf{I}_g - m\mathbf{S}^2(\boldsymbol{r}_g^b) \tag{3.22}$$

The skew-symmetric matrix $\mathbf{S}(\cdot)$ is defined by

$$\mathbf{S}(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.23}$$

where $\boldsymbol{a} = [a_1, a_2, a_3]^\top$.

[1] Finding added mass parameters are not covered in this thesis, but [34] has extensive information about the subject. Other useful sources for finding such parameters are [36, 37], where strip theory is used to retrieve the added mass parameters.

Coriolis and centripetal matrix

The rigid body Coriolis and centripetal matrix $\mathbf{C}_{RB}(\boldsymbol{\nu}) = -\mathbf{C}_{RB}^{\top}(\boldsymbol{\nu})$ can be represented with angular velocities only, which is useful when creating a model using the relative velocity $\boldsymbol{\nu}_r$. The 6 DOF matrix is retrieved from [34, (3.57)]:

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} m\mathbf{S}(\boldsymbol{\omega}) & -m\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\boldsymbol{r}_g^b) \\ m\mathbf{S}(\boldsymbol{r}_g^b)\mathbf{S}(\boldsymbol{\omega}) & -\mathbf{S}(\mathbf{I}_b\boldsymbol{\omega}) \end{bmatrix} \tag{3.24}$$

When placing the o_b so that we get starboard-port symmetry (in the middle of the ship along the y axis) in 3 DOF the Coriolis-centripetal matrix becomes

$$\mathbf{C}_{RB} = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix}. \tag{3.25}$$

The added mass Coriolis and centripetal matrix $\mathbf{C}_A(\boldsymbol{\nu}_r) = -\mathbf{C}_A^{\top}(\boldsymbol{\nu}_r)$ is

$$\mathbf{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v_r + Y_{\dot{r}} r \\ 0 & 0 & -X_{\dot{u}} u_r \\ -Y_{\dot{v}} v_r - Y_{\dot{r}} r & X_{\dot{u}} u_r & 0 \end{bmatrix}. \tag{3.26}$$

Damping matrix

The damping model used is that of [34, Section 6.5], amended with cubic surge damping and Munk-moment compensation, as described in [38]. It has the form

$$\mathbf{D}(\boldsymbol{\nu}_r) = \mathbf{D}_L + \mathbf{D}_{NL}(\boldsymbol{\nu}_r) \tag{3.27}$$

where

$$\mathbf{D}_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \tag{3.28}$$

and

$$\mathbf{D}_{NL}(\boldsymbol{\nu}_r) = \begin{bmatrix} d_{NL,11}(\boldsymbol{\nu}_r) & 0 & 0 \\ 0 & d_{NL,22}(\boldsymbol{\nu}_r) & d_{NL,23}(\boldsymbol{\nu}_r) \\ 0 & d_{NL,32}(\boldsymbol{\nu}_r) & d_{NL,33}(\boldsymbol{\nu}_r) \end{bmatrix}. \tag{3.29}$$

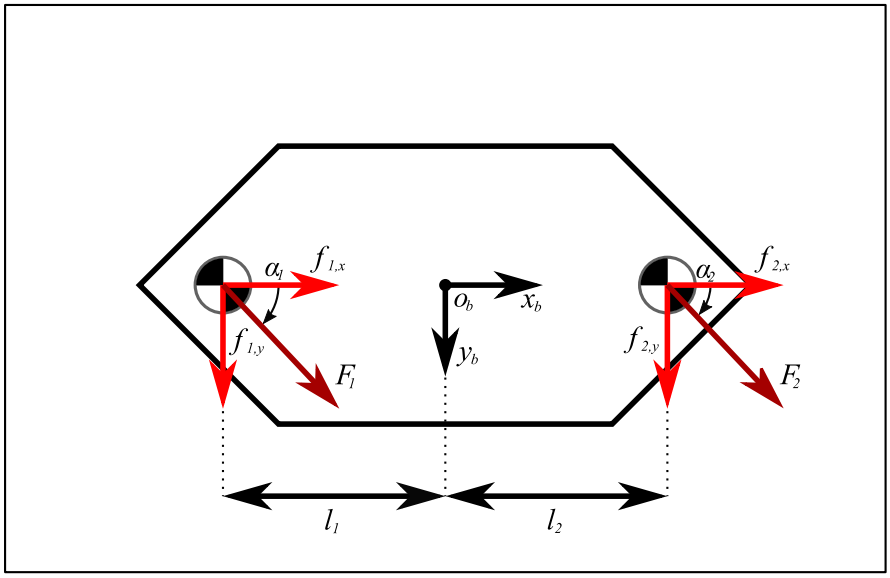


Figure 3.4: Illustration of symmetric actuator layout for ferries.

The linear part of the damping is $\mathbf{D}_L$, and $\mathbf{D}_{NL}$ is the nonlinear part. Its components are

$$d_{NL,11}(\boldsymbol{\nu}_r) = -X_{|u|u}|u_r| - X_{u^3}u_r^2 \tag{3.30a}$$
$$d_{NL,22}(\boldsymbol{\nu}_r) = -Y_{|v|v}|v_r| - Y_{|r|v}|r| \tag{3.30b}$$
$$d_{NL,23}(\boldsymbol{\nu}_r) = -Y_{|v|r}|v_r| - Y_{|r|r}|r| - Y_{ur}u_r \tag{3.30c}$$
$$d_{NL,32}(\boldsymbol{\nu}_r) = -N_{|v|v}|v_r| - N_{|r|v}|r| - N_{uv}u_r \tag{3.30d}$$
$$d_{NL,33}(\boldsymbol{\nu}_r) = -N_{|v|r}|v_r| - N_{|r|r}|r| - N_{ur}u \tag{3.30e}$$

Irrotational constant ocean currents

According to Property 8.1 in [34] the rigid body kinetics satisfies

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} = \mathbf{M}_{RB}\dot{\boldsymbol{\nu}}_r + \mathbf{C}_{RB}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r \tag{3.31}$$

This is used to retrieve the form of (3.15) for the kinetics, and to not keep track of both $\boldsymbol{\nu}$ and $\boldsymbol{\nu}_r$ for simulation purposes.

3.1.4 Control forces and actuation

Common for almost every new ferry built for operation in Norway is the thruster configuration where an azimuth thruster is placed in each end of the ship [16]. This is illustrated in Figure 3.4. This configuration allows for high maneuverability

while docking and undocking. Additionally, a thruster in each end means the ferry does not have to rotate, and that cars can drive off without backing up.

While two azimuth thrusters allow full actuation (the ship will in fact be over-actuated) in surge, sway and yaw during docking, the sway forces are heavily reduced during high speeds, such that the ship becomes underactuated during transit. During transit the aft thruster has the most authority over yaw moment, and is used as the main propulsion, while the fore thruster either turns freely or is used as auxiliary propulsion. Controllable pitch propellers are able to change the pitch so that the fore thruster may turn freely in an efficient manner while not being used. Fixed pitch propellers contribute a relatively high drag force when turning freely, so it is more efficient to let it provide 30–40 % of the propulsion force [16].

Thruster model for docking

These details are abstracted away in the ship model by creating a separate thruster model for docking and transit. The docking model will have a control vector $\boldsymbol{u}$ of dimension 4, where $\boldsymbol{u} = [\boldsymbol{F}^\top, \boldsymbol{\alpha}^\top]^\top = [F_1, F_2, \alpha_1, \alpha_2]^\top$. The input to each actuator is a force F_i and an angle α_i, for $i \in \{1, 2\}$. This requires a mapping from the control vector $\boldsymbol{u}$ to decomposed forces and moment $\boldsymbol{\tau} = [X, Y, N]$:

$$\boldsymbol{\tau}(\boldsymbol{u}) = \mathbf{T}(\boldsymbol{\alpha})\boldsymbol{F}, \tag{3.32}$$

where

$$\mathbf{T}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 \\ \sin\alpha_1 & \sin\alpha_2 \\ -l_1\sin\alpha_1 & l_2\sin\alpha_2 \end{bmatrix}. \tag{3.33}$$

Here l_i denotes the length from the origin o_b to the azimuth thruster i.

For allocation purposes it may be favorable to represent the forces generated by each thruster in decomposed form:

$$\boldsymbol{f} = \begin{bmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{bmatrix}. \tag{3.34}$$

The relationship between $\boldsymbol{f}$ and $\boldsymbol{u}$ is then

$$\alpha_i = \operatorname{atan2}(f_{i,y}, f_{i,x}) \tag{3.35a}$$

$$F_i = \left\| [f_{i,x}, f_{i,y}]^\top \right\|_2 \tag{3.35b}$$

$$i \in \{1, 2\}.$$

With this representation the force vector $\boldsymbol{\tau}$ becomes

$$\boldsymbol{\tau} = \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -l_1 & 0 & l_2 \end{bmatrix}}_{\mathbf{T}} \boldsymbol{f}. \tag{3.36}$$

Thruster model for transit

The transit model will have a control vector of dimension 2, where $\boldsymbol{u} = [F, \delta]^\top$, F is the force magnitude of the aft azimuth thruster, and δ is its angle. The force vector $\boldsymbol{\tau}$ will in this case become

$$\boldsymbol{\tau} = \begin{bmatrix} F\cos\delta \\ F\sin\delta \\ -l_1 F\sin\delta \end{bmatrix}. \tag{3.37}$$

This is only valid for small angles δ, as the cross flow will reduce the efficiency of lateral forces. As δ is small in a transit operation, a simplification of (3.37) useful for control design is

$$\boldsymbol{\tau} = \begin{bmatrix} F \\ F\delta \\ -l_1 F\delta \end{bmatrix}. \tag{3.38}$$

3.1.5 3 DOF model summary

For completeness, the ship-model kinematics and kinetics are collected here:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\overbrace{(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c)}^{\boldsymbol{\nu}} \tag{3.39a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}}, \tag{3.39b}$$

where

$$\begin{aligned} \boldsymbol{\eta} &= [x, y, \psi]^\top, \\ \boldsymbol{\nu}_r &= [u_r, v_r, r]^\top, \text{ and} \\ \boldsymbol{\tau} &= [X, Y, N]^\top. \end{aligned}$$

3.2 Workspace, configuration space and actuation

When designing control systems, it is important to understand the difference between the workspace and the configuration space. The control objective is defined in the workspace, and the pose of the vehicle exists in the configuration space. From Fossen's work in [34] we have the following definitions:

Definition 3.1 (Configuration space)**.** The n-dimensional configuration space is the space of possible positions and orientations that a craft may attain, possibly subject to external constraints.

In our 3 DOF example for maneuvering, the positions x and y, and the heading ψ define the three-dimensional configuration space

$$\mathcal{C} = \mathbb{R}^2 \times \mathrm{S}\,. \tag{3.40}$$

Definition 3.2 (Workspace)**.** The workspace is a reduced space of dimension $m \leq n$ in which the control objective is defined.[2]

The workspace is denoted $\mathcal{W}$. When controlling heading, the control objective is only ψ, and the workspace has dimension 1.

Using Definition 3.1, we define when a ship is fully actuated or underactuated. When the number of control inputs available is r, we get the following definitions:

Definition 3.3 (Underactuated marine craft)**.** A marine craft is underactuated if it has fewer control inputs than generalized coordinates ($r < n$).

Definition 3.4 (Fully actuated marine craft)**.** A marine craft is fully actuated if it has equal or more control inputs than generalized coordinates ($r \geq n$).

A ship with a rudder and propellers has two control inputs ($r = 2$), while a ship (at low speeds) with two azimuth thrusters has $r = 4$ — two angles and two magnitudes of thruster force. An azimuth thruster's control inputs may also be seen as a force decomposed in the x and y directions.

Another source with useful information on the operating spaces is the book of Spong et al. [39]. Section 1.1 treats the basics of the operating spaces, and Section 5.1 treats the configuration space with respect to path and trajectory planning.

[2] In [34], the reduced space has dimension $m < n$, but an example where the dimension of the workspace equals the configuration space is a 3 DOF DP operation, where we may wish to control both x, y and ψ. Here $m = n = 3$, hence $m \leq n$.

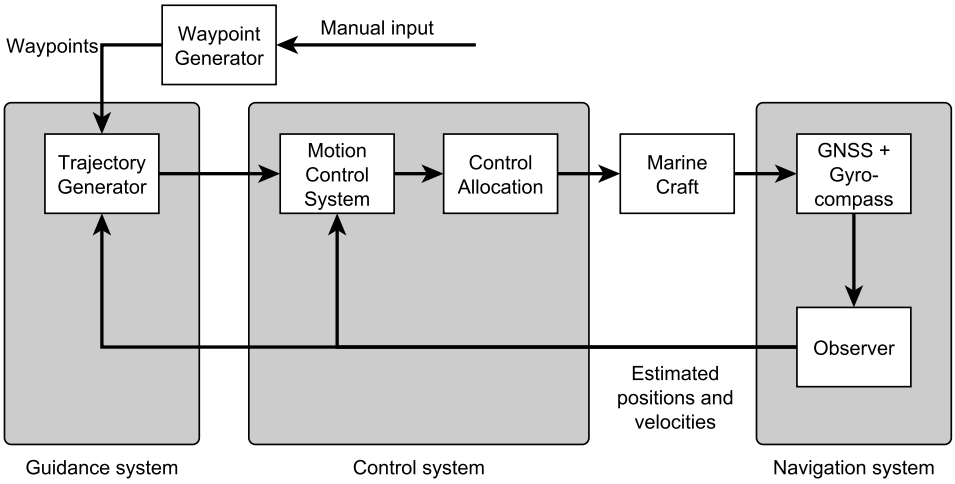


Figure 3.5: GNC concept block diagram. Inspired by [34, Figure 9.1].

3.3 Low-level control

In the literature for marine craft, airplanes and space vehicles, motion control is often divided into guidance, navigation and control (GNC). Fossen defines these concepts as seen in Figure 3.5 in [34]. The guidance systems translate the predefined waypoints into a trajectory, and output the necessary course and speed which will keep the vessel on the trajectory. The control system uses this data as set points, and controls the actuators (e.g. rudder angle and propeller speed) such that the vessel converges to the specified course and speed. The navigation system estimates the position and velocity of the vessel, which is used as feedback for the other systems.

This is how a transit autopilot may be set up. Each of the blocks in Figure 3.5 may have a varying level of complexity, depending on the ship's area of operation. The trajectory generator may be a simple line-of-sight (LOS) guidance law, or it may be a more advanced path following kinematic controller, which uses a dynamical model of the vessel to generate feasible velocities [34]. The motion control system may be a set of simple single-input single-output (SISO) PID-controllers, or it may be a more advanced multiple-input multiple-output (MIMO) controller, linear or nonlinear.

3.3.1 Combined backstepping controller

A controller based on integral backstepping is developed by Fossen et al. in [40]. It is based on the model

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{3.41a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau} + \mathbf{R}(\psi)^\top \boldsymbol{b} \tag{3.41b}$$

where $\boldsymbol{b}$ are slowly varying force disturbances in NED. The controller gives a uniformly globally asymptotically stable (UGAS) origin of the error states z_1, $\boldsymbol{z}_2$ and $\tilde{\boldsymbol{b}}$, where $z_1 = \psi - \psi^d$, $\boldsymbol{z}_2 = \boldsymbol{\nu} - \boldsymbol{\alpha}$, $\tilde{\boldsymbol{b}} = \hat{\boldsymbol{b}} - \boldsymbol{b}$,

$$\boldsymbol{\alpha} = \begin{bmatrix} u^d, & 0, & -k_1 z_1 + \dot{\psi}^d \end{bmatrix}^\top \tag{3.42}$$

and $\hat{\boldsymbol{b}}$ is the disturbance estimator

$$\dot{\hat{\boldsymbol{b}}} = \boldsymbol{\Gamma}\mathbf{R}\boldsymbol{z}_2 \tag{3.43}$$

Here the estimator gains $\boldsymbol{\Gamma}$ is positive definite, and $k_1 > 0$.

The control law is

$$\boldsymbol{\tau} = \mathbf{M}\dot{\boldsymbol{\alpha}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\alpha} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\alpha} - \mathbf{R}^\top(\psi)\hat{\boldsymbol{b}} - \boldsymbol{h}z_1 - \mathbf{K}_2\boldsymbol{z}_2 \tag{3.44}$$

where $\boldsymbol{h} = [0, 0, 1]^\top$ and $\mathbf{K}_2 = \mathrm{diag}\{k_{2,1}, k_{2,2}, k_{2,3}\}$. This control law requires a fully actuated ship, but may be extended to underactuated ships by selecting α_2 such that a relationship $\tau_2 = 0$ is obtained [40]. Using this relationship, a dynamic state α_2 is retrieved from the following equation, where $\mathbf{N} = \mathbf{C} + \mathbf{D}$ is used for notational brevity.

$$\begin{aligned} \tau_2 &= 0 \\ \tau_2 &= m_{22}\dot{\alpha}_2 + m_{23}\dot{\alpha}_3 + n_{21}\alpha_1 + n_{22}\alpha_2 + n_{23}\alpha_3 + \sin\psi\hat{b}_1 - \cos\psi\hat{b}_2 - k_{2,2}z_{2,2} = 0 \\ m_{22}\dot{\alpha}_2 &= -m_{23}\dot{\alpha}_3 - n_{21}\alpha_1 - n_{22}\alpha_2 - n_{23}\alpha_3 - \sin\psi\hat{b}_1 + \cos\psi\hat{b}_2 + k_{2,2}z_{2,2} \end{aligned} \tag{3.45}$$

In the simulations performed in this thesis, the relationship $\tau_2 = 0$ is changed to $\tau_2 = -\frac{1}{l_1}\tau_3$ due to the thruster model (3.37), which gives a different dynamic state α_2:

$$\begin{aligned} \tau_2 &= -\frac{1}{l_1}\tau_3 \\ \tau_2 &= m_{22}\dot{\alpha}_2 + m_{23}\dot{\alpha}_3 + n_{21}\alpha_1 + n_{22}\alpha_2 + n_{23}\alpha_3 + \sin\psi\hat{b}_1 - \cos\psi\hat{b}_2 - k_{2,2}z_{2,2} \\ &= -\frac{1}{l_1}(m_{32}\dot{\alpha}_2 + m_{33}\dot{\alpha}_3 + n_{31}\alpha_1 + n_{32}\alpha_2 + n_{33}\alpha_3 - k_{2,3}z_{2,3} - \hat{b}_3) \\ &\qquad\qquad (l_1 m_{22} + m_{32})\dot{\alpha}_2 = \\ &\qquad l_1(-m_{23}\dot{\alpha}_3 - n_{21}\alpha_1 - n_{22}\alpha_2 - n_{23}\alpha_3 - \sin\psi\hat{b}_1 + \cos\psi\hat{b}_2 + k_{2,2}z z_{2,2}) \\ &\qquad\qquad -m_{33}\dot{\alpha}_3 + k_{2,3}z_{2,3} - n_{31}\alpha_1 - n_{32}\alpha_2 - n_{33}\alpha_3 + k_{2,3}z_{2,3} + \hat{b}_3 \end{aligned} \tag{3.46}$$

Reference models

To generate the time differentiated reference signals, a reference model is used for both surge and heading. The surge reference model is of order 2 to achieve smooth $\dot{u}^d$. It is retrieved from [34, Section 10.2.1] and has the form

$$\ddot{u}_r^d + 2\zeta_u \omega_{n,u} \dot{u}_r^d + \omega_{n,u}^2 u_r^d = \omega_{n,u}^2 u_r^{\text{ref}} \tag{3.47}$$

where $\omega_{n,u} > 0$ and $\zeta_u > 0$. The state-space representation is

$$\begin{bmatrix} \dot{u}^d \\ \ddot{u}^d \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_{n,u}^2 & -2\zeta_u \omega_{n,u} \end{bmatrix} \begin{bmatrix} u^d \\ \dot{u}^d \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_{n,u}^2 \end{bmatrix} u^{\text{ref}} \tag{3.48}$$

For the heading reference signals, a third order model is desired, as smooth r^d and $\dot{r}^d$ are needed. The following model is used:

$$\frac{\psi^d}{\psi^{\text{ref}}}(s) = \frac{\omega_{n,\psi}^3}{(s + \omega_{n,\psi})(s^2 + 2\zeta_\psi \omega_{n,\psi} s + \omega_{n,\psi}^2)} \tag{3.49}$$

The state-space representation is

$$\dot{\psi}^d = r^d \tag{3.50a}$$

$$\dot{r}^d = \ddot{r}^d \tag{3.50b}$$

$$\ddot{r}^d = -(2\zeta_\psi + 1)\omega_{n,\psi} \dot{r}^d - (2\zeta + 1)\omega_{n,\psi}^2 r^d + \omega_{n,\psi}^3 (\psi^{\text{ref}} - \psi^d) \tag{3.50c}$$

Each of these states are saturated at yaw rate and acceleration limits by $r_{\max}^d$ and $\dot{r}_{\max}^d$, respectively, to improve feasibility.

3.3.2 Control allocation

As stated in Section 3.1.4, a ferry with two azimuth thrusters will be overactuated during docking operations. This means that there are an infinite number of combinations of control inputs $\boldsymbol{u}$ that will produce a given force vector $\boldsymbol{\tau}$. For ease of reference, the decomposed relationship between the control inputs and force will be repeated here:

$$\begin{aligned} \boldsymbol{\tau}(\boldsymbol{u}) &= \mathbf{T}\boldsymbol{f} \\ &\Updownarrow \\ \begin{bmatrix} X \\ Y \\ N \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & -l_1 & 0 & l_2 \end{bmatrix} \begin{bmatrix} f_{1,x} \\ f_{1,y} \\ f_{2,x} \\ f_{2,y} \end{bmatrix} \end{aligned} \tag{3.51}$$

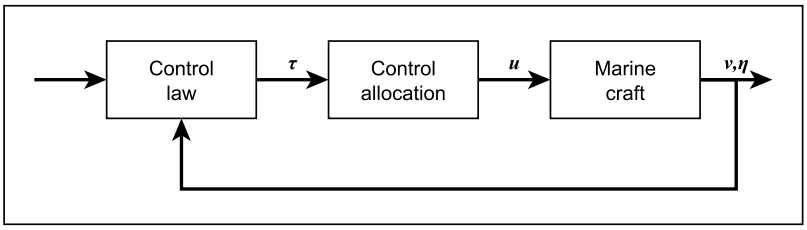


Figure 3.6: Control allocation in a feedback control system. A control law feeds the desired virtual force vector $\boldsymbol{\tau}$ to the control allocation algorithm, which produces the appropriate control vector $\boldsymbol{u}$ for the actuators. Inspired by [34, Figure 12.25].

A reasonable choice would be to select the combination of control inputs that uses the least amount of power. Figure 3.6 illustrates the concept of a control allocation setup in a motion control system.

Fossen's book [34, Section 12.3] introduces many concepts for optimal control allocation. The examples presented next are gathered from there. Additionally, a thorough review of control allocation methods for overactuated systems is found in Johansen and Fossen's survey [41].

Explicit solution to the unconstrained allocation problem

When no constraints are placed on the controls $\boldsymbol{u}$, the control allocation problem may be seen as an unconstrained least squares optimization problem.

$$\begin{aligned} &\min_{\boldsymbol{f}} \boldsymbol{f}^\top \mathbf{W} \boldsymbol{f} \\ &\text{s.t. } \boldsymbol{\tau} = \mathbf{K} \boldsymbol{f}, \end{aligned} \tag{3.52}$$

where $\mathbf{W} > 0$ is a weighting matrix. This problem has an explicit solution, according to [34]:

$$\boldsymbol{f} = \mathbf{T}^\dagger \boldsymbol{\tau}, \tag{3.53}$$

where

$$\mathbf{T}^\dagger = \mathbf{W}^{-1}\mathbf{T}^\top(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^\top)^{-1} \tag{3.54}$$

is the generalized inverse.

Explicit solution to the constrained allocation problem

Tøndel et al. have developed an explicit solution to the constrained quadratic programming (QP) problem [42], which is extended to marine applications by

Johansen et al. in [43]. The constrained problem is formulated as

$$\begin{aligned} &\min_{\boldsymbol{f},\boldsymbol{s},\bar{f}} \{\boldsymbol{f}^\top \mathbf{W}\boldsymbol{f} + \boldsymbol{s}^\top \mathbf{Q}\boldsymbol{s} + \beta\bar{f}\} \\ &\text{s.t. } \mathbf{T}\boldsymbol{f} = \boldsymbol{\tau} + \boldsymbol{s} \\ &\quad \boldsymbol{f}_{\min} \leq \boldsymbol{f} \leq \boldsymbol{f}_{\max} \\ &\quad -\bar{f} \leq f_{1,x}, f_{1,y}, f_{2,x}, f_{2,y} \leq \bar{f} \end{aligned} \tag{3.55}$$

where $\boldsymbol{s} \in \mathbb{R}^3$ are slack variables which ensure feasibility. The matrix $\mathbf{Q} > 0$ is penalization on the slack variables, $\bar{f} = \max_i |f_{i,j}|$ is the largest force among $\boldsymbol{f}$, and β penalizes the largest force.

In general this type of problem has to be solved by an iterating algorithm, but, as mentioned, an approach for explicit solutions of the parametric QP is developed in [42] and [43].

Iterative solutions with varying $\boldsymbol{\alpha}$

The previous control allocation examples involve decomposing the thruster forces into x and y components. In reality the azimuth thruster angles α_i have limitations on both amplitude and rate of change. In addition, the limitations on $f_{i,j}$ is only a weak approximation of the real-world situation, where the limitation is in fact on the force magnitude F_i. Johansen et al. suggest the following nonconvex nonlinear optimization problem [44]:

$$\begin{aligned} \min_{\boldsymbol{F},\boldsymbol{\alpha},\boldsymbol{s}} &\left\{ \sum_{i=1}^{2} \bar{P}_i |F_i|^{3/2} + \boldsymbol{s}^\top \mathbf{Q}\boldsymbol{s} + (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)^\top \boldsymbol{\Omega} (\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) \right. \\ &\left. + \frac{\varrho}{\varepsilon + \det(\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha}))} \right\} \\ \text{s.t. } &\mathbf{T}(\boldsymbol{\alpha})\boldsymbol{F} = \boldsymbol{\tau} + \boldsymbol{s} \\ &\boldsymbol{F}_{\min} \leq \boldsymbol{F} \leq \boldsymbol{F}_{\max} \\ &\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max} \\ &\Delta\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha}_{\max} \end{aligned} \tag{3.56}$$

where

- $\bar{P}_i |f_i|^{3/2}$ is power consumption for each thruster, where $\bar{P}_i$ is a positive weight.
- $\boldsymbol{s}^\top \mathbf{Q}\boldsymbol{s}$ is the penalization on the slack variables $\boldsymbol{s}$.
- $\boldsymbol{\Omega}$ is the matrix which penalizes changes in $\boldsymbol{\alpha}$, reducing wear and tear on the azimuth thrusters.

- $\boldsymbol{F}_{\min} \leq \boldsymbol{F} \leq \boldsymbol{F}_{\max}$ and $\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max}$ are saturations on azimuth forces and angles.
- $\Delta\boldsymbol{\alpha}_{\min} \leq \boldsymbol{\alpha} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha}_{\max}$ limits the rate of movement of $\boldsymbol{\alpha}$. $\boldsymbol{\alpha}_0$ is the azimuth angles at the previous sample.
- A compensation term

$$\frac{\varrho}{\varepsilon + \det(\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^{\top}(\boldsymbol{\alpha}))} \tag{3.57}$$

  is included to avoid singular configurations. ϱ and ε are scalars that tune this term.

This nonconvex nonlinear problem is hard to solve, and may not be feasible for real-time implementations. A linear approximation of the forces and angles may be made to reduce this to a QP problem:

$$\boldsymbol{F} = \boldsymbol{F}_0 + \Delta\boldsymbol{F} \tag{3.58a}$$
$$\boldsymbol{\alpha} = \boldsymbol{\alpha}_0 + \Delta\boldsymbol{\alpha} \tag{3.58b}$$

The resulting QP problem then becomes [44]:

$$\begin{aligned}
\min_{\Delta\boldsymbol{F},\Delta\boldsymbol{\alpha},\boldsymbol{s}} \Bigg\{ & (\boldsymbol{F}_0 + \Delta\boldsymbol{F})^{\top}\mathbf{P}(\boldsymbol{F}_0 + \Delta\boldsymbol{F}) + \boldsymbol{s}^{\top}\mathbf{Q}\boldsymbol{s} + \Delta\boldsymbol{\alpha}^{\top}\boldsymbol{\Omega}\Delta\boldsymbol{\alpha} \\
& + \frac{\partial}{\partial\boldsymbol{\alpha}}\left(\frac{\varrho}{\varepsilon + \det(\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^{\top}(\boldsymbol{\alpha}))}\right)\Bigg|_{\boldsymbol{\alpha}_0} \cdot \Delta\boldsymbol{\alpha} \Bigg\} \\
\text{s.t. } & \boldsymbol{s} + \mathbf{T}(\boldsymbol{\alpha}_0)\Delta\boldsymbol{F} + \tfrac{\partial}{\partial\boldsymbol{\alpha}}(\mathbf{T}(\boldsymbol{\alpha}_0)\boldsymbol{F})|_{\boldsymbol{\alpha}_0,\boldsymbol{F}_0} \cdot \Delta\boldsymbol{\alpha} = \boldsymbol{\tau} - \mathbf{T}(\boldsymbol{\alpha}_0)\boldsymbol{F}_0 \\
& \boldsymbol{F}_{\min} - \boldsymbol{F}_0 \leq \Delta\boldsymbol{F} \leq \boldsymbol{F}_{\max} - \boldsymbol{F}_0 \\
& \boldsymbol{\alpha}_{\min} - \boldsymbol{\alpha}_0 \leq \Delta\boldsymbol{\alpha} \leq \boldsymbol{\alpha}_{\max} - \boldsymbol{\alpha}_0 \\
& \Delta\boldsymbol{\alpha}_{\min} \leq \Delta\boldsymbol{\alpha} \leq \Delta\boldsymbol{\alpha}_{\max}
\end{aligned} \tag{3.59}$$

This problem is convex and may be solved by a standard QP solver.

Chapter 4

Industrial Motion Control Systems from a Ferry Perspective

A modern ship consists of multitudes of advanced equipment for navigation and motion control, communication, ship and obstacle detection, naval maps, machinery, propulsion, and steering. After leaving the harbor area and upon entering open waters, a modern ship may transit on its own, with crew present only to monitor the ship's machinery and to intervene in case a COLREGS situation occurs. This is made possible by state-of-the-art control equipment.

While a large ocean going ship moves automatically from point A to point B, the fjord-going ferries are mostly controlled manually. As an example, the *Korsfjord* carries only a heading autopilot for control, leaving it up to the crew to compensate for currents and wind [45].

This chapter will introduce equipment and layouts commonly used in ships, such as actuators, autopilot systems, navigation systems and more. The content is heavily based on Golding's work in [46], with some updates from recent sources and publications.

4.1 Equipment types

An industrial motion control system for ships commonly consists of modular commercial off-the-shelf (COTS) units. These units have specified responsibilities, and communicate with each other using protocols defined by National Marine Electronics Association (NMEA) and IEC61162 [47, 48]. The modular units may be divided into three categories: measurement systems, actuators, and information processing units.

Examples of sensor systems are:

- Position measurement:

  Global navigation satellite system (GNSS): Satellite based positioning system, determining position using trilateration.

  Inertial navigation system (INS): System using inertial sensors such as accelerometers and gyros to estimate velocity and position. Often fused with GNSS with the help of a Kalman filter.

  Hydroacoustic position sensor: System using transceivers and transponders to estimate position with sound waves. One or more transponders are fastened to the seabed, and a transceiver on-board the ship pings it and measures time of flight.

- Heading measurement:

  Gyrocompass: Non-magnetic compass which uses a fast-spinning disc to exploit *gyroscopic precession* to determine deviation from *geodetic north*.

  Magnetic compass: Device which determines deviation from *magnetic north* by letting a magnetized needle align with the surrounding magnetic field.

  GNSS compass: Using two separate GNSS antennas, heading may be determined by measuring the vector of the baseline between the antennas.

- Attitude measurement:

  Attitude and heading reference system (AHRS): System using inertial sensors (accelerometers, gyroscopes, and heading information) to determine a vehicle's attitude (roll and pitch).

- Ship identification and tracking:

  Radar: System using microwaves to determine range, velocity, and angle to objects. A radio pulse is emitted, which is reflected by any object, and the reflection is measured by the radar.

  Automatic radar plotting aid (ARPA): A system based on radar, which tracks and calculates the velocity vector of objects detected by radar.

  Automatic identification system (AIS): Ship tracking system, where each ship communicates its position and velocity (among other information) by radio to nearby vessels.

- Wind measurement:

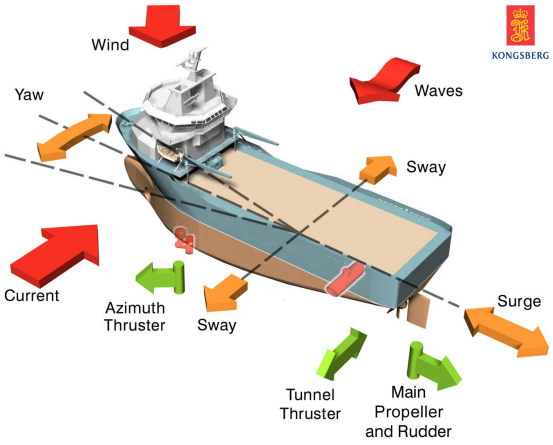


Figure 4.1: Concept illustration of DP systems. © Kongsberg.

Anemometer: Device used for measuring wind speed and direction. The anemometer may be based on mechanical or ultrasonic principles. Mechanical anemometers consist of rotating hemispheres mounted on a bar. Ultrasonic anemometers measure wind speed and direction in two or three dimensions.

- Relative water speed measurement:

  Acoustic Doppler current profiler (ADCP): System measuring relative water velocity using sonar pulses and the Doppler effect of reflected sound waves.

More information about these types of sensors are found in Section 7.7.

Examples of actuators are rudders, propellers, azimuth thrusters, and roll stabilizers. These provide the force necessary to move the ship in transit and docking.

The sensor and actuator modules provide inputs to, and receive outputs from, information processing units, such as waypoint managers, heading and course autopilots, tracking controllers, and dynamic positioning (DP) systems.

4.2 Architecture and capability of DP systems

A DP system is designed to replace an anchor or mooring at deep sea, or at places where it is infeasible to use such equipment. DP uses the ship's thruster capabilities to keep it stationary at a point or in an area. The ship may also hold a desired heading. Figure 4.1 is a concept illustration of how a DP system works. This might be interesting for autonomous ferries in cases where the ferry needs to

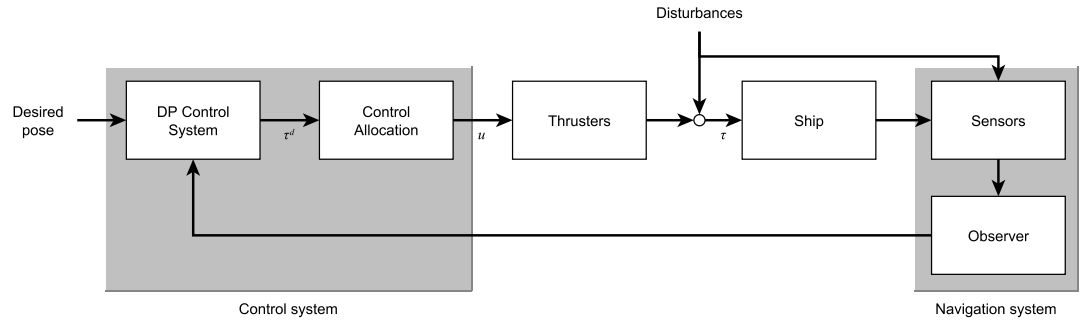


Figure 4.2: Block diagram for DP systems.

wait before docking, or other situations requiring the ship to be stationary in the water.

Advanced ship models, state estimators, and sensors are used to estimate environmental forces, and counteract these, so that the vessel achieves the desired pose. As the nature of DP operations entails keeping an arbitrary position and heading, a ship with such capabilities must have full actuation. Design of a DP system is covered in the work of Sørensen et al. in [49]. Information regarding navigation and control systems for DP is available in Fossen's book [34]. A simple overview of a DP system is found in Figure 4.2.

In order to keep a constant position and heading, a DP ship's *navigation system* must be very accurate. Commonly used positioning equipment includes acoustic systems and differential GPS (DGPS). For heading, a gyrocompass provides accurate heading measurements.

An estimator is able to provide the necessary information of states and environmental forces to be able to counteract them. Commonly used estimators include Kalman filters, nonlinear passive observers, and wave filters. This information is fed to a control system, which uses it to generate the necessary counter forces.

Many traditional and advanced control designs are capable of providing well-performing DP capability. Proportional-integral-derivative (PID) control and linear-quadratic-Gaussian (LQG) control is commonly used. These controllers output desired force and moment in surge, sway, and yaw, which will keep the ship in the desired position.

A fully actuated ship (in 3 DOF) is able to provide an arbitrary force in surge and sway, as well as a desired yaw moment. The procedure for translating this desired force to actuator inputs (such as rpm and direction) is called *control allocation*. Optimal control allocation uses an objective function to minimize the amount of energy spent on creating the desired force vector. A compilation of control allocation algorithms is found in Johansen and Fossen's survey [41]. A

(a) Tunnel thruster. CC BY Wikimedia user *Dr. Karl-Heinz Hochhaus*.

(b) Schottel azimuth thruster. CC BY Siemens AG.

(c) Ship propeller. By the US government.

Figure 4.3: Images of different thruster types.

simple method for control allocation is weighted least squares, which has a closed-form solution when no inequality constraints are imposed. More information is also found in [34, Section 12.3] and in Section 3.3.2.

To attain full or redundant actuation, a DP ship may use azimuth thrusters, tunnel thrusters, and fixed propellers with rudders. Ships that do not have DP capabilities may also have additional thrusters to ease docking or reduce the need for tug boats. Figure 4.3 shows photos of different types of thrusters.

4.3 Architecture and capability of autopilot systems

The complexity of autopilot systems varies. Simple autopilots keep a ship's heading or course steady, while more advanced autopilots include guidance systems that keep the ship on a predefined track. While heading and course autopilots are included in the more advanced systems, they are relatively simple, and [34] treats them in depth.

A track controller consists of a guidance system, which provides a course (and speed) reference to a course controller. A navigation system provides the controller with position data, and observers and additional sensors provides information about wind and wave forces. Track controllers may be divided into two groups: cascaded and single-loop architecture.

4.3.1 Cascaded track controllers

Cascaded track controllers have an inner course or heading controller, which receives course and velocity information from an observer based on compass and

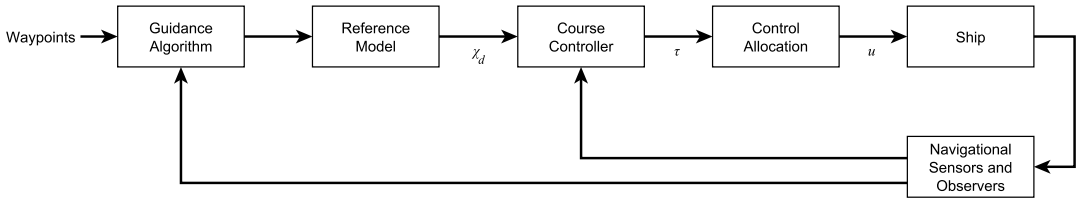


Figure 4.4: Cascaded track controller block diagram.

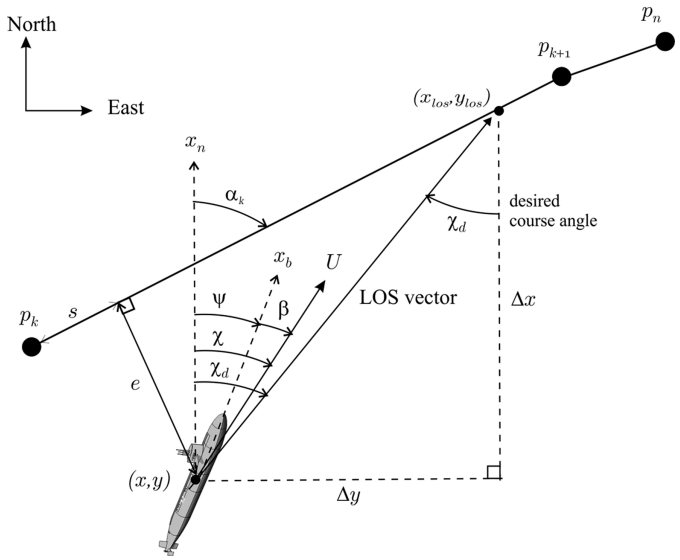


Figure 4.5: LOS guidance where the desired course angle χ_d is chosen to point towards the LOS intersection point (x_{los}, y_{los}). From [34]. © 2011 John Wiley & Sons, Ltd.

GNSS information. The course error is fed to a controller which calculates an appropriate rudder action to stabilize the error, and the actual course converges to the desired value.

The course reference is retrieved from the outer guidance loop. This loop gets position information from GNSS and/or observers, and uses this to generate an appropriate course or heading to keep the ship on the predefined track. The principle is seen in Figure 4.4. The guidance controller may be based on geometrical or model based principles, often augmented with heuristics which improve the controller quality. A geometrical principle called LOS guidance is described in Fossen's book [34]. Figure 4.5 shows this principle where the ship is guided towards a LOS intersection point (x_{los}, y_{los}). Information about guidance is also found in Section 5.5. Guidance controllers based on curved paths do also exist, and the theory behind them is explained in Section 5.5.

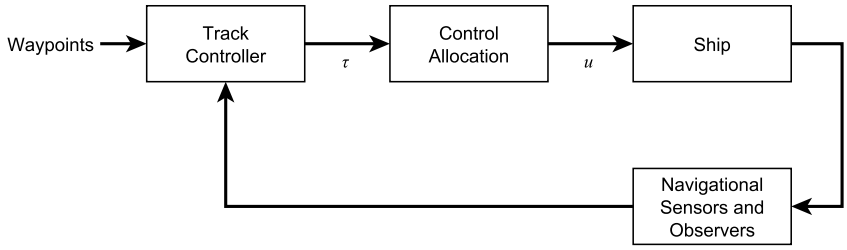


Figure 4.6: Single-loop track controller block diagram.

4.3.2 Single-loop track controllers

While a cascaded system closes the course and track loop successively, a single-loop track controller feeds all state information (position, velocity, heading, observed states, etc.) to a track controller. This controller type provides a desired force vector, which is subsequently translated to actuator control signals. The concept is illustrated in Figure 4.6. A model-based approach using LQG is presented in Holzhüter's paper [50]. Another model-based approach using methods based on heuristics and PID-principles is found in Källström's work in [51].

4.3.3 Physical architecture

As noted in [46] and in Section 4.1, most commercially available autopilot systems for ships consist of modular COTS products, which communicate using standard protocols. Figure 4.7 shows a concept illustration of the physical architecture of an industrial autopilot system.

Autopilot system developers often also create or augment existing protocols with proprietary protocols that enhance communication between their equipment. More integrated approaches are also found, where it is impossible to separate the subsystems as standalone physical components.

4.4 Actuator layouts and categories

Golding introduces categories for maneuverability of ships in [46]. The categories are:

Category A No/low maneuverability.

Category B Low/moderate maneuverability.

Category C Moderate/high maneuverability (DP ships).

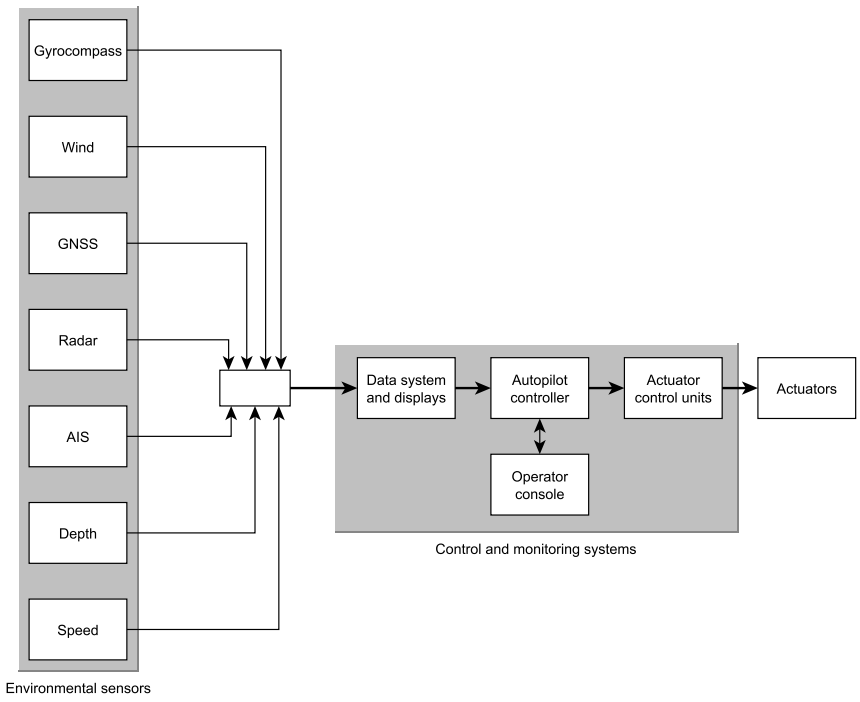


Figure 4.7: Conceptual block diagram of physical autopilot architecture.

Large ocean-going ships which have high requirements on efficiency and power output, and low requirements on maneuverability, as docking is handled by tug boats, fall into Category A. These ships are characterized by only having stern propellers with rudders. Ferries, which need to be able to dock without tug boats, may be put into Category B. These ships usually have some form of auxiliary thrusters for docking, such as tunnel thrusters or azimuth, or in the case of most ferries, an azimuth thruster in each end. Category C includes DP vessels, and ships in this category have a multitude of thrusters. There are often several azimuth thrusters and tunnel thrusters employed to keep the ship stationary in challenging conditions.

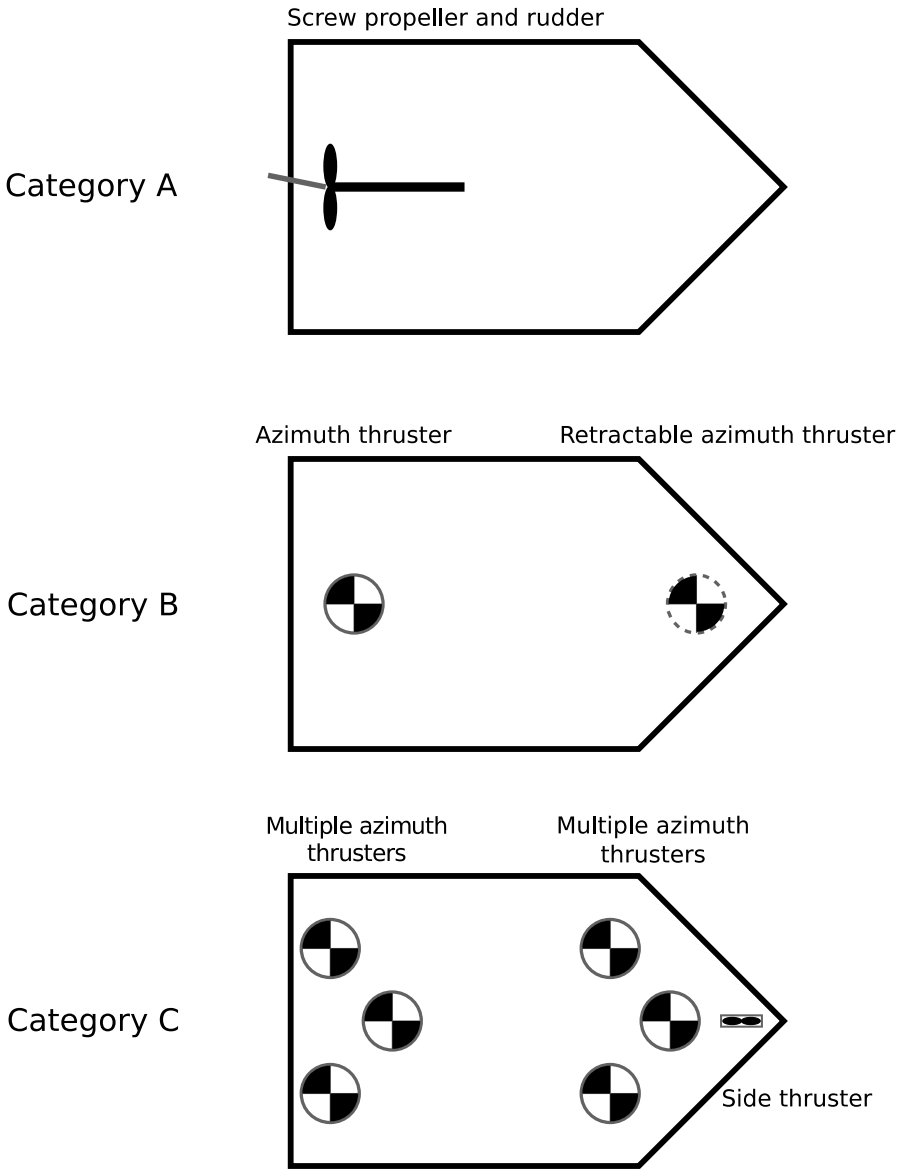


Figure 4.8: Examples of actuator layout categories. Ships in Category A may also have a set of two stern propellers and rudders, or just one or two stern azimuth thrusters, eliminating the need for rudders. Ships in Category B may also carry non-retractable bow azimuth thrusters (like most ferries), or no bow azimuth thruster, but side thrusters in the front. Redundancy is the deciding factor of whether a ship falls into Category C, where several different configurations may fulfill these requirements.

Chapter 5

Path Planning and High-Level Control

When a ship automatically transits from one location to another it follows a pre-planned path. Finding a reasonable path from A to B is a common challenge in robotics and ship motion control. Figure 5.1 shows an overview of the building blocks in a control system, where the highlighted blocks are relevant to this chapter. Figure 5.2 displays a possible sequence of events where a ferry goes through path planning, path following, and collision avoidance.

When designing motion control systems for an autonomous ferry, it is important to maintain the priorities introduced in Section 1.2, repeated here:

1. Safety
2. Area of operation
3. Punctuality
4. Green

The *safety* priority is mostly related to collision avoidance, but also means that the path planning algorithm must account for reefs and other static obstacles. Another aspect of safety is to avoid areas with unfavorable weather conditions, where such situations may occur. The choice of path planning algorithm, optimization criterion, and motion control algorithms affect energy consumption.

This chapter contains a thorough presentation of path planning methods and an introduction to path following — making an underactuated ship follow the path specified by the planning algorithms.

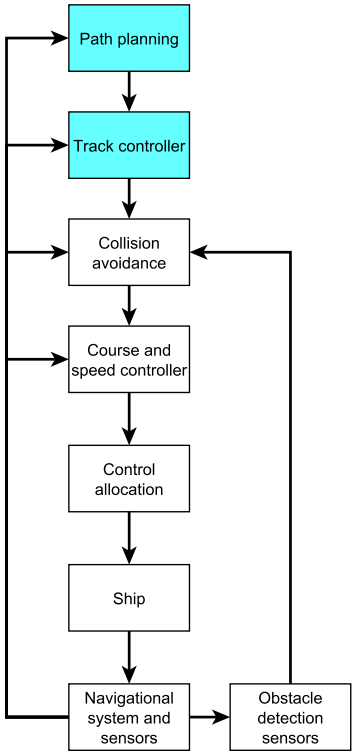


Figure 5.1: Block diagram of a motion control system. The highlighted blocks are relevant to the discussion of path planning.

5.1 Path planning

A crucial part of automatic motion control for ships is path planning. Path planning is the procedure of generating a path from a start position to a final position, subject to some constraints. A relevant example is moving a ferry from one dock to another, while avoiding static obstacles like reefs and land. An algorithm which solves the path planning problem is a *global method*.

The path planning problem is in [52] formulated as

$$P_s \xrightarrow{\coprod r(\theta)} P_f, \tag{5.1}$$

where P_s and P_f denote the start and end pose, respectively, $r(\theta)$ denotes the generated path as a function of a path variable, and $\coprod$ denotes the constraints imposed on the path.

There are two main categories of methods used for path planning: Roadmap methods and parametrized path generation methods. Roadmap methods generate waypoints that if connected form a trajectory from one place to another.

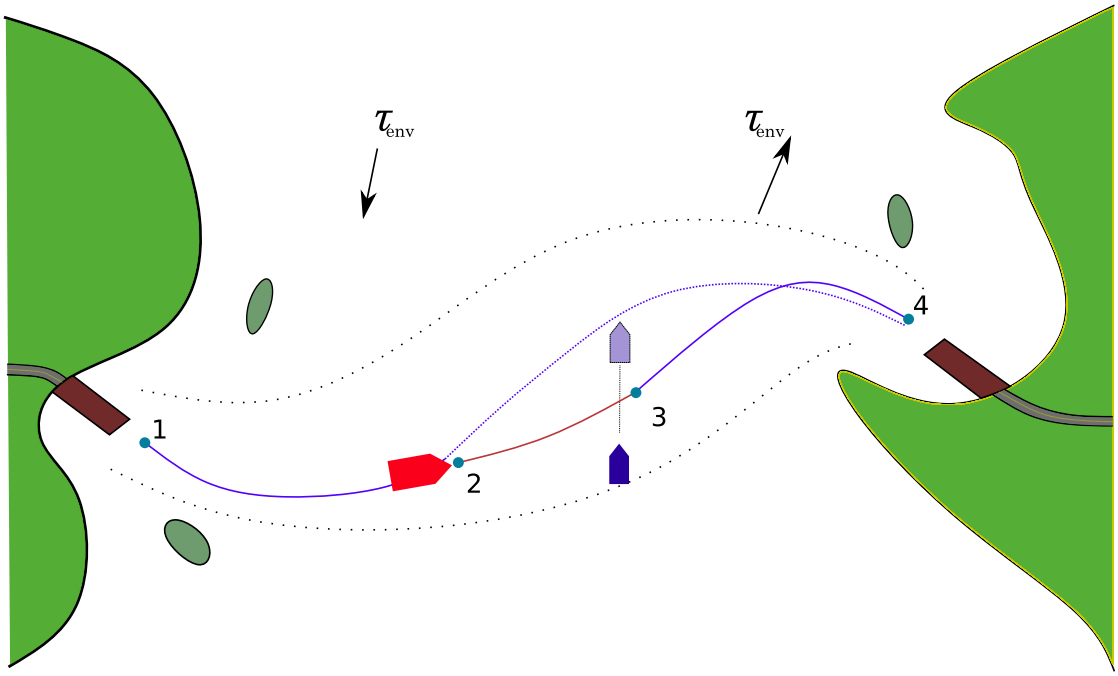


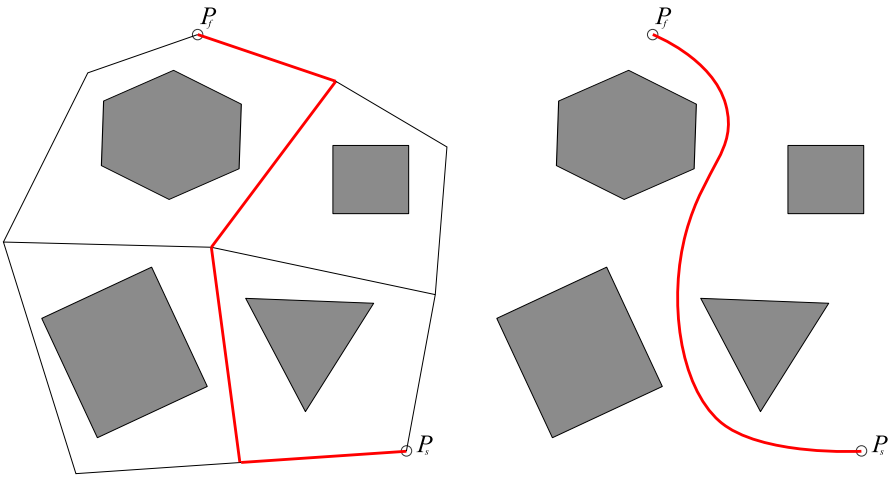
Figure 5.2: Example of a path planning scenario for a ferry. The ferry undocks from the port and starts to follow a nominal trajectory at *1*. At *2*, another ship is detected which demands the ferry to follow the crossing situation described in COLREGS. A new trajectory is calculated at *3*, which takes the ferry to the end of the transit, where it will begin a docking process at *4*.

Optimization-based parametrized path generation methods methods generate some trajectory which starts in P_s and ends in P_f, where the trajectory is optimal in some sense.

5.2 Roadmap methods

Roadmap methods are a class of planning methods which is based on generating waypoints that if connected form a trajectory from one place to another. The concept is illustrated in Figure 5.3. There are several methods for generating roadmaps, among them are *Voronoi diagrams*, *visibility graphs*, and *cell decomposition*. Common to these methods are that the resulting output is only points that connect a non-colliding path from P_s to P_f. Most of them do not inherently include dynamic constraints, which requires the path planning system to connect the points in a feasible manner.

Roadmap methods may be compared in terms of computational complexity.



(a) Result of roadmap generation. **(b)** After smoothing.

Figure 5.3: Illustration of the roadmap path planning concept.

The main distinction is *combinatorial* versus *sampling-based* methods. Combinatorial planning is also called exact planning, and searches the whole continuous space for roadmap vertices, which is inherently computationally expensive. Sampling-based methods are probabilistic in nature, and often have shorter running times for high-dimensional problems.

5.2.1 Voronoi diagram

Using Voronoi diagrams for path planning is done by first creating points at the vertices of each obstacle, and then create a roadmap with edges that are maximally distant to those points. The resulting tree is then used to search for a shortest path from P_s to P_f. Subsequent path smoothing is required. The Voronoi diagram method is *combinatorial*.

Candeloro et al. [53] present a path planning method using Voronoi diagram results connected with straight lines and curves of Fermat's spiral segments. Figure 5.4 illustrates the workflow of the path-planning module, where the authors use the Dijkstra-Yen algorithm [54] to find the shortest path from P_s to P_f, removes collinear waypoints, checks for obstacle clearance, and removes other unnecessary waypoints. Subsequently, a path smoothing module creates Fermat's spirals between the straight segments, which makes the trajectory curvature-continuous[1].

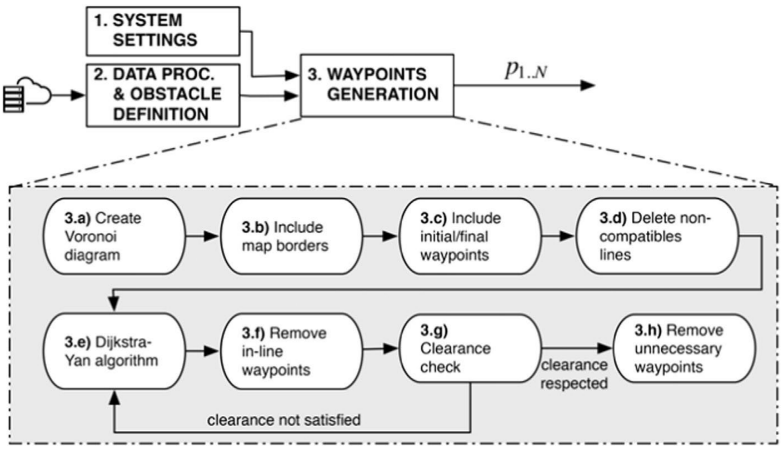


Figure 5.4: Flowchart of the method used for path planning in [53]. Subsequent to this operation, a path smoothing module connects straight line segments with Fermat's spirals. © 2017 Elsevier Ltd.

An example of the results is seen in Figure 5.5.

5.2.2 Cell decomposition

Cell decomposition for path planning is to divide the environment into a grid of non-overlapping cells. Every cell which has an obstacle in it is marked as obstructed, and a search algorithm connects adjacent free cells with straight lines. Figure 5.6 illustrates the concept. The cell decomposition method is *combinatorial*. The method is referred to as the cell decomposition method by Tsourdos et al. in [52], but also goes under the name *occupancy grid*. More information about the occupancy grid is found in Moravec and Elfes (1985) [57].

5.2.3 Visibility graph

Visibility graph planning is a method which is based on that the shortest trajectory between two points in a map obstructed by polygons, includes some of the polygons' edges. The Visibility graph method is *combinatorial*. The graph is illustrated in Figure 5.7. The method involves drawing polygons around the obstacles, creating a graph. The graph contains the polygon vertices plus the start and end points. The edges are formed by connecting all points whose straight lines do not

[1] Curvature continuity implies that two connecting segments have both the same direction and radius of the curvature at the connection point. This is denoted G^2 continuity. See [52]. G^n geometric continuity is equivalent to C^n continuity of the arc length parametrized path [55]. See [56] for more information on Fermat's spiral.

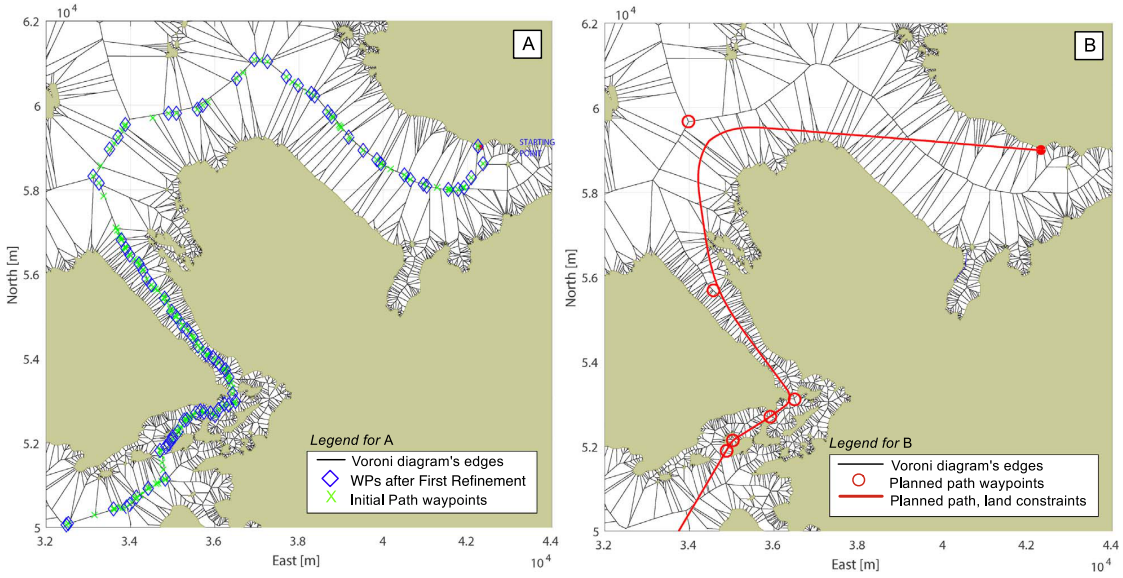


Figure 5.5: Illustration of a path generated by a Voronoi diagram method in [53]. (A) shows the output from the Voronoi diagram search, and (B) shows the trajectory smoothed with Fermat's spirals. © 2017 Elsevier Ltd.

intersect any polygon. Subsequently, the shortest path from P_s to P_f is found by using a shortest path algorithm (like Dijkstra or A*) on the graph.

A notable source on visibility graphs is Lozano-Pérez' article [58], which introduces an algorithm for generating the polygons and the visibility graph, and solving it. Casalino et al. also use the visibility graph method to create an initial path, and introduces lower level collision avoidance techniques in [59]. This is more relevant to Chapter 7.

5.2.4 Probabilistic roadmap

Kavraki et al. present a probabilistic method (sampling based) for motion planning in [61], the probabilistic roadmap (PRM). The method consists of two phases. The first phase is called the *learning phase*, and involves placing nodes randomly in the free configuration space with a given distribution. The nodes are connected, and the corresponding edges correspond to feasible paths.

The second phase is called the *query phase*. Here the start and end configurations are connected to the nodes, and the resulting tree is searched for a path between these points.

Geraerts and Overmars compare several implementations of PRMs where, among other aspects, the sampling distributions and collision checking techniques vary [62]. An illustration of the development of a PRM algorithm is found in

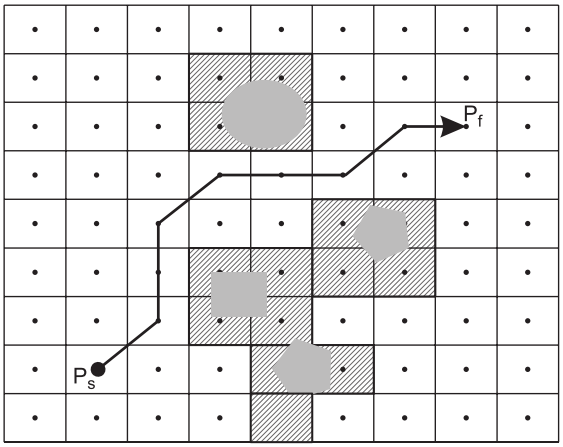


Figure 5.6: Illustration of the cell decomposition path planning method. From [52]. © 2010 John Wiley & Sons, Ltd.

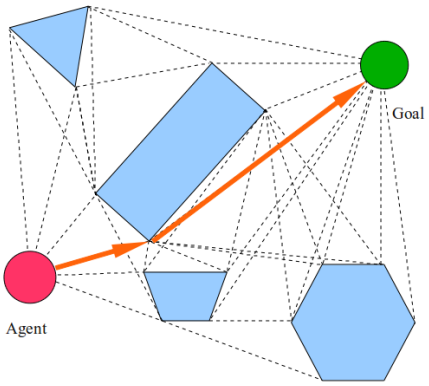


Figure 5.7: Visibility graph illustration. From [60]. © 2017 UBM.

Figure 5.8.

5.2.5 Rapidly-exploring random trees

Rapidly-exploring random trees (RRT) is a *probabilistic* (sampling-based) method for generating an initial path. RRT was first introduced in LaValle's work in [63], as a method for finding a trajectory in an expanded configuration space $\mathcal{X} = \mathrm{T}(\mathcal{C})$, called the *tanget bundle*[2]. The algorithm may be used to find a kinematic path from P_s to P_f in the configuration space containing obstacles. Constraints may also be placed on the velocities, such that the result includes dynamics of the vehicle. The obstacles are denoted by $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$, and $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$ denotes the

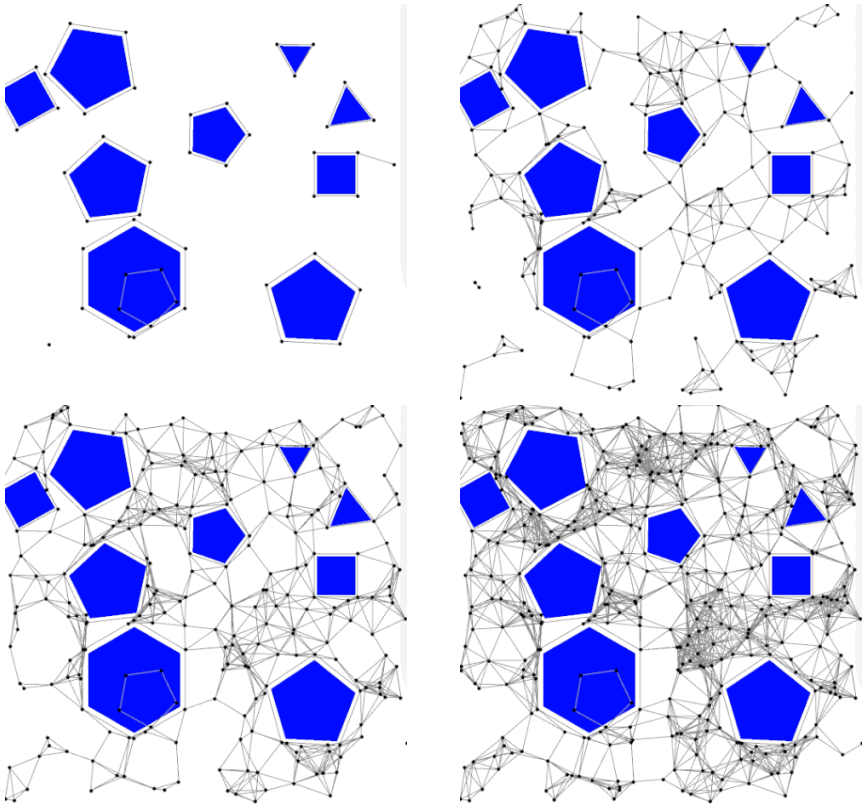


Figure 5.8: Development of PRM algorithm. Eric O. Scott.

allowable configurations.

Algorithm 1 is pseudocode for the RRT method. The algorithm takes as inputs $\boldsymbol{x}_{\text{init}}$, the starting state; K, the number of vertices to produce; and Δt, the integration step of the dynamic model $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$.

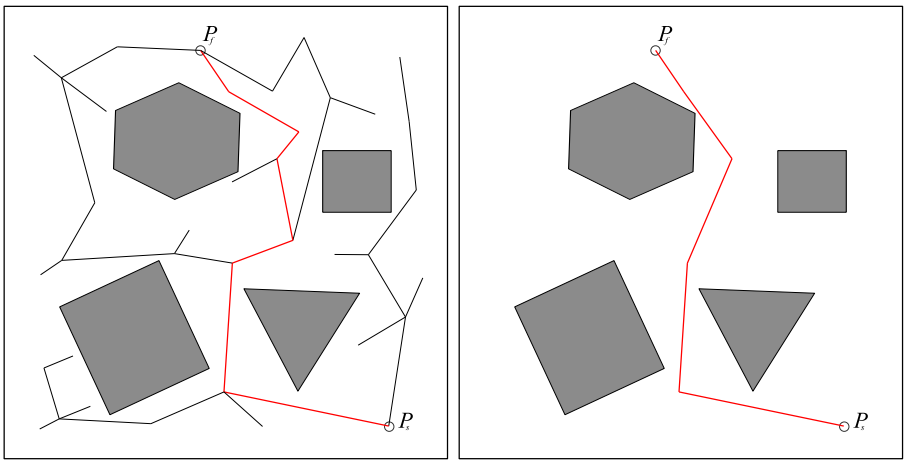
For each iteration, a new state is selected randomly from $\mathcal{X}$ (step 1:4). The nearest neighbor is then selected by some metric function from the existing vertices (1:5). The new input $\boldsymbol{u}$ is selected such that the distance from $\boldsymbol{x}_{\text{near}}$ to $\boldsymbol{x}_{\text{rand}}$ is minimized, subject to that the state remains in $\mathcal{X}_{\text{free}}$ (1:6). This involves some collision detection algorithm (Mirtich's V-Clip algorithm is mentioned in [63]). NEW_STATE (1:7) propagates the state from $\boldsymbol{x}_{\text{near}}$ to $\boldsymbol{x}_{\text{new}}$. Finally, the new vertex $\boldsymbol{x}_{\text{new}}$ is added, along with an edge between $\boldsymbol{x}_{\text{near}}$ and $\boldsymbol{x}_{\text{new}}$. An illustration of the path generated by this algorithm is seen in Figure 5.9a.

Subsequent to this algorithm, the shortest path from P_s to P_f is found using some search algorithm (e.g. Dijkstra). After that a waypoint reduction scheme is applied, removing unnecessary vertices in the path. See Figure 5.9b.

[2] The tangent bundle contains both the configuration and its velocity.

Algorithm 1 RRT algorithm. From [63].

```
1: function GENERATE_RRT(x_init, K, Δt)
2:     T.init
3:     for k = 1 to K do
4:         x_rand ← RANDOM_STATE();
5:         x_near ← NEAREST_NEIGHBOUR(x_rand, T);
6:         u ← SELECT_INPUT(x_rand, x_near);
7:         x_new ← NEW_STATE(x_near, u, Δt);
8:         T.add_vertex(x_new);
9:         T.add_edge(x_near, x_new, u);
10:    return T
```



(a) Initial RRT path. **(b)** After waypoint reduction.

Figure 5.9: Illustration of the RRT method for path planning.

Other helpful sources on RRT are LaValle's follow-up work in [64], and a master's thesis by Hvamb [65].

5.3 Optimal control and optimization methods

Optimization methods revolve around defining an objective function along with some constraints, which when solved provides a trajectory from P_s to P_f that is *optimal* or *optimized*[3] in some sense, defined by the objective function. A common way to describe an optimal control problem is:

Definition 5.1 (Optimization problem)**.**

Minimize

$$J(\boldsymbol{x}(t), \boldsymbol{u}(t), t_f) = E(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} F(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\,\mathrm{d}t \tag{5.2a}$$

subject to

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \tag{5.2b}$$
$$\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, t) \leq 0 \tag{5.2c}$$
$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, t) = 0 \tag{5.2d}$$

Here $J(\cdot)$ is the objective function to be minimized, (5.2b) describes the system dynamics, and $\boldsymbol{g}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are inequality and equality constraints, respectively.

Some optimization problems may be solved analytically, and Pontryagin's famous maximum (or minimum) principle provides necessary conditions for an optimal solution. In [67], Boissonnat et al. find an optimal solution for the *Dubins path* problem using the minimum principle. However, the vast majority of optimization problems have optimums which are impossible to find with analytical methods, and require numerical methods to find an optimized solution.

The subsequent sections will introduce the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation, useful for solving optimal control problems. Additionally, two methods for solving optimal control problems are introduced: The semi-Lagrangian (SL) scheme, which solves the HJB, and the pseudospectral (PS) method, which instead of solving the HJB, approximates the dynamics with polynomials and discretizes these, before solving the resulting problem as a nonlinear program (NLP)[4].

5.3.1 Hamilton-Jacobi-Bellman equation

The HJB is a partial differential equation which is useful for solving optimal control problems (such as the one in Definition 5.1). It relates closely to Pontryagin's maximum principle [68], and provides a sufficient condition of optimality for optimal control problems [69]. The HJB also gives way to methods for solving the optimization problem numerically, by discretization.

[3] The difference between an *optimal* solution and an *optimized* solution is described in [66]. Essentially, an optimal solution is one which is an actual global optimum to the optimization problem. In cases where such an optimum does not exist, or more commonly, where the method used to solve the optimization problem can only provide an approximation because of discretization and computational limitations, an *optimized* solution may be provided.

[4] The PS method may also be used to solve the HJB, but in the literature referred to in this thesis the PS method is used for solving the approximated problem.

With the problem stated as in Definition 5.1, the HJB is:

$$\begin{cases} \frac{\partial}{\partial t}V(\boldsymbol{x},t) + \max\{\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \cdot \nabla_x V(\boldsymbol{x},t) + F(\boldsymbol{x},\boldsymbol{u},t)\} = 0, \ (\boldsymbol{x},t) \in \mathbb{R}^d \times (0,T) \\ V(\boldsymbol{x},t) = E(\boldsymbol{x}(t_f),t_f), \quad \boldsymbol{x} \in \mathbb{R}^d \end{cases} \tag{5.3}$$

The solution of this equation is

$$V(\boldsymbol{x},t) = \inf_{\boldsymbol{u}\in\mathcal{U}} J(\boldsymbol{x},\boldsymbol{u},t), \tag{5.4}$$

and is called the *value function* (in some texts this function is called the *return function*). As stated in an article by Sundar and Shiller [70], the value function has a unique minimum and is key to find the solution to Definition 5.1. The solution of this problem will in turn provide an optimized trajectory from P_s to P_f.

5.3.2 Semi-Lagrangian scheme

The SL scheme may be used to solve the HJB. This has successfully been attempted in [71], where Roald defined the path planning problem and solved it by using forward Euler approximation for discretization and linear interpolation. The solution was tested on models of Dubins car, Reeds-Shepp car and a 4 state boat model in 3 DOF (x, y, ψ, u).

As stated in [71], the SL scheme was proposed by Courant et al. in 1952 [72]. The method is based on fluid dynamics, and is a combination of the Lagrangian and Eulerian approaches to describing the motion of fluid parcels.

The Lagrangian approach is to describe the flow of each particle in a reference frame that moves with the particle. On the other hand, the Eulerian approach is to look at the fluid parcel's motion from a fixed reference frame. More information on the Lagrangian scheme can be found in an article by Bowman et al. [73].

The SL scheme uses a fixed grid, but with the description of the advective derivative in a Lagrangian frame — keeping the other spatial derivatives in an Eulerian frame.

Other important aspects of the SL scheme are reconstruction by interpolation between the discretized elements, and approximation of the continuous system by discretization. More information on these subjects are found in the book by Falcone and Ferretti [74].

A summary of the steps needed to solve an optimal control problem follows:

1. Find a discretized version of the HJB by introducing a time step Δt.
2. Build a structured grid of the solution space.
3. Solve the value function in the structured grid.

4. The structured grid now contains a monotonically decreasing path from a starting point to its minimum. Find this path.

5. Implement the control action resulting from the computation of the value function.

5.3.3 Pseudospectral method

PS methods may be used for solving the HJB, and thus solving an optimal control problem, but the focus of this section is a way of approximating the NLP in Definition 5.1 and solving it by using Legendre pseudospectral discretization methods. This is done with success in both [71] and an article by Gong et al. [75], using the MATLAB software package DIDO by Elissar Global [76]. Roald's master thesis [71] is also followed up in an article by Lekkas et al. [77].

The basic principles of the PS method gathered from [75] where they use the *Lobatto collocation points* follow:

1. Approximate the states $\boldsymbol{x}(t)$ by Lagrange polynomials of order N, $\boldsymbol{x}^N(t)$.

2. Transform the time domain to $[-1, 1]$ by an affine transformation.

3. Approximate the derivative of the state by a differentiation matrix $\mathbf{D}$, such that $\dot{x}_i(t_k) \approx \dot{x}_i^N(t_k) = \sum_{j=0}^{N} D_{kj} x_i^N(t_j)$ for $i = 1, 2, \ldots, N_x$. Here N_x is the number of state variables in $\boldsymbol{x}$.

4. Reformulate the optimization criteria (constraints and objective) in terms of the new states and discrete dynamics.

5. Solve the new optimization problem using a spectral algorithm (e.g. [78]).

Gauss, Radau and Lobatto collocation points

Collocation points are the points in the time domain where the optimization constraints are enforced. There are three main methods of selecting these points: Legendre-Gauss (LG), Legendre-Gauss-Radau (LGR) and Legendre-Gauss-Lobatto (LGL). The performance of these schemes are discussed in detail in the works of Garg et al. in [79]. The significant differences are that the LG, LGR and LGL collocation points lie on the open interval $\tau = (-1, 1)$, the half-open interval $\tau = [-1, 1)$ or $\tau = (-1, 1]$, and the closed interval $\tau = [-1, 1]$, respectively. This is illustrated in figures 5.10 and 5.11. One of the conclusions in [79] is that the methods using LG and LGR seemed to be more accurate than the method using LGL points.

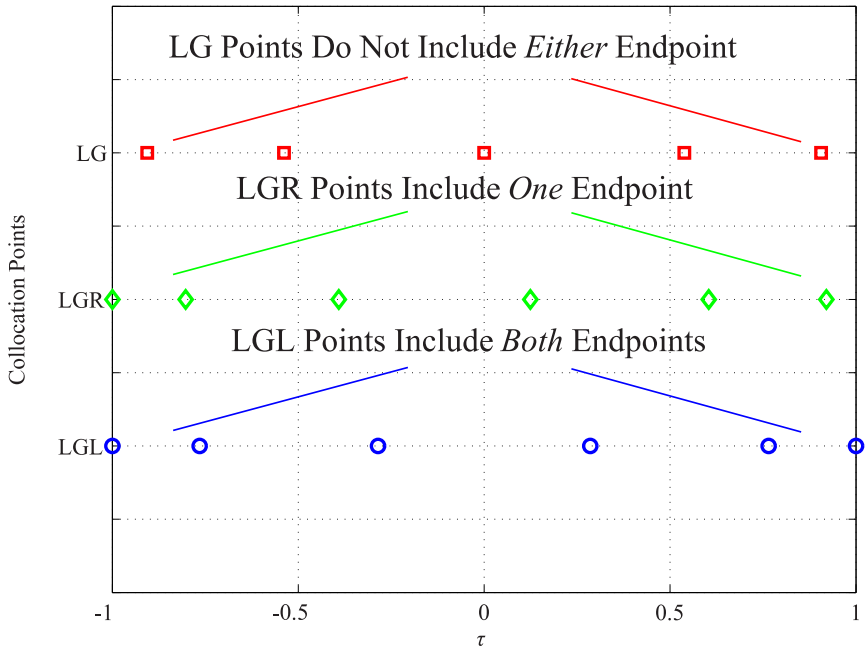


Figure 5.10: Difference between LG, LGR and LGL collocation points. From [79]. © 2009 D. Garg, M. A. Patterson, W. W. Hager, and A. V. Rao.

5.3.4 Optimization objective

The objective function (5.2a) decides in which respect the solution is optimized. Several choices may be made, such as minimum time, minimum length, and minimum energy. The latter is very relevant to the motion control of ferries, and interesting considering the main priorities from the introduction in this chapter.

In the different cases, the objective function has the following forms:

- Minimum time:

$$J(\boldsymbol{x}, \boldsymbol{u}, t_f) = \int_{t_0}^{t_f} 1 \, \mathrm{d}\tau = t_f - t_0 \tag{5.5}$$

- Minimum length (two-dimensional problem):

$$J(\boldsymbol{x}, \boldsymbol{u}, t_f) = \int_{t_0}^{t_f} \sqrt{\dot{x}^2 + \dot{y}^2} \, \mathrm{d}\tau \tag{5.6}$$

- Minimum energy:

$$J(\boldsymbol{x}, \boldsymbol{u}, t_f) = \int_{t_0}^{t_f} \left(|X u_r| + |Y v_r| + |N r| \right) \mathrm{d}\tau \tag{5.7}$$

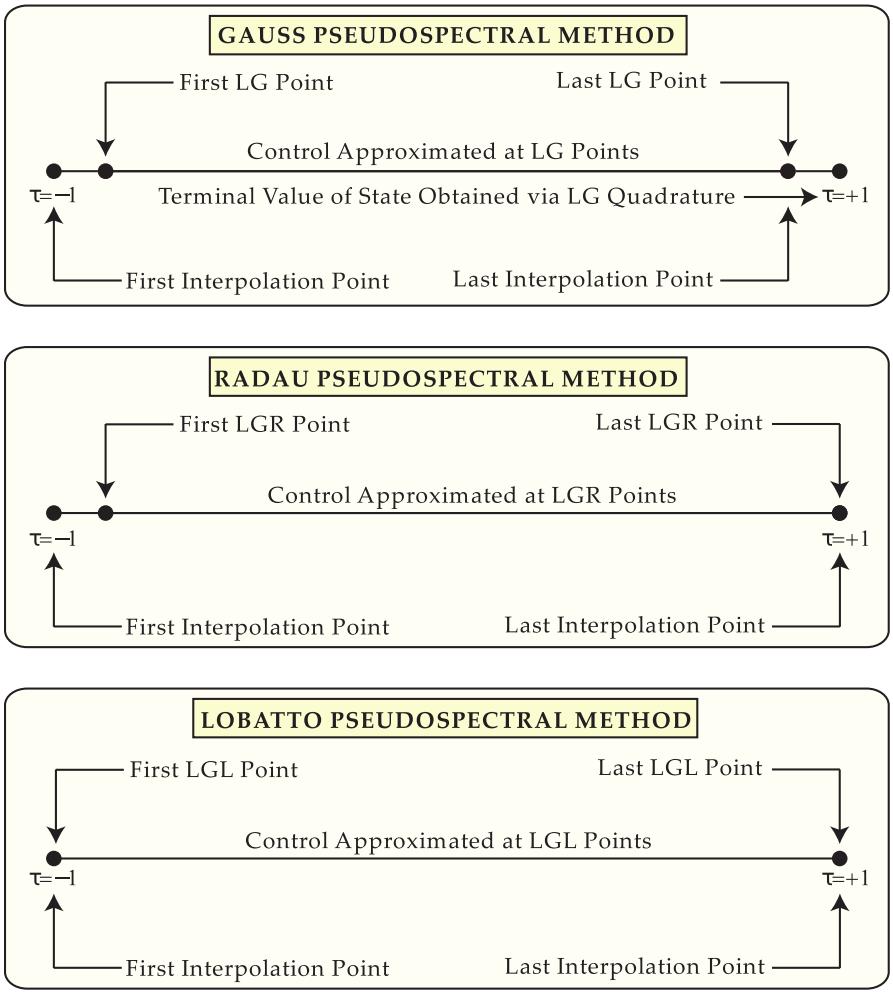


Figure 5.11: Difference between the Gauss, Radau and Lobatto PS methods. From [79]. © 2009 D. Garg, M. A. Patterson, W. W. Hager, and A. V. Rao.

Here X, Y, N and u_r, v_r, r are the elements of the vectors $\boldsymbol{\tau}$ and $\boldsymbol{\nu}_r$, respectively.

In addition to selecting an appropriate objective function, the constraints of the problem must be defined. The dynamic constraints are already described by (5.2b), but there are additional constraints. These may be saturations on the control vector $\boldsymbol{u}$, static obstacles which have to be represented in the inequality constraints (5.2c), and initial and end points on the path, P_s and P_f. The punctuality condition from the start of this chapter also imply an inequality constraint on the problem.

To summarize, the priority list from the start of the chapter may be connected to constraints and the objective function:

1. Safe — The inequality constraints from Definition 5.1 (5.2c) must be set to prohibit trajectories that cross obstacles.

3. Punctual — To make the ferry reach the harbor in time, the inequality constraints should also include some maximum end time.

4. Green — In order to make the trajectory *energy optimized*, some version of (5.7) may be included in the objective function (5.2a).

5.3.5 Models used for optimization

The obvious choice for (5.2b) is the 3 DOF dynamical model (3.39), which is repeated here for convenience:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}_z(\psi)\overbrace{(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c)}^{\boldsymbol{\nu}} \tag{5.8a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \boldsymbol{\tau}_{\text{wave}} \tag{5.8b}$$

The model has order 6 with the state vector $\boldsymbol{x} = [x, y, \psi, u_r, v_r, r]^\top$. An issue with using this model for optimization is the so-called *curse of dimensionality*, coined by Bellman [80].

The curse of dimensionality is the issue which arises when the computational complexity and memory requirements increases substantially with problem dimension. An alternative to the 6th order model is to use a model of order 5, where the turning rate is considered a command, rather than the result of applied moments. One may also remove the sway velocity v, by assuming that it is negligible, resulting in a 4th order model.

Reducing the model by removing sway and turning-rate dynamics decreases the computational complexity, but comes at the cost of not being able to account for energy used to compensate for currents and wind in these dimensions. Considering the holistic nature of the optimization methods, it is preferable to include these dynamics to exploit those advantages.

5.3.6 Data required for optimization

While it is relatively simple to formulate the models and constraints in an optimization problem, identifying and obtaining the actual data needed to be able to solve the problem is harder.

Firstly, obtaining an accurate ship model is difficult on its own. Finding accurate mass, Coriolis-centripetal, and drag matrices requires experimentation or advanced modeling. Approximations are commonly used in control design, and often yield satisfactory results.

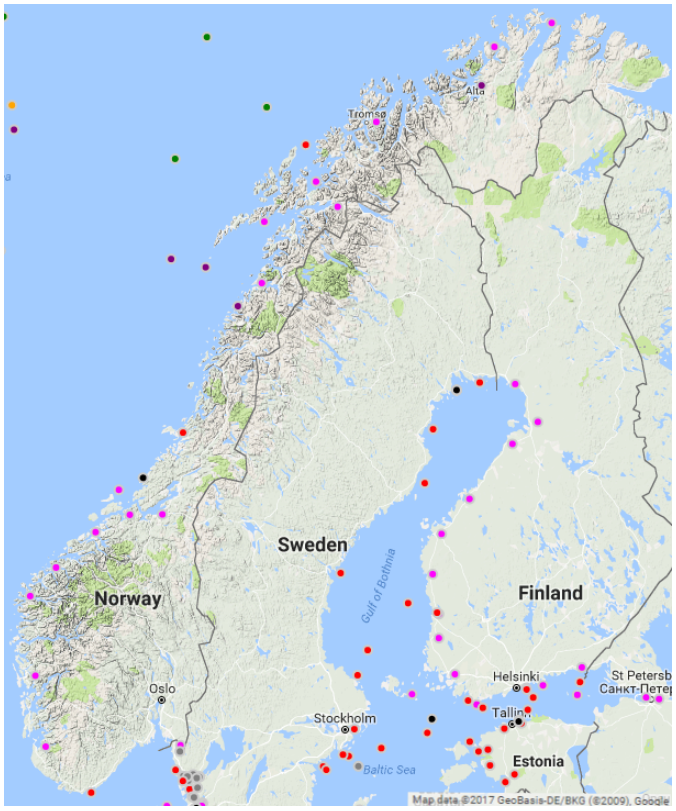


Figure 5.12: EMODnet sensor locations. Most entries are sea level sensors.

Secondly, obtaining accurate translations from wind speed to forces is generally hard. Several models are found in Fossen's book [34, Chapter 8]. Models for waves and their influence to external forces are also found there.

Thirdly, obtaining real-life up-to-date data on current velocity, and wave information is not trivial.

EMODnet

European Marine Observation and Data Network (EMODnet) is a source of information on ocean currents, wind and waves that could be potentially useful for real-time ferry operations [81]. It is a portal where data is collected from a large number of observatories around Europe, and is visualized and available through an application programming interface (API). Such data could be useful when initializing values of unknown parameters (such as current velocities of an area) to speed up the convergence of estimators.

Data for the Norwegian coast is sparse, as seen in Figure 5.12. Using such a portal with autonomous ferries will require new sensors to be connected to the

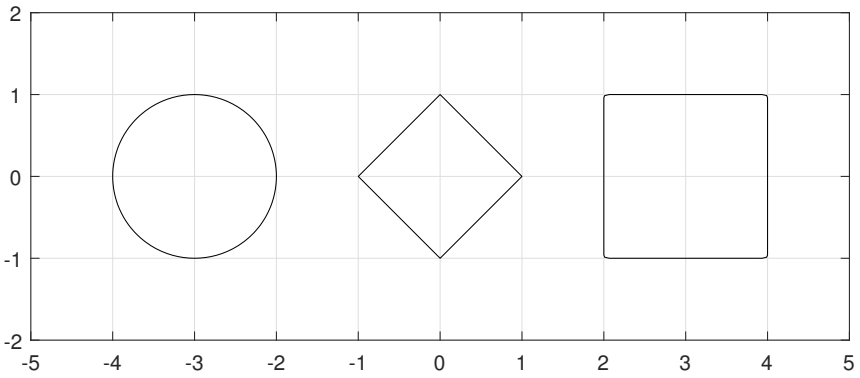


Figure 5.13: Differentiable obstacle representations. From left to right, (5.9), (5.10) and (5.11). The sizes in each direction are 1.

network.

Useful sensors which should be connected to the network for each ferry route are high frequency (HF) ocean current radars and anemometers for current and wind velocities, respectively.

5.3.7 Representation of static obstacles

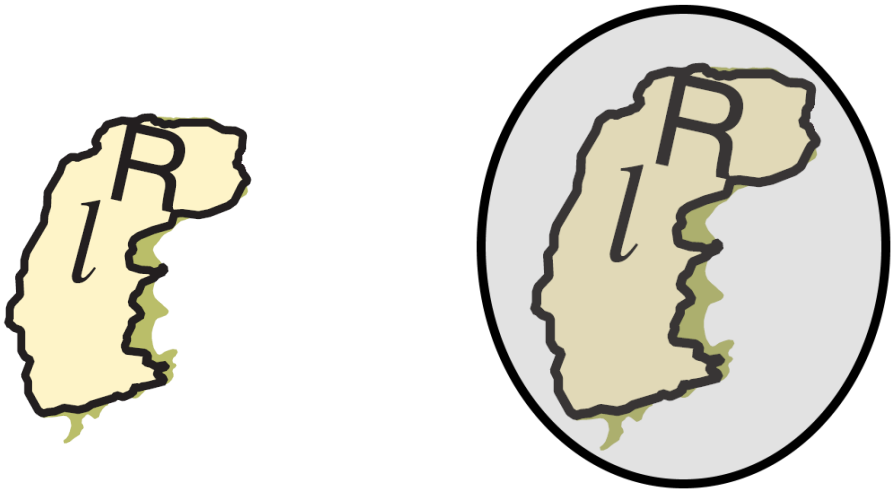
According to Gong et al. in [75], using differentiable algebraic functions to represent obstacles is computationally effective when using optimization methods. Variations of the *p-norm* may be used to represent ellipses and rectangles:

$$\text{Ellipse:} \quad \left(\frac{x - x_i}{x_{s,i}}\right)^2 + \left(\frac{y - y_i}{y_{s,i}}\right)^2 - 1 \geq 0 \tag{5.9}$$

$$\text{Skewed rectangle:} \quad \left(\frac{|x - x_i|}{x_{s,i}}\right) + \left(\frac{|y - y_i|}{y_{s,i}}\right) - 1 \geq 0 \tag{5.10}$$

$$\text{Straight rectangle:} \quad \left(\left(\frac{x - x_i}{x_{s,i}}\right)^P + \left(\frac{y - y_i}{y_{s,i}}\right)^P\right)^{\frac{1}{P}} - 1 \geq 0 \tag{5.11}$$

Here $P = 100$ for a sufficiently square-edged rectangle, and the variables x_i and y_i denote the center of obstacle i. The constants $x_{s,i}$ and $y_{s,i}$ indicate the size of each obstacle. They represent minor and major radii for the ellipse, half a diagonal for the skewed rectangle, and half a side for the straight rectangle. Figure 5.13 depicts the shapes.



(a) **(b)**

Figure 5.14: Map of Litla Bogøyna in Bergen. Shape (a) shows the island, which is in itself not easy to represent using generic functions. A solution is to draw an ellipse around it, shown in (b), which also serves as a safety zone around the island.

Figure 5.14 shows how an ellipse may serve as a representation of an island. The ellipse also serves as a safety zone around the obstacle. Including some navigable water in the ellipse would not affect the path planning algorithm, as this is an area the ferry would not enter anyway.

In addition to these shapes, Lewis and Ross cover methods to represent polygon approximations as differentiable functions in [82].

5.4 Comparison of roadmap and optimization methods

Roadmap and optimization methods are fundamentally different. While both require a representation of the static obstacles (a map) of the area, the way of finding a trajectory from P_s to P_f differs. Figure 5.15 gives an overview of the path planning methods.

In general the roadmap methods are computationally lighter than the optimization methods. The roadmap methods are well established in computer science and in robotics. Several well tested techniques of connecting points smoothly exist in the literature. However, it is not straight-forward to connect ship dynamics or energy consumption to most of these methods. The road map methods exploit

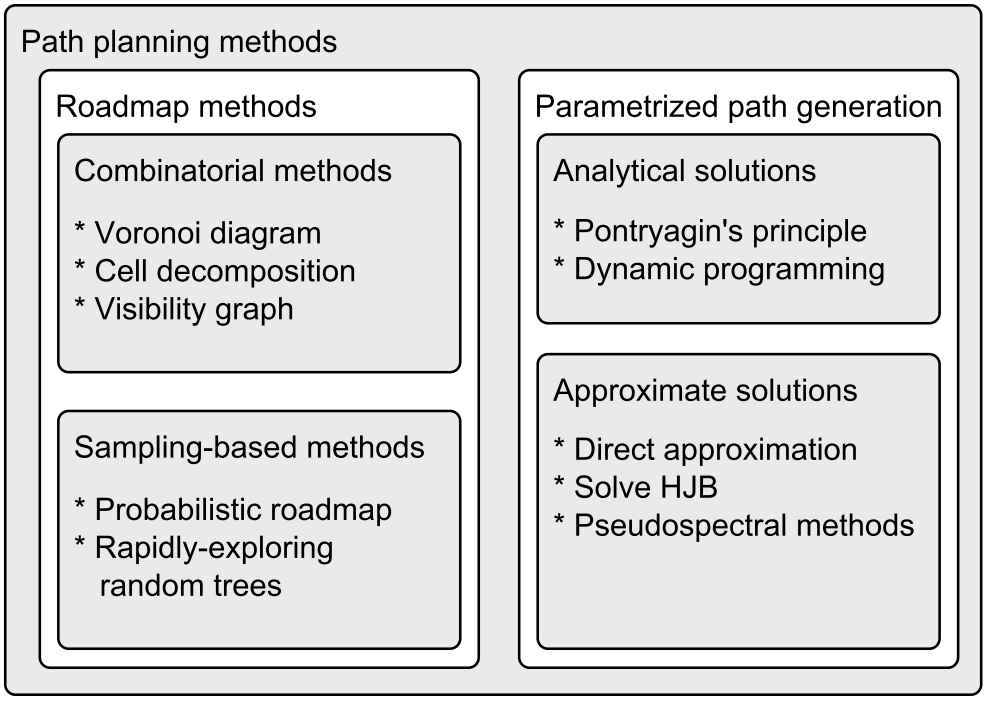


Figure 5.15: Overview of path planning methods. The figure is made with help from A. Lekkas and V. T. Wivestad.

a number of well thought out heuristics which produce a safe and feasible path for a vehicle to follow. After refinement the output from these methods produce straight lines connected with curves.

Optimization methods inherently include dynamics and saturation constraints. It is holistic in the sense that priorities of energy consumption, punctuality, and geographical constraints may be inserted into the problem statement intuitively. In addition, environmental influences, such as wind and ocean currents, have their natural place in the dynamic equations. The downside is that solving these dynamic optimization problems is computationally heavy, although pseudospectral methods have reduced this complexity substantially. The output of optimization methods is a "complete path," in the form $\boldsymbol{p}^d(\theta) = [x^d(\theta), y^d(\theta)]^\top$.

In conclusion, the optimization methods have significant advantages over the roadmap methods when it comes to energy optimized path planning, once the issues regarding computational complexity are overcome. Ever cheaper computing power and improvements in solver algorithms point to optimization methods as natural choices for path planning.

5.5 Path following

Once a path has been generated, either by the means of a roadmap and waypoints, or by solving an optimization problem, the next challenge is to follow the path with an underactuated ship. Fossen introduces several concepts for path following in Chapter 10 of [34]. The solutions are divided into path following for straight-line paths and curved paths. The straight-line path-following solutions work well with most roadmap methods, although they don't require any smoothing operations between the straight line segments. The algorithms used for straight-line path following are usually only dependent on a sequence of waypoints which describes the path.

Path following for curved paths requires some parametrization of the path from P_s to P_f. Algorithms designed to follow such paths are useful when the path is generated by an optimization scheme, as these methods output a path as a function of some path parameter θ, e.g.

$$\boldsymbol{p}^d(\theta) = \begin{bmatrix} x^d(\theta) \\ y^d(\theta) \end{bmatrix}.$$

5.5.1 Straight-line path following

Path following for straight-line paths are mostly based on LOS steering laws, which may be divided further into enclosure-based steering and lookahead-based steering. Both of these approaches seek to drive the cross track error e to 0.

In the 2D inertial reference frame $\{n\}$ a straight-line path is defined by the two waypoints $\boldsymbol{p}_k = [x_k, y_k]^\top$ and $\boldsymbol{p}_{k+1} = [x_{k+1}, y_{k+1}]^\top$. The constant α_k is then the path-tangential angle with respect to north, defined by

$$\alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k), \tag{5.12}$$

and illustrated in Figure 5.16. The function atan2 is the four-quadrant arctan-function. Geometrically it can be seen that the along-track distance and cross-track errors are described by

$$s = (x(t) - x_k)\cos\alpha_k + (y(t) - y_k)\sin\alpha_k \tag{5.13a}$$
$$e = -(x(t) - x_k)\sin\alpha_k + (y(t) - y_k)\cos\alpha_k \tag{5.13b}$$

In addition to the algorithms mentioned in this section, some sources for straight-line path following include an article by Breivik and Fossen [83], Breivik's PhD thesis [84], and a paper by Pettersen and Lefeber [85].

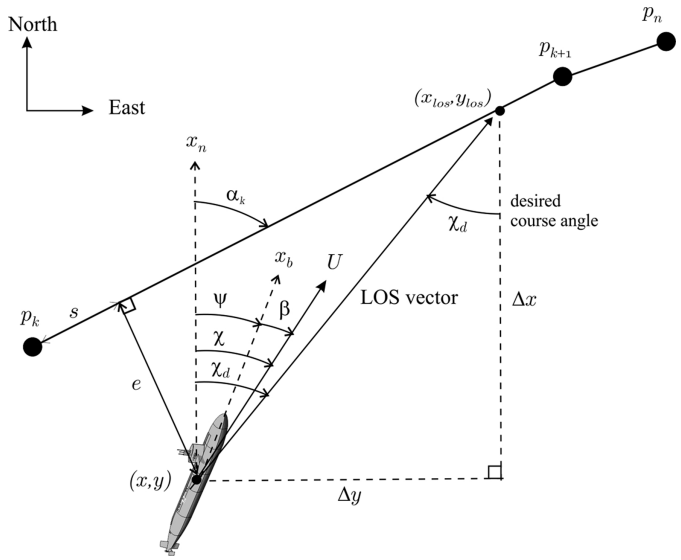


Figure 5.16: LOS guidance parameters. From [34]. © 2011 John Wiley & Sons, Ltd.

Enclosure-based steering

Enclosure-based steering involves placing an imaginary circle around the vehicle's position with radius R, and pointing the desired course χ^d towards the intersection between this circle and the line from $\boldsymbol{p}_k$ to $\boldsymbol{p}_{k+1}$. The circle will intersect at two points, and the point which is closest to $\boldsymbol{p}_{k+1}$ is chosen. The intersecting point is denoted $\boldsymbol{p}_{los} = [x_{los}, y_{los}]^\top$.

A steering law is introduced:

$$\chi^d = \text{atan2}(y_{los} - y(t), x_{los} - x(t)) \tag{5.14}$$

To calculate the unknown intersection point, the following equations must be solved:

$$(x_{los} - x(t))^2 + (y_{los} - y(t))^2 = R^2 \tag{5.15}$$

$$\begin{aligned} \tan \alpha_k &= \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \\ &= \frac{y_{los} - y_k}{x_{los} - x_k} \end{aligned} \tag{5.16}$$

A thorough derivation of the solution of these equations is found in Fossen's book [34, Section 10.3.2]. A notable disadvantage is that this approach requires $R \geq |e|$.

Lookahead-based steering

The lookahead-based approach is motivated by missile guidance, and has several advantages over the enclosure-based approach [34]. It is valid for all cross-track errors, and it is less computationally expensive.

Lookahead-based steering separates the desired course in two parts:

$$\chi^d = \chi_p + \chi_r(e) \tag{5.17}$$

where

$$\chi_p = \alpha_k. \tag{5.18}$$

The *velocity-path relative angle* χ_r is set to

$$\chi_r(e) = \arctan\left(-\frac{e}{\Delta}\right), \tag{5.19}$$

where Δ is the lookahead distance.

The steering law can be extended to include integral action to compensate for ocean currents and other disturbances:

$$\chi_r(e) = \arctan\left(-\frac{e}{\Delta} - K_i \int_0^t e(\tau)\,\mathrm{d}\tau\right) \tag{5.20}$$

Waypoint manager

Common to both straight-line path-following laws is the need of a waypoint manager. The waypoint manager's responsibility is to propagate the waypoints so that the correct line-segment is followed. The basic task is to increment to the next waypoint once the vessel is close enough to the previous waypoint in some sense.

A simple switching condition is introduced: Circle of acceptance. The circle of acceptance switches to the next waypoint once the ship is within a circle of the current goal waypoint:

$$\left\|\boldsymbol{p}_{k+1} - \boldsymbol{p}(t)\right\|_2^2 \leq R^2 \tag{5.21}$$

5.5.2 Path following for curved paths

With respect to optimization methods, path following for parametrized curved paths are more interesting than straight-line methods. The reason is as stated earlier that optimization methods don't necessarily produce a sequence of waypoints, but two-dimensional function of a path parameter, namely

$$\boldsymbol{p}^d(\theta) = \begin{bmatrix} x^d(\theta) \\ y^d(\theta) \end{bmatrix}. \tag{5.22}$$

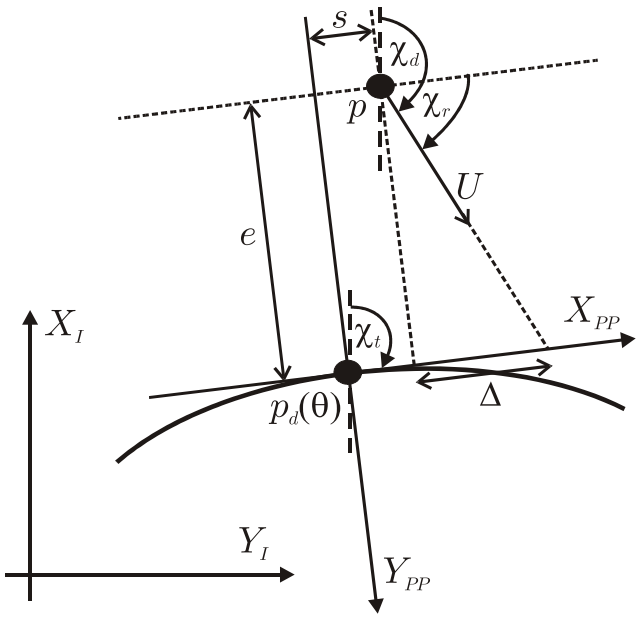


Figure 5.17: Geometric relationships in path-following controller from [86]. © 2004 IEEE.

A path-following controller for this situation motivated by LOS methods is presented by Breivik and Fossen in [86]. It is summarized here:

The along-track and cross-track errors are

$$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} \cos\chi_t & \sin\chi_t \\ -\sin\chi_t & \cos\chi_t \end{bmatrix} \begin{bmatrix} x - x^d \\ y - y^d \end{bmatrix}, \tag{5.23a}$$

and the path-tangential angle is

$$\chi_t(\theta) = \arctan\frac{y^{d\prime}(\theta)}{x^{d\prime}(\theta)}. \tag{5.23b}$$

The control law is

$$\chi_r(e) = \arctan\left(-\frac{e}{\Delta}\right), \tag{5.23c}$$

and the desired course is

$$\chi^d(\theta, e) = \chi_t(\theta) + \chi_r(e). \tag{5.23d}$$

The path parameter dynamics are

$$\dot{\theta} = \frac{U\cos(\chi(t) - \chi_t(\theta)) + \gamma s}{\sqrt{x^{d\prime}(\theta)^2 + y^{d\prime}(\theta)^2}}, \tag{5.23e}$$

with the requirements

$$\Delta > 0;\ \gamma > 0;\ U \geq U_{\min} > 0. \tag{5.23f}$$

The design parameter Δ is can be thought of as the lookahead-distance, as in the straight-line path-following controller in Section 5.5.1. The other design parameter γ controls the speed of the dynamics of a virtual target projected on to the path. Figure 5.17 illustrates the geometric relationships of the course and track parameters. The low-level controller from Section 3.3.1 takes desired heading ψ^d as input, which requires the use of sideslip compensation. This is done by assigning

$$\psi^d = \chi^d + \beta \tag{5.24}$$

This path-following controller has the origin $[s, e]^\top = \mathbf{0}$ proven to be UGAS and uniformly locally exponentially stable (ULES). From (5.23e) we also see that the path parameter θ increases as long as the along-track error s is non-negative. This implies both that a ship with course χ^d and speed U will follow the path (5.22), and that θ will increase, so that the ship moves from P_s to P_f.

Micaelli and Samson introduce an alternative approach based on a unicycle-like robot in [87]. Other resources for curved-path control include the works by Samson [88], and an article by Lapierre et al. [89].

Chapter 6

Optimal Path Planning and Control for a 3-DOF Ship Model Using DIDO

Path planning using pseudospectral optimization algorithms is examined in Chapter 5. The idea is to exploit a dynamical model to create a feasible path which is optimal in some sense. This chapter documents experiments performed with the optimal control software DIDO by Elissar Global. Additionally, results from simulations where a guidance controller is implemented to follow the generated paths are presented.

6.1 About DIDO

DIDO is optimal control software developed by I. Michael Ross and Elissar Global [76]. It is a numerical optimization extension to Matlab, and is based on pseudospectral methods. DIDO uses LGL collocation points with Legendre polynomials to solve the optimal control problem [90]. After the problem is coded in the DIDO format, the algorithm spends some time before outputting a solution containing the control action, states and cost, along with information about the dual problem. DIDO recognizes the solution as one of these four classifications:

Extremal: A feasible local minimum is found within the algorithm tolerance.

Good: A feasible solution close to or at a local minimum is found.

Suboptimal: A feasible solution is found which is not optimal.

Infeasible: DIDO couldn't find a solution within the given constraints.

6.2 Optimization setup

This section contains information about how the DIDO environment is set up. The experiments are run on a desktop computer with a 3.20 GHz 4-core Intel i5 CPU with 16 GB RAM and no GPU. The operating system is Windows 7 64-bit.

6.2.1 Obstacle representation

In the experiments which contain obstacles, they are represented as inequality constraints. Some examples are found in Section 5.3.7. DIDO supports these constraints as *path constraints*. A function $c_i(\boldsymbol{x}) \geq 0$ represents obstacle i, and is constrained to an interval $[0, C_i]$, where $C_i > 0$ is a large constant. DIDO does not support infinite intervals, thus the large constant is needed.

6.2.2 Dynamic model

To apply dynamic constraints, the model in (3.39) is used, repeated here:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi) \overbrace{(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c)}^{\boldsymbol{\nu}} \tag{6.1a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau} \tag{6.1b}$$

Wind and wave forces are not taken into account, and current forces are contained in the model by using relative velocities. The thruster model used is the *transit model* (3.37) from Section 3.1.4:

$$\boldsymbol{\tau}(\boldsymbol{u}) = \begin{bmatrix} F\cos\delta \\ F\sin\delta \\ -l_1 F\sin\delta \end{bmatrix} \tag{6.2}$$

Limits are placed such that $F \in [0, 8\,\mathrm{N}]$ and $\delta \in [-30°, 30°]$. This gives a maximum surge speed of approximately 1.0 m/s. These low values are due to the small-scale model being used, as noted in the next paragraphs.

The model is transformed to state-space form, by selecting $\boldsymbol{x} = [\boldsymbol{\eta}^\top, \boldsymbol{\nu}_r^\top]^\top \in \mathbb{R}^{N_x}$ where $N_x = 6$. The dynamic equations become

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{\eta}} \\ \dot{\boldsymbol{\nu}}_r \end{bmatrix} = \begin{bmatrix} \mathbf{R}(\psi)\,(\boldsymbol{\nu}_r + \boldsymbol{\nu}_c) \\ \mathbf{M}^{-1}\left(-\mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r - \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \boldsymbol{\tau}\right) \end{bmatrix} \tag{6.3}$$

Parameters and scaling

The model parameters used in the experiments are from the work of Sørensen et al. in [38]. The parameters are found in Table A.1, and is from a small-scale model ship called CyberShip II. CyberShip II has a length of 1.255 m and weighs 23.8 kg. Appendix A contains the complete model and parameters, as well as a section on how to perform upscaling of the parameters to a larger ship. The experiments use the original small ship model.

6.2.3 Variable scaling and balancing

In DIDO's user manual it is recommended that the magnitude of each state variable is smaller than 10 and larger than 1 on average. For the state-space model (6.3) this may not be feasible. E.g. the ship may travel hundreds or thousands of meters during an experiment, and thus the state variables will grow significantly. A way to mend this is to introduce *designer units*. The state $\bar{\boldsymbol{x}} = \mathbf{S}_{\boldsymbol{x}}\boldsymbol{x}$ is used, where $\mathbf{S}_{\boldsymbol{x}} = \text{diag}\left\{\frac{1}{U_x}, \frac{1}{U_y}, \frac{1}{U_\psi}, \frac{1}{U_u}, \frac{1}{U_v}, \frac{1}{U_r}\right\}$. The positive constant $U_i > 0$ for state $i \in \{x, y, \psi, u, v, r\}$ is the designer unit for that state. It is used to scale the expected value of state i up or down to the desired magnitude.

Scaling is also done for time, where the relationship $\bar{t} = S_t t,\ S_t = \frac{1}{U_t}$ is used to scale the time to a reasonable magnitude (same as states). This creates a new relationship for the derivative with respect to time:

$$\dot{\bar{\boldsymbol{x}}} = \frac{\mathrm{d}\bar{\boldsymbol{x}}}{\mathrm{d}\bar{t}} = \frac{\mathrm{d}\mathbf{S}_{\boldsymbol{x}}\boldsymbol{x}}{\mathrm{d}S_t t} = \frac{\mathbf{S}_{\boldsymbol{x}}}{S_t}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \frac{\mathbf{S}_{\boldsymbol{x}}}{S_t}\dot{\boldsymbol{x}} \tag{6.4}$$

In addition to scaling on states and time, scaling on the cost function J is also performed, which has improved feasibility of the solutions. A new cost $\bar{J} = S_J J$ is introduced, where $S_J = \frac{1}{U_J}$.

In the process of solving the optimal control problem, DIDO generates as many *costates* as there are variables. The costates $\boldsymbol{\lambda} \in \mathbb{R}^{N_x}$ are a part of the *dual problem*, and more information on the dual problem is found in Ross' book [91]. For a fast and accurate solution of the optimal control problem, it is important that the states and costates are *balanced*. This means that each state x_i and its costate λ_i has approximately the same magnitude. After scaling the variables and cost using designer units, $\bar{\boldsymbol{\lambda}}$ also changes according to the relationship

$$\bar{\boldsymbol{\lambda}} = S_J \mathbf{S}_{\boldsymbol{x}}^{-1}\boldsymbol{\lambda} \tag{6.5}$$

Thus a combination of scaling the states to a reasonable magnitude and scaling the cost is performed, so that the costates are balanced.

6.2.4 Using an initial guess

Some of the experiments result in infeasible solutions, although it would seem that the problem is actually feasible. DIDO may struggle to satisfy the dynamic constraints for complex state-space equations. Use of an initial guess is supported, and results have shown that it increases the chance of a feasible solution for a complex problem.

In the experiments where an initial guess is used, a Runge-Kutta (RK) simulation of the dynamic system is performed, where the initial conditions represent a stationary ship pointing in an arbitrary direction. Force in the surge direction is applied from start, and the simulation is run for a set time. The states, time and control action are recorded for use as an initial guess. The trajectory is sampled as close as possible to the LGL collocation points, and scaled using the designer units mentioned in Section 6.2.3, and fed to the DIDO algorithm before solving the optimal control problem.

Using an initial guess introduces a bias to the solution. If a local minimum exists close to the initial guess, it is likely that DIDO will find this solution. It is therefore important to either use an initial guess which is in the proximity of where it is believed an actual optimal solution lies, or to use several initial guesses to compare the costs if the solutions differ.

6.2.5 Objective function

Three different cost functions are mentioned in Section 5.3.4. Of those, the following two functions are used in the experiments:

Minimum time:

$$J = t_f \tag{6.6}$$

This function minimizes the end time t_f, which when optimized, will provide the fastest possible route to the end goal.

Minimum energy:

$$J = \int_0^{t_f} \left(|Xu_r| + |Yv_r| + |Nr| \right) \mathrm{d}\tau \tag{6.7}$$

Some criticism to this cost function is that it is not necessarily energy spent to move the ship, but rather *work* done by the actuator, which will be a lower value than energy spent. This function has a discontinuous derivative when either of the included variables are 0, which sometimes give a *good* or *suboptimal* solution, rather than *optimal*. Some information on the related L^1 optimal control problem, which includes absolute value operations in the cost function, is available in [91].

6.2.6 Testing feasibility

For each solution DIDO creates, the feasibility is tested by applying the generated control signals to a RK simulation of the model. The trajectory from the optimal control solution is compared to the simulated trajectory, and should overlap. None of the experiments shown here have any significant deviation from the simulated trajectories.

6.2.7 Result figures

The results include the following figures:

Path: Map with obstacle and path, including scaled ship polygons evenly spaced. Current direction is indicated with an arrow.

Pose: Four time series, consisting of north and east position, heading and sideslip.

Absolute velocities: Four time series, consisting of ground speed, surge and sway velocities, and yaw rate.

Relative velocities: Three time series, consisting of relative speed and relative surge and sway velocities.

Controls: Thruster force and angle as time series.

For the obstacle-free experiments, the two cost functions minimum energy and minimum time are shown superimposed on the path plot, and side by side on the other plots. The maneuver times are different for the minimum-energy and minimum-time runs, thus the time-axis is labeled accordingly. Minimum energy uses a green color, and minimum time uses blue. The plots use spline interpolations of the collocation points produced by DIDO.

6.3 Obstacle-free scenarios

These scenarios are run with two cost functions: Minimum time and minimum energy. The initial absolute velocities are zero, and are run with and without current. The scenario conditions are found in Table 6.1. The maximum final time $t_{f,\max}$ has to be set for each optimization run. For the time-optimal cost function, this value is insignificant as long as it is high enough, as the objective is to reach the end condition in minimum time. For the energy-optimal cost function however, the algorithm will use as much of the available time as possible, thus the setting of $t_{f,\max}$ is significant. For the obstacle-free scenarios we have $t_{f,\max} = 30\,\mathrm{s}$. The optimization algorithm is run with 30 nodes for every solution.

Table 6.1: Obstacle-free scenario conditions.

Scenario	Section	Initial conditions $(x_0, y_0, \psi_0, u_0, v_0, r_0)$ $(\mathrm{m}, \mathrm{m}, ^\circ, \mathrm{m/s}, \mathrm{m/s}, ^\circ/\mathrm{s})$	Final conditions (x_f, y_f, ψ_f) $(\mathrm{m}, \mathrm{m}, ^\circ)$
Short diagonal	6.3.1	$(-1, -1, 0, 0, 0, 0)$	$(1, 1, 180)$
Short U-turn	6.3.2	$(0, -1, 0, 0, 0, 0)$	$(0, 1, 180)$
Long straight	6.3.3	$(0, -10, 90, 0, 0, 0)$	$(0, 10, \sim)$

6.3.1 Short diagonal

This scenario has the goal of moving in a 90 degree curve while turning 180 degrees across a diagonal of $2\sqrt{2}\,\mathrm{m}$.

No current

The results are found in Table 6.2 and in figures 6.1 through 6.5.

Table 6.2: Short diagonal, no current: Results.

Current velocity		
V_x	0	m/s
V_y	0	m/s
Minimum-energy run		
Cost	8.633	J
t_f	30	s
Run time	38	s
Initial guess	No	
Classification	Good	
Minimum-time run		
t_f	6.738	s
Run time	16	s
Initial guess	No	
Classification	Extremal	

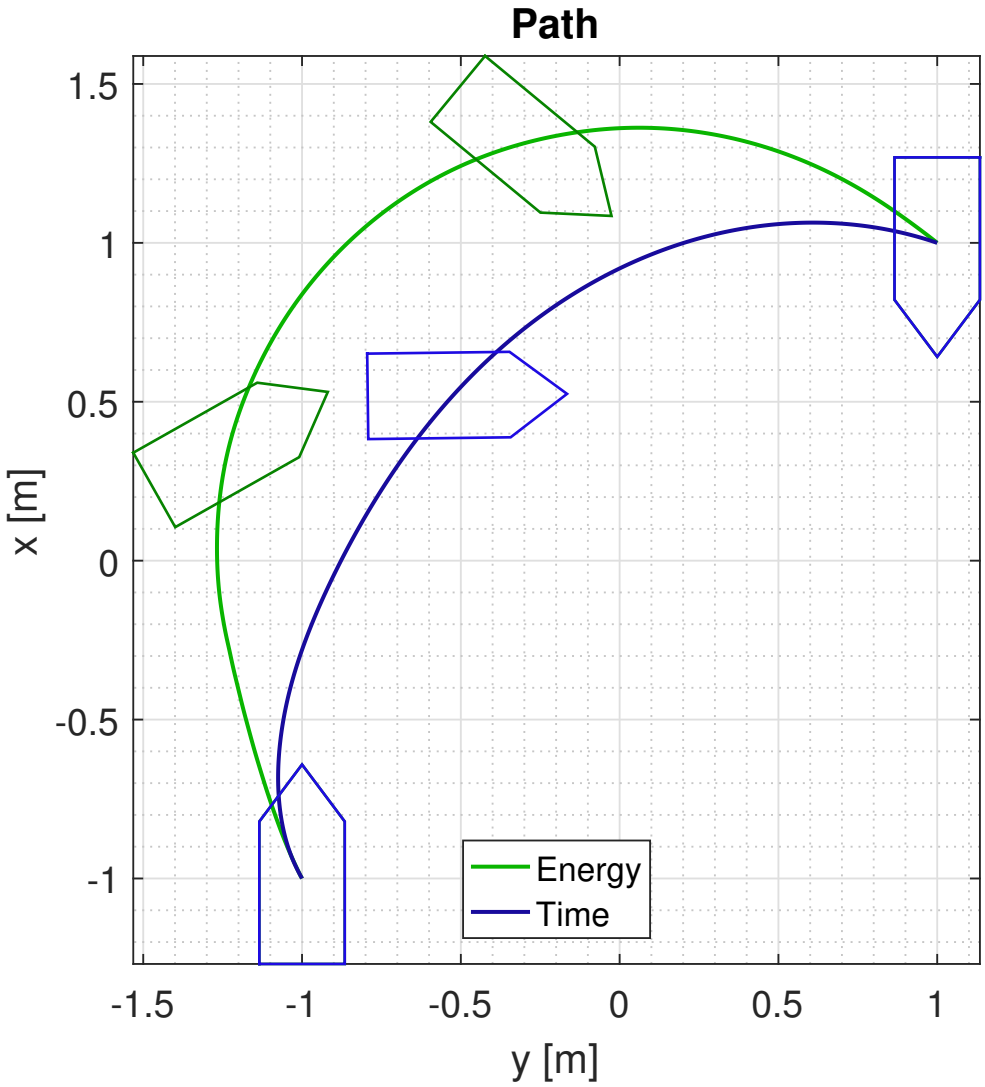


Figure 6.1: Short diagonal, no current: Path.

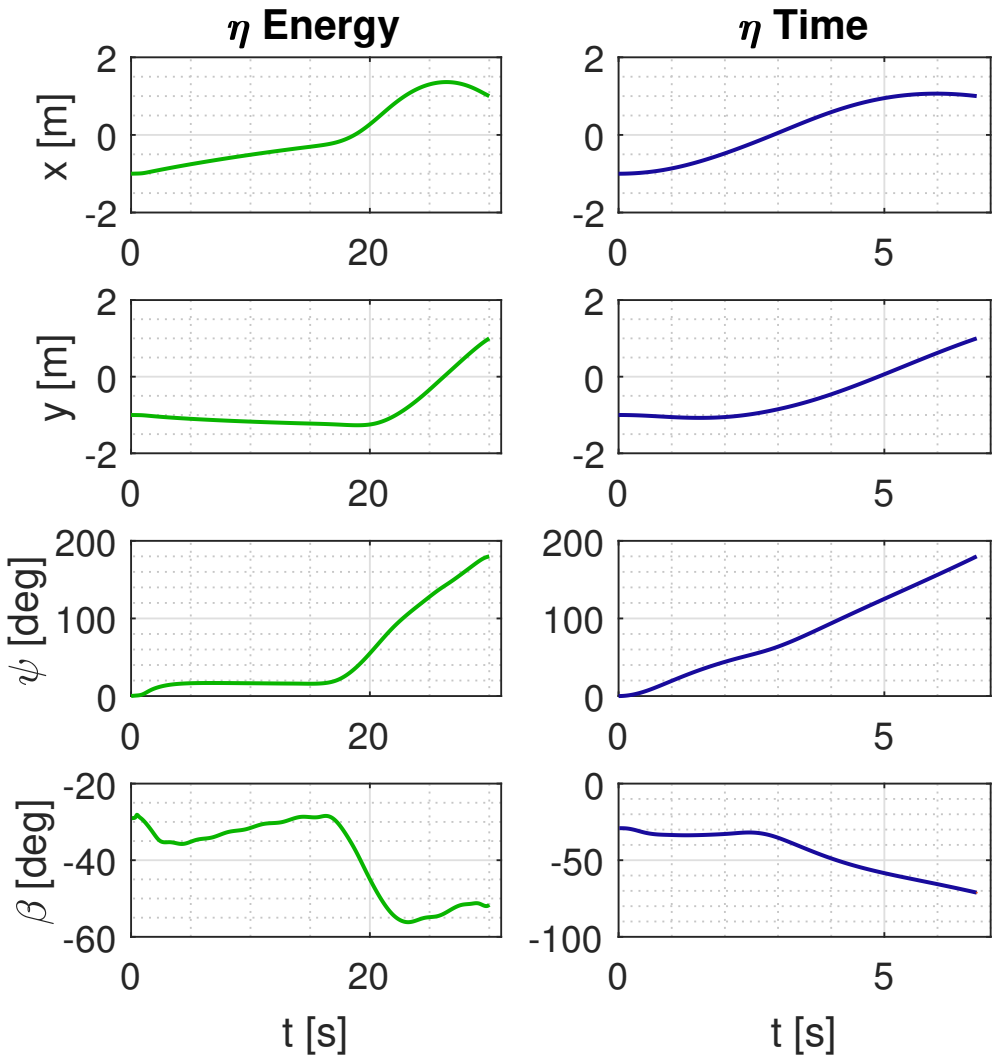


Figure 6.2: Short diagonal, no current: Pose.

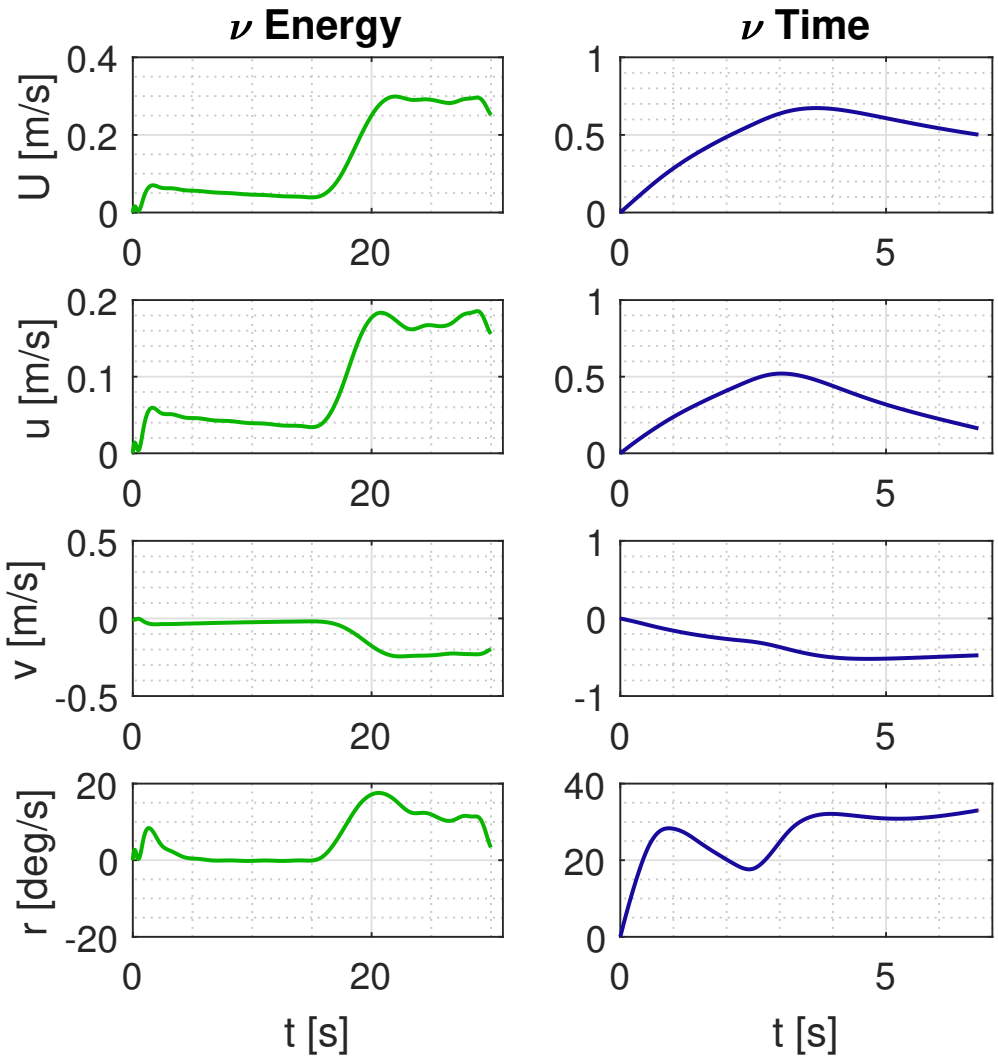


Figure 6.3: Short diagonal, no current: Absolute velocities.

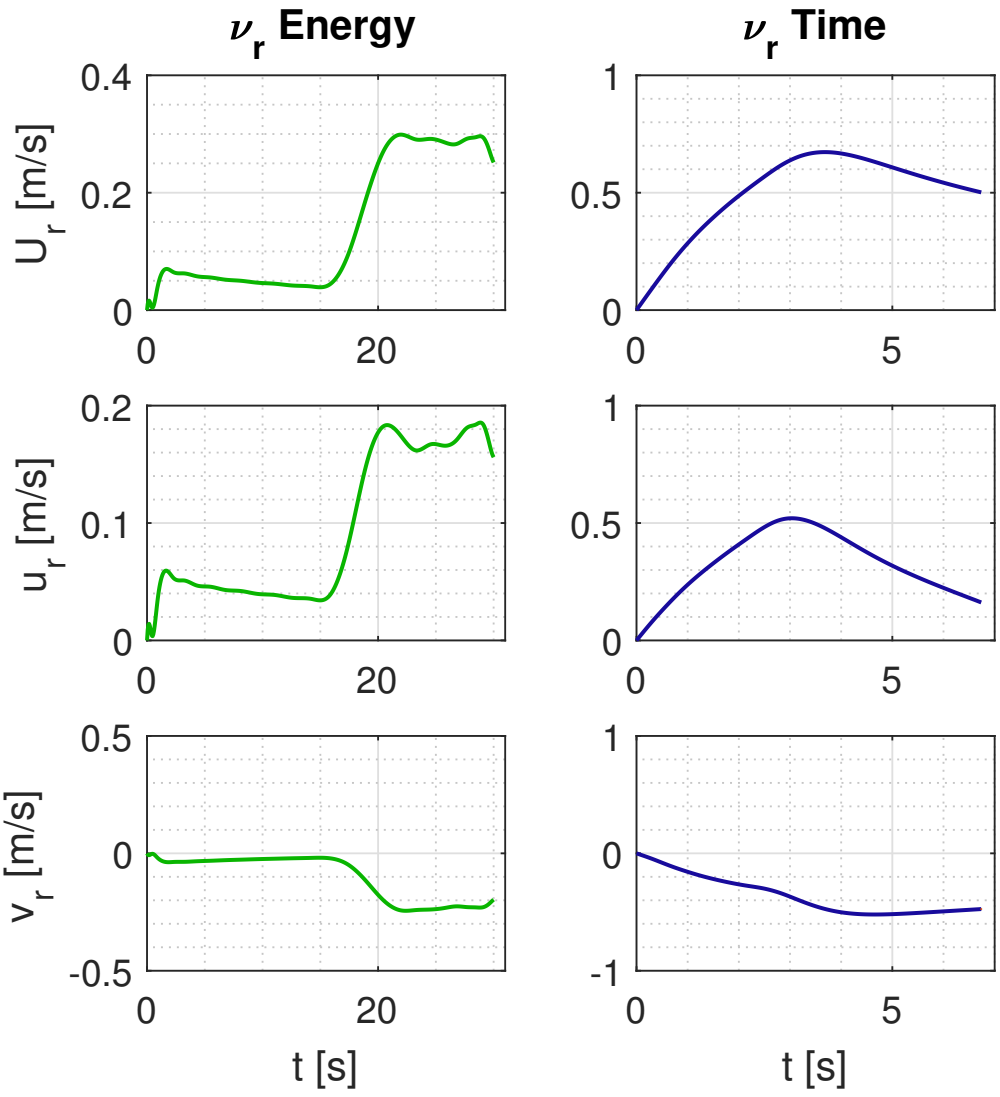


Figure 6.4: Short diagonal, no current: Relative velocities.

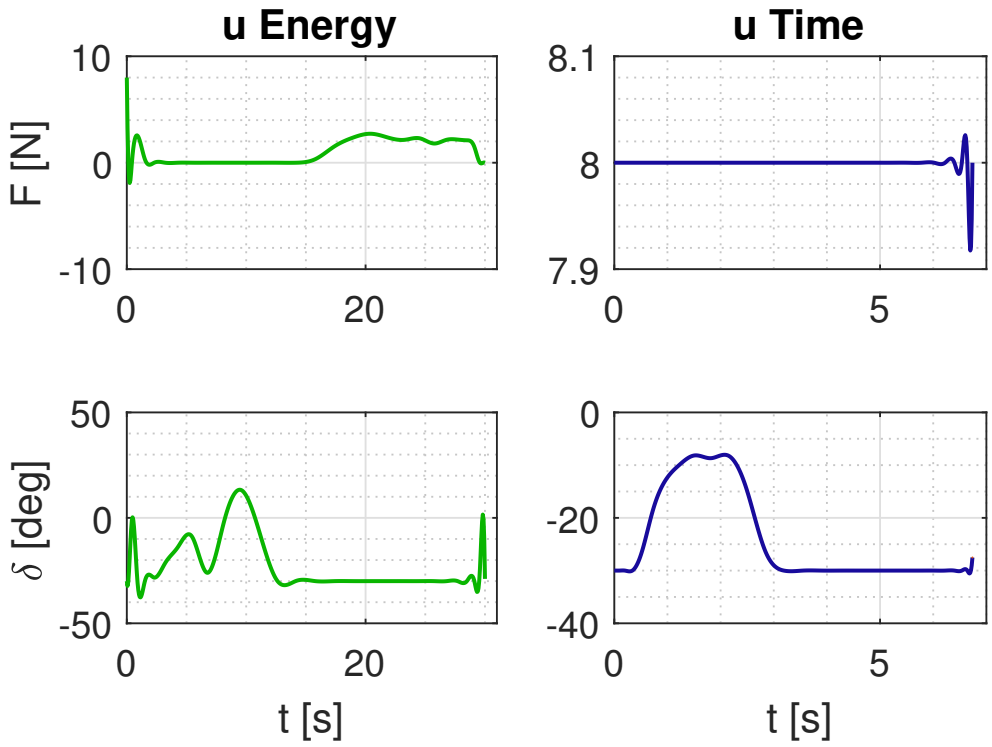


Figure 6.5: Short diagonal, no current: Control inputs.

Northeastern current

The results are found in Table 6.3 and in figures 6.6 through 6.10.

Table 6.3: Short diagonal, northeastern current: Results.

Current velocity		
V_x	0.15	m/s
V_y	0.15	m/s
Minimum-energy run		
Cost	8.768	J
t_f	10.64	s
Run time	28	s
Initial guess	Yes	
Classification	Suboptimal	
Minimum-time run		
t_f	6.609	s
Run time	25	s
Initial guess	No	
Classification	Extremal	

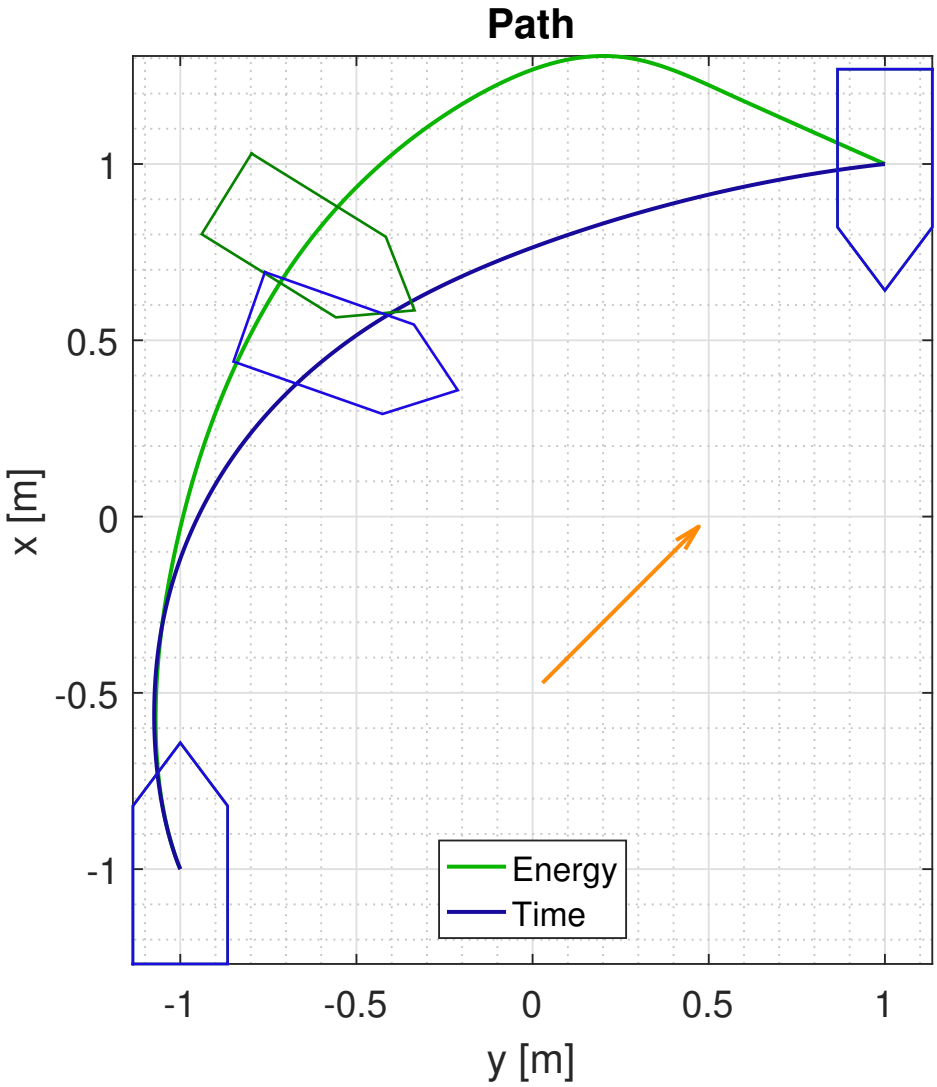


Figure 6.6: Short diagonal, northeastern current: Path.

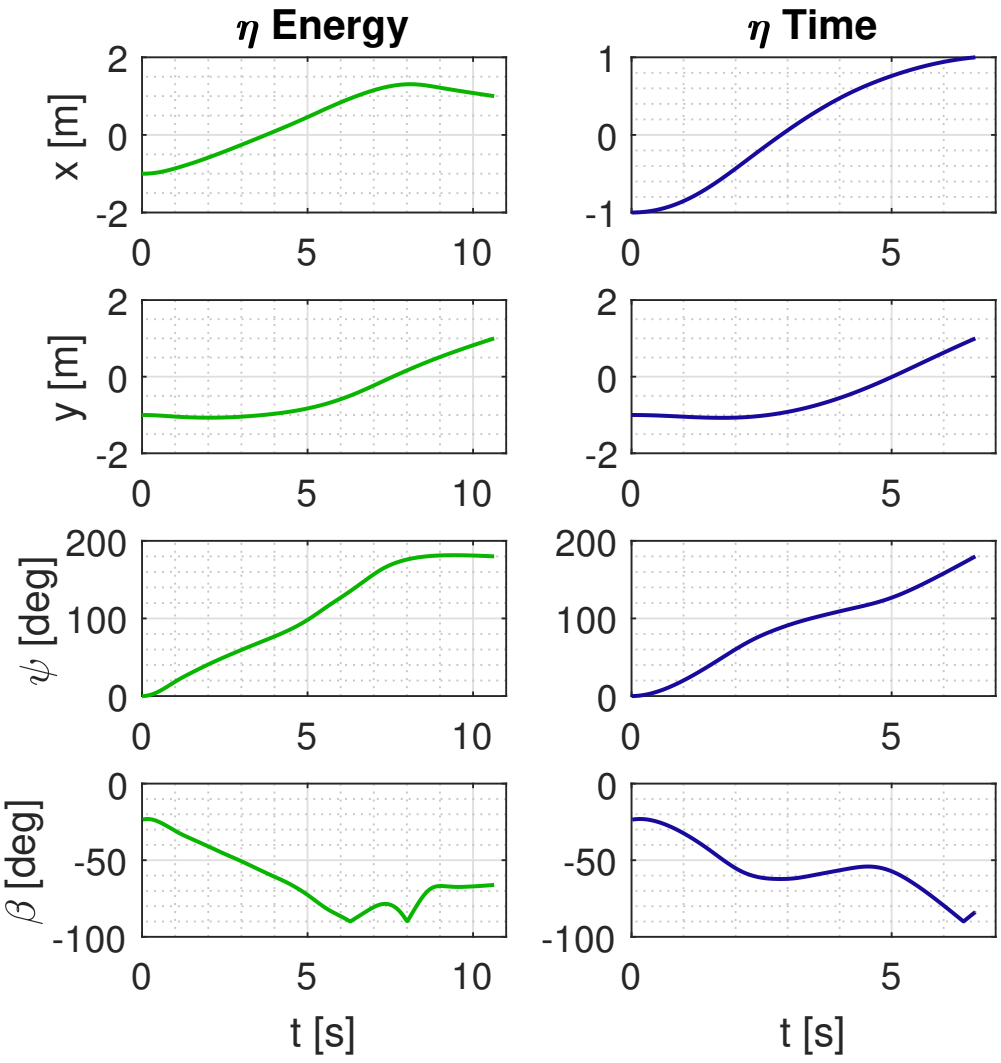


Figure 6.7: Short diagonal, northeastern current: Pose.

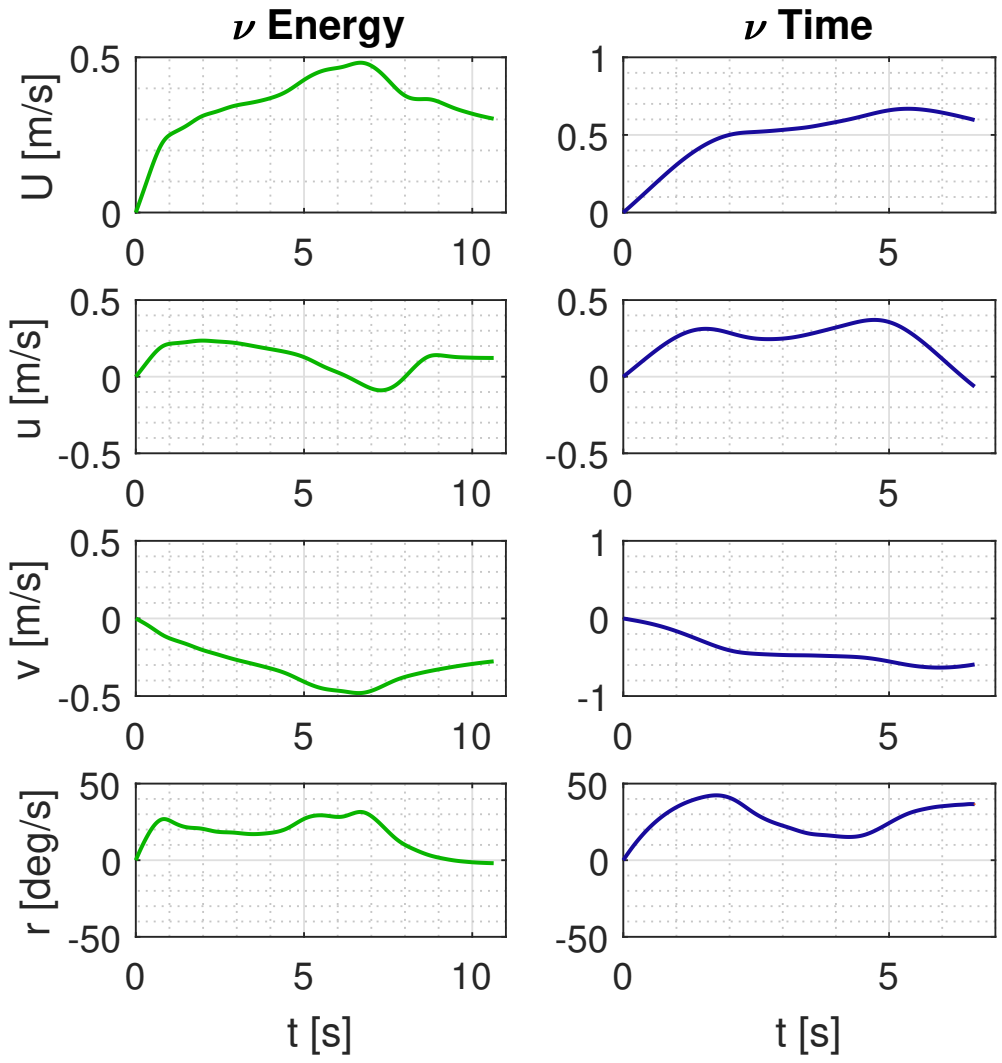


Figure 6.8: Short diagonal, northeastern current: Absolute velocities.

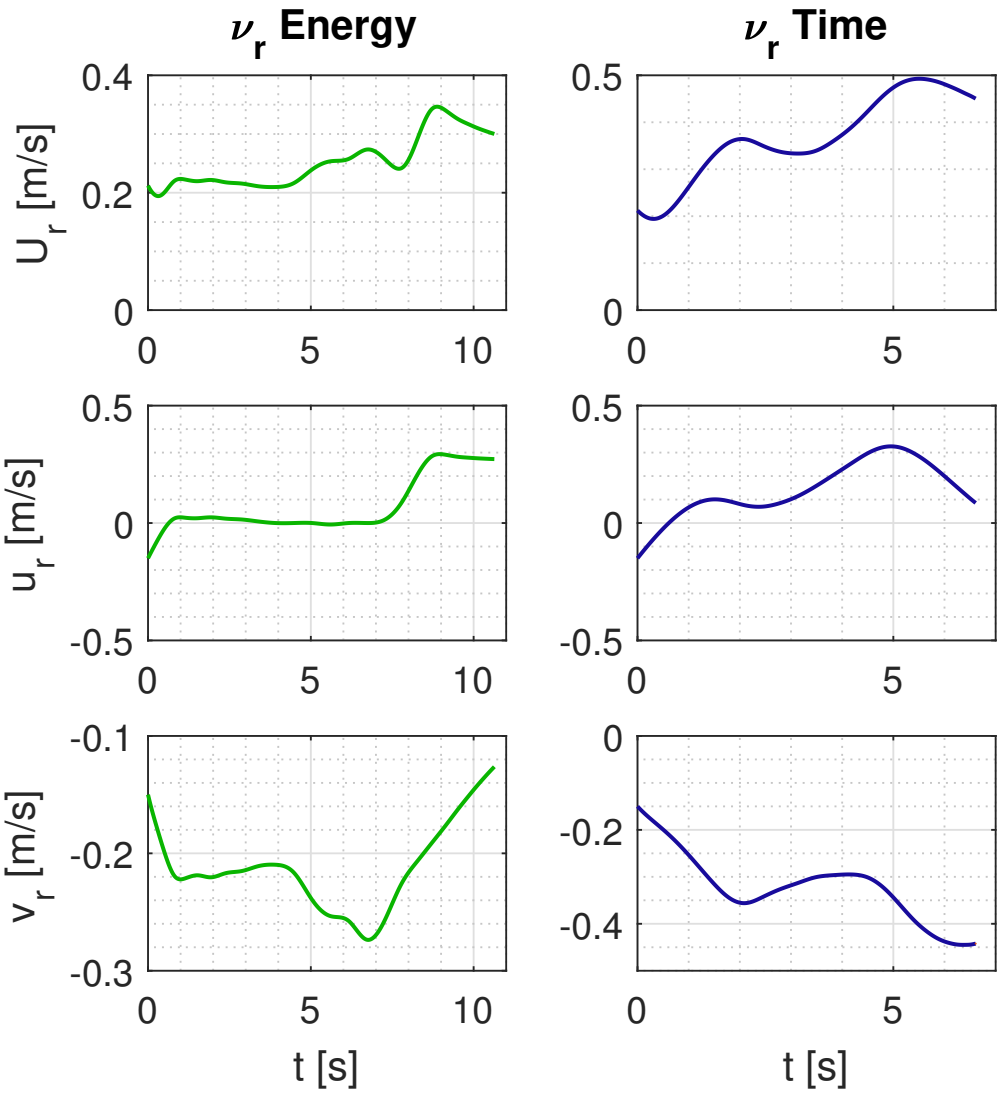


Figure 6.9: Short diagonal, northeastern current: Relative velocities.

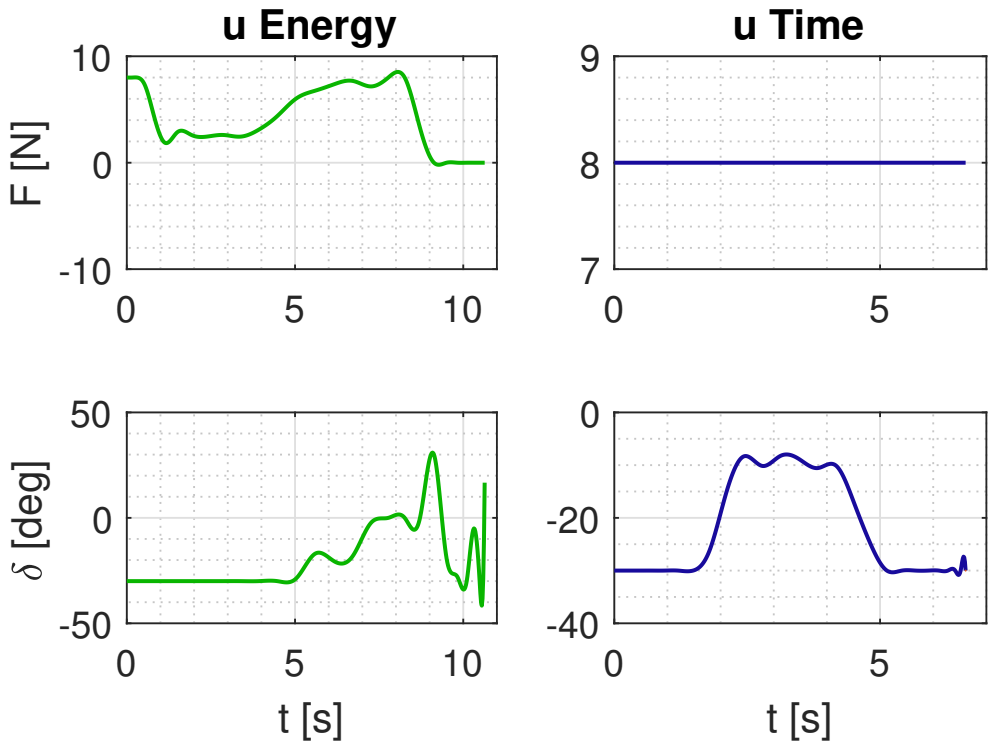


Figure 6.10: Short diagonal, northeastern current: Control inputs.

6.3.2 Short U-turn

This scenario has the goal of performing a U-turn with an offset of 2 m east.

No current

The results are found in Table 6.4 and in figures 6.11 through 6.15.

Table 6.4: Short U-turn, no current: Results.

Current velocity		
V_x	0	m/s
V_y	0	m/s
Minimum-energy run		
Cost	10.34	J
t_f	20.5	s
Run time	17	s
Initial guess	Yes	
Classification	Suboptimal	
Minimum-time run		
t_f	7.778	s
Run time	21	s
Initial guess	No	
Classification	Good	

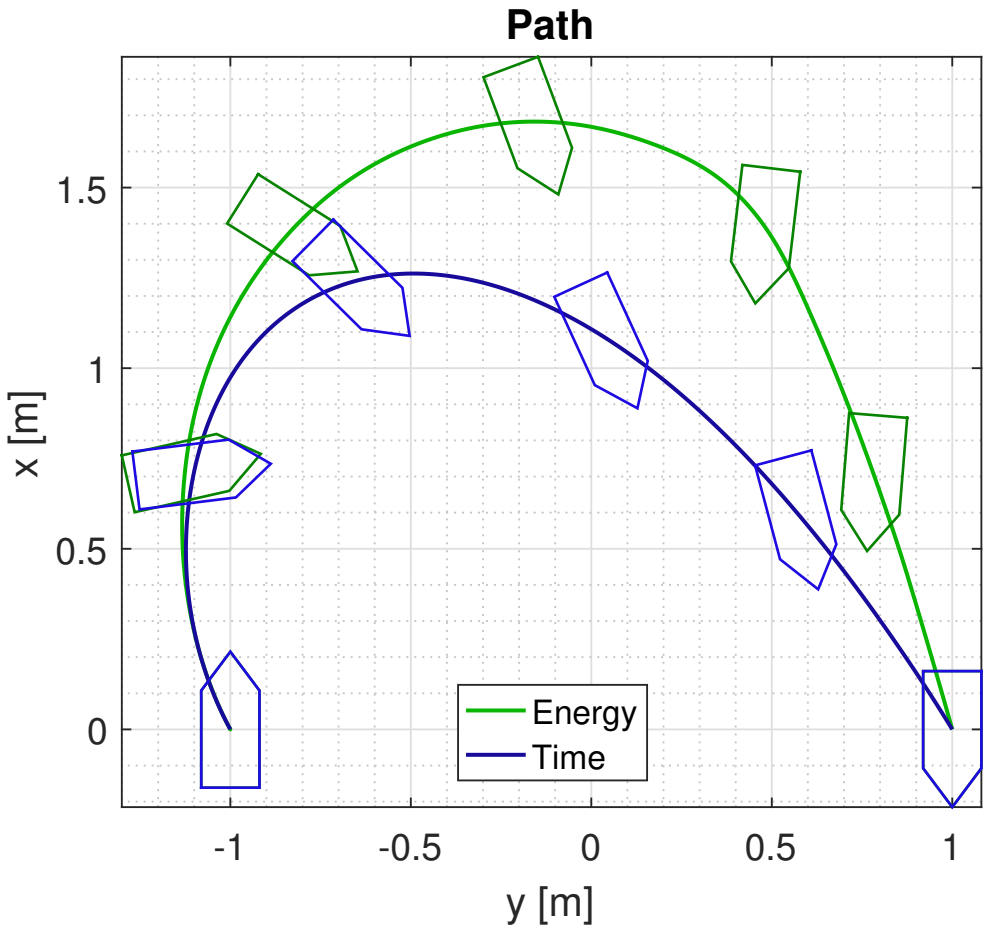


Figure 6.11: Short U-turn, no current: Path.

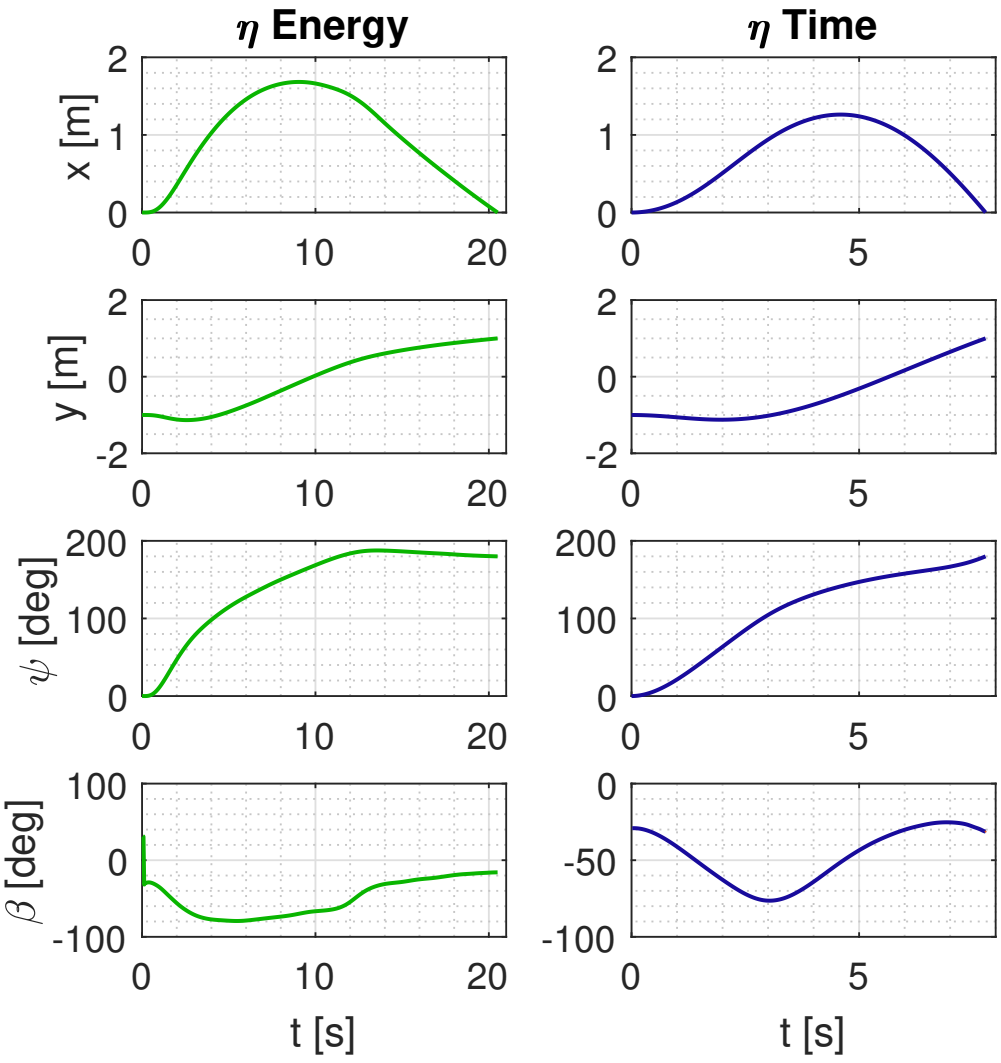


Figure 6.12: Short U-turn, no current: Pose.

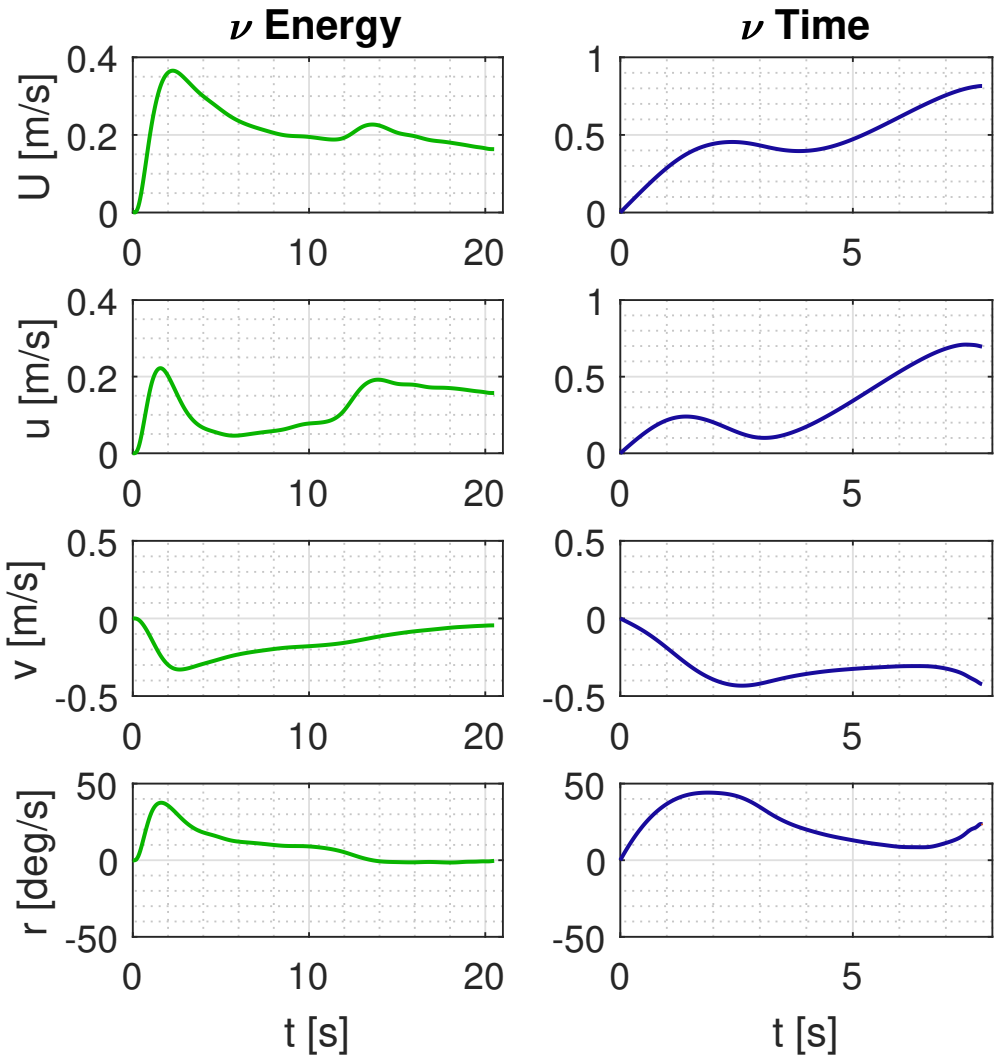


Figure 6.13: Short U-turn, no current: Absolute velocities.

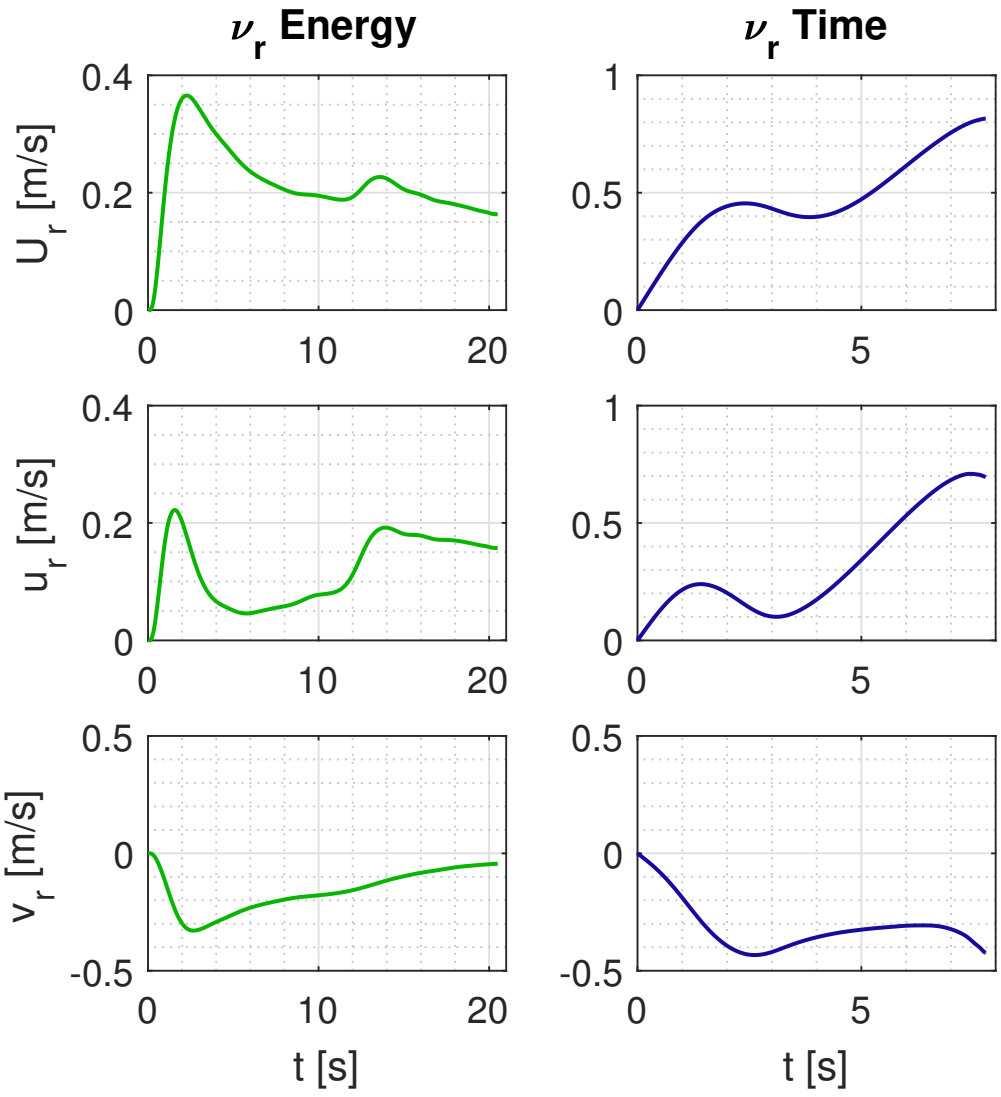


Figure 6.14: Short U-turn, no current: Relative velocities.

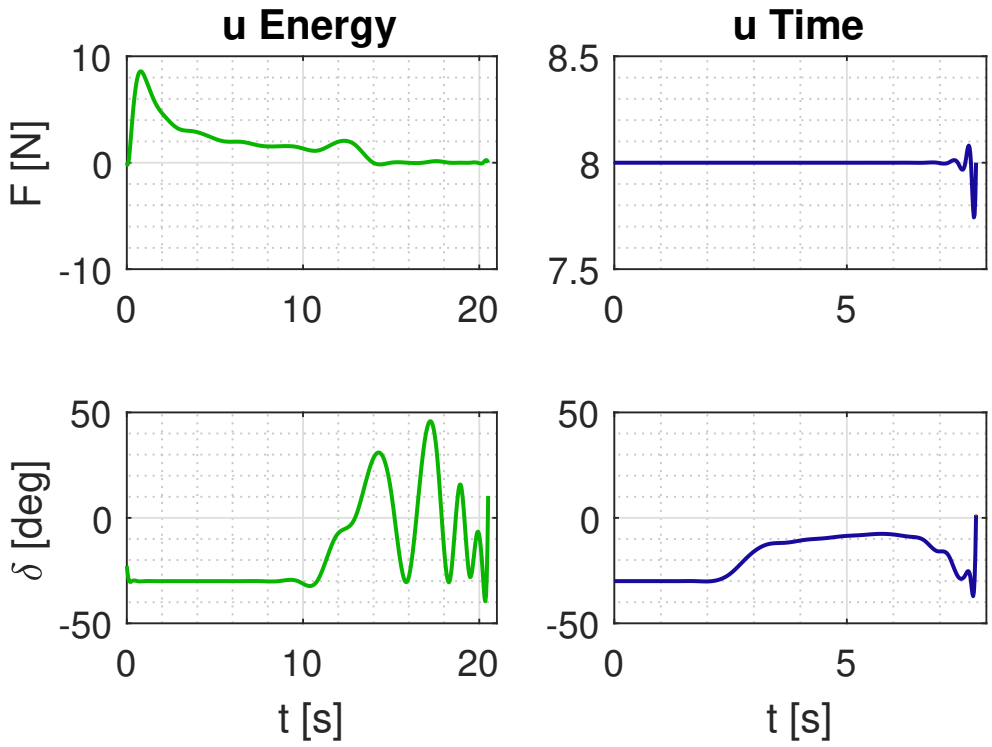


Figure 6.15: Short U-turn, no current: Control inputs.

Northern current

The results are found in Table 6.5 and in figures 6.16 through 6.20.

Table 6.5: Short U-turn, northern current: Results.

Current velocity		
V_x	0.2	m/s
V_y	0	m/s
Minimum-energy run		
Cost	9.296	J
t_f	30	s
Run time	61	s
Initial guess	Yes	
Classification	Suboptimal	
Minimum-time run		
t_f	9.607	s
Run time	20	s
Initial guess	No	
Classification	Good	

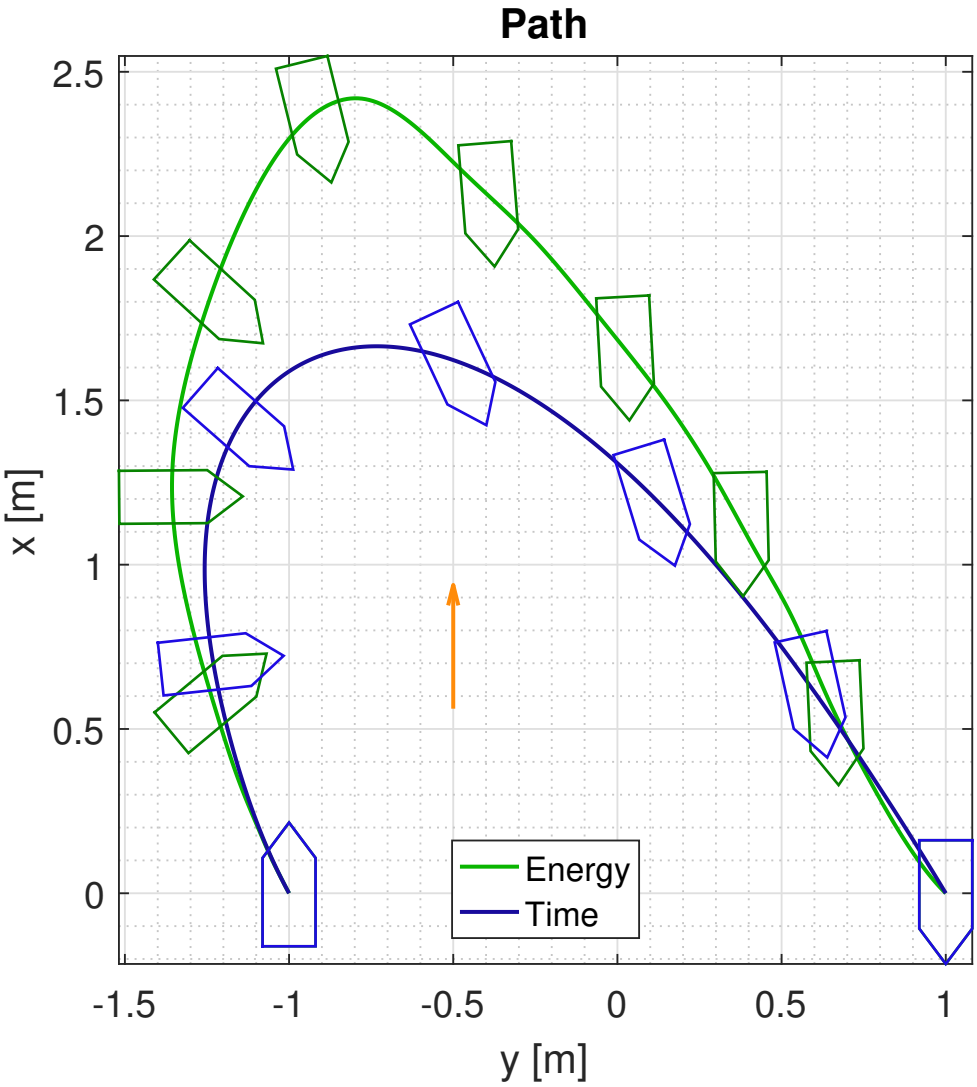


Figure 6.16: Short U-turn, northern current: Path.

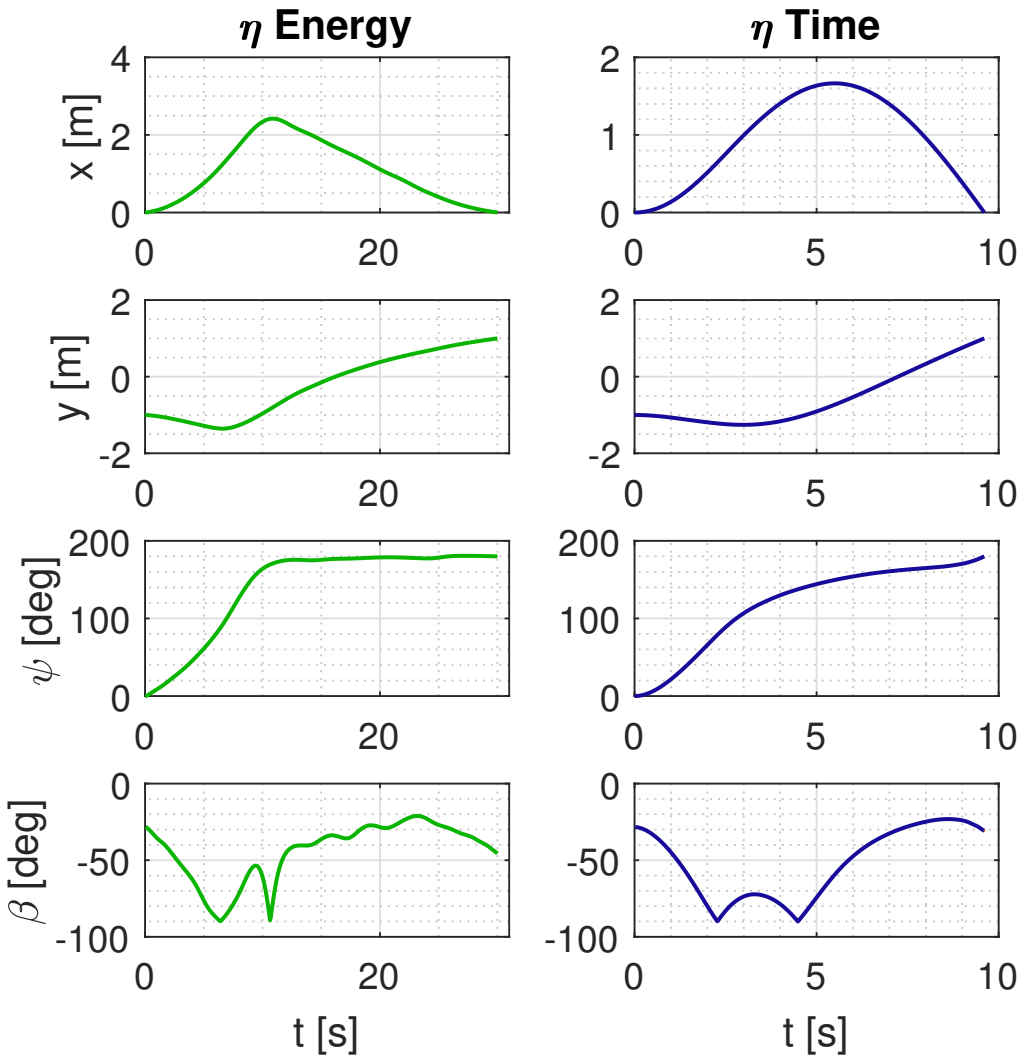


Figure 6.17: Short U-turn, northern current: Pose.

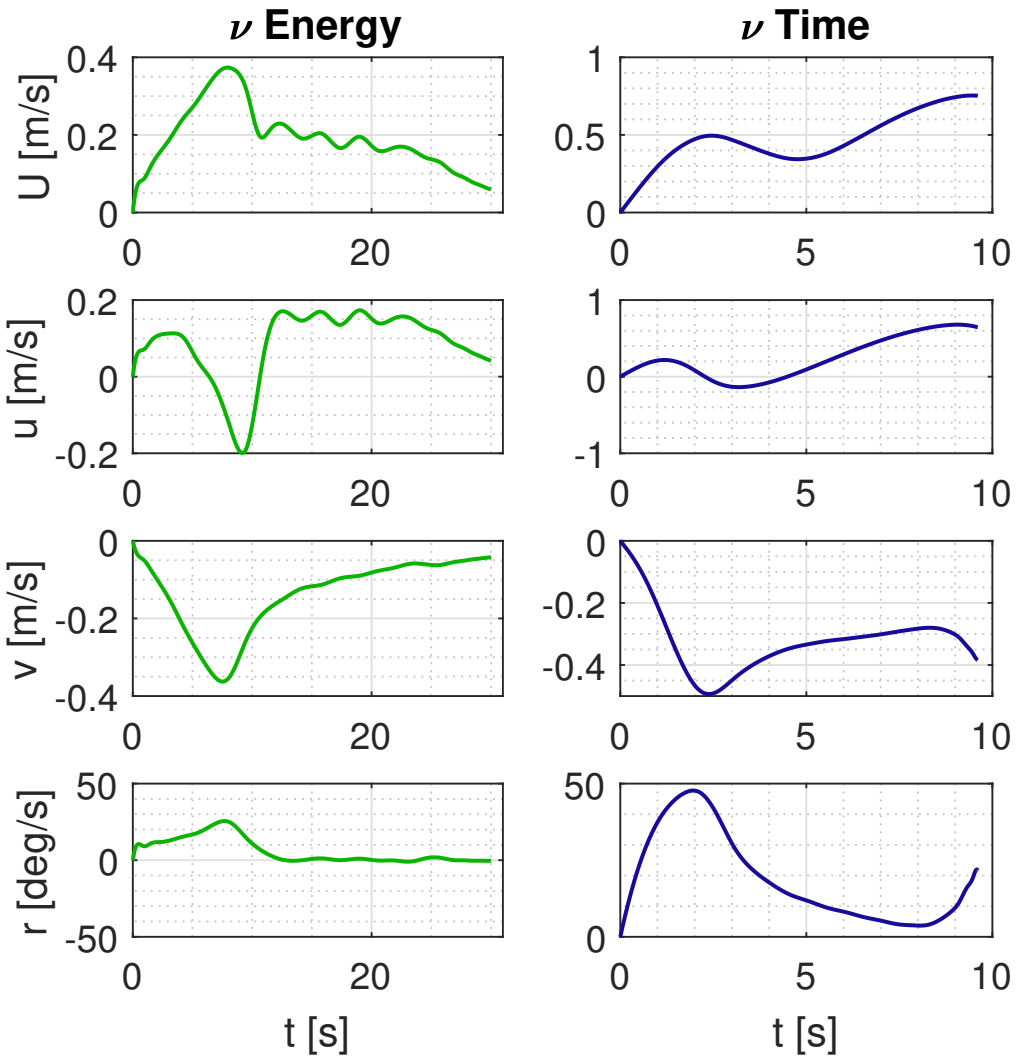


Figure 6.18: Short U-turn, northern current: Absolute velocities.

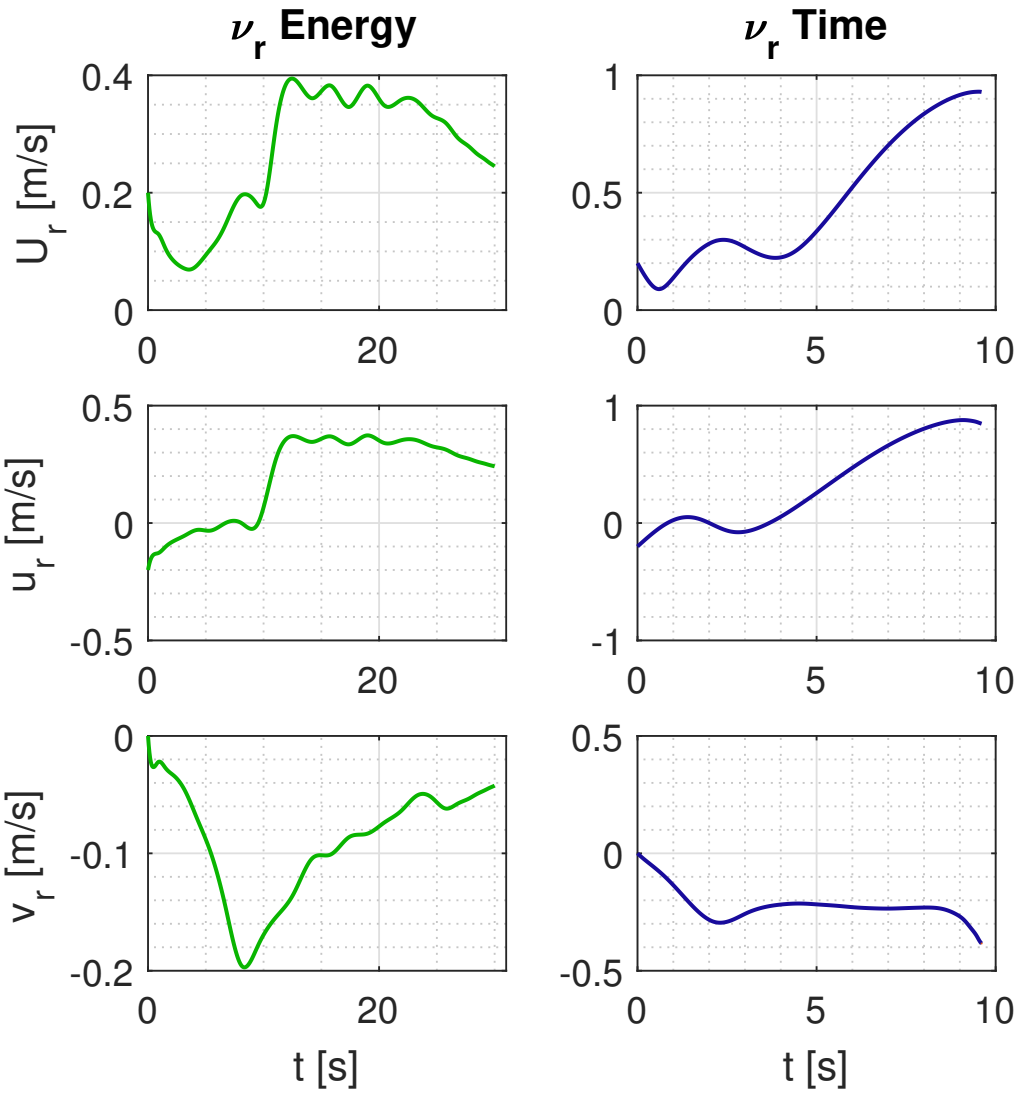


Figure 6.19: Short U-turn, northern current: Relative velocities.

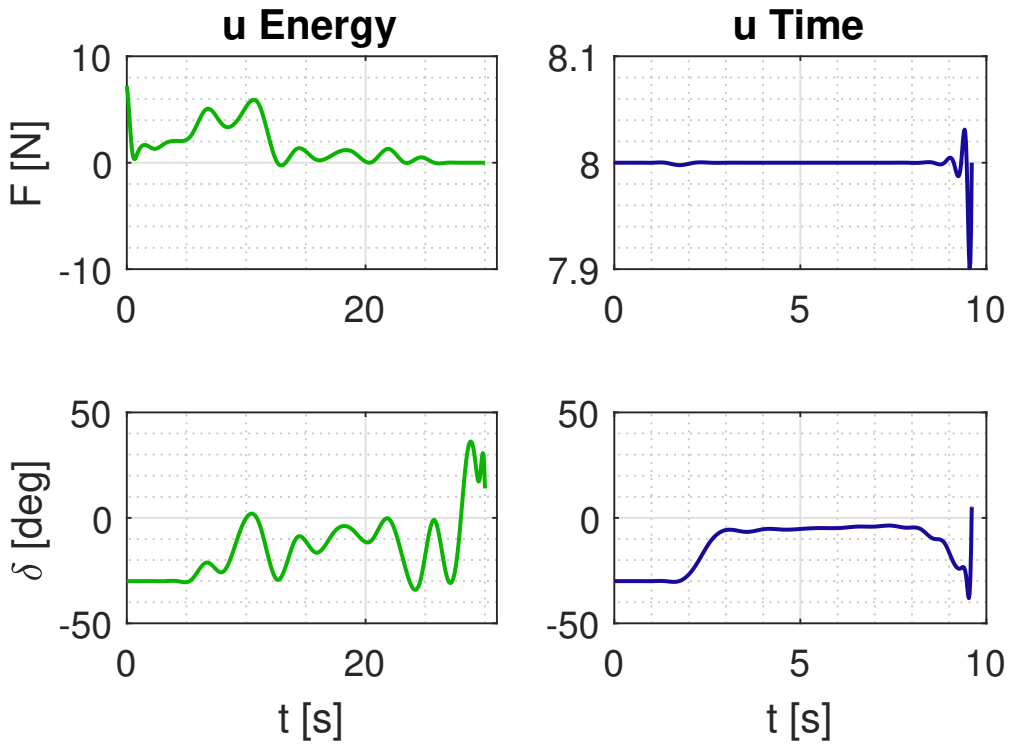


Figure 6.20: Short U-turn, northern current: Control inputs.

6.3.3 Long straight

This scenario is a 20 m movement from west to east. No restrictions are put on final heading, as opposed to the previous scenarios.

No current

The results are found in Table 6.6 and in figures 6.21 through 6.25.

Table 6.6: Long straight, no current: Results.

Current velocity		
V_x	0	m/s
V_y	0	m/s
Minimum-energy run		
Cost	64.56	J
t_f	30	s
Run time	16	s
Initial guess	Yes	
Classification	Extremal	
Minimum-time run		
t_f	21.98	s
Run time	16	s
Initial guess	Yes	
Classification	Extremal	

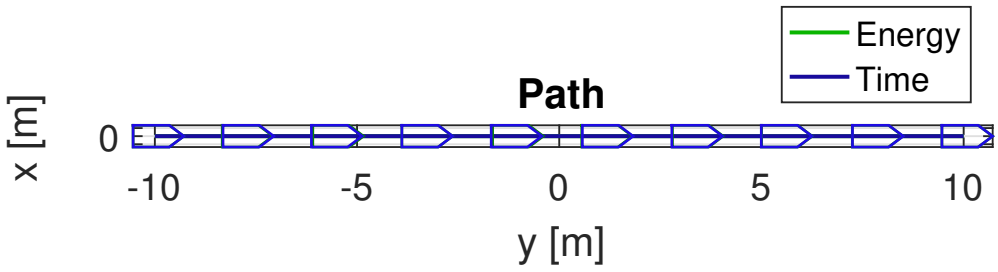


Figure 6.21: Long straight, no current: Path.

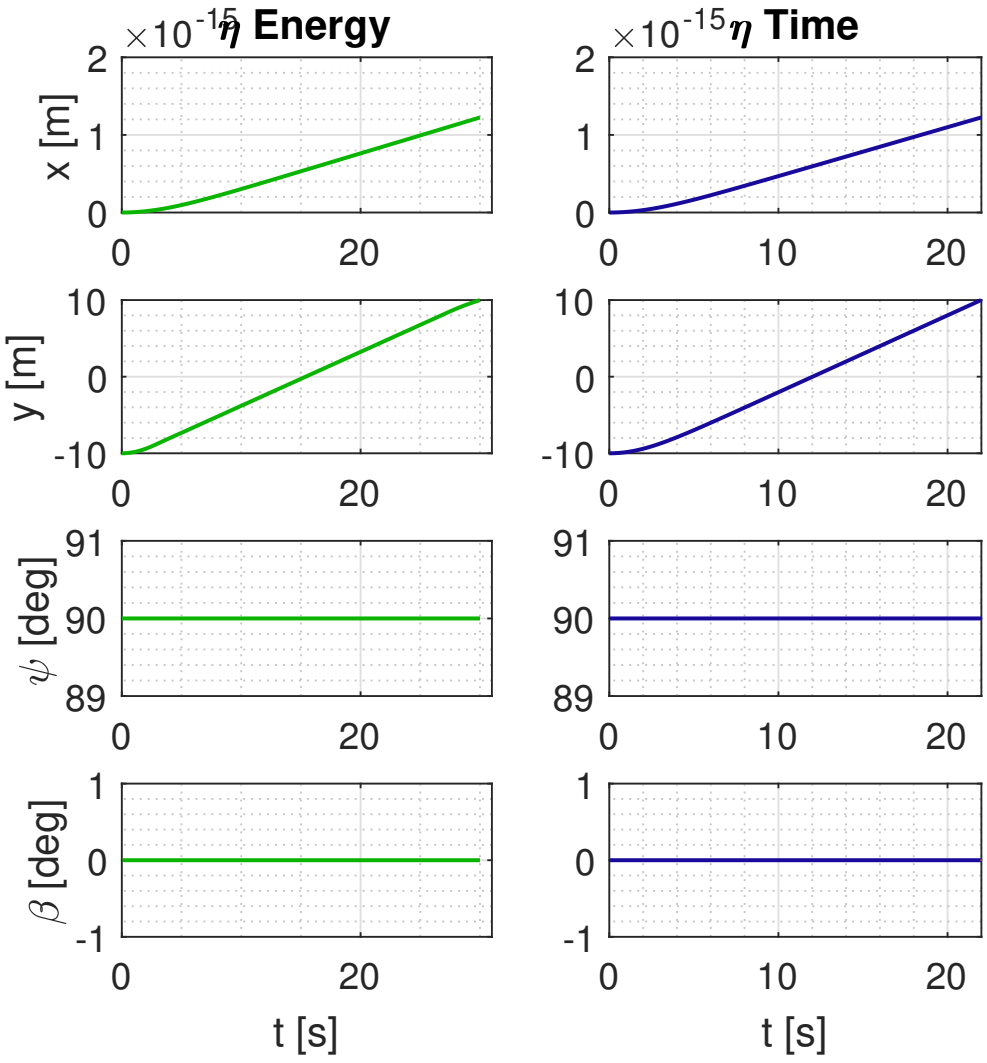


Figure 6.22: Long straight, no current: Pose.

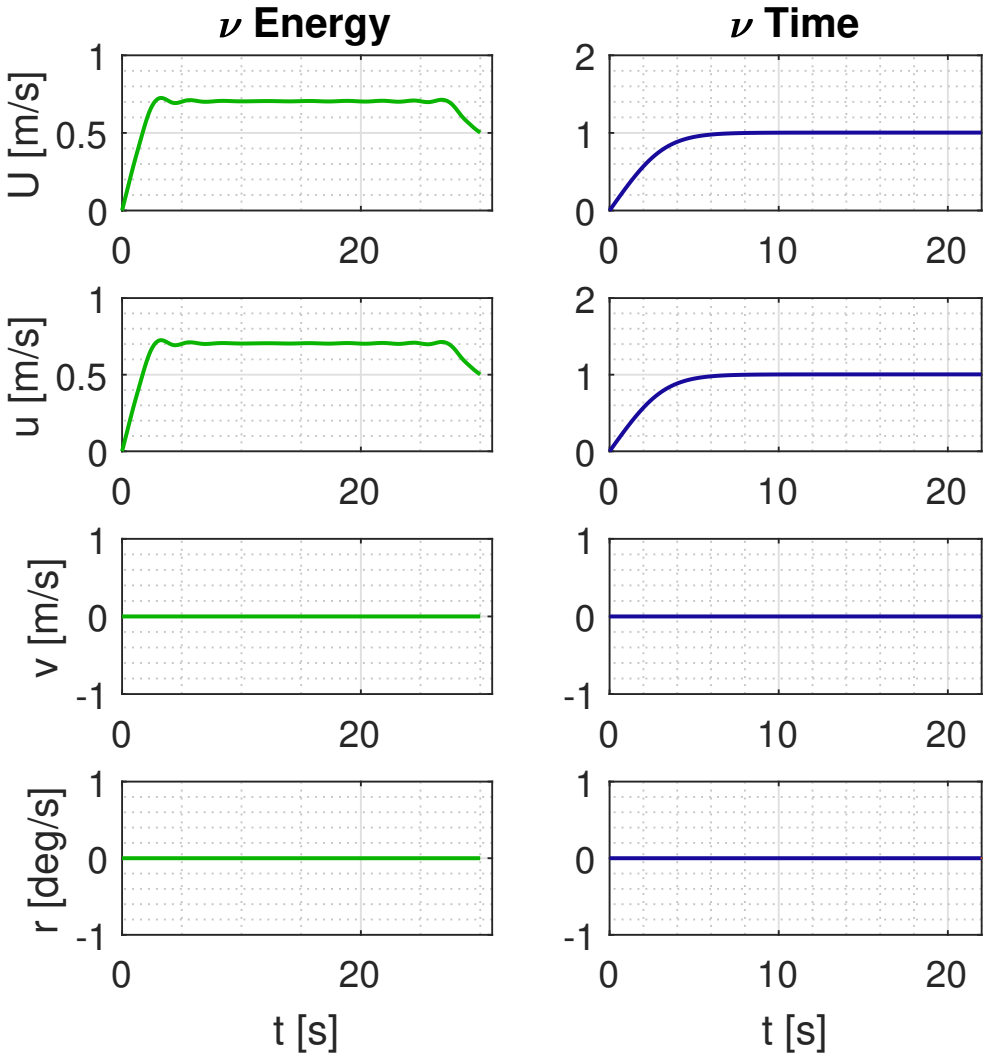


Figure 6.23: Long straight, no current: Absolute velocities.

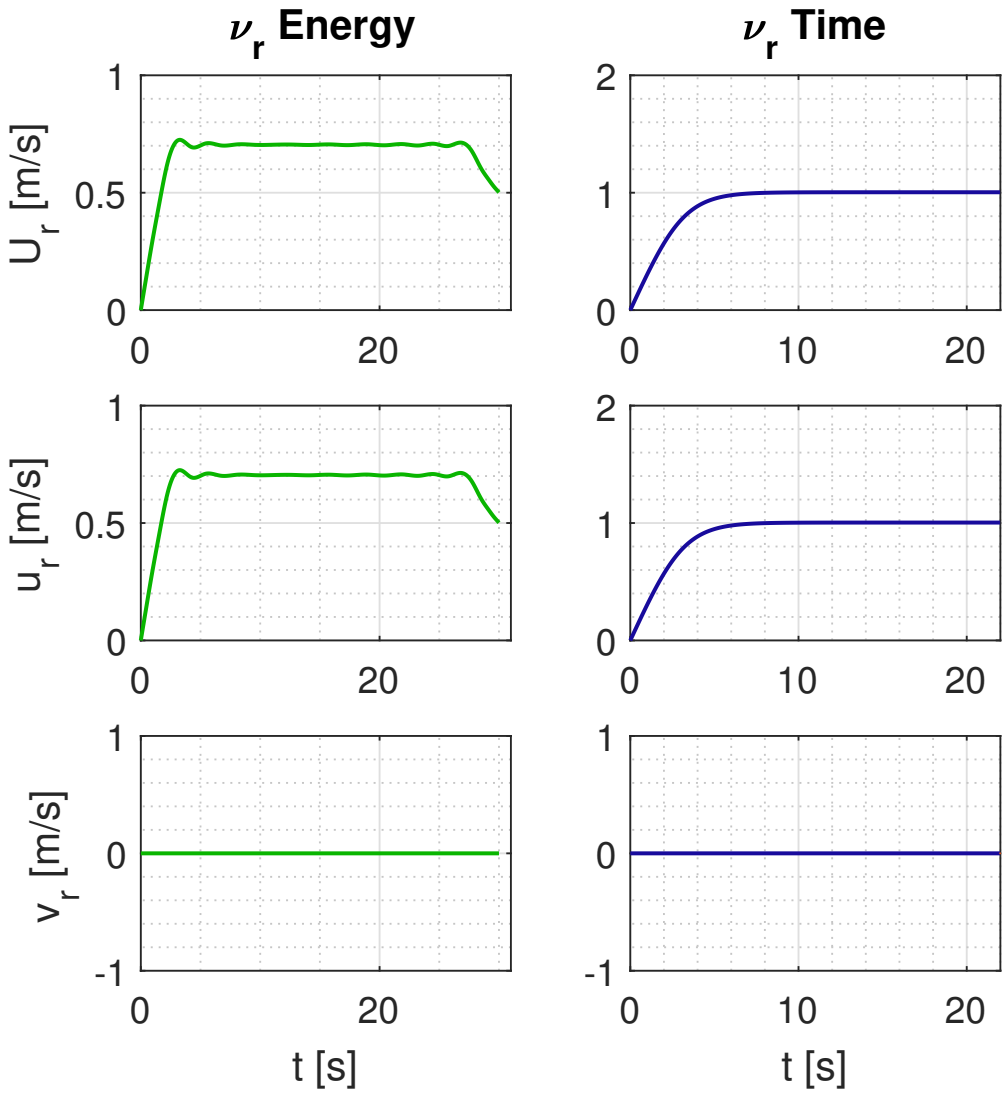


Figure 6.24: Long straight, no current: Relative velocities.

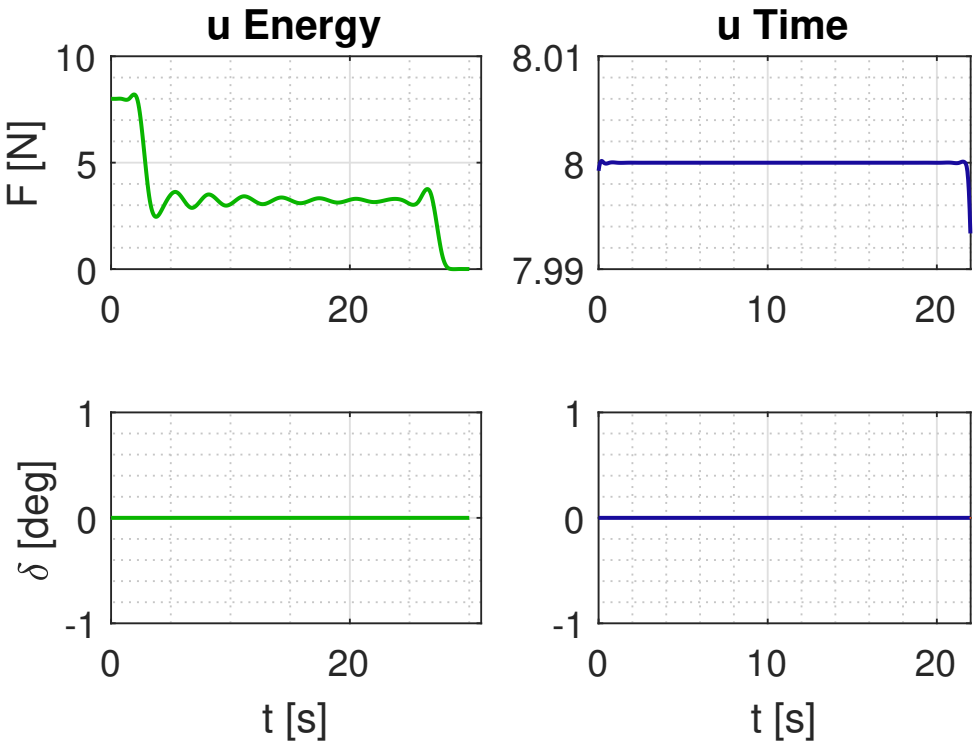


Figure 6.25: Long straight, no current: Control inputs.

Southern current

The results are found in Table 6.7 and in figures 6.26 through 6.30.

Table 6.7: Long straight, southern current: Results.

Current velocity		
V_x	-0.5	m/s
V_y	0	m/s
Minimum-energy run		
Cost	136.4	J
t_f	30	s
Run time	20	s
Initial guess	Yes	
Classification	Suboptimal	
Minimum-time run		
t_f	25.30	s
Run time	26	s
Initial guess	Yes	
Classification	Extremal	

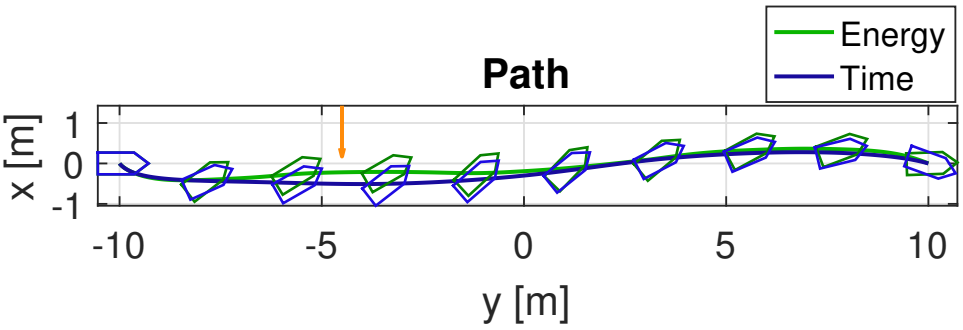


Figure 6.26: Long straight, southern current: Path.

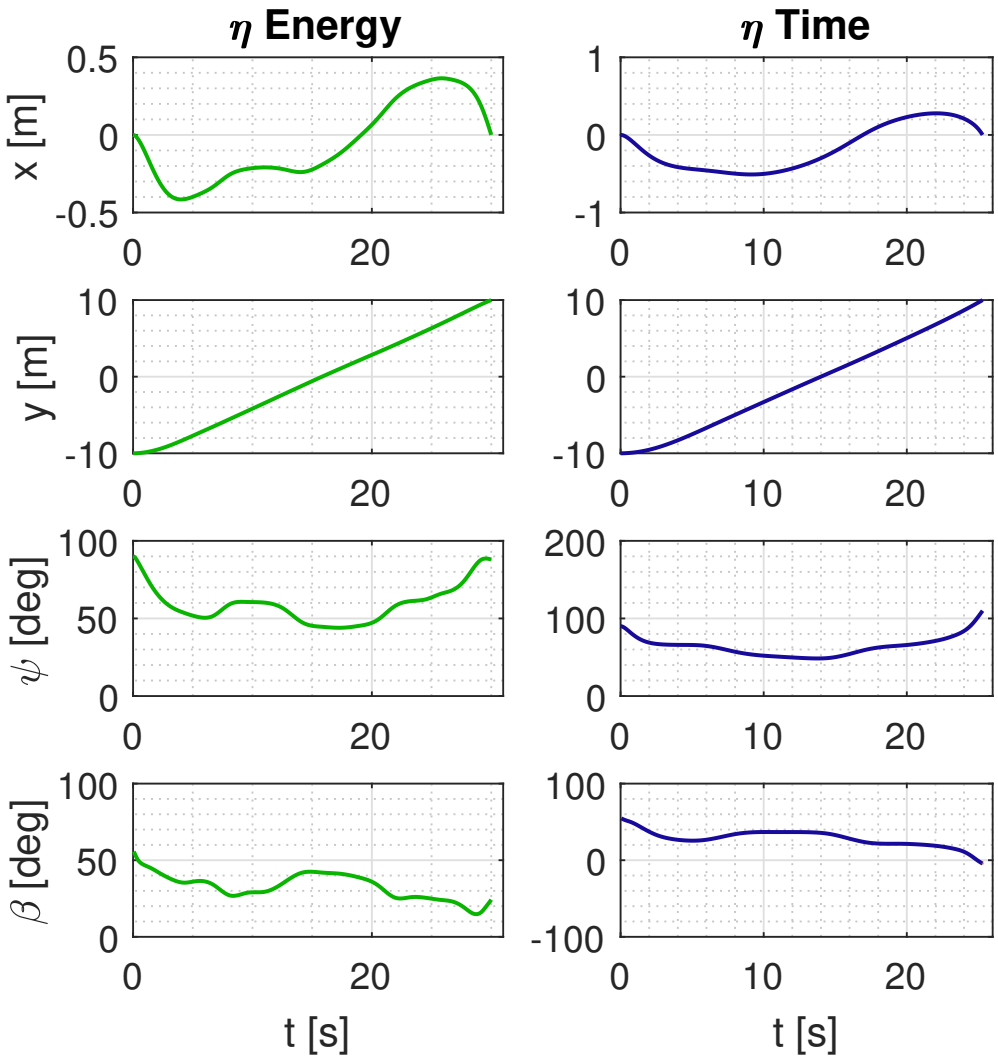


Figure 6.27: Long straight, southern current: Pose.

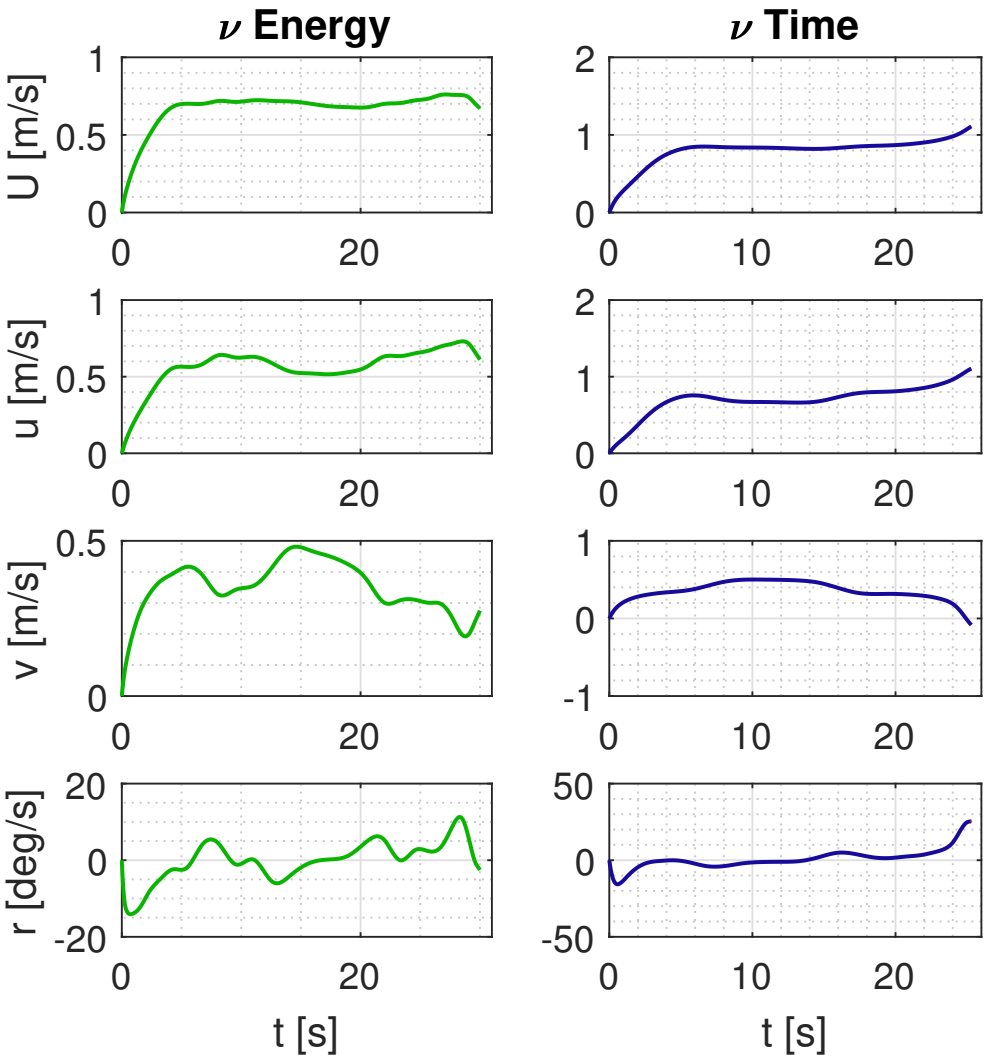


Figure 6.28: Long straight, southern current: Absolute velocities.

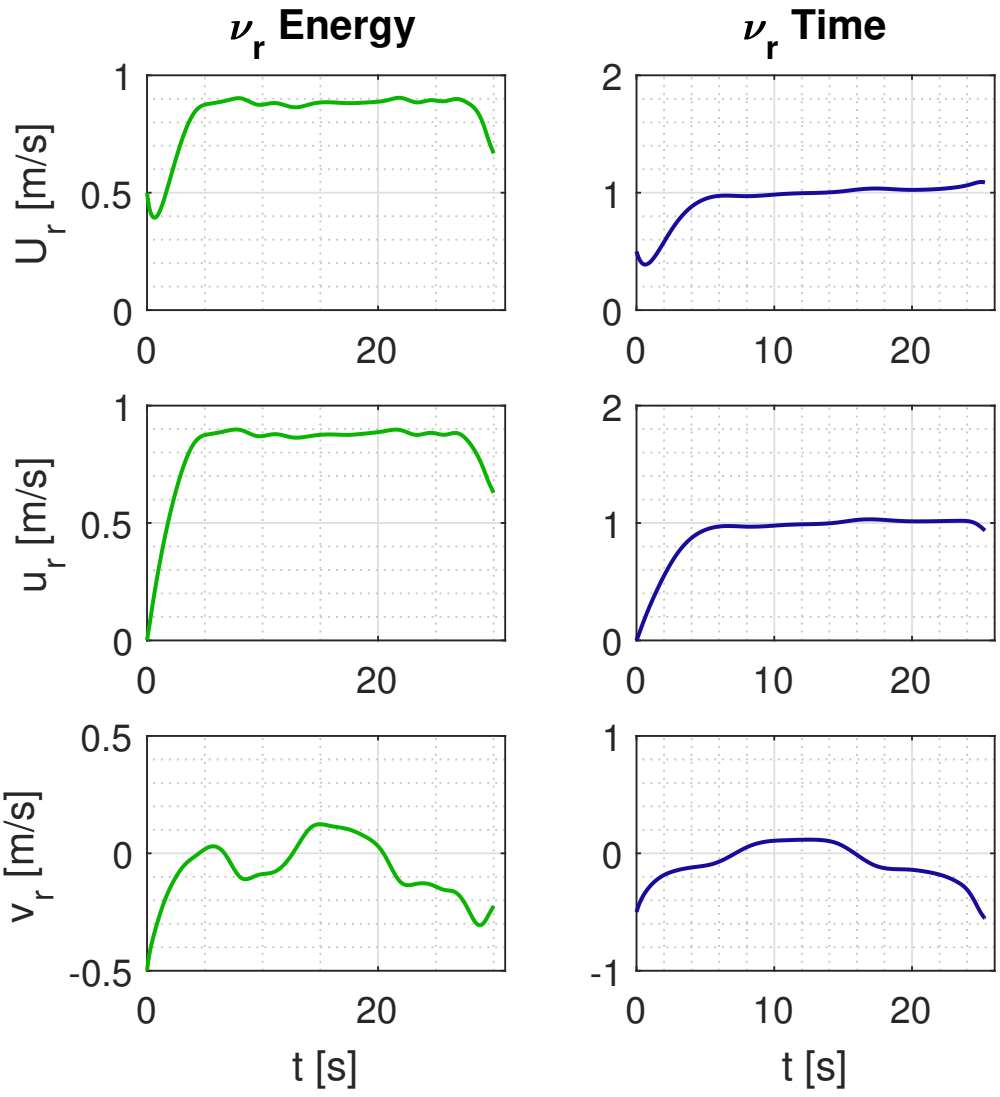


Figure 6.29: Long straight, southern current: Relative velocities.

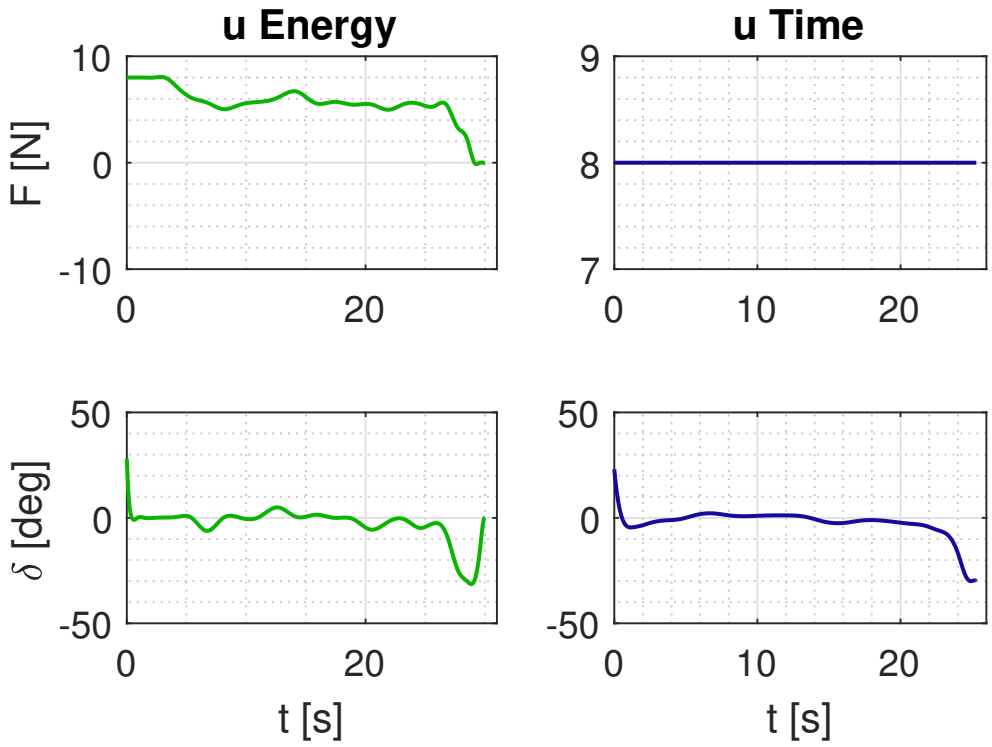


Figure 6.30: Long straight, southern current: Control inputs.

6.3.4 Discussion

From the results in this section it is pointed out that all scenarios produce feasible paths, verified by the method described in Section 6.2.6. All of the experiments with the minimum-time cost function are classified as either *extremal* or *good*, while some of the minimum-energy experiments are *suboptimal*. The reason for the minimum-energy experiments have suboptimal solutions may be due to the absolute value operators in the cost function.

Other than the straight-line scenario, the routes for minimum-time and minimum-energy experiments differ significantly. Minimum-time routes are characterized by having sharper turns and shorter paths, while the minimum-energy routes use longer time and less control action to complete.

Current affects the cost functions differently as well. In the U-turn scenario with northern current (Figure 6.16) the minimum-energy ship spends more time with the heading against the current than the minimum-time ship, to reduce drag.

In the U-turn scenarios, the minimum-energy ship seems to expend less energy with northern current compared to the no-current experiment, which is unintuitive. However, both the results are classified as *suboptimal* by DIDO, and no conclusion may be drawn. In general the minimum-energy experiments have higher costs when ocean currents are present.

6.4 Single-obstacle scenarios

These scenarios are run with only the minimum energy cost function. The initial absolute velocities are $\boldsymbol{\nu}_0 = [0.2\,\mathrm{m/s}, 0\,\mathrm{m/s}, 0\,^\circ/\mathrm{s}]^\top$. The scenarios are run with and without ocean current. The maximum final time $t_{f,\max}$ is set to 120 s. The algorithm is run with 30 nodes. Figure 6.31 shows the map including the obstacle. The initial pose is $\boldsymbol{\eta}_0 = [38\,\mathrm{m}, 6\,\mathrm{m}, 180^\circ]^\top$, and the final conditions are $x_f = y_f = 44\,\mathrm{m}$. There are no restrictions on final heading. The obstacle is an ellipse represented by

$$c_1(\boldsymbol{x}) = \left(\frac{x - 38\,\mathrm{m}}{25\,\mathrm{m}}\right)^2 + \left(\frac{y - 25\,\mathrm{m}}{6\,\mathrm{m}}\right)^2 - 1 \geq 0 \tag{6.8}$$

The results are collected in Table 6.8.

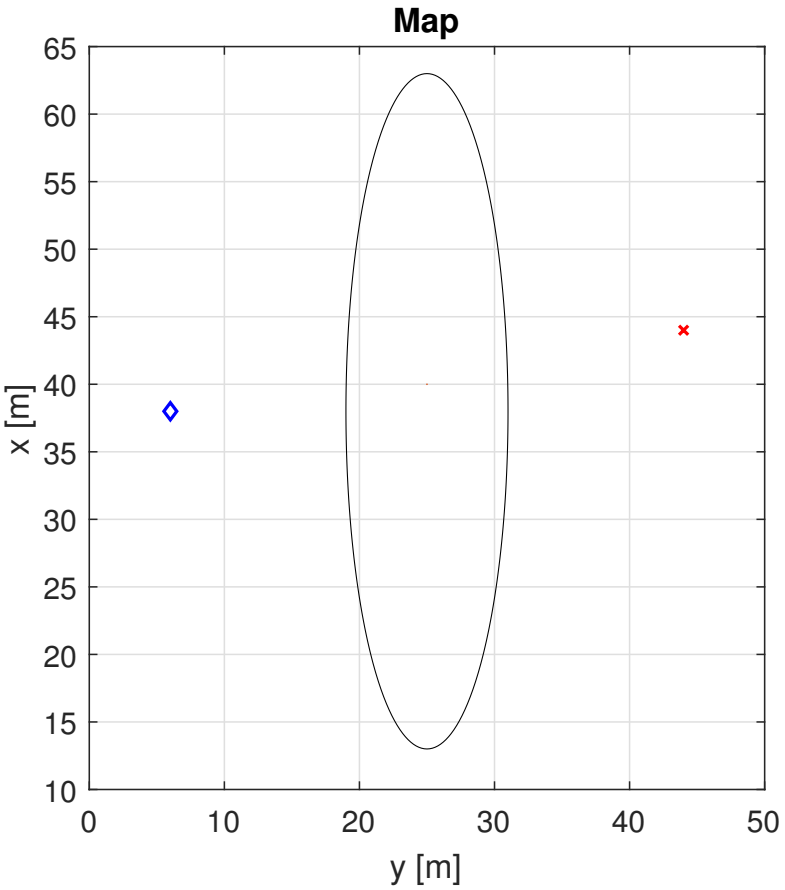


Figure 6.31: Map with obstacle. The blue diamond represents the starting position, and the red cross is the goal position.

Table 6.8: Single obstacle: Results

Current	(V_x, V_y) (m/s, m/s)	Cost J	t_f s	Run time s	Initial guess	Classification
None	$(0, 0)$	142.7	120	121	Yes	Good
Eastern	$(0, 0.3)$	112.3	120	31	Yes	Good
Western	$(0, -0.3)$	413.5	120	32	Yes	Suboptimal

6.4.1 No current

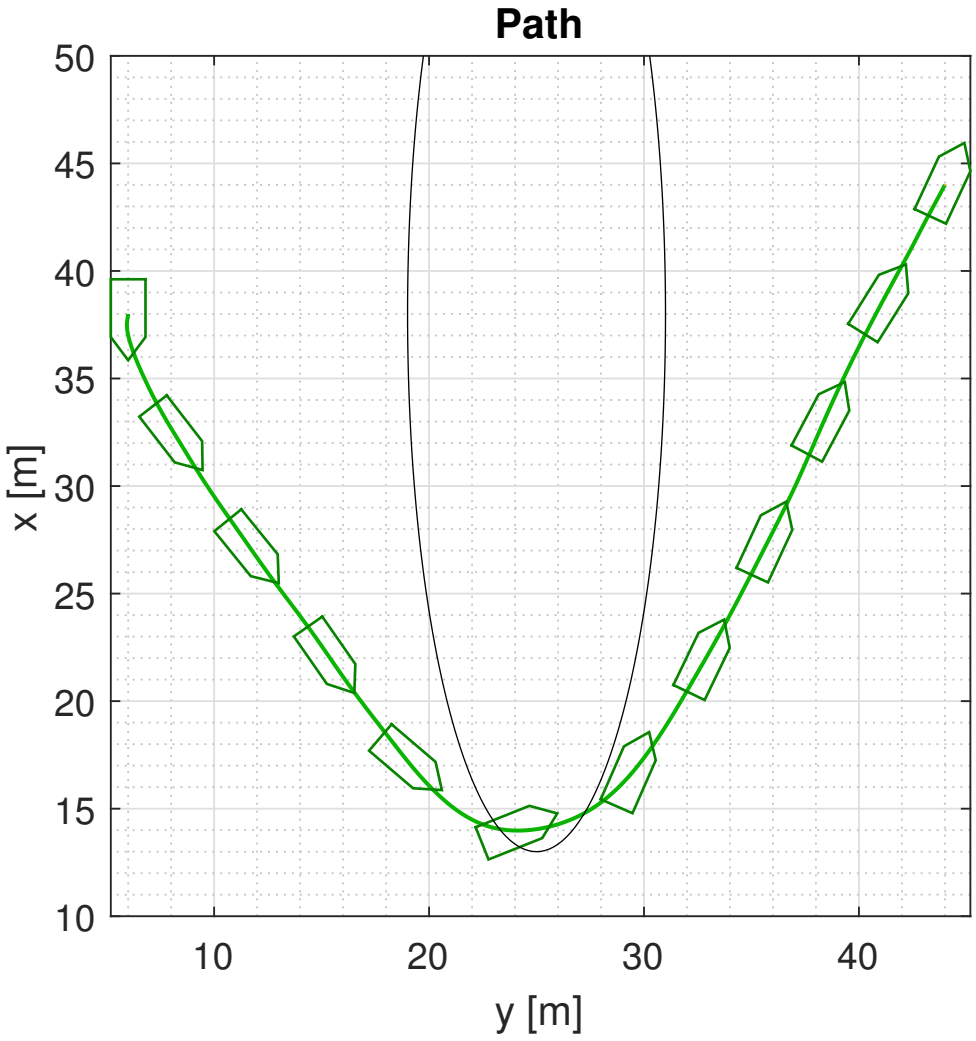


Figure 6.32: Single obstacle, no current: Path.

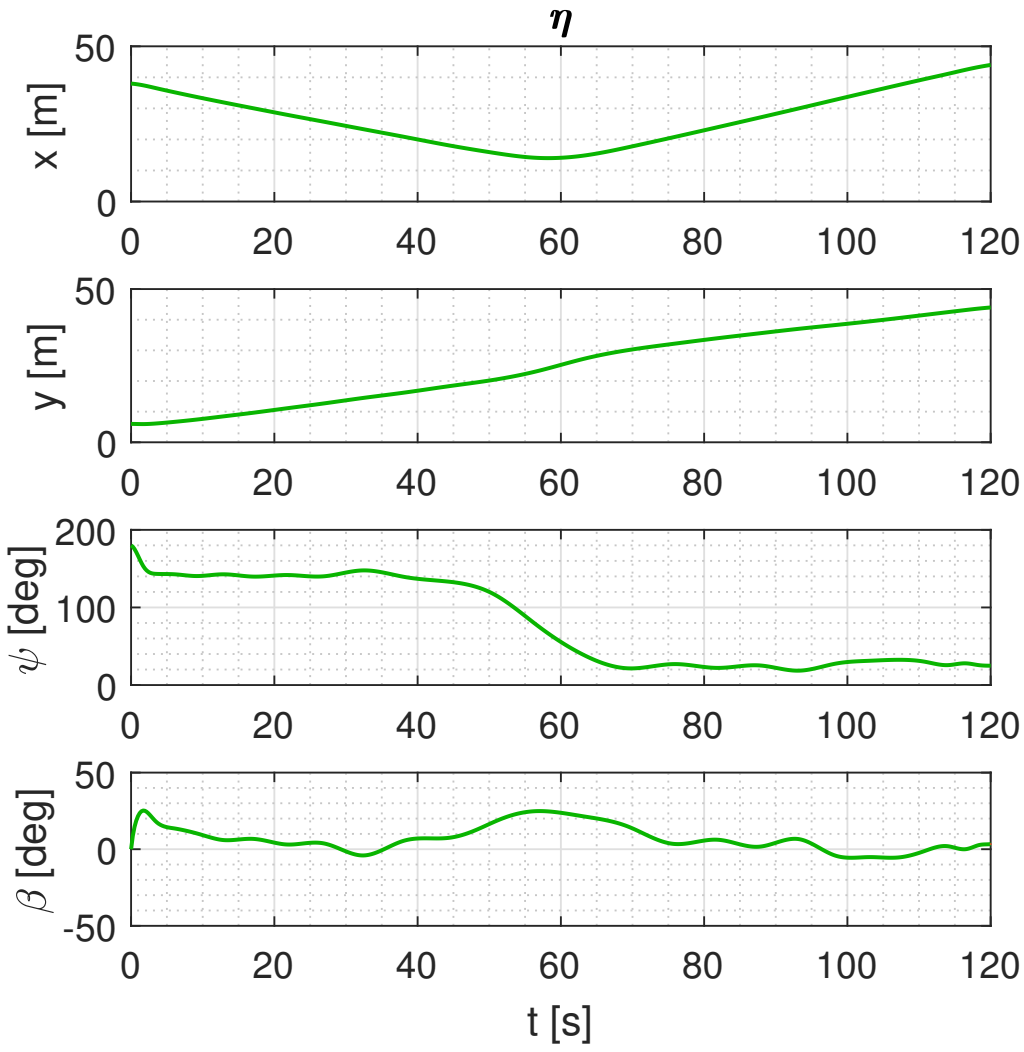


Figure 6.33: Single obstacle, no current: Pose.

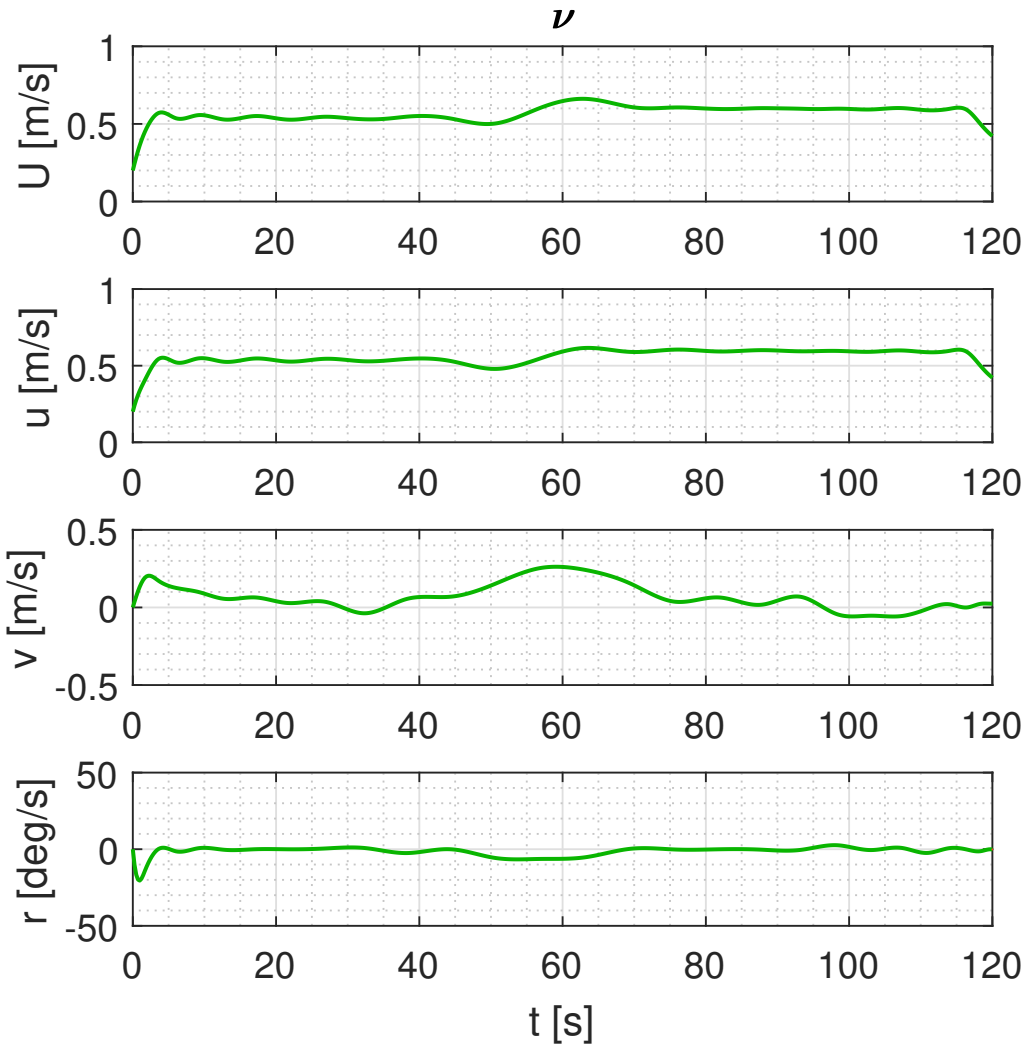


Figure 6.34: Single obstacle, no current: Absolute velocities.

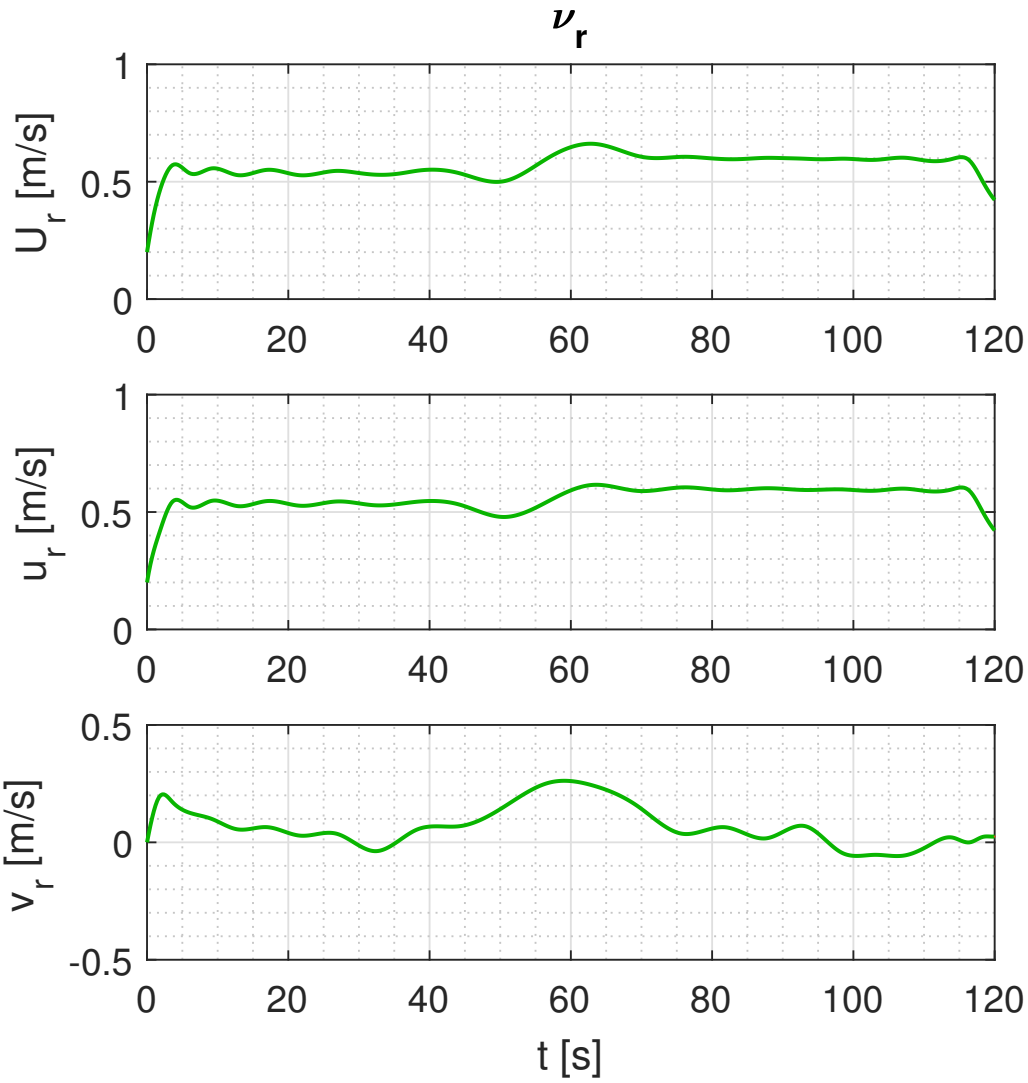


Figure 6.35: Single obstacle, no current: Relative velocities.

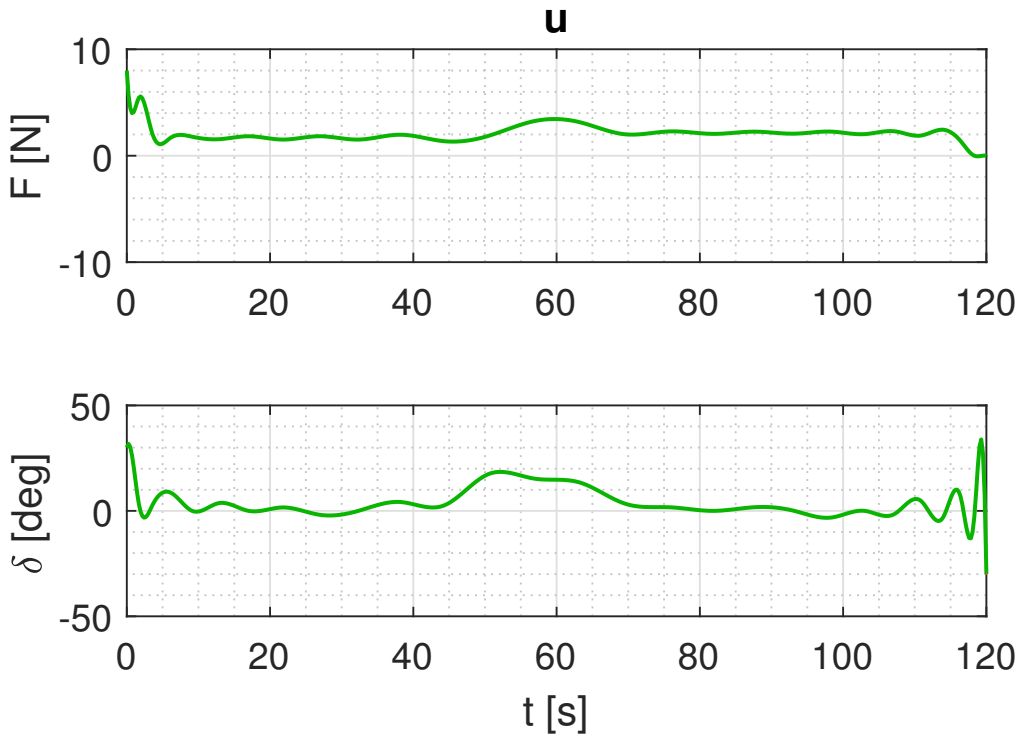


Figure 6.36: Single obstacle, no current: Control inputs.

6.4.2 Eastern current

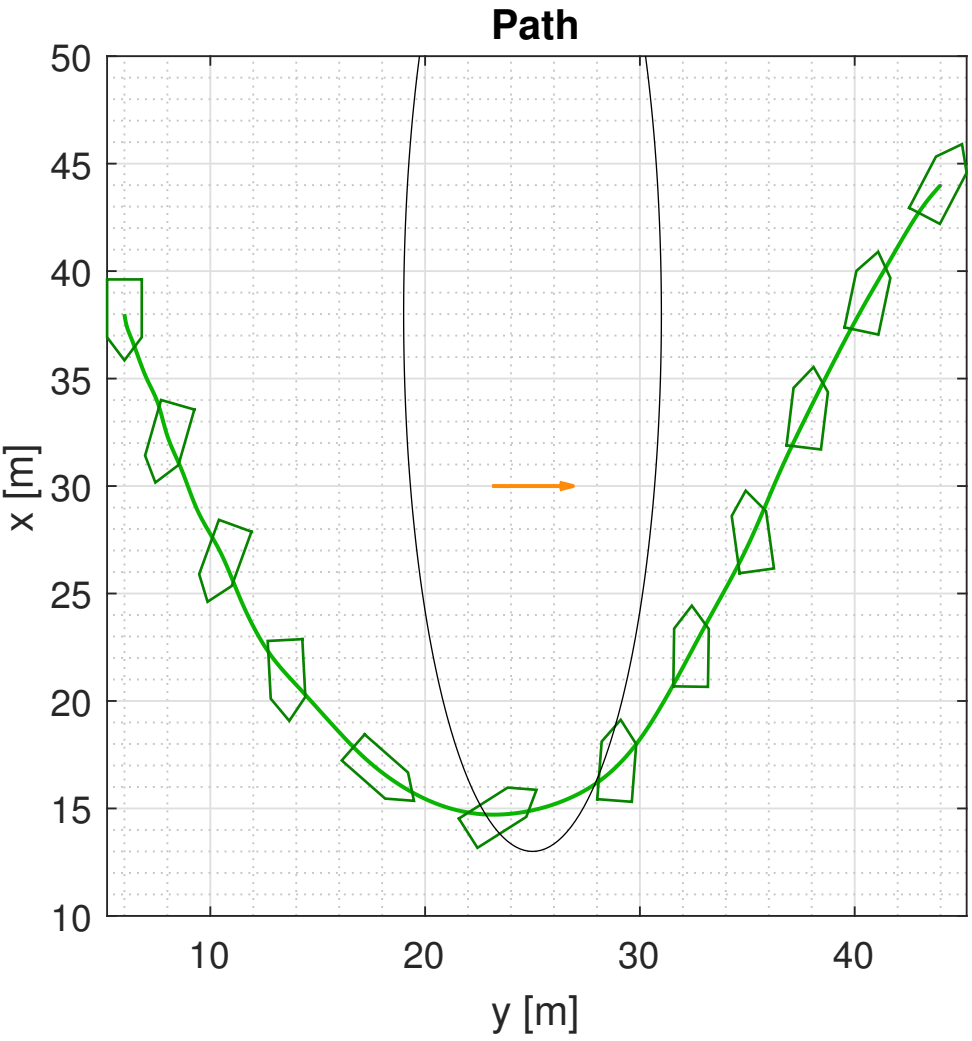


Figure 6.37: Single obstacle, eastern current: Path.

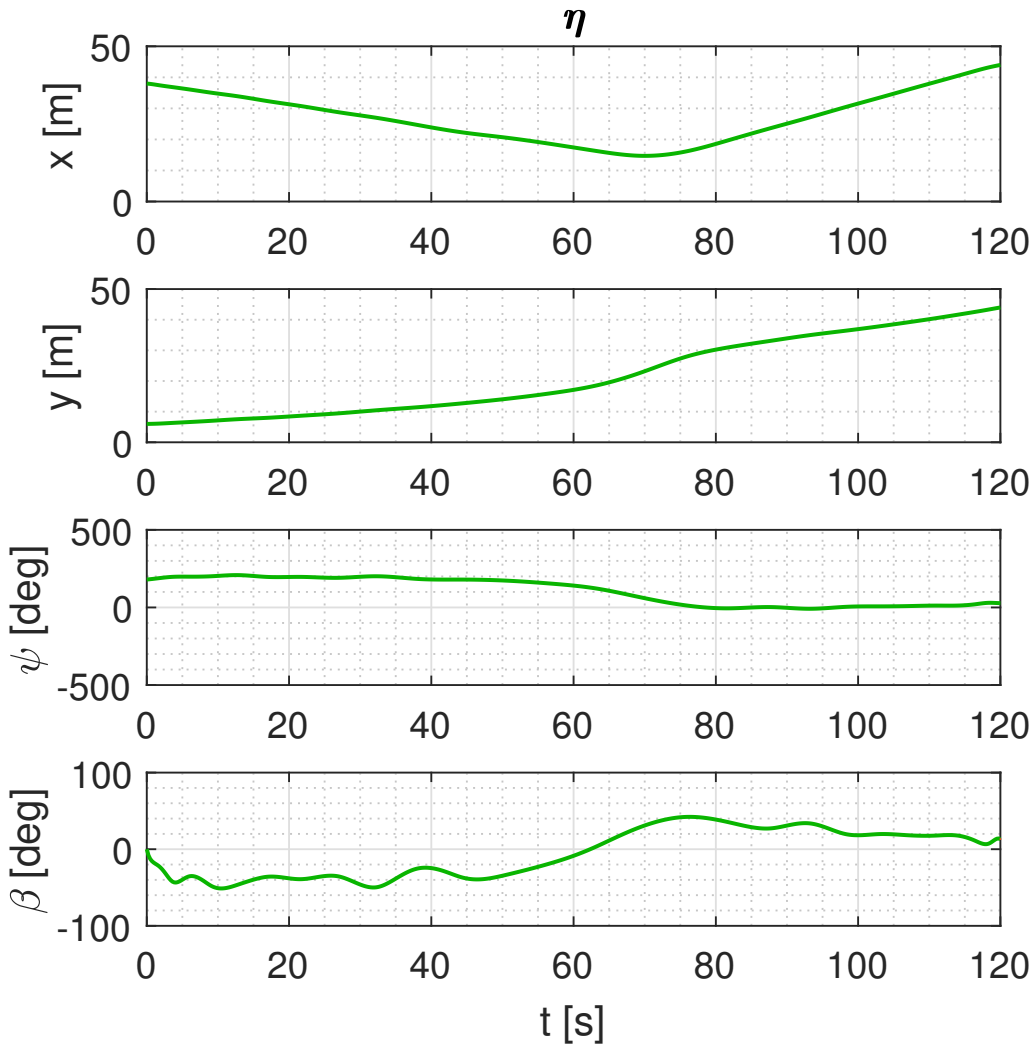


Figure 6.38: Single obstacle, eastern current: Pose.

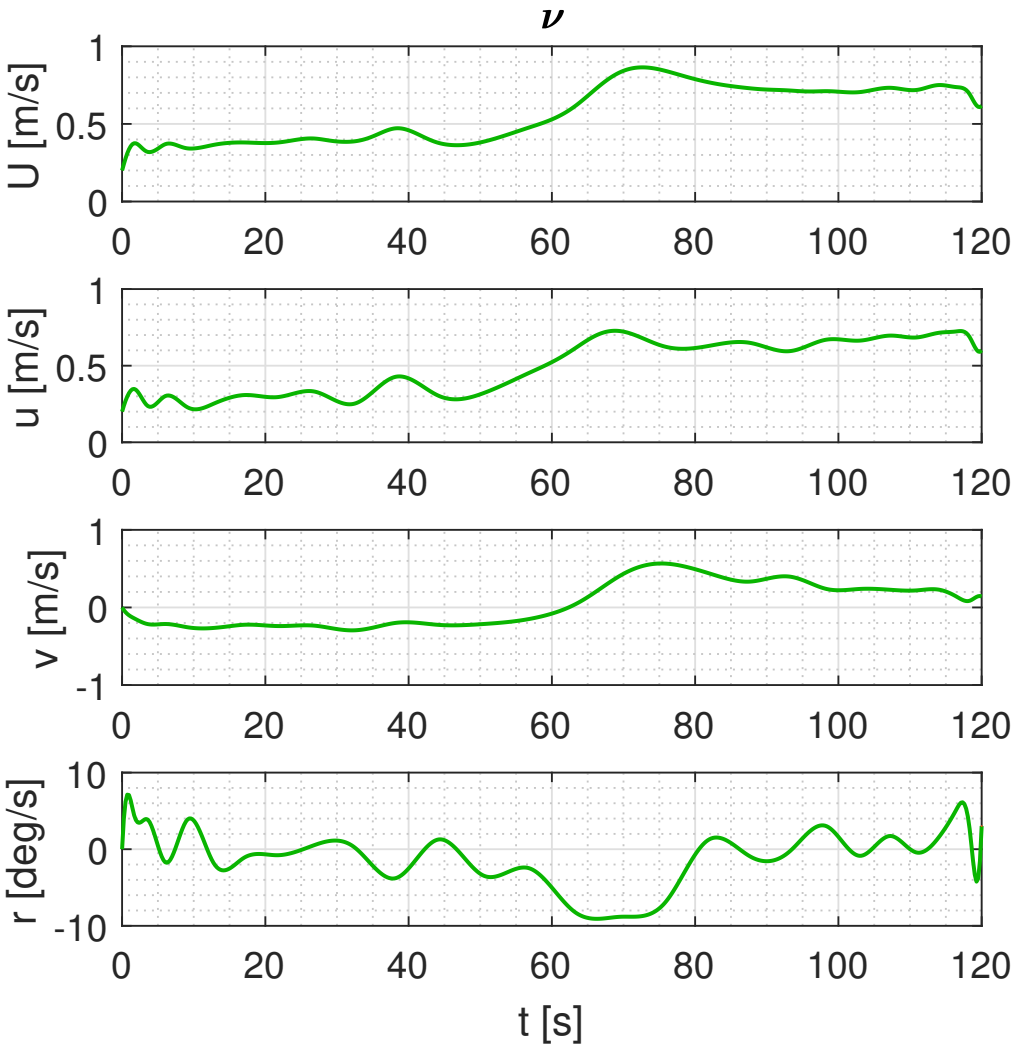


Figure 6.39: Single obstacle, eastern current: Absolute velocities.

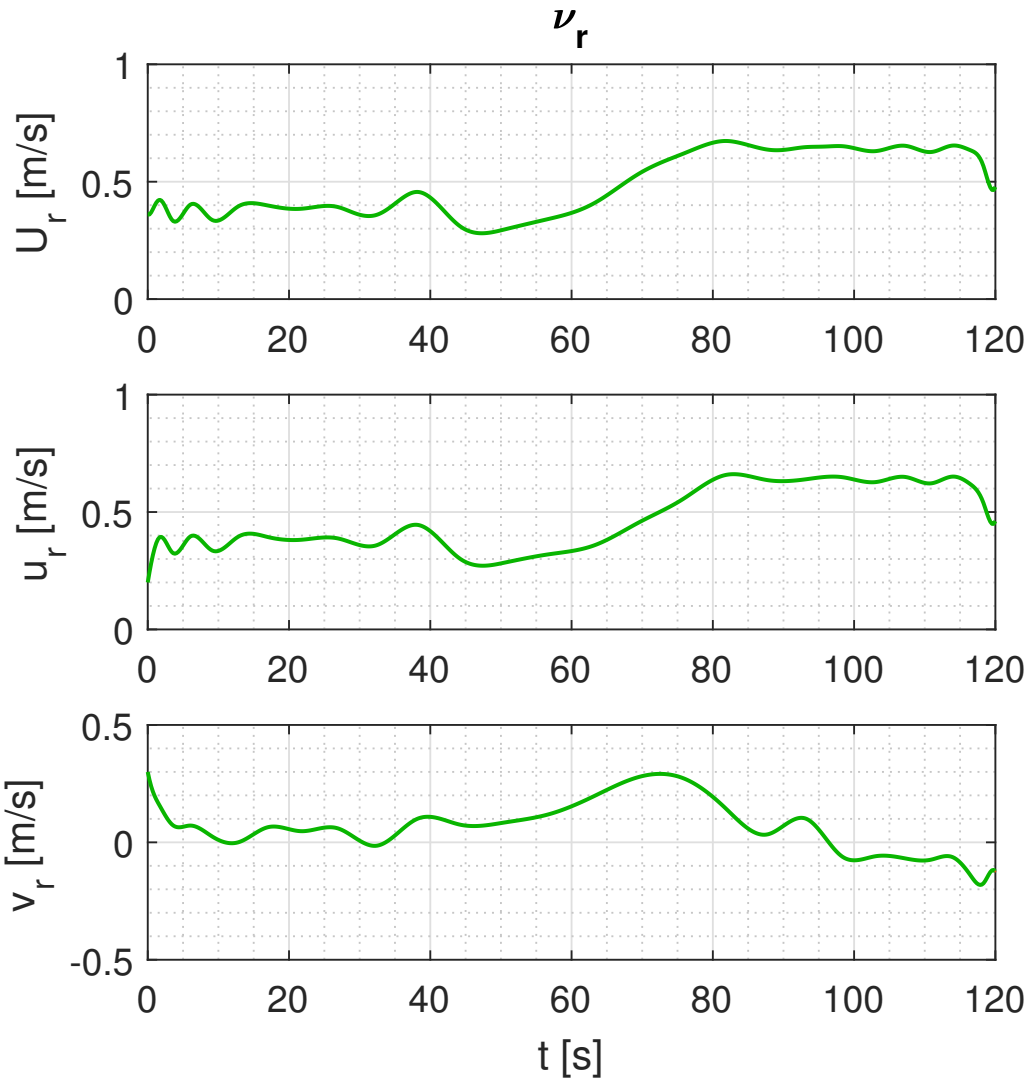


Figure 6.40: Single obstacle, eastern current: Relative velocities.

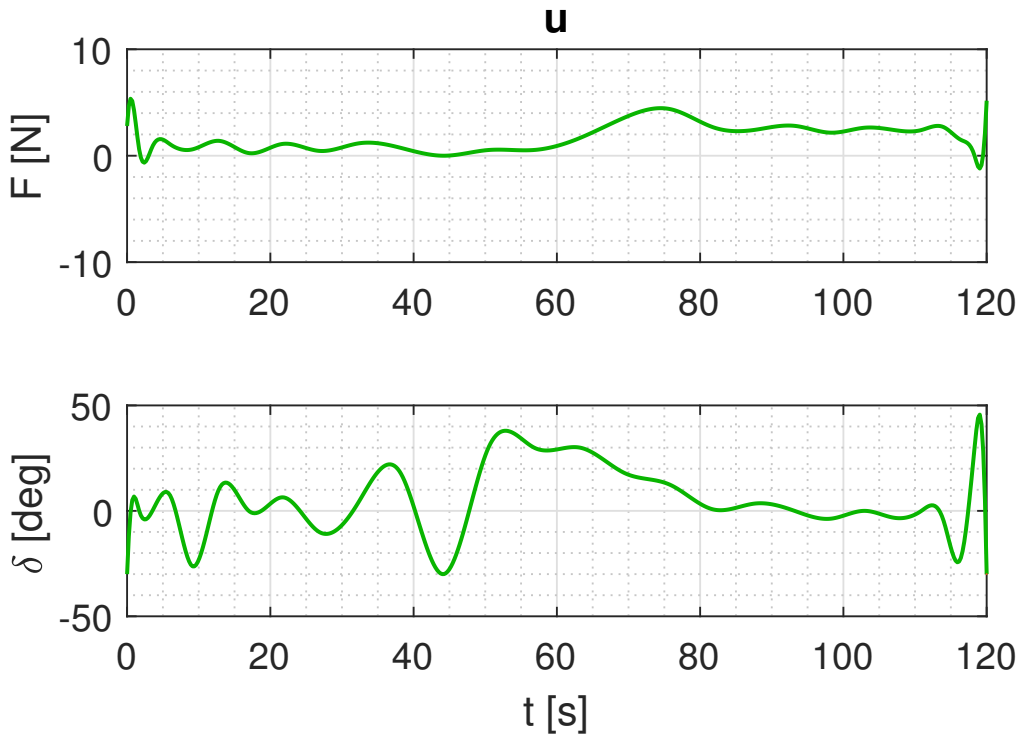


Figure 6.41: Single obstacle, eastern current: Control inputs.

6.4.3 Western current

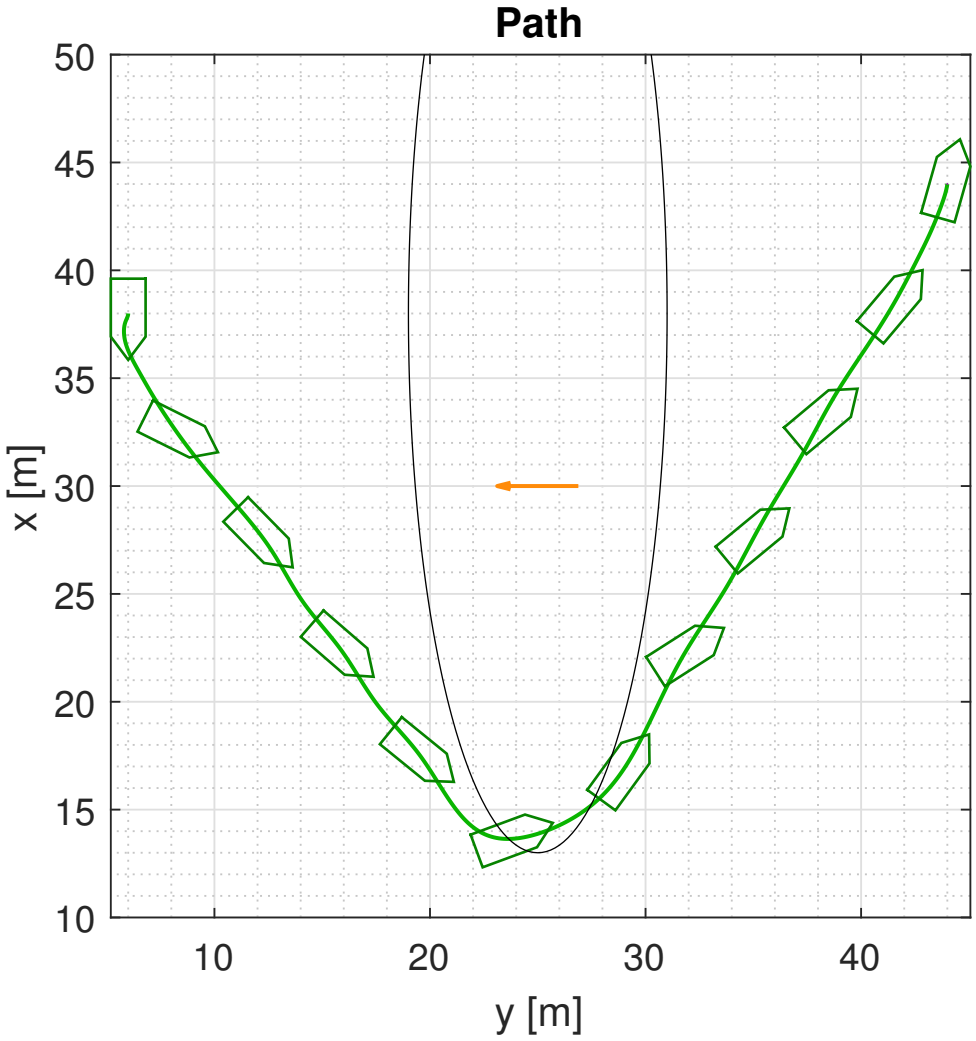


Figure 6.42: Single obstacle, western current: Path.

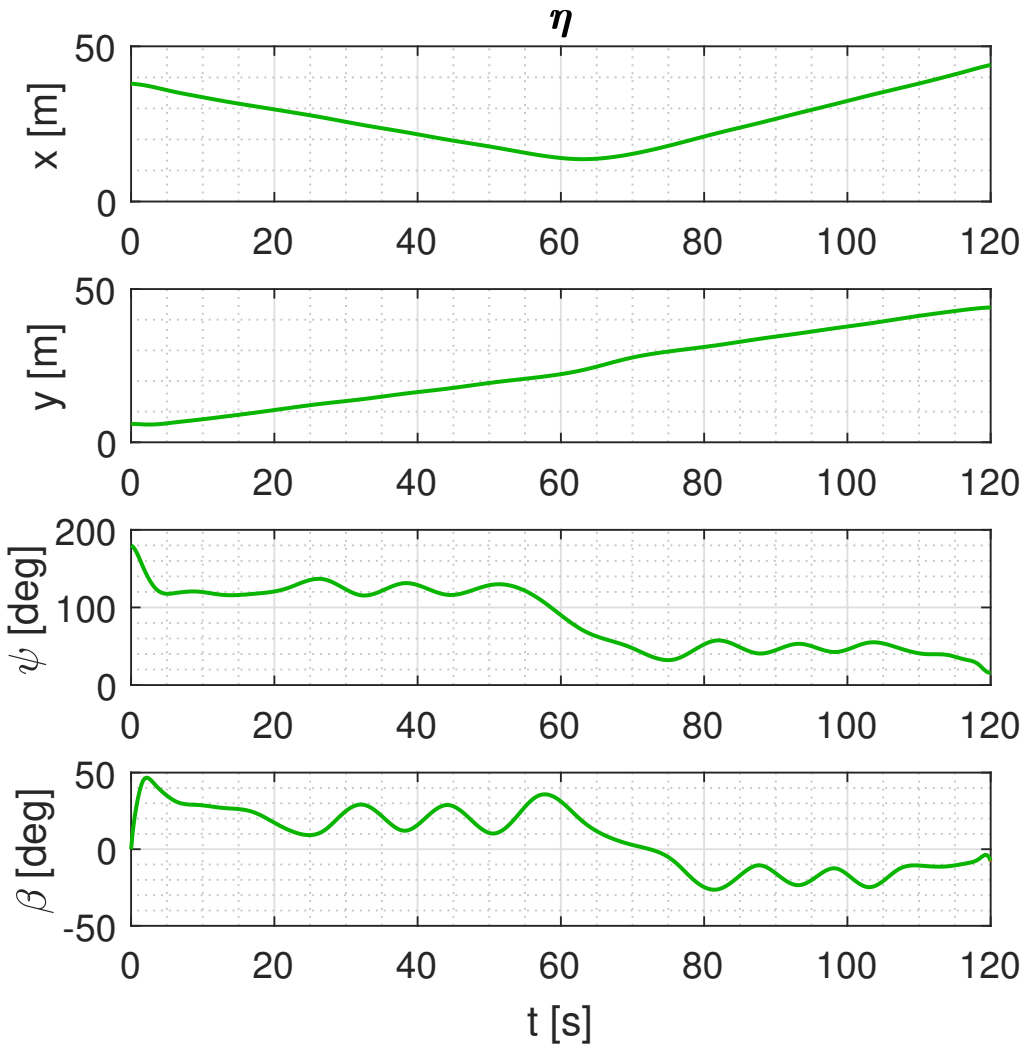


Figure 6.43: Single obstacle, western current: Pose.

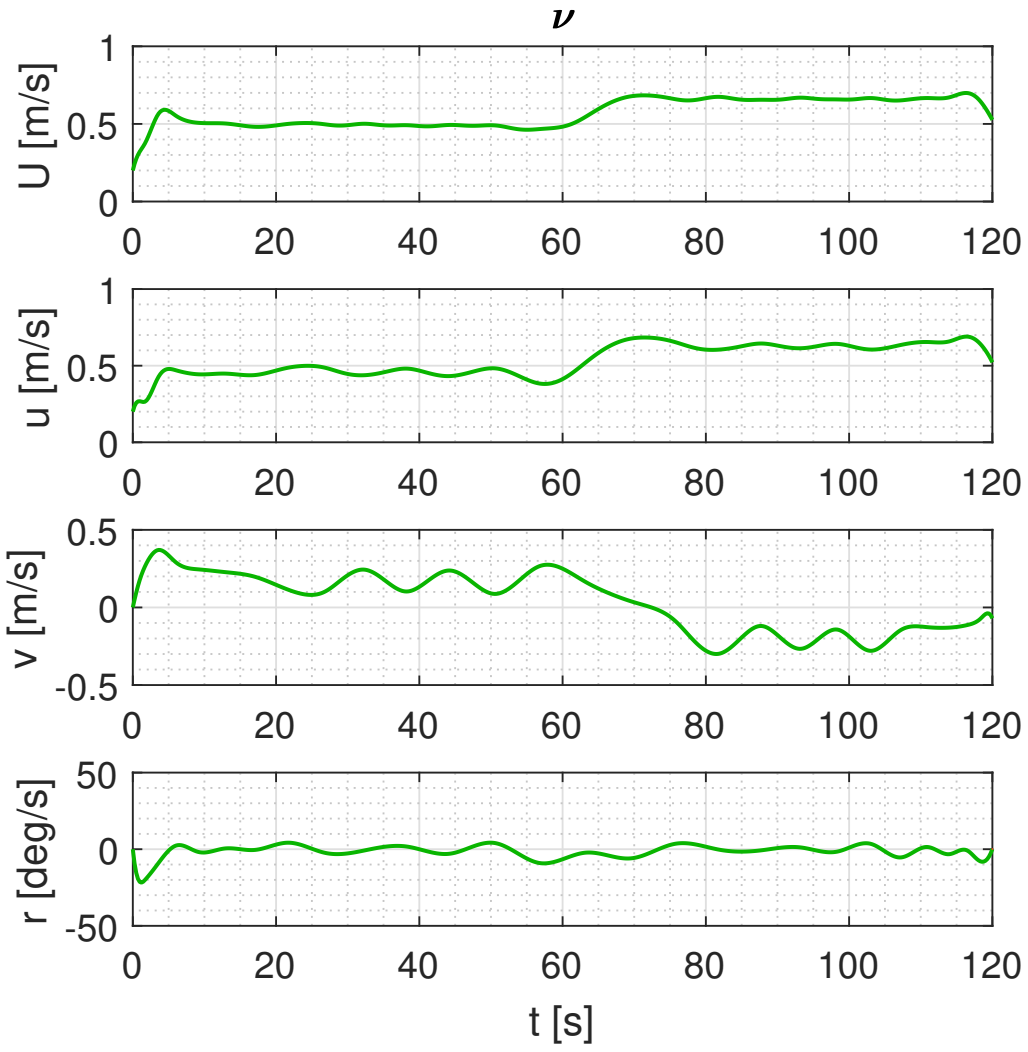


Figure 6.44: Single obstacle, western current: Absolute velocities.

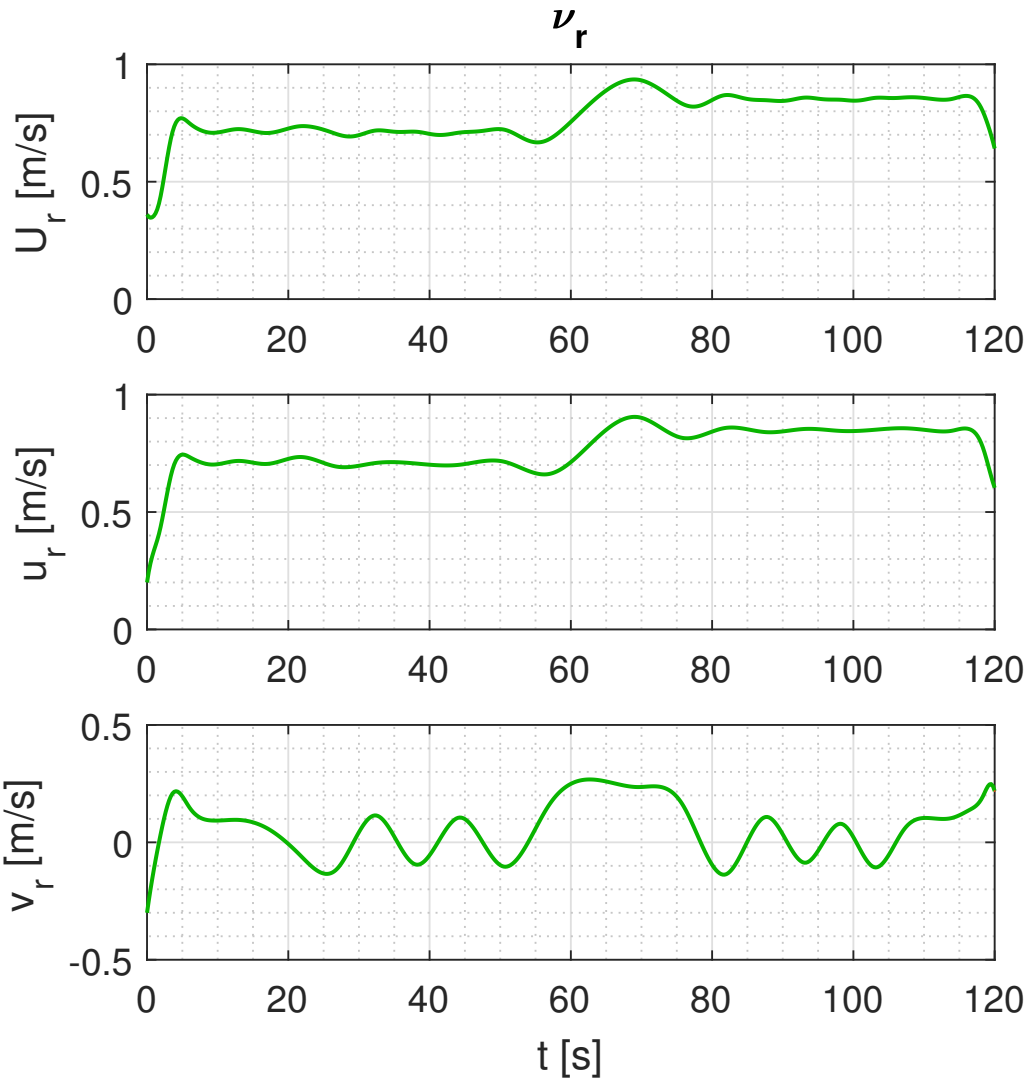


Figure 6.45: Single obstacle, western current: Relative velocities.

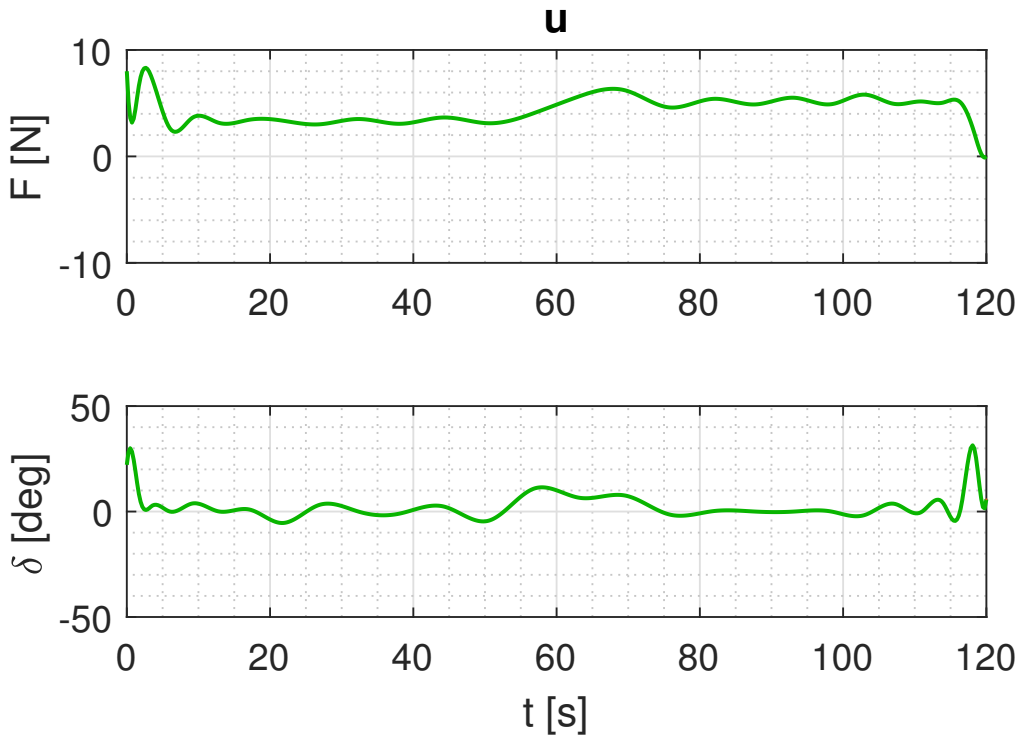


Figure 6.46: Single obstacle, western current: Control inputs.

6.4.4 Discussion

The results in this section are verified to be feasible by the method described in Section 6.2.6. For all experiments the path crosses the obstacle boundary on its southern edge. This is due to the PS algorithm only enforces constraints at the collocation points, thus interpolated solutions will not go clear of obstacles. This may be mended by using more collocation points, at the expense of computational time, or by using larger safety margins in the obstacle representations.

The different current conditions do not significantly alter the path, but heading and energy costs are significantly affected. As expected, western currents have significantly higher cost than both eastern currents and no currents. This is due to the higher relative velocity through the maneuver, which is important to the cost function. The ship uses approximately 190 % more energy traveling against the current compared to with no currents. With eastern currents, the ship uses approximately 20 % less energy. Observing the ship heading with eastern currents (Figure 6.37), the ship travels mostly perpendicular to the current direction, to exploit the drag effects. With western currents (Figure 6.42), the ship tries to

Table 6.9: Controller parameters

Parameter	Value	Parameter	Value
γ	100	$\omega_{n,\psi}$	2.0
Δ	$3L$	ζ_ψ	0.9
k_1	10	$r^d_{\max}$	0.9
$k_{2,1}$	10	$\dot{r}^d_{\max}$	0.9
$k_{2,2}$	1	$\omega_{n,u}$	0.9
$k_{2,3}$	10	ζ_u	0.9
$\mathbf{\Gamma}$	$\mathbf{I}_{3\times 3}$		

head parallel to the current most of the time. With the exception of the no-current scenario, the algorithm spends around 30 seconds on completing the optimization.

6.5 Open-loop and closed-loop simulations

In this section open-loop and closed-loop simulations are performed of the results in Section 6.4 — the single-obstacle scenarios. The open-loop simulations are performed by performing a RK 45 simulation of (6.3) with the same initial conditions used in the optimization. Spline-interpolated control signals $\boldsymbol{u}$ from the optimization are fed to the ship model through the actuator model.

The closed-loop simulations are performed using the low-level controller from Section 3.3.1 with dynamic α_2 as shown in (3.46). The force vector produced from the low-level controller is fed directly into the ship model, without using an actuator model and control allocation. The guidance algorithm from Section 5.5.2 is employed, where the desired path $\boldsymbol{p}^d(\theta)$ and a desired surge speed $u^d(\theta)$ are taken from the optimization. The heading from the optimization is ignored, and the guidance and control systems control the heading. The path parameter θ is the optimization time variable. Table 6.9 contains the parameters used in the simulations.

Each of the scenarios produce two figures, one for the path, and one for the kinematic guidance errors s (along-track error) and e (cross-track error). The path figures in this section contain the obstacle and four path legends:

NOM is the nominal path produced by the path planning algorithm.

ON is the open-loop simulation without disturbance.

OD is the open-loop simulation with disturbance.

CN is the open-loop simulation without disturbance.

CD is the closed-loop simulation with disturbance.

The disturbance is an additional current of 0.15 m/s south added to the original scenario. The results containing the energy cost for each simulation are found in Table 6.10 for reference.

Table 6.10: Single obstacle simulation: Results

Current	Nominal cost J	Open-loop no dist. cost J	Open-loop with dist. cost J	Closed-loop no dist. cost J	Closed-loop with dist. cost J
None	142.7	142.8	140.0	171.4	252.9
Eastern	112.3	114.6	109.1	197.1	254.9
Western	413.5	414.0	411.1	432.6	550.6

6.5.1 No current

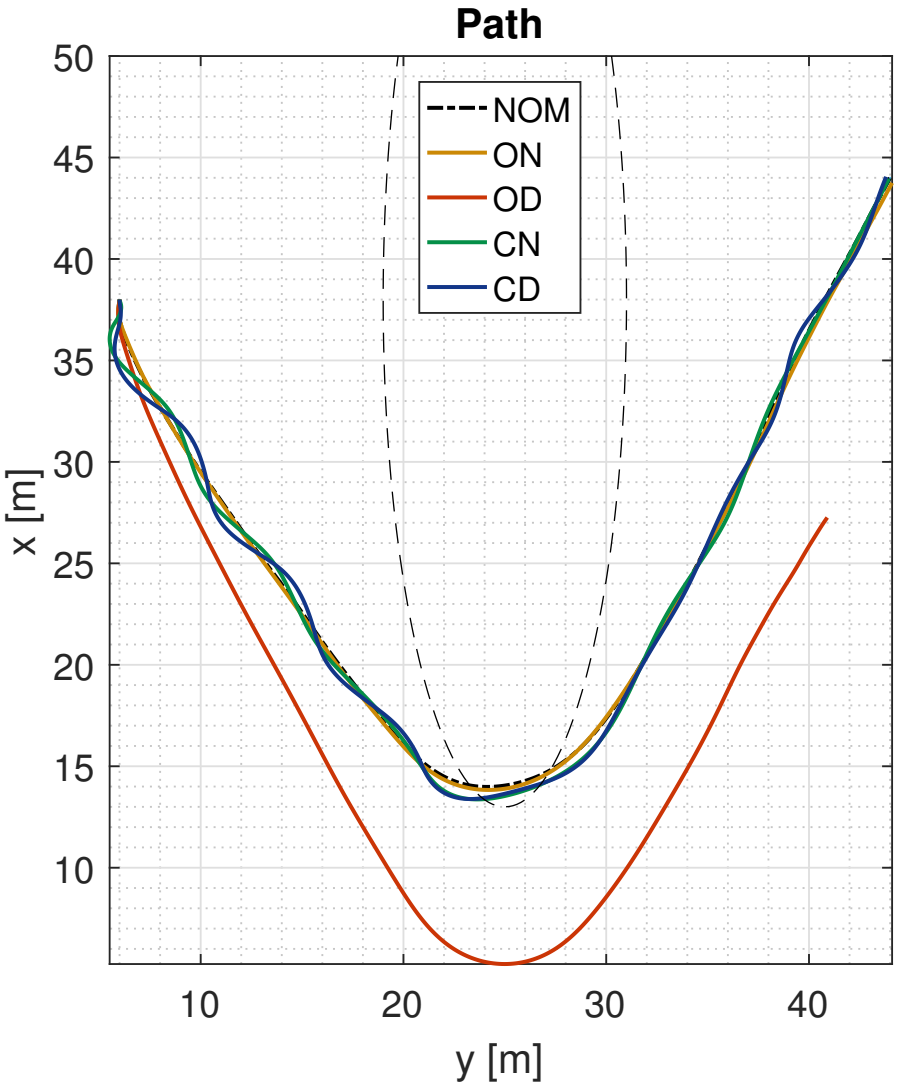


Figure 6.47: Open and closed loop, no current: Path.

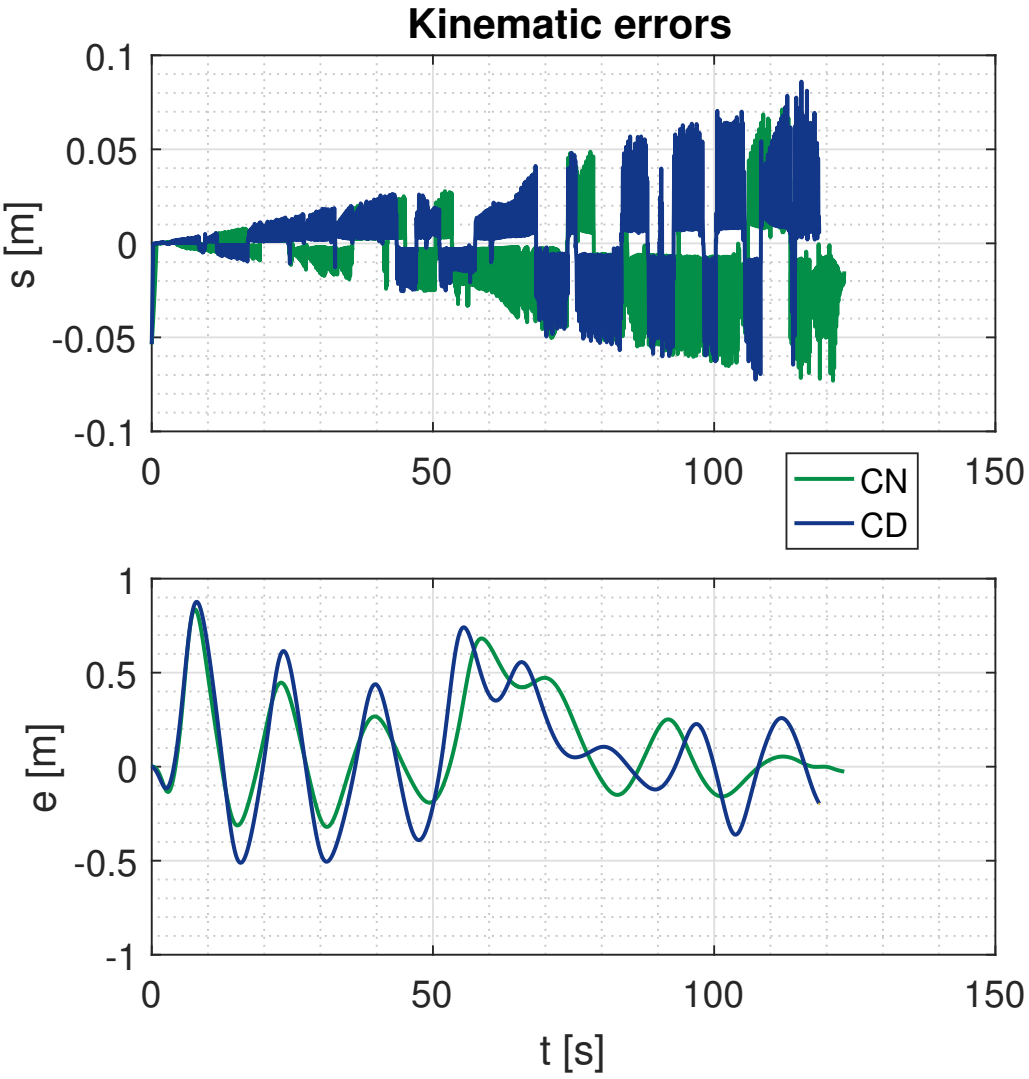


Figure 6.48: Open and closed loop, no current: Kinematic errors.

6.5.2 Eastern current

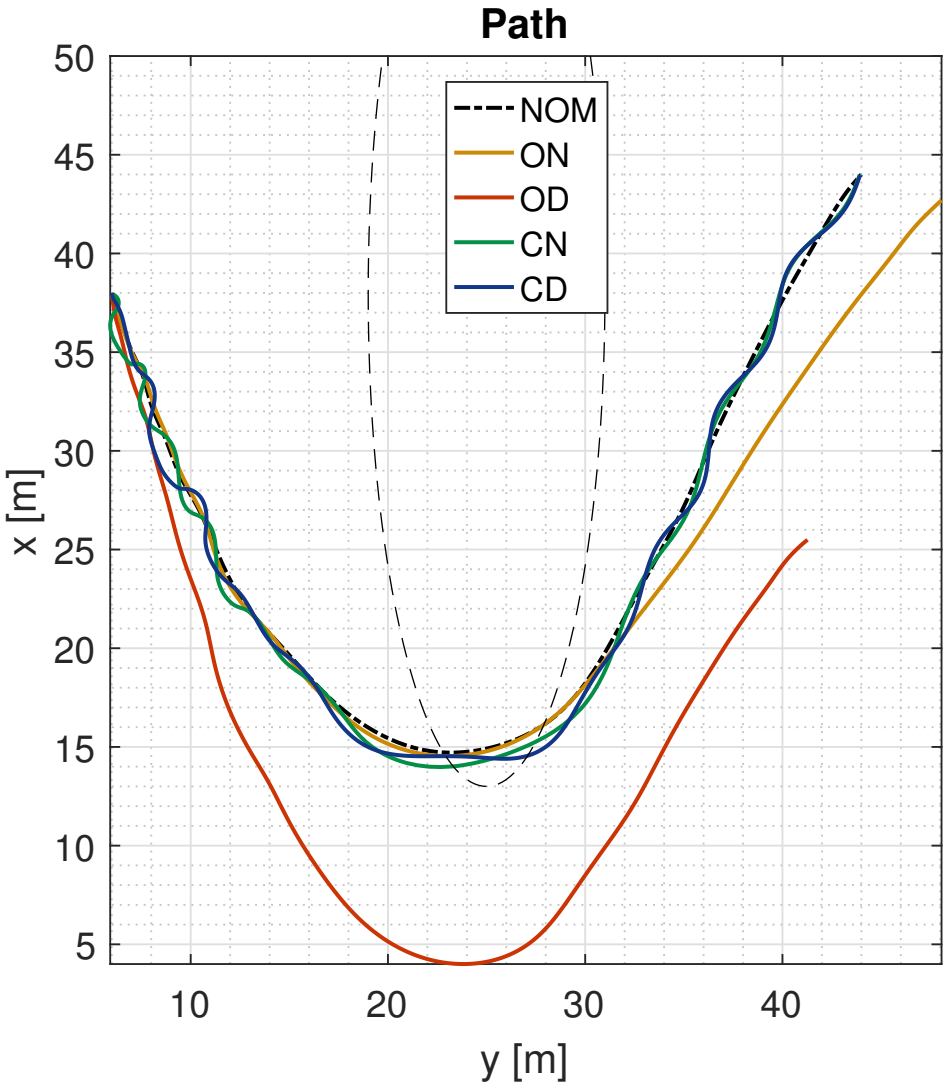


Figure 6.49: Open and closed loop, eastern current: Path.

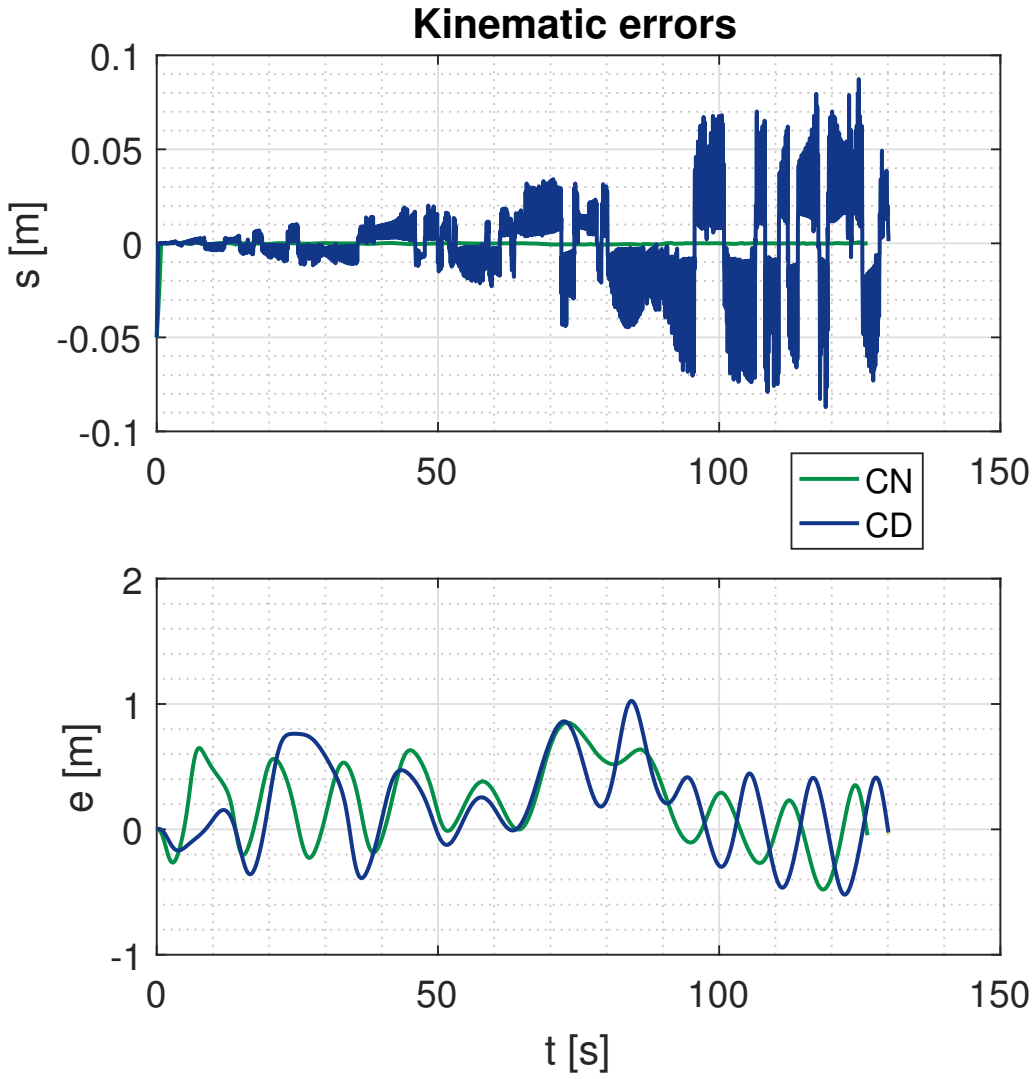


Figure 6.50: Open and closed loop, eastern current: Kinematic errors.

6.5.3 Western current

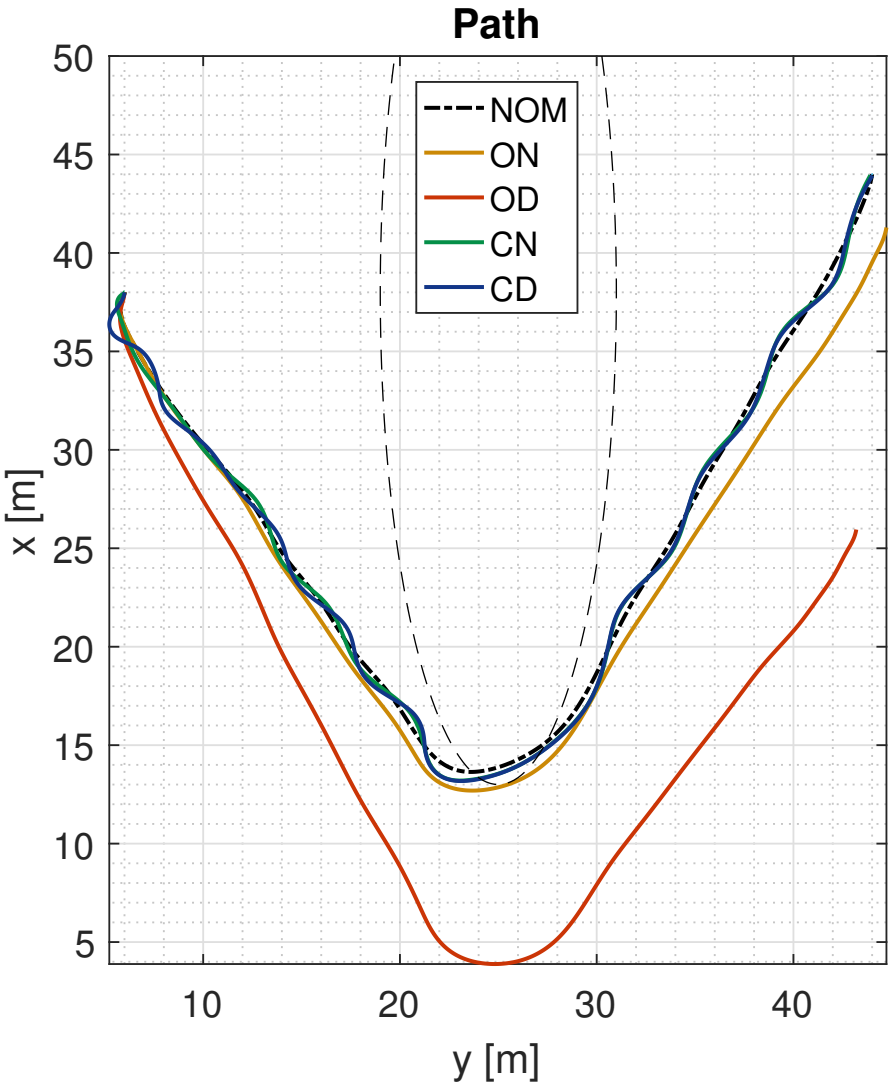


Figure 6.51: Open and closed loop, western current: Path.

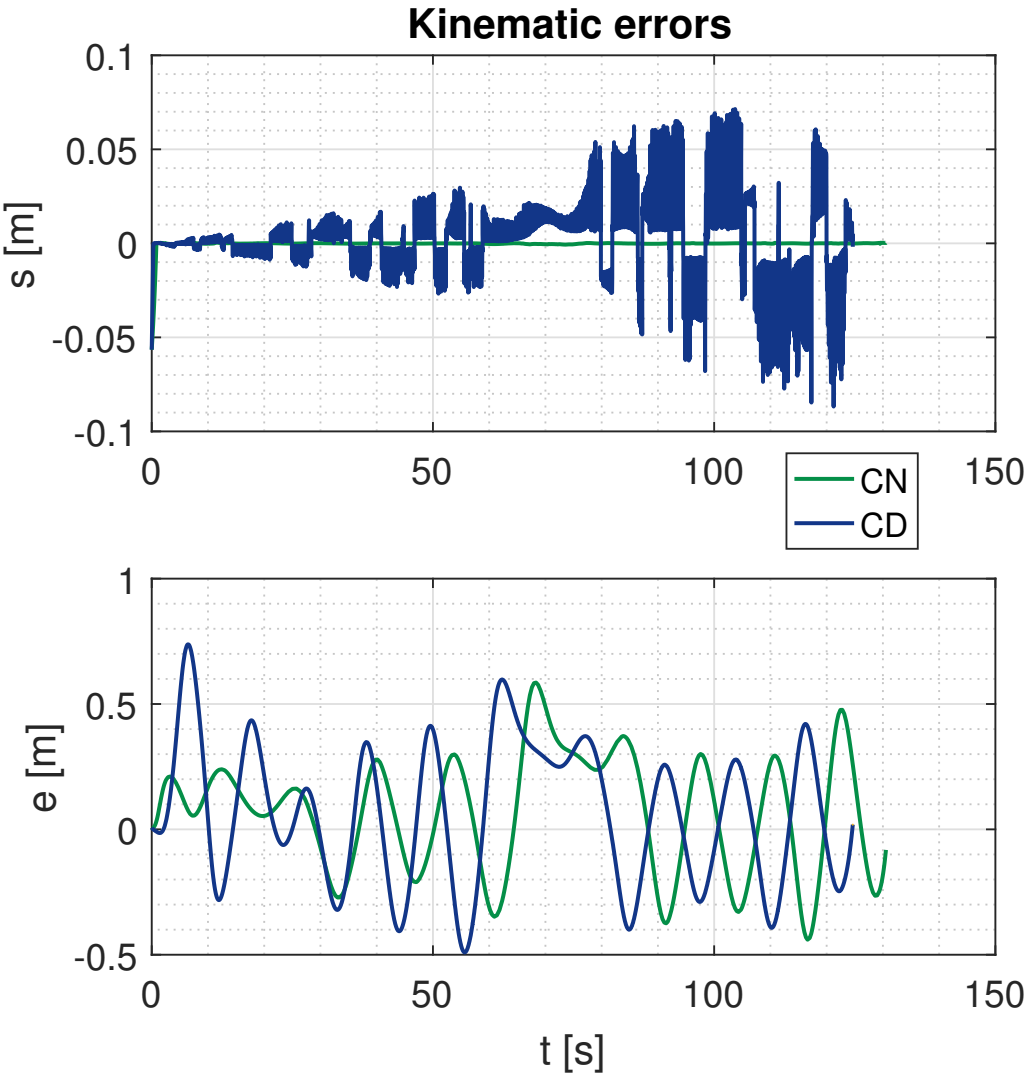


Figure 6.52: Open and closed loop, western current: Kinematic errors.

6.5.4 Discussion

With small deviations, the open-loop simulations using the nominal control inputs follow the nominal paths. As expected, the path is shifted south when the disturbance is added. This implies the need of a guidance and control system, which is also presented in the results. By closing the loop, the ship follows the nominal path closely. Some oscillations are seen, which may be caused by inadequately tuned controllers, but the cross-track errors stay below 1 m for all ocean current scenarios.

Table 6.10 shows the costs from the simulations. Closing the loop gives from 5 % to 80 % increase in energy cost compared to nominal in equal conditions. Adding the disturbance increases the energy cost by an additional 30 % to 50 %.

The results show that the optimization algorithm creates feasible paths that are possible to follow using guidance principles.

Chapter 7

Collision Avoidance

A crucial requirement for autonomous vessels are that they must be able to detect and avoid any obstacle, be it a kayak, diver or another ship. This chapter will introduce the different concepts encountered in collision avoidance, such as *local* and *global* methods, as well as the concepts of *reactive* and *deliberate* methods.

Some relevant studies on collision avoidance for autonomous ships are presented here: Statheros et al. introduce collision avoidance concepts for autonomous ships in [92]. Tam et al. reviews studies on collision avoidance in [93].

After introducing the collision avoidance hierarchy, more attention will be paid to the local methods, which focus on avoiding obstacles that are close to the ship. The global methods are mainly path planning methods augmented with kinematic information of obstacles, and are treated thoroughly in Chapter 5. Lastly, sensors used for navigation and obstacle detection are considered, along with a short introduction on sensor fusion. Figure 7.1 shows a motion control system relevant to collision avoidance.

7.1 Preliminaries

To formalize the definition of collision avoidance, some notation will be introduced. A ship exists in the workspace $\mathcal{W} = \mathbb{R}^2$, which is the two-dimensional world. The area of the ship can be seen as a subset of the workspace: $\mathcal{A} \subset \mathcal{W}$. Here $\mathcal{A}$ is a function of the configuration vector $\boldsymbol{\eta} \in \mathcal{C}$:

$$\mathcal{A}(\boldsymbol{\eta}) = \mathcal{A} : \mathcal{C} \to \mathcal{W} \tag{7.1}$$

As stated in Section 3.2, the configuration space consists of a point in $\mathcal{W}$ and a heading angle: $\mathcal{C} = \mathcal{W} \times \mathrm{S}$.

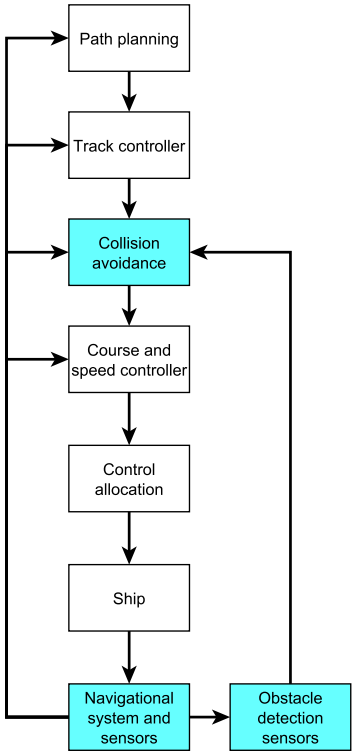


Figure 7.1: Block diagram of a motion control system. The highlighted blocks are relevant to the discussion of collision avoidance.

We let $\mathcal{O} \subset \mathcal{W}$ define the part of the workspace which is occupied with obstacles. Then the obstructed configuration space becomes

$$\mathcal{C}_{\text{obs}} = \{\boldsymbol{\eta} \in \mathcal{C} | \mathcal{A}(\boldsymbol{\eta}) \cap \mathcal{O} \neq \emptyset\}. \tag{7.2}$$

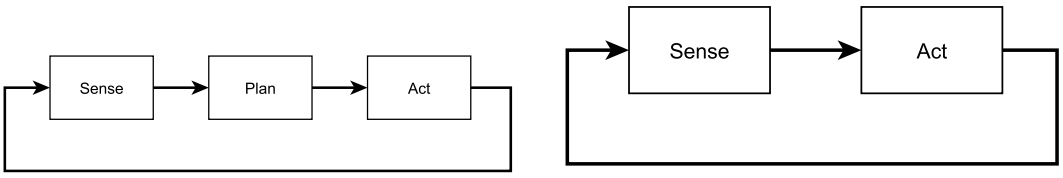
The complement of the obstructed configuration space is the free configuration space:

$$\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \mathcal{C}_{\text{obs}}. \tag{7.3}$$

The goal of collision avoidance is to stay outside of danger of entering $\mathcal{C}_{\text{obs}}$.

7.2 Deliberate and reactive methods

Deliberate methods are based on the *sense, plan, act paradigm*. The *sense* part consists of discovering information about the environment. The *plan* part involves making decisions based on the sensed information, the known environment (a map) and the goal. The *act* part is to perform the actions that were planned.



(a) Deliberate concept. **(b)** Reactive concept.

Figure 7.2: Block diagrams of deliberate and reactive methods.

Should unforeseen circumstances occur, it is necessary to replan the actions while considering the new information.

Reactive methods sense and act continuously, and react to new information as it appears. A characteristic of reactive methods is that they do not involve planning, but immediately react to sensed information. Reactive methods are generally faster than deliberate methods. The differences of the methods are illustrated in Figure 7.2.

An observation to make when studying collision avoidance methods are that most local methods are reactive, and the global path planning methods are inherently deliberate. Notwithstanding this observation, one can construct a global reactive system, as well as a local deliberate system. Local versus global is a matter of the scope of the information used, and reactive versus deliberate depends on whether the system creates a plan.

7.3 Global methods

Global methods take into account all available information in the workspace. If the paths of nearby vessels are known, path planning algorithms may include their kinematic information to generate paths that stay clear of other ships. The path planning algorithm then becomes a global collision avoidance method, as it includes the map *and* kinematic information of other obstacles.

Two concepts of global collision avoidance methods are listed here, and they are given a thorough review in Chapter 5.

Roadmap methods: An approach based on generating waypoints which when connected form a feasible path from A to B.

Optimization methods: An approach based on generating a *complete* path from A to B, optimized to some objective, subject to dynamic constraints.

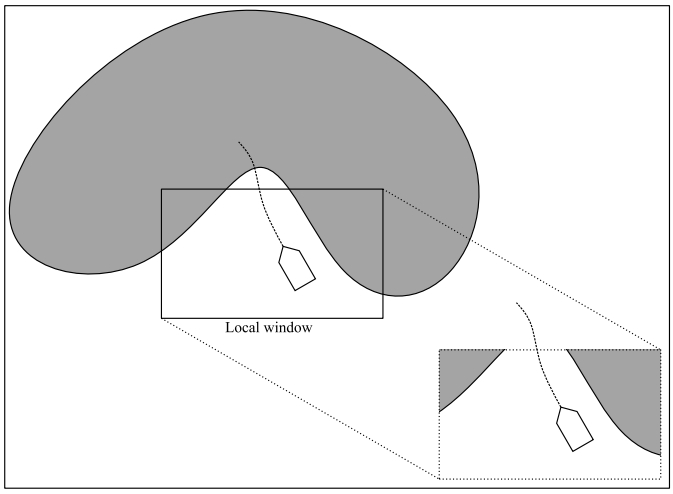


Figure 7.3: Illustration of a local convergence issue.

7.4 Local methods

Local methods are often reactive in nature, as they only focus on avoiding local obstacles, and do not include planning. These methods are *local*, because they don't include the "full picture," such as the entire map. The local methods only react to the environment around the vessel. Because of this, a local method is not guaranteed to converge to the goal. Figure 7.3 illustrates this issue. On the other hand, the local methods are orders of magnitude faster than global methods in general.

While global information may be surveyed beforehand or gathered from a map, local information, such as position and velocity of surrounding obstacles, is harder to obtain. Because of the strict requirement of being able to sense and avoid *all* obstacles, equally strict requirements are placed on the sensors and algorithms that are responsible for detecting such obstacles. In addition, when multiple sensors are used for this purpose, the idea of *sensor fusion* is useful and necessary to combine the measurements. Among feasible sensors are: Radar, LIDAR and cameras. These sensors are discussed in detail in Section 7.7.3.

A selection of local collision avoidance methods are listed here, and a more thorough introduction is provided later in this chapter:

- Dynamic window
- Velocity obstacles
- Artificial potential field

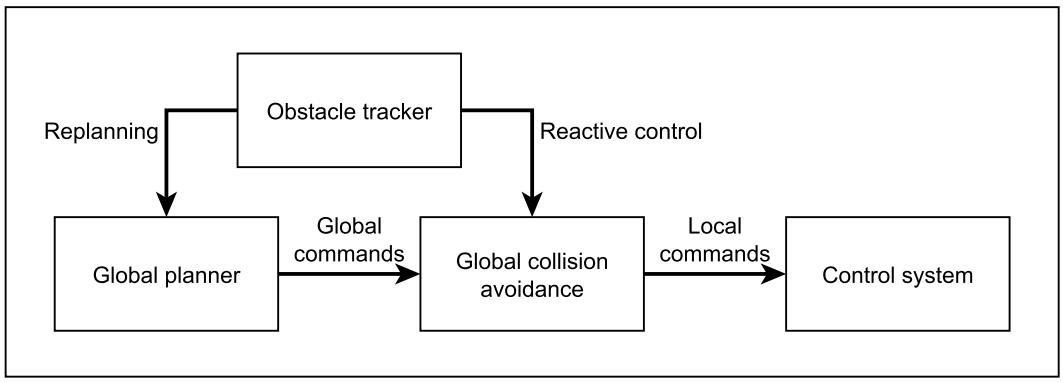


Figure 7.4: Layered hybrid collision avoidance scheme.

7.5 Hybrid approaches

From the previous two sections it is clearly seen that either a global or a local method alone is not enough for a ship to operate on its own. The global methods lack the ability to react quickly to changes in the surroundings, while the local methods lack guaranteed convergence. Hybrid approaches are attempts to combine the complementing local and global methods.

Loe suggests a method where a combination of cell decomposition with A* search and the RRT algorithm provides the global path, and the dynamic window algorithm provides local collision avoidance [32]. By tuning cost functions of both the RRT algorithm and the dynamic window, the system is able to handle COLREGS situations well.

Casalino et al. suggest a three-layered architecture for obstacle avoidance in [59]. The first layer is a global planner which only takes into account static obstacles, and the *visibility graph* method is used to generate a path. The second layer takes care of kinematic data of moving obstacles, assuming constant velocities of the tracked obstacles. A bounding box of a tracked obstacle is extrapolated assuming constant velocity, to where an intercept would take place. A visibility-graph like approach is employed to create a path around the projected track. The third layer is not discussed in the paper, but is a last-resort option, where the sole focus is to avoid imminent collision.

Loe's method of combining a global and a local algorithm involves feeding a desired course from the global path tracker, to the local dynamic window algorithm. The dynamic window algorithm selects headings and speeds that are as close to the global commands as possible, while making sure collisions are avoided. An illustration is provided in Figure 7.4.

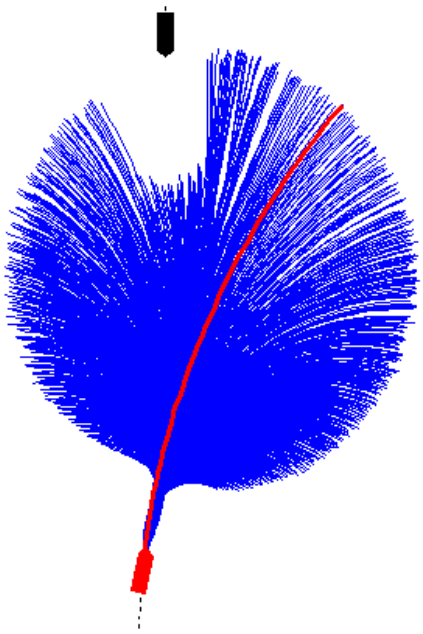


Figure 7.5: Allowable trajectories generated by the dynamic window algorithm. From [32]. © 2008 Øivind A. G. Loe.

Rather than planning the global path once, and trying to return to that path after every reactive encounter, it is possible to replan regularly, or when triggered by such an event. This may ensure optimality even after moving away from the nominal path. In addition, new kinematic information about surrounding objects may be included in the global algorithm, which will adapt to the new situation.

7.6 A selection of local collision avoidance methods

A selection of local collision avoidance methods are introduced in this section. In addition to this, two recent resources on COLREGS-compliant collision avoidance are provided: Woerner presents a multi-objective optimization concept for COLREGS-compliant collision avoidance in [94]. Additionally, Benjamin introduces a COLREGS-compliant method for collision avoidance, involving selecting behavior modes according to the COLREGS in [95].

7.6.1 Dynamic window

The goal of the dynamic window algorithm is to provide a set of desired *velocity pairs* which are "safe" for avoiding collision. The velocity pair is a surge speed u and a yaw rate r. Loe uses this algorithm to generate feasible surge speeds and yaw rates in [32].

The dynamic window algorithm was first proposed by Fox et al. as a motion planning algorithm for a car-like robot [96]. As the name suggests, the algorithm works with the set of velocity pairs (u, r) which are reachable within the *dynamic window* from the ship's current state. The dynamic window is the time interval T, and the reachable velocities are

$$\begin{aligned} V_d = &\{(u,r) \in \mathbb{R} \times \mathbb{R} \mid \\ &u \in [u^* + \dot{u}_{\min} T,\ u^* + \dot{u}_{\max} T] \wedge r \in [r^* - \dot{r}_{\max} T,\ r^* + \dot{r}_{\max}]\}. \end{aligned} \tag{7.4}$$

Here $\dot{u}_{\min}$, $\dot{u}_{\max}$ and $\dot{r}_{\max}$ are the maximum accelerations and decelerations of surge speed and yaw rate, respectively. Additionally, the algorithm works with the set limited by the upper and lower bounds on the velocities u and r, V_s. Finally, the set V_a contains the velocities which guarantee that the ship does not collide with an obstacle:

$$\begin{aligned} V_a = \Big\{(u,r) \in \mathbb{R} \times \mathbb{R} \,|\, u \leq \sqrt{-2 \cdot \mathrm{dist}(u,r) \cdot \dot{u}_{\min}} \\ \wedge\ |r| \leq \sqrt{2 \cdot \mathrm{dist}(u,r) \cdot \dot{r}_{\max}}\Big\} \end{aligned} \tag{7.5}$$

The search in the feasible space is defined here:

Definition 7.1 (Dynamic window search)**.**

Maximize

$$G(u,r) = \sigma(\alpha \cdot \mathrm{heading}(u,r) + \beta \cdot \mathrm{dist}(u,r) + \gamma \cdot \mathrm{velocity}(u,r)) \tag{7.6a}$$

s.t.

$$(u,r) \in V_r \tag{7.6b}$$

σ denotes a low pass filter which reduces noise. α, β and γ are tuning parameters which place weights on the different cost functions. The functions heading(u,r) and velocity(u,r) make the search prefer yaw rates and surge speeds that are as close to the global desired values as possible. The function dist(u,r) makes the search prefer the surge speeds and yaw rates that make the ship travel as far as possible. In addition we have $V_r = V_d \cap V_s \cap V_a$.

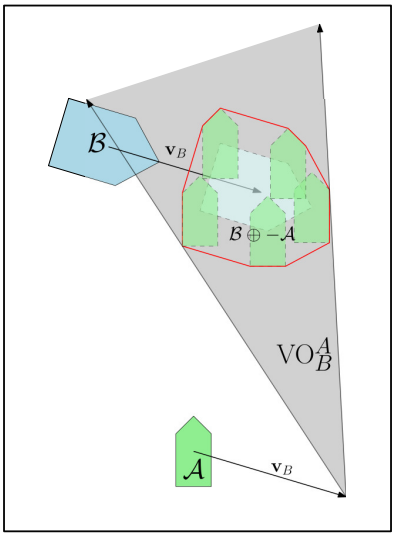


Figure 7.6: Illustration of velocity obstacles. From [100]. © 2016 Bjørn-Olav H. Eriksen.

COLREGS may be accounted for by including a cost function which penalizes situations described by the regulations. This may be based on the angle of approach between the two vessels. Such an implementation is done with success in [32].

Other variations of the algorithm include a method called *global dynamic window approach*, where global information is included in the objective function. This variation was introduced by Brock and Khatib in [97]. Eriksen provides an alternative trajectory prediction and search space for the dynamic window algorithm in [98].

7.6.2 Velocity obstacles

Velocity obstacles is a way of avoiding collisions by determining obstacles in the velocity space. Kinematic information is included by assuming linear velocities, and safe velocities are determined by projecting the areas of both the ship and the targets ahead in time. Myre uses velocity obstacles for COLREGS collision avoidance in [30]. That work is based on the work of Kuwata et al. in [99].

The velocity obstacle becomes a cone in the velocity space. For the ship A with shape $\mathcal{A}$ trying to avoid the obstacle B with shape $\mathcal{B}$, the velocity obstacle is described by the following equation:

$$VO_B^A(\boldsymbol{v}_B) = \{\boldsymbol{v}_A \mid \lambda(\boldsymbol{p}_A, \boldsymbol{v}_A - \boldsymbol{v}_B) \cap (\mathcal{B} \oplus -\mathcal{A} \neq \emptyset)\} \tag{7.7}$$

Here, λ is a ray described by

$$\lambda(\boldsymbol{p}_i, \boldsymbol{v}_i) = \{\boldsymbol{p}_i + t\boldsymbol{v}_i \mid t \geq 0\}, \tag{7.8}$$

and $\boldsymbol{p}_i$ and $\boldsymbol{v}_i$ are the position and velocity of ship i, respectively. In addition, we have the following operations:

$$\text{Minowski sum:} \quad \mathcal{A} \oplus \mathcal{B} = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{A}, \boldsymbol{b} \in \mathcal{B}\} \tag{7.9}$$

$$\text{Reflection:} \quad -\mathcal{A} = \{-\boldsymbol{a} \mid \boldsymbol{a} \in \mathcal{A}\} \tag{7.10}$$

The concept is illustrated in Figure 7.6, where the gray cone is the velocity obstacle.

The velocity space is discretized, and a cost function based on time to collision and deviation from global reference is applied to find the optimal velocity vector. COLREGS compliance is implemented by creating new constraints based on the angle of approach between the vessels.

7.6.3 Artificial potential field

Khatib introduced the artificial potential field method in 1985 [101]. The basis of the method is a potential field defined by the goal position $\boldsymbol{x}_d$ and obstacles $\mathcal{O}$:

$$U_{\text{art}} = U_{\boldsymbol{x}_d}(\boldsymbol{x}) + U_{\mathcal{O}}(\boldsymbol{x}) \tag{7.11}$$

Here $\boldsymbol{x}$ is the current position of the robot, $U_{\boldsymbol{x}_d}$ is the attractive field of the goal, and $U_{\mathcal{O}}$ is the repulsive field from the obstacles. With this field defined, the method imposes a desired force on the robot trying to reach the goal, based on the gradient of U_{art}:

$$\boldsymbol{F} = -\nabla U_{\text{art}}(\boldsymbol{x}) \tag{7.12}$$

Heuristics may be applied to achieve asymptotic convergence, such as dissipation on the attractive field.

The artificial potential field method has major drawbacks, which include local minima where repulsive forces cancel the attractive forces, prohibiting convergence to the goal. Oscillatory motion is often encountered near multiple obstacles or narrow corridors. The method assumes a fully actuated robot, applying forces in all directions — this is not feasible for an underactuated ship. There is no straight-forward way of implementing the COLREGS in this algorithm.

7.6.4 Comparison of local collision avoidance algorithms

Both the dynamic window and velocity obstacle algorithms consider the feasible directions and speeds which avoid collision. As opposed to the dynamic window

algorithm, velocity obstacles considers the obstacle's velocity to project its location ahead in time. Both algorithms use an optimization search to find the best velocity based on some cost on deviation from a global velocity reference, and other measurements. COLREGS are admissible in both algorithms, either in the optimization search or in the rules which decide the feasible velocities. The dynamic window algorithm considers the ship dynamics, while velocity obstacles does not. The dynamic window algorithm does not account for movement of the obstacles, but treats them as static objects.

The artificial potential field algorithm is simple and intuitive to understand and implement, but suffers from several drawbacks as mentioned in Section 7.6.3. Among them are local minima, oscillatory behavior, and no straight-forward way of implementing the COLREGS.

7.7 Sensors used for navigation and obstacle detection

Any algorithm used for collision avoidance requires information about the positions and possibly the velocities of nearby obstacles. As safety is the first priority of any ferry operation, it is imperative that the ship is able to detect and track dynamic obstacles. In addition to obstacle detection, navigation sensors are also required for most collision avoidance algorithms, as well as guidance algorithms. This section introduces a range of navigation and obstacle detection sensors.

7.7.1 Equipment for position measurement

Sensors available for position measurement may be divided into local and global techniques. Global sensors measure a receiver's position relative to the earth-centered, earth-fixed (ECEF) reference frame. The prime examples are GNSS and their augmented versions DGPS and real-time kinematic (RTK), which improve accuracy. Local sensors are based on local stations with known positions, and some kind of transponder mounted on the vessel. These systems measure position relative to a local frame. The most relevant local techniques are acoustics such as ultra-short baseline (USBL), short baseline (SBL), and long baseline (LBL). Other systems include radar-based (Artemis, RADius), and laser-based (SpotTrack, Fanbeam, and CyScan). Information on these systems may be found in the following references: [102, 103, 104, 105].

Both the global and local position sensors are commonly augmented or fused with INS. This gives a faster update to the position estimate, and improves accuracy.

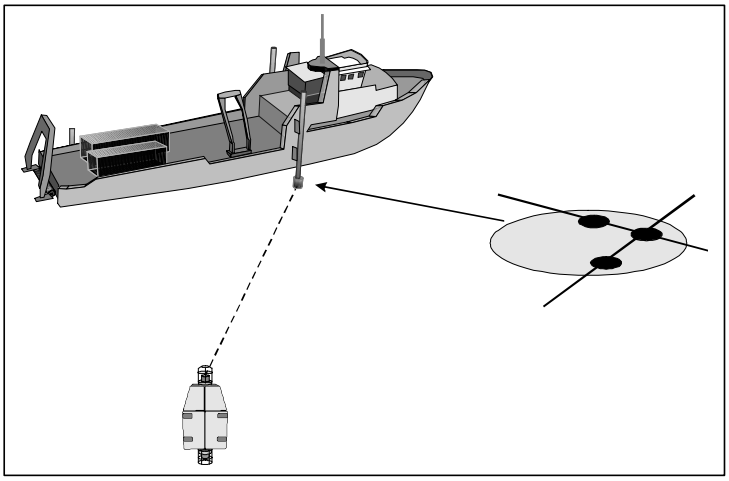


Figure 7.7: Illustration of USBL with a ship and an AUV. In the case of ferry navigation, the transponder mounted on the AUV would be mounted on the seabed instead. From [110]. © 1998 MTS Dynamic Positioning Committee.

Satellite systems (GNSS)

GNSS on its own may not be accurate enough for navigation, with Global Positioning System (GPS) having accuracy of less than 7.8 m [106].

- DGPS uses stations with exact known positions, and transmits the measured position error data to local receivers using radio communication. This information is then used to improve accuracy, and is correct to 0.5 m to 1 m plus approximately 0.2 m for each 100 km distance from the reference station [107]. This gives substantial increase in usefulness for navigation and docking operations.
- RTK works by considering the *phase* of the satellite's carrier signal, in addition to the information contained within the signal. A base station, whose position is known exactly, measures the carrier signal's phase and transmits this data to the vessel. The vessel in turn compares this data to its own phase measurement, and calculates the relative position of the station and itself with centimeter precision [108, 109].

Acoustic systems

There are several methods that are suitable for determining relative position using acoustic methods. Useful information regarding acoustic methods is found in [110] and [111].

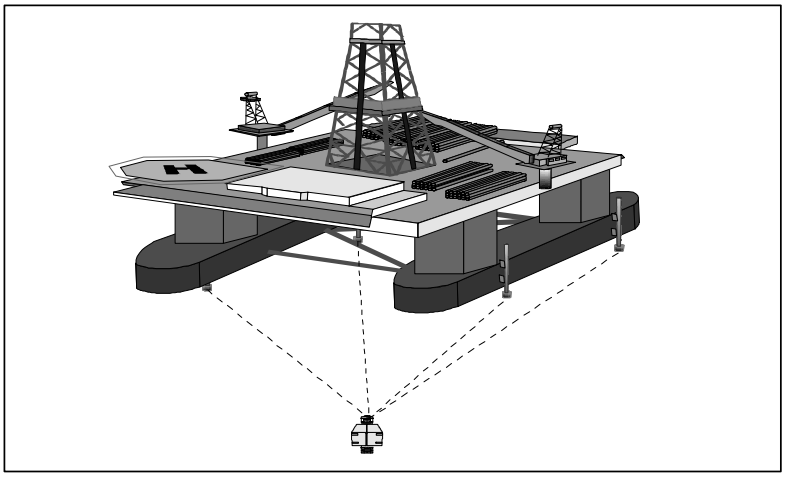


Figure 7.8: Illustration of SBL with a drilling platform and an AUV. In the case of ship navigation, the transceivers would be mounted on the ferry, and the transponder mounted on the AUV would be mounted on the seabed instead. From [110]. © 1998 MTS Dynamic Positioning Committee.

- USBL (also known as super-short baseline (SSBL)) uses a single transceiver mounted under the ship to measure range and angle to a fixed transponder, mounted at a known location on the seabed. Figure 7.7 illustrates this. The transceiver sends a sonic signal which is returned by the seabed transponder, and the run time is proportional to the distance. The transceiver consists of an array of transducers, whose phase shift of the return signal is translated to *target angle*. Combining the angle and range gives a 2D position measurement. Due to the nature of the position calculation, an error in angle measurement makes the position error increase with distance from the transponder. Accuracy of 0.1° in angle and 0.1 % (0.5 m at 500 m) is observed in the works of Watson et al. [112]. Inverted ultra-short baseline (iUSBL) is a form of USBL where the transceiver is mounted on the seabed and the transponder on the ship.

- SBL uses three or more baseline transceivers mounted on the ship, and a single transponder mounted on the seabed. This is illustrated in Figure 7.8. One of the transceivers sends a sonic signal which is received by the transponder. The transponder then returns the signal which is received by the transceivers on board the ship. The distance from each of the transceivers to the seabed transponder is proportionate to the echo delay, and a position is calculated by trilateration. Accuracies of 1 % of water depth are mentioned in the book of Milne [111] about Sintra-Alcatel Acoustic Measuring System, but this reference is from 1983, and information about SBL accuracy is sparse in the

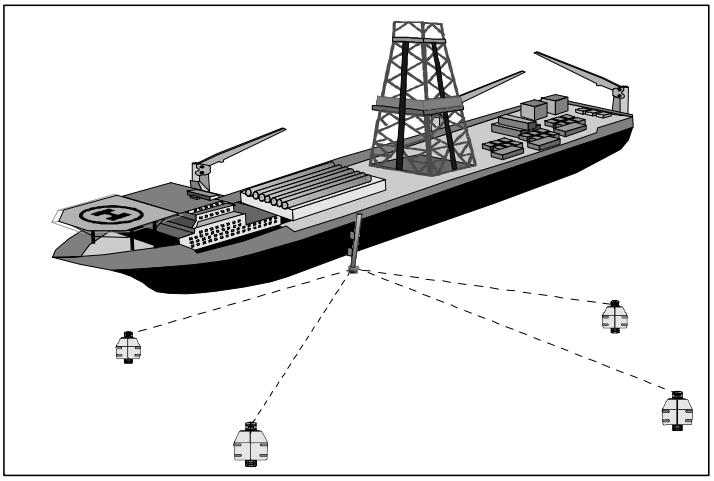


Figure 7.9: Illustration of LBL with a ship. From [110]. © 1998 MTS Dynamic Positioning Committee.

literature. Large spacing between the on-board transceivers will increase accuracy.

- LBL works by having multiple transponders mounted on the seabed, which respond to a sonic signal transmitted by a transceiver mounted on a ship. This is seen in Figure 7.9. The range from each sensor is determined by echo delay, and the position is determined by trilateration. Accuracy of LBL systems are independent of water depth, but dependent on frequency and spacing of the seabed transponders. Typical accuracy goes from 2 m to 5 m with low frequency (LF) (8 kHz to 16 kHz) to less than 0.01 m with very high frequency (VHF) (200 kHz to 300 kHz). Increased accuracy comes at a cost; while LF reaches tens of kilometers, VHF may stop before 100 meters [110].

Radar and laser systems

Radar and laser local positioning systems work with the same principle, although they have differing technology. They consist of a rotating transmitter and scanner on board the ship, which transmits a pulse signal in the horizontal plane. A stationary reflector placed on land or on a platform reflects this signal, which is read by the on-board scanner. The echo delay is proportionate to the distance from the reflector, and the angle is read out from an encoder. The equipment mentioned in this section is most commonly used in DP operations.

- Artemis MK V is a radar system which works in ranges up to 5 km [102]. It has a mobile station mounted on a ship, which communicates with a fixed station through radar microwaves. The rotating mobile radar station aligns

to the fixed station and measures the range. The angle is read out from an encoder. This equipment has an accuracy of 1 m distance, and 0.02° angle. Radar range measurements are resistant to bad weather conditions, such as snow, heavy rain and fog.

- Fanbeam is a laser system which uses a laser with a 20° beam as the mobile transmitter [105]. The laser rotates and aligns to a fixed reflective station, most commonly a *retroreflector prism*. The range and angle measurements also work by echo delay and encoder, respectively. The laser pulses with a frequency of 7.5 kHz. The Fanbeam has a recommended range of less than 500 m for DP operations, and maximum 2 km in good weather conditions. The performance is degraded in bad weather conditions. The accuracy is ±20 cm and ±0.02° for range and angle, respectively.

- CyScan is another laser system that works much in the same way as Fanbeam [105]. The major difference is that it rotates at a constant speed of $360\,^\circ\,\mathrm{s}^{-1}$ and emits laser pulses at 30 kHz. Range and angle is measured by echo delay and encoder, respectively. The range is more than 1000 m using a prism reflector as the target, 400 m with an aluminum plate, and 200 m with a cylindrical target. The accuracy is 10 cm at 200 m range, and degrades with increasing range. It is possible to use multiple targets to enhance accuracy and provide redundancy.

7.7.2 Equipment for heading measurement

Measuring only position is not sufficient to avoid obstacles. The position measurement is only an indication of where a certain point on the ship's body is located. The heading of the ship must also be measured.

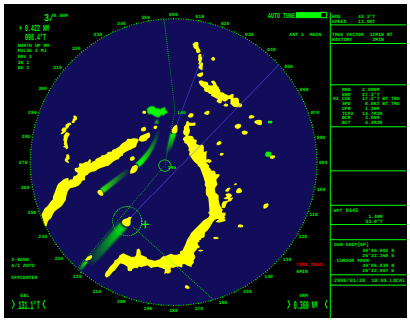
The ship takes up an area in the workspace $\mathcal{W} = \mathbb{R}^2$. The map

$$\mathcal{A}(\boldsymbol{\eta}) = \mathcal{A} : \mathcal{C} \to \mathcal{W} \tag{7.13}$$

describes the subset of $\mathcal{W}$ taken up by the ship, defined by the pose $\boldsymbol{\eta} = [x, y, \psi]^\top$ and the vessel shape.

Position measurement instruments determine x and y, while the most common sensor to determine ψ is a gyrocompass. The gyrocompass is a non-magnetic compass which is based on *gyroscopic precession* to determine heading. It finds true north as opposed to magnetic north, and is not affected by magnetic disturbances, as opposed to a magnetic compass.

(a) Ship radar. CC BY Wikimedia user *High Contrast*.



(b) Radar screen. CC BY Wikipedia user *Mysid*.

Figure 7.10: Radar.

7.7.3 Equipment for obstacle detection

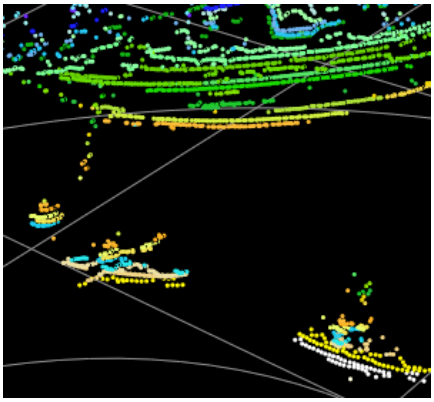
Knowing the pose of the ferry is not enough to avoid all obstacles. Comparing the position with precisely drawn map is only sufficient to avoid static obstacles, such as land and reefs. Nearby traffic must also be detected and avoided. A ship's crew can easily detect and avoid obstacles using only vision. Detecting such obstacles autonomously is quite challenging. Sensors suitable for detection include cameras, radar, light detection and ranging (LIDAR), laser range, ultrasound above water, and sonar sensors for detecting obstacles under water. These sensors have their own strengths and weaknesses, and are often used to complement each other.

Long range sensors

The sensors in this section may be used to detect obstacles at a long range (from a few meters to kilometers), and are useful for detecting obstacles during transit.

- Stereo cameras are used to generate a depth map which may in turn be used to identify obstacles [114, 115]. Care must be taken to correctly calibrate the cameras, in order to have an accurate depth map. This type of sensor works well for detecting obstacles close to the ship, but suffer from reflection in water and noise in bad weather conditions.
- Radar equipment detects obstacles by emitting microwaves which reflect off solid surfaces. The reflection is measured by the radar equipment, and the time of flight is proportionate to the distance to the target. As the radar is rotating, the angle to the target can also be measured. This creates a map with reflections, and may be used for obstacle detection. Radar based detection is described in the paper of Almeida et al. [116]. Fused radar and vision based detection for road obstacles is described in the works of Bertozzi

(a) Velodyne HDL-64E LIDAR sensor. © Velodyne.



(b) LIDAR scan of kayakers. From [113]. © 2010 SPIE.

Figure 7.11: LIDAR.

et al. [117] and the article of Kato et al. [118]. Figure 7.10 shows a radar screen and a photo of a radar. A radar works with long ranges (kilometers), but are often blind the first hundred meters [114].

- LIDAR is a technology quite similar to radar, the main difference being that it uses waves from the near-infrared, visible and ultraviolet parts of the spectrum. This is done to generate a point cloud, which in turn is used to detect obstacles. Detection of boats, kayaks, buoys, and other obstacles using LIDAR is treated in Halterman and Bruch's paper [113]. LIDAR may detect obstacles at a shorter range than radar, and with greater depth resolution than vision. Ranges down to 8 m is observed in the paper of Elkins et al. [114]. Figure 7.11 shows a LIDAR sensor and scan result.
- An active sonar is used for obstacle detection in a single forward-pointing beam under water in Heidarsson and Sukhatme's work [119]. As surface obstacles always have an underwater part, this sensor may be used for detecting obstacles. The experiment was successful with low pitch and roll angles, but as the sonar is detecting objects close to the surface, large attitude angles will cause the beam to point out of the water, which is a source of noise.

Automatic identification system (AIS)

AIS is an automatic tracking system for ships. A ship equipped with AIS transmits its position, heading and speed over VHF. This information is available to ships in the proximity, and is used for collision avoidance. The following data is transmitted:

- Maritime Mobile Service Identity (MMSI), a nine-digit unique ship identifier;
- navigation status, such as "at anchor" or "under way using engine";
- rate of turn;
- speed over ground;
- position;
- course over ground;
- true heading;
- true bearing; and
- timestamp.

Using AIS in object detection and tracking is not recommended as the sole source of information. The three main reasons for this are that not all ships carry AIS equipment, they may turn the equipment off, and the AIS data is not always reliable. As mentioned in the paper of Wilthil et al. [120], some of the issues with data reliability include *out-of-order message arrival*, and *sampling time quantization errors*.

Out-of-order message arrival is when the coordinated universal time (UTC) timestamps are not monotonously increasing for sequential incoming AIS messages. The proposed solution in [120] is to discard "old" messages. This solution introduces a reduction of sampling rate proportionate to the number of discarded messages, which is adequate if the resulting sampling time is small enough to capture significant target dynamics.

Sampling time quantization errors introduce significant measurement errors when the actual AIS sampling time is not an integer multiple of the UTC seconds. This error is introduced as a Gaussian error source to the position measurement, and for speeds over approximately $2\,\mathrm{m\,s^{-1}}$ it becomes the dominating position error for AIS.

In [120] the filtered AIS information is used for cross-validation and estimation of process noise. For further information on the reliability of AIS information, Harati-Mokhtari et al. provide a thorough analysis of the error sources and some suggestions of improvement in [121].

Proximity sensors

The sensors mentioned in this section is used for detection of obstacles in the immediate proximity of the ship.

Using multiple above-water ultrasound acoustic sensors for obstacle detection is treated in Spange's thesis [122]. Here the sensors are used for mapping the area

Table 7.1: Comparison of position measurement sensors. Accuracy and price level are given in orders of magnitude of meters and USD, respectively.

System type	Accuracy	Price level
GNSS	10^1	10^2
DGPS[1]	10^0	$10^{3–4}$
RTK GNSS	10^{-2}	10^3
USBL	10^0	10^5
LBL[2]	10^0	10^5
Radar	10^0	10^5
Laser	10^{-2}	10^5

around a ship to create a map of obstacles not detected by an on-board LIDAR sensor. A laser range sensor may be used for the same purpose.

7.7.4 Recommendation of sensor package for navigation and obstacle avoidance

As seen in tables 7.1 and 7.2, both price and accuracy vary greatly between the available sensors. The goal is to have a sensor package which is able to detect items as small as divers and kayaks with an acceptable accuracy, at a reasonable cost.

For navigation it is suggested to use DGPS or RTK to determine an accurate position. Combined with a gyrocompass, the pose estimation is assumed to be sufficiently accurate.

For obstacle detection, a combination of radar, LIDAR and stereo camera is suggested, together with AIS for cross validation. This combination complements each other range-wise. Radar has the farthest range, LIDAR covers the mid-range and cameras are necessary to use up close. This selection may be further complemented with proximity sensors to avoid immediate threats.

[1]DGPS solutions fused with IMU data for commercial purposes may have centimeter (10^{-2}) accuracy, and prices in the 10^4 USD range.

[2]LBL is available as an option to Kongsberg's HiPAP system, which mainly uses USBL for determining position.

[3]A camera does not provide depth information on its own, thus only angular accuracy is provided. Fusing two cameras for stereo vision will provide depth information, given correct calibration and their ability to connect points from different images. This is a complex endeavor and is not explored in this thesis.

[4]Significantly shorter range for small obstacles.

[5]The speed of sound in water is relatively slow, so a sonar used for long range detection will have a long sampling time.

Table 7.2: Comparison of obstacle detection sensors. Range and range accuracy are given in orders of magnitude of meters, angle accuracy is given in order of magnitude of degrees, and price level is given in order of magnitude of USD.

System type	Range	Range accuracy	Angle accuracy	Price level
Stereo camera[3]	-	-	10^{-2}	10^{4}
Radar[4]	10^{4}	10^{0}	10^{0}	10^{3}
LIDAR	10^{2}	10^{-1}	10^{-1}	10^{4}
Active sonar[5]	10^{4}	10^{0}	10^{0}	10^{3}

7.8 Sensor fusion and target tracking

Several sensors used for obstacle detection are mentioned in Section 7.7.3. While they give information regarding obstacles on their own, the raw data has very high bandwidth, and it is not straight-forward to combine the sensor information. Sensor fusion is a concept used to combine the data from multiple sensors to give basic output, such as a position and velocity of an obstacle for target tracking. The process is complex, as the sensor data usually is noisy, and may reflect waves, land and weather-phenomena.

A common way of performing sensor fusion for target tracking is to generate a list of tracks to keep. This list is populated by initialization and termination logic, which adds or removes tracks based on if there are several subsequent measurements in a region. Each object in the list has a predicted position and velocity vector associated with it. An important task is to associate measured data points with a list entry. This is called *data association*. A method for probabilistic data association is treated in [120]. Here the normalized innovation squared (NIS)[6] is used to associate a measurement with a track, shown by

$$\text{NIS} = (\boldsymbol{z}_k^i - \hat{\boldsymbol{z}}_k)^\top \mathbf{S}_k (\boldsymbol{z}_k^i - \hat{\boldsymbol{z}}_k) < \gamma_G, \tag{7.14}$$

where $\boldsymbol{z}_k^i$ is a measured point, $\hat{\boldsymbol{z}}_k$ is the priori estimate, and $\mathbf{S}_k$ is the covariance of the predicted measurement. The positive constant γ_G is a threshold used to determine if the measurement should be validated.

Filtering and prediction of target movement is commonly done using Bayesian filtering, such as the much-used Kalman filter. The *posteriori* estimate is generated by a weighted sum of the *priori* estimate and measurements. Other topics relevant to sensor fusion and tracking include

[6] Innovation is the difference between the predicted measurement and the actual measurement.

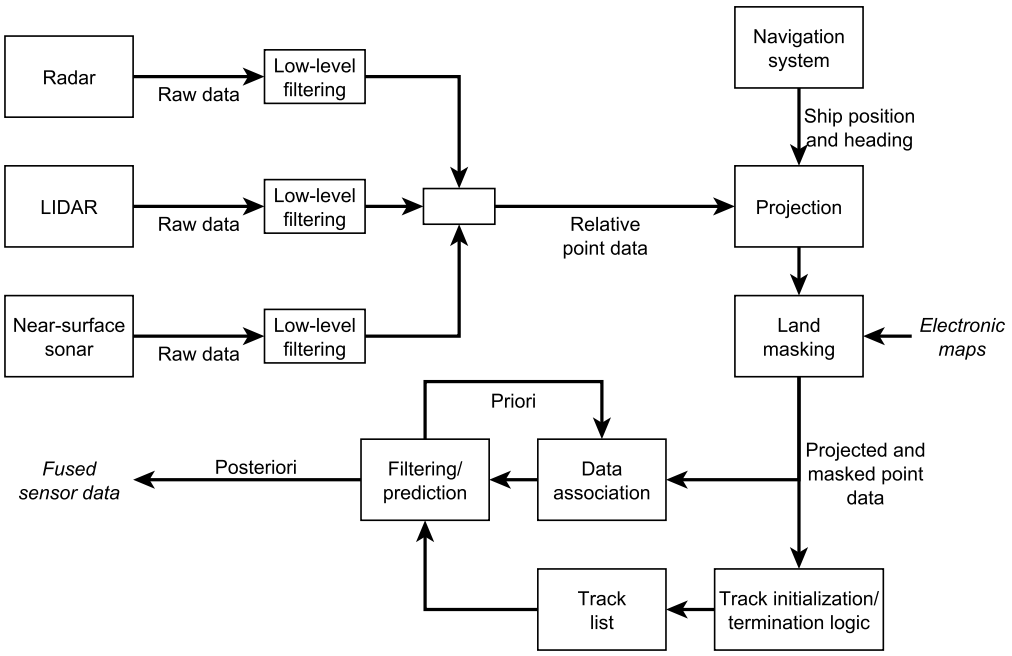


Figure 7.12: An instance of a sensor fusion pipeline concept, where the goal is to track targets using multiple sensors.

Multiple hypothesis tests (MHT) Measurement history is included, and multiple hypotheses of target movement are compared to give the best estimate.

Multiple motion models Multiple models for motion are compared. Useful if the targets have highly varying degrees of dynamics (such as a speed boat versus an ocean liner).

An overview of a possible sensor fusion pipeline is seen in Figure 7.12. This set-up is similar to the one seen in [120]. Point data is projected on an earth-fixed reference frame before points over known land are removed through *land masking*. The remaining points are then used in both track initialization/termination logic, and in data association and filtering.

Sensor fusion is also treated briefly in [114], where a software pack developed by Daniel H. Wagner Associates [123] is used to combine measurements. This software uses multiple of the mentioned techniques to fuse sensor data into a tactical picture.

Chapter 8

Possible Solutions for Automatic Docking

Docking is a challenging task for the ferry crew due to the many different layouts of harbors, some of which make the docking process difficult. Ferries crashing into harbors occur disturbingly often. The following news articles are a small selection of accidents: [124, 125, 126, 127, 128]. These incidents may be attributed to demanding harbor layouts, and the wind and current conditions.

The skills of operating a ferry during docking lies entirely within the ship crew. Considering what a demanding task this is implies that much can be gained in terms of safety from developing automatic docking solutions.

A description of the docking process is to *move the ship with low speed from pose A in the proximity of the harbor, to pose B lying right next to it, while simultaneously avoiding all static and dynamic obstacles.*

8.1 Current state of docking and mooring

During the final stages of transit, the ship's crew aligns the ferry so that the harbor is reachable with the least amount of maneuvering. The captain takes manual control of the thrusters, approaching the harbor slowly while keeping close attention to all parts of the ship and obstacles. After making a controlled low-speed collision with the harbor, some thrust is applied towards it, such that the ferry is safely connected during loading and offloading [45]. This reduces the need for personnel, as well as the load time, and is common with small ferries and short stops, where traditional mooring is not needed. For a battery-powered ferry, this solution may be infeasible, as it requires some power to keep the ship steady. Additionally, depending on the charging method and schedule, the ferry may need

to be connected to a precise point on the dock during the loading and offloading period.

8.2 Algorithms used for automatic docking

As there are stationary objects very close to the ferry, automatic docking requires high accuracy of both position measurements and motion control algorithms. In addition, it is imperative that the system is aware of all obstacles, static or dynamic. Although few examples of automatic harbor docking have been found in the literature, significant research has been conducted on related fields, such as docking autonomous underwater vehicles (AUVs) to underwater stations, or to moving surface vehicles.

An excerpt of algorithms relevant to automatic docking is presented here. To classify the algorithms, each is described with a section below, and are listed in Table 8.1 (page 170) with the properties *vehicle type, scenario, method, and sensors*.

8.2.1 Recursive fuzzy procedure by Rae and Smith (1992)

A fuzzy docking procedure is treated in Rae and Smith's paper [129], which allows an AUV to dock with a stationary underwater vehicle. This procedure is defined recursively and is developed by investigating how human operators dock the vehicle through remote operation or in simulators. The recursive method is illustrated in Figure 8.1. The results from this method are gathered from a simulation, thus no navigation sensors were discussed.

Algorithm properties

Vehicle type: AUV; Scenario: Docking to moving underwater platform; Method: Recursive fuzzy path planning; Sensors: No sensors, simulation only.

8.2.2 Discretized fuzzy vector field maps by Teo et al. (2015)

Another scheme based on discretized fuzzy vector field maps for desired heading is presented in the works of Teo et al. [130]. This method is developed for use with AUVs docking to a station with fixed position and orientation under water. Figure 8.2 shows how an area around the docking station is discretized into cells, which contain a desired heading vector leading the AUV to the docking station.

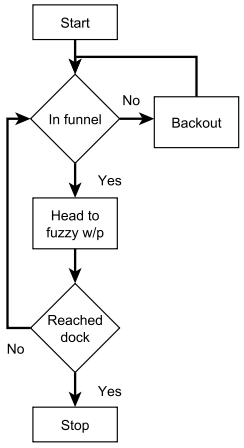


Figure 8.1: Flowchart of the fuzzy docking solution presented in [129]. The described "fuzzy waypoint" is determined by a heuristic map dependent on the vehicle's position relative to the docking base.

An iUSBL system is used as the main homing sensor, fused with an INS system for navigation.

Algorithm properties

Vehicle type: AUV; Scenario: Docking to fixed underwater platform; Method: Discretized fuzzy vector field maps; Sensors: iUSBL fused with INS.

8.2.3 A* path planning using LIDAR and proximity sensors by Spange (2016)

Object detection and path planning using LIDAR, ultrasound equipment, and the A* path planning algorithm has been attempted in Spange's thesis [122]. Docking experiments were done on a test ship mimicking a leisure boat in ideal environments, with varying results. Several actuator configuration modes were used, varying between redundant, fully actuated and underactuated. The fully actuated and redundant modes were satisfactory, but the author had trouble getting the underactuated modes to work. There were also some issues where the ship stopped too early or bumped into obstacles.

Algorithm properties

Vehicle type: Both fully and underactuated surface ship; Scenario: Docking to harbor; Method: Object detection with LIDAR and proximity sensors, with A*

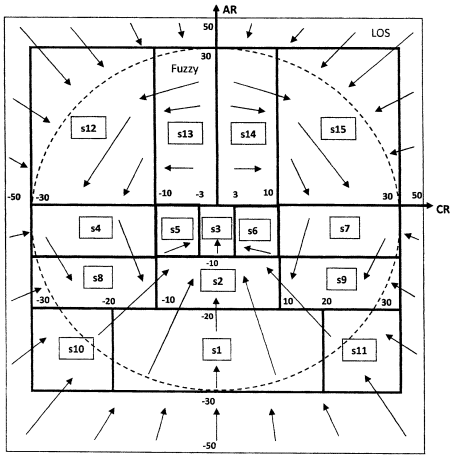


Figure 8.2: Discretized heading vector field, as presented in [130]. © 2015 IEEE.

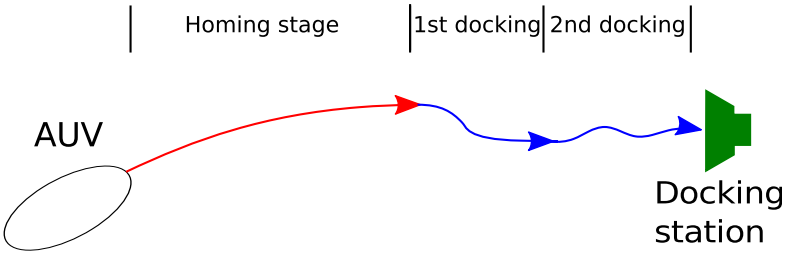


Figure 8.3: Illustration of docking stages presented in [131].

path planning; Sensors: LIDAR, ultrasonic rangefinder.

8.2.4 Repeated path planning and vision based docking by Hong et al. (2003)

A combination of repeated path planning and vision based docking is presented in the works of Hong et al. [131]. When the AUV is within a set distance (10 m in the paper) of a docking station, the first docking stage is entered. Here a docking path is generated and updated at set intervals, from the AUV to a predicted position of the docking station (which is moving slightly). When a stable position prediction is met, the second docking stage is entered. In this stage, the AUV uses real time vision tracking of the docking station to approach it safely. Figure 8.3 illustrates the docking stages.

Algorithm properties

Vehicle type: AUV; Scenario: Docking to moving underwater dock; Method: Repeated path planning and vision tracking; Sensors: LBL, USBL and camera.

8.2.5 Path following with PID control and DGPS by Andersen (1998)

Docking of a fully actuated surface ship to a harbor is described in Andersen's thesis [132]. A predefined path is followed using a model-based PID controller. The ship is equipped with a DGPS receiver for precise position measurement. In addition a method for docking the ship to a floating dock is developed, using a transponder for positioning.

Algorithm properties

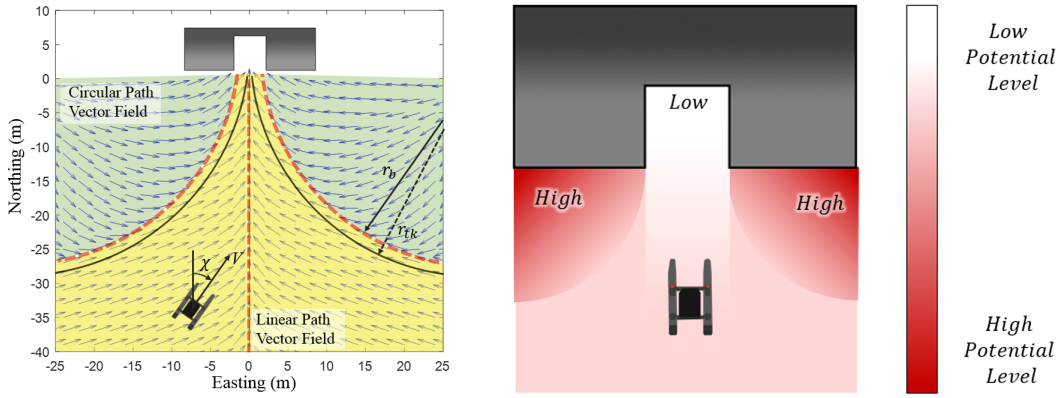
Vehicle type: Fully actuated surface ship; Scenario: Docking to fixed and moving surface harbor; Method: Predefined path following and PID control; Sensors: DGPS.

8.2.6 Docking to moving ship using target tracking by Loberg (2010)

Docking of AUVs to unmanned surface vehicles (USVs) and USVs to manned ships is treated in Loberg's thesis [133]. The scenarios presented in [133] are without static or dynamic obstacles, and with moving targets, which limits their usefulness for ferries docking to a harbor. The author employs a two-stage docking procedure for AUV to USV; a homing stage, and a more precise docking stage. The homing stage involves moving the USV in the proximity of the AUV, using a pure pursuit guidance scheme. The docking stage starts by assigning a target point for the USV to track and continues by letting the USV move in a straight line, while the AUV tracks a new target point described relative to the USV. A conference article on this algorithm is published by Breivik and Loberg in [134].

Algorithm properties

Vehicle type: Underactuated USV and AUV; Scenario: Docking to moving surface ship; Method: Target tracking; Sensors: No sensors, simulation only.



(a) (b)

Figure 8.4: Illustration of docking methods with velocity vector fields in (a) and artificial potential fields in (b). From [135] and [136], respectively. © 2016 International Society of Offshore and Polar Engineers, and Korean Society of Ocean Engineers.

8.2.7 Docking to harbor using vector fields and artificial potential fields by Woo and Kim (2016)

Recent research by Woo and Kim introduces two methods of performing docking of an underactuated ship to a static harbor. They present a vector field method in [135], where the desired velocity vector is determined by the ship's position. The vector field is composed by fields that give circular and linear paths. See Figure 8.4a for an illustration.

A similar approach where the gradients of repulsive and attractive potential fields are used to determine the desired heading. Gaussian and sigmoid functions are used to compose the potential field. The resulting field is illustrated in Figure 8.4b.

Algorithm properties

Vehicle type: Underactuated USV; Scenario: Docking to static harbor; Method: Vector field and artificial potential field; Sensors: No sensors, simulation only.

8.2.8 Algorithm deficiencies

The algorithms mentioned thus far are designed to work in a static environment with no dynamic obstacles. An algorithm used for docking a ferry in a harbor will need to consider these issues. Development and implementation of a suitable

docking algorithm must therefore be completed to achieve the goal of autonomous ferries.

Table 8.1: Comparison of selected docking algorithms.

Section	Vehicle type	Scenario	Method	Sensors
8.2.1	AUV	Docking to moving underwater platform	Recursive fuzzy path planning	No sensors, simulation only
8.2.2	AUV	Docking to fixed underwater platform	Discretized fuzzy vector field maps	iUSBL fused with INS
8.2.3	USV	Docking to harbor	Object detection with LIDAR and proximity sensors, with A* path planning	LIDAR, ultrasonic rangefinder
8.2.4	AUV	Docking to moving underwater dock	Repeated path planning and vision tracking	LBL, USBL and camera
8.2.5	USV and AUV	Docking to moving surface ship	Target tracking	No sensors, simulation only
8.2.7	USV	Docking to static harbor	Vector field and artificial potential field	No sensors, simulation only

Chapter 9

Automation Potential in Ferry Operations

Today's ferries are mostly dependent on human control for holding course, speed and docking. In the case of a takeover, crossing, or a head-on situation, the ship's personnel makes sure COLREGS are followed. The docking and mooring processes are also carried out by the personnel. In addition, ticketing and guiding of vehicles is performed manually. As mentioned in the introduction, several universities and companies are focusing on autonomy in ferries and other shipping operations. Rolls-Royce with their auto-crossing system and Kongsberg's *Yara Birkeland* are two examples of that [5, 6, 137].

This chapter is a discussion on the challenges which stand between today's situation and autonomous ferries. A characterization of autonomous ferries is given, and some reflection on the challenges that need to be solved is provided. The definition of autonomous ferries may vary depending on the level of autonomy that is considered. In this thesis, the characterizations are divided into those that relate to *motion control*, and those that do not.

9.1 Motion control characteristics

Related to motion control, the following criteria are considered:

- Automatic docking and undocking
- Automatic path planning
- Automatic path following
- Automatic collision avoidance

Challenges associated with automatic docking are treated in Chapter 8. The algorithms treated there are far from ready to use directly in a docking operation on a normal quay. They are mostly used in static environments with no dynamic obstacles. Two possibilities are recognized for improving this situation. One is to "dumb down" the docking environment, by creating docking quays customized for autonomous operation. This could involve closing the area off for unwanted traffic, and creating a system which guides or moves the ferry to the quay. The other possibility is to further develop docking algorithms such that they are safe to use in a "normal" docking environment. Extensive development and testing is needed to ensure that the operation does not endanger people or material.

Path planning is treated in Chapter 5. Algorithms providing a feasible path are quite mature. Roadmap methods from robotics are able to quickly generate a path between two points while avoiding obstacles. Optimization-based path planning shows promising results. These methods are not as extensively used, but are able to provide paths that are optimal in regards to energy, time or distance, and inherently include system dynamics which gives feasible paths. The drawback of optimization-based methods is computational time, and the need for quite accurate mathematical models. Currently roadmap methods are orders of magnitude faster than optimization-based methods. Developments in pseudospectral optimization theory as well as faster computers may change that in the future.

Guidance algorithms for path following are also covered in Chapter 5. Systems implementing these types of algorithms are common and commercially available.

Chapter 6 covers the use of a PS optimal control algorithm for path planning. Feasible time-optimized and energy-optimized paths are generated for a nonlinear 3-DOF ship model. Additionally, by implementing a guidance controller, it is shown that the path-planning algorithm produces feasible paths. With few exceptions, the optimization algorithm produces a path in less than 30 seconds. This provides validation for the concept.

Collision avoidance has seen significant development in recent years. Autonomous ships and self-driving cars have been the main drivers behind this research. Chapter 7 considers local collision avoidance algorithms for ships. Some of those may be modified to include the COLREGS. This is necessary when the autonomous ship is operating in an open area, where the crews of other ships have expectations on the behavior of nearby vessels. To be able to avoid collisions it is necessary to detect obstacles. Sensors feeding information to collision avoidance algorithms must be able to detect large obstacles like tankers and other ferries, smaller obstacles like leisure boats, and even smaller obstacles like kayakers or swimmers. This is done by the means of combining complementary and redundant sensors in sensor fusion algorithms. Elkins et al. have demonstrated sensor fusion and collision avoidance for a military boat [114], but these kinds of systems

Figure 9.1: Picture of AutoPASS solution. From [139]. © 2015 Kjell Arne Steinsvik, Møre-Nytt.

are not widespread.

9.2 Other characteristics

Apart from motion control, the following criteria are considered:

- Automatic ticketing
- Automatic guiding of vehicles
- Automatic mooring and unmooring
- Automatic charging

Ferry tickets are mostly sold by operators on board the ferry. A few connections have automatic ticketing systems, where the vehicle is registered automatically through a ticketing system, and an invoice is sent to the vehicle owner. The automatic system is a part of AutoPASS[1], which many vehicles already have. This system is illustrated in Figure 9.1. From 2017, domestic ferry connections will start moving to the AutoPASS system [138]. The AutoPASS system has been tested on the Flakk–Rørvik connection and is a promising solution for use in autonomous ferries.

[1] AutoPASS is a system for automatic toll road registrations using an electronic tag. More information may be found at `http://www.autopass.no`.

Figure 9.2: Cavotec automated mooring system. © 2017 Cavotec.

Part of the ferry personnel's job is to guide vehicles into the ferry and signal them to drive out. This is in order to use the space efficiently and to balance the weight evenly. To the best of the author's knowledge, no system for doing this automatically exists.

Currently, mooring is done without ropes, but with thrusters applying force towards the quay. This may be infeasible for battery-powered ferries, as it requires some power to keep the ship steady. An alternative is to use an automatic mooring system. Automatic mooring systems have been developed by Cavotec [140] and Mampaey [141], which use vacuum to secure ships to the dock. A photo of Cavotec's system is found in Figure 9.2. These systems may allow for free heave movement, as well as some rolling and pitching. The Port of Helsinki has started using the system from Cavotec on its West Harbor [142]. Combined with automatic docking, this type of system has potential for being used as a fast and safe automatic mooring solution.

In addition to automated mooring systems, Cavotec produces an automatic plug-in system for charging of ships. Figure 9.3 shows a photo of the system, consisting of a plug that is lowered from a tower into a specially designed compartment on the ship. Wärtsilä and Cavotec are also cooperating to develop a combined mooring and wireless charging system based on inductive power [143].

Figure 9.3: Cavotec automated charging system. © 2017 Cavotec.

Chapter 10

Conclusion and Further Work

This thesis brings together many topics related to autonomous shipping, and more specifically autonomous ferries. Technical and formal challenges related to autonomy in shipping are presented, and relevant solutions are discussed. Numerous resources on the central topics of path planning, collision avoidance and automatic docking are provided.

An introduction to commercial and industrial motion control equipment is provided. This includes architecture of autopilot and DP systems, as well as an overview of actuator configurations and maneuverability categories.

On the topic of path planning, a distinction has been made between roadmap methods and parametrized path generation. Among roadmap methods, both combinatorial methods and sampling-based methods have been looked into. These methods usually lead to the final refined paths after using a number of heuristics. Of parametrized path-generation methods, algorithms based on optimal control such as the semi-Lagrangian method and the pseudospectral method have been explored.

Further research on path planning using the pseudospectral method is performed. The optimal-control software DIDO is used to generate paths using a nonlinear 3-DOF ship model. The DIDO setup is able to provide a feasible trajectory for an energy-based cost function in 30 seconds. A guidance algorithm for curved paths is used to follow the generated path in simulations with unknown disturbances. This shows that the generated paths are feasible from a practical point of view.

A review of local collision-avoidance algorithms is given. Some of these are COLREGS-compliant, which is a necessity for autonomous ferries operating in trafficked areas. In addition to collision-avoidance algorithms, a range of sensors used for obstacle detection is presented, along with accuracy and price estimates for these. A brief introduction to tracking and sensor fusion is also given.

Several algorithms used for automatic docking of ships and underwater vehicles are reviewed. Common to these algorithms are that they have not been tested with moving obstacles. Further research and development of automatic docking algorithms for surface vessels must be completed.

Semi-autonomous ferries where transit is performed by automation are currently being tested and developed, e.g. by Rolls-Royce [6]. A higher level of autonomy, where collision avoidance and docking, is further ahead in time. Substantial research and development on sensors, obstacle detection, collision avoidance algorithms and docking algorithms is necessary to reach widespread full autonomy.

Emission-free propulsion for ferries is underway in the Norwegian fjords. Several new ships are built with batteries as the sole source of energy. Limitations on battery capacity constrain the number of ferry connections such ships may operate. Prioritizing energy efficiency in all aspects of motion control allows emission-free ferries to operate more connections. This is an area of focus for developers of autonomous and semi-autonomous ships, and improvements are seen in e.g. Rolls-Royce's auto-crossing system.

To achieve the goal of energy-efficient autonomous ferries, several topics require attention in further work:

- Compare the energy costs of paths generated by roadmap algorithms to paths generated by optimization-based algorithms.
- Investigate the combination of a global optimization-based path planning algorithm, e.g. PS, with a local COLREGS-compliant collision-avoidance algorithm.
- Combine an ocean-current estimator with the PS path planning algorithm in order to re-plan during transit.
- Implement the optimization-based path planning algorithm with guidance on a physical model ship to validate the concept and to compare cost estimates with real costs.
- Develop an automatic docking algorithm suitable for ferries. Such an algorithm should take moving obstacles into account.

Appendix A

Ship Model, Parameters and Scaling

The ship model used for simulation and control design in this thesis is acquired from the work of Sørensen et al. in [38]. Some changes are made to adapt their model to the problems reviewed in this thesis. The model of Sørensen et al. does not include relative velocities in the system equations. In addition, the $\mathbf{C}_{RB}$ matrix is changed to a representation independent of linear velocities. These changes are made before presenting the outcome in this chapter.

The model is in 3 DOF, and has the form

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \tag{A.1a}$$

$$\mathbf{M}\dot{\boldsymbol{\nu}}_r + \mathbf{C}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r + \mathbf{D}(\boldsymbol{\nu}_r)\boldsymbol{\nu}_r = \boldsymbol{\tau}, \tag{A.1b}$$

where $\mathbf{M} \in \mathbb{R}^{3\times3}$, $\mathbf{C}(\boldsymbol{\nu}_r) \in \mathbb{R}^{3\times3}$, $\mathbf{D}(\boldsymbol{\nu}_r) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\tau} \in \mathbb{R}^3$. The kinematic transformation matrix $\mathbf{R} \in \mathrm{SO}(3)$ is given by

$$\mathbf{R} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.2}$$

The inertia matrix is composed by $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$, where

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \quad \text{and} \tag{A.3}$$

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix}. \tag{A.4}$$

The Coriolis and centripetal matrix is composed by $\mathbf{C}(\boldsymbol{\nu}_r) = \mathbf{C}_{RB}(\boldsymbol{\nu}_r)+\mathbf{C}_A(\boldsymbol{\nu}_r)$, where

$$\mathbf{C}_{RB} = \begin{bmatrix} 0 & -mr & -mx_g r \\ mr & 0 & 0 \\ mx_g r & 0 & 0 \end{bmatrix} \text{ and} \tag{A.5}$$

$$\mathbf{C}_A(\boldsymbol{\nu}_r) = \begin{bmatrix} 0 & 0 & -c_{A,13}(\boldsymbol{\nu}_r) \\ 0 & 0 & c_{A,23}(\boldsymbol{\nu}_r) \\ c_{A,13}(\boldsymbol{\nu}_r) & -c_{A,23}(\boldsymbol{\nu}_r) & 0 \end{bmatrix}. \tag{A.6}$$

The added mass elements of $\mathbf{C}_A$ are

$$c_{A,13}(\boldsymbol{\nu}_r) = -Y_{\dot{v}} v_r - \frac{1}{2}(N_{\dot{v}} + Y_{\dot{r}})r \text{ and} \tag{A.7a}$$

$$c_{A,23}(\boldsymbol{\nu}_r) = -X_{\dot{u}} u_r. \tag{A.7b}$$

The damping matrix is composed by $\mathbf{D}(\boldsymbol{\nu}_r) = \mathbf{D}_L + \mathbf{D}_{NL}(\boldsymbol{\nu}_r)$, where

$$\mathbf{D}_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \text{ and} \tag{A.8}$$

$$\mathbf{D}_{NL}(\boldsymbol{\nu}_r) = \begin{bmatrix} d_{NL,11}(\boldsymbol{\nu}_r) & 0 & 0 \\ 0 & d_{NL,22}(\boldsymbol{\nu}_r) & d_{NL,23}(\boldsymbol{\nu}_r) \\ 0 & d_{NL,32}(\boldsymbol{\nu}_r) & d_{NL,33}(\boldsymbol{\nu}_r) \end{bmatrix}. \tag{A.9}$$

The elements of the nonlinear damping matrix are

$$d_{NL,11}(\boldsymbol{\nu}_r) = -X_{|u|u}|u_r| - X_{u^3} u_r^2, \tag{A.10a}$$

$$d_{NL,22}(\boldsymbol{\nu}_r) = -Y_{|v|v}|v_r| - Y_{|r|v}|r|, \tag{A.10b}$$

$$d_{NL,23}(\boldsymbol{\nu}_r) = -Y_{|v|r}|v_r| - Y_{|r|r}|r| - Y_{ur} u_r, \tag{A.10c}$$

$$d_{NL,32}(\boldsymbol{\nu}_r) = -N_{|v|v}|v_r| - N_{|r|v}|r| - N_{uv} u_r \text{ and} \tag{A.10d}$$

$$d_{NL,33}(\boldsymbol{\nu}_r) = -N_{|v|r}|v_r| - N_{|r|r}|r| - N_{ur} u_r, \tag{A.10e}$$

with

$$Y_{ur} = X_{\dot{u}} \tag{A.11a}$$

$$N_{uv} = -(Y_{\dot{v}} - X_{\dot{u}}) \text{ and} \tag{A.11b}$$

$$N_{ur} = Y_{\dot{r}}. \tag{A.11c}$$

All required parameters are found in Table A.1.

Table A.1: Ship model parameter values of CyberShip II. From [38].

Parameter	Value	Units	Parameter	Value	Units
L	1.255	m	Y_v	−0.88965	kg/s
m	23.8	kg	$Y_{\|v\|v}$	−36.47287	kg/m
x_g	0.046	m	N_v	0.03130	kg m/s
I_z	1.760	kg m^2	$N_{\|v\|v}$	3.95645	kg
$X_{\dot{u}}$	−2.0	kg	$Y_{\|r\|v}$	−0.805	kg
$Y_{\dot{v}}$	−10.0	kg	Y_r[1]	0.1052	kg m/s
$Y_{\dot{r}}$	0	kg m	$Y_{\|v\|r}$	−0.845	kg
$N_{\dot{v}}$	0	kg m	$Y_{\|r\|r}$	−3.450	kg m
$N_{\dot{r}}$	−1.0	kg m^2	$N_{\|v\|r}$	0.080	kg m
X_u	−0.72253	kg/s	$N_{\|r\|v}$	0.130	kg m
$X_{\|u\|u}$	−1.32742	kg/m	N_r	−1.900	kg m^2/s
X_{u^3}	−5.86643	kg s/m^2	$N_{\|r\|r}$	−0.750	kg m^2

A.1 Upscaling of the CyberShip II model using the bis system

The parameters in Table A.1 are from a 1:70 scale replica of a supply ship, with a length of 1.255 m and mass $m = 23.8$ kg. See Figure A.1. The *bis system* for parameter normalization by Norrbin [146] is used to scale the parameters up to a ship of size $L = 100$ m and $m = 2\,000\,000$ kg, which is a typical size for a ferry. More information on the bis system is found in [34].

Table A.2 shows the normalization variables used in the bis system. The expression $\mu = \frac{m}{\rho\nabla}$ is the density ratio, which is 1 for floating ships. This comes from $m = \rho\nabla$, which simplifies the entries in Table A.2, and gives the rightmost column.

Being equivalent for this use, the subscript $(\cdot)_r$ is dropped from the velocity variables in the transformations. The notation $(\cdot)_1$ is used for the parameters belonging to CyberShip II, and $(\cdot)_2$ is used to denote the parameters belonging to the ferry-sized ship. The normalization variables from Table A.2 are used to find

[1]The value of Y_r from [38] is −7.250. With this magnitude and negative sign, the damping effects from yaw rate has destabilizing effects on guidance. A clockwise turn (positive r) made to correct the course will cause negative sway forces, which again gives a counter-clockwise course change, thus causing positive feedback in the guidance loop. The values from [38] are collected from [144], where the parameters are generated by performing adaptive maneuvering experiments. The value for Y_r used here is taken from [40].

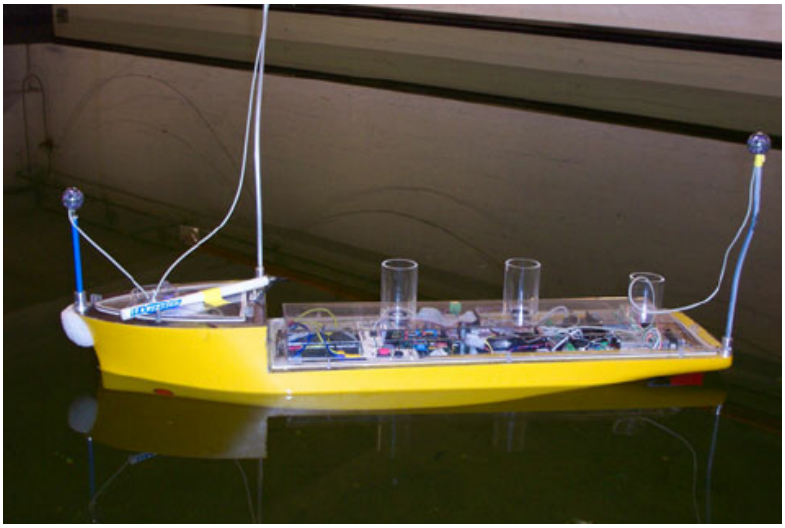


Figure A.1: CyberShip II. From [145]. © 2005 Elsevier Ltd.

the transformation from $\dot{\boldsymbol{\nu}}_1$ to $\dot{\boldsymbol{\nu}}_2$, $\boldsymbol{\nu}_1$ to $\boldsymbol{\nu}_2$, and from $\boldsymbol{\tau}_1$ to $\boldsymbol{\tau}_2$:

$$\dot{\boldsymbol{\nu}}_2 = \mathbf{T}_{\dot{\boldsymbol{\nu}}}\dot{\boldsymbol{\nu}}_1 \rightarrow \dot{\boldsymbol{\nu}}_1 = \mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1}\dot{\boldsymbol{\nu}}_2 \tag{A.12a}$$

$$\boldsymbol{\nu}_2 = \mathbf{T}_{\boldsymbol{\nu}}\boldsymbol{\nu}_1 \rightarrow \boldsymbol{\nu}_1 = \mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2 \tag{A.12b}$$

$$\boldsymbol{\tau}_2 = \mathbf{T}_{\boldsymbol{\tau}}\boldsymbol{\tau}_1 \rightarrow \boldsymbol{\tau}_1 = \mathbf{T}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\tau}_2 \tag{A.12c}$$

where

$$\mathbf{T}_{\dot{\boldsymbol{\nu}}} = \text{diag}\{1, 1, \lambda^{-1}\} \rightarrow \mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1} = \text{diag}\{1, 1, \lambda\} \tag{A.13a}$$

$$\mathbf{T}_{\boldsymbol{\nu}} = \text{diag}\{\sqrt{\lambda}, \sqrt{\lambda}, \lambda^{-\frac{1}{2}}\} \rightarrow \mathbf{T}_{\boldsymbol{\nu}}^{-1} = \text{diag}\{\lambda^{-\frac{1}{2}}, \lambda^{-\frac{1}{2}}, \sqrt{\lambda}\} \tag{A.13b}$$

$$\mathbf{T}_{\boldsymbol{\tau}} = \text{diag}\{\sigma, \sigma, \sigma\lambda\} \rightarrow \mathbf{T}_{\boldsymbol{\tau}}^{-1} = \text{diag}\{\sigma^{-1}, \sigma^{-1}, \sigma^{-1}\lambda^{-1}\} \tag{A.13c}$$

and

$$\lambda = \frac{L_2}{L_1} \tag{A.14a}$$

$$\sigma = \frac{m_2}{m_1}. \tag{A.14b}$$

The main assumptions are that the dimensionless accelerations, velocities and forces are equal for the model ship and a larger ship with the same characteristics. For a linear acceleration, velocity and force, $\dot{u}$, u and X, respectively, that gives

Table A.2: Bis system normalization variables. From [34].

Unit	Normalization variable	Simplified
Length	L	L
Mass	$\mu\rho\nabla$	m
Inertia moment	$\mu\rho\nabla L^2$	mL^2
Time	$\sqrt{\frac{L}{g}}$	$\sqrt{\frac{L}{g}}$
Reference area	$\mu\frac{2\nabla}{L}$	$\frac{2\nabla}{L}$
Position	L	L
Angle	1	1
Linear velocity	$\sqrt{Lg}$	$\sqrt{Lg}$
Angular velocity	$\sqrt{\frac{g}{L}}$	$\sqrt{\frac{g}{L}}$
Linear acceleration	g	g
Angular acceleration	$\frac{g}{L}$	$\frac{g}{L}$
Force	$\mu\rho g\nabla$	mg
Moment	$\mu\rho g\nabla L$	mgL

us the following relationships:

$$\dot{u}'' = \frac{1}{g}\dot{u}_1 = \frac{1}{g}\dot{u}_2 \rightarrow \dot{u}_1 = \dot{u}_2 \tag{A.15}$$

$$u'' = \frac{1}{L_1 g}u_1 = \frac{1}{L_2 g}u_2 \rightarrow u_1 = \frac{L_1}{L_2}u_2 \tag{A.16}$$

$$X'' = \frac{1}{m_1 g}X_1 = \frac{1}{m_2 g}X_2 \rightarrow X_1 = \frac{m_2}{m_1}X_2, \tag{A.17}$$

where g is the acceleration due to gravity, and the double prime symbol $()''$ is used to denote the dimensionless states. For angular acceleration, velocity and moment, $\dot{r}$, r, N, respectively, we have these relationships:

$$\dot{r}'' = \frac{L_1}{g}\dot{r}_1 = \frac{L_2}{g}\dot{r}_2 \rightarrow \dot{r}_1 = \frac{L_2}{L_1}\dot{r}_2 \tag{A.18}$$

$$r'' = \sqrt{\frac{L_1}{g}}r_1 = \sqrt{\frac{L_2}{g}}r_2 \rightarrow r_1 = \sqrt{\frac{L_2}{L_1}}r_2 \tag{A.19}$$

$$N'' = \frac{1}{m_1 g L_1}N_1 = \frac{1}{m_2 g L_2}N_2 \rightarrow N_1 = \frac{m_2 L_2}{m_1 L_1}N_2 \tag{A.20}$$

Using the notation $\mathbf{N}_i(\boldsymbol{\nu}) = \mathbf{C}_i(\boldsymbol{\nu}) + \mathbf{D}_i(\boldsymbol{\nu})$, we can use the following model to find $\mathbf{M}_2$, $\mathbf{C}_2$ and $\mathbf{D}_2$:

$$\mathbf{M}_1\dot{\boldsymbol{\nu}}_1 + \mathbf{N}(\boldsymbol{\nu}_1)\boldsymbol{\nu}_1 = \boldsymbol{\tau}_1 \tag{A.21}$$

Substituting the acceleration, velocity and torque vectors using (A.12), we get the following relationships:

$$\begin{aligned} \mathbf{M}_1\mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1}\dot{\boldsymbol{\nu}}_2 + \mathbf{N}_1(\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2)\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2 &= \mathbf{T}_{\boldsymbol{\tau}}^{-1}\boldsymbol{\tau}_2 \\ \underbrace{\mathbf{T}_{\boldsymbol{\tau}}\mathbf{M}_1\mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1}}_{\mathbf{M}_2}\dot{\boldsymbol{\nu}}_2 + \underbrace{\mathbf{T}_{\boldsymbol{\tau}}\mathbf{N}_1(\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2)\mathbf{T}_{\boldsymbol{\nu}}^{-1}}_{\mathbf{N}_2(\boldsymbol{\nu}_2)}\boldsymbol{\nu}_2 &= \boldsymbol{\tau}_2 \end{aligned} \tag{A.22}$$

This gives the new system matrices

$$\begin{aligned} \mathbf{M}_{RB,2} &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{M}_{RB,1}\mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1} \\ &= \begin{bmatrix} m_2 & 0 & 0 \\ 0 & m_2 & m_2\lambda x_{g,1} \\ 0 & m_2\lambda x_{g,1} & \lambda^2\sigma I_{z,1} \end{bmatrix}, \end{aligned} \tag{A.23}$$

$$\begin{aligned} \mathbf{C}_{RB,2}(\boldsymbol{\nu}_2) &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{C}_{RB,2}(\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2)\mathbf{T}_{\boldsymbol{\nu}}^{-1} \\ &= \begin{bmatrix} 0 & 0 & -m_2(\lambda x_{g,1}r_2 + v_2) \\ 0 & 0 & m_2u_2 \\ m_2(\lambda x_{g,1}r_2 + v_2) & -m_2u_2 & 0 \end{bmatrix}, \end{aligned} \tag{A.24}$$

$$\begin{aligned} \mathbf{M}_{A,2} &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{M}_{A,1}\mathbf{T}_{\dot{\boldsymbol{\nu}}}^{-1} \\ &= \begin{bmatrix} -\sigma X_{\dot{u},1} & 0 & 0 \\ 0 & -\sigma Y_{\dot{v},1} & -\lambda\sigma Y_{\dot{r},1} \\ 0 & -\lambda\sigma N_{\dot{v},1} & -\lambda^2\sigma N_{\dot{r},1} \end{bmatrix}, \end{aligned} \tag{A.25}$$

$$\begin{aligned} \mathbf{C}_{A,2}(\boldsymbol{\nu}_2) &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{C}_{A,2}(\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2)\mathbf{T}_{\boldsymbol{\nu}}^{-1} \\ &= \begin{bmatrix} 0 & 0 & -c_{A,13,2} \\ 0 & 0 & c_{A,23,2} \\ c_{A,13,2} & -c_{A,23,2} & 0 \end{bmatrix} \end{aligned} \tag{A.26}$$

with

$$c_{A,13,2}(\boldsymbol{\nu}_2) = -\sigma Y_{\dot{v},1}v_2 - \frac{1}{2}(\sigma\lambda N_{\dot{v},1} + \sigma\lambda Y_{\dot{r},1})r_2 \tag{A.27a}$$

$$c_{A,23,2}(\boldsymbol{\nu}_2) = -\sigma X_{\dot{u},1}u_2, \tag{A.27b}$$

$$\begin{aligned} \mathbf{D}_{L,2} &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{D}_{L,2}\mathbf{T}_{\boldsymbol{\nu}}^{-1} \\ &= \begin{bmatrix} -\frac{\sigma}{\sqrt{\lambda}}X_{u,1} & 0 & 0 \\ 0 & -\frac{\sigma}{\sqrt{\lambda}}Y_{v,1} & -\sigma\sqrt{\lambda}Y_{r,1} \\ 0 & -\sigma\sqrt{\lambda}N_{v,1} & -\sigma\lambda^{\frac{3}{2}}N_{r,1} \end{bmatrix}, \end{aligned} \tag{A.28}$$

and

$$\begin{aligned}\mathbf{D}_{NL,2}(\boldsymbol{\nu}_2) &= \mathbf{T}_{\boldsymbol{\tau}}\mathbf{D}_{NL,2}(\mathbf{T}_{\boldsymbol{\nu}}^{-1}\boldsymbol{\nu}_2)\mathbf{T}_{\boldsymbol{\nu}}^{-1} \\ &= \begin{bmatrix} d_{NL,11,2}(\boldsymbol{\nu}_2) & 0 & 0 \\ 0 & d_{NL,22,2}(\boldsymbol{\nu}_2) & d_{NL,23,2}(\boldsymbol{\nu}_2) \\ 0 & d_{NL,32,2}(\boldsymbol{\nu}_2) & d_{NL,33,2}(\boldsymbol{\nu}_2) \end{bmatrix}\end{aligned} \tag{A.29}$$

with

$$d_{NL,11,2}(\boldsymbol{\nu}_2) = -\frac{\sigma}{\lambda}X_{|u|u,1}|u_2| - \sigma\lambda^{-\frac{3}{2}}X_{u^3,1}u_2^2 \tag{A.30a}$$

$$d_{NL,22,2}(\boldsymbol{\nu}_2) = -\frac{\sigma}{\lambda}Y_{|v|v,1}|v_2| - \sigma Y_{|r|v,1}|r_2| \tag{A.30b}$$

$$d_{NL,23,2}(\boldsymbol{\nu}_2) = -\sigma Y_{|v|r,1}|v_2| - \sigma\lambda Y_{|r|r,1}|r_2| - \sigma Y_{ur,1}u_2 \tag{A.30c}$$

$$d_{NL,32,2}(\boldsymbol{\nu}_2) = -\sigma N_{|v|v,1}|v_2| - \sigma\lambda N_{|r|v,1}|r_2| - \sigma N_{uv,1}u_2 \tag{A.30d}$$

$$d_{NL,33,2}(\boldsymbol{\nu}_2) = -\sigma\lambda N_{|v|r,1}|v_2| - \sigma\lambda^2 N_{|r|r,1}|r_2| - \sigma\lambda N_{ur,1}u_2. \tag{A.30e}$$

A.2 Parameters for the upscaled CyberShip II model

The upscaled model may be put into the same form as the CyberShip II system, (A.3) through (A.11), by comparing the expressions from (A.23) through (A.30) with new parameters with subscript 2. Table A.3 shows the upscaled parameters, where the length and mass of the new ship is taken from *Korsfjord*[2], which ferries the Flakk–Rørvik connection in Trondheim, whose size is typical among ferries. In the rest of the thesis subscripts 1 and 2 will be dropped from both the parameters and the model variables.

[2] Lightship weight of *Korsfjord* is reported to be 1866 metric tons by the ship operators. However, including fluids and a typical load the weight is closer to 2000 tons, which is the number used in Table A.3.

Table A.3: Ship model parameter values for upscaled model. The units are the same as in Table A.1, and are here omitted for brevity.

Param.	Expression	Value	Param.	Expression	Value
L_2	L_2	120	$Y_{v,2}$	$\frac{\sigma}{\sqrt{\lambda}}Y_{v,1}$	-7.65×10^3
m_2	m_2	2.00×10^6	$Y_{\|v\|v,2}$	$\frac{\sigma}{\lambda}Y_{\|v\|v,1}$	-3.21×10^4
$x_{g,2}$	$\lambda x_{g,1}$	4.39	$N_{v,2}$	$\sigma\sqrt{\lambda}N_{v,1}$	2.57×10^4
$I_{z,2}$	$\lambda^2\sigma I_{z,1}$	1.35×10^9	$N_{\|v\|v,2}$	$\sigma N_{\|v\|v,1}$	3.32×10^5
$X_{\dot{u},2}$	$\sigma X_{\dot{u},1}$	-1.68×10^5	$Y_{\|r\|v,2}$	$\sigma Y_{\|r\|v,1}$	-6.76×10^4
$Y_{\dot{v},2}$	$\sigma Y_{\dot{v},1}$	-8.40×10^5	$Y_{r,2}$	$\sigma\sqrt{\lambda}Y_{r,1}$	8.64×10^4
$Y_{\dot{r},2}$	$\lambda\sigma Y_{\dot{r},1}$	0	$Y_{\|v\|r,2}$	$\sigma Y_{\|v\|r,1}$	-7.10×10^4
$N_{\dot{v},2}$	$\lambda\sigma N_{\dot{v},1}$	0	$Y_{\|r\|r,2}$	$\sigma\lambda Y_{\|r\|r,1}$	-2.77×10^7
$N_{\dot{r},2}$	$\lambda^2\sigma N_{\dot{r},1}$	-7.68×10^8	$N_{\|v\|r,2}$	$\sigma\lambda N_{\|v\|r,1}$	6.43×10^5
$X_{u,2}$	$\frac{\sigma}{\sqrt{\lambda}}X_{u,1}$	-6.21×10^3	$N_{\|r\|v,2}$	$\sigma N_{\|r\|v,1}$	1.09×10^4
$X_{\|u\|u,2}$	$\frac{\sigma}{\lambda}X_{\|u\|u,1}$	-1.67×10^3	$N_{r,2}$	$\sigma\lambda^{\frac{3}{2}}N_{r,1}$	-1.49×10^8
$X_{u^3,2}$	$\sigma\lambda^{-\frac{3}{2}}X_{u^3,1}$	-527	$N_{\|r\|r,2}$	$\sigma\lambda^2 N_{\|r\|r,1}$	-5.76×10^8

Bibliography

[1] C. Paris. "Norway Takes Lead in Race to Build Autonomous Cargo Ships." In: *The Wall Street Journal* (2017-07-22). URL: https://www.wsj.com/articles/norway-takes-lead-in-race-to-build-autonomous-cargo-ships-1500721202 (visited on 2017-08-20).

[2] E. Parry. "Amsterdam to get world's first fleet of autonomous boats." In: *Amsterdam Institute for Advanced Metropolitan Solutions* (2016-09-19). URL: http://www.ams-institute.org/news/roboat/ (visited on 2017-04-27).

[3] Zero. "Nyheter om utslippsfri og fossilfri ferge- og skipsfart." Norwegian. In: *Zero* (2017). URL: https://www.zero.no/nyheter-utslippsfri-fossilfri-ferge-skipsfart/.

[4] T. Stensvold. "Denne fergen er revolusjonerende. Men passasjerene merker det knapt. Bli med om bord i Ampere." Norwegian. In: *Teknisk Ukeblad* (2015-03-20). URL: https://www.tu.no/artikler/denne-fergen-er-revolusjonerende-men-passasjerene-merker-det-knapt/222522 (visited on 2017-04-20).

[5] T. Stensvold. "Første skritt mot autonome skip: Fjord1-ferge på autopilot." Norwegian. In: *Teknisk Ukeblad* (2016-10-19). URL: https://www.tu.no/artikler/forste-skritt-mot-autonome-skip-fjord1-ferge-pa-autopilot/359531 (visited on 2017-04-20).

[6] Rolls-Royce. "Rolls-Royce to supply first automatic crossing system to Norwegian ferry company Fjord1." In: *Rolls-Royce* (2016-10-08). URL: https://www.rolls-royce.com/media/press-releases/yr-2016/18-10-2016-rr-to-supply-first-automatic-crossing-system-to-norwegian-ferry-company-fjord1.aspx.

[7] K. Heggdal. "Trondheim kan bli først i verden på førerløse passasjerferjer." Norwegian. In: *Adresseavisen* (2016-06-09). URL: http://www.adressa.no/nyheter/trondheim/2016/06/09/Trondheim-kan-bli-f%C3%

B8rst-i-verden-p%C3%A5-f%C3%B8rerl%C3%B8se-passasjerferjer-12862564.ece.

[8] Vegdirektoratet. *Ferjestatistikk 2012*. Norwegian. Håndbok V620. 2014. URL: http://www.vegvesen.no/fag/trafikk/Trafikkdata/Ferjestatistikk.

[9] E. Farstad. *Transportytelser i Norge 1946–2015*. Norwegian. Tech. rep. 1544/2016. Transportøkonomisk institutt, 2016. ISRN: 978-82-480-1831-5. URL: https://www.toi.no/publikasjoner/transportytelser-i-norge-1946-2015-article34139-8.html.

[10] Vegdirektoratet. *Ferjeleier-1 Ferjeleiers landområder. Planlegging, prosjektering*. Norwegian. Håndbok V430. 1999.

[11] Miljødirektoratet. "Klimagassutslipp fra transport." Norwegian. In: *Miljøstatus* (2017-06-02). URL: http://www.miljostatus.no/tema/klima/norske-klimagassutslipp/utslipp-av-klimagasser-fra-transport/ (visited on 2017-08-20).

[12] T. Stensvold. "DNV GL har kartlagt: Disse skipstypene slipper ut mest." Norwegian. In: *Teknisk Ukeblad* (2015-06-15). URL: https://www.tu.no/artikler/dnv-gl-har-kartlagt-disse-skipstypene-slipper-ut-mest/223951.

[13] Oslo Economics, Inventura, and DNV GL. *Tiltaksanalyse for utvikling av ferjemarkedet på lang sikt*. Tiltaksutvikling og konsekvensanalyse. Norwegian. Tech. rep. 2016-22. Oslo Economics, 2016.

[14] Oslo Economics, Inventura, and DNV GL. *Premissanalyser – tiltaksanalyse for utvikling av ferjemarkedet på lang sikt*. Norwegian. Tech. rep. 2016-23. Oslo Economics, 2016.

[15] Skipsrevyen. *Skipsrevyen. Nordens ledende maritime magasin*. Norwegian. 2017. URL: http://www.skipsrevyen.no/ (visited on 2017-05-02).

[16] H. K. Dyrli. *Teknisk informasjon om ferjer i Norge*. Private communication. Email communication between Hans Kristian Dyrli in Multi Maritime AS and Glenn Bitar, regarding technical information about ferries in Norway. 2017-06-06.

[17] A. Lyngnes. *Informasjon om Horten-Moss-ferja*. Norwegian. Private communication. Email conversation between Aashild Lyngnes in Bastø Fosen and Glenn Bitar, regarding the Horten–Moss ferry connection. 2017-04-19.

[18] W. Langes. *Wagenfähren in Europa*. German. 2016. URL: http://www.fjordfaehren.de/ (visited on 2017-04-10).

[19] Bastø Fosen AS. *Bastø Fosen*. Norwegian. 2017. URL: http://basto-fosen.no/ (visited on 2017-04-13).

[20] Norled. *Norled*. Norwegian. 2017. URL: https://www.norled.no/ (visited on 2017-04-13).

[21] G. de Ruiter and S. de Ruiter. *Fjonesundet Kafe og Ferje*. Norwegian. 2017. URL: http://fjoneferga.no/ (visited on 2017-04-14).

[22] L. Haugstøyl and G. Fjone. *Fjone*. Norwegian. 2017. URL: http://www.fjone.no/ (visited on 2017-04-14).

[23] Samferdselsdepartementet. *Forskrift om transport med ferje*. Norwegian. Ed. by Stortinget. 2003-03-26. URL: https://lovdata.no/dokument/SF/forskrift/2003-03-26-403 (visited on 2017-04-10).

[24] Nærings- og fiskeridepartementet. *Forskrift om forebygging av sammenstøt på sjøen. Sjøveisreglene*. Norwegian. Ed. by Stortinget. 1975-12-01. URL: https://lovdata.no/dokument/SF/forskrift/1975-12-01-5/ (visited on 2017-04-10).

[25] Justis- og beredskapsdepartementet. *Lov om sjøfarten. Sjøloven*. Norwegian. Ed. by Stortinget. 1994-06-24. URL: https://lovdata.no/dokument/NL/lov/1994-06-24-39 (visited on 2017-04-10).

[26] Nærings- og fiskeridepartementet. *Forskrift om skipsførerens og rederiets plikter i tilfelle straffbare handlinger av alvorlig art begås om bord og melding om savnede personer. Forskrift om skipsførerens og rederiets plikter*. Norwegian. Ed. by Stortinget. 2005-03-01. URL: https://lovdata.no/dokument/SF/forskrift/2005-03-01-235 (visited on 2017-04-19).

[27] Samferdselsdepartementet. *Lov om havner og farvann. Havne- og farvannsloven*. Norwegian. Ed. by Stortinget. 2009-04-17. URL: https://lovdata.no/dokument/NL/lov/2009-04-17-19 (visited on 2017-04-10).

[28] Vegdirektoratet. *Ferjekai. Prosjektering*. Norwegian. Håndbok V431. 2017. URL: http://www.vegvesen.no/fag/teknologi/Bruer/Nyhetsarkiv+bru/ny-h%C3%A5ndbok-v431-ferjekai-prosjektering.

[29] IMO. *Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)*. International Maritime Organization. 1972. URL: http://www.imo.org/en/About/Conventions/ListOfConventions/Pages/COLREG.aspx (visited on 2017-03-20).

[30] H. Myre. "Collision Avoidance for Autonomous Surface Vehicles Using Velocity Obstacle and Set-Based Guidance." MA thesis. Norwegian University of Science and Technology, 2016. URL: http://hdl.handle.net/11250/2403593.

[31] T. Stenersen. "Guidance System for Autonomous Surface Vehicles." MA thesis. Norwegian University of Science and Technology, 2015. URL: http://hdl.handle.net/11250/2352498.

[32] Ø. A. G. Loe. "Collision Avoidance for Unmanned Surface Vehicles." MA thesis. Norwegian University of Science and Technology, 2008. URL: http://hdl.handle.net/11250/259696.

[33] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman. "A method for protocol-based collision avoidance between autonomous marine surface craft." In: *Journal of Field Robotics* 23.5 (2006), pp. 333–346. DOI: 10.1002/rob.20121.

[34] T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley-Blackwell, 2011-04. ISBN: 978-1-119-99149-6. DOI: 10.1002/9781119994138.

[35] O. M. Faltinsen. *Hydrodynamics of High-Speed Marine Vehicles*. Cambridge university press, 2005. ISBN: 9781139447935.

[36] D. Clarke. "A Two-Dimensional Strip Method for Surface Ship Hull Derivatives: Comparison of theory with Experiments on a Segmented Tanker Model." In: *Journal of Mechanical Engineering Science* 4.7 (1972), pp. 53–61. URL: http://journals.sagepub.com/toc/jmsa/14/7.

[37] D. Clarke, P. Gedling, and G. Hine. "Application of manoeuvring criteria in hull design using linear theory." In: *Naval Architect* (1983), pp. 45–68.

[38] M. E. N. Sørensen, M. Breivik, and B. H. Eriksen. "A Ship Heading and Speed Control Concept Inherently Satisfying Actuator Constraints." In: *1st IEEE Conference on Control Technology and Applications*. (2017-08-27). Kohala Coast, Hawaii, 2017.

[39] M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. Ed. by C. F. Shultz, M. Clendenny, D. Kellogg, and K. Santor. John Wiley and Sons Ltd, 2005. 496 pp. ISBN: 978-0-471-64990-8. URL: http://www.ebook.de/de/product/5124811/mark_spong_seth_hutchinson_mathukumalli_vidyasagar_robot_modeling_and_control.html.

[40] T. I. Fossen, M. Breivik, and R. Skjetne. "Line-of-sight path following of underactuated marine craft." In: *Proceedings of the 6th IFAC MCMC, Girona, Spain*. (2003-09). Girona, Spain, 2003, pp. 244–249. DOI: 10.1016/S1474-6670(17)37809-6.

[41] T. A. Johansen and T. I. Fossen. "Control allocation — A survey." In: *Automatica* 49.5 (2013-05), pp. 1087–1103. DOI: 10.1016/j.automatica.2013.01.035.

[42] P. Tøndel, T. A. Johansen, and A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions." In: *Automatica* 39.3 (2003-03), pp. 489–497. DOI: 10.1016/s0005-1098(02)00250-9.

[43] T. A. Johansen, T. I. Fossen, and P. Tøndel. "Efficient Optimal Constrained Control Allocation via Multiparametric Programming." In: *Journal of Guidance, Control, and Dynamics* 28.3 (2005-05), pp. 506–515. DOI: 10.2514/1.10780.

[44] T. A. Johansen, T. I. Fossen, and S. P. Berge. "Constrained Nonlinear Control Allocation With Singularity Avoidance Using Sequential Quadratic Programming." In: *IEEE Transactions on Control Systems Technology* 12.1 (2004-01), pp. 211–216. DOI: 10.1109/tcst.2003.821952.

[45] J. K. Olsen. *Information about ferry operations onboard MF* Korsfjord. Private communication. Gathered from conversations with Chief Mate Olsen on board MF *Korsfjord* during a round trip on the connection Flakk–Rørvik. 2017-04-26.

[46] B. K. Golding. *Industrial Systems for Guidance and Control of Marine Surface Vessels*. Project report. Department of Engineering Cybernetics. Norwegian University of Science and Technology, 2004.

[47] National Marine Electronics Association. *National Marine Electronics Association - NMEA*. 2016. URL: http://www.nmea.org/ (visited on 2017-04-12).

[48] IEC. *Maritime navigation and radiocommunication equipment and systems - Digital interfaces*. Standard IEC 61162. International Electrotechnical Commission, 2016.

[49] A. J. Sørensen, S. I. Sagatun, and T. I. Fossen. "Design of a dynamic positioning system using model-based control." In: *Control Engineering Practice* 4.3 (1996-03), pp. 359–368. DOI: 10.1016/0967-0661(96)00013-5.

[50] T. Holzhüter. "LQG approach for the high-precision track control of ships." In: *IEEE Proceedings - Control Theory and Applications* 144.2 (1997-03), pp. 121–127. DOI: 10.1049/ip-cta:19971032.

[51] C. G. Källström. "Autopilot and track-keeping algorithms for high-speed craft." In: *Control Engineering Practice* 8.2 (2000-02), pp. 185–190. DOI: 10.1016/s0967-0661(99)00167-7.

[52] A. Tsourdos, B. White, and M. Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles*. John Wiley & Sons, Inc., 2010-11-23. 190 pp. ISBN: 978-0-470-74129-0. URL: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470741295.html.

[53] M. Candeloro, A. M. Lekkas, and A. J. Sørensen. "A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels." In: *Control Engineering Practice* 61 (2017-04), pp. 41–54. DOI: 10.1016/j.conengprac.2017.01.007.

[54] J. Y. Yen. "An algorithm for finding shortest routes from all source nodes to a given destination in general networks." In: *Quarterly of Applied Mathematics* 27.4 (1970-01), pp. 526–530. DOI: 10.1090/qam/253822.

[55] B. Barsky and T. DeRose. "Geometric continuity of parametric curves: three equivalent characterizations." In: *IEEE Computer Graphics and Applications* 9.6 (1989-11), pp. 60–69. DOI: 10.1109/38.41470.

[56] A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen. "Continuous-Curvature Path Generation Using Fermat's Spiral." In: *Modeling, Identification and Control* 34.4 (2013), pp. 183–198. DOI: 10.4173/mic.2013.4.3.

[57] H. Moravec and A. Elfes. "High resolution maps from wide angle sonar." In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation.* (1985-03-25). St. Louis, MO, USA: Institute of Electrical and Electronics Engineers, 1985-03. DOI: 10.1109/robot.1985.1087316.

[58] T. Lozano-Pérez and M. A. Wesley. "An Algorithm for Planning Collision-free Paths Among Polyhedral Obstacles." In: *Commun. ACM* 22.10 (1979-10), pp. 560–570. ISSN: 0001-0782. DOI: 10.1145/359156.359164.

[59] G. Casalino, A. Turetta, and E. Simetti. "A three-layered architecture for real time path planning and obstacle avoidance for surveillance USVs operating in harbour fields." In: *Oceans 2009-Europe.* (2009-05-11). IEEE. Bremen, Germany, 2009, pp. 1–8. DOI: 10.1109/OCEANSE.2009.5278104.

[60] M. Klingensmith. *Overview of Motion Planning.* 2013-09-07. URL: http://www.gamasutra.com/blogs/MattKlingensmith/20130907/199787/Overview_of_Motion_Planning.php (visited on 2017-06-01).

[61] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces." In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580. DOI: 10.1109/70.508439.

[62] R. Geraerts and M. H. Overmars. "A Comparative Study of Probabilistic Roadmap Planners." In: *Springer Tracts in Advanced Robotics.* Springer Berlin Heidelberg, 2004, pp. 43–57. DOI: 10.1007/978-3-540-45058-0_4.

[63] S. M. LaValle. *Rapidly-Exploring Random Trees: A New Tool for Path Plannning.* 1998. URL: http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.1853.

[64] S. M. LaValle and J. Kuffner Jr. *Rapidly-Exploring Random Trees: Progress and Prospects*. 2000. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.1387.

[65] K. Hvamb. "Motion Planning Algorithms for Marine Vehicles." MA thesis. Norwegian University of Science and Technology, 2015. URL: http://hdl.handle.net/11250/2352514.

[66] J.-P. Laumond, N. Mansard, and J.-B. Lasserre. "Optimality in Robot Motion: Optimal versus Optimized Motion." In: *Communications of the ACM* 57.9 (2014-09), pp. 82–89. DOI: 10.1145/2629535.

[67] J.-D. Boissonnat, A. Cerezo, and J. Leblond. "Shortest paths of bounded curvature in the plane." In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*. (1992-05-12). Nice, France: IEEE Comput. Soc. Press, 1992-05, pp. 2315–2320. DOI: 10.1109/robot.1992.220117.

[68] L. C. Evans. *An Introduction to Mathematical Optimal Control Theory*. Lecture notes. Version 0.2. University of California, Berkeley, 2010. URL: https://math.berkeley.edu/~evans/ (visited on 2017-05-26).

[69] L. Cesari. *Optimization—Theory and Applications: Problems with Ordinary Differential Equations*. Stochastic Modelling and Applied Probability. Springer, 1983. ISBN: 0387906762.

[70] S. Sundar and Z. Shiller. "Optimal obstacle avoidance based on the Hamilton-Jacobi-Bellman equation." In: *IEEE Transactions on Robotics and Automation* 13.2 (1997-04), pp. 305–310. DOI: 10.1109/70.563653.

[71] A. L. Roald. "Path Planning for Vehicle Motion Control Using Numerical Optimization Methods." MA thesis. Norwegian University of Science and Technology, 2015. URL: http://hdl.handle.net/11250/2352517.

[72] R. Courant, E. Isaacson, and M. Rees. "On the solution of nonlinear hyperbolic differential equations by finite differences." In: *Communications on Pure and Applied Mathematics* 5.3 (1952-08), pp. 243–255. DOI: 10.1002/cpa.3160050303.

[73] J. C. Bowman, M. A. Yassaei, and A. Basu. "A fully Lagrangian advection scheme." In: *Journal of Scientific Computing* 64.1 (2015), pp. 151–177. DOI: 10.1007/s10915-014-9928-8.

[74] M. Falcone and R. Ferretti. *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*. Society for Industrial and Applied Mathematics, 2013-12. ISBN: 978-1-61197-304-4. DOI: 10.1137/1.9781611973051.

[75] Q. Gong, R. Lewis, and M. Ross. "Pseudospectral Motion Planning for Autonomous Vehicles." In: *Journal of Guidance, Control, and Dynamics* 32.3 (2009-05), pp. 1039–1045. DOI: 10.2514/1.39697.

[76] Elissar Global. *DIDO optimal control software*. 2017. URL: http://www.elissarglobal.com/academic/products/ (visited on 2017-06-12).

[77] A. M. Lekkas, A. L. Roald, and M. Breivik. "Online Path Planning for Surface Vehicles Exposed to Unknown Ocean Currents Using Pseudospectral Optimal Control." In: *IFAC-PapersOnLine* 49.23 (2016), pp. 1–7. DOI: 10.1016/j.ifacol.2016.10.313.

[78] Q. Gong, F. Fahroo, and I. M. Ross. "Spectral Algorithm for Pseudospectral Methods in Optimal Control." In: *Journal of Guidance, Control, and Dynamics* 31.3 (2008-05), pp. 460–471. DOI: 10.2514/1.32908.

[79] D. Garg, M. A. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington. "An overview of three pseudospectral methods for the numerical solution of optimal control problems." 2009.

[80] R. Bellman. *Dynamic Programming*. Dover Publications, 2013. ISBN: 978-0-486-42809-3.

[81] EMODnet. *European Marine Observation and Data Network*. European Union. 2017. URL: http://emodnet.eu/ (visited on 2017-06-05).

[82] L. R. Lewis and I. M. Ross. *A Pseudospectral Method for Real-Time Motion Planning and Obstacle Avoidance*. Research rep. ADA478686. Naval Postgraduate School, 2007-11-01. 23 pp. URL: http://www.dtic.mil/docs/citations/ADA478686.

[83] M. Breivik and T. I. Fossen. "Path following of straight lines and circles for marine surface vessels." In: *IFAC Proceedings Volumes* 37.10 (2004-07), pp. 65–70. DOI: 10.1016/s1474-6670(17)31709-3.

[84] M. Breivik. "Topics in Guided Motion Control of Marine Vehicles." PhD thesis. Norwegian University of Science and Technology, 2010. URL: http://hdl.handle.net/11250/259494.

[85] K. Y. Pettersen and E. Lefeber. "Way-point tracking control of ships." In: *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*. (2001-12-04). Orlando, FL, USA: IEEE, 2001. DOI: 10.1109/cdc.2001.980230.

[86] M. Breivik and T. I. Fossen. "Path following for marine surface vessels." In: *Oceans '04 MTS/IEEE Techno-Ocean '04*. (2004-11-09). Kobe, Japan: IEEE, 2004. DOI: 10.1109/oceans.2004.1406507.

[87] A. Micaelli and C. Samson. *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*. Research rep. RR-2097. Inria, 1993. URL: https://hal.inria.fr/inria-00074575/.

[88] C. Samson. "Path following and time-varying feedback stabilization of a wheeled mobile robot." In: *Proceedings of the ICARCV'92*. Valbonne, France, 1992.

[89] L. Lapierre, D. Soetanto, and A. Pascoal. "Nonlinear path following with applications to the control of autonomous underwater vehicles." In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475)*. (2003-12-09). Maui, HI, USA: IEEE, 2003. DOI: 10.1109/cdc.2003.1272781.

[90] I. M. Ross and F. Fahroo. "Legendre Pseudospectral Approximations of Optimal Control Problems." In: *New Trends in Nonlinear Dynamics and Control and their Applications*. Springer Berlin Heidelberg, 2004-05, pp. 327–342. DOI: 10.1007/978-3-540-45056-6_21.

[91] I. M. Ross. *A Primer on Pontryagin's Principle in Optimal Control*. Second. Collegiate Publishers, 2015. ISBN: 9780984357116. URL: http://www.upcitemdb.com/upc/9780984357116.

[92] T. Statheros, G. Howells, and K. M. Maier. "Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques." In: *Journal of Navigation* 61.01 (2007-12). DOI: 10.1017/s037346330700447x.

[93] C. Tam, R. Bucknall, and A. Greig. "Review of Collision Avoidance and Path Planning Methods for Ships in Close Range Encounters." In: *Journal of Navigation* 62.03 (2009-06), pp. 455–476. DOI: 10.1017/s0373463308005134.

[94] K. L. Woerner. "COLREGS-Compliant Autonomous Collision Avoidance Using Multi-Objective Optimization with Interval Programming." MA thesis. Massachusetts Institute of Technology, 2014. URL: https://woerner.mit.edu/colregs-compliant-autonomous-collision-avoidance-using-multi-objective-optimization-interval.

[95] M. R. Benjamin. *Autonomous COLREGS Modes and Velocity Functions*. Tech. rep. Massachusetts Institute of Technology, 2017-05-16. URL: http://hdl.handle.net/1721.1/109146.

[96] D. Fox, W. Burgard, and S. Thrun. "The dynamic window approach to collision avoidance." In: *IEEE Robotics & Automation Magazine* 4.1 (1997-03), pp. 23–33. DOI: 10.1109/100.580977.

[97] O. Brock and O. Khatib. "High-speed navigation using the global dynamic window approach." In: *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*. (1999-05-10). Detroit, MI, USA: IEEE, 1999. DOI: 10.1109/robot.1999.770002.

[98] B. H. Eriksen. "Horizontal Collision Avoidance for Autonomous Underwater Vehicles." MA thesis. Norwegian University of Science and Technology, 2015. URL: http://hdl.handle.net/11250/2352502.

[99] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger. "Safe Maritime Autonomous Navigation With COLREGS, Using Velocity Obstacles." In: *IEEE Journal of Oceanic Engineering* 39.1 (2014-01), pp. 110–119. ISSN: 0364-9059. DOI: 10.1109/JOE.2013.2254214.

[100] B. H. Eriksen. "Collision Avoidance for Autonomous Surface Vehicles." Unpublished technical note at the Norwegian University of Science and Technology. 2016.

[101] O. Khatib. "Real-time obstacle avoidance for manipulators and mobile robots." In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*. (1985-03-25). St. Louis, MO, USA, 1985. DOI: 10.1109/robot.1985.1087247.

[102] International Marine Contractors Association. *A Review of the Artemis Mark V Positioning System*. Tech. rep. M 174. International Marine Contractors Association (IMCA), 2016-06. URL: https://www.imca-int.com/publications/210/a-review-of-the-artemis-mark-v-positioning-system/.

[103] Kongsberg Maritime. *Relative position reference system, RADius*. 2017. URL: https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/57E47945903AB766C12570F3003061A8?OpenDocument.

[104] Kongsberg Maritime. *Laser-based position reference system. SpotTrack*. 2017. URL: https://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/DA020AE36CADEE97C1257BE400499329?OpenDocument.

[105] International Marine Contractors Association. *A Review of Marine Laser Positioning Systems*. Tech. rep. M 170. International Marine Contractors Association (IMCA), 2003-11. URL: https://www.imca-int.com/publications/195/a-review-of-marine-laser-positioning-systems-part-1-mk-iv-fanbeam-and-part-2-cyscan/.

[106] USA Department of Defense. *Global Positioning System Standard Positioning Service Performance Standard*. Tech. rep. USA Department of Defense, 2008-09. URL: http://www.gps.gov/technical/ps/.

[107] L. S. Monteiro, T. Moore, and C. Hill. "What is the accuracy of DGPS?" In: *Journal of Navigation* 58.2 (2005-05), pp. 207–225. DOI: 10.1017/s037346330500322x.

[108] Y. Feng and J. Wang. "GPS RTK performance characteristics and analysis." In: *Positioning* 1.13 (2008). URL: http://file.scirp.org/Html/376.html.

[109] E. Gakstatter. "Centimeter-Level RTK Accuracy More and More Available — for Less and Less." In: *GPS World* (2014-02-04). URL: http://gpsworld.com/centimeter-level-rtk-accuracy-more-and-more-available-for-less-and-less/ (visited on 2017-05-04).

[110] K. Vickery. "Acoustic positioning systems. A practical overview of current systems." In: *Proceedings of the 1998 Marine Technology Society (MTS) Dynamic Positioning Conference*. (1998-10-13). 1998-10.

[111] P. Milne. *Underwater Acoustic Positioning Systems*. Spon Press, 1983. ISBN: 978-0872010123.

[112] M. Watson, C. Loggins, and Y. Ochi. "A new high accuracy super-short baseline (SSBL) system." In: *Proceedings of 1998 International Symposium on Underwater Technology*. (1998-04-17). Tokyo, Japan: IEEE, 1998-04-17. DOI: 10.1109/ut.1998.670093.

[113] R. Halterman and M. Bruch. "Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection." In: *Unmanned Systems Technology XII*. (2010-04-05). Ed. by G. R. Gerhart, D. W. Gage, and C. M. Shoemaker. Orlando, FL, USA: SPIE, 2010-04. DOI: 10.1117/12.850611.

[114] L. Elkins, D. Sellers, and W. R. Monach. "The Autonomous Maritime Navigation (AMN) project: Field tests, autonomous and cooperative behaviors, data fusion, sensors, and vehicles." In: *Journal of Field Robotics* 27.6 (2010-09), pp. 790–818. DOI: 10.1002/rob.20367.

[115] T. Huntsberger, H. Aghazarian, A. Howard, and D. C. Trotz. "Stereo vision-based navigation for autonomous surface vessels." In: *Journal of Field Robotics* 28.1 (2010-12), pp. 3–18. DOI: 10.1002/rob.20380.

[116] C. Almeida, T. Franco, H. Ferreira, A. Martins, R. Santos, J. M. Almeida, J. Carvalho, and E. Silva. "Radar based collision detection developments on USV ROAZ II." In: *OCEANS 2009-EUROPE*. (2009-05-11). Bremen, Germany: IEEE, 2009-05. DOI: 10.1109/oceanse.2009.5278238.

[117] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. C. Antonello, and M. Miglietta. "Obstacle detection and classification fusing radar and vision." In: *2008 IEEE Intelligent Vehicles Symposium*. (2008-06-04). Eindhoven, Netherlands: IEEE, 2008-06. DOI: 10.1109/ivs.2008.4621304.

[118] T. Kato, Y. Ninomiya, and I. Masaki. "An obstacle detection method by fusion of radar and motion stereo." In: *IEEE Transactions on Intelligent Transportation Systems* 3.3 (2002-09), pp. 182–188. DOI: 10.1109/tits.2002.802932.

[119] H. K. Heidarsson and G. S. Sukhatme. "Obstacle detection and avoidance for an Autonomous Surface Vehicle using a profiling sonar." In: *2011 IEEE International Conference on Robotics and Automation*. (2011-05-09). Shanghai, China: IEEE, 2011-05. DOI: 10.1109/icra.2011.5980509.

[120] E. F. Wilthil, A. L. Flåten, and E. F. Brekke. "A Target Tracking System for ASV Collision Avoidance based on the PDAF." Unpublished preprint of an article at the Norwegian University of Science and Technology. 2017.

[121] A. Harati-Mokhtari, A. Wall, P. Brooks, and J. Wang. "Automatic Identification System (AIS): Data Reliability and Human Error Implications." In: *Journal of Navigation* 60.03 (2007-08), p. 373. DOI: 10.1017/s0373463307004298.

[122] J. Spange. "Autonomous Docking for Marine Vessels Using a Lidar and Proximity Sensors." MA thesis. Norwegian University of Science and Technology, 2016. URL: http://hdl.handle.net/11250/2440566.

[123] Daniel H. Wagner Associates. *Data Fusion*. 2013. URL: http://www.wagner.com/category/data-fusion/ (visited on 2017-05-09).

[124] C. Sanden, S. Strøm, and S. Rydland. "– Folk skrek og det var ganske fælt." Norwegian. In: *Norsk Rikskringkasting* (2013-07-25). URL: https://www.nrk.no/hordaland/ferge-dundret-i-kaien-pa-utne-1.11150719 (visited on 2017-05-02).

[125] G. Forland, S. Steinum, B. H. Johansen, and O.-C. Olsen. "Her krasjer ferga inn i kaia for andre gang i dag." Norwegian. In: *Norsk Rikskringkasting* (2014-07-07). URL: https://www.nrk.no/nordland/flere-skadet-etter-fergekollisjon-1.11818838 (visited on 2017-05-02).

[126] S. Lysvold. "*Godfjord* krasjet for femte gang på ett år." Norwegian. In: *Norsk Rikskringkasting* (2015-09-21). URL: https://www.nrk.no/nordland/_godfjord_-krasjet-for-femte-gang-pa-ett-ar-1.12564332 (visited on 2017-05-02).

[127] J. Ødegård. "Ulykkesferge krasjet for sjette gang: – Folk er forbannet." Norwegian. In: *Norsk Rikskringkasting* (2016-08-08). URL: https://www.nrk.no/nordland/ulykkesferge-krasjet-for-sjette-gang_-_-folk-er-forbannet-1.13077659 (visited on 2017-05-02).

[128] L. Hesjedal. "Fergeleie ødelagt etter krasj." Norwegian. In: *Helgelands Blad* (2017-02-12). URL: https://www.hblad.no/nyheter/ferge/samferdsel/fergeleie-odelagt-etter-krasj/s/5-23-62878 (visited on 2017-05-02).

[129] G. Rae and S. Smith. "A Fuzzy Rule Based Docking Procedure For Autonomous Underwater Vehicles." In: *OCEANS '92 Mastering the Oceans Through Technology*. (1992-10-26). Newport, RI, USA: IEEE, 1992. DOI: 10.1109/oceans.1992.607638.

[130] K. Teo, B. Goh, and O. K. Chai. "Fuzzy Docking Guidance Using Augmented Navigation System on an AUV." In: *IEEE Journal of Oceanic Engineering* 40.2 (2015-04), pp. 349–361. DOI: 10.1109/joe.2014.2312593.

[131] Y.-H. Hong, J.-Y. Kim, J.-h. Oh, P.-M. Lee, B.-H. Jeon, and K.-H. Oh. "Development of the homing and docking algorithm for AUV." In: *The Thirteenth International Offshore and Polar Engineering Conference*. (2003-05-25). International Society of Offshore and Polar Engineers. Hawaii, USA, 2003. URL: https://www.onepetro.org/conference-paper/ISOPE-I-03-112.

[132] L. Andersen. "Følgeregulering av skip." Norwegian. MA thesis. Norwegian University of Science and Technology, 1998.

[133] J. E. Loberg. "Planar Docking Algorithms for Underactuated Marine Vehicles." MA thesis. Norwegian University of Science and Technology, 2010. URL: http://hdl.handle.net/11250/259957.

[134] M. Breivik and J.-E. Loberg. "A Virtual Target-Based Underway Docking Procedure for Unmanned Surface Vehicles." In: *IFAC Proceedings Volumes* 44.1 (2011-01), pp. 13630–13635. DOI: 10.3182/20110828-6-it-1002.02969.

[135] J. Woo and N. Kim. "Vector Field based Guidance Method for Docking of an Unmanned Surface Vehicle." In: *Proceedings of the Twelfth (2016) Pacific-Asia Offshore Mechanics Symposium*. (2016-10-07). Gold Coast, Australia, 2016-10, pp. 276–281. URL: https://www.onepetro.org/conference-paper/ISOPE-P-16-103.

[136] J. Woo and N. Kim. "Design of Guidance Law for Docking of Unmanned Surface Vehicle." Korean. In: *Journal of Ocean Engineering and Technology* 30 (3 2016), pp. 208–213. DOI: 10.5574/KSOE.2016.30.3.208.

[137] Kongsberg. *YARA and KONGSBERG enter into partnership to build world's first autonomous and zero emissions ship*. 2017-05-09. URL: https://www.km.kongsberg.com/ks/web/nokbg0238.nsf/AllWeb/98A8C576AEFC85AFC125811A0037F6C4?OpenDocument (visited on 2017-05-10).

[138] K. Herskedal. "Autopass på ferjer fra 1. januar 2017." Norwegian. In: *Våre Veger* (2014-03-20). URL: https://www.vareveger.no/artikler/autopass-pa-ferjer-fra-1-januar-2017/196462 (visited on 2017-04-25).

[139] M. Myklebust. "Snart kan du betale med Autopass på ferja." Norwegian. In: *Møre-Nytt* (2015-12-04). URL: http://www.morenytt.no/nyheiter/2015/12/04/Snart-kan-du-betale-med-Autopass-p%C3%A5-ferja-11886550.ece (visited on 2017-06-14).

[140] Cavotec. *Automated Mooring Systems*. 2014. URL: http://www.cavotec.com/en/your-applications/ports-maritime/automated-mooring (visited on 2017-04-26).

[141] Mampaey Offshore Industries. *Intelligent Docklocking System*. 2016. URL: http://mampaey.com/the-intelligent-docklocking-system/ (visited on 2017-04-26).

[142] Port of Helsinki. *Automooring system speeds up ship mooring*. 2016. URL: http://www.portofhelsinki.fi/en/making-new/developing-west-harbour/automooring-system-speeds-ship-mooring (visited on 2017-04-26).

[143] Wärtsilä Corporation. "Wärtsilä and Cavotec to develop world's first marine wireless charging and mooring concept." In: *Wärtsilä global website* (2016-01-25). URL: https://www.wartsila.com/media/news/25-01-2016-wartsila-and-cavotec-to-develop-worlds-first-marine-wireless-charging-and-mooring-concept (visited on 2017-08-23).

[144] R. Skjetne, Ø. N. Smogeli, and T. I. Fossen. "A Nonlinear Ship Manoeuvering Model: Identification and adaptive control with experiments for a model ship." In: *Modeling, Identification and Control* 25.1 (2004), pp. 3–27. DOI: 10.4173/mic.2004.1.1.

[145] R. Skjetne, T. I. Fossen, and P. V. Kokotović. "Adaptive maneuvering, with experiments, for a model ship in a marine control laboratory." In: *Automatica* 41.2 (2005-02), pp. 289–298. DOI: 10.1016/j.automatica.2004.10.006.

[146] N. H. Norrbin. "Theory and Observations on the Use of a Mathematical Model for Ship Maneuvering in Deep and Confined Waters." In: *Proceedings of the 8th Symposium on Naval Hydrodynamics*. (1970-08-24). Pasadena, California, 1970, pp. 807–904. URL: `https://archive.org/details/hydrodynamicsino00symp`.