Nora Åsheim

Autonomous ship maneuvering with guaranteed safety

Master's thesis in Marine Technology
Supervisor: Roger Skjetne
Co-supervisor: Mathias Marley
June 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Norwegian University of
Science and Technology

Nora Åsheim

Autonomous ship maneuvering with guaranteed safety

Master's thesis in Marine Technology
Supervisor: Roger Skjetne
Co-supervisor: Mathias Marley
June 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Marine Technology

NTNU
Kunnskap for en bedre verden

NTNU Trondheim
Norwegian University of Science and Technology
Department of Marine Technology

MASTER OF TECHNOLOGY THESIS DEFINITION (30 SP)

Name of the candidate: Åsheim, Nora

Field of study: Marine cybernetics

Thesis title (Norwegian): Autonom skipsmanøvrering med garantert sikkerhet

Thesis title (English): Autonomous ship maneuvering with guaranteed safety

Background

Under auto-voyaging for autonomous ships, the ship is usually set to follow a nominal path constructed prior to the mission. The nominal path is free of known static obstacles. During the voyage, unknown or dynamic obstacles may be detected along the path, forcing the vessel to deviate from the nominal path. Once the obstacle is passed the vessel typically returns to the nominal path. However, in many situations re-planning of the desired path may be a more natural choice. This may even be advantageous in presence of multiple obstacles, where re-planning may eliminate the need for multiple evasive maneuvers.

The traditional maneuvering problem may be solved using a control Lyapunov function (CLF), with the vessel states (e.g., position, orientation, and velocity) and a scalar path variable $s \in \mathbb{R}$ as arguments. Safety (e.g., obstacle avoidance) may be ensured by combining the CLF with a control barrier function (CBF). In traditional CLF-CBF control designs, the CBF works in parallel with the CLF; that is, the CBF and the CLF act on the same control inputs. In such a setup the desired path remains unchanged during evasive maneuvers. Moreover, the CBF design must explicitly take the vessel dynamics into account, complicating the design process.

In this project, we will instead use a two-dimensional path variable, $s \in \mathbb{R}^2$, where s_1 is the along-path distance and s_2 is in the normal direction to the path. The desired vessel position $p_d(s(t))$ may then be interpreted as a virtual planar vessel. By employing CBFs to restrict the evolution of the path variable $s(t)$ it may be ensured that the path traced out by the virtual vessel is safe. This enables using CBF-designs for idealized kinematic vehicles, of which several solutions exist in literature.

With the CBF acting on the guidance level, obstacle avoidance capabilities are achieved without requiring explicit knowledge of the vessel dynamics. However, safety is only ensured if the vessel can follow the virtual vessel with sufficient accuracy.

Scope of Work

1. Perform a background and literature review to provide information and relevant references on:
   - CLF-based control design, maneuvering control design, and path-following.
   - Collision avoidance algorithms for autonomous marine vehicles.
   - Control barrier functions (CBFs) for safety assurance, including applications for collision avoidance of autonomous robots.

   Write a list with abbreviations and definitions of terms and symbols, relevant to the literature study and project report.
2. Establish a dynamic vessel model, for use in simulations. The vessel model shall contain uncertainty in the dynamics. This may for instance be a fully actuated 3-DOF maneuvering model with an unknown constant bias. Design a CLF-based controller that solves the traditional maneuvering problem for straight-line paths traced out by $q = p_d(s)$.
3. Design and implement CBF-based safety critical controllers for selected kinematic vessel models. This may for instance be particle models (with linear velocity or acceleration as control input), or unicycle models (with heading or heading rate as control input).

4. Modify the CLF-based controller from item 2 above to accommodate a 2-dimensional "path variable" $\xi \in \mathbb{R}^2$. Let the virtual point $q = p_d(\xi)$ emulate the kinematic vessel considered in item 3, with the CBF-based safety-critical controller giving obstacle avoidance capabilities, i.e., design the desired dynamics $\dot{\xi} = f_d(\xi, t)$ according to your virtual vessel designs(s). Explore the feasibility of the path traced out by $q(t) = p_d(\xi(t))$ for each virtual vessel model during obstacle avoidance maneuvers.

Specifications

Every weekend throughout the project period, the candidate shall send a status email to the supervisor and co-advisors, providing two brief bulleted lists: 1) work done recent week, and 2) work planned to be done next week.

The scope of work may prove to be larger than initially anticipated. By the approval from the supervisor, described topics may be deleted or reduced in extent without consequences with regard to grading.

The candidate shall present personal contribution to the resolution of problems within the scope of work. Theories and conclusions should be based on mathematical derivations and logic reasoning identifying the steps in the deduction.

The report shall be organized in a logical structure to give a clear exposition of background, problem/research statement, design/method, analysis, and results. The text should be brief and to the point, with a clear language. Rigorous mathematical deductions and illustrating figures are preferred over lengthy textual descriptions. The report shall have font size 11 pts., and it is not expected to be longer than 70 A4-pages, 100 B5-pages, from introduction to conclusion, unless otherwise agreed. It shall be written in English (preferably US) and contain the elements: Title page, abstract, preface (incl. description of help, resources, and internal and external factors that have affected the project process), acknowledgement, project definition, list of symbols and acronyms, table of contents, introduction (project background/motivation, objectives, scope and delimitations, and contributions), technical background and literature review, problem formulation, method, results and analysis, conclusions with recommendations for further work, references, and optional appendices. Figures, tables, and equations shall be numerated. The original contribution of the candidate and material taken from other sources shall be clearly identified. Work from other sources shall be properly acknowledged using quotations and a Harvard citation style (e.g. natbib Latex package). The work is expected to be conducted in an honest and ethical manner, without any sort of plagiarism and misconduct, which is taken very seriously by the university and will result in consequences. NTNU can use the results freely in research and teaching by proper referencing, unless otherwise agreed.

The thesis shall be submitted with an electronic copy to the main supervisor and department according to NTNU administrative procedures. The final revised version of this thesis definition shall be included after the title page. Computer code, pictures, videos, dataseries, etc., shall be included electronically with the report.

Start date: 15 January, 2021 **Due date:** As specified by the administration.

Supervisor: Roger Skjetne
Co-advisor(s): Mathias Marley

Signature(s): Roger Skjetne Digitally signed by Roger Skjetne
Date: 2021.05.31 12:06:22 +02'00'

Abstract

During a voyage, autonomous ships usually follow a nominal path constructed prior to the voyage. Known, static obstacles can be avoided in the planning process, but unknown or dynamic obstacles cannot. Hence, such obstacles may force the vessel to deviate from the nominal path, and typically return to the path after an evasive maneuver is performed. However, it may be more advantageous to re-plan the desired path, particularly in situations with multiple obstacles. This could eliminate the need for several evasive maneuvers. By implementing a two-dimensional path variable with an along-path distance and a path normal distance, the desired vessel position can be interpreted as a virtual planar vessel. By employing control barrier functions to restrict the evolution of the path variable, the path traced out by the virtual vessel can be ensured safe. Yet, safety is only ensured for the autonomous ship if it is able to follow the path. Therefore, the feasibility of such paths must be investigated. A comparison of various methods is of value to determine which provides a feasible path or the most feasible path.

A system consisting of a virtual vessel at guidance level, along with a line-of-sight guidance law was created. Three virtual vessel models from literature have been implemented. These were a first order particle model, a unicycle model, and a potential function. The virtual vessels' safety-critical controllers employed barrier functions, referred to as control barrier functions (CBFs), and quadratic programming to produce safe control inputs to the virtual vessel models. The simulations were performed in MATLAB, and thus the MATLAB function "quadprog" was used to perform quadratic programming optimization. Different CBFs from literature were implemented depending on the virtual vessel model. These included both hybrid and non-hybrid CBFs. The line-of-sight guidance produced a course angle reference for the autonomous ship based on the cross-track error to the position of the virtual vessel. The autonomous ship should perform trajectory tracking of the virtual vessel. A maneuvering model for an idealized ship was therefore implemented. The control system consisted of a backstepping controller. A control allocation was outside the scope of the thesis.

The results showed a varying degree of feasibility of the paths traced out. The most promising result was produced by the unicycle virtual vessel model, with a non-hybrid CBF and a modified obstacle in the safety-critical controller. This conclusion is largely based on the along- and cross-track errors observed, and the magnitude and smoothness of the control inputs required to perform trajectory tracking. The unicycle model with a hybrid CBF performed fairly well, but experienced some switching back and forth between CBFs during the evasive maneuver. The switching between CBFs caused undesirable transient response in the system. The particle model, which employed linear velocities as control inputs, resulted in a path with sharp turns, that the ship struggled to track to a certain degree. This resulted in sub-optimal behavior, that may not be realistic, although the ship appeared to be able to track the virtual vessel during the simulations. The potential function required unrealistically high forces and moment for the ship to perform trajectory tracking. Hence, the path traced out was deemed infeasible.

Several improvements are discussed. Examples are further tuning of individual parameters in the different methods, and the inclusion of a control allocation in the control system. In addition, an extension into path following, and the addition of several, and potentially dynamic, obstacles are of interest in future work. Furthermore, only current forces were added in the simulations. Including more environmental forces is an important extension of the environment to make it more realistic. However, such an extension would require the inclusion of an observer, which was outside the scope of this thesis.

Sammendrag

Autonome skip følger vanligvis en nominell bane under en reise. Denne banen planlegges gjerne før reisen begynner. Kjente, statiske hindringer kan unngås gjennom planleggingen av den nominelle banen, i motsetning til ukjente eller dynamiske hindringer. Slike hindringer kan derfor tvinge skipet til å divergere fra den nominelle banen, og typisk returnere til banen etter at hindringen er passert. Å returnere til banen er derimot ikke nødvendigvis optimalt. Å heller replanlegge banen kan hindre flere unnamanøvre hvis det for eksempel er mange hindringer i området. Ved å implementere en todimensjonal banevariabel med en langsgående banevariabel og en normal banevariabel, så kan den ønskede skipsposisjonen tolkes som et virtuelt skip. Ved å benytte kontroll barriere funksjoner for å begrense utviklingen til banevariabelen så kan banen det virtuelle skipet tenger ut gjøres trygg. For at banen skal være trygg for det autonome skipet så må banen i tillegg være mulig for skipet å følge. Derfor må banens gjennomførbarhet i forhold til skipets evne til å følge den undersøkes. En sammenligning av diverse metoder er verdifullt slik at man kan bestemme hvilke metoder som resulterer i gjennførbare baner, og eventuelt til hvilken grad.

Et system bestående av et virtuelt skip på guidance nivå sammen med en line-of-sight guidance lov ble designet. Tre virtuelle skipsmodeller fra literaturen ble også implementert. De var en partikkelmodell, en enhjulssykkelmodell og en potensialfunksjon. De virtuelle skipenes sikkerhetskritiske kontrollere benyttet seg av barriere funksjoner, referert til som kontroll barriere funksjoner (CBFer), samt kvadratisk programmering for å finne de trygge kontrollinputene. Simuleringene ble gjennomført i MATLAB, og derfor ble den innebygde funksjonen "quadprog" benyttet til å gjennomføre kvadratisk programmering. Forskjellige CBFer fra literaturen ble implementert, avhengig av den virtuelle skipsmodellen. Disse var både hybride of ikke-hybride CBFer. Line-of-sight guidance produserte en kursvinkel referanse for det autonome skipet basert på "cross-track" error til posisjonen til det virtuelle skipet. Det autonome skipet skulle gjennomføre banesporing. En manøvreringsmodell for et ideelt skip ble derfor implementert. Kontrolsystemet besto av en backsteppingkontroller. Kontrollallokering var utenfor omfanget av denne oppgaven.

Resultatene viste en variende grad av gjennomførbarhet for banene. Det mest lovende resultatet ble produsert av det virtuelle skipet modellert som en enhjulssykkelmodell med en ikke-hybrid CBF og en modifisert hindring i den sikkerthetskristiske kontrolleren. Denne konklusjonen er i stor grad basert på sporingsavikene i langsgående retning og normalt på banen, samt størrelse på og glatthet i kontrollinputene til det autonome skipet. Enhjulssykkelmodellen med en hybrid CBF utførte banesporing relativt bra, men med et par skifter frem og tilbake mellom CBFene under unnamanøveren som forårsaket uønskede transienter i systemet. Partikkelmodellen som benyttet seg av lineære hastigheter som kontrollinput, resulterte i en bane med skarpe svinger som skipet til en viss grad hadde vanskeligheter med å spore. Dette resulterte i sub-optimal oppførsel som ikke nødvendigvis er realistisk selv om skipet tilsynelatende klarte å spore det virtuelle skipet under simuleringene. Potensialfunksjonen krevde urealistisk høye krefter og moment for å gjennomføre banesporing.

Flere forslag til forbedringer diskuteres. Eksempler er å tune videre på individuelle parametere i de forskjellige metodene, og inkludere kontrollallokering i kontrollsystemet. I tillegg så forslås det å utvide fra banesporing til banefølging, samt legge til flere og dynamiske hindringer. Videre så består miljøkreftene kun av strøm. Flere miløkrefter ville derfor gjort miljøet mer realistisk. Dette krever en observer som var utenfor omfanget av dette arbeidet.

Preface

Five years ago I started my studies of Marine Technology at the Norwegian University of Science and Technology. Due to an interest in autonomous surface vessels in particular, I chose to specialize in Marine Cybernetics two and a half years ago. During my exchange semester in Lisbon I developed an interest in path planning through a robotics course at Instituto Superior Tcnico. This was the major motivation for the topic of this thesis.

The past year has been greatly affected by the COVID-19 pandemic. Fortunately, we were quite used to the situation by the time this master's thesis was commenced. Hence, the project definition could be adjusted to the expectations of what the semester would be like. Therefore, the focus was the implementations and simulations in MATLAB, not lab work. This was done to make sure that the thesis work was not interrupted by lock-downs during the semester.

Communication with my supervisor and co-advisor was almost solely done through Microsoft Teams. This is a good channel for digital communication, however it is less personal than face-to-face communication. This has at times challenging. However, my supervisor Prof. Roger Skjetne and my co-advisor Mathias Marley, has been readily available for meetings online, which I am grateful for.

Firstly, I would like to thank my supervisor, Prof. Roger Skjetne, for valuable guidance, insight, and literature throughout the semester. The designs implemented were developed with great help from my co-advisor, Mathias Marley, whom also provided me with useful literature, as well as serving as a sparring partner when needed throughout the design process. For that I owe him a huge thanks. My fellow master's student, Jon Magnus Moen, also served as a good sparring partner in the control design process, and so a thanks is directed to him as well.

From a young age my parents has been an inspiration to me. I would therefore like to thank them for inspiring me to choose an engineering degree at the Norwegian University of Science and Technology. A special thanks is directed to my mother for inspiring me choose Marine Technology, which I have enjoyed immensely.

Lastly, the past five years would not have been the same without my closest friends. I owe them a huge thank you for making these years the best years of my life so far.

The past semester has been one of the most challenging, yet rewarding semesters of my University career. With this thesis, it is with both great happiness, and some sadness, that I conclude my five years at the Norwegian University of Science and Technology.

Trondheim, 10th of June, 2021

N.Åsheim

Nora Åsheim

Contents

List of Figures

List of Tables

Nomenclature

Acronyms and symbols not included in the nomenclature will be defined in the text. Some symbols are similar, but distinction should be clear from context.

Acronyms

ASV	Autonomous surface vessel
CBF	Control barrier function
CLF	Control Lyapunov function
COLAV	Collision avoidance
COLREGS	International Regulations for Preventing Collisions at Sea
DOF	Degree of freedom
DP	Dynamic positioning
GNC	Guidance, Navigation, and Control
LOS	Line-of-sight
NE	North-East frame
OS	Own-ship
QP	Quadratic programming
SMC	Sliding mode control
UGAS	Uniformly globally asymptotically stable
UGES	Uniformly globally exponentially stable
VS	Virtual ship
WP	Way-point

Symbols

$\alpha(B)$	CBF class $\mathcal{K}$ function
α_1	Backstepping virtual control input
β	Crab angle
β_c	Flow direction of the current

β_s	Sideslip angle
χ	Course angle
χ_d	Desired course angle
Δt	Simulation time step in seconds
δ	Switching logic inequality constant
Δ_{LOS}	Line-of-sight look-ahead distance
$\emptyset$	Empty set
η	Position and orientation of the own-ship in the inertial frame
λ	Regularization parameter
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathcal{C}$	Safe set
$\mathcal{C}_u$	Unsafe set
$\mathcal{H}$	Hybrid dynamical system
$\mathcal{L}$	Set of logic variables
$\mathcal{S}^1$	Unit circle
μ	Synergy gap
ν	Linear and angular velocities of the own-ship in the body-fixed frame
ν_c	Current velocity in body-fixed frame
ν_r	Relative velocity in body-fixed frame
ω	Angular velocity
ω_{sat}	Angular velocity saturation limit
π_p	Angle between inertial frame and path tangential frame
ψ	Yaw angle
ψ_v	Set of critical orientations
ρ	Fresh water density
σ	Modified obstacle safety margin
τ	Vector of generalized control forces and moment
ξ	Two-dimensional path variable
ζ	Distance between obstacle and modified obstacle
$\{b\}$	Body-frame notation

$\{n\}$	Inertial-frame notation
$\{p\}$	Path tangential frame notation
A_{qp}	Quadratic programming inequality matrix
$B(\cdot)$	Barrier function
b_{qp}	Quadratic programming inequality vector
C	Flow set
$C(\nu)$	Coriolis and centripetal matrix
C_d	Drag coefficient
D	Jump set
$D(\nu)$	Damping matrix
$d(q)$	Distance from virtual ship to obstacle boundary
d_{lim}	Minimum distance from virtual ship to obstacle boundary before barrier function is activated
k	Particle model constant forcing VS back to nominal path
k_1	Critical orientations shifting angle
K_i	Controller gain matrix, $i = 1, 2, I$
L	Euclidean distance between subsequent way-points
l	Logic variable
l_r	Inertia radius in yaw
lb	Quadratic programming lower bound
M	Mass matrix
N	Unit normal vector
P	Function shifting the critical orientations
p	Position of the own-ship in the inertial frame
$P(q)$	Potential function
p_m	Modified obstacle position in the inertial frame
p_{obs}	Position of obstacle in the inertial frame
Q	Positive definite quadratic programming matrix
q	Position of the virtual vessel in the inertial frame
$R(\cdot)$	Rotation matrix
r_m	Modified safety radius around the obstacle

r_s	Safety radius around the obstacle
S	Rotation matrix corresponding to 90^o counterclockwise rotation
s	One-dimensional path variable
$S(\cdot)$	Skew-symmetric matrix
T	Unit tangent vector
t	Time in seconds
T_i	Controller time constant matrix, $i = 1, 2$
t_i	CBF time constant
U	Control input
U_B	Set of safe control inputs
U_c	Total current speed
u_d	Desired forward speed
U_r	Relative speed
U_s	Total speed
ub	Quadratic programming upper bound
V	Nominal control input
$V(\cdot)$	Control Lyapunov function
$v(\xi)$	Reference value for $\dot{\xi}$
x_e^p	Along-track error
y_e^p	Cross-track error
Z	Unit heading vector
z	State vector
z_i	Backstepping error variable, $i = 1, 2$
B	Breadth of own-ship
D	Draft of own-ship
L	Length of own-ship
m	Rigid body mass

Chapter 1

Introduction

1.1 Background and motivation

During auto-voyaging, autonomous ships usually follow a nominal path constructed prior to the mission. This path is free of known static obstacles, yet unknown or dynamic obstacles may be encountered during the voyage. These obstacles will force the vessel to deviate from the path. Once the obstacles are passed the vessel usually returns to the nominal path. However, in many cases, re-planning of the desired path could be advantageous, such as cases with multiple obstacles present. It could eliminate the need for several evasive maneuvers.

The traditional maneuvering problem can be solved using a control Lyapunov function (CLF), with the vessel states and a scalar path variable, $s \in \mathbb{R}$, as arguments. By combining a CLF with a control barrier function (CBF), safety can be ensured. Traditionally, the CBF works in parallel with the CLF in CLF-CBF control designs, where the CBF and CLF will act on the same control input. This leaves the desired path unchanged during evasive maneuvers. In addition, the CBF design must take the vessel dynamics into account, which complicates the design process. These limitations and complications motivates the topic of this thesis.

In the work, a two-dimensional path variable, $\xi \in \mathbb{R}^2$, will be used. ξ_1 is the along-path distance and ξ_2 is the normal direction to the path. The desired position of the vessel, $q = p_d(\xi(t))$, can be interpreted as a virtual planar vessel. By employing a CBF to limit the evolution of the path variable $\xi(t)$, the path traced out by the virtual vessel is ensured safe. With the CBF acting on the guidance level, obstacle avoidance capabilities are achieved without explicit knowledge of the vessel dynamics. However, safety is only ensured if the vessel can follow the virtual vessel with sufficient accuracy. Hence, several barrier functions and virtual vessel models will be explored, and the feasibility of each combination will be investigated.

1.2 Objectives and scope of work

The objective of the thesis is to design a system where a vessel tracks trajectories traced out by virtual vessels and explore the feasibility of the trajectories. During the voyage, an obstacle is encountered and barrier functions are employed to restrict the evolution of the path variable, ensuring that the virtual vessel trajectories are safe for the virtual vessel, at least. To achieve this objective, the following tasks are defined:

1. Perform a background and literature review on:
    - CLF-based control design, maneuvering control design, and path-following.

    - Collision avoidance algorithms for autonomous marine vehicles.
    - CBFs for safety assurance, including applications for collision avoidance of autonomous robots.

2. Establish a dynamic vessel model, for use in simulations. The vessel model shall contain uncertainty in the dynamics.

3. Design a CLF-based controller that solves the traditional maneuvering problem for straight-line paths traced out by $q = p_d(s)$

4. Design and implement CBF-based safety critical controllers for selected kinematic vessel models.

5. Modify the CLF-based controller from item 3 to accommodate a two-dimensional path variable, $\xi \in \mathbb{R}^2$. Let the virtual point, $q = p_d(\xi)$, emulate the kinematic vessel considered in item 4, with the safety-critical controller giving obstacle avoidance capabilities.

6. Explore the feasibility of the path traced out by $q(t) = p_d(\xi(t))$ for each virtual vessel model during obstacle avoidance maneuvers.

The scope of the thesis is limited by the assumptions and delimitation in Section 2.1.1.

1.3 Literature review

1.3.1 Control Lyapunov function based control design

Control Lyapunov functions can be employed to perform an integrator backstepping control design. According to Fossen (2021), the idea of integrator backstepping appeared almost simultaneously in Koditschek (1987), Sontag & Sussmann (1988), Tsinias (1989), and Byrnes & Isidori (1989). Developments and extensions of these are found in many later publications, such as ideas of adaptive and nonlinear backstepping designs which are described in Krstic et al. (1995). Fossen (2021) focuses on practical designs with implementation considerations for mechanical systems by exploring nonlinear properties of the systems. Fleischer (2020) uses designs explored in Fossen (2011) in the backstepping design in her thesis. This design is implemented with a one-dimensional path variable. Jensen (2020) uses a similar design with a two-dimensional path variable, citing Skjetne (2020*b*). The backstepping design in Skjetne (2020*b*) is a cascade-backstepping control design, which the control design in this thesis is based on. Skjetne (2005) presents constructive designs based on ISS (input-to-state stable) backstepping and adaptive backstepping to solve the maneuvering problem.

Another application of CLFs in control design is sliding mode control (SMC). Various methods of SMC with applications to marine crafts exists and appears in e.g. Yoerger & Slotine (1985), Healey & Lienard (1993), McGookin et al. (2000*a*) and McGookin et al. (2000*b*). Fossen (2021) discusses conventional integral SMC and the adaptive-gain super-twisting algorithm (STA). Conventional integral SMC can be used to design robust autopilots for heading, course, speed and depth control. In addition to this, STA has the advantage that it drives the sliding variable and its derivatives to zero in the presence of disturbances and uncertainties (Fossen 2021). Skjetne (2005) also presents a design based on SMC to solve the maneuvering problem, in addition to the backstepping designs mentioned above.

1.3.2 Maneuvering control design and path-following

Skjetne (2005) states that in many applications the most significant objective is to steer an object along a desired path. Control problems for these applications are usually divided into a geometric and a dynamic task. The geometric task aims to force the output of the system, usually the position, to reach and follow a desired output as a function of a path variable. The dynamic task is usually a specified speed assignment. In the tracking problem, these tasks are merged into one. In the path following problem the output should converge to and follow the desired path without any dynamic requirements along the path. Skjetne (2005) focuses on bridging the gap between these two concepts, resulting in the maneuvering control concept. In maneuvering, the problem is separated into two tasks, where path following is the first and most important task. The second task, which has a lower priority, is to satisfy a desired dynamic behaviour along the path. It is satisfied when tracing the path perfectly. However, the dynamic behaviour can be sacrificed if the vessel struggles to trace the path in order to achieve a more accurate path following (Skjetne 2005). Skjetne (2005) focuses on a one-dimensional path variable, while Skjetne (2020*b*) introduces a two-dimensional path variable in dynamic positioning (DP) conditions, i.e. assuming low speeds and a fully actuated vessel. A fully actuated vessel is able to control all its degrees of freedom (DOFs) (Fossen 2021). Jensen (2020) uses the design in Skjetne (2020*b*) in harbour maneuvering.

Furthermore, several motion control scenarios are found in Fossen (2021), including path tracking and path following. For the tracking problem, if there is no available information about the path in advance, target tracking is possible. The goal is then to track a moving or stationary object without information about the future motion of the object. Examples include line-of-sight (LOS) guidance and pure-pursuit guidance. With regards to path following, Fossen (2021) presents multiple design methods, such as linear design methods and LOS guidance laws for path following using course autopilots. Examples of LOS guidance laws for path following is look-ahead- and enclosure-based LOS steering, which are classifications of proportional LOS guidance laws made in Breivik & Fossen (2009). Proportional LOS guidance laws are described in Fossen (2021, Ch. 12.4), and finds the desired course angle using a proportional gain (see (12.79) in Fossen (2021, pp. 333)).

1.3.3 Collision avoidance algorithms for autonomous marine vehicles

Huang et al. (2020) defines collision avoidance (COLAV) as "a process in which one ship (manned or unmanned) departs from its planned trajectory to avoid a potential undesired contact a certain time in the future". Techniques to avoid collisions are called collision prevention techniques. Huang et al. (2020) offers an overview of collision prevention techniques based on motion prediction, conflict detection and conflict resolution. Strengths and weaknesses of COLAV methods are discussed, where limitations and new challenges are highlighted. Examples of such methods include rule-based methods and re-planning algorithms. A rule-based method could be incorporating the International Regulations for Preventing Collisions at Sea (COLREG) (IMO 1972) into the rule system guiding collision avoidance. Examples of re-planning algorithms are graph searching algorithms, where the goal is to find an optimal collision-free path. In this regard, Huang et al. (2020) compares research done on manned and unmanned ships, and how research in the domains can benefit from each other.

COLAV is also known as sense-and-avoid (Brekke et al. 2019). The COLAV system must perceive surrounding obstacles in order to perform evasive maneuvers. Brekke et al. (2019) provide a "bird-eye overview" of developments and experiments performed in the Autosea project focusing on algorithm design for target tracking and COLAV. According to Brekke et al. (2019), target tracking is an important tool to perform evasive maneuvers, as the unknown motion of obstacles must be estimated. Tracking methods perform data association to link detections made from subsequent images, which can be used to establish tracks and their kinematic attributes. To avoid obstacles, methods vary from path planning to reactive methods integrated into the control system. Path planning methods are described as global methods where an obstacle-free path is constructed prior to a mission. Reactive methods is a form of local methods where obstacles are detected and avoided during a mission. A form of reactive COLAV is the implementation of CBFs in the motion control system (Marley, Skjetne, Breivik & Fleischer 2020). Such a method with CBFs based on COLREGS is proposed in Thyri et al. (2020). The COLREGS are used to define a collision-free set. From this, a CBF is formulated and applied as an inequality constraint in a Guidance, Navigation and Control (GNC) system, with an optimization-based trajectory tracking or thrust allocation system.

1.3.4 Control barrier functions for safety assurance

Ames et al. (2019) provide an introduction to and overview of recent work on CBFs, and their use to verify and enforce safety properties in the context of safety-critical controllers. According to Ames et al. (2019), one of the main drivers of research on safety and CBFs is the recent interest in autonomous systems, particularly as many are expected to operate in unknown or unstructured environments. This makes enforcing safety properties difficult. The other is the recent introduction of CBFs suggesting that many control design techniques based on Lyapunov and control Lyapunov functions can be suitably modified to address safety considerations. Ames et al. (2019) discusses applications of CBFs to several robotic systems, such as the “stepping stone” problem, where a robot must walk safely on a series of stepping stones.

Marley, Skjetne, Breivik & Fleischer (2020) highlight the need for autonomous systems to be resilient, that is the control system must be stable, robust, and safe. Of these, the paper focuses on the concepts of safety and robustness, and proposes a CBF-based hybrid kinematic controller for obstacle avoidance. The hybrid controller ensures robustness, while the CBF ensures safety. The performance of the controller is illustrated by simulations using an underactuated ship. A non-hybrid CBF is unable to ensure safety of vehicles required to maintain a non-zero forward speed. This is due to vanishing control authority when the vehicle is pointed directly at the obstacle (Marley et al. 2021). Marley et al. (2021) therefore proposes synergistic CBFs for nonholonomic vehicles where the orientations with vanishing control authority are shifted, inducing a penalty for passing an obstacle in one direction or the other. The preferred direction is determined by a logic variable.

Basso et al. (2020) presents a unified framework for safety-critical control of ASVs for maneuvering, dynamic positioning (DP), and control allocation, with safety guarantees in the presence of unknown ocean currents. It utilizes CLF and CBF based quadratic programs (QP). Integral action is added to the CLFs to counteract the effect of the currents. Current estimates are constructed for robust CBF design, and analytic conditions under which the estimates guarantee safety are defined. The framework is verified using simulations with a double-ended passenger ferry.

1.4 Contributions

The main contribution of this thesis is a comparison of different trajectories in trajectory tracking during evasive maneuvers for an autonomous ship, focusing on the feasibility. The trajectories are traced out by a virtual vessel (VS), and different VS models are explored in combination with various barrier functions to activate the evasive maneuvers of the VS. That is, they are incorporated into the VS control design to create the safety-critical controllers for the VS. The feasibility of the trajectories are investigated through simulations in MATLAB with an OS modelled as an idealized maneuvering model. The OS controller uses a backstepping design method with a two-dimensional path variable. The thesis thus combines the exploration of various VS models, (control) barrier functions (C)BFs, and a two-dimensional path variable to give an indication of the feasibility of the different methods in trajectory tracking. The conclusions can be used to determine the focus of further research into collision avoidance methods using virtual vessels with safety-critical controllers based on (C)BFs, and two-dimensional path variables.

1.5 Outline

The report is structured as follows. **Chapter 2** contains the problem formulation, with the system description, assumptions and delimitation, and problem statement. **Chapter 3** presents a two dimensional path parametrization ξ. This is followed by **Chapter 4**, which presents the different barrier functions and control barrier functions used to activate the evasive maneuvers of the VS. The VS models, which trace out the trajectory $q(t) = p_d(\xi(t))$, are presented in **Chapter 5**. These include a particle model, a unicycle model, and a potential function. The OS maneuvering model is described in **Chapter 6**. The simulation code algorithm, simulation parameter values, and simulation results are presented in **Chapter 7**, with some discussions of the individual results. The final discussion and conclusions are made in **Chapter 8**, along with suggestions for further work.

Chapter 2

Problem formulation

2.1 System description

The International Maritime Organization identifies four degrees of ship autonomy (IMO n.d.). These are:

1. Ship with automated processes and decision support.
2. Remotely controlled ship with seafarers on board.
3. Remotely controlled ship without seafarers on board.
4. Fully autonomous ship.

Degree 4 is the subject of this thesis. IMO (n.d.) describes this as the case where "the operating system of the ship is able to make decisions and determine actions by itself". These vessels are known as ASVs (autonomous surface vessel).

Marine crafts can in general be described to move in six DOFs. This requires six independent coordinates to determine position and orientation of the craft (Fossen 2021). Fossen (2021) defines the position in the North-East-Down frame (NED), and the orientation by the Euler angles (roll, pitch and yaw). The Euler angles describes the orientation of the body-fixed reference frame with respect to the NED-frame. The body-frame motions are defined as surge, sway, heave, roll, pitch and yaw. An illustration of motions in 6-DOFs is included in Figure 2.1.

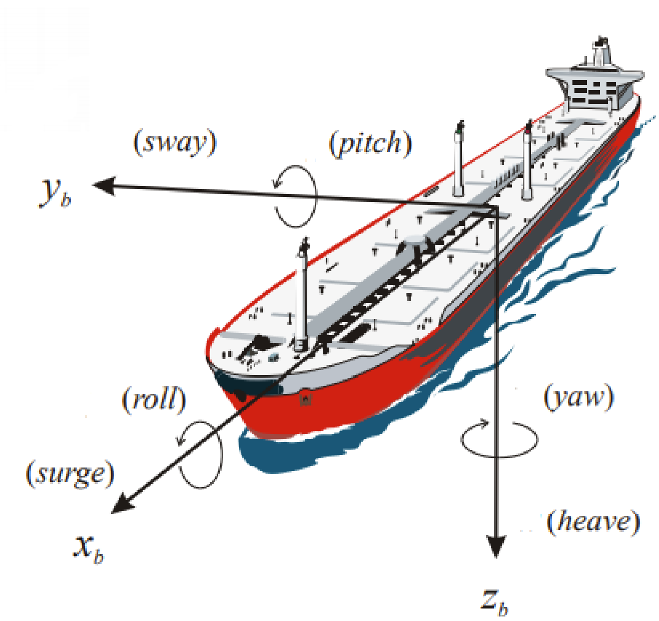


Figure 2.1: 6-DOF motions of a surface vessel. *Adapted from Fossen (2020).*

In the case of an ASV, it is assumed metacentrically stable (Fossen 2021). That means that the restoring forces will resist inclinations in heave, roll, and pitch away from the equilibrium points (Fossen 2021, pp. 68). These are at zero inclination. It it thus reasonable to neglect motions in heave, roll, and pitch. The result is a 3-DOF kinematic equation of motion,

$$\dot{\eta} = R(\psi)\nu, \tag{2.1}$$

where $\eta := [x \quad y \quad \psi]^\top$ is the ASV's position and orientation in the North-East-frame (NE). $\nu := [u \quad v \quad r]^\top$ denotes the ASV's linear and angular velocities in body-frame (SNAME 1950). $R(\psi)$ denotes the rotation matrix from body- to NE-frame.

$$R(\psi) := \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

For local navigation (navigation within a geographical area of about 10 km × 10 km), the NE-frame is assumed inertial (Fossen 2021). Figure 2.2 shows the rotation from the inertial NE-frame, denoted by $\{n\}$, to body-frame, denoted by $\{b\}$. This thesis concerns maneuvering at open sea in the transit phase of the voyage. When the ship is not in an evasive state, the speed is close to constant, and it is assumed that the speed in surge direction u is much larger than the speed in sway direction v. Consequently the total speed of the ASV is $U_s = \sqrt{u^2 + v^2} \approx u$. A constant, irrotational (no angular velocity) current, is applied to the ASV in the simulations. Figure 2.2 shows the effects of such a current. The angle between the true North and U_s is known as the course angle χ. The angle between the body-frame x-axis, x_b, and U_s is called the crab angle, β, (also known as the drift angle), while the angle between x_b and the total relative speed between the ASV and the current, U_r, is the sideslip angle β_s. The course angle is defined as $\chi := \psi + \beta$. The relative course angle is defined as $\chi_r := \psi + \beta_s$. If there is no current present, $\beta_s = \beta$ as $U_r = \sqrt{u_r^2 + v_r^2} = \sqrt{(u - u_c)^2 + (v - v_c)^2}$, as the current's surge and sway velocities $u_c = v_c = 0$. The ASV in this thesis is not exposed to any other environmental disturbances or forces other than current during the simulations.

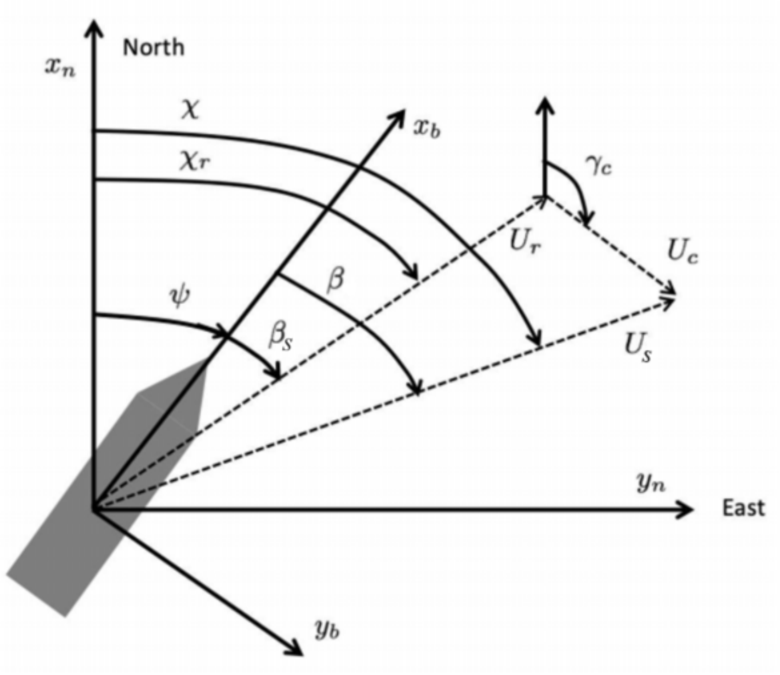


Figure 2.2: Ocean current triangle in the horizontal plane. *Adapted from Fossen (2021).*

The control objective in the simulations is to make the ASV track a desired trajectory. From now on the ASV will be referred to as the own-ship (OS). The term is from Huang et al. (2020), where it refers to the ship under control. The trajectory is traced out by a virtual vessel (VS), which is either defined by a particle model, a unicycle model, or a potential function. The particle model is a first order particle model with linear velocities as control input. Linear velocities are the control inputs in the potential function model as well, while the control inputs in the unicycle models are the heading rate and linear acceleration in surge. The control inputs are in the case of the particle model and unicycle model limited by a CBF, thus limiting the evolution of the path variable ξ. The design with a potential function utilizes a barrier function to limit the gradient of the potential function, and the control input is given by the negative valued gradient of the potential function. The vessel models of the VSs and the OS are addressed in Chapter 5 and 6 respectively, and therefore not discussed in detail in this section. The OS control system is based on a backstepping control design (Section 6.3). A visual overview of the total system in the form of a flow diagram is provided in Figure 2.3.

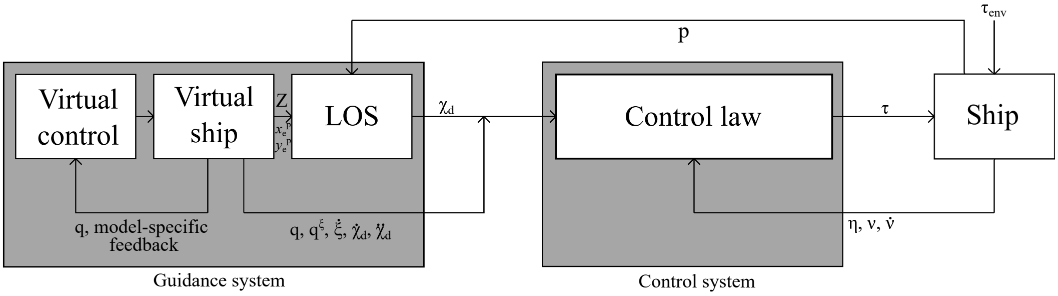


Figure 2.3: System flow diagram.

The guidance system consists of the VS and a LOS guidance law. The VS outputs the desired position q, the derivative of the desired position with respect to the path variable, q^{ξ} (ξ denotes the path variable), the time derivative of the path variable, $\dot{\xi}$, and the desired course rate $\dot{\chi}_d$ and its derivative. The outputs is used as input to the control system. In addition, it outputs the VS heading unit vector Z, which is used to compute the along-track and cross-track errors of the OS with respect to the VS position. The tracking errors in combination with Z is utlized in the LOS guidance law to find the desired course angle χ_d. The OS controller outputs the forces in surge and sway, and moment in yaw in the vector τ. τ is used in the vessel model to determine body-frame accelerations $\dot{\nu}$, which gives the body-frame velocities ν through Euler integration. By (2.1), ν gives the NE-frame position and heading vector, η. η, ν and $\dot{\nu}$ is fed back to the control system, while the OS position p is fed back to the LOS guidance law.

2.1.1 Assumptions and delimitation

The assumptions and delimitation of the system design and simulations are as follows:

- The OS is sufficiently metacentrically stable. This results in a 3-DOF, horizontal plane vessel model with dynamics in surge, sway, and yaw. Motions in heave, roll, and pitch are neglected.
- Local navigation results in an inertial NE-frame. Hence, the forces from the Earth's rotation are negligible (Fossen 2021).
- The ASV is assumed to be a rigid-body. This assumption removes the need to consider forces acting between individual mass elements. According to Fossen (2021), this assumption, along with the assumption of an inertial NE-frame, is necessary to derive the equations of motion.
- The vessel model is extracted from Marley (2020*b*). This model is assumed fore-aft and port-starboard symmetric. This ensures diagonal rigid-body and added mass matrices with midship as reference point. The hydrodynamic damping matrix is also assumed diagonal. These assumptions mean that the only coupling terms are due to Coriolis forces.
- The OS is assumed fully actuated in the horizontal plane. It can thus control all 3 DOFs independently. There are no limitations on the surge and sway forces, and yaw moment produced, as it is outside the scope of the thesis.
- The only environmental force, τ_{env}, is the current force. It is not measured and acts as an unknown bias.
- It is assumed that the obstacles encountered are static and that the OS is able to detect the obstacles.
- The positions of the OS and the obstacle are measured perfectly, i.e. there is no uncertainty or noise in the measurements.
- Inclusion of the COLREGS in the design of the barrier functions is outside the scope of the thesis.

2.2 Problem statement

The research question addressed in the thesis is: To what degree are the paths traced out by the virtual vessels feasible during obstacle avoidance maneuvers for the vessel model implemented, and how do they compare? The motivation for the thesis is explored in Section 1.1. In short, the implementation of a two-dimensional path variable allows re-planning of the VS path. In combination with barrier functions, such as CBFs, acting on a guidance level, the limitations and complications of a traditional CLF-CBF design are eliminated. However, safety is only ensured if the OS is able to track the trajectories traced out by the VS. Therefore, several designs are explored and compared to determine which of the implementation(s) is(are) the most feasible, and to what degree.

The nominal path is constructed from a known, initial position in the NE-frame. An obstacle is placed on the path, with a specified safety radius. The vessels must stay outside this safety radius. The (C)BF decides when the VS activates an evasive maneuver and when, or if, the VS should return to the nominal path. The simulations end if the VS is unable to pass the obstacle, or the OS reaches a specified point in the NE-frame along the trajectory it is tracking. The implementations and simulations are performed in MATLAB and the flow diagram in Figure 2.3 is a good representation of the structure of the MATLAB code.

The guidance system consists of the safety-critical controller, the VS model, and the LOS guidance law. Their purpose is to form the desired trajectory, and activate an evasive maneuver if required. The safety-critical controller and VS model computes the desired position and heading unit vector of the VS, along with other outputs fed to the OS controller. The evolution of the path variable is limited by the (C)BFs implemented for each VS model. The purpose of the LOS guidance law is to compute the desired course angle for the OS and feed it to the OS controller. The computation is based on the cross-track error and the heading unit vector of the VS.

The backstepping control design in the control system computes the generalized forces and moments in each DOF based on the outputs from the guidance system and the feedback from the OS vessel model. The control objective is to track the desired trajectory. An integral effect is included in the control design to reduce the steady-state error in sway direction due to the applied current.

The position, velocity, and heading of the OS and the obstacle are assumed known or perfectly measured, and available at all times. With this assumption, no observer is implemented. Hence, no navigation system block is included in the flow diagram, although this is part of a Guidance, Navigation, and Control (GNC) system on an ASV. For simplicity, the system will still be referred to as the GNC system.

All of the above systems and their implementations are presented in the following chapters.

Chapter 3

Path parametrization

According to Skjetne (2005), the primary control objective of a control system is often to steer an object towards a desired path, denoted the geometric task of the system. The dynamic task of the system could be a speed assignment. In a tracking problem, such as the one addressed in this thesis, these two tasks are merged into one. Skjetne (2005) presents the two tasks as follows:

1. Geometric task: force output y to converge to a desired path y_d, parametrized by the path variable s, such that
$$\lim_{t\to\infty} |y(t) - y_d(s(t))| = 0.$$
2. Dynamic task: force $\dot{s}$ to converge to a desired speed assignment $v(s,t)$, such that
$$\lim_{t\to\infty} |\dot{s}(t) - v(s(t),t)| = 0.$$

3.1 Path planning and path generation

Before defining the path parametrization, a short introduction to path planning and path generation is given. Path planning and generation are parts of the guidance system in a GNC system where way-points (WPs), a reference speed u_d, a reference heading ψ_{ref}, and lines/curves connecting the points are generated. The WPs are given as Cartesian coordinates in the NE-frame ($p = [x \;\; y]^\top$), and their purpose is to "indicate changes in direction, speed, and altitude along a desired path" (Fossen 2021). In order to facilitate a feasible path for the vessel to follow, the WP generation must follow some restrictions related to the vessel's specifications and dynamics. An example is the turning rate that the vessel can perform, resulting in a limit on the curvature of the path, where successive WP placement is significant. Re-planning of the WPs can be done to consider changes in weather, or to perform evasive maneuvers. The path generator connects the points using straight lines or curved paths. Fossen (2021) discusses several methods for path generation, such as path generation using straight lines and inscribed circles, and straight-line paths based on circles of acceptance. The first connects the WPs using straight lines connected to circles with a specified radii surrounding the WPs. The downside of this method is that the turning rate along straight lines equals 0, while it has a constant value along the arc of the circle in steady-state. This results in a jump in the turning rate in the transition phase between the straight line and the arc, leading to a small offset in cross-tracking. The second method uses straight-line segments and a switching mechanism when the vessel enters a radius of acceptance around the next WP, resulting in a jump to the following WP in the list of WPs. Another possibility is to switch to the following WP when the vessel is within a certain along-track distance. This may be a more intuitive and suitable switching mechanism.

This thesis concerns straight-line nominal paths in trajectory tracking scenarios. The following section presents a straight-line path parametrization using an along-path and path normal variable. The path variable identifies where along a path, or path section, the vessel is located. The path parametrization describes this mathematically.

3.2 Path parametrization using two path variables

This section is based on Skjetne (2020*a*) and Marley (2021*b*). An illustration of the path parametrization with a two-dimensional path variable is included in Figure 3.1.

Define the two-dimensional path variable as $\xi := [\xi_1 \;\; \xi_2]^\top \in \mathbb{R}^2$. $\xi_1 \in \mathbb{R}$ is the along-path variable and $\xi_2 \in \mathbb{R}$ is a path normal variable. Let $d_1,\ d_2 \in \mathbb{R}^2$ be the two points defining the straight line path p_d where $q = p_d(\xi)$ is the current position of the virtual ship (VS) tracing out the desired path. Defining the unit tangent vector $T \in \mathcal{S}^1$ and the unit normal vector $N \in \mathcal{S}^1$ as

$$T := \frac{d_2 - d_1}{|d_2 - d_1|}, \quad N := ST = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} T, \tag{3.1}$$

where S is a rotation matrix corresponding to a 90^o counterclockwise rotation and $\mathcal{S}^1$ denotes the unit circle. A straight-line can be parametrized by the along-path variable as

$$q := \xi_1(d_2 - d_1) + d_1. \tag{3.2}$$

Introducing the path normal variable, the path parametrization becomes

$$q := \xi_1(d_2 - d_1) + \xi_2 S(d_2 - d_1) + d_1 = |d_2 - d_1| \left[\xi_1 \tfrac{(d_2-d_1)}{|d_2-d_1|} + \xi_2 S \tfrac{(d_2-d_1)}{|d_2-d_1|} \right] + d_1. \tag{3.3}$$

Defining $L := |d_2 - d_1| \in \mathbb{R}$ (the norm of $d_2 - d_1$, i.e. the Euclidean distance between the two points) and simplifying, the desired path is parametrized by

$$q := L(\xi_1 T + \xi_2 N) + d_1. \tag{3.4}$$

When $\xi_2 = 0$, (3.4) reduces to (3.2). Differentiating (3.4) with respect to ξ gives

$$q^\xi = \begin{bmatrix} q^{\xi_1} & q^{\xi_2} \end{bmatrix} = L \begin{bmatrix} T & N \end{bmatrix} \in \mathbb{R}^{2\times 2}. \tag{3.5}$$

The reference value for the derivative of the path variable (the speed assignment), $\dot{\xi}$, is defined as $v(\xi) := [v_1(\xi) \;\; v_2(\xi)]^\top$. The derivative of q is

$$\dot{q}(\xi) = q^\xi \dot{\xi}. \tag{3.6}$$

To maintain a constant reference speed u_d, the requirement

$$|v(\xi)| = \frac{|u_d|}{L} \tag{3.7}$$

is introduced. The speed assignment becomes

$$\dot{\xi} \to v(\xi). \tag{3.8}$$

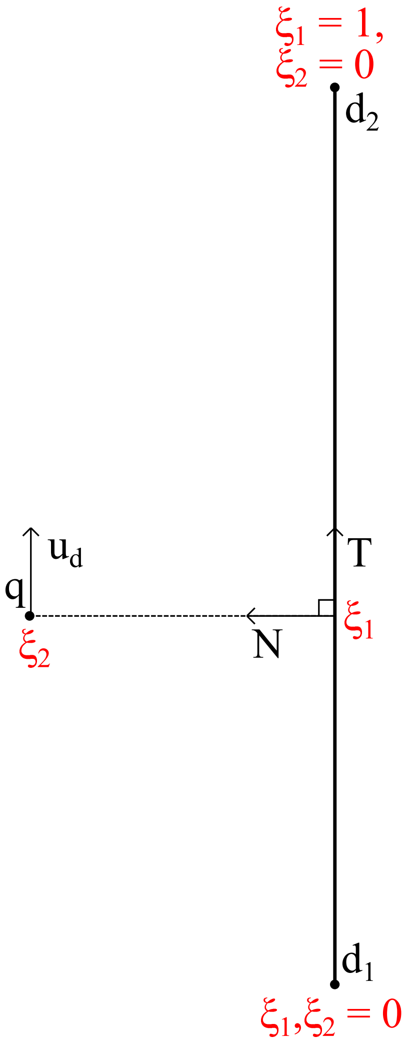


Figure 3.1: Path showing q at an offset from the straight-line nominal path, such as during evasive maneuvers, where ξ_2 is activated.

Chapter 4

Barrier functions

Ames et al. (2019) defines a safety-critical system as a system where safety is a major design consideration. They distinguish the liveness property, or the requirement that “good” things eventually happen, and safety, the requirement that “bad” things does not happen. An example of a liveness property could be asymptotic stability as this means that an asymptotically stable equilibrium point is eventually reached (see Theorem 4.1 in Khalil (2015) for definition of an asymptotically stable equilibrium point). Invariance can be defined as a safety property as any trajectory starting inside an invariant set will not reach the complement of the set being the locus where “bad” things occur. According to Ames et al. (2019), the study of safety dynamical systems, $\dot{z} = f(z)$, dates back to the 1940’s with Nagumo’s Theorem, which gives the necessary and sufficient conditions for set invariance (Nagumo 1942). Note that z refers to the state vector. Given the system $\dot{z} = f(z)$ with $z \in \mathbb{R}^n$, assume that the safe set $\mathcal{C}$ is the superlevel set of a smooth function $h : \mathbb{R}^n \to \mathbb{R}$, that is, $\mathcal{C} = \{z \in \mathbb{R}^n : h(z) > 0\}$, and that $\frac{\partial h}{\partial z}(z) \neq 0$ for all z such that $h(z) = 0$. Then the necessary and sufficient conditions for set invariance is given by Nagumo’s Theorem based on the derivative of h on the boundary of $\mathcal{C}$:

$$\mathcal{C} \text{ is invariant} \Leftrightarrow \dot{h}(z) \; \geq \; 0 \;\; \forall \;\; z \in \partial\mathcal{C},$$

where $\partial\mathcal{C}$ is the boundary of $\mathcal{C}$. In the 2000’s, barrier certificates were introduced in Prajna & Jadbabaie (2004) and Prajna (2006), as a tool to formally prove safety of hybrid and nonlinear systems (Ames et al. 2019). In that case, an unsafe set $\mathcal{C}_u$ and a set of initial states, $\mathcal{C}_0$ is considered, along with a function $B : \mathbb{R}^n \to \mathbb{R}$. $B(z) \leq 0$ for all $z \in \mathcal{C}_0$ and $B(z) > 0$ for all $z \in \mathcal{C}_u$. B is a barrier certificate if

$$\dot{B}(z) \leq 0 \Rightarrow \mathcal{C} \text{ is invariant.}$$

By picking $\mathcal{C} = \mathcal{C}_u^c$ (safe set $\mathcal{C}$ is the complement of the unsafe set $\mathcal{C}_u$) and $B(z) = -h(z)$, the barrier certificate becomes $\dot{h}(z) \geq 0$, implying that $\mathcal{C}$ is invariant. These conditions reduce to Nagumo’s Theorem on the boundary. In order to address open dynamical systems, such as control systems given by $\dot{z} = f(z) + g(z)U$, where U is the control input, required moving from invariant sets to controlled invariant sets (Ames et al. 2019). This requires a suitably designed controller. The extension of the barrier certificate led to the first definition of a CBF in Wieland & Allgwer (2007). The following section describes the barrier functions and CBFs utilized in the control design in Chapter 5. Details regrading the controllers and the virtual vessel designs are presented there, and therefore not discussed in this chapter.

4.1 Simple control barrier function

A simple form of barrier function dependent on the virtual ship (VS) position q presented in Marley, Skjetne, Breivik & Fleischer (2020) is

$$B(q) := |q - p_{obs}| - r_s. \tag{4.1}$$

Note that (4.1) gives $B(q) \geq 0$, i.e. the conditions above changes sign. This is a matter of definition of $B(q)$. (4.1) becomes a CBF with the condition

$$\dot{B}(q) = \frac{(q - p_{obs})^\top}{|q - p_{obs}|}(\dot{q} - \dot{p}_{obs}) = \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi \dot{\xi} = \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi U =: L_g B(q) U \geq -\alpha(B). \tag{4.2}$$

$L_g B(q)$ is the Lie derivative of $B(q)$ in the direction of g, as the affine control system is $\dot{q} = f(q) + g(q)U = q^\xi \dot{\xi} = q^\xi U$ (Marley 2021*b*, Section 3.1.1). p_{obs} is the position of the obstacle in the NE-frame such that $p_{obs} = [x_{obs} \;\; y_{obs}]^\top$. Assuming that the obstacle is static, $\dot{p}_{obs} = 0$. r_s is the safety radius around p_{obs}, of which the VS and OS must stay outside when passing the obstacle. The control input U is defined as $\dot{\xi}$. Along the straight-line path $U = V$, where V is the nominal control input. However, when the CBF condition no longer holds with this control input, U is determined through quadratic programming, such as presented in Thyri et al. (2020). This is discussed further in Chapter 5. The safe set, $\mathcal{C}$, for q is defined as

$$\mathcal{C} := \{q \in \mathbb{R}^2 \;:\; B(q) \geq 0\}, \tag{4.3}$$

$$\partial\mathcal{C} := \{q \in \mathbb{R}^2 \;:\; B(q) = 0\}, \tag{4.4}$$

$\alpha(B)$ in (4.2) is a class $\mathcal{K}$ function (strictly increasing and $\alpha(0) = 0$ (Khalil 2015, pp. 144)), which ensures forward invariance of the set $\mathcal{C}$ without restricting any sublevel set of $\mathcal{C}$ to be forward invariant (any q belonging to $\mathcal{C}$ will remain in $\mathcal{C}$ for all future time instants) by allowing $\dot{B}(q)$ to become increasingly negative as the value of $B(q)$ increases (Thyri et al. 2020). Thyri et al. (2020) also proposes a linear function such that

$$\alpha(B(q)) = \gamma B(q), \tag{4.5}$$

where $\gamma > 0$ is a constant, sufficiently small to ensure a region $0 < B(q) < B_{lim}$. This restricts U to reasonable inputs with respect to the own-ship (OS) dynamics, in an attempt to make the trajectory traced out by the VS feasible for the OS to track.

Marley (2021*a*, Lemma 3) shows that a non-hybrid control law (hybrid dynamical systems are introduced in Section 4.2) defined by quadratic programming, such as the above, has equilibrium points along the boundary of the obstacle. That is, $U = \mathbf{0}$. This is not desirable, and hence Marley (2021*a*) presents a synergistic CBF design based on Marley et al. (2021), which shifts the undesired equilibria. Together with a QP based controller, this can render a compact set (closed and bounded) UGAS (uniformly globally asymptotically stable) in the presence of obstacles (Marley 2021*a*). Such a CBF design from Marley et al. (2021) is presented in the following section.

4.2 Synergistic control barrier function

For vehicles required to maintain a non-zero forward speed, ordinary CBFs cannot be utilized to ensure safety. This is due to vanishing control authority when the vehicle is oriented directly towards the obstacle. Marley et al. (2021) therefore propose synergistic CBFs for obstacle avoidance for nonholonomic vehicles, such as rudder controlled ships. These are constructed by shifting the orientations with vanishing control authority, which induces a penalty for passing the obstacle clockwise or counterclockwise. A logic variable, l, determines the preferred direction.

The introduction of l means that the system becomes a hybrid dynamical affine control system of the form

$$\mathcal{H} \; : \; \begin{cases} \dot{z} = f(z) + g(z)U, & z \in C, \\ z^{+} \in H(z), & z \in D. \end{cases} \tag{4.6}$$

This is a special case of hybrid inclusions discussed in Goebel et al. (2012), where u denotes the control input. The state $z \in \mathbb{R}^n$ may evolve in both continuous time (flow) and discrete time (jumps). C refers to the flow set and D refers to the jump set. $l \in \mathcal{L}$ evolves in discrete time only. The barrier function is given by $B \; : \; \mathtt{Z} \to \mathbb{R}$ where $\mathtt{Z} \subset \mathbb{R}^n$ is continuously differentiable and defines the set

$$\mathcal{C} \coloneqq \{z \in \mathtt{Z} \; : \; B(z) \leq 0\}, \tag{4.7}$$

$$\partial\mathcal{C} \coloneqq \{z \in \mathtt{Z} \; : \; B(z) = 0\}. \tag{4.8}$$

Define $\mathcal{Z} \coloneqq \mathtt{Z} \times \mathcal{L}$ and let B be a continuously differentiable function $B \; : \; \mathcal{Z} \to \mathbb{R}$. Then (4.3) and (4.4) can be modified to include the logic variable l.

$$\mathcal{C} \coloneqq \{(z,l) \in \mathrm{Int}(\mathcal{Z}) \; : \; B(z,l) \leq 0\}, \tag{4.9}$$

$$\partial\mathcal{C} \coloneqq \{z,l) \in \mathrm{Int}(\mathcal{Z}) \; : \; B(z,l) = 0\}, \tag{4.10}$$

Denoting the control input as U, a requirement for $B(z,l)$ to be a CBF for the control system is that there exists an $\alpha(B)$, as mentioned in Section 4.1, such that

$$L_f B(z,l) + L_g B(z,l)U \leq -\alpha(B(z,l)), \tag{4.11}$$

where $L_f B(z,l)$ and $L_g B(z,l)$ are the Lie derivatives (directional derivative) of B (see Khalil 2015, pp. 509-510). Marley et al. (2021) derive a synergistic CBF candidate as follows.

Synergistic CBF candidate

Let ψ_{ref} be the heading of the VS. As mentioned, control authority vanishes if the VS is directed directly towards p_{obs}. Define $\psi_v \subset \mathcal{Z}$ as

$$\psi_v \coloneqq \{(z,l) \in \mathcal{Z} \; : \; L_g B(z,l) = 0 \text{ and } L_f B(z,l) \geq 0\}, \tag{4.12}$$

where ψ_v is the set of ψ_{ref} with vanishing control authority (set of critical orientations). Define

$$M(z) \coloneqq \min_{l \in \mathcal{L}} B(z,l). \tag{4.13}$$

The synergy gap is consequently defined as

$$\mu \coloneqq \inf_{(z,l) \in \psi_v} (B(z,l) - M(z)). \tag{4.14}$$

If $\mu > 0$, then B is a synergistic CBF candidate for the system. Selecting a δ such that $0 < \delta < \mu$, the jump and flow sets become

$$D \coloneqq \{(z,l) \in \mathcal{Z} \; : \; M(z) - B(z,l) \leq -\delta\}, \tag{4.15}$$

$$C \coloneqq (\mathbb{R}^n \diagdown D) \cup \partial D. \tag{4.16}$$

The affine control system becomes

$$\mathcal{H} \ : \ \begin{cases} \dot{z} = f(z) + g(z)U, \ \dot{l} = 0, & (z, l) \in C, \\ l^+ \in H(z), \ z^+ = z, & (z, l) \in D, \end{cases} \tag{4.17}$$

where $H(z) \coloneqq \{l \in \mathcal{L} \ : \ B(z,l) = M(z)\}$. If there exists an $\alpha(B)$ satisfying (4.11), B is a synergistic CBF candidate.

The CBFs in Marley et al. (2021) consist of a non-hybrid CBF, which loses control authority when the VS is directed directly towards p_{obs} and a synergistic CBF with shifted orientations clockwise or counterclockwise based on l, resulting in a preferred turning direction. The VS model used in combination with these CBFs is a unicycle model, which is discussed in Section 5.3. These two CBFs are eventually combined into a new synergistic CBF, which ensures safety even when the VS is in the interior of the safety radius r_s. This extends the logic variable l such that

$$l \coloneqq \begin{bmatrix} l_0 \\ l_1 \end{bmatrix} \in \mathcal{L}_0 \times \mathcal{L}_1 =: \mathcal{L}, \tag{4.18}$$

where $\mathcal{L}_0 \coloneqq \{0, 1\}$ and $\mathcal{L}_1 \coloneqq \{-1, 1\}$. When $l_0 = 1$ the VS is in an evasive mode, while l_1 assigns preferred turning direction.

Non-hybrid CBF

The non-hybrid CBF is as follows:

$$B_0(q, Z) \coloneqq r_s - |q - p_{obs}| - t_0 u_d \frac{(q - p_{obs})^\top}{|q - p_{obs}|} Z, \tag{4.19}$$

where $B_0 \ : \ \mathsf{Z} \times \mathcal{S}^1 \to \mathbb{R}$ ($\mathcal{S}^1 \coloneqq \{Z \in \mathbb{R}^2 \ : \ Z^\top Z = 1\}$ denotes the unit circle), $Z \in \mathcal{S}^1$ is the unit heading vector, q is the position of the VS, r_s is the safety radius around the obstacle position p_{obs}, and the last term is the relative velocity between the VS and the obstacle weighted by a time constant $t_0 \in \mathbb{R}_{>0}$. u_d is the desired forward speed of the VS. Note that (4.19) resembles (4.1), except it has the opposite sign and (4.1) does not take the relative velocity into consideration.

The states of the system are therefore $(q, Z, l) \in \mathbb{R}^2 \times \mathcal{S}^1 \times \mathcal{L}$, and with the unicycle modelling described in Section 5.3, the affine control system is

$$\begin{bmatrix} \dot{q} \\ \dot{Z} \end{bmatrix} = f(Z) + g(Z)U, \quad \dot{l} = 0, \tag{4.20}$$

$$f(Z) \coloneqq \begin{bmatrix} Z u_d \\ 0 \end{bmatrix}, \quad g(Z) \coloneqq \begin{bmatrix} 0 \\ SZ \end{bmatrix}. \tag{4.21}$$

The Lie derivatives of B_0 are

$$L_f B_0(q, Z) = -u_d \frac{(q - p_{obs})^\top}{|q - p_{obs}|} Z - t_0 \frac{u_d^2}{|q - p_{obs}|} \left(\frac{(q - p_{obs})^\top}{|q - p_{obs}|} S^\top Z \right)^2, \tag{4.22}$$

$$L_g B_0(q, Z) = -t_0 u_d \frac{(q - p_{obs})^\top}{|q - p_{obs}|} SZ. \tag{4.23}$$

Control authority vanishes for

$$\psi_{v0} = \{(q, Z, l) \in \mathcal{Z} \;:\; L_g B_0(q, Z) = 0 \text{ and } L_f B_0(q, Z) \geq 0\}, \tag{4.24}$$

and this occurs when the VS is directed directly towards or away from the obstacle. The safe set is defined by B_0 as

$$\mathcal{C}_0 \coloneqq \{(q, Z, l) \in \mathrm{Int}(\mathcal{Z}) \;:\; B_0(q, Z) \leq 0\}. \tag{4.25}$$

$\psi_{v0} \cap \mathcal{C}_0 \neq \emptyset$, motivating a hybrid CBF.

Synergistic CBF

The synergistic CBF $B_1 \;:\; \mathtt{Z} \times \mathcal{S}^1 \times \mathcal{L}_1 \to \mathbb{R}$ must be able to shift the set of critical orientations. To enable this, Marley et al. (2021) introduces a function $P \;:\; \mathtt{Z} \times \mathcal{S}^1 \times \mathcal{L}_1 \to [-1, 1]$,

$$P \coloneqq P(q, Z, l_1) \coloneqq \frac{(q - p_{obs})^\top}{|q - p_{obs}|} R(Z^{l_1 k_1}) Z, \tag{4.26}$$

with $k_1 \in \left(0, \frac{\pi}{2}\right)$ and

$$R(Z^{l_1 k_1}) \coloneqq \begin{bmatrix} \cos(l_1 k_1) & -\sin(l_1 k_1) \\ \sin(l_1 k_1) & \cos(l_1 k_1) \end{bmatrix}. \tag{4.27}$$

$R(Z^{l_1 k_1})$ shifts the critical orientations by $l_1 k_1$ radians in the counterclockwise direction. B_1 is then defined as

$$B_1(q, Z, l_1) \coloneqq r_s - |q - p_{obs}| - t_1 u_d (P(q, Z, l) - \sin(k_1)). \tag{4.28}$$

$t_1 \in \mathbb{R}_{>0}$ is a time constant. $\sin(k_1)$ ensures forward invariance of $\mathcal{C}_1$ (see proposition 3 in Marley et al. (2021)), where B_1 defines $\mathcal{C}_1$ as

$$\mathcal{C}_1 \coloneqq \{(q, Z, l) \in \mathrm{Int}(\mathcal{Z}) \;:\; B_1(q, Z, l_1) \leq 0\}. \tag{4.29}$$

The directional derivatives of B_1 are

$$L_f B_1(q, Z, l_1) = -u_d \frac{(q - p_{obs})^\top}{|q - p_{obs}|} Z - t_1 u_d L_f P(q, Z, l1), \tag{4.30}$$

$$L_g B1(q, Z, l_1) = -t_1 u_d L_g P(q, Z, l_1), \tag{4.31}$$

where the directional derivatives of P are

$$L_f P(q, Z, l1) = \frac{u_d}{|q - p_{obs}|} \frac{(q - p_{obs})^\top}{|q - p_{obs}|} S Z \frac{(q - p_{obs})^\top}{|q - p_{obs}|} R(Z^{l_1 k_1}) S Z, \tag{4.32}$$

$$L_g P(q, Z, l1) = \frac{(q - p_{obs})^\top}{|q - p_{obs}|} R(Z^{l_1 k_1}) S Z. \tag{4.33}$$

The set of critical orientations becomes

$$\psi_{v1} \coloneqq \{(q, Z, l) \in \mathcal{Z} \;:\; L_g B_1(q, Z, l_1) = 0 \text{ and } L_f B_1(q, Z, l_1) \geq 0\}. \tag{4.34}$$

The synergy for B_1 is defined as

$$\mu_1 \coloneqq \inf_{(q,Z,l) \in \psi_{v1}} \{B_1(q, Z, l_1) - M_1(q, Z)\}, \tag{4.35}$$

where

$$M_1(q, Z) \coloneqq \min_{l_1 \in \mathcal{L}_1} B_1(q, Z, l_1). \tag{4.36}$$

Combination of the non-hybrid CBF and the synergistic CBF
The final CBF combining the non-hybrid CBF and the synergistic CBF is

$$B(q, Z, l) \coloneqq (1 - l_0)B_0(q, Z) + l_0 B_1(q, Z, l_1), \tag{4.37}$$

where the logic variable l_0 decides whether $B = B_0$ or $B = B_1$, and therefore also the directional derivatives of B. B defines the set

$$\mathcal{C} \coloneqq \{(q, Z, l) \in \operatorname{int}(\mathcal{Z}) \ : \ B(q, Z, l) \leq 0\}. \tag{4.38}$$

The critical orientations are defined as

$$\psi_v \coloneqq \{(q, Z, l) \in \mathcal{Z} \ : \ L_g B(q, Z, l) = 0 \text{ and } L_f B(q, Z, l) > 0\}, \tag{4.39}$$

and the synergy gap is

$$\mu \coloneqq \inf_{(q,Z,l) \in \psi_v} \{B(q, Z, l) - M(q, Z)\}, \tag{4.40}$$

with

$$M(q, Z) \coloneqq \min_{l \in \mathcal{L}} B(q, Z, l). \tag{4.41}$$

For the control input to be safe, the requirement is

$$L_f B(q, Z, l) + L_g B(q, Z, l)U \leq -\alpha(B(q, Z, l)), \tag{4.42}$$

where $\alpha(B)$ is defined by (4.5).

With the assumptions that:

1. the set of control inputs is a compact convex set (a closed and bounded set, which contains the entire segment connecting two points) containing a neighbourhood of the origin,
2. there exists a control law, which is continuously differentiable, such that the compact set $\mathcal{A} \subset \mathbb{R}^2$ is UGAS for the obstacle free system,
3. the intersection of $\mathcal{A}$ and the obstacle domain is empty,

Marley (2021*a*, Theorem 1) proves that $\mathcal{A}$ is rendered UGAS with a similar synergistic CBF design as presented in this section.

4.3 Barrier function with modified obstacle

This section presents a barrier function which will be used in combination with a potential field to steer the VS to a point, in this case the origin. The design is extracted from Marley (2020*a*). Define $d(q) \ : \ \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ as the distance from the VS to the obstacle,

$$d(q) \coloneqq \begin{cases} |q - p_{obs}| - r_s, & |q - p_{obs}| > r_s \\ 0, & |q - p_{obs}| \leq r_s, \end{cases} \tag{4.43}$$

and the barrier function $B \ : \ \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ as

$$B(d(q)) \coloneqq \begin{cases} (d(q) - d_{lim})^2 \ln\left(\frac{d_{lim}}{d(q)}\right), & 0 \leq d(q) \leq d_{lim} \\ 0, & d(q) > d_{lim}. \end{cases} \tag{4.44}$$

d_{lim} denotes the minimum distance away from the obstacle boundary before the barrier function B is activated. The gradient of B is

$$\nabla B(d(q)) = \frac{\partial B(d(q))}{\partial d(q)} \nabla d(q), \tag{4.45}$$

where the partial derivative of B with respect to d is

$$\frac{\partial B(d(q))}{\partial d(q)} = \begin{cases} \frac{(d(q)-d_{lim})}{d(q)} \left(d_{lim} - d(q) + 2d(q)\ln\left(\frac{d_{lim}}{d(q)}\right)\right), & 0 < d(q) \leq d_{lim} \\ 0, & d(q) > d_{lim}, \end{cases} \tag{4.46}$$

and the gradient of $d(q)$ is

$$\nabla d(q) = \begin{cases} \frac{(q-p_{obs})}{|q-p_{obs}|}, & |q-p_{obs}| > r_s \\ 0, & |q-p_{obs}| \leq r_s. \end{cases} \tag{4.47}$$

$\frac{\partial B(d(q))}{\partial d(q)}$ is continuous for all $d(q) > 0$ and undefined for $d(q) = 0$, and $\nabla d(q)$ is continuous everywhere except for on the obstacle boundary. As Section 5.4 will show, there is a saddle point along the North-axis. All trajectories with $q_2 \neq 0$ will converge to the origin due to the potential function, however all trajectories with $(q_1 \in [p_{obs} + r_s, \infty), q_2 = 0)$ will converge to the saddle point. This can be prevented by creating a modified obstacle, which grazes the original obstacle radius on one side, while adding a safety margin on the other side.

A logic variable $l \in \mathcal{L} := \{-1, 1\}$ is used to redefine the obstacle position as

$$p_m := p_{obs} + \begin{bmatrix} 0 \\ l\sigma \end{bmatrix}, \tag{4.48}$$

where $\sigma > 0$ is the additional safety margin. The switching logic for l is described in Section 5.4. The obstacle radius becomes $r_m := r_s + \sigma$, and the modified distance $d_m : \mathbb{R}^2 \times \mathcal{L} \to \mathbb{R}_{\geq 0}$ is defined by

$$d_m(q) := \begin{cases} |q-p_m| - r_m, & |q-p_m| > r_m \\ 0, & |q-p_m| \leq r_m. \end{cases} \tag{4.49}$$

The barrier function is thus

$$B(d_m(q)) := \begin{cases} (d_m(q) - d_{lim})^2 \ln\left(\frac{d_{lim}}{d_m(q)}\right), & 0 \leq d_m(q) \leq d_{lim} \\ 0, & d_m(q) > d_{lim}. \end{cases} \tag{4.50}$$

4.4 Control barrier function with modified obstacle

Another possible CBF is a combination of the non-hybrid CBF, B_0, presented in Marley et al. (2021) with the modified obstacle presented in Marley (2020*a*). The modified obstacle position p_m and the modified radius r_m is as defined in Section 4.3. With the inclusion of a modified obstacle, (4.19) is defined as

$$B_0(q, Z) := r_m - |q-p_m| - t_0 u_d \frac{(q-p_m)^\top}{|q-p_m|} Z, \tag{4.51}$$

and the directional derivatives are

$$L_f B_0(q, Z) = -u_d \frac{(q - p_m)^\top}{|q - p_m|} Z - t_0 \frac{u_d^2}{|q - p_m|} \left(\frac{(q - p_m)^\top}{|q - p_m|} S^\top Z \right)^2, \tag{4.52}$$

$$L_g B_0(q, Z) = -t_0 u_d \frac{(q - p_m)^\top}{|q - p_m|} S Z. \tag{4.53}$$

The set of safe control inputs will be defined by

$$L_f B_0(q, Z) + L_g B_0(q, Z) U \leq -\alpha(B_0(q, Z)), \tag{4.54}$$

where $\alpha(B)$ is given in (4.5).

Chapter 5

Virtual vessel

The guidance system of the own-ship (OS) constitutes a virtual vessel (VS) and a line-of-sight (LOS) guidance law. They produce the input to the control system. The OS control system aims to track the trajectory traced out by the VS. This chapter presents the different models used to simulate the motion of the VS, along with the safety critical controllers, which employ the (control) barrier functions ((C)BFs) presented in Chapter 4. The LOS guidance law providing input directly to the OS control system is presented in Chapter 6.

5.1 Quadratic programming

In a quadratic programming (QP) problem, the goal is to maximize or minimize an objective function, which must satisfy some specified bounds, and linear equality and/or inequality constraints. A general example of a minimization QP problem extracted from MathWorks (n.d.*b*) is

$$f(x) = \min_{x} \left\{ \frac{1}{2} x^\top Q x + f^\top x \right\}, \tag{5.1}$$

subject to

$$\begin{aligned} A_{qp} x &\leq b_{qp}, \\ A_{eq} x &= b_{eq}, \\ lb \leq x &\leq ub. \end{aligned} \tag{5.2}$$

The QP problem minimizes the objective function $f(x)$ by minimizing the vector x. Q is a positive definite matrix and f is a vector. A_{qp} and A_{eq} are matrices, and b_{qp} and b_{eq} are vectors. These describe the inequality and equality constraints. The problem bounds are given by lb and ub, which denote the lower bound and upper bound respectively.

In the VS control designs, QP is used in most of the simulations to find a safe control input by minimizing the difference between a nominal control input V and the safe control input U. That is $x = (U - V)$, while $f = \mathbf{0}$.

5.2 Particle model

The particle model implemented is a first order particle model with linear velocities as control inputs. The nominal control input, V, is defined by the time derivative of the path variable, $\dot{\xi}$. In the trajectory tracking scenarios in the simulations, there is no feedback from the vessel states to the evolution of ξ, such that $\dot{\xi}$ equals the speed assignment $v(\xi)$. $v_1 = \frac{u_d}{L}$, where u_d is a desired surge speed, and $L := |d_2 - d_1|$. d_1 and d_2 are the first and last way-points (WPs) of a path section. v_2 is a negative signed hyperbolic tangent function dependent on the path normal variable ξ_2. The hyperbolic tangent function allows the VS to gradually return to the nominal path if desirable. A constant k decides how rapidly the VS returns to the nominal path. If $k = 0$ the VS will continue straight ahead after an evasive maneuver. To maintain a constant reference speed, the requirement presented in (3.7) is $|v(\xi)| = \frac{u_d}{L}$. Since $V = v(\xi)$, then $|V| = \frac{u_d}{L}$. V_2 is the negative signed hyperbolic tangent function weighted by k, and thus V_1 is derived from $|V| = \sqrt{V_1^2 + V_2^2} = \frac{u_d}{L}$ (Marley 2021*b*).

$$V := \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{u_d}{L}^2 - V_2^2} \\ -k\tanh(\xi_2) \end{bmatrix}. \tag{5.3}$$

When the vessel is travelling along the straight-line nominal path, the control input to the particle model U equals the nominal control input V. However, during an obstacle encounter V is not able to perform an evasive maneuver. QP is therefore employed to find a safe control input U with a minimum difference from V. The objective function is

$$f(U - V) = \min_{(U-V)} \left\{ \frac{1}{2}(U - V)^\top Q(U - V) \right\} \in \mathbb{R}^2 \tag{5.4}$$

To encourage the vessel to turn rather than stop at the boundary of the obstacle, the Q-matrix weights the control input corresponding to the surge velocity, $U_1 - V_1$, higher than the control input difference corresponding to the sway velocity, $U_2 - V_2$. That is, $Q_{11} > Q_{22}$. The cross-diagonal terms, Q_{12} and Q_{21} are set to 0. Q_{12} and Q_{21} not equal to 0 will adjust the equilibrium point along the boundary of the obstacle. Regardless of the weighting of the diagonal terms, the VS stagnates at the equilibrium point. With the cross-diagonal terms set to 0, it is located on the axis running directly through the obstacle.

The QP inequality constraint is derived from the CBF as follows:

$$B(q) := |q - p_{obs}| - r_s, \tag{5.5}$$

$$\dot{B}(q) = \frac{(q - p_{obs})^\top}{|q - p_{obs}|}(\dot{q} - \dot{p}_{obs}) = \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi \dot{\xi} = \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi U \geq -\alpha(B), \tag{5.6}$$

with $\alpha(B) := \gamma B$. $\gamma = \frac{1}{T_B}$, where T_B is a suitable time constant chosen such that both the VS and the OS is able to start the evasive maneuver in time to avoid the obstacle. Switching the sign of the inequality in (5.6) results in

$$-\frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi U \leq \alpha(B). \tag{5.7}$$

From (5.7), the set of safe control inputs is derived as

$$U_B := \left\{ U \in \mathcal{U} \ : \ -\frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi U - \alpha(B) \leq 0 \right\}, \tag{5.8}$$

where $\mathcal{U} \in \mathbb{R}^2$ is a set of control inputs. (5.7) is similar to the QP inequality constraint except $x = U - V$, not U. By adding $\frac{(q-p_{obs})^\top}{|q-p_{obs}|} q^\xi V$ on both sides, the following expression emerges,

$$-\frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi (U - V) \leq \alpha(B) + \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi V. \tag{5.9}$$

The QP inequlity is

$$A_{qp}(U - V) \leq b_{qp}. \tag{5.10}$$

A_{qp} and b_{qp} are thus

$$A_{qp} = -\frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi, \tag{5.11}$$

and

$$b_{qp} = \alpha(B) + \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi V. \tag{5.12}$$

The QP problem is therefore to solve (5.4) subject to the linear inequality constraint

$$-\frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi (U - V) \leq \alpha(B) + \frac{(q - p_{obs})^\top}{|q - p_{obs}|} q^\xi V. \tag{5.13}$$

The QP problem is solved using the MATLAB "quadprog" function (MathWorks n.d.a). U is found by adding V to the QP solution as it is solved for $U - V$. The path is parametrized as in Section 3.2. The virtual vessel model with linear velocity as control input is therefore

$$q := L(\xi_1 T + \xi_2 N) + d_1, \tag{5.14}$$

$$\dot{q} = q^\xi \dot{\xi} = q^\xi U = L \begin{bmatrix} T & N \end{bmatrix} U. \tag{5.15}$$

$\dot{\xi} = U$ such that

$$\xi = \xi + U\Delta t, \tag{5.16}$$

where Δt is the simulation time step. Recall that T is the path tangent vector and N is the path normal vector. The unit heading vector is Z. This is given as

$$Z = \begin{cases} \frac{\dot{q}}{|\dot{q}|}, & |\dot{q}| > 0 \\ T, & |\dot{q}| = 0 \end{cases}, \tag{5.17}$$

where $|\dot{q}|$ is the speed of the VS. Effectively, (5.17) is a map from a particle model to a unicycle model. This becomes obvious in the following section, where $|\dot{q}|$ corresponds to u_d in (5.19).

5.3 Unicycle model

The unicycle model is implemented as presented in Marley et al. (2021). The unicycle kinematics are given by

$$\dot{Z} = SZ\omega, \tag{5.18}$$

and

$$\dot{q} = Zu_d. \tag{5.19}$$

$\dot{Z}$ is the kinematic equation for motion along the unit circle, and $\dot{q}$ is the linear x and y velocities of the VS in NE-frame. $\omega = \dot{\psi}_{ref}$ is the heading rate, which along with the linear acceleration in surge, $\dot{u}_d$, forms the control input $U := [\omega \quad \dot{u}_d]^\top$. The unit heading vector $Z := [\cos(\psi_{ref}) \quad \sin(\psi_{ref})]^\top$, the desired forward speed u_d, and the VS position q are found through Euler integration:

$$\begin{aligned} Z &= \frac{Z}{|Z|} + \dot{Z}\Delta t, \\ u_d &= u_d + \dot{u}_d \Delta t, \\ q &= q + \dot{q}\Delta t. \end{aligned} \tag{5.20}$$

The forward speed u_d is assumed constant such that $\dot{u}_d = 0$.
The affine control system is thus

$$\begin{bmatrix} \dot{q} \\ \dot{Z} \end{bmatrix} = f(Z) + g(Z)\omega, \quad \dot{l} = 0. \tag{5.21}$$

l is the logic variable discussed in Section 4.2.

The safe control input U is found through quadratic programming (QP) from the objective function defined in (5.4). The inequality constraint is determined based on the inequality in (4.42), where $U = U_1$ as $U_2 = 0$ and constant. Hence, $x = U_1 - V_1 \in \mathbb{R}$ and the objective function is

$$\min_{(U_1 - V_1)} \left\{ \frac{1}{2}(U_1 - V_1)^\top Q(U_1 - V_1) \right\} \in \mathbb{R} \tag{5.22}$$

The inequality constraint is derived as follows:

$$L_f B(q, Z, l) + L_g B(q, Z, l) U_1 \leq -\alpha(B(q, Z, l)), \tag{5.23}$$

with $\alpha(B) \coloneqq \gamma B$ where $\gamma = \frac{1}{T_B}$. The set of safe control inputs is

$$U_B \coloneqq \{U_1 \in \mathcal{U} \ : \ L_f B(q, Z, l) + L_g B(q, Z, l) U_1 + \alpha(B(q, Z, l)) \leq 0\}, \tag{5.24}$$

where $\mathcal{U} \in \mathbb{R}$ is a set of control inputs. By moving $L_f B(q, Z, l)$ to the right side of (5.23) and subtracting $L_g B(q, Z, l) V_1$ on both sides the result is

$$L_g B(q, Z, l)(U_1 - V_1) \leq -\alpha(B(q, Z, l)) - L_f B(q, Z, l) - L_g B(q, Z, l) V_1. \tag{5.25}$$

Therefore,

$$A_{qp} = L_g B(q, Z, l) \tag{5.26}$$

and

$$b_{qp} = -\alpha(B(q, Z, l)) - L_f B(q, Z, l) - L_g B(q, Z, l) V_1. \tag{5.27}$$

For the synergistic CBF presented in Section 4.2, the CBF employed depends on the logic variable l_0. If $M(q, Z) - B(q, Z, l) \leq -\delta$ for $\delta > 0$ and $\delta < \mu$ (recall that μ is the synergy gap defined in (4.40)), then l_0 is toggled, i.e. it switches from 0 to 1 or vice versa. If l_0 is toggled to 1, then depending on (4.36), l_1 may be toggled as well, i.e. $l_1^+ = -l_1$. This decides the preferred turning direction. The jump and flow sets are defined similarly to (4.15) and (4.16),

$$D \coloneqq \{(q, Z, l) \in \mathcal{Z} \ : \ M(q, Z) - B(q, Z, l) \leq -\delta\}, \tag{5.28}$$

$$C \coloneqq ((\mathbb{R}^2 \times \mathcal{S}^1 \times \mathcal{L}) \diagdown D) \cup \partial D. \tag{5.29}$$

Recall that $q \in \mathbb{R}^2$, $Z \in \mathcal{S}^1$ and $l \in \mathcal{L}$. To determine δ, a saturation limit for the heading rate ω_{sat} is chosen. ω_{sat} should not be higher than what is a reasonable turning rate for the OS. The time constant in B_1 is selected such that $t_1 = \omega_{sat}^{-1}$. The time constant in B_0 is given as $t_0 = t_1(\sin(k_1) + 1)$. This is based on proposition 5 in Marley et al. (2021), which states that B has a non-zero synergy gap if $t_0 > t_1(\cos(k_1) + \sin(k_1)) > 0$, and if this is satisfied $\mu = \min\{\mu_1, u_d(t_0 - t_1(\cos(k_1) + \sin(k_1))\} > 0$. Thus, $t_0 = t_1(\sin(k_1) + 1)$ yields $\mu = \mu_1 > 0$. This is because $\mu_1 = t_1 u_d(1 - \cos(k_1))$ ((58) in Marley et al. (2021)). A possible $\delta > 0$, which is used in the simulations, is $\delta = \frac{\mu}{4}$.

In the case of the CBF with a modified obstacle, the CBF corresponding to $l_0 = 0$ is utilized. The modified obstacle position p_m is adjusted based on a logic variable $l \in \{-1, 1\}$. The modified safety radius r_m creates a circle which graces the original safety radius r_s on one side, while adding an additional safety margin on the opposite side. This induces a preferred turning direction towards the side where $r_m = r_s + \sigma$ graces r_s. The switching logic for l is given as

$$|q - p_m(-l)| - |q - p_m(+l)| \leq -\delta, \tag{5.30}$$

with p_m as defined in (4.48). Defining the set $\mathcal{Q} := \mathbb{R}^2 \times \mathcal{L}$ The jump and flow sets can thus be defined as

$$D := \{(q, l) \in \mathcal{Q} \ : \ |q - p_m(-l)| - |q - p_m(+l)| \leq -\delta\}, \tag{5.31}$$

$$C := ((\mathbb{R}^2 \times \mathcal{L}) \setminus D) \cup \partial D. \tag{5.32}$$

δ is determined differently in the case with a modified obstacle than in the case with a hybrid synergistic CBF. The top figure in Figure 5.1 shows the modified obstacles and radii, in relation to the original obstacle position, p_{obs}, and the position of the VS, q. $\zeta := |p_m - p_{obs}|$ is the Euclidean distance between the obstacle at p_{obs} and the modified obstacle at p_m. The bottom figure in Figure 5.1 illustrates the switching logic, which is such that given ζ and a distance $\beta_{r_m} := d_\beta(r_m + t_0 u_d)$, where $\mathrm{c} = \sqrt{(\beta_{r_m})^2 + (2\zeta)^2}$, a value $\Delta := \mathrm{c} - \beta_{r_m}$ is defined. δ is then given as $\delta = -\frac{1}{2}\Delta$. In this case, $t_0 = \omega_{sat}^{-1}$ as $t_0 = t_1 := \omega_{sat}^{-1}$ if $k_1 = 0$, which is the case if no adjustment of the critical orientations is desired. d_β is a constant, which must be suitably tuned.

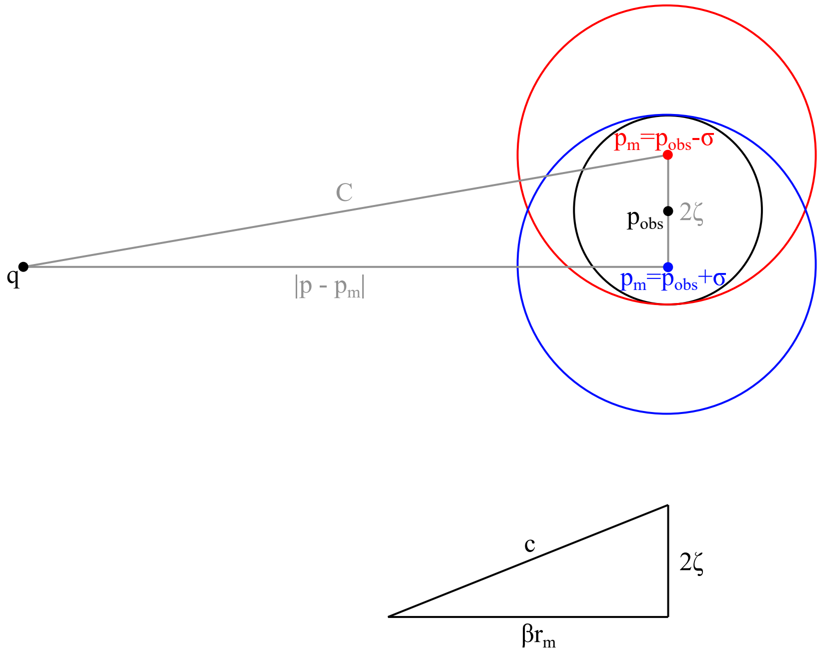


Figure 5.1: Modified obstacles and switching logic triangle.

The upper and lower bounds of the QP problem depends on the saturation limit of the heading rate ω_{sat}. That is, $\omega \in [-\omega_{sat}, \omega_{sat}]$. Since $U_1 = \omega$ and the QP problem minimizes $U_1 - V_1$, the upper bound is $ub = \omega_{sat} - V_1$ and the lower bound is $lb = -\omega_{sat} - V_1$.

The nominal control input V_1 is determined in two ways in the simulations. The first is simply

$$V_1 := -Z_2 = -\sin(\psi_{ref}). \tag{5.33}$$

The second is the method presented in Marley et al. (2021), which uses the line-of-sight (LOS) algorithm presented in Fossen (2021) with desired path along the North-axis ($y = 0$). The LOS orientation vector is defined as in Marley, Skjetne, Breivik & Fleischer (2020) as

$$Z_{LOS} = \frac{1}{\sqrt{\Delta_{LOS} + q_2^2}} \begin{bmatrix} \Delta_{LOS}^2 \\ -q_2 \end{bmatrix} \in \mathcal{S}^1. \tag{5.34}$$

Δ_{LOS} is the look-ahead distance. The desired orientation vector for the VS is the LOS orientation vector rotated by the matrix containing the path tangent and path normal vectors,

$$Z_d = \begin{bmatrix} T & N \end{bmatrix} Z_{LOS}. \tag{5.35}$$

When $T = [1 \;\; 0]^\top$ and $N = [0 \;\; 1]^\top$, $Z_d = Z_{LOS}$. The nominal control law for V_1 is in this case a non-hybrid version of the hybrid kinematic feedback controller presented in Marley, Skjetne & Teel (2020), such that

$$V_1 := -\omega_{sat} \frac{\tilde{Z}_2}{\sqrt{1 - \lambda^2 \tilde{Z}_1^2}}, \tag{5.36}$$

where $0 < \lambda < 1$ is a regularization parameter, which, if sufficiently high, makes V_1 approximately constant for most $Z_d \in \mathcal{S}^1$, and

$$\tilde{Z} = \begin{bmatrix} Z_d & SZ_d \end{bmatrix}^\top Z = R(Z_d)Z. \tag{5.37}$$

$R(Z_d) \in SO(2)$ denotes a map from the unit circle $\mathcal{S}^1 := \{Z \in \mathbb{R}^2 \;:\; Z^\top Z = 1\}$ to planar rotations $SO(2) := \{R \in \mathbb{R}^{2\times 2} \;:\; R^\top R = I,\; det(R) = 1\}$.

The QP problem is therefore to solve (5.22) subject to the linear inequality constraint

$$L_gB(q, Z, l)(U_1 - V_1) \leq -\alpha(B(q, Z, l)) - L_fB(q, Z, l) - L_gB(q, Z, l)V_1 \tag{5.38}$$

and the upper and lower bounds

$$\begin{aligned} lb &= -\omega_{sat} - V_1, \\ ub &= \omega_{sat} - V_1. \end{aligned} \tag{5.39}$$

5.4 Potential function

The potential function presented in this section is described in Marley (2020*a*), which is based on Sanfelice et al. (2006).

With no obstacle present, the potential function dependent on the position of the VS, q, is defined as

$$P_0(q) := \frac{1}{2}(q_1^2 + q_2^2). \tag{5.40}$$

The gradient with respect to q is

$$\nabla P_0(q) = \begin{bmatrix} \frac{\partial P_0(q)}{\partial q_1} \\ \frac{\partial P_0(q)}{\partial q_2} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}. \tag{5.41}$$

The control input considered is the VS velocity, $\dot{q}$. Choosing $\dot{q} = -\nabla P_0$ results in the time derivative of P_0 in (5.42).

$$\dot{P}_0(q) = \nabla P_0^\top \dot{q} = -\nabla P_0^\top \nabla P_0 = -(q_1^2 + q_2^2) \leq 0. \tag{5.42}$$

(5.42) implies global asymptotic stability (GAS) by Theorem 4.1 and Theorem 4.2 in Khalil (2015). To include an obstacle, a new potential function P is defined, where the barrier function $B(d(q))$ is presented in (4.44),

$$P := P_0 + B(d(q)). \tag{5.43}$$

The gradient of P with respect to q is

$$\nabla P = \nabla P_0 + \nabla B(d(q)), \tag{5.44}$$

where

$$\nabla B(d(q)) = \frac{\partial B(d)}{\partial d} \nabla d(q). \tag{5.45}$$

The partial derivative of B with respect to d and the gradient of $d(q)$ is given in (4.46) and (4.47) respectively. Like mentioned in Section 4.3, the partial derivative of B is continuous for all $d(q) > 0$ and undefined for $d(q) = 0$. $d(q)$ is continuous everywhere except for on the boundary of the obstacle. Intuitively, if $d(q)$ is undefined, then $B(d(q))$ is undefined as well. Choosing the control input $\dot{q} = -\nabla P$ gives $\dot{P} = -(\nabla P_0 + \nabla B)^\top (\nabla P_0 + \nabla B) \leq 0$, which implies stability by Lyapunov theory (Khalil 2015, pp. 114). However, there is a saddle point at $\nabla P_0 + \nabla B = 0$. This point is located along the North-axis. If a trajectory starts out along this axis, no incentive will be given to diverge from the axis when the VS is closing in on the obstacle boundary as $\dot{q}_2 = 0$. Hence, p_{obs} must be shifted to induce a $\dot{q}_2 \neq 0$, enabling the VS to perform an evasive maneuver. Hence, the modified obstacle is introduced. $d_m(q)$ and $B(d_m(q))$ is presented in (4.49) and (4.50) respectively. The new potential function is

$$P_m(q) = P_0 + B(d_m(q)). \tag{5.46}$$

This allows all trajectories starting near, or on, the North-axis to converge to the origin. The logic variable deciding the preferred turning direction by adjusting p_m one direction or the other, toggles ($l^+ = -l$) if the Euclidean distance between the VS position and the origin, $|q|$, is greater than the Euclidean distance between the modified obstacle position and the origin, $|p_m|$, and $lq_2 > -\mu_P |q|$. $\mu_P = \beta_P \frac{\sigma}{|p_m|}$, where β_P is a tuning parameter. The first condition ensures that the VS is further away from the origin than the obstacle, that is, the VS has not passed the obstacle yet. The second condition ensures that the VS chooses the most suitable turning direction, i.e. it turns port when q_2 is located on the port side of the modified obstacle, and starboard when q_2 is located on the starboard side. The flow and jump sets can therefore be defined as

$$D := \{(q, l) \in \mathcal{Q} \ : \ |q| > |p_m|, \ lq_2 > -\mu_P |q|\}, \tag{5.47}$$

$$C := ((\mathbb{R}^2 \times \mathcal{L}) \diagdown D) \cup \partial D. \tag{5.48}$$

q is determined through Euler integration of $\dot{q}$. The unit heading vector Z is given in (5.17).

Chapter 6

Vessel model and controller

The “Ship” block in the flow diagram presented in Figure 2.3 represents the own-ship (OS) maneuvering model. The model in this chapter is presented in Marley (2020*b*), which uses models and principles from Fossen (2011) and Faltinsen (1993). The tracking controller uses a line-of-sight (LOS) course reference, and the LOS guidance law is presented in Section 6.2. Furthermore, it is based on a backstepping control design, and it provides the control input to the maneuvering model. The control design is presented in Section 6.3.

6.1 Vessel model

Recall that the kinematic equation relating the body-frame velocities $\nu = [u \;\; v \;\; r]^\top$ and the OS pose $\eta = [x \;\; y \;\; \psi]^\top$ is given by

$$\dot{\eta} = R(\psi)\nu, \quad R(\psi) := \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.1}$$

If there is no current present and the vessel is travelling along a straight line, the course angle χ equals the yaw angle ψ. However, that is not the case in the simulations presented in Chapter 7. Hence, a course angle χ and a crab angle β must be calculated. Let $U_s := \sqrt{u^2 + v^2} = \sqrt{\dot{x}^2 + \dot{y}^2}$ be the ship speed. χ is derived from the fact that it satisfies

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} U_s \cos(\chi) \\ U_s \sin(\chi) \end{bmatrix}. \tag{6.1}$$

Thus $\tan(\chi) = \frac{\dot{y}}{\dot{x}}$ and χ is given by

$$\chi = \operatorname{atan2}(\dot{y}, \dot{x}). \tag{6.2}$$

The crab angle β is derived similarly where

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} U_s \cos(\beta) \\ U_s \sin(\beta), \end{bmatrix} \tag{6.3}$$

where $\tan(\beta) = \frac{v}{u}$ and therefore

$$\beta = \operatorname{atan2}(v, u). \tag{6.4}$$

atan2 maps to $[-\pi \;\; \pi]$. The course angle, yaw angle, and crab angle are related through

$$\chi = \psi + \beta. \tag{6.5}$$

The ocean current triangle in Figure 2.2 illustrates the relationships in (6.1) to (6.5). x and y corresponds to x_n and y_n respectively.

The simulations consider a constant, irrotational current, i.e. the speed U_c and flow direction β_c is constant. The current velocities in the inertial NE-frame $\{n\}$ are given by

$$\nu_c^n = \begin{bmatrix} u_c^n \\ v_c^n \\ r_c^n \end{bmatrix} = \begin{bmatrix} U_c \cos(\beta_c) \\ U_c \sin(\beta_c) \\ 0 \end{bmatrix}, \tag{6.6}$$

while the body-fixed current velocities are

$$\nu_c = \begin{bmatrix} u_c \\ v_c \\ r_c \end{bmatrix} = \begin{bmatrix} U_c \cos(\beta_c - \psi) \\ U_c \sin(\beta_c - \psi) \\ 0 \end{bmatrix}. \tag{6.7}$$

The relative velocity between the OS and the current $\nu_r = [u_r \;\; v_r \;\; r_r]^\top$ is defined as

$$\nu_r := \nu - \nu_c = \begin{bmatrix} u - u_c \\ v - v_c \\ r \end{bmatrix}. \tag{6.8}$$

The relative accelerations become

$$\dot{\nu}_r = \dot{\nu} - \dot{\nu}_c = \begin{bmatrix} \dot{u} - U_c \sin(\beta_c - \psi)r \\ \dot{v} - U_c \; (-\cos(\beta_c - \psi))r \\ \dot{r} \end{bmatrix}. \tag{6.9}$$

$\dot{U}_c = 0$, $\dot{\beta}_c = 0$, and $\dot{\psi} = r$. The relative speed is defined as $U_r := \sqrt{u_r^2 + v_r^2}$. The angle between the relative speed and the body-fixed x-axis is called the sideslip angle β_s, and it satisfies

$$\nu_r = \begin{bmatrix} U_r \cos(\beta_s) \\ U_r \sin(\beta_s) \\ 0 \end{bmatrix}, \tag{6.10}$$

such that $\tan(\beta_s) = \frac{v_r}{u_r}$. Hence,

$$\beta_s = \text{atan2}(v_r, u_r). \tag{6.11}$$

The equation of motion for the OS is given by

$$M_{rb}\dot{\nu} + M_a\dot{\nu}_r + C_{rb}(\nu)\nu + C_a\nu + D(\nu_r)\nu_r = \tau. \tag{6.12}$$

M_{rb} is the rigid-body mass matrix, M_a is the added mass matrix, C_{rb} is the rigid-body Coriolis matrix, C_a is the added mass Coriolis matrix, and D is the damping matrix. The added mass and rigid-body matrices are diagonal due to assumed port-starboard and fore-aft symmetry. There are no centripetal forces when the rigid-body and added mass matrices are diagonal. The damping matrix is also assumed diagonal. τ denotes the forces in surge and sway, and moment in yaw. Collecting the mass terms into $M := M_{rb} + M_a$ and solving (6.12) with respect to $\dot{\nu}$, knowing $\dot{\nu}_r$ from (6.9), results in the following expression,

$$\dot{\nu} = M^{-1}(\tau - D(\nu_r)\nu_r - C_{rb}(\nu)\nu - C_a(\nu_r)\nu_r + M_a\dot{\nu}_c). \tag{6.13}$$

6.2 Line-of-sight guidance

The OS controller receives a desired course angle, χ_d, from an LOS guidance law. The trajectory that the OS should track is the trajectory traced out by the VS. Hence, the LOS guidance law must use the cross-track error between the position of the OS, p, and the position of the VS, q. Figure 6.1 visualizes LOS guidance with desired course angle χ_d. The cross-track error is denoted y_e^p. x_e^p denotes the along-track error. The superscript p refers to the path-tangential coordinate system notation $\{p\}$. p_i^n and p_{i+1}^n denotes the WPs of the path section, similar to d_1 and d_2 as mentioned earlier.

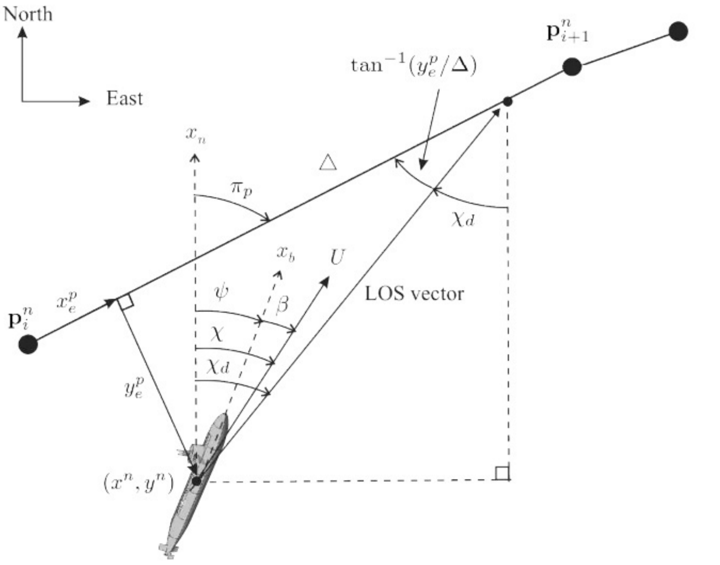


Figure 6.1: Line-of-sight guidance with desired course angle χ_d. *Adapted from Fossen (2021).*

The tracking error vector in the path tangential coordinate system is calculated as in (12.55) in Fossen (2021),

$$\begin{bmatrix} y_e^p \\ x_e^p \end{bmatrix} = \begin{bmatrix} Z & SZ \end{bmatrix}^\top (p - q), \tag{6.14}$$

where S is the rotation matrix corresponding to a 90^o counterclockwise rotation, and $[Z \quad SZ]^\top$ corresponds to the transposed rotation matrix of π_p. π_p is the angle between the inertial NE-frame, $\{n\}$, and the path tangential frame, $\{p\}$. With the defined lookahead-distance Δ_{LOS}, the LOS orientation vector is given as in Marley, Skjetne, Breivik & Fleischer (2020),

$$Z_{LOS} := \frac{1}{\sqrt{\Delta_{LOS}^2 + (y_e^p)^2}} \begin{bmatrix} \Delta_{LOS} \\ -y_e^p \end{bmatrix}. \tag{6.15}$$

The desired orientation vector in the inertial $\{n\}$ NE-frame is

$$Z_d = \begin{bmatrix} Z & SZ \end{bmatrix} Z_{LOS} = \begin{bmatrix} \cos(\chi_d) \\ \sin(\chi_d) \end{bmatrix}. \tag{6.16}$$

resulting in the desired course angle

$$\chi_d = \operatorname{atan2}(Z_{d2}, Z_{d1}). \tag{6.17}$$

Figure 6.2 illustrates the the steps in (6.14) to (6.17), where the deviation in x- and y-direction in the NE-frame is rotated to the path tangential frame to find the along- and cross-track errors. These are subsequently used to find the LOS orientation vector, which must be rotated back to the NE-frame before the desired course angle is extracted. To make the figure more compact, the VS straight-line nominal path is along the East-axis. However, in the simulations the nominal path is along the North-axis. The principle is the same in either case. A close-up of the details in Figure 6.2 is included in Figure 6.3.

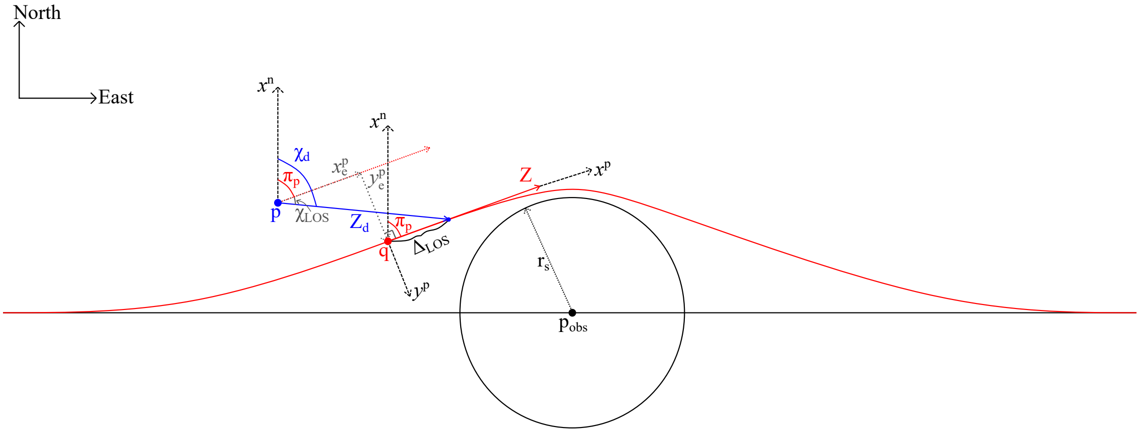


Figure 6.2: Line-of-sight guidance for OS trajectory tracking with desired course angle.

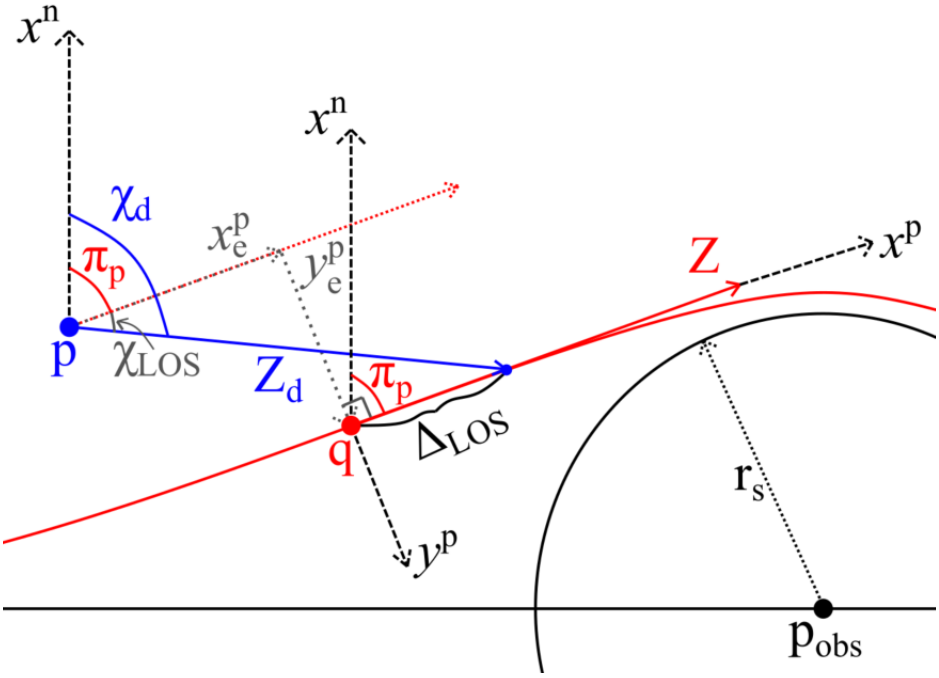


Figure 6.3: Close up of line-of-sight guidance for OS trajectory tracking with desired course angle from Figure 6.2.

6.3 Control design

Given the OS vessel model in (6.13), a control design model can be derived. This is a simplified mathematical model, which is detailed enough to describe the main physical characteristics of the system (Sorensen 2013). The model is given by

$$M\dot{\nu} = \tau - D\nu + R(\psi)^\top b. \tag{6.18}$$

b is a vector of bias estimates, usually provided by an observer. However, the simulations only have an unknown bias from the current, for which there is no estimate. Hence, $b = \mathbf{0}$. M and D is as specified in Section 6.1. Using this model, a control law enabling trajectory tracking is designed using a backstepping design methodology. The control design constructs a feedback control law through recursive construction of a control Lyapunov function (CLF) (Fossen 2021). CLFs are defined in Section 6.3.1. Initially a control design using a one-dimensional path variable is presented in Section 6.3.2. This is extended to a two-dimensional path variable in Section 6.3.3. Section 6.3.3 also introduces an integral term to compensate for the unknown current that the OS experiences in the simulation studies.

6.3.1 Control Lyapunov functions

Definition 16.2 in Fossen (2021) defines CLFs as follows based on Artstein (1983) and Sontag (1983):

Definition (Control Lyapunov function)
A smooth positive definite and radially unbounded function $V : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is called a control Lyapunov function for

$$\dot{x} = f(x, u), \tag{6.19}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^r$ if

$$\inf_{u \in \mathbb{R}^r} \left\{ \frac{\partial V}{\partial x}(x) f(x, u) \right\} < 0, \quad \forall x \neq 0. \tag{6.20}$$

x is the state vector and u denotes the control input. Radially unbounded refers to the condition that $V(x) \to \infty$ as $|x| \to \infty$ (Khalil 2015, pp. 123). Section 16.3.2 in Fossen (2021) gives a simple example of an integrator backstepping design using CLFs as defined above. The steps of that design is recognized in the control designs in the following sections.

6.3.2 Control design with one path variable

The backstepping control design goes through two steps to derive the control law for τ. Recall that τ is the vector containing the generalized control forces and moment. The design presented in this section uses the one-dimensional path variable, s. This is extended to the two-dimensional path variable, ξ, in Section 6.3.3. The designs are based on Skjetne (2020*b*), Jensen (2020) and Fleischer (2020), with adaptions to perform trajectory tracking.

Step 1 Initially, an error variable z_1, which should be controlled to 0, is defined as

$$z_1 = R(\psi)^\top [\eta - \eta_d(s)], \tag{6.21}$$

where η_d is the desired path parametrized by s. Differentiating (6.21) with respect to time gives

$$\dot{z}_1 = \dot{R}(\psi)^\top [\eta - \eta_d] + R(\psi)^\top [\dot{\eta} - \eta_d^s \dot{s}] = -S(r) z_1 + \nu - R(\psi)^\top \eta_d^s \dot{s}, \tag{6.22}$$

where

$$S(r) = \begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{6.23}$$

and $\dot{R}(\psi)^\top = -R(\psi)^\top S(r)$. The step 1 CLF is

$$V_1 \coloneqq \frac{1}{2} z_1^\top z_1. \tag{6.24}$$

Differentiating V_1 with respect to time yields

$$\dot{V}_1 = z_1^\top \dot{z}_1 = -z_1^\top S(r) z_1 + z_1^\top \nu - z_1^\top R(\psi)^\top \eta_d^s \dot{s}. \tag{6.25}$$

The error variable for the second CLF is

$$z_2 \coloneqq \nu - \alpha_1(t, s, \eta), \tag{6.26}$$

such that

$$\dot{V}_1 = -z_1^\top S(r) z_1 + z_1^\top z_2 + z_1^\top [\alpha_1 - R(\psi)^\top \eta_d^s \dot{s}]. \tag{6.27}$$

In trajectory tracking $\dot{s} = v(s)$. To make $\dot{V} < 0$ (negative definite), the virtual control input is

$$\alpha_1 = -K_1 z_1 + R(\psi)^\top \eta_d^s v(s), \quad K_1 = K_1^\top > 0, \tag{6.28}$$

and thus

$$\dot{z}_1 = -S(r) z_1 + z_2 + \alpha_1(t, s, \eta) - R(\psi)^\top \eta_d^s \dot{s} = -K_1 z_1 - S(r) z_1 + z_2, \tag{6.29}$$

and

$$\dot{V}_1 = -z_1^\top K_1 z_1 + z_1^\top z_2, \tag{6.30}$$

as

$$-z_1^\top S(r) z_1 = -\begin{bmatrix} z_{11} & z_{12} & z_{13} \end{bmatrix} \begin{bmatrix} 0 & -r & 0 \\ r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{11} \\ z_{12} \\ z_{13} \end{bmatrix} \tag{6.31}$$

$$= -\begin{bmatrix} z_{11} & z_{12} & z_{13} \end{bmatrix} \begin{bmatrix} -r z_{12} \\ r z_{11} \\ 0 \end{bmatrix} \tag{6.32}$$

$$= r z_{11} z_{12} - r z_{11} z_{12} + 0 = 0. \tag{6.33}$$

Step 2 Recall that the error variable $z_2 \coloneqq \nu - \alpha_1(t, s, \eta)$ should be controlled to 0. Differentiating with respect to time yields

$$\dot{z}_2 = \dot{\nu} - \dot{\alpha}_1. \tag{6.34}$$

Multiplying (6.34) by M, the term $M\dot{\nu}$ appears, which is given in (6.18), resulting in

$$M \dot{z}_2 = \tau - D\nu + R(\psi)^\top b - M \dot{\alpha}_1. \tag{6.35}$$

The second CLF is

$$V_2 = \frac{1}{2} z_2^\top M z_2 + V_1, \tag{6.36}$$

and V_2 differentiated with respect to time yields

$$\dot{V}_2 = z_2^\top M \dot{z}_2 + \dot{V}_1 = z_2^\top M(\dot{\nu} - \dot{\alpha}) - z_1^\top K_1 z_1 + z_1^\top z_2. \tag{6.37}$$

Substituting the term $M\dot{\nu}$ with (6.18) gives

$$\dot{V}_2 = -z_1^\top K_1 z_1 + z_2^\top [z_1 + \tau - D\nu + R(\psi)^\top b - M\dot{\alpha}] \tag{6.38}$$

yielding the control law

$$\tau = -z_1 + D\alpha_1 - R(\psi)^\top b + M\dot{\alpha}_1 - K_2 z_2, \quad K_2 = K_2^\top > 0. \tag{6.39}$$

such that

$$\dot{V}_2 = -z_1^\top K_1 z_1 - z_2^\top (K_2 + D) z_2 < 0, \tag{6.40}$$

and thus the equilibrium point $[z_1 \quad z_2]^\top = \mathbf{0}$ is uniformly globally exponentially stable (UGES).

Tuning Let $T_1 = \text{diag}(T_x, T_y, T_{psi})$ be the diagonal matrix containing time constants for the z_1-subsystem. Set $K_1 = T_1^{-1}$. To ensure that the z_2-subsystem is faster than the z_1-subsystem, T_2 should be less than T_1. With $T_2\dot{z}_2 = -z_2$, then inserting (6.39) into (6.35), results in an expression for K_2, where $K_2 = MT_2^{-1} - D + \frac{z_1}{\dot{z}_2}T_2^{-1}$.

6.3.3 Control design with two path variables

The simulation studies use a two-dimensional path variable $\xi = [\xi_1 \quad \xi_2]^\top \in \mathbb{R}^2$, and thus the backstepping control design must be extended to include both the path tangent and path normal variable. Note that the design in this section uses a heading reference. In the simulations, this is replaced by a desired course angle produced by the LOS guidance law, and its derivatives. The reason why a heading reference is not used in the simulations is explained in Chapter 7, and the consequences are discussed.

In this design, the position and heading control is decoupled and later combined into a virtual control α_1. Therefore,

$$\eta_d(t,\xi) := \begin{bmatrix} p_d(\xi_1, \xi_2) \\ \psi_d(t, \xi_1, \xi_2), \end{bmatrix} \tag{6.41}$$

where $q = p_d(\xi)$ is the position of the VS and ψ_d is the heading reference. The speed assignment is defined by

$$v(t,\xi) := \begin{bmatrix} v_1(t, \xi_1, \xi_2) \\ v_2(t, \xi_1, \xi_2) \end{bmatrix}. \tag{6.42}$$

The rotation matrix $R_2(\psi)$ is defined as

$$R_2(\psi) := \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \end{bmatrix}, \tag{6.43}$$

such that

$$\dot{p} = R_2(\psi)\nu_p, \quad \nu_p := \begin{bmatrix} u \\ v \end{bmatrix}. \tag{6.44}$$

The error variables of the two subsystems are

$$z_{1,p} := R_2(\psi)^\top[p-q], \qquad z_{1,\psi} := \psi - \psi_d(t,\xi_1,\xi_2), \qquad z_1 := \mathrm{col}(z_{1,p}, z_{1,\psi}), \tag{6.45}$$
$$z_{2,p} := \nu_p - \alpha_p, \qquad z_{2,\psi} := r - \alpha_\psi, \qquad z_2 := \mathrm{col}(z_{1,\psi}, z_{2,\psi}), \tag{6.46}$$
$$\alpha_1 := \mathrm{col}(\alpha_p, \alpha_\psi). \tag{6.47}$$

Step 1 First, the tracking design for position is done. Differentiating $z_{1,p}$ with respect to time gives

$$\dot{z}_{1,p} = \dot{R}_2(\psi)^\top[p-q] + R_2(\psi)^\top[\dot{p} - q^{\xi_1}\dot{\xi}_1 - q^{\xi_2}\dot{\xi}_2] \tag{6.48}$$
$$= -S_2(r)z_{1,p} + z_{2,p} + \alpha_p - R_2(\psi)^\top[q^{\xi_1}v_1 + q^{\xi_2}v_2]. \tag{6.49}$$

where

$$S_2(r) = \begin{bmatrix} 0 & -r \\ r & 0 \end{bmatrix}. \tag{6.50}$$

The first CLF for position is

$$V_{1,p} := \frac{1}{2}z_{1,p}^\top z_{1,p}. \tag{6.51}$$

Differentiating (6.51) with respect to time yields

$$\dot{V}_{1,p} = z_{1,p}^\top \dot{z}_{1,p} = z_{1,p}^\top z_{2,p} + z_{1,p}^\top(\alpha_p - R_2(\psi)^\top[q^{\xi_1}v_1 + q^{\xi_2}v_2]). \tag{6.52}$$

Recall that $-z_{1,p}^\top S_2(r)z_{1,p} = 0$. The virtual control law α_p is

$$\alpha_p = -K_{1,p}z_{1,p} + R_2(\psi)^\top[q^{\xi_1}v_1 + q^{\xi_2}v_2], \quad K_{1,p} = K_{1,p}^\top > 0, \tag{6.53}$$

such that

$$\dot{V}_{1,p} = -z_{1,p}^\top K_{1,p}z_{1,p} + z_{1,p}^\top z_{2,p}. \tag{6.54}$$

Inserting α_p into (6.49) gives

$$\dot{z}_{1,p} = -S_2(r)z_{1,p} + z_{2,p} - K_{1,p}z_{1,p}. \tag{6.55}$$

Next, consider the heading control design. Assuming that $\dot{\psi}_d$ is available, differentiating $z_{1,\psi}$ with respect to time yields

$$\dot{z}_{1,\psi} = \dot{\psi} - \dot{\psi}_d = r - \dot{\psi}_d = z_{2,\psi} + \alpha_\psi - \dot{\psi}_d. \tag{6.56}$$

The first CLF for heading is defined as

$$V_{1,\psi} := \frac{1}{2}z_{1,\psi}^2. \tag{6.57}$$

Differentiated with respect to time, the resulting expression is

$$\dot{V}_{1,\psi} = z_{1,\psi}\dot{z}_{1,\psi} = z_{1,\psi}[z_{2,\psi} + \alpha_\psi - \dot{\psi}_d]. \tag{6.58}$$

Hence, the virtual control for heading is

$$\alpha_\psi = -k_{1,\psi}z_{1,\psi} + \dot{\psi}_d, \quad k_{1,\psi} > 0. \tag{6.59}$$

Inserting (6.59) into (6.58) yields

$$\dot{V}_{1,\psi} = z_{1,\psi}z_{2,\psi} - z_{1,\psi}k_{1,\psi}z_{1,\psi}. \tag{6.60}$$

Step 2 The second CLF for position is

$$V_{2,p} := V_{1,p} + \frac{1}{2} z_{2,p}^\top M_p z_{2,p} \tag{6.61}$$

where M_p denotes the first two rows of the mass matrix M. The derivative of (6.61) yields

$$\dot V_{2,p} = \dot V_{1,p} + z_{2,p}^\top M_p \dot z_{2,p} = -z_{1,p}^\top K_{1,p} z_{1,p} + z_{2,p}^\top z_{1,p} + z_{2,p}^\top M_p [\dot\nu_p - \dot\alpha_p]. \tag{6.62}$$

Inserting $M_p \dot\nu_p$ into (6.62) gives

$$\dot V_{2,p} = -z_{1,p}^\top K_{1,p} z_{1,p} + z_{2,p}^\top [z_{1,p} - D_p \nu_p + R_2(\psi)^\top b_p + \tau_p - M_p \dot\alpha_p]. \tag{6.63}$$

The control law is thus

$$\tau_p = -z_{1,p} + D_p \alpha_p - R_2(\psi)^\top b_p + M_p \dot\alpha_p - K_{2,p} z_{2,p}, \quad K_{2,p} = K_{2,p}^\top > 0, \tag{6.64}$$

such that

$$\dot V_{2,p} = -z_{1,p}^\top K_{1,p} z_{1,p} - z_{2,p}^\top (K_{2,p} + D) z_{2,p}. \tag{6.65}$$

The second CLF for heading is

$$V_{2,\psi} := V_{1,\psi} + \frac{1}{2} z_{2,\psi} M_\psi z_{2,\psi}, \tag{6.66}$$

where M_ψ denotes the last row of the mass matrix. The time derivative of $V_{2,\psi}$ is

$$\dot V_{2,\psi} = V_{1,\psi} + z_{2,\psi} M_\psi \dot z_{2,\psi} = z_{1,\psi} z_{2,\psi} - z_{1,\psi} k_{1,\psi} z_{1,\psi} + z_{2,\psi} M_\psi [\dot r - \dot\alpha_\psi]. \tag{6.67}$$

Inserting for $M_\psi \dot r$ gives

$$\dot V_{2,\psi} = -z_{1,\psi} k_{1,\psi} z_{1,\psi} + z_{2,\psi} [z_{1,\psi} - D_\psi r + b_\psi + \tau - M_\psi \dot\alpha_\psi], \tag{6.68}$$

yielding the control law

$$\tau_\psi = -z_{1,\psi} + D_\psi \alpha_\psi - b_\psi + M_\psi \dot\alpha_\psi - k_{2,\psi} z_{2,\psi}, \quad k_{2,\psi} > 0. \tag{6.69}$$

The total control law τ is therefore

$$\tau = -z_1 + D\alpha_1 - R(\psi)^\top b + M\dot\alpha_1 - K_2 z_2, \quad K_2 = K_2^\top > 0, \tag{6.70}$$

which is familiar from (6.39). (6.70) requires $\dot\alpha_1$, which is given by

$$\dot\alpha_1 := \mathrm{col}(\dot\alpha_p, \dot\alpha_\psi), \tag{6.71}$$

where

$$\begin{aligned} \dot\alpha_p &= -K_{1,p} \dot z_{1,p} + \dot R_2(\psi)^\top [q^{\xi_1} v_1 + q^{\xi_2} v_2] &(6.72)\\ &= -K_{1,p} \dot z_{1,p} - S_2(r) R_2(\psi)^\top [q^{\xi_1} v_1 + q^{\xi_2} v_2] &(6.73)\\ &= K_{1,p}(S_2(r) z_{1,p} - \nu_p) + (K_{1,p} - S_2(r)) R_2(\psi)^\top [q^{\xi_1} v_1 + q^{\xi_2} v_2], &(6.74) \end{aligned}$$

with $\dot z_{1,p}$ as given in (6.49), and

$$\begin{aligned} \dot\alpha_\psi &= -k_{1,\psi} \dot z_{1,\psi} + \ddot\psi_d &(6.75)\\ &= -k_{1,\psi}(z_{2,\psi} + \alpha_\psi - \dot\psi_d) + \ddot\psi_d &(6.76)\\ &= -k_{1,\psi}(r - \dot\psi_d) + \ddot\psi_d. &(6.77) \end{aligned}$$

The OS is exposed to a constant, unknown current. As mentioned there is no bias estimate available, thus $b = \mathbf{0}$. The current must be compensated for using an integral term in the control law in (6.70). The resulting control law is

$$\tau = -z_1 + D\alpha_1 + M\dot{\alpha}_1 - K_2 z_2 - K_I e_1, \quad K_I = K_I^\top > 0, \tag{6.78}$$

where K_I is the integral gain matrix. e_I denotes the column vector containing the integrated along- and cross-track errors, and the deviation from the heading reference ψ_d, such that

$$e_I = e_I + e\Delta t, \quad e := \begin{bmatrix} x_e^p \\ y_e^p \\ \psi_e \end{bmatrix}. \tag{6.79}$$

In practice, (6.78) becomes a PID-controller. This is no longer guaranteed UGAS, but should at least be locally stable, and able to satisfy the tracking control objective.

Tuning The time constants for the $z_{1,p}$-subsystem is given by $T_{1,p} = \text{diag}(T_x, T_y)$, and $K_{1,p} = T_{1,p}^{-1}$. $T_{1,\psi}$ is the time constant for the $z_{1,\psi}$-subsystem, where $k_{1,\psi} = T_{1,\psi}^{-1}$. The time constant for the z_2-subsystem is T_2. For the z_2-subsystem to be faster than the z_1-subsystem, $T_2 < \text{diag}(T_{1,p}, T_{1,\psi})$. All time constants are determined by tuning. With $T_2\dot{z}_2 = -z_2$, then inserting (6.39) into (6.35), results in the expression for K_2, where $K_2 = MT_2^{-1} - D + \frac{z_1}{\dot{z}_2}T_2^{-1}$.

Chapter 7

Simulation studies

This chapter presents the simulation parameters, the simulation results, and discussions related to the individual results. The final discussion and conclusions with regards to the research question are made in Section 8.1. The simulations concern a virtual vessel (VS) following a straight-line nominal reference path, on which an obstacle is placed. The obstacle has a circular boundary surrounding it of which the VS must stay outside. This is ensured through the barrier functions presented in Chapter 4. The control objective of the own-ship (OS) is to track the trajectory traced out by the VS. The feasibility of each trajectory is the focus of the discussions. All the figures discussed in the result sections are included in larger format in the Appendix A to G, along with the remaining figures produced, which are not mentioned in the discussions. Note that the LOS guidance presented in Section 6.2 produces a desired course angle, not a desired heading angle. Hence, the OS controller controls the OS heading to the desired course angle. This results in a steady-state offset between the heading output and the desired course angle input. However, the resulting course angle goes to the desired course angle, and the cross-track errors goes to zero as desired. Conversion to desired heading angle, $\psi_d = \chi_d - \beta = \chi_d - \text{atan2}(v, u)$ was not possible during the simulations due to an algebraic loop as the controller controls the sway velocity v as well as the heading rate r, and thus the heading. That is, v depends on ψ_d and ψ_d depends on v. The conversion is done after the simulation is completed and it is included in the simulation plots. The desired heading rate is computed from $r_d = \dot{\psi}_d = \dot{\chi}_d - \dot{\beta}$.
A code algorithm for the simulation file is included in Section 7.1.

7.1 Code algorithm

Algorithm 1: main.m

```
Result: trajectory tracking simulation results
Define OS model parameters
input  : VS model index identifier
while VS model index identifier $\neq 1$ or 2 or 3 or 4 do
    input  : VS model index identifier
end
input  : Obstacle position
while Obstacle position $\notin \mathbb{R}^{2\times 1}$ do
    input  : Obstacle position
end
if VS model index identifier $\neq 2$ then
    input  : End of path
    while End of path $\notin \mathbb{R}^{2\times 1}$ do
        input  : End of path
    end
    else
        End of path = $[0 \;\; 0]^\top$                /* due to potential function */
end
input  : VS initial position
while VS initial position $\notin \mathbb{R}^{2\times 1}$ do
    input  : VS initial position
end
Parametrize path
Create VS models
Set simulation parameter values
while Not end of path do
    Compute $q$, $q^\xi$, $\dot{\xi}$, $Z$
    Compute tracking errors
    Compute $\chi_d$ with LOS
    Compute $\dot{\chi}_d$
    Compute ocean current values
    Compute $\tau$ with backstepping control
    Compute $\dot{\nu}$ for maneuvering model
    $\dot{\eta} = R(\psi)\nu$
    $\nu = \nu + \dot{\nu}\Delta t$
    $\eta = \eta + \dot{\eta}\Delta t$
    Store results
    if $\dot{q}_x < 10^{-5}$ then
        break
    end
end
Post-process results
Plot results
```

7.2 Simulation parameters

The following subsections present the simulation parameters. Several of the parameter values are extracted from relevant references from Chapters 4, 5 and 6.

7.2.1 Virtual vessel simulation parameters

Table 7.1 contains the simulation parameters and their values used in all the simulations. The tables following Table 7.1 include the simulation specific parameters.

Table 7.1: Overview of simulation parameters for the virtual vessels and desired trajectory (*see (7.16)).

Parameter	Symbol	Values	Unit
Obstacle position	p_{obs}	$[1000 \;\; 0]^\top$	m
Euclidean distance between subsequent WPs	L	1	m
Path tangent vector	T	$[1 \;\; 0]^\top$	-
Path normal vector	N	$[0 \;\; 1]^\top$	-
Desired surge velocity	u_d	$0.5 u_{crit}{}^*$	m/s
Obstacle radius	r_s	100	m
Derivative of q with respect to path variable ξ	q^ξ	$\text{diag}(1,1)$	m
Derivative of ξ with respect to time	$\dot{\xi}$	$\dot{q}$	m/s

q^ξ and $\dot{\xi}$ are the result of the path parametrization where $q^\xi = L[T \quad N] = \text{diag}(1,1)$, and $\dot{q} = q^\xi \dot{\xi} = \dot{\xi}$.

Table 7.2: Overview of simulation parameters for the particle model with CBF.

Parameter	Symbol	Values	Unit
Simulation time step	Δt	0.1	s
End of path	WP_{end}	$[2500 \;\; 0]^\top$	m
Reference speed for path variable	u_{ref}	u_d/L	m/s
Constant forcing VS back to nominal path	k	$0.1 u_{ref}$	m/s
CBF time constant	T_B	50	s
$\alpha(B)$ constant	γ	$1/T_B$	s^{-1}
QP positive definite matrix terms	Q_{11}, Q_{12}, Q_{21}, Q_{22}	7, 0, 0, 1	-

Table 7.3: Overview of simulation parameters for the potential function with modified obstacle.

Parameter	Symbol	Values	Unit
Simulation time step	Δt	0.01	s
Initial VS position	d_1	$[2500 \;\; 0]^\top$	m
End of path	WP_{end}	$[0 \;\; 0]^\top$	m
Additional safety margin	σ	$0.4 r_s$	m
Distance to obstacle boundary when B is activated	d_{lim}	$2 r_s$	m
Switching logic tuning variable	β_P	0.001	-

Table 7.4: Overview of simulation parameters for the unicycle model with synergistic CBF.

Parameter	Symbol	Values	Unit
Simulation time step	Δt	0.1	s
Initial VS position	d_1	$[0 \ \ 0]^\top$	m
End of path	WP_{end}	$[2500 \ \ 0]^\top$	m
Saturation limit of angular velocity	ω_{sat}	$2\pi/180$	rad/s
Shift angle of critical orientations	k_1	$\pi/2.5$	rad
B_1 time constant	t_1	ω_{sat}^{-1}	s
B_0 time constant	t_0	$t_1(\sin(k_1)+1)$	s
Synergy gap	μ	$t_1 u_d(1-\cos(k_1))$	m
Switching logic constant	δ	$\mu/4$	m
LOS look-ahead distance	Δ_{LOS}	500	m
Regularization parameter	λ	0.9	-
CBF time constant	T_B	50	s
$\alpha(B)$ constant	γ	$1/T_B$	s^{-1}
QP term	Q	1	-

Table 7.5: Overview of simulation parameters for the unicycle model with modified obstacle.

Parameter	Symbol	Values	Unit
Simulation time step	Δt	0.1	s
Initial VS position	d_1	$[0 \ \ 0]^\top$	m
End of path	WP_{end}	$[2500 \ \ 0]^\top$	m
Saturation limit of angular velocity	ω_{sat}	$2\pi/180$	rad/s
B_0 time constant	t_0	ω_{sat}^{-1}	s
LOS look-ahead distance	Δ_{LOS}	500	m
Regularization parameter	λ	0.9	-
CBF time constant	T_B	50	s
$\alpha(B)$ constant	γ	$1/T_B$	s^{-1}
Additional safety margin	σ	$0.4r_s$	m
Switching logic distance constant	d_β	10	-
QP term	Q	1	-

Note that t_0 equals t_1 from Table 7.4, as $k_1 = 0$, i.e. the critical orientations are not shifted.

7.2.2 Vessel model and controller simulation parameters

Table 7.6 contains the simulation parameters the OS and the current. The ship parameters are extracted from Marley (2020*b*) where they are based on engineering judgement. The mass, damping, Coriolis, and controller gain time constant matrices are presented at the end of the section.

Table 7.6: Overview of simulation parameters for the OS and the current.

Parameter	Symbol	Values	Unit
Ship length	L	100	m
Ship breadth	B	L/10	m
Ship draft	D	L/20	m
Fresh water density	ρ	1000	kgm^{-3}
Rigid-body mass	m	LBD $\times$ ρ	kg
Inertia radius in yaw	l_r	L/4	m
Reference surge velocity for linearized damping	u_o	6	m/s
Reference sway velocity for linearized damping	v_o	2	m/s
Reference yaw velocity for linearized damping	r_o	$2\pi/180$	rad/s
Drag coefficient in surge	C_{dx}	0.5	-
Drag coefficient in sway	C_{dy}	1	-
LOS look-ahead distance	Δ_{LOS}	100	m
Current speed	U_c	1	m/s
Current direction in NE-frame	β_c	$\pi/4$	rad

Notice that $u_d = 0.1u_{crit}$ is the desired surge velocity for the VS, not the OS, hence it is not included in the above table.

The mass matrices are defined as

$$M_{rb} := \text{diag}(m, m, ml_r^2), \tag{7.1}$$

$$M_a := \text{diag}(a_{11}, a_{22}, a_{33}) = \text{diag}(0.05m, 0.3m, a_{22}l_r^2), \tag{7.2}$$

$$M := M_{rb} + M_a. \tag{7.3}$$

Damping is selected as the sum of linear and quadratic damping. According to Marley (2020*b*), the overall damping level may be unrealistically high. However, the chosen values presented next are believed to give a reasonable ratio between damping level in the different DOFs. The surge and sway quadratic damping terms are selected as

$$d_{11q} := \frac{1}{2}\text{BD}C_{dx} \times \rho, \quad d_{22q} := \frac{1}{2}\text{LD}C_{dy} \times \rho. \tag{7.4}$$

Yaw damping is found by integrating the yaw-induced sway force along the hull, resulting in the following quadratic damping term

$$d_{33q} := C_{dy}\text{D}\frac{\text{L}^4}{64}\rho. \tag{7.5}$$

The linear damping terms are included through the quadratic damping force at the reference velocities in Table 7.6.

$$d_{11} := d_{11q}u_o, \quad d_{22} := d_{22q}v_o, \quad d_{33} := d_{33q}r_o. \tag{7.6}$$

The damping matrices are thus

$$D_l := \text{diag}(d_{11}, d_{22}, d_{33}), \tag{7.7}$$

$$D_{nl}(\nu) := \text{diag}(d_{11q}|u_r|, d_{22q}|v_r|, d_{33q}|r|), \tag{7.8}$$

$$D(\nu) := D_l + D_{nl}(\nu). \tag{7.9}$$

Marley (2020*b*) derives the rigid-body and added mass Coriolis matrices as follows. The rigid-body forces, evaluated in a rotating reference frame about the center of gravity, are given by

$$M_{rb}\dot{\nu} + C_{rb}(\nu)\nu := \begin{bmatrix} m(\dot{u} - vr) \\ m(\dot{v} + ur) \\ ml_r^2\dot{r} \end{bmatrix}. \tag{7.10}$$

With the assumption that there are no external forces

$$\dot{u} = vr, \ \dot{v} = -ur, \tag{7.11}$$

such that, by (15) in Marley (2020*b*),

$$\dot{U}_s = 0. \tag{7.12}$$

Hence,

$$C_{rb}(\nu)\nu := \begin{bmatrix} -mvr \\ mur \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} 0 & -mr & 0 \\ mr & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{7.13}$$

The added mass Coriolis forces are derived from the hydrodynamic inertia loads, which are given by

$$M_a\dot{\nu}_r + C_a(\nu_r)\nu_r := \begin{bmatrix} a_{11}\dot{u}_r - a_{22}v_r r \\ a_{22}\dot{v}_r + a_{11}u_r r \\ a_{33}\dot{r} + (a_{22} - a_{11})u_r v_r \end{bmatrix}. \tag{7.14}$$

Thus,

$$C_a(\nu_r)\nu_r := \begin{bmatrix} -a_{22}v_r r \\ a_{11}u_r r \\ (a_{22} - a11)u_r v_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & -a_{22}v_r \\ 0 & 0 & a_{11}u_r \\ a_{22}v_r & -a_{11}u_r & 0 \end{bmatrix} \nu_r, \tag{7.15}$$

where the term $(a_{22} - a_{11})u_r v_r$ is the destabilizing Munk moment. Section 4 of Marley (2020*b*) investigates the directional stability and the pivot point, which is the point around which the ship is yawing (Fossen 2021, pp. 179). These are not elaborated on here, however the resulting critical velocity for directional stability, i.e. the maximum surge velocity before the ship becomes directionally unstable, is

$$u_{crit} = \sqrt{\frac{d_{22}d_{33}}{m_{11}(a_{22} - a_{11})}}, \tag{7.16}$$

which is used to determine the desired surge velocity u_d presented in Table 7.1.

The time constant matrices yielding the controller gain matrices K_1 and K_2 are

$$T_1 = \text{diag}([T_{1,x} \ \ T_{1,y} \ \ T_{1,\psi}]) = \begin{bmatrix} 30 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 15 \end{bmatrix}, \tag{7.17}$$

$$T_2 = \text{diag}([T_{2,p} \ \ T_{2,y} \ \ T_{2,\psi}]) = \begin{bmatrix} 15 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 12 \end{bmatrix}, \tag{7.18}$$

The integral gain matrix is given as

$$K_I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{7.19}$$

The only non-zero term as the second diagonal term, as the intention of K_I is to compensate for the steady-state error in sway direction.

7.3 Particle model with control barrier function

The simulations presented in this section concern a VS modelled as a particle model with linear velocities in surge and sway as control inputs. To ensure that the VS stays outside of the obstacle boundary, the CBF in Section 4.1 is employed. Two cases are investigated. In the first case, the VS, and hence the OS, stagnates at the obstacle boundary. In the second case, the VS successfully avoids the obstacle, and the OS is able track the trajectory that the VS traces out.

7.3.1 Failed performance of evasive maneuver

This simulation effectively demonstrates a weakness in the VS particle model with the non-hybrid CBF and the optimization based controller. The safe control input for the VS is determined by quadratic programming (QP) through the minimization problem

$$\min_{(U-V)} \left\{ \frac{1}{2}(U-V)^\top \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} (U-V) \right\},$$

where the Q-matrix terms are as presented in Table 7.2. Hence, Q is a positive definite diagonal matrix. Somewhere along the boundary of the obstacle there will be an equilibrium point. With $Q = \text{diag}(7, 1)$ and the obstacle situated on the nominal reference path, the equilibrium points is located at the intersection between the nominal reference path and the obstacle boundary. This results in stagnation for the VS, and therefore the OS, at the obstacle boundary as Figure 7.1 illustrates. Looking at Figure 7.2, this occurs because the safe control input produced by the QP problem is $U = [0 \;\; 0]^\top$. Figures 7.3, 7.4, 7.5, and 7.6 confirm that the OS velocities and control inputs go to zero.

Close to the equilibrium point, the sway control inputs produced are very small, eventually leading to stagnation at the boundary as well. Furthermore, there are initial conditions close to these points where the VS is able to avoid the obstacle, but the maneuver is such that the OS position, p, enters the safety radius of the obstacle. Hence, the VS must have an initial position far enough away from the nominal path at the North-axis to enable the OS to track the VS trajectory, while staying outside the obstacle boundary. This is at a distance of around two meters on either side, approximately. Hence, the following section discusses a simulation with a VS initial position, and thus a straight-line nominal path, of two meters east of the North-axis.

In all the simulation plots with surge and sway velocities, the velocities subscripted r refers to the relative velocities with respect to the current. The velocities subscripted d is $\dot{q}$ rotated by the desired heading angle, which is calculated from the desired course angle after the simulation is completed. Recall that the controller uses desired course angle, not desired heading angle. This results in a steady-state offset between the actual and desired surge and sway velocities. The offset is present throughout the simulations.

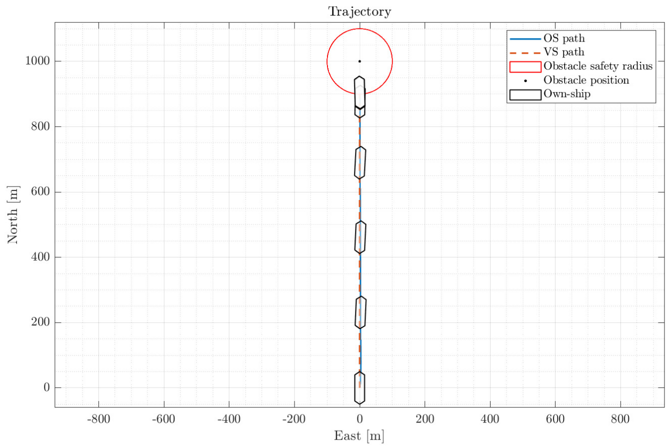


Figure 7.1: VS and OS trajectory with VS as a particle model starting at the North-axis.

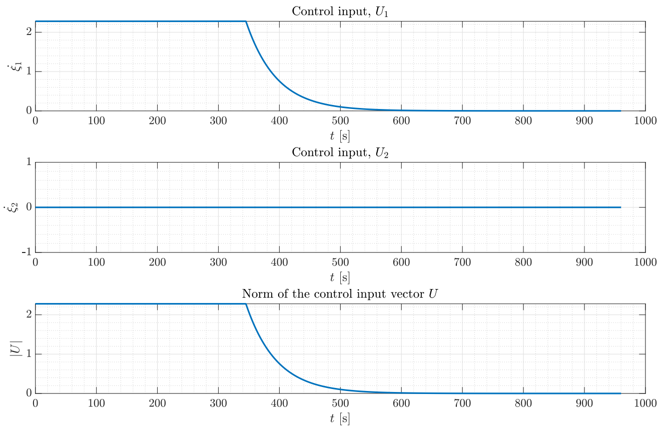


Figure 7.2: VS control inputs with VS as a particle model starting at the North-axis.

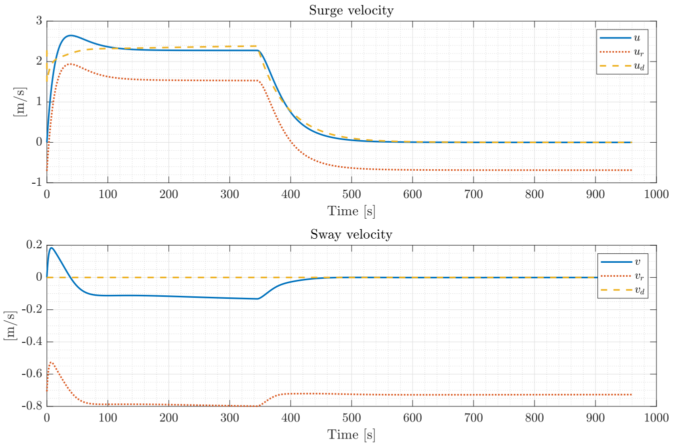


Figure 7.3: Surge and sway velocities with VS as a particle model starting at the North-axis.

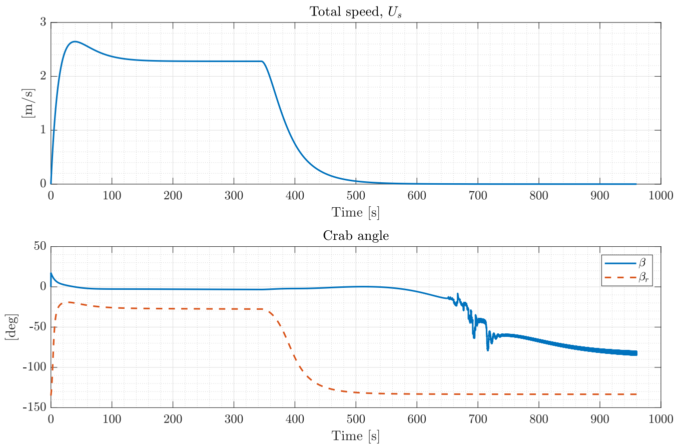


Figure 7.4: Total speed and crab angle with VS as a particle model starting at the North-axis.

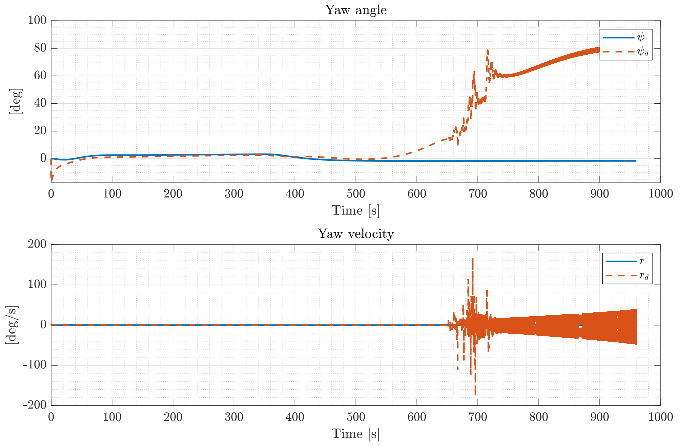


Figure 7.5: Yaw angle and velocity with VS as a particle model starting at the North-axis.

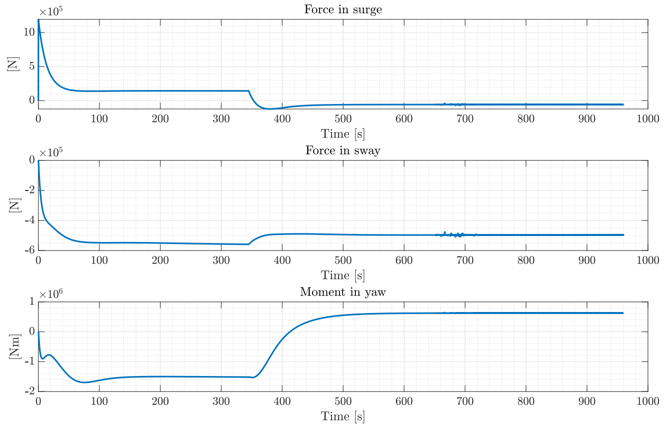


Figure 7.6: OS control inputs with VS as a particle model starting at the North-axis.

7.3.2 Successful performance of evasive maneuver

The VS is placed at a distance of two meters east of the North-axis. Figure 7.7 shows that VS is able to perform a successful evasive maneuver and that the OS is able to track the trajectory. The trajectory traced out by the VS follows the boundary of the obstacle closely. The proximity to the obstacle boundary could be extended by increasing the CBF time constant T_B, however to make comparison between the different implementations easier, the simulation parameters they have in common were kept the same. The particle model uses linear velocities in x- and y-directions as control inputs (Figure 7.8), and thus does not consider yaw rate saturation such as the unicycle models in Sections 7.5 and 7.6. The result is a sharp turn due to sudden changes in the velocities. This sudden change induces a transient response in the system causing an oscillation in the surge and sway velocities, which is observed in Figure 7.9. Figure 7.9 also shows that the OS must reduce its surge velocity by approximately 1 m/s in order to perform the evasive maneuver. This happens at between $t = 350$ s and $t = 500$ s, approximately. Looking at Figure 7.10, it is clear that this reduction in surge speed results in increased tracking errors in this time window. This behaviour is not optimal for a transit mode of operation where the more realistic behaviour would be to keep the surge speed close to constant, and utilize the yaw moment to perform the evasive maneuver. However, with the nature of the particle model, this is not possible.

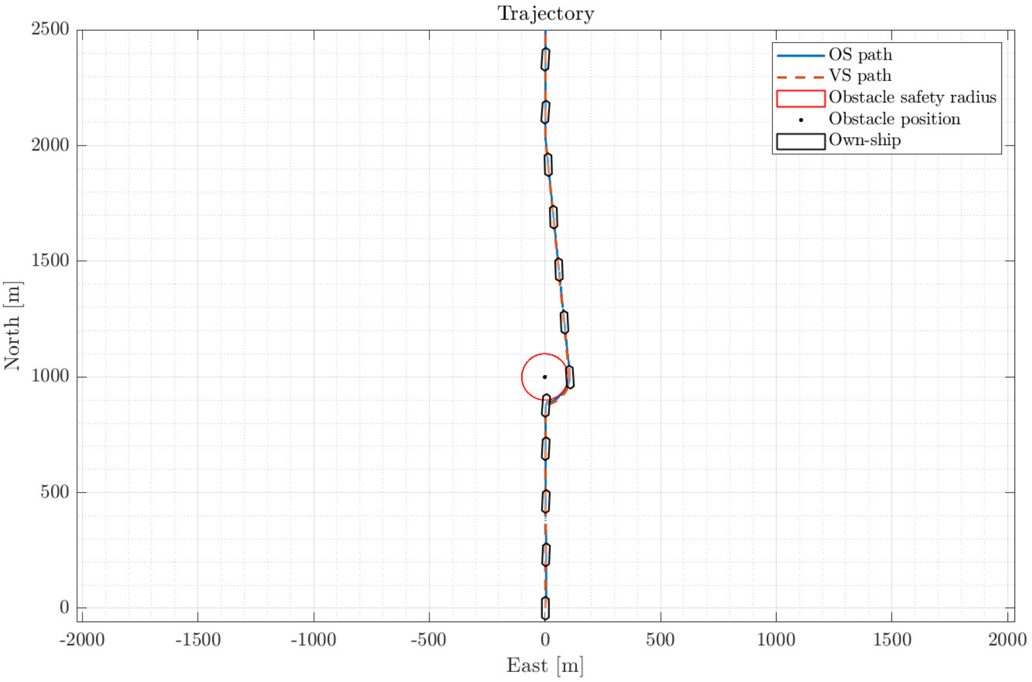


Figure 7.7: OS trajectory with VS as particle model starting east of the North-axis.

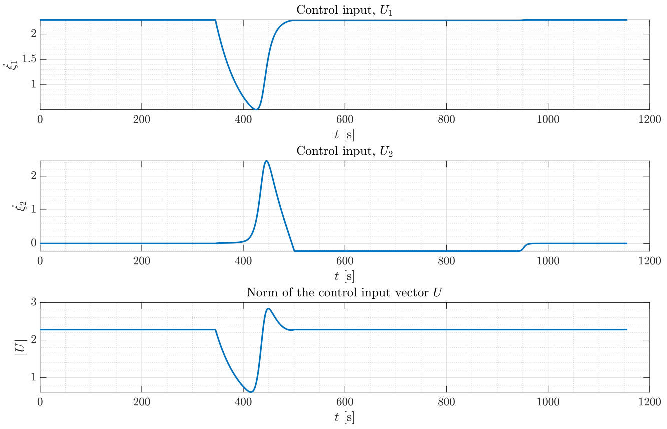


Figure 7.8: VS control inputs with VS as particle model starting east of the North-axis.

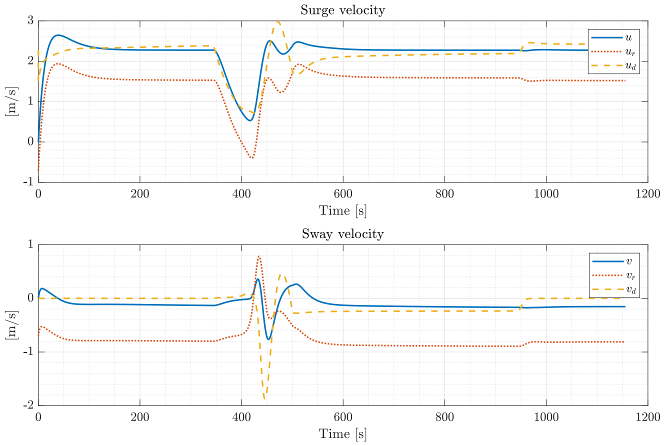


Figure 7.9: Surge and sway velocities with VS as particle model starting east of the North-axis.

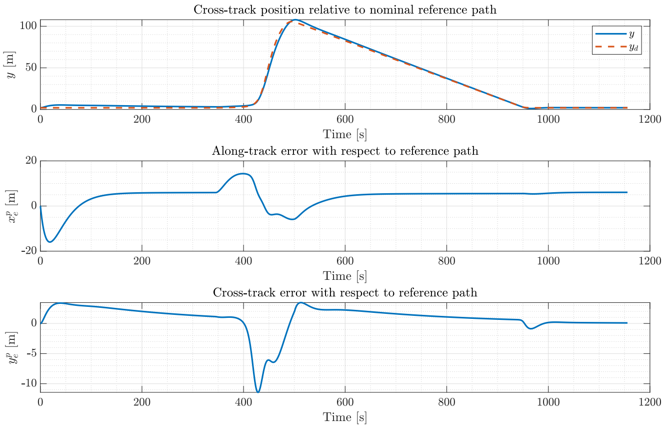


Figure 7.10: Tracking errors with VS as particle model starting east of the North-axis.

In addition to the development of the velocities, the OS also requires a much larger yaw moment, see Figure 7.11, to perform the evasive maneuver compared to what is required by other VS models presented in this thesis. This is due to the large and sudden change in desired course rate, resulting from the LOS guidance dependence on the cross-track error. The OS is only partially able to follow the desired course rate, and therefore the desired heading rate, at the beginning of the evasive maneuver particularly, see Figures 7.12 and 7.13. Hence, the close up of the trajectories in Figure 7.14 shows an overshoot before the OS is able to adjust and turn, which is not desirable. Note that the OS appears to enter the obstacle boundary in Figure 7.14. The dimensions of the plotted vessel polygon is not to scale based on the model. The purpose is to give a visual representation of the vessel along its path with the corresponding heading.

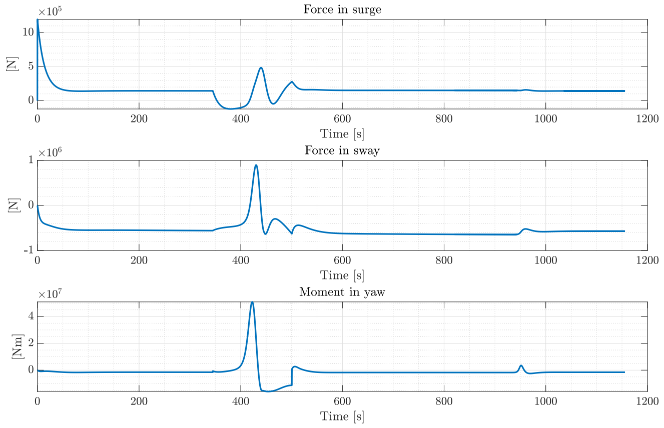


Figure 7.11: OS control inputs with VS as particle model starting east of the North-axis.

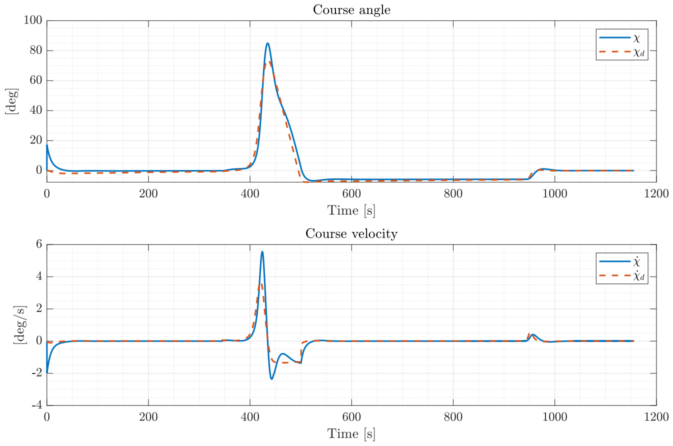


Figure 7.12: Course angle and velocity with VS as particle model starting east of the North-axis.

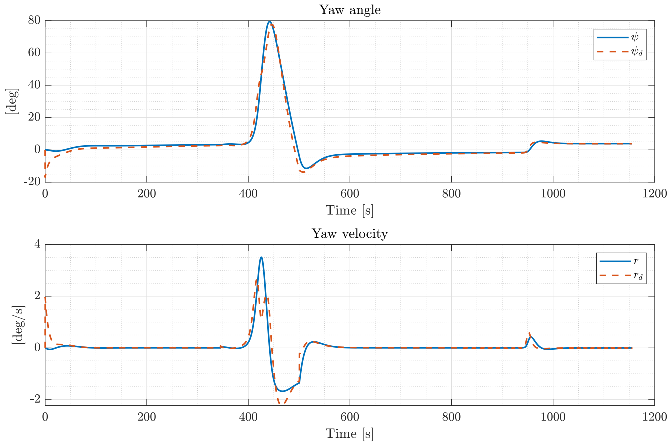


Figure 7.13: Yaw angle and velocity with VS as particle model starting east of the North-axis.

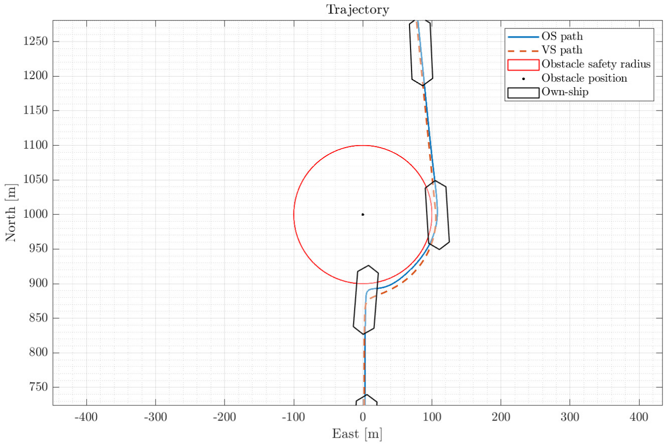


Figure 7.14: Close up of OS and VS trajectories in Figure 7.7.

7.4 Potential function with barrier function and modified obstacle

This simulation models the VS as a potential function. The VS control input is $U := \dot{q} = -\nabla P_m = -(\nabla P_0 + \nabla B(d_m(q)))$, i.e. the gradient of the potential function. $\nabla P_0 = q$, that is, the gradient of P_0 equals the position of the VS. B refers to the barrier function in (4.50). When the VS is far away from the path end, which is in the origin of the NE-frame in this case, $\dot{q}$, and therefore U, is large. Hence, the vessel is not able to track the VS trajectory as it is incapable of producing large enough surge and sway forces. The figures included effectively illustrates this. Figure 7.15 shows the trajectory plot with the potential field lines, Figure 7.16 shows the surge and sway velocities, Figure 7.17 shows the yaw angle and rate, and Figure 7.18 shows the OS control forces and moment. Since the simulations are conducted without limitations on the forces and moments the OS controller produces high control outputs, however despite these, the OS is still not able to track the trajectory. These outputs are not assumed realistic, but included to show the infeasibility of the trajectory traced out by the VS.

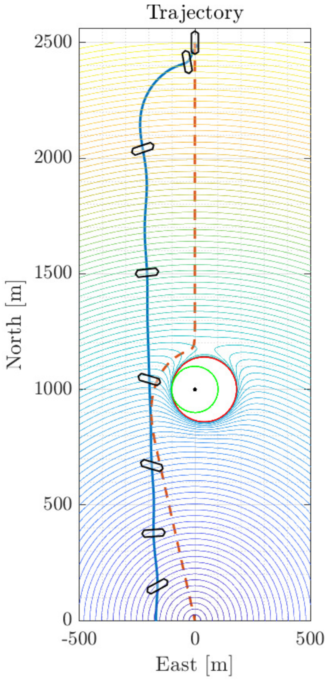


Figure 7.15: OS trajectory with VS as a potential function. The potential field contours show $\sqrt{P(q)}$ to improve resolution around the origin.

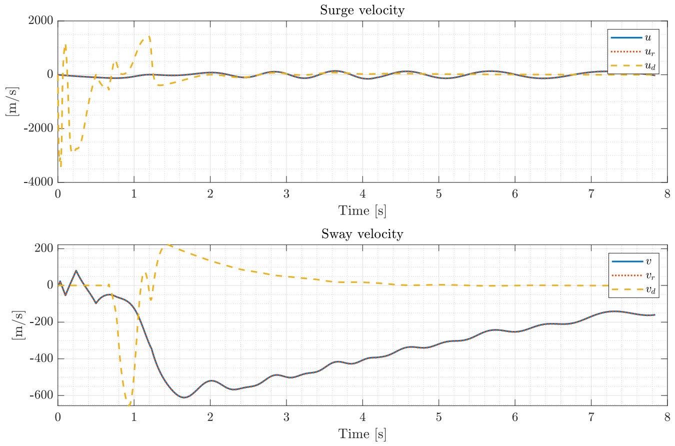


Figure 7.16: Surge and sway velocities with VS a as potential function.

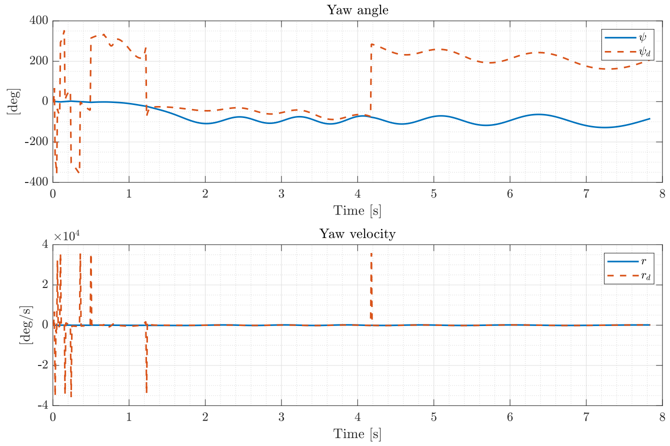


Figure 7.17: Yaw angle and velocity with VS as a potential function.

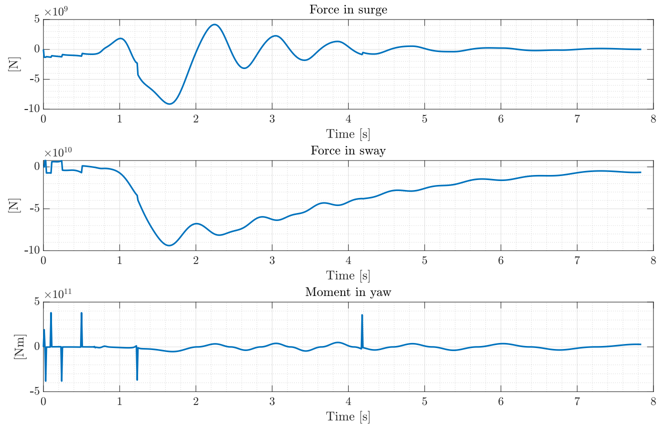


Figure 7.18: OS control inputs with VS as a potential function.

7.5 Unicycle model with synergistic control barrier function

This section concerns a VS modelled as a unicycle model, with angular velocity, ω, and linear acceleration in surge, $\dot{u}_d$, as control inputs. The surge velocity is assumed constant such that $\dot{u}_d = 0$. The angular velocity is therefore the control input determined through QP, as discussed in Section 5.3. Two different nominal control inputs for ω, V_1, are simulated. Method 1 is the LOS guidance method presented in (5.34) to (5.37), and method 2 is $V_1 = -Z_2$ (see (5.33)). Z refers to the unit heading vector. The CBF employed is the synergistic CBF from Section 4.2.

7.5.1 Successful performance of evasive maneuver with method 1

The nominal reference path for the LOS guidance law is along the North-axis. Figures 7.19 and 7.20 indicate that the trajectory traced out is feasible for the OS to track. The tracking errors in Figure 7.21 are also fairly small, particularly the cross-track error. Looking at the velocities u and v in Figure 7.22, some oscillations are observed. This is likely due to a transient response as the VS control inputs, and thus eventually the OS control input, change rapidly when the logic variable indicating the evasive mode, l_0, is toggled. This switches the employed CBF from the non-hybrid CBF, B_0, to the hybrid CBF, B_1, and vice versa. Using black circles, Figure 7.23 shows when l_0 toggles. As the VS is approaching the obstacle, l_0 toggles immediately. Note that during the evasive maneuver, l_0 toggles to 0 for a period of time when the vessel is far enough away to satisfy the switching logic condition. However, it toggles back to 1 during the return to the nominal path, before settling on $l_0 = 0$ after the obstacle is passed. This is the source of the sudden changes in the control inputs as the hybrid CBF utilized switches back and forth between B_0 and B_1 during the evasive maneuver. It induces some sharp changes in the OS control input, see Figure 7.24, particularly in the yaw moment (which is reflected in the desired yaw rate in Figure 7.25) and sway force. However, the yaw moment required is much lower than in the previous simulations, which makes the sudden increases or decreases more feasible, although not optimal.

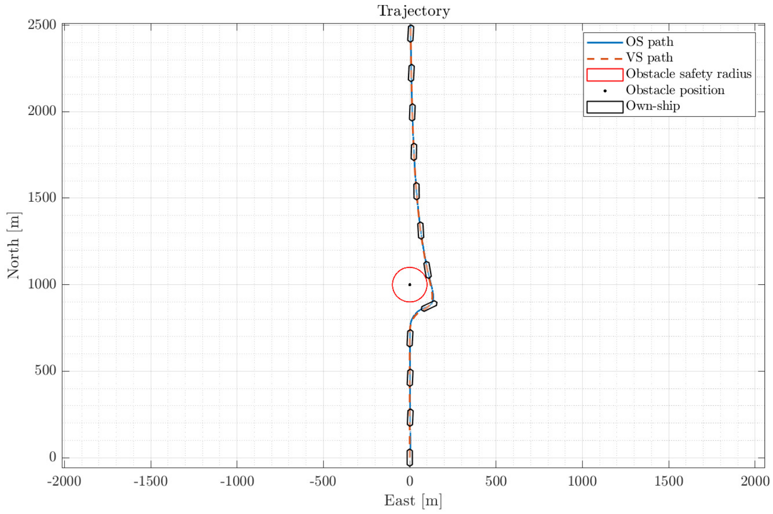


Figure 7.19: OS trajectory with VS as a unicycle model with a synergistic CBF and LOS guidance.

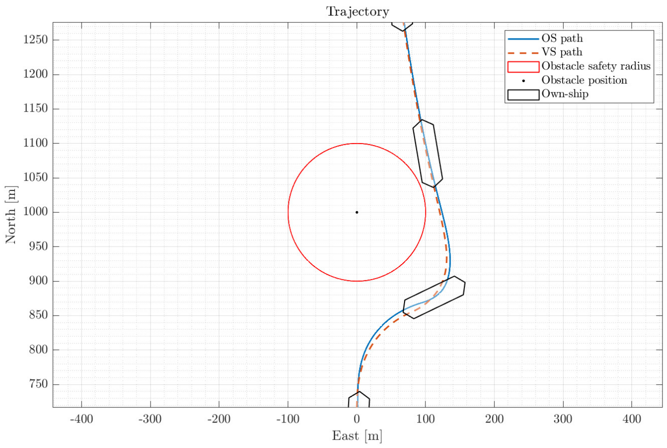


Figure 7.20: Close up of OS and VS trajectories in Figure 7.19.

.

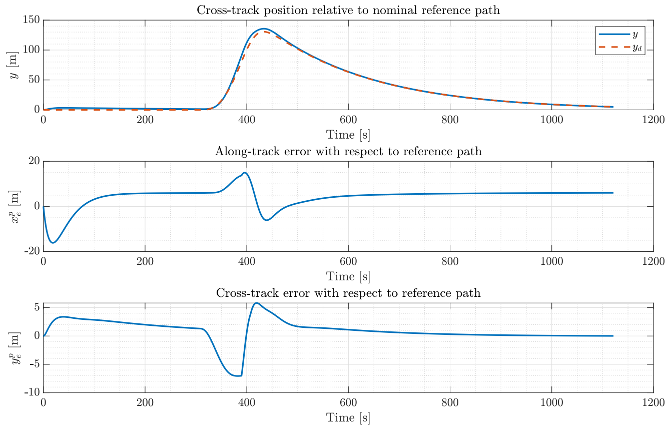


Figure 7.21: Tracking errors with VS as a unicycle model with a synergistic CBF and LOS guidance.

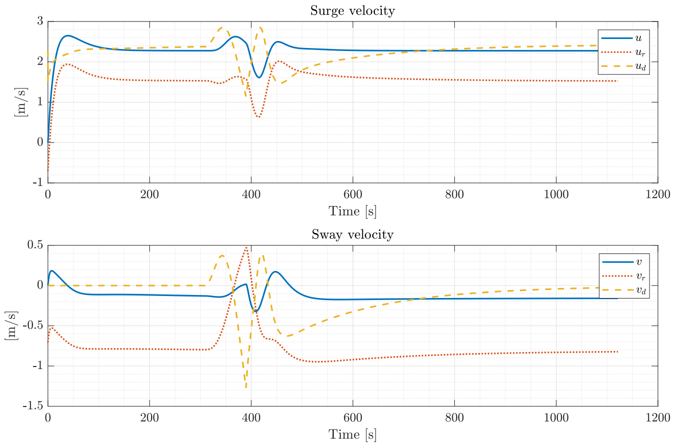


Figure 7.22: Surge and sway velocities with VS as a unicycle model with a synergistic CBF and LOS guidance.

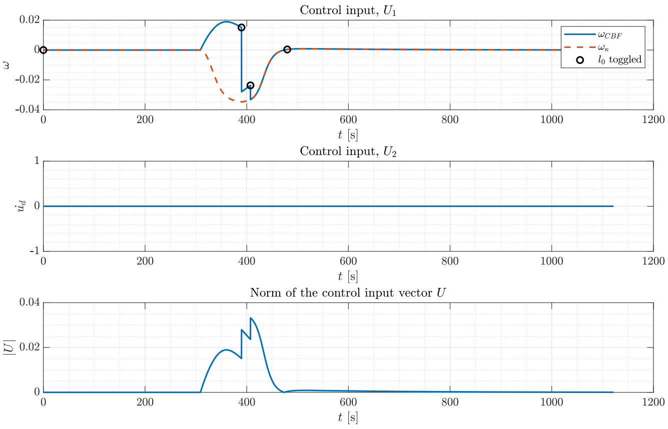


Figure 7.23: VS control inputs with VS as a unicycle model with a synergistic CBF and LOS guidance.

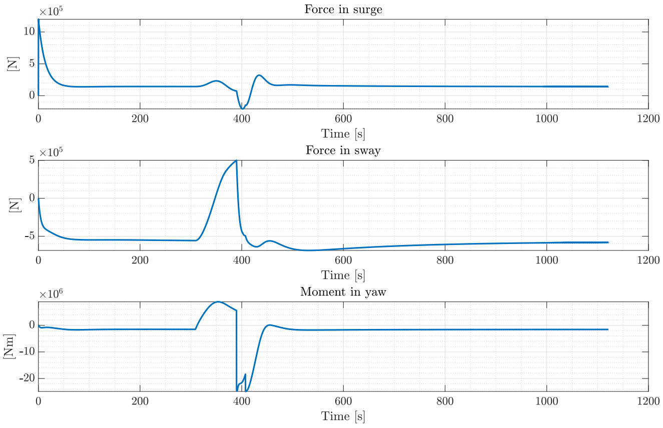


Figure 7.24: OS control inputs with VS as a unicycle model with a synergistic CBF and LOS guidance.

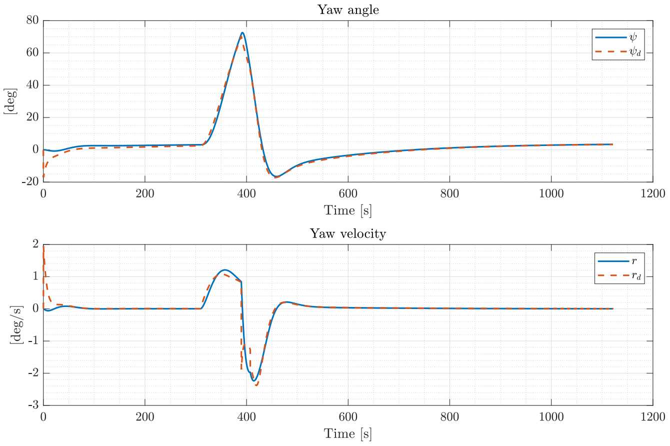


Figure 7.25: Yaw angle and velocity with VS as a unicycle model with a synergistic CBF and LOS guidance.

7.5.2 Successful performance of evasive maneuver with method 2

The following figures display a very similar behaviour as the previous subsection. The nominal control input $V_1 = -Z_2$ is not plotted in the figure with the VS control inputs (Figure 7.30) as in Section 7.5.1, because V_1 is much larger than U_1. Thus U_1 is not discernible. Similar to the LOS guidance law simulations, the VS control input changes rapidly during the evasive maneuver due to deactivation and activation of B_1 by l_0. Comparing the OS control input figures, Figure 7.24 and Figure 7.31, the first figure produces the smoothest control inputs. Comparing Figures 7.23 and 7.30, which contain the VS control inputs, it is clear that Figure 7.23 shows the smoothest control inputs as well. Thus, it is evident that the LOS method performs the best out of the two methods given the conditions in these simulations. However, there is not a great difference. Therefore, whether the desire is to return to the nominal reference path for the VS, or not, must be considered before making a judgement of which method is the most suitable.

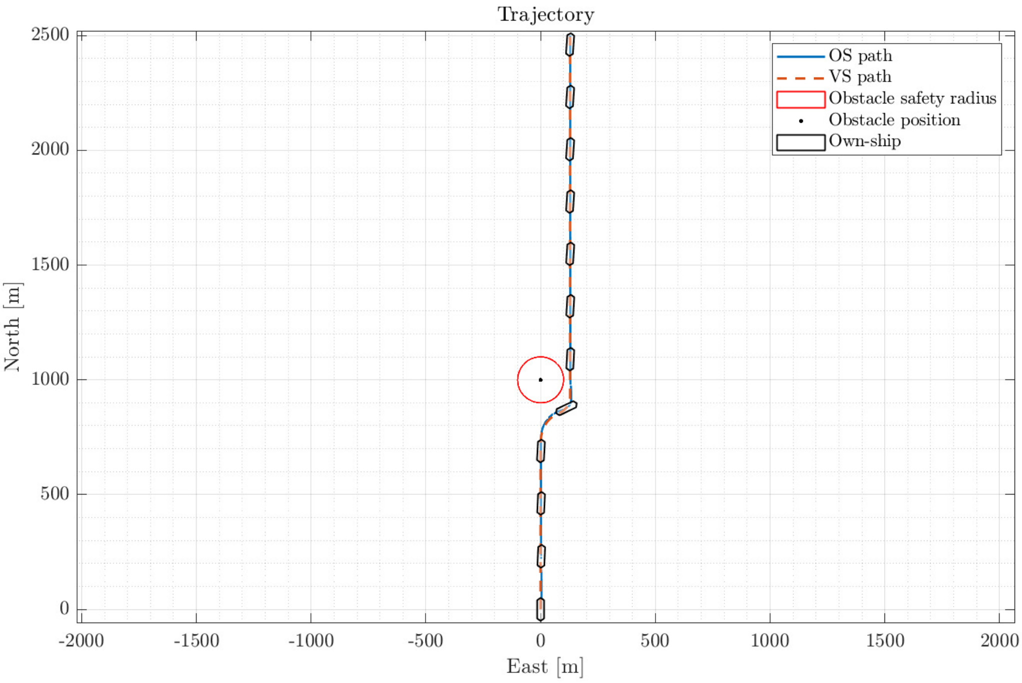


Figure 7.26: OS trajectory with VS as a unicycle model with a synergistic CBF and $V_1 = -Z_2$.

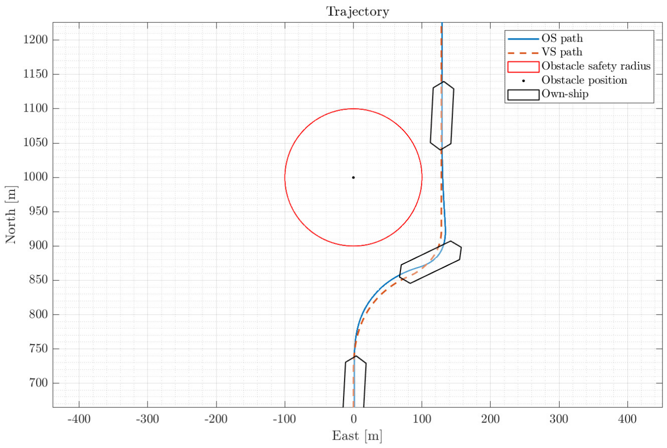


Figure 7.27: Close up of OS and VS trajectories in Figure 7.26.

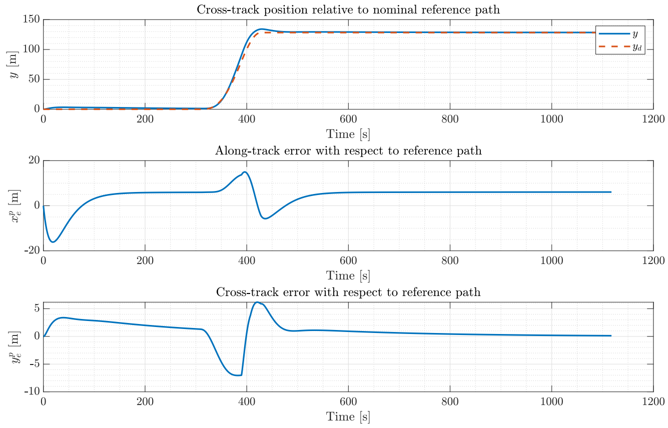


Figure 7.28: Tracking errors with VS as a unicycle model with a synergistic CBF and $V_1 = -Z_2$.

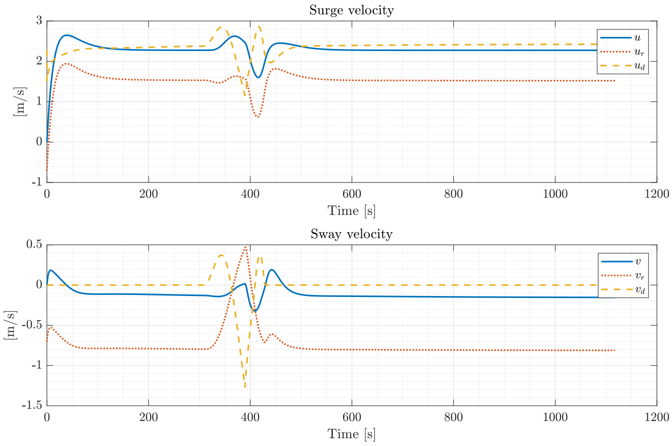


Figure 7.29: Surge and sway velocities with VS as unicycle model with a synergistic CBF and $V_1 = -Z_2$.

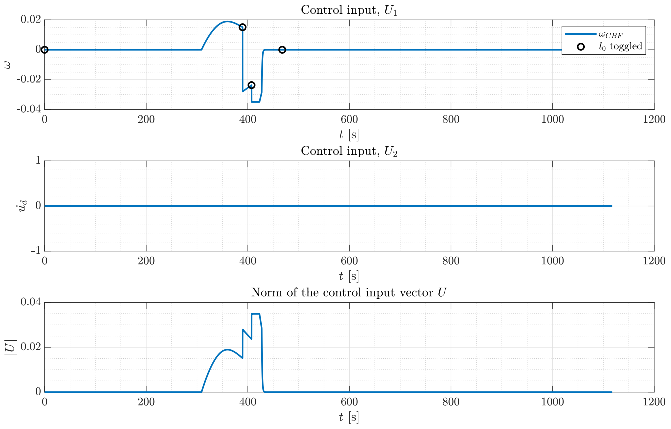


Figure 7.30: VS control inputs with VS as a unicycle model with a synergistic CBF and $V_1 = -Z_2$.

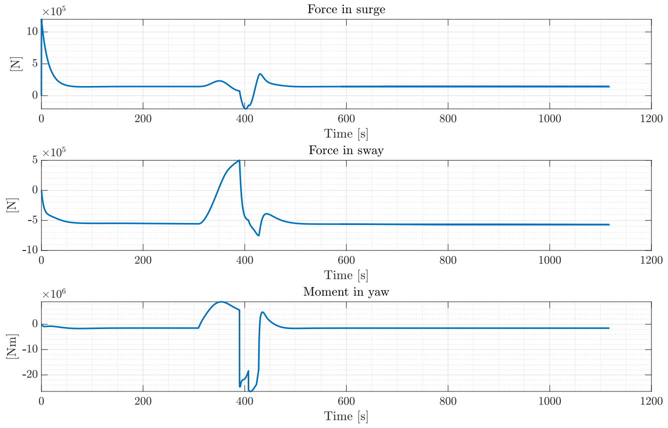


Figure 7.31: OS control inputs with VS as a unicycle model with a synergistic CBF and $V_1 = -Z_2$.

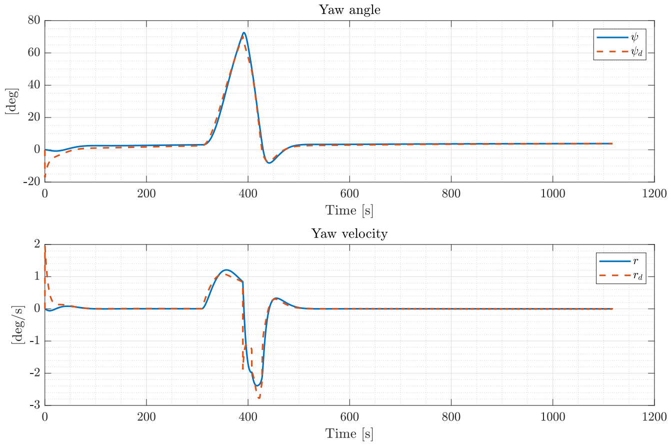


Figure 7.32: Yaw angle and velocity with VS as a unicycle model with a synergistic CBF and $V_1 = -Z_2$.

7.6 Unicycle model with control barrier function and modified obstacle

The simulations in Section 7.4 use a modified obstacle to induce a preferred turning direction for the VS. However, as discussed, the potential function implemented does not produce a feasible trajectory for the OS to track. The simulations conducted in this section, on the other hand, use the modified obstacle in combination with the non-hybrid CBF B_0 from the simulations in Section 7.5. As in the previous section, the VS is modelled as a unicycle model, with angular velocity ω and linear acceleration in surge $\dot{u}_d$ as control inputs. The surge velocity is assumed constant and thus $\dot{u}_d = 0$. The angular velocity is the control input determined through QP, and two different nominal control inputs are simulated. The first method determines the nominal angular velocity though the LOS guidance method presented in (5.34) to (5.37). The second method determines the nominal angular velocity through $V_1 = -Z_2$ where Z refers to the unit heading vector.

7.6.1 Successful performance of evasive maneuver with method 1

The VS nominal reference path is along the North-axis. Looking at the trajectory plots in Figure 7.33 and 7.34, this trajectory seems to be the most feasible for the OS to track compared to the previous simulations in Section 7.3 to 7.5. The tracking errors are confirmed to be small in Figure 7.35. Furthermore, the VS and OS control inputs displayed in Figure 7.36 and 7.37 respectively, are smooth, resulting in smooth surge and sway velocities in Figure 7.45, and yaw rate in Figure 7.39. In addition, Figure 7.37 shows that the yaw moment is utilized in a larger degree and the sway force in a lesser degree to perform the evasive maneuver, compared to the previous simulations. This utilization of sway force and yaw moment is more realistic with regards to real life situations in the transit phase of a voyage.

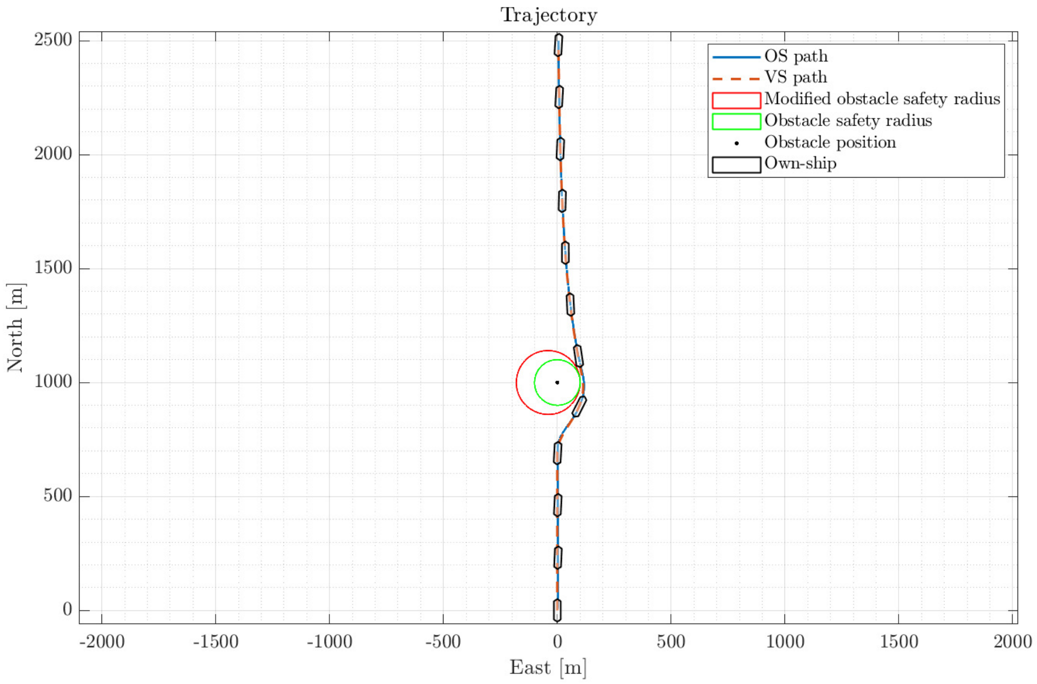


Figure 7.33: OS trajectory with VS as a unicycle model with a modified obstacle and LOS guidance.

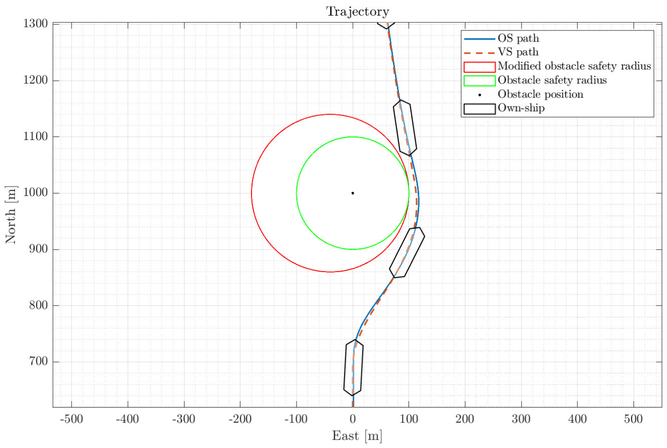


Figure 7.34: Close up of OS and VS trajectories in Figure 7.33.

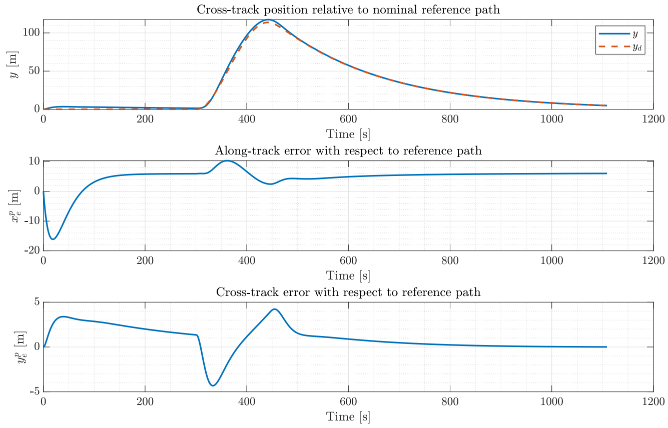


Figure 7.35: Tracking errors with VS as a unicycle model with a modified obstacle and LOS guidance.

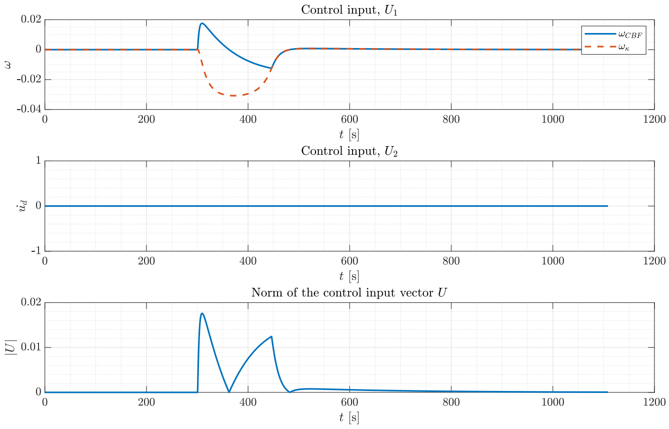


Figure 7.36: VS control inputs with VS as a unicycle model with a modified obstacle and LOS guidance.

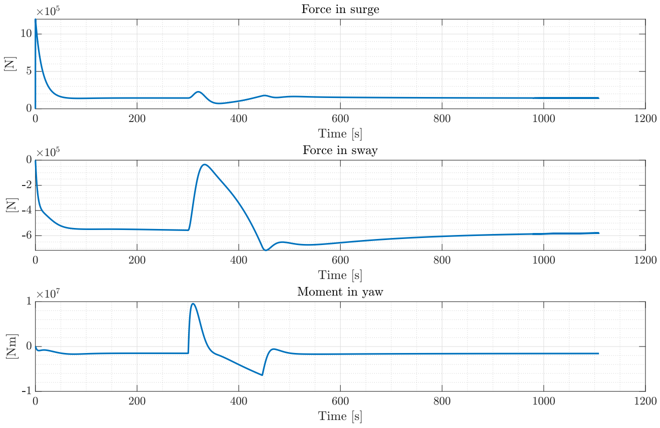


Figure 7.37: OS control inputs with VS as a unicycle model with a modified obstacle and LOS guidance.

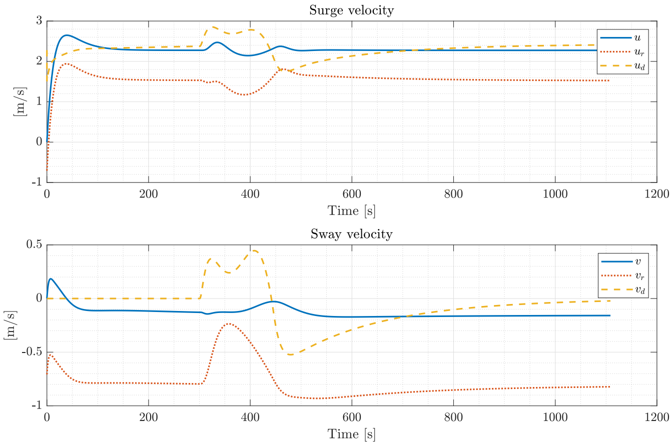


Figure 7.38: Surge and sway velocities with VS as a unicycle model with a modified obstacle and LOS guidance.

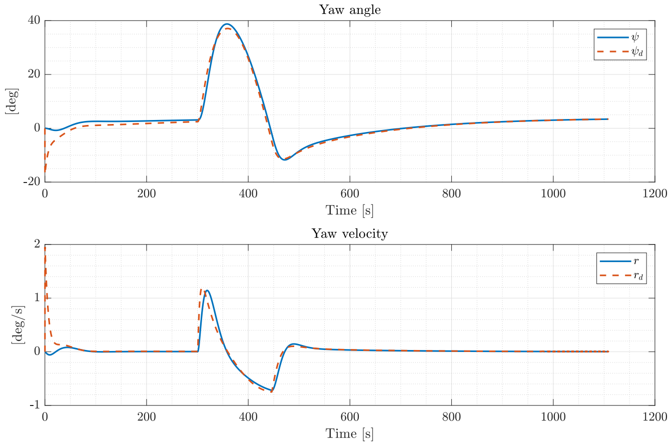


Figure 7.39: Yaw angle and velocity with VS as a unicycle model with a modified obstacle and LOS guidance.

7.6.2 Successful performance of evasive maneuver with method 2

The simulation in this section shows almost identical results compared to Section 7.6.1. The major difference between the two, is whether or not the goal is to return to the nominal reference path along the North-axis after the evasive maneuver is performed. If a return to the nominal path is desired, the implementation with LOS guidance is suitable. However, if the goal is to continue straight ahead after the obstacle is passed, the implementation with $V_1 = -Z_2$ seems to perform well. Since both methods in this section seems feasible, further discussion of this simulation is included in Section 8.1, where it is compared to the all the simulations conducted in the thesis.

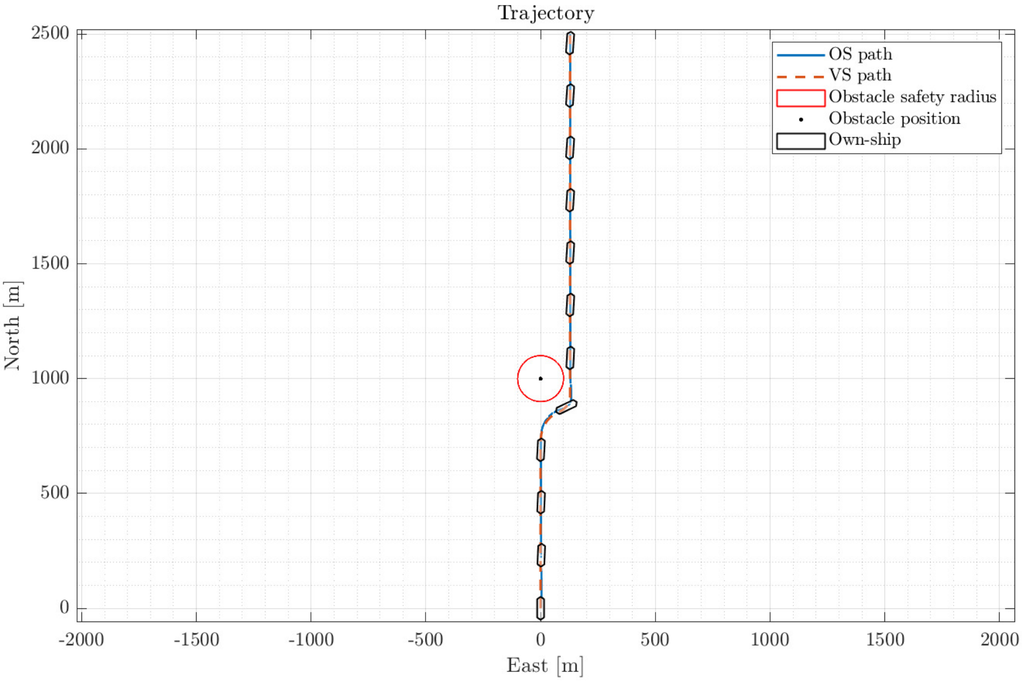


Figure 7.40: OS trajectory with VS as a unicycle model with a modified obstacle and $V_1 = -Z_2$.

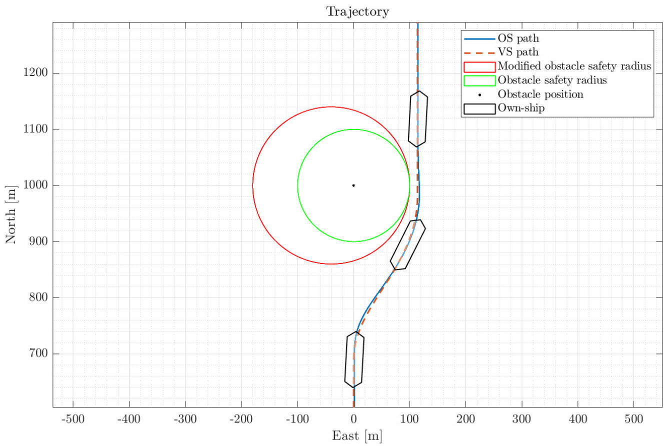


Figure 7.41: Close up of OS and VS trajectories in Figure 7.40.

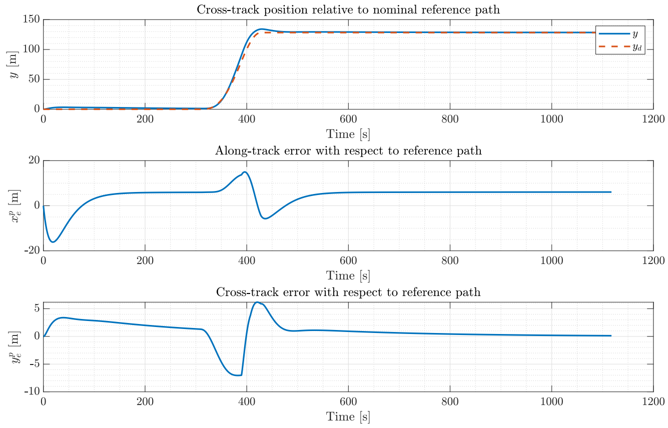


Figure 7.42: Tracking errors with VS as a unicycle model with a modified obstacle and $V_1 = -Z_2$.

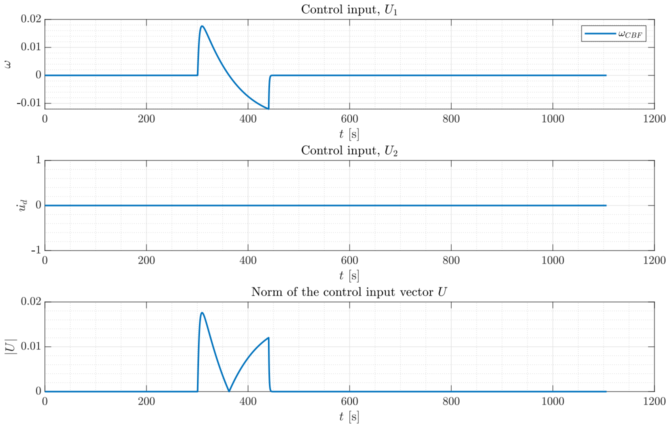


Figure 7.43: VS control inputs with VS as a unicycle model with modified obstacle.

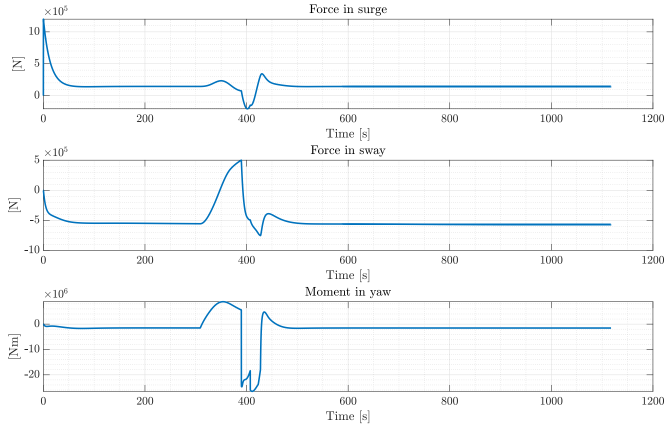


Figure 7.44: OS control inputs with VS as a unicycle model with a modified obstacle and $V_1 = -Z_2$.

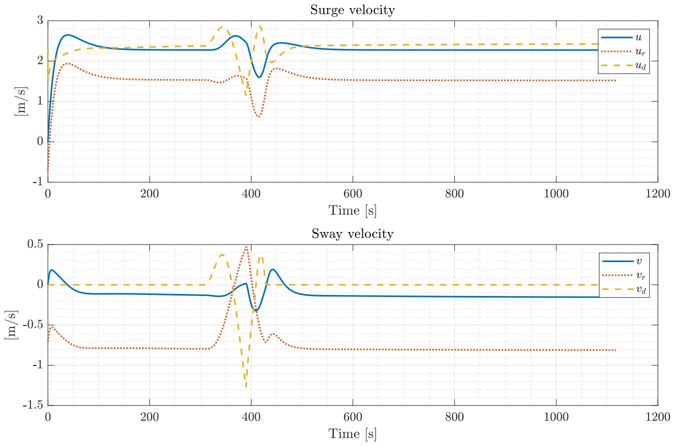


Figure 7.45: Surge and sway velocities with VS as a unicycle model with a modified obstacle and $V_1 = -Z_2$.

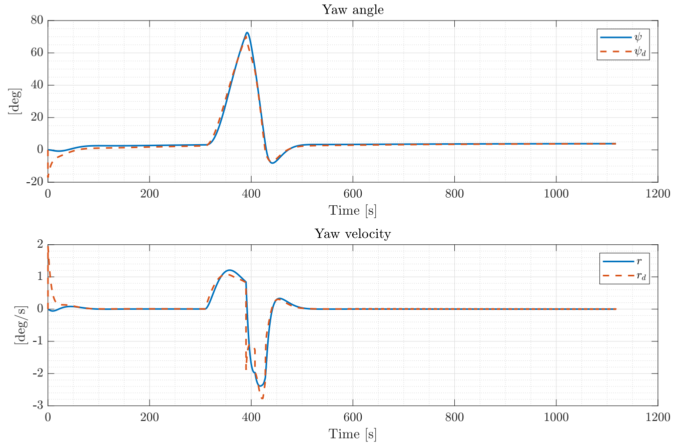


Figure 7.46: Yaw angle and velocity with VS as a unicycle model with a modified obstacle and $V_1 = -Z_2$.

Chapter 8

Discussion, conclusions, and further work

8.1 Discussion and conclusions

An introduction to path parametrization using a two-dimensional path variable ξ is given. This is used to define the position of a VS, $q = p_d(\xi)$. Three different VS models are presented. These are a particle model, a unicycle model, and a potential function. The VS traces out a trajectory for the OS to track. Along the VS straight-line nominal reference path, an obstacle is encountered. The VS must perform an evasive maneuver in order to allow the OS to avoid the obstacle. This is done by employing a barrier function or a control barrier function, which should ensure that the VS is able to evade the obstacle. Four different methods of employing barrier functions with assigned VS models are presented. The thesis aims to investigate the feasibility of the trajectories during obstacle maneuvers. In order to answer this research question, the simulations in Chapter 7 were conducted.

The first simulation concerning a VS particle model has some challenges. The particle model does not consider yaw dynamics, and thus utilizes surge and sway velocity to perform the evasive maneuver. The OS is not able to replicate these sudden changes, causing an overshoot in the initial phase. To perform the maneuver initiated by the VS, the OS requires a large yaw moment as the tracking errors, particularly the cross-track error, increases. The LOS course reference depends on the cross-track error, inducing a sharp change in the yaw rate. The system also experiences sudden changes in surge and sway velocities, which causes transients. The result is some oscillatory behaviour in the velocities, and a significant decrease in the surge velocity. Hence, although the OS is able to track the trajectory, i.e. it is feasible under the conditions in the simulations, it is not optimal with respect to real life operations. Furthermore, there is an equilibrium point along the obstacle boundary, which can cause the vessels to stagnate.

The second simulation conducted uses a potential function to model the VS. This produced an unfeasible trajectory as the VS speed in x- and y-directions are too large for the OS to be able to track the trajectory. In addition, the potential function guides the VS to the origin. Hence, adjustments have to be made to generalize the end of the path. However, such an adjustment was not made as the trajectory is not feasible, and therefore the other methods are of greater interest.

The unicycle model with a synergistic CBF performs fairly well. The path appears smoother and the OS control forces and moment required to successfully perform trajectory tracking are much smaller than in the foregoing simulations. Due to toggling back and forth between evasive and non-evasive mode, there are some transient response observed in the system. This occurs because the switching logic condition for l_0 is satisfied as the OS is far enough away from the boundary at a point during the evasive maneuver. During the return to the nominal path, B_1 is once again activated. The two switches in l_0 causes U_1 to change rapidly as B_0 and B_1 produces different control inputs in the QP based controller. These switches can be removed by decreasing the time constant T_B as this reduces the distance to the obstacle boundary before the evasive maneuver is performed. Intuitively, this prevents the VS from journeying far enough away from the boundary to toggle l_0. However, it reduces the margin the OS has to the boundary. In addition, this path closer to the boundary requires greater forces and moments from the OS to track the VS.

The second method with this VS model and CBF uses $V_1 = -Z_2$. It displays very similar results compared to the LOS method. The difference is that the VS controller, and thus the OS controller, changes control input even more rapidly as l_0 is toggled. It is not a great difference with regards to the OS control inputs, hence the most significant consideration concerning these two methods is whether or not a return to the VS nominal reference path is desired. If a return is desired, LOS guidance of the VS is the most suitable method.

The simulations conducted with a VS unicycle model, a modified obstacle, and a non-hybrid CBF, seems to perform the best of the methods implemented. This is the case for both of the VS nominal controllers, where V_1 is based on LOS guidance in the first method, and $V_1 = -Z_2$ in the second method. Similar to the synergistic CBF simulations, the most suitable of these two methods is determined by whether the desire is to return to the VS nominal reference path along the North-axis after the evasive maneuver. Looking at the figures, these simulations has the smallest tracking errors, the smoothest control inputs and resulting velocities, and hence the conclusion is that these are the most feasible trajectories for the OS to track. In addition, the VS path has softer turns allowing the simulated OS to perform more realistically with regards to transit phase maneuvering around obstacles. That is, the surge speed is close to constant and the evasive maneuver is to a larger degree performed using change in yaw angle, rather than force in sway.

8.2 Further work

A clear point for further work is to extend the OS vessel model used in the simulations. The vessel model implemented is the maneuvering model of an idealized ship from Marley (2020*b*). The vessel is port-starboard and fore-aft symmetric, hence the mass and damping matrices are diagonal. However, that is not the case for most real life vessels, which are usually only port-starboard symmetric. In addition, the OS is only exposed to a constant current. A vessel in transit will usually encounter various currents, as well as wave and wind forces. Such conditions should be included to create a more realistic environment. Furthermore, the GNC system does not include an observer, or a control allocation with limits on forces and yaw moment. It is assumed that the required measurements are available and free of noise. However, that is not realistic. In addition, a more accurate conclusion with regards to trajectory feasibility would be possible with a control allocation and limits on the thrusters. Although such an extension is outside the scope of the thesis, it would be an interesting and significant addition to the GNC system.

With regards to the VS models and (C)BFs there are improvements to note for further work. The particle model simulation has the issue that there is an equilibrium point at the intersection between the straight-line nominal path when the obstacle is located on the path. A solution could be to create a hybrid Q-matrix in the QP problem. A hybrid Q-matrix means that depending on the location of the VS relative to the obstacle, Q is shifted. Hence, the equilibrium point along the obstacle boundary is shifted. Q should be tuned such that the equilibrium point moves to a sufficient distance away from the point where the VS will make an evasive maneuver. That is, the equilibrium point will not cause stagnation at the boundary.

Another issue with the particle model is that the transient behaviour the system experiences. This is a result of the fact that the control inputs are the linear velocities in surge and sway, i.e. it is a first order particle model. Implementing a second order particle model with linear accelerations as control inputs could create smoother VS control inputs, which reduce the transient response.

The time constant, T_B, used to activate the CBFs through $\alpha(B)$ was kept constant throughout the simulations. However, it was observed that tuning this variable to each simulation could have improved the results for some of the simulations. By increasing T_B in the particle model simulation, the evasive maneuver is initiated at an earlier stage, preventing the OS from entering the obstacle boundary at any time if the evasive maneuver is performed successfully by the VS. The synergistic CBF simulation, on the other hand, yielded results resembling the results of the simulations with the non-hybrid CBF and a modified obstacle, if T_B was reduced. This initiated the evasive maneuver at a later stage, hence the VS did not journey far enough away from the obstacle boundary to toggle l_0 out of the evasive mode. However, the required OS forces and moments to track this trajectory were much higher. Thus, the conclusion that the simulations with the non-hybrid CBF and a modified obstacle performs the best would have been unchanged. One can also observe that the CBFs in (4.1) and (4.19) are very similar. The latter has an additional term with the relative velocity between the obstacle and the VS, weighted by a time constant t_0. By including such a term in (4.19), a similar effect as increasing T_B is achieved, and so this could be considered as well.

These simulations only considers one static obstacle. The inclusion of more than one obstacle, where some are dynamic, is possible. By including dynamic obstacles such as other vessels, COLREGS (International Regulations for Preventing Collisions at Sea, (IMO 1972)) could be included in the CBFs as well to ensure that the OS is in line with the regulations.

This thesis focused on trajectory tracking, i.e. there is no feedback from the OS's states to the evolution of the path variable ξ. However, this could be extended to look at maneuvering designs using gradient-type update laws which provide such feedback. By implementing a gradient update law influencing the evolution of ξ, issues regarding the OS's ability to keep up with ξ, such as the case with the potential function, could be reduced.

Bibliography

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K. & Tabuada, P. (2019), Control barrier functions: Theory and applications, *in* '2019 18th European Control Conference (ECC)', pp. 3420–3431.

Artstein, Z. (1983), 'Stabilization with relaxed controls', *Nonlinear Analysis: Theory, Methods & Applications* **7**(11), 1163–1173.

Basso, E. A., Thyri, E. H., Pettersen, K. Y., Breivik, M. & Skjetne, R. (2020), Safety-critical control of autonomous surface vehicles in the presence of ocean currents, *in* 'IEEE Conference on Control Technology and Applications', Vol. 4.

Breivik, M. & Fossen, T. I. (2009), Guidance laws for autonomous underwater vehicles, *in* A. V. Inzartsev, ed., 'Intelligent Underwater Vehicles', I-Tech Education and Publishing, chapter 4.

Brekke, E. F., Wilthil, E. F., Eriksen, B.-O. H., Kufoalor, D. K. M., Helgesen, . K., Hagen, I. B., Breivik, M. & Johansen, T. A. (2019), The autosea project: Developing closed-loop target tracking and collision avoidance systems, *in* 'Journal of Physics', 1357.

Byrnes, C. I. & Isidori, A. (1989), 'New results and examples in nonlinear feedback stabilization', *Systems & Control Letters* **12**(5), 437–442.

Faltinsen, O. M. (1993), *Sea loads on ships and offshore structures*, Cambridge University Press, Cambridge, UK.

Fleischer, C. (2020), Optimal path-planning on a bio-inspired neural network guidance model for autonomous surface vessels, Master's thesis, Norwegian University of Science and Technology, Faculty of Engineering, Department of Marine Technology, NO-7491 Trondheim, Norway.

Fossen, T. I. (2011), *Handbook of Marine Craft Hydrodynamics and Motion Control*, 1 edn, John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom.

Fossen, T. I. (2020), 'Chapter 2 - kinematics'. Lecture notes. TTK4190 Guidance and Control of Vehicles, Norwegian University of Science and Technology.

Fossen, T. I. (2021), *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2 edn, John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom.

Goebel, R., Sanfelice, R. G. & Teel, A. R. (2012), *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*, Princeton University Press, Princeton (NJ).

Healey, A. & Lienard, D. (1993), 'Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles', *IEEE Journal of Oceanic Engineering* **18**(3), 327–339.

Huang, Y., Chen, L., Chen, P., Negenborn, R. R. & van Gelder, P. H. A. J. M. (2020), 'Ship collision avoidance methods: State-of-the-art', *Safety Science* **121**, 451–472.

IMO (1972), *COLREG : Convention on the International Regulations for Preventing Collisions at Sea, 1972*, International Maritime Organization, London.

IMO (n.d.), 'Autonomous shipping'. Available at: `https://www.imo.org/en/MediaCentre/HotTopics/Pages/Autonomous-shipping.aspx` [Accessed: May 12, 2021].

Jensen, J. S. (2020), Reactive path-planning for autonomous harbor maneuvering, Master's thesis, Norwegian University of Science and Technology, Faculty of Engineering, Department of Marine Technology, NO-7491 Trondheim, Norway.

Khalil, H. K. (2015), *Nonlinear Systems*, 3 edn, Pearson Education Limited, Edinburgh Gate, Harlow, Essex CM20 2JE, United Kingdom.

Koditschek, D. E. (1987), Adaptive techniques for mechanical systems, *in* 'Fifth Yale Workshop on Applications of Adaptive Systems Theory', pp. 259–265.

Krstic, M., Kanellakopoulos, I. & Kokotovic, P. V. (1995), *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc, 605 Third Avenue, New York, New York, United States.

Marley, M. (2020*a*), 'Robust obstacle avoidance for autonomous vessels'. Technical Note. Norwegian University of Science and Technology (NTNU).

Marley, M. (2020*b*), 'Technical note: Maneuvering model of an idealized ship'. Norwegian University of Science and Technology (NTNU).

Marley, M. (2021*a*), 'Technical note: A synergistic cbf design for ugas in presence of obstacles.'. Version Revision B. Norwegian University of Science and Technology (NTNU).

Marley, M. (2021*b*), 'Technical note: Maneuvering control design using two path variables'. Version Revision B. Norwegian University of Science and Technology (NTNU).

Marley, M., Skjetne, R., Breivik, M. & Fleischer, C. (2020), 'A hybrid kinematic controller for resilient obstacle avoidance of autonomous ships', *IOP Conference Series: Materials Science and Engineering* **929**, 012022.

Marley, M., Skjetne, R. & Teel, A. R. (2020), A kinematic hybrid feedback controller on the unit circle suitable for orientation control of ships, *in* '2020 59th IEEE Conference on Decision and Control (CDC)', pp. 1523–1529.

Marley, M., Skjetne, R. & Teel, A. R. (2021), Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles, *in* '2021 American Control Conference (ACC) New Orleans, USA, May 25-28, 2021'.

MathWorks (n.d.*a*), 'quadprog'.
URL: *https://se.mathworks.com/help/optim/ug/quadprog.html*

MathWorks (n.d.*b*), 'Quadratic programming'.
URL: *https://se.mathworks.com/discovery/quadratic-programming.html*

McGookin, E. W., Murray-Smith, D. J., Li, Y. & Fossen, T. I. (2000*a*), 'Experimental results of supply ship autopilot optimisation using genetic algorithms', *Transactions of the Institute of Measurement and Control* **22**(2), 141–178.

McGookin, E. W., Murray-Smith, D. J., Li, Y. & Fossen, T. I. (2000*b*), 'Ship steering control system optimisation using genetic algorithms', *Control Engineering Practice* **8**(4), 429–443.

Nagumo, M. (1942), Über die lage der integralkurven gewöhnlicher differentialgleichungen, *in* 'Proceedings of the Physico-Mathematical Society of Japan', Vol. 24 of *3*, pp. 551–559.

Prajna, S. (2006), 'Barrier certificates for nonlinear model validation', *Automatica* **42**(1), 117–126.

Prajna, S. & Jadbabaie, A. (2004), Safety verification of hybrid systems using barrier certificates, *in* R. Alur & G. J. Pappas, eds, 'Hybrid Systems: Computation and Control', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 477–492.

Sanfelice, R., Messina, M., Emre Tuna, S. & Teel, A. (2006), Robust hybrid controllers for continuous-time systems with applications to obstacle avoidance and regulation to disconnected set of points, *in* '2006 American Control Conference', pp. 6 pp.–.

Skjetne, R. (2005), The Maneuvering Problem, PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology, Mathematics, and Electrical Engineering, Department of Engineering Cybernetics, NO-7491 Trondheim, Norway.

Skjetne, R. (2020*a*), 'Marine control systems II, Lecture 8: Maneuvering control design'. Lecture notes. TMR4243 Marine Control Systems II, Norwegian University of Science and Technology.

Skjetne, R. (2020*b*), 'Technical note: Maneuvering-based guidance design for dynamic positioning'. Version Revision B (Draft). Norwegian University of Science and Technology (NTNU).

SNAME (1950), *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*, Technical and Research Bulletin 1-5, The Society of Naval Architects and Marine Engineers, pp. 1–15.

Sontag, E. (1983), 'A lyapunov-like characterization of asymptotic controllability', *Siam Journal on Control and Optimization* **21**, 461–471.

Sontag, E. D. & Sussmann, H. J. (1988), 'Further comments on the stabilizability of the angular velocity of a rigid body', *Systems & Control Letters* **12**, 437–442.

Sorensen, A. J. (2013), *Marine Control Systems Propulsion and Motion Control of Ships and Ocean Structures*, 3 edn, Kompendieforlaget, Nardoveien 12, 7005 Trondheim, Norway, chapter 8, p. 233.

Thyri, E. H., Basso, E. A., Breivik, M., Pettersen, K. Y., Skjetne, R. & Lekkas, A. M. (2020), Reactive collision avoidance fos asvs based on control barrier functions, *in* 'IEEE Conference on Control Technology and Applications (CCTA)', IEEE, pp. 380–387.

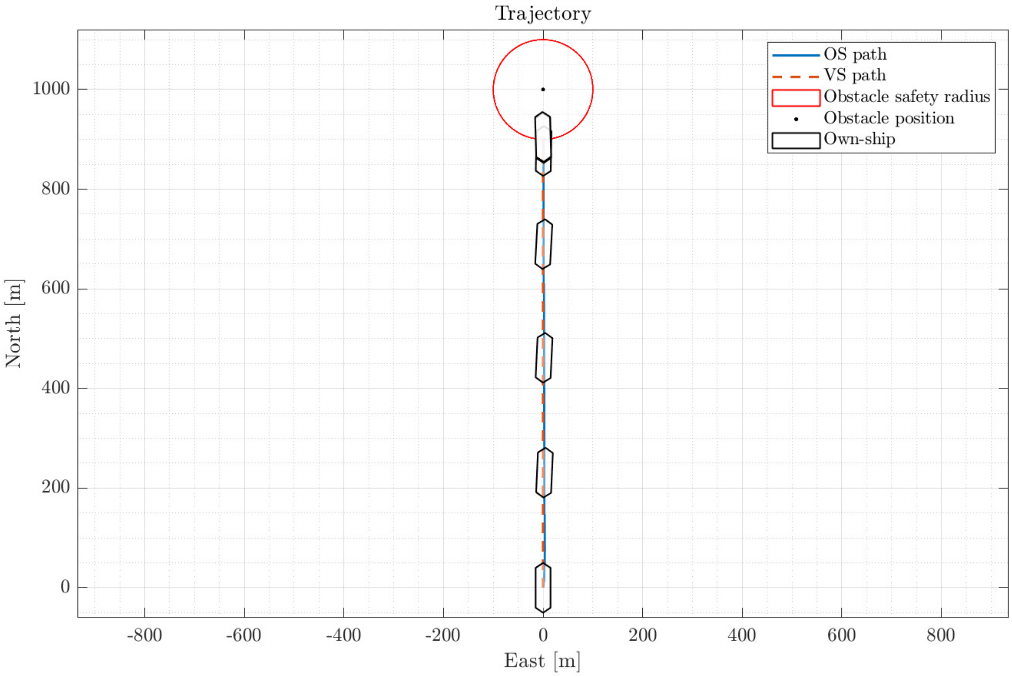
Tsinias, J. (1989), 'Sufficient lyapunov-like conditions for stabilization', *Mathematics of Control, Signals and Systems* **2**, 343–357.

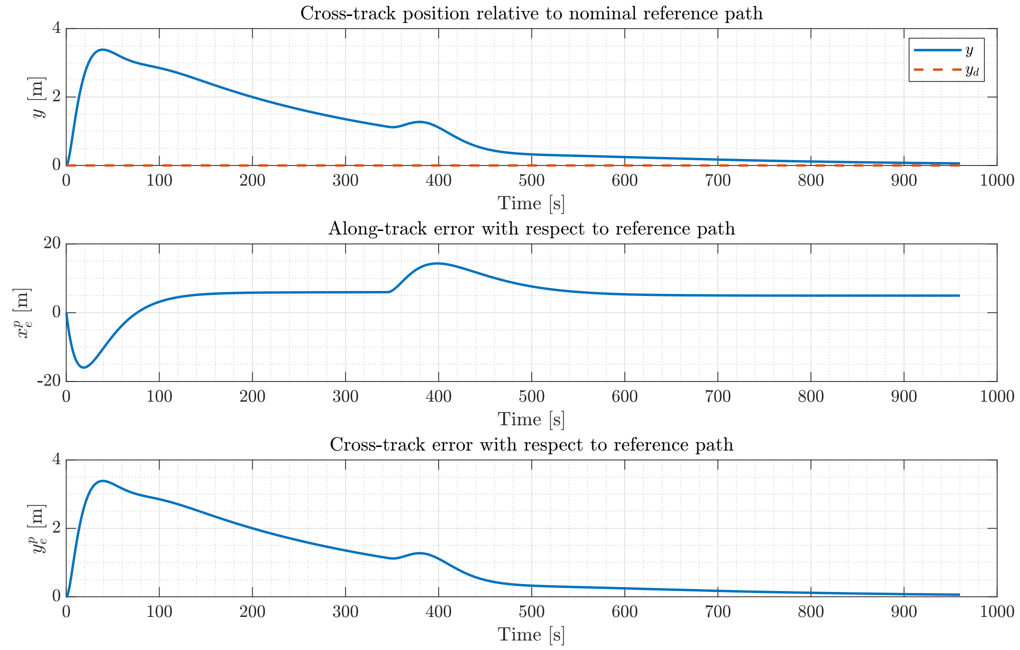
Wieland, P. & Allgwer, F. (2007), 'Constructive safety using control barrier functions', *IFAC Proceedings Volumes* **40**(12), 462–467. 7th IFAC Symposium on Nonlinear Control Systems.

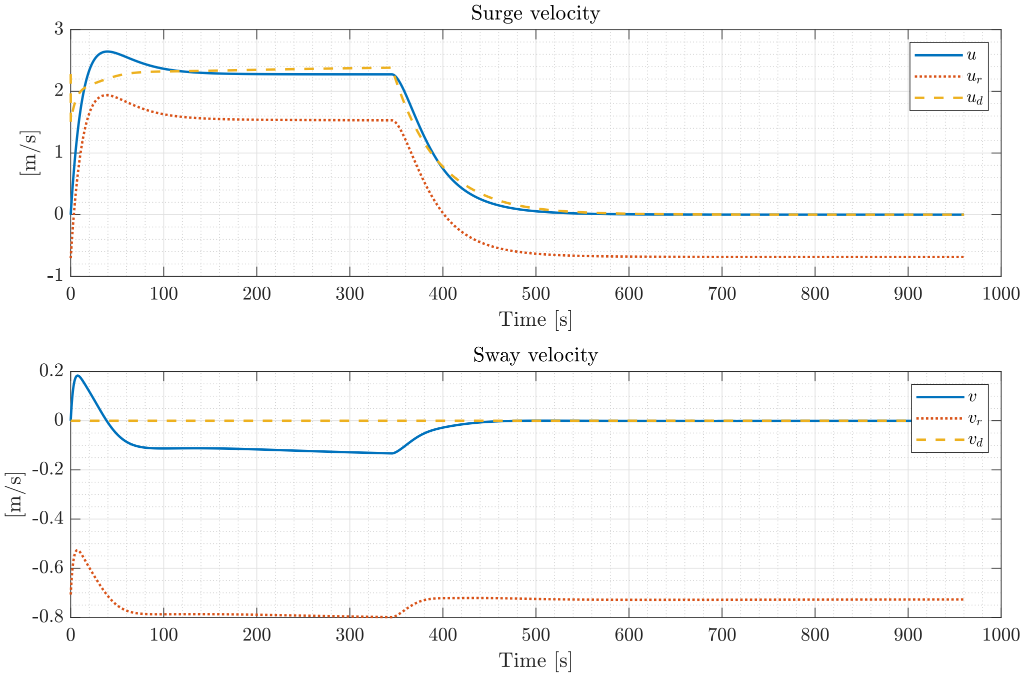
Yoerger, D. & Slotine, J. (1985), 'Robust trajectory control of underwater vehicles', *IEEE Journal of Oceanic Engineering* **10**(4), 462–470.

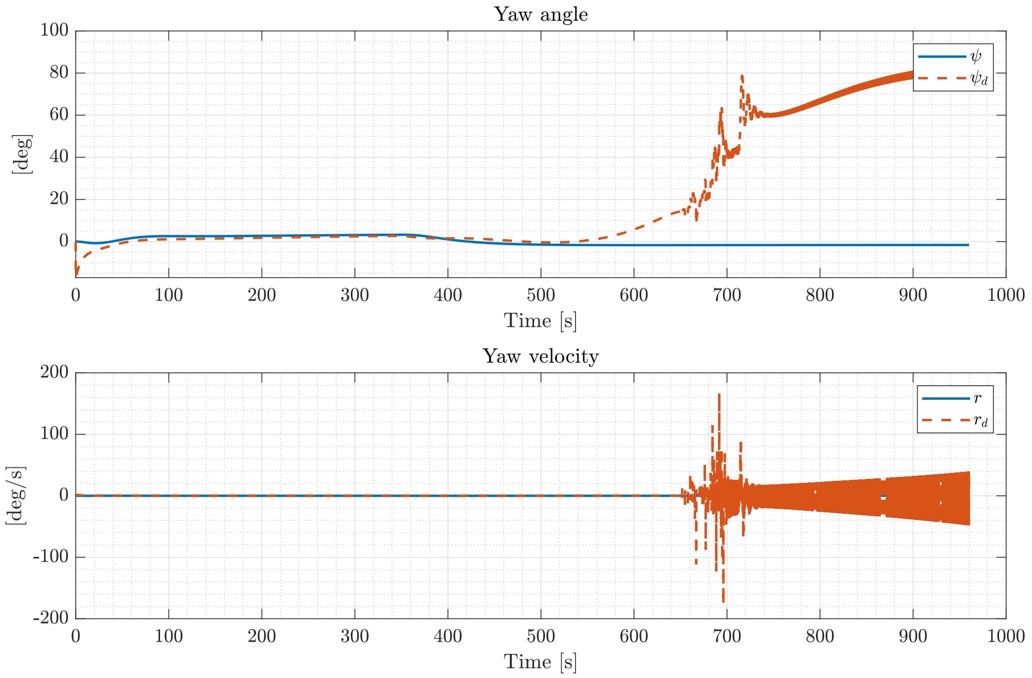
Appendix

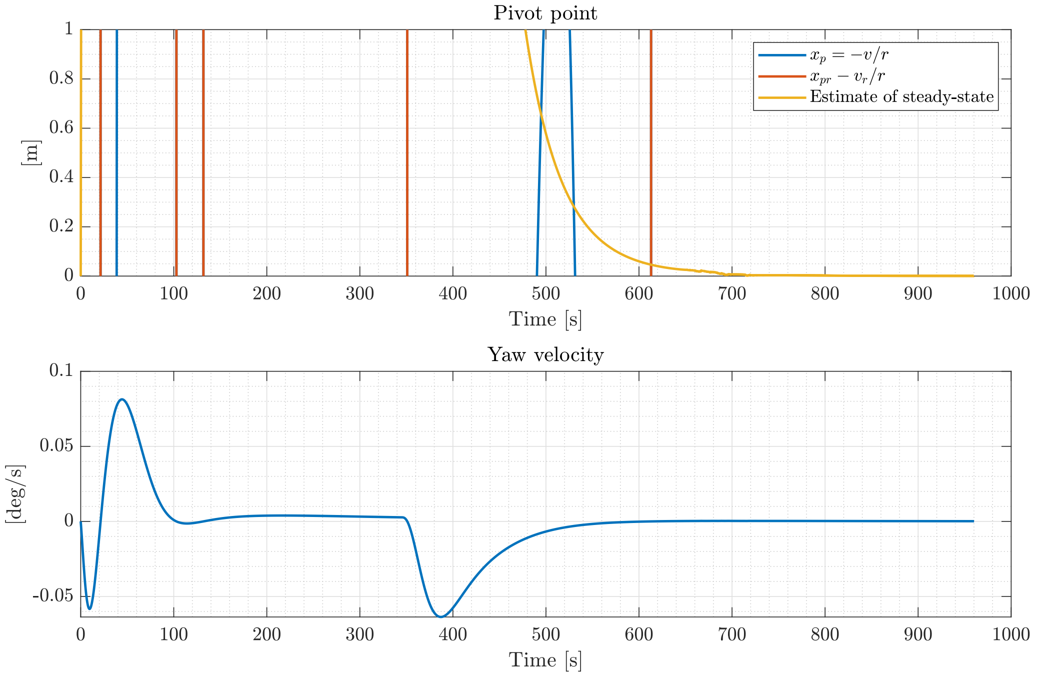
A Figures from Section 7.3.1

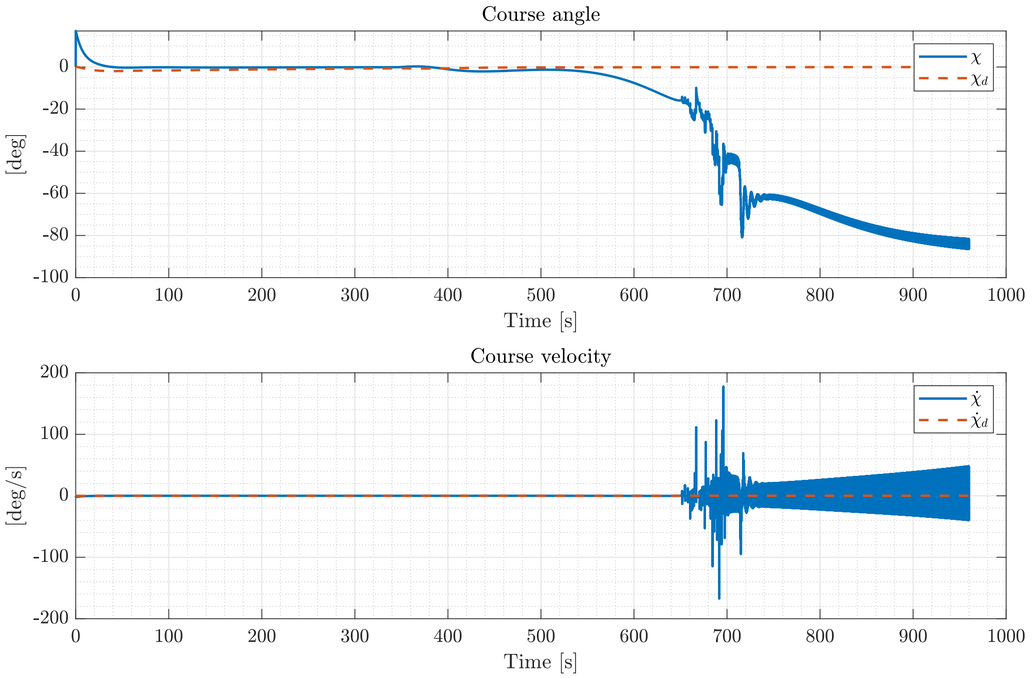


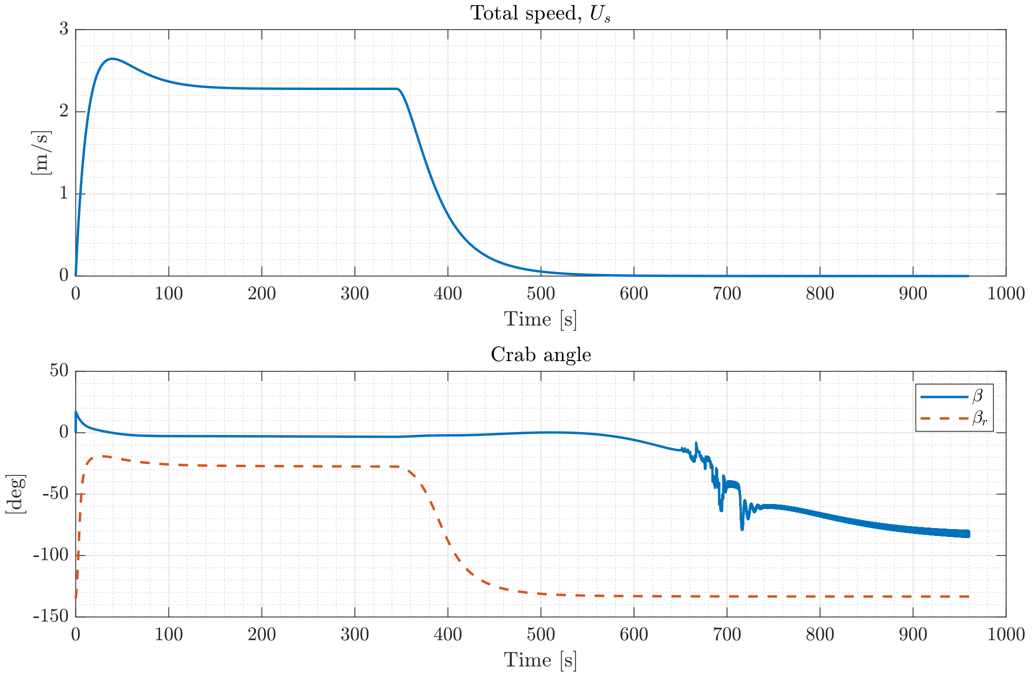

Cross-track position relative to nominal reference path
y [m]
y
y_d
Time [s]
Along-track error with respect to reference path
x_e^p [m]
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
Time [s]
0
100
200
300
400
500
600
700
800
900
1000
4
2
0
20
-20

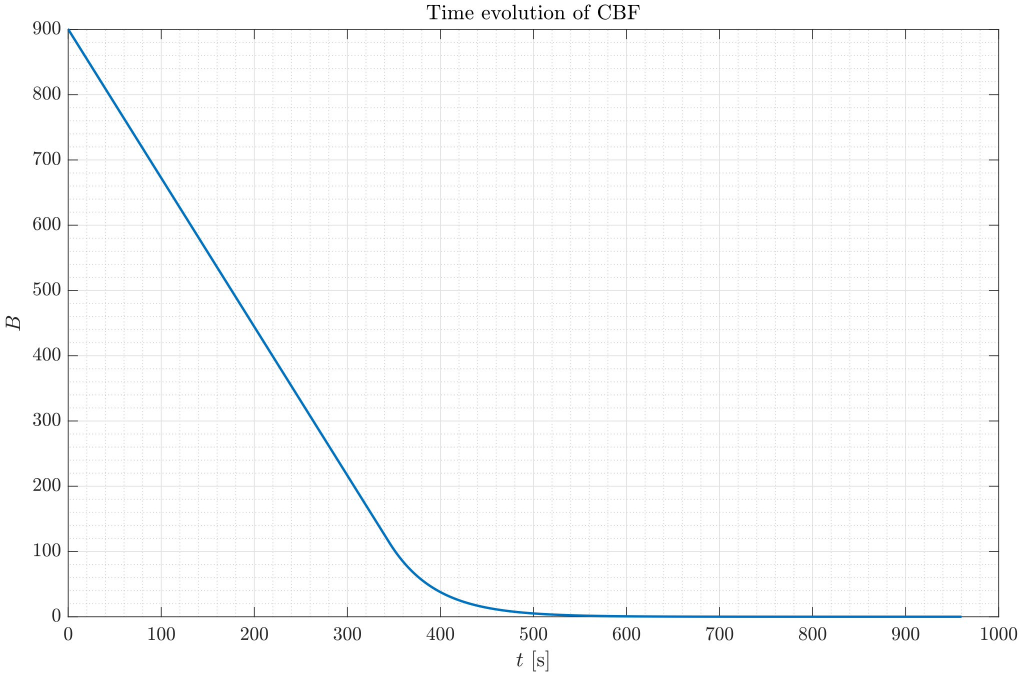

Surge velocity
[m/s]
u
u_r
u_d
Time [s]
3
2
1
0
-1
Sway velocity
[m/s]
v
v_r
v_d
Time [s]
0.2
0
-0.2
-0.4
-0.6
-0.8
0
100
200
300
400
500
600
700
800
900
1000

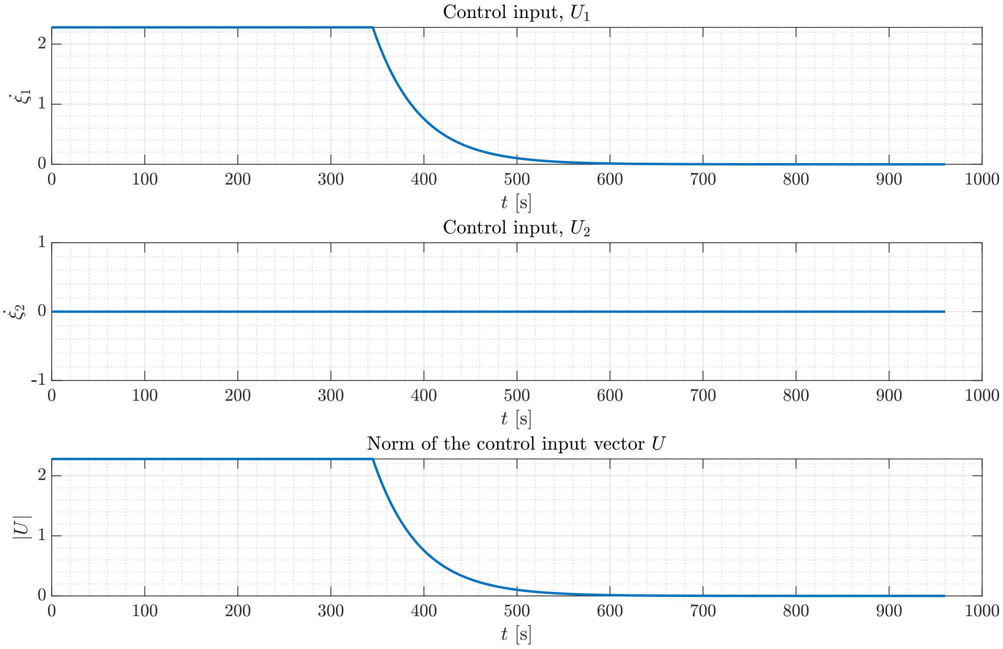

Yaw angle
100
80
60
40
20
0
[deg]
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
ψ
ψ_d
Yaw velocity
200
100
0
-100
-200
[deg/s]
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
r
r_d

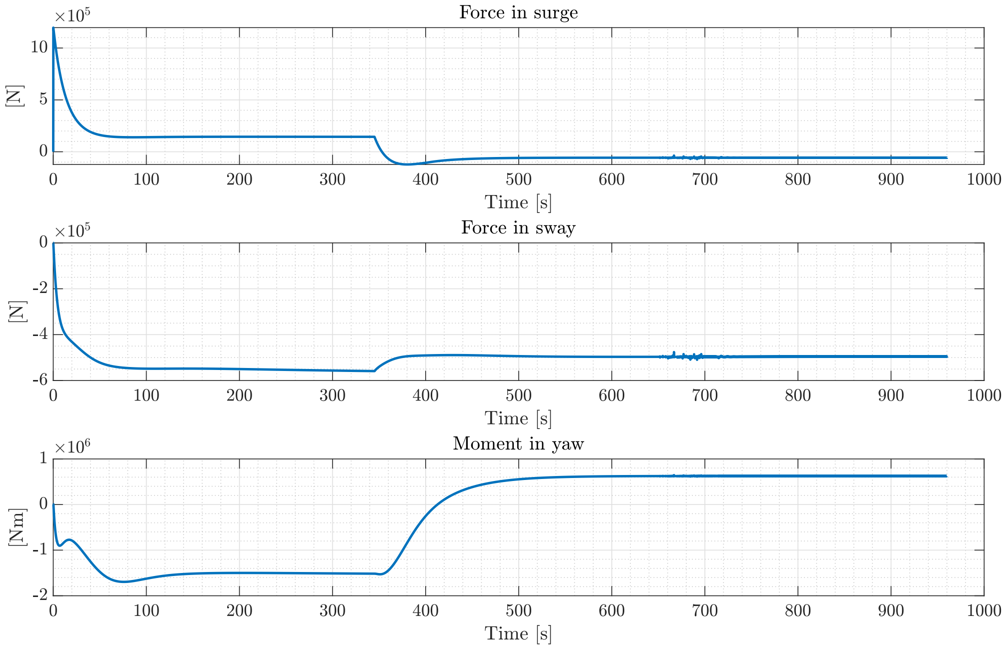

Pivot point
1
0.8
0.6
0.4
0.2
0
[m]
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
Yaw velocity
0.1
0.05
0
-0.05
[deg/s]
0
100
200
300
400
500
600
700
800
900
1000
Time [s]


Course angle
[deg]
0
-20
-40
-60
-80
-100
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
χ
χ_d
Course velocity
[deg/s]
200
100
0
-100
-200
Time [s]
$\dot{\chi}$
$\dot{\chi}_d$

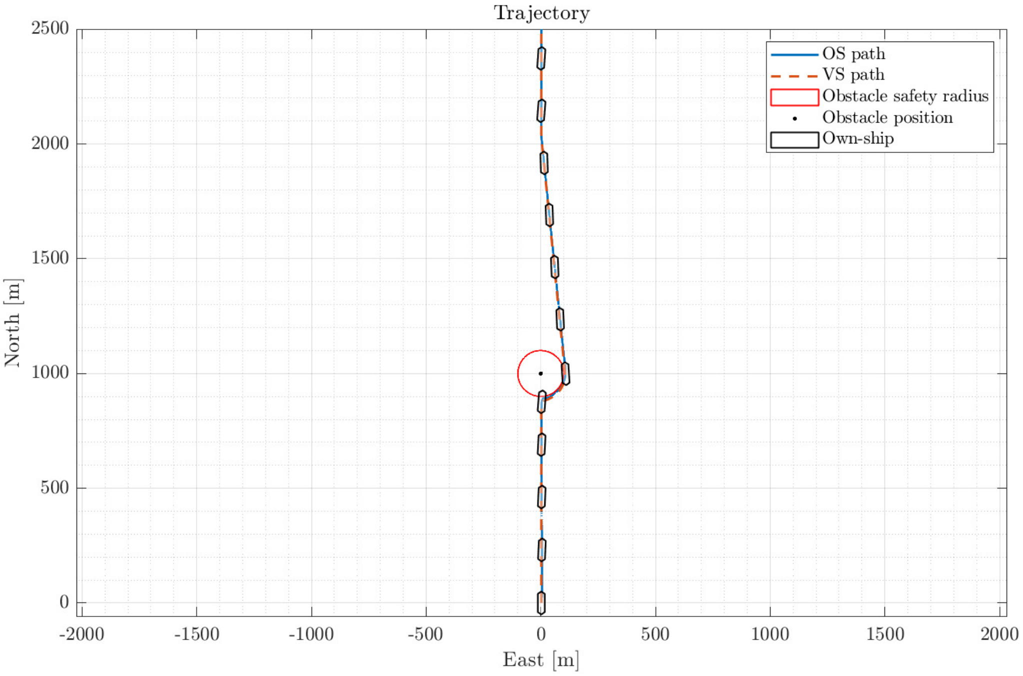

Total speed, U_s
[m/s]
3
2
1
0
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
Crab angle
[deg]
50
0
-50
-100
-150
Time [s]
β
β_r

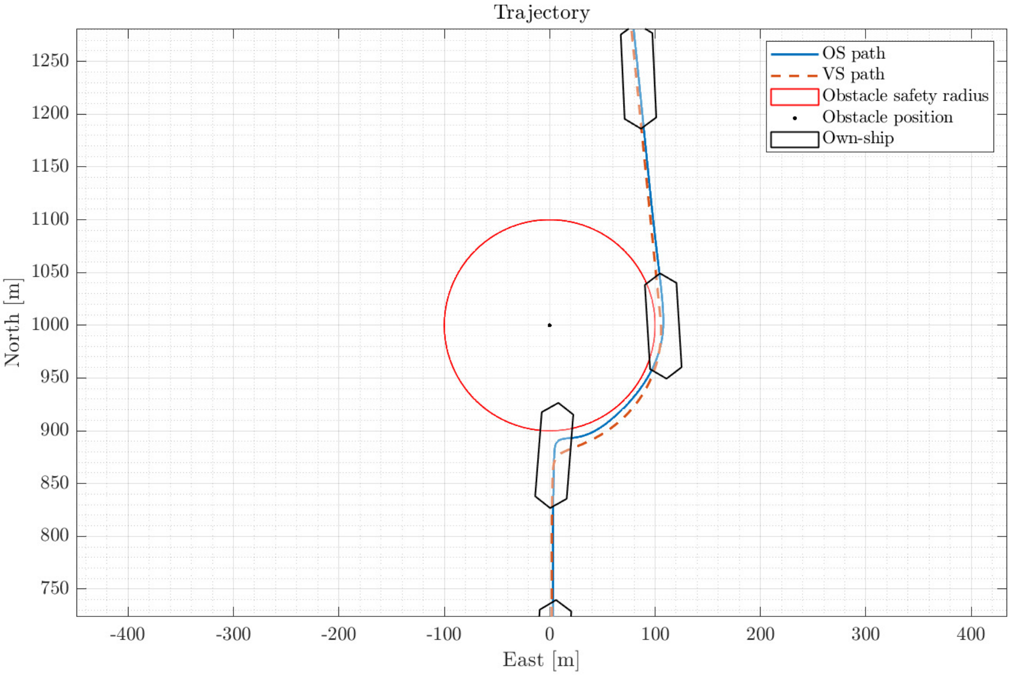

Time evolution of CBF
B
900
800
700
600
500
400
300
200
100
0
0
100
200
300
400
500
600
700
800
900
1000
t [s]

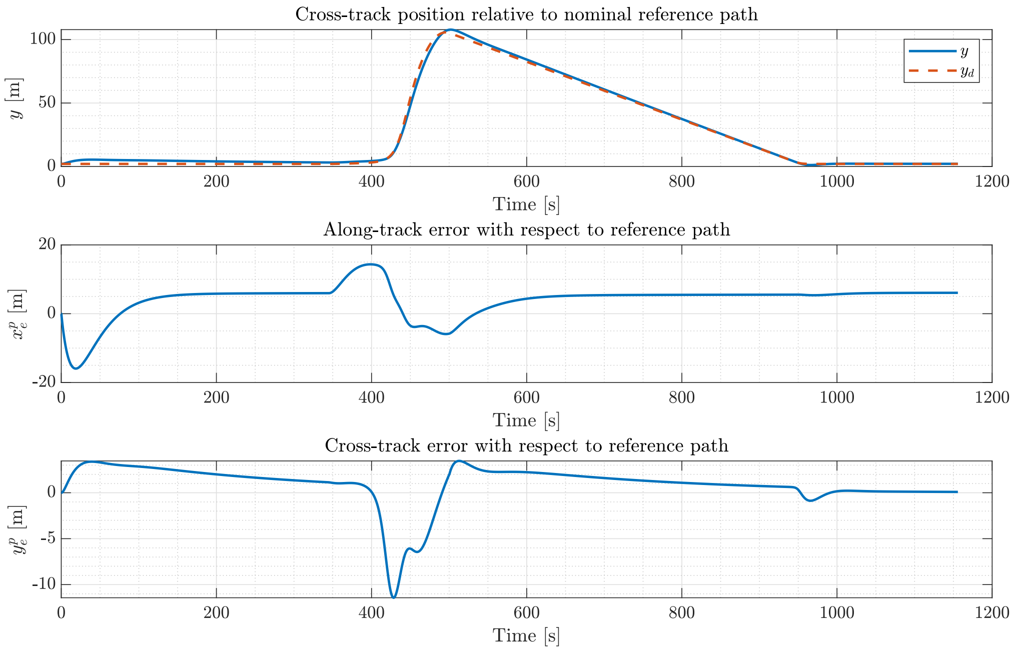

Control input, U_1
$\dot{\xi}_1$
2
1
0
t [s]
Control input, U_2
$\dot{\xi}_2$
1
0
-1
t [s]
Norm of the control input vector U
$|U|$
2
1
0
0
100
200
300
400
500
600
700
800
900
1000
t [s]

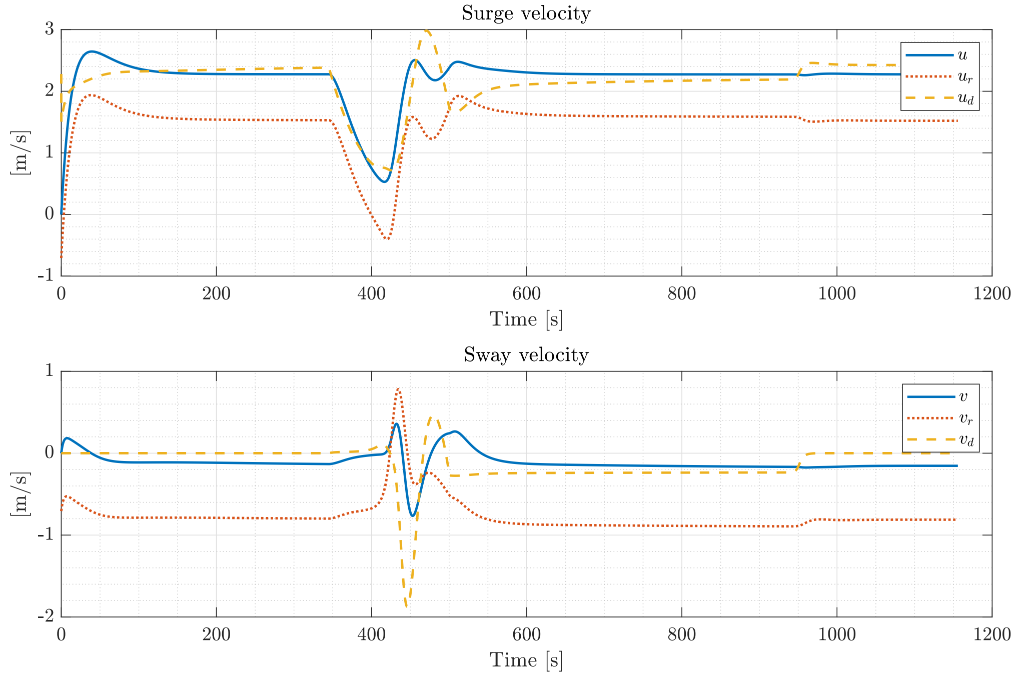

Force in surge
×10^5
[N]
10
5
0
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
Force in sway
×10^5
[N]
0
-2
-4
-6
0
100
200
300
400
500
600
700
800
900
1000
Time [s]
Moment in yaw
×10^6
[Nm]
1
0
-1
-2
0
100
200
300
400
500
600
700
800
900
1000
Time [s]

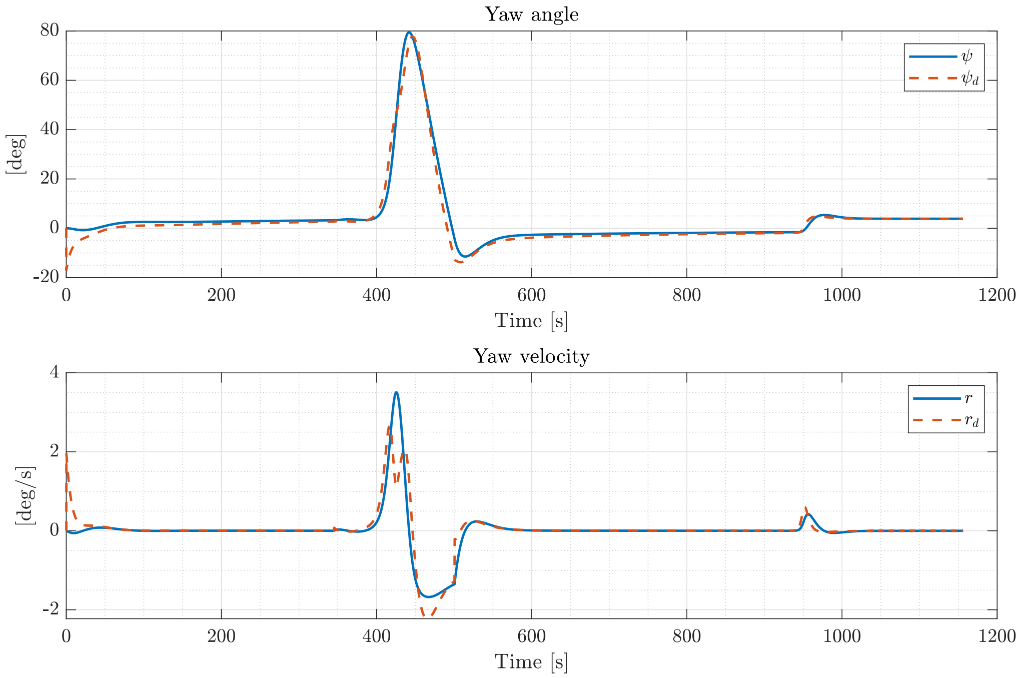
B Figures from Section 7.3.2

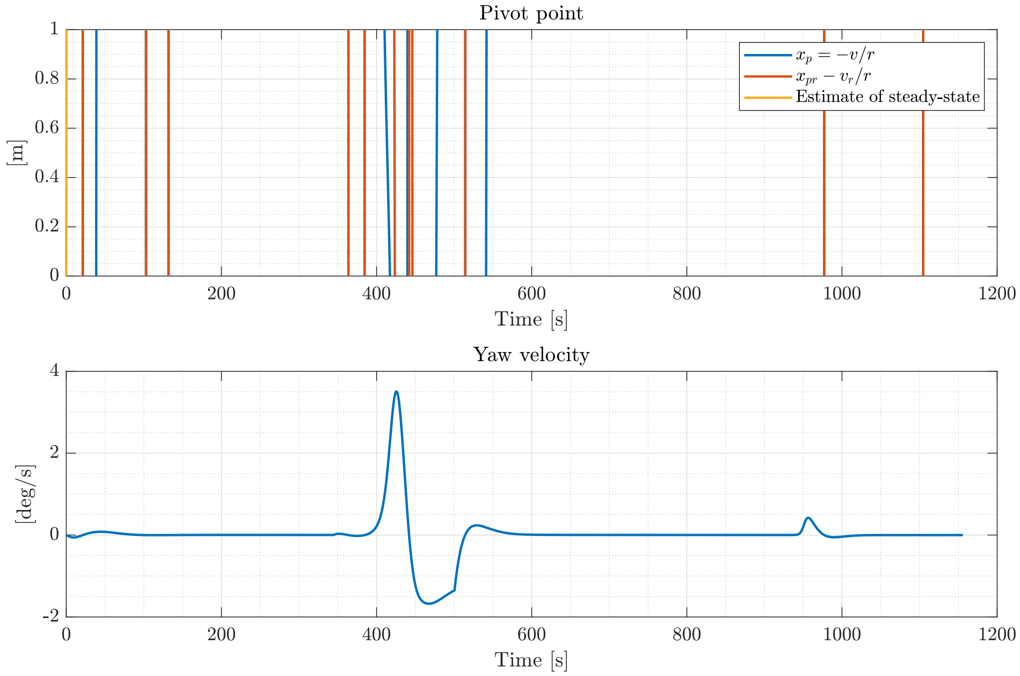


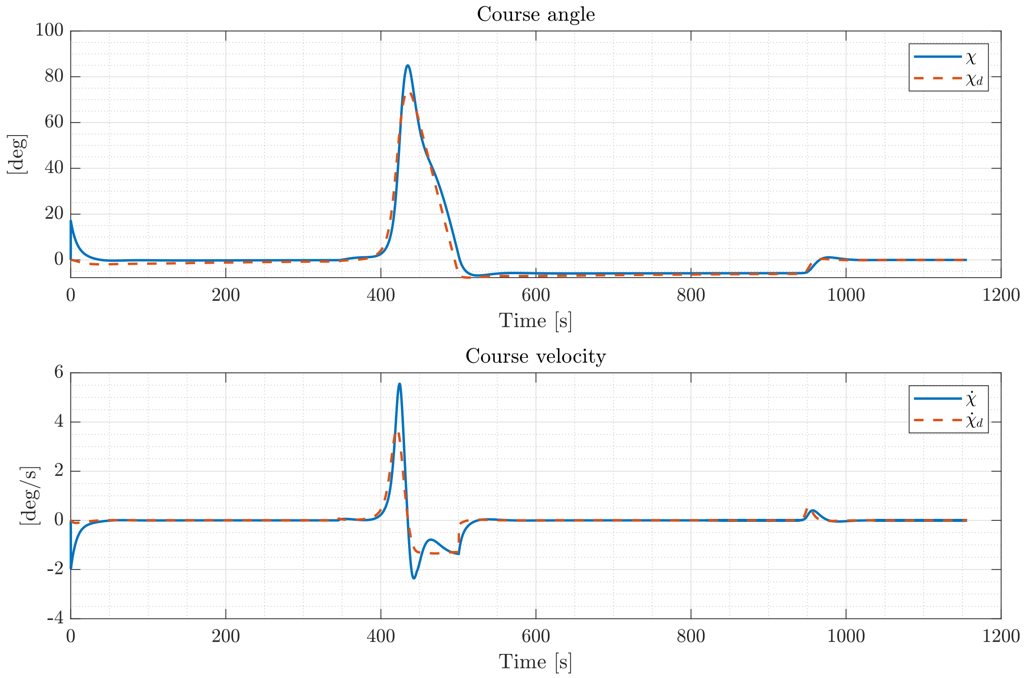


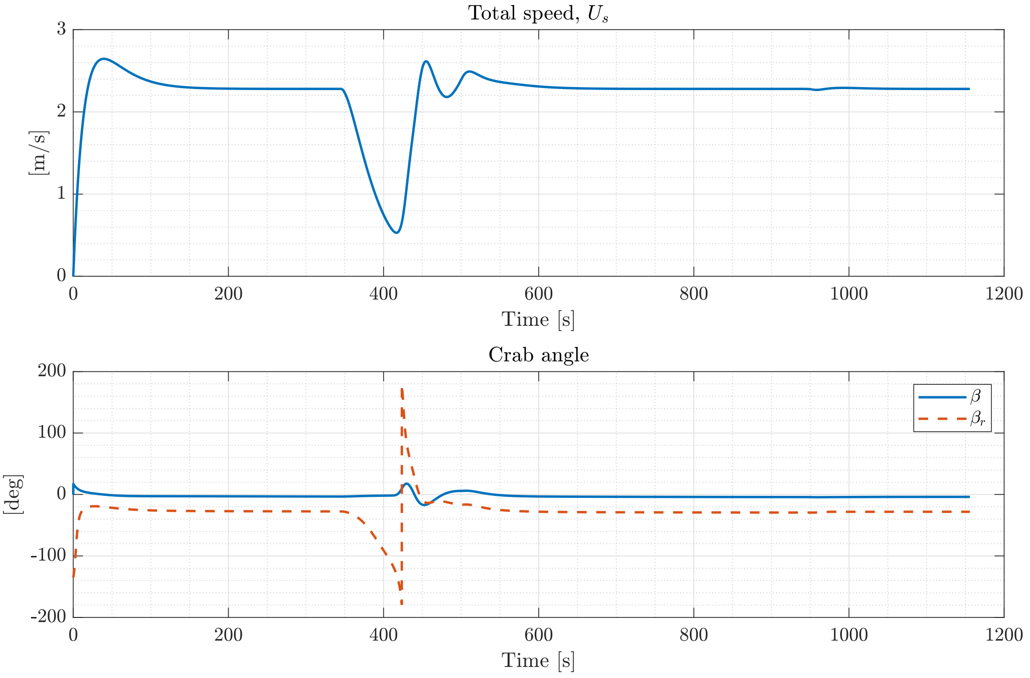

Cross-track position relative to nominal reference path
y [m]
100
50
0
0
200
400
600
800
1000
1200
Time [s]
y
y_d
Along-track error with respect to reference path
x_e^p [m]
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
0
-5
-10
0
200
400
600
800
1000
1200
Time [s]

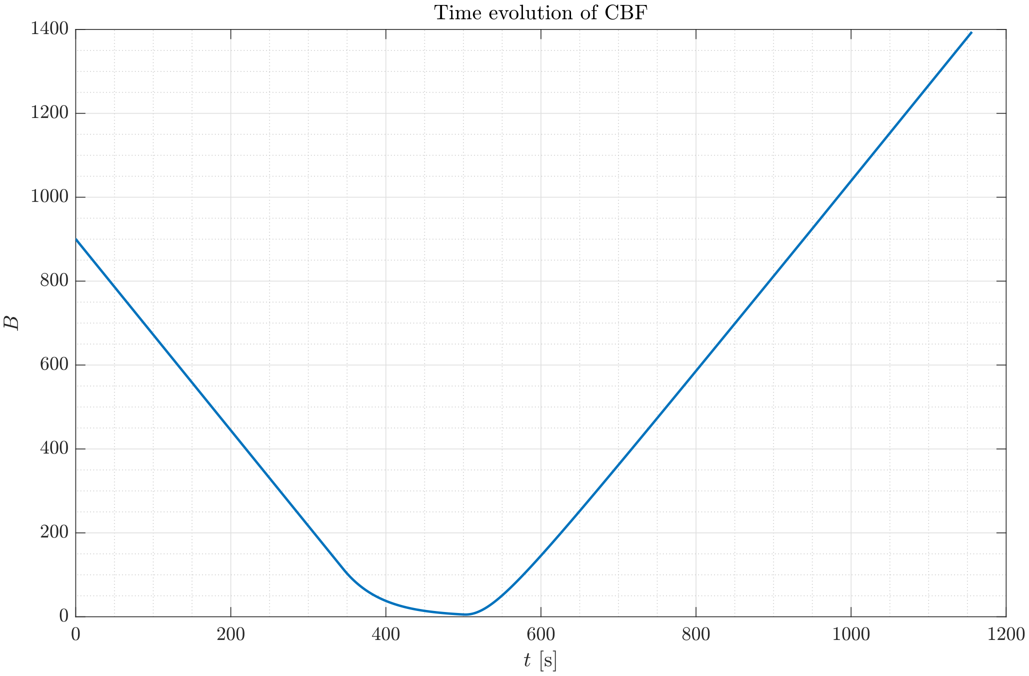

Surge velocity
[m/s]
3
2
1
0
-1
0
200
400
600
800
1000
1200
Time [s]
u
u_r
u_d
Sway velocity
[m/s]
1
0
-1
-2
0
200
400
600
800
1000
1200
Time [s]
v
v_r
v_d

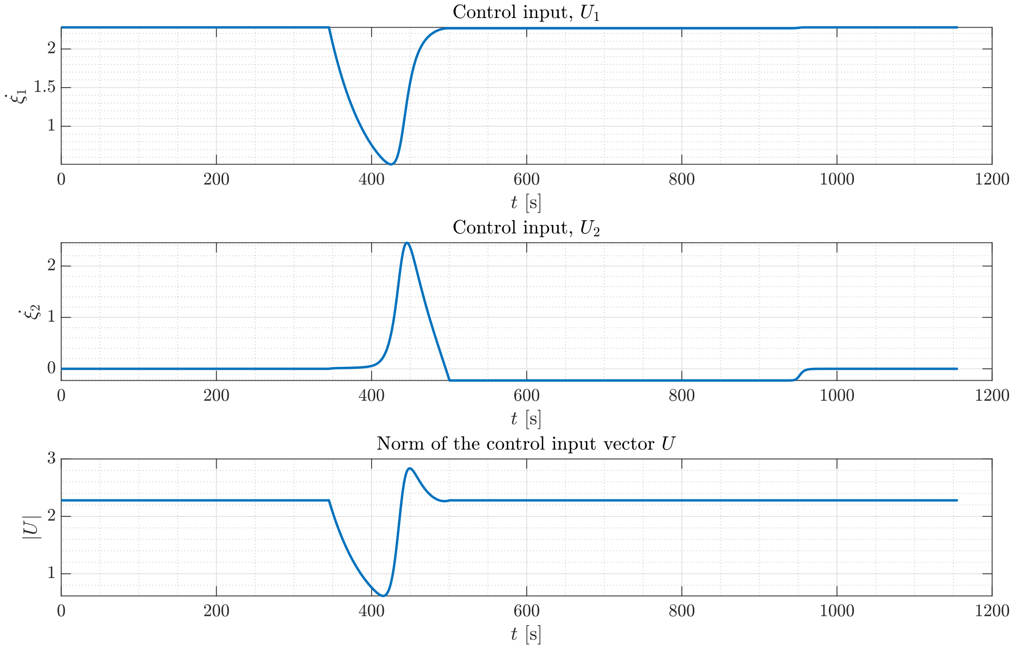

Yaw angle
ψ
ψ_d
[deg]
80
60
40
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
Yaw velocity
r
r_d
[deg/s]
4
2
0
-2
Time [s]

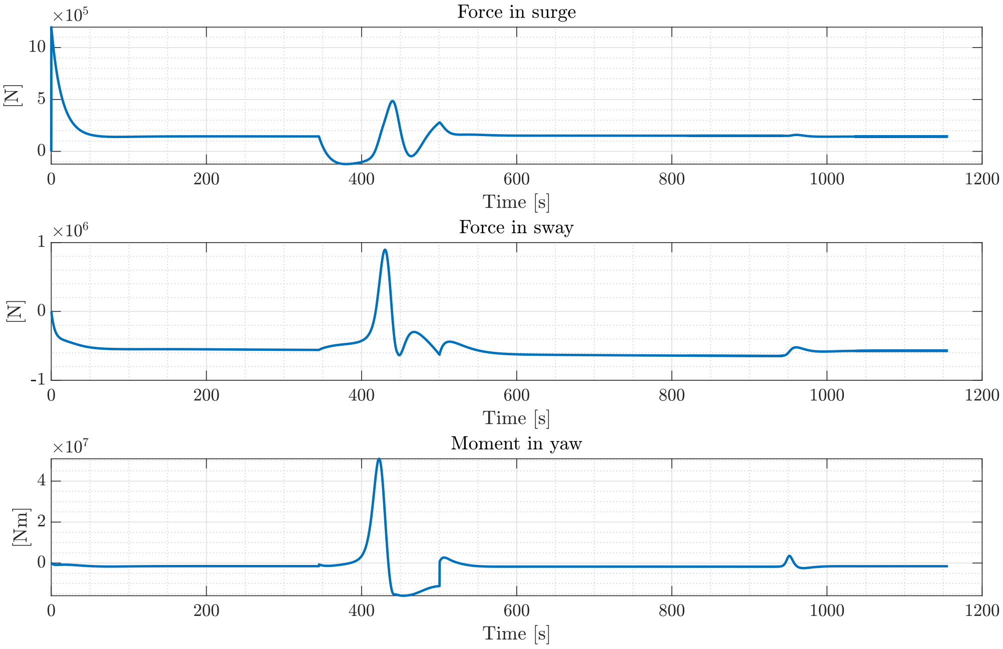

Pivot point
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
[m]
1
0.8
0.6
0.4
0.2
0
0
200
400
600
800
1000
1200
Time [s]
Yaw velocity
[deg/s]
4
2
0
-2
Time [s]


Course angle
[deg]
100
80
60
40
20
0
0
200
400
600
800
1000
1200
Time [s]
χ
χ_d
Course velocity
[deg/s]
6
4
2
0
-2
-4
0
200
400
600
800
1000
1200
Time [s]
$\dot{\chi}$
$\dot{\chi}_d$

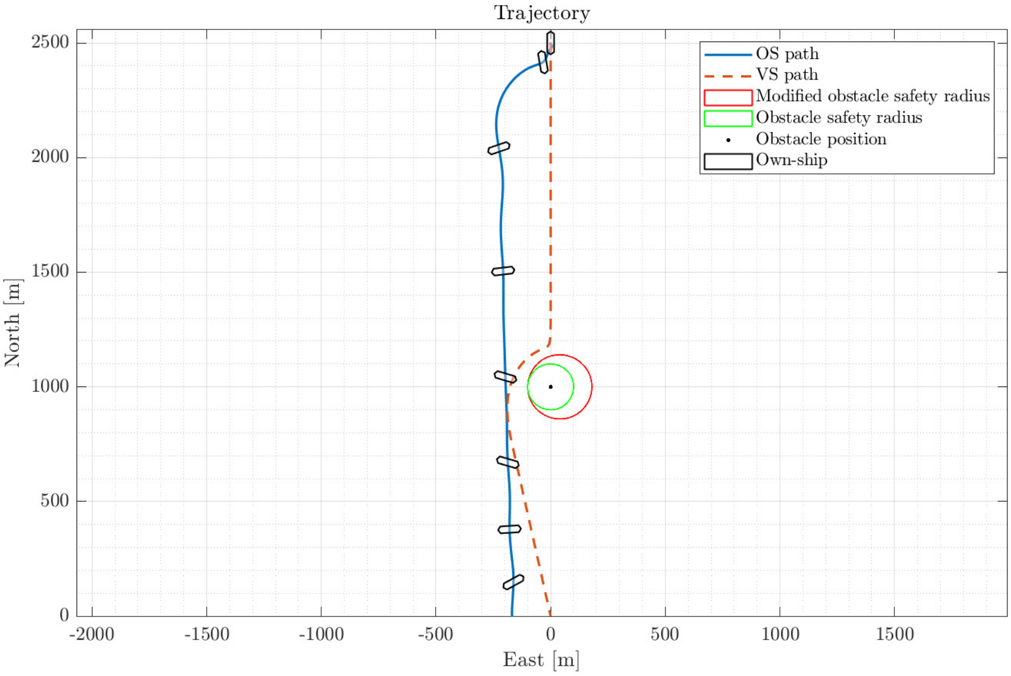

Total speed, U_s
[m/s]
3
2
1
0
0
200
400
600
800
1000
1200
Time [s]
Crab angle
[deg]
200
100
0
-100
-200
0
200
400
600
800
1000
1200
Time [s]
β
β_r

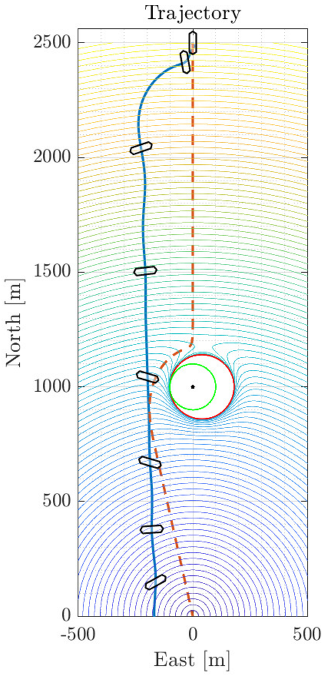

Time evolution of CBF
B
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
t [s]

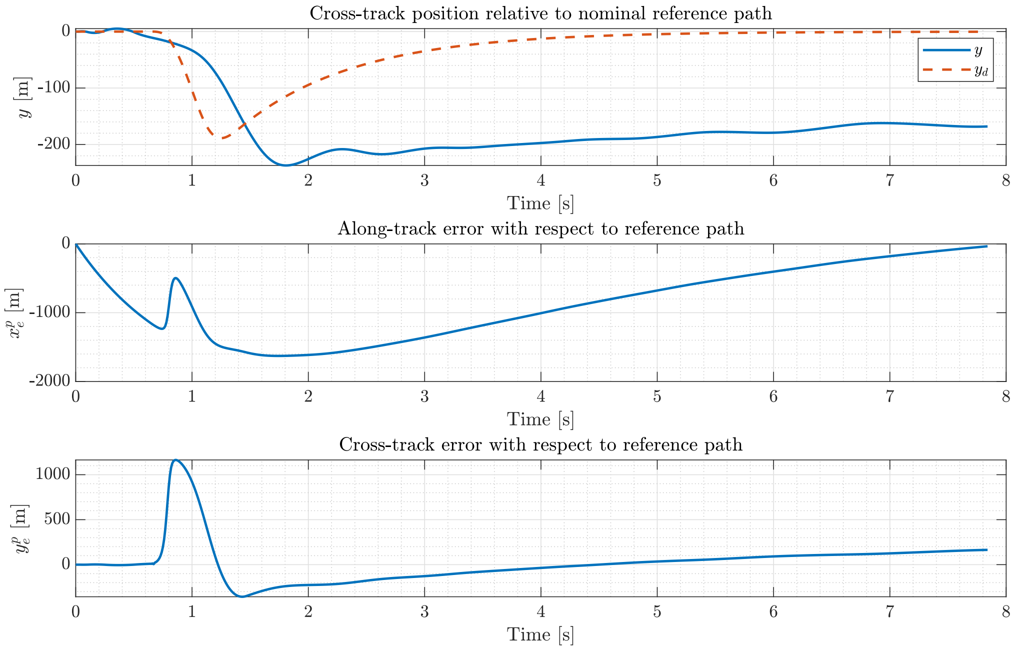

Control input, U_1
$\dot{\xi}_1$
2
1.5
1
0
200
400
600
800
1000
1200
t [s]
Control input, U_2
$\dot{\xi}_2$
2
1
0
0
200
400
600
800
1000
1200
t [s]
Norm of the control input vector U
$|U|$
3
2
1
0
200
400
600
800
1000
1200
t [s]


Force in surge
×10^5
10
5
0
[N]
0
200
400
600
800
1000
1200
Time [s]
Force in sway
×10^6
1
0
-1
[N]
0
200
400
600
800
1000
1200
Time [s]
Moment in yaw
×10^7
4
2
0
[Nm]
0
200
400
600
800
1000
1200
Time [s]

C Figures from Section 7.4

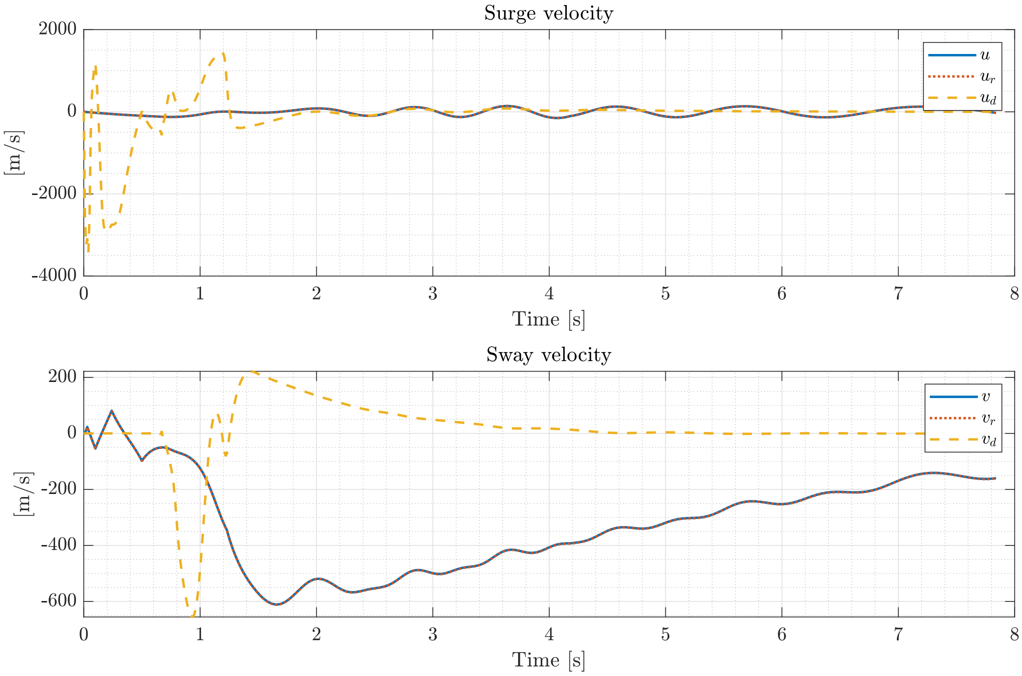




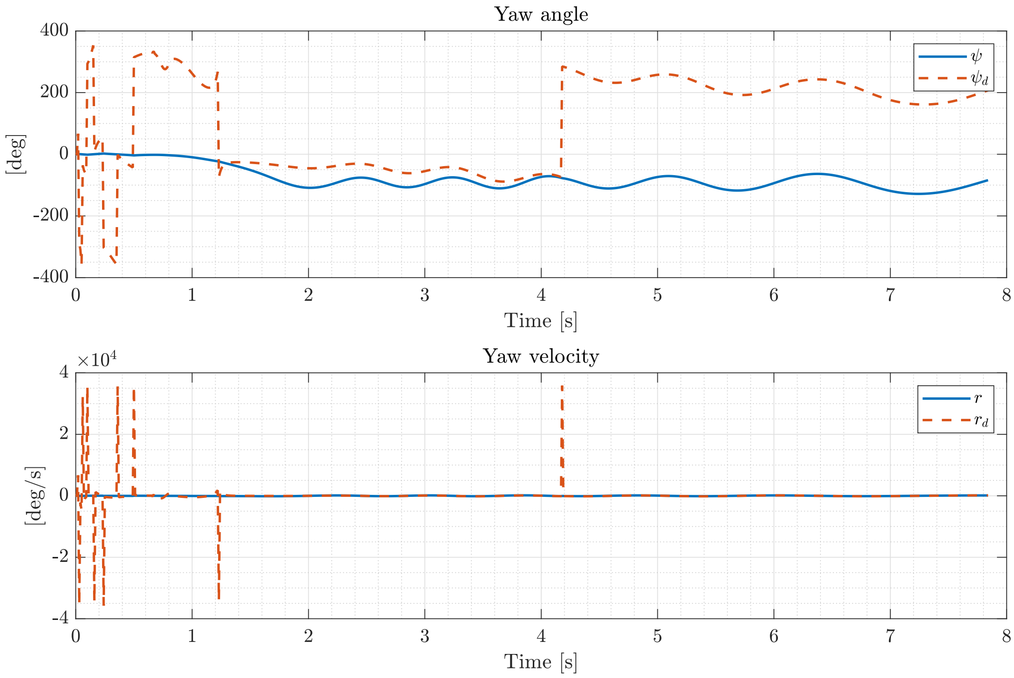

Cross-track position relative to nominal reference path

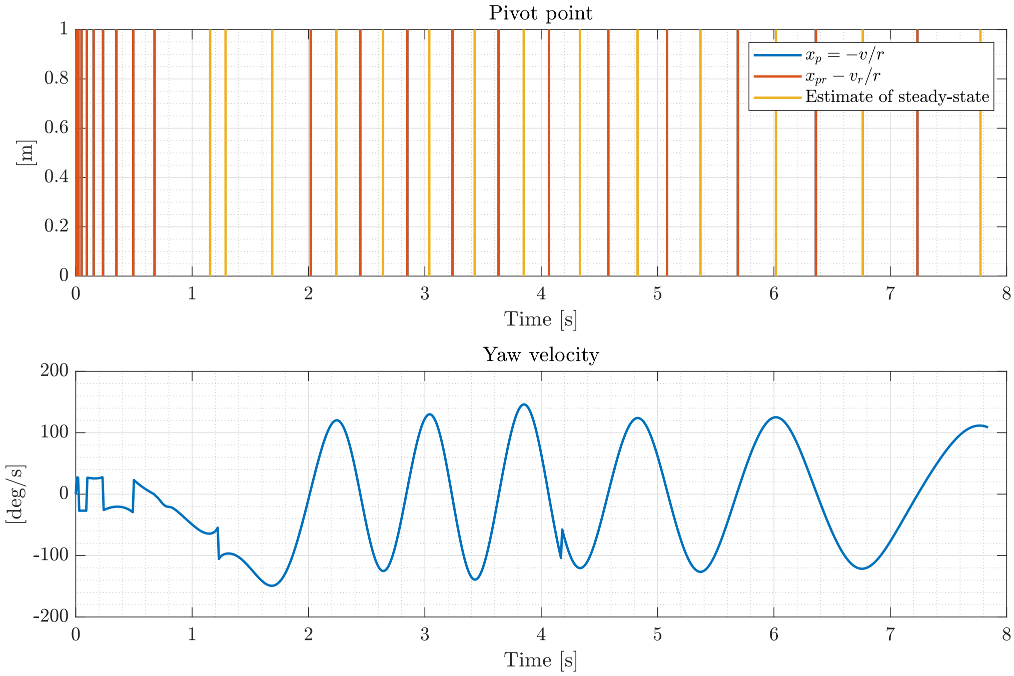
Along-track error with respect to reference path

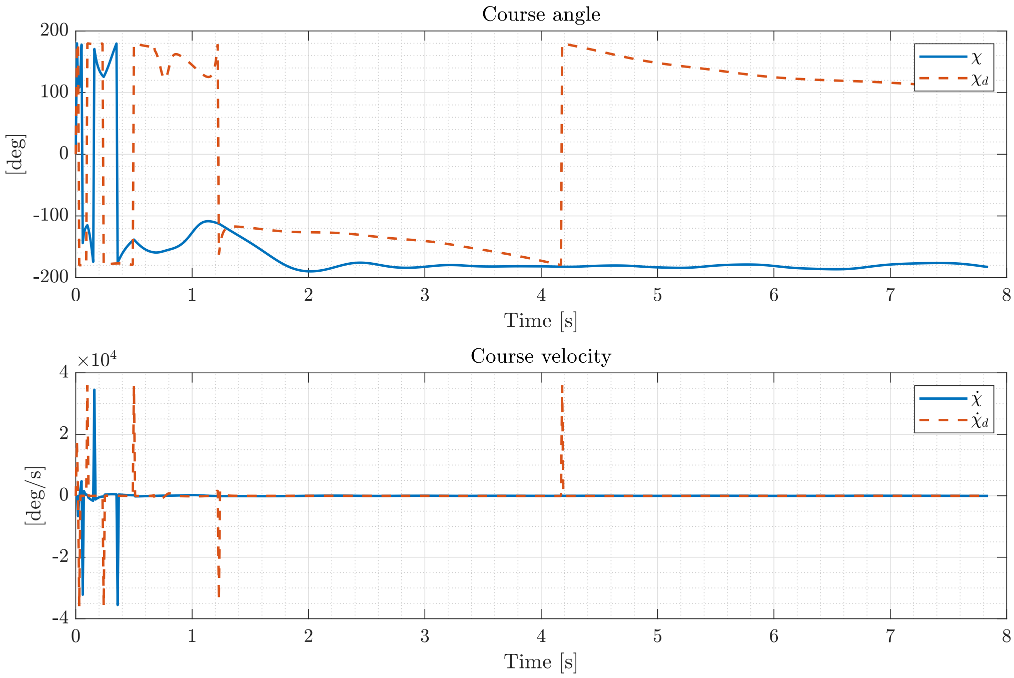
Cross-track error with respect to reference path

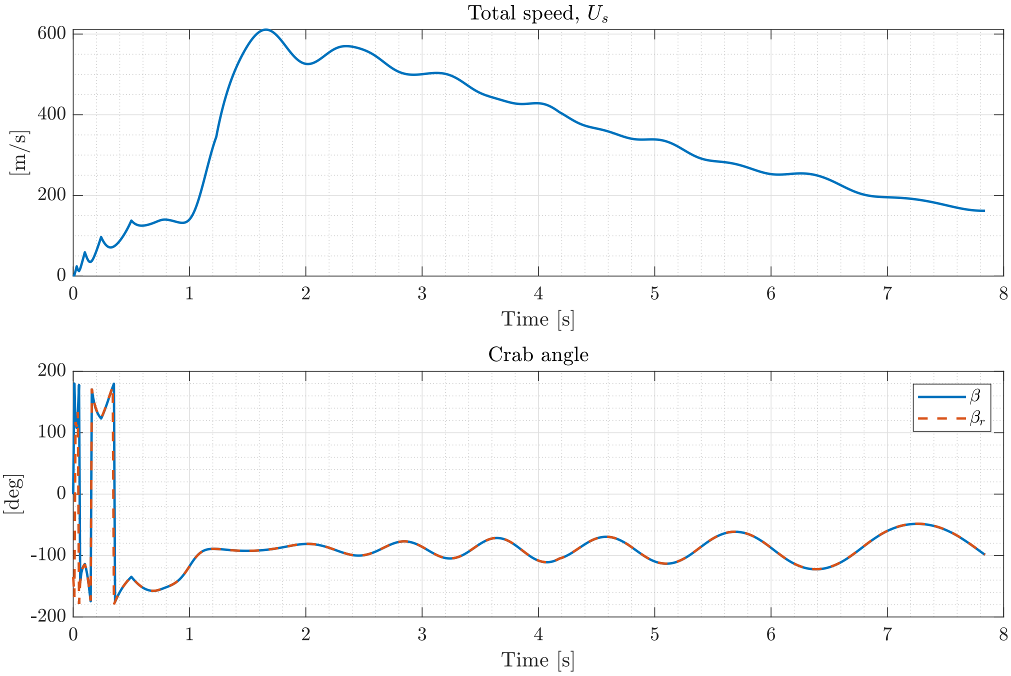

Surge velocity

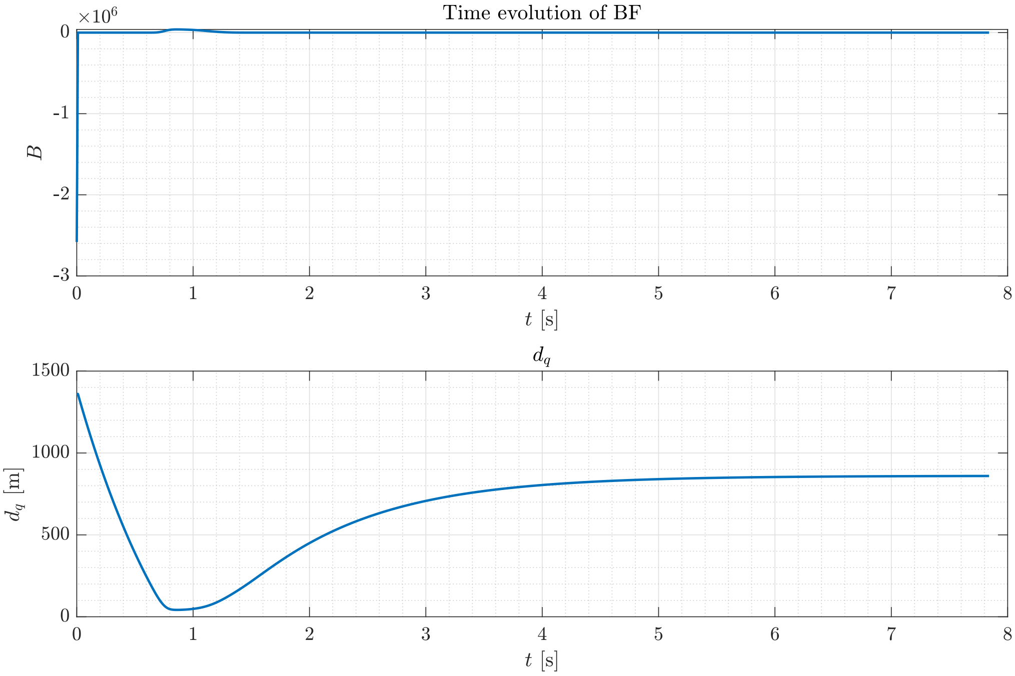
Sway velocity

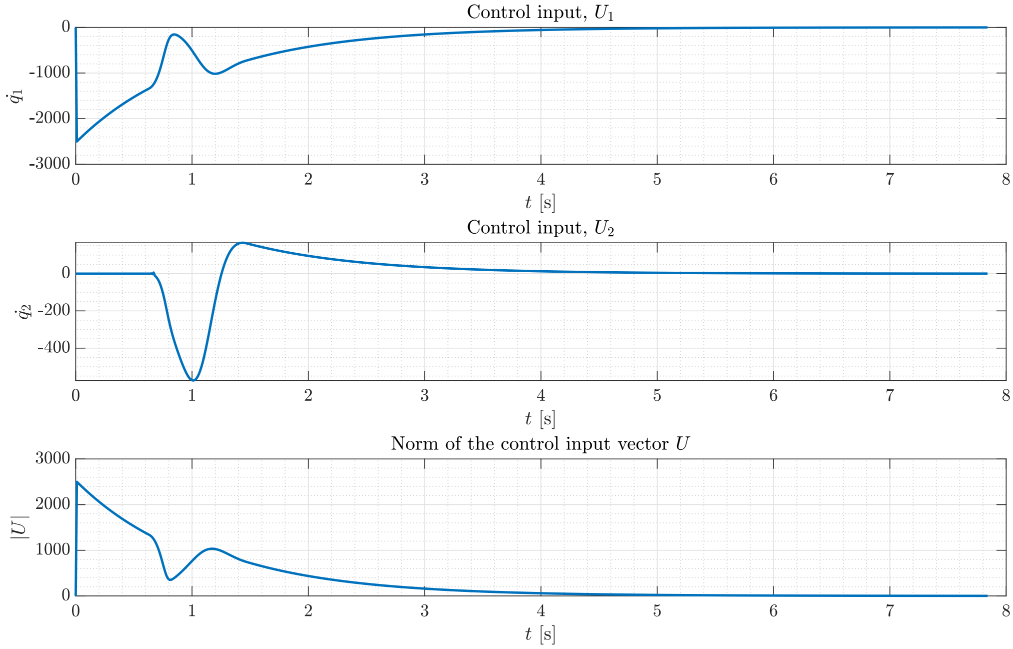

Yaw angle
400
200
0
-200
-400
[deg]
ψ
ψ_d
0
1
2
3
4
5
6
7
8
Time [s]
Yaw velocity
$\times 10^4$
4
2
0
-2
-4
[deg/s]
r
r_d
0
1
2
3
4
5
6
7
8
Time [s]

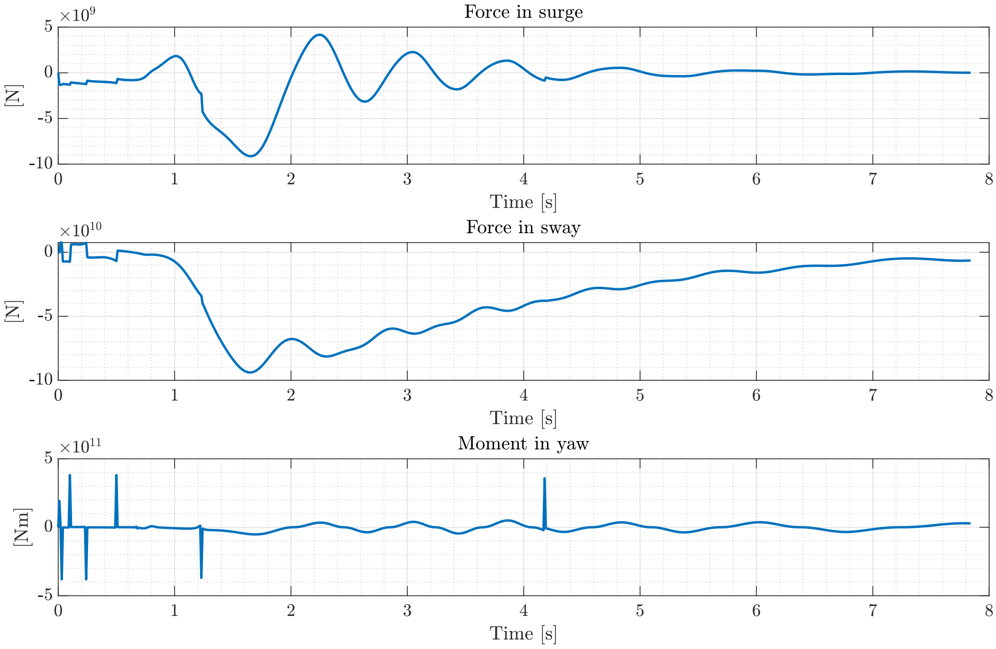

Pivot point
1
0.8
0.6
0.4
0.2
0
[m]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
0
1
2
3
4
5
6
7
8
Time [s]
Yaw velocity
200
100
0
-100
-200
[deg/s]
0
1
2
3
4
5
6
7
8
Time [s]


Course angle
200
100
0
-100
-200
[deg]
χ
χ_d
0
1
2
3
4
5
6
7
8
Time [s]
Course velocity
$\times 10^4$
4
2
0
-2
-4
[deg/s]
$\dot{\chi}$
$\dot{\chi}_d$
0
1
2
3
4
5
6
7
8
Time [s]

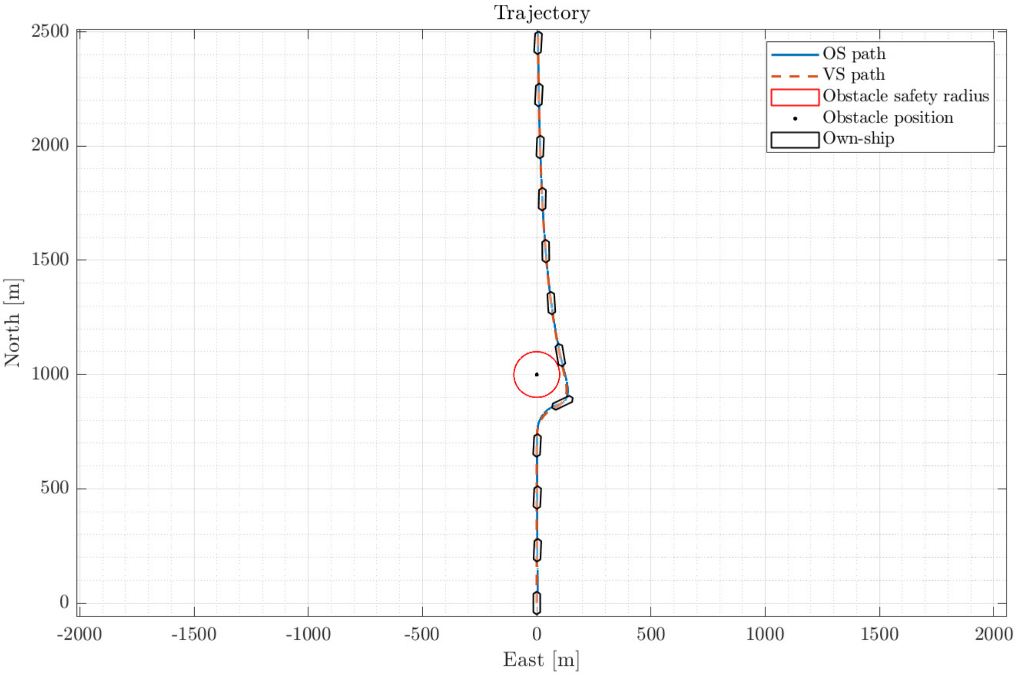

Total speed, U_s
600
400
200
0
[m/s]
0
1
2
3
4
5
6
7
8
Time [s]
Crab angle
200
100
0
-100
-200
[deg]
β
β_r
0
1
2
3
4
5
6
7
8
Time [s]

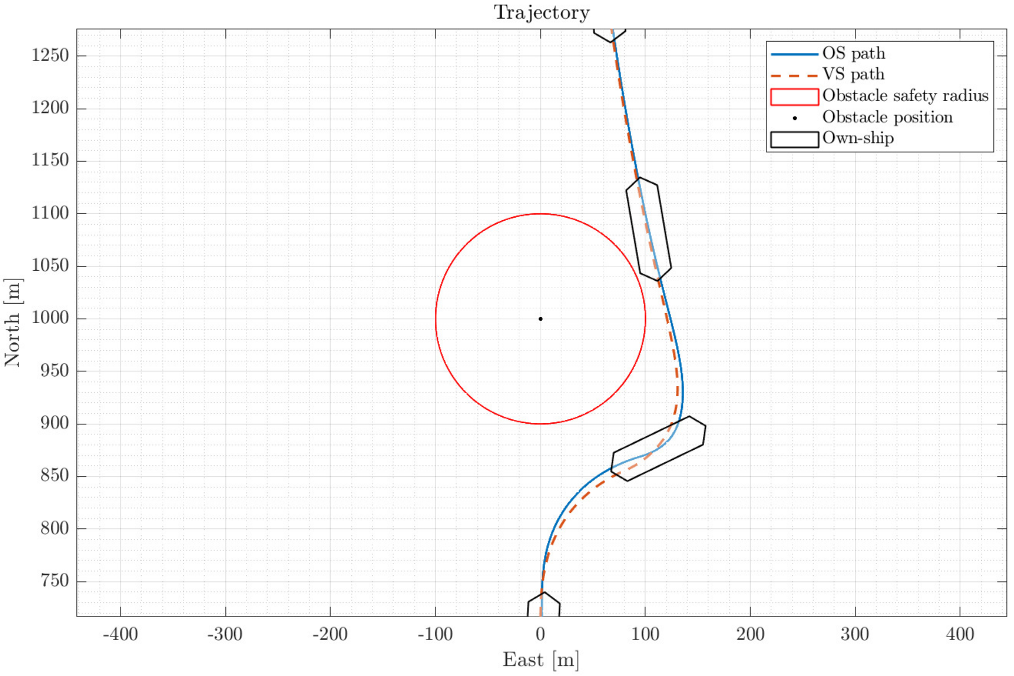

$\times 10^6$
Time evolution of BF
B
0
-1
-2
-3
0 1 2 3 4 5 6 7 8
t [s]
d_q
d_q [m]
1500
1000
500
0
0 1 2 3 4 5 6 7 8
t [s]

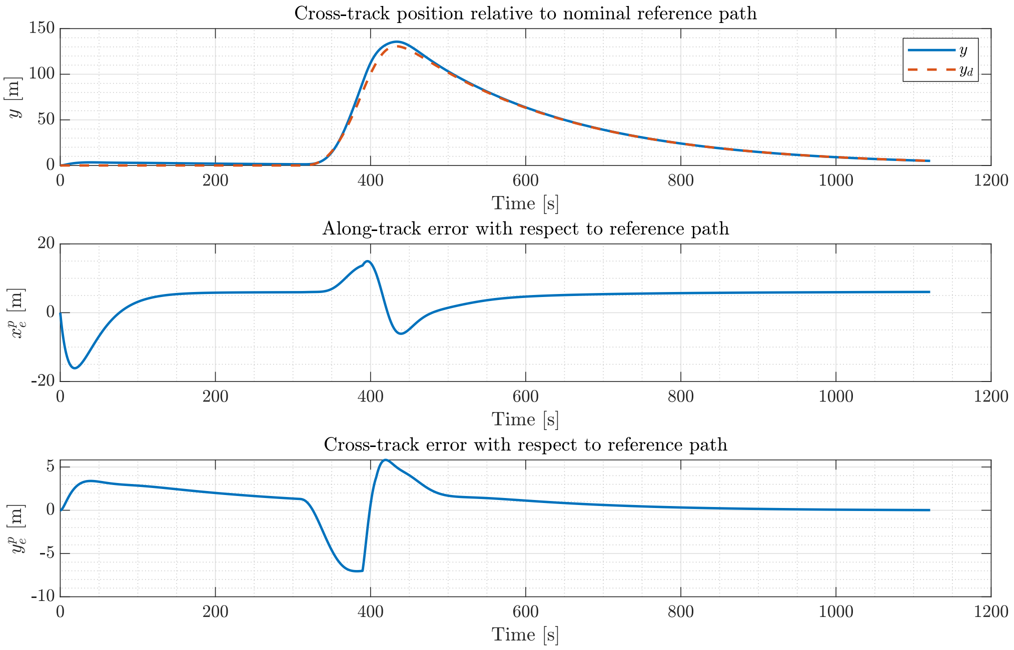

Control input, U_1
$\dot{q}_1$
0
-1000
-2000
-3000
0 1 2 3 4 5 6 7 8
t [s]
Control input, U_2
$\dot{q}_2$
0
-200
-400
0 1 2 3 4 5 6 7 8
t [s]
Norm of the control input vector U
$|U|$
3000
2000
1000
0
0 1 2 3 4 5 6 7 8
t [s]

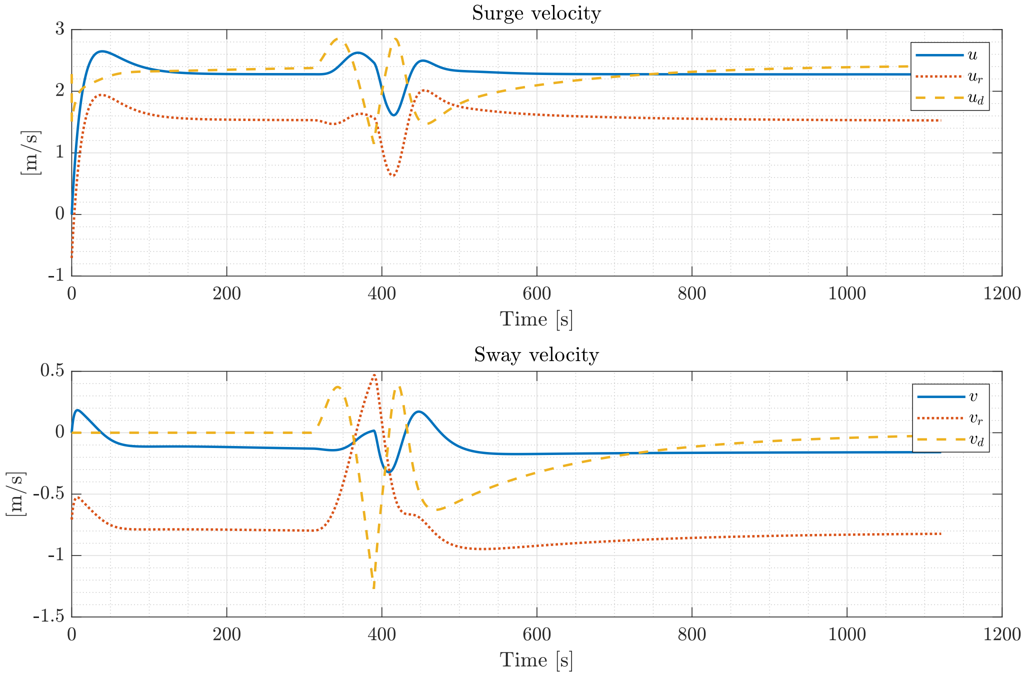

Force in surge
×10^9
[N]
5
0
-5
-10
0
1
2
3
4
5
6
7
8
Time [s]
Force in sway
×10^10
[N]
0
-5
-10
0
1
2
3
4
5
6
7
8
Time [s]
Moment in yaw
×10^11
[Nm]
5
0
-5
0
1
2
3
4
5
6
7
8
Time [s]

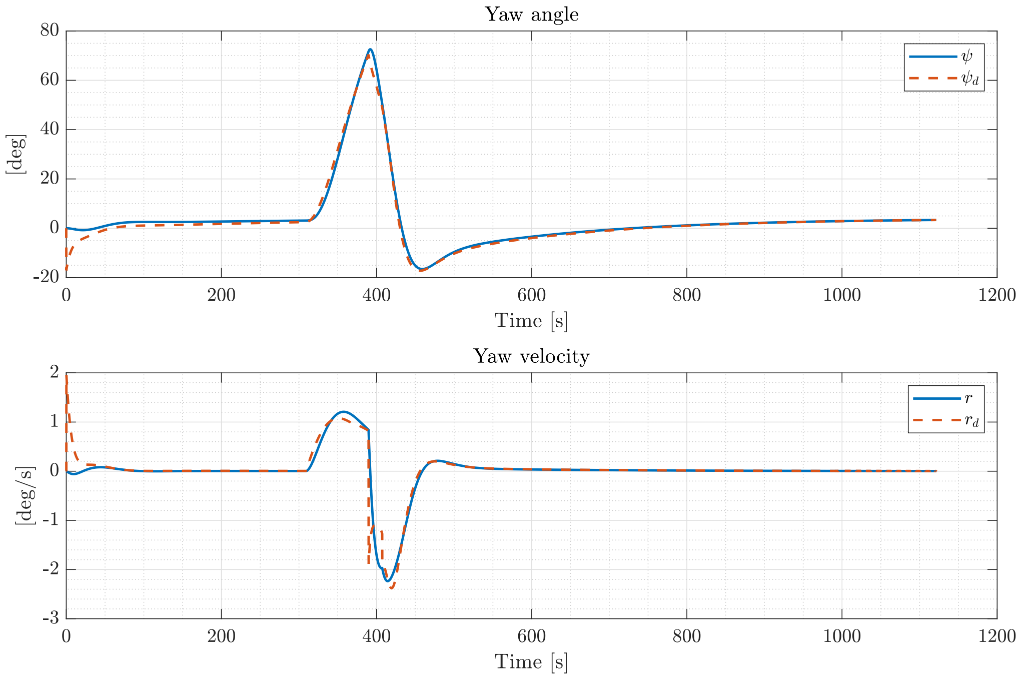
D Figures from Section 7.5.1

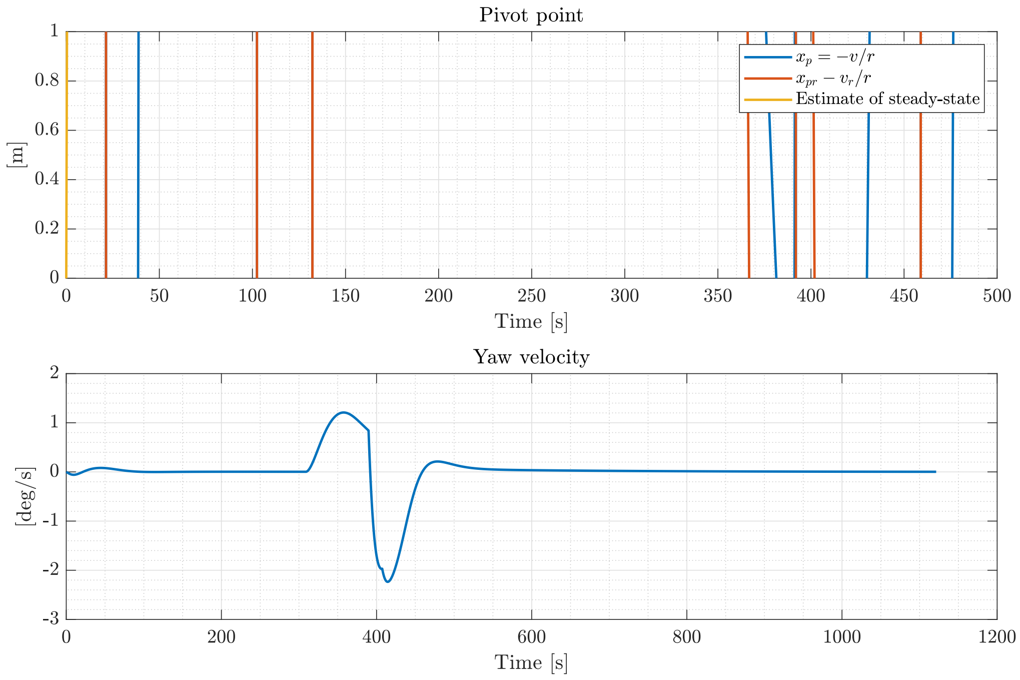


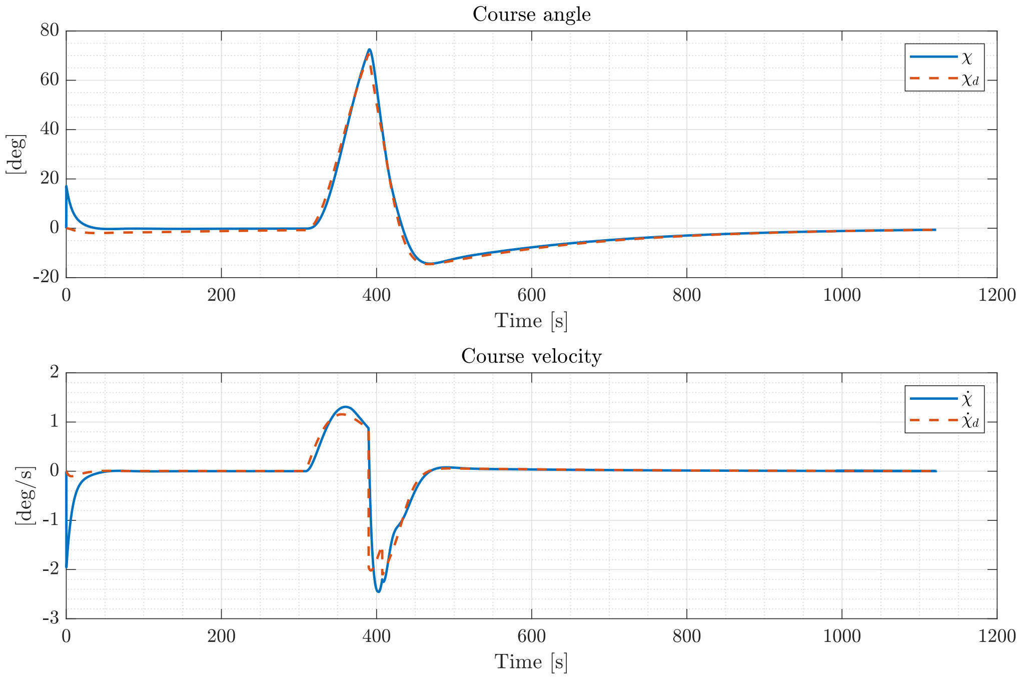


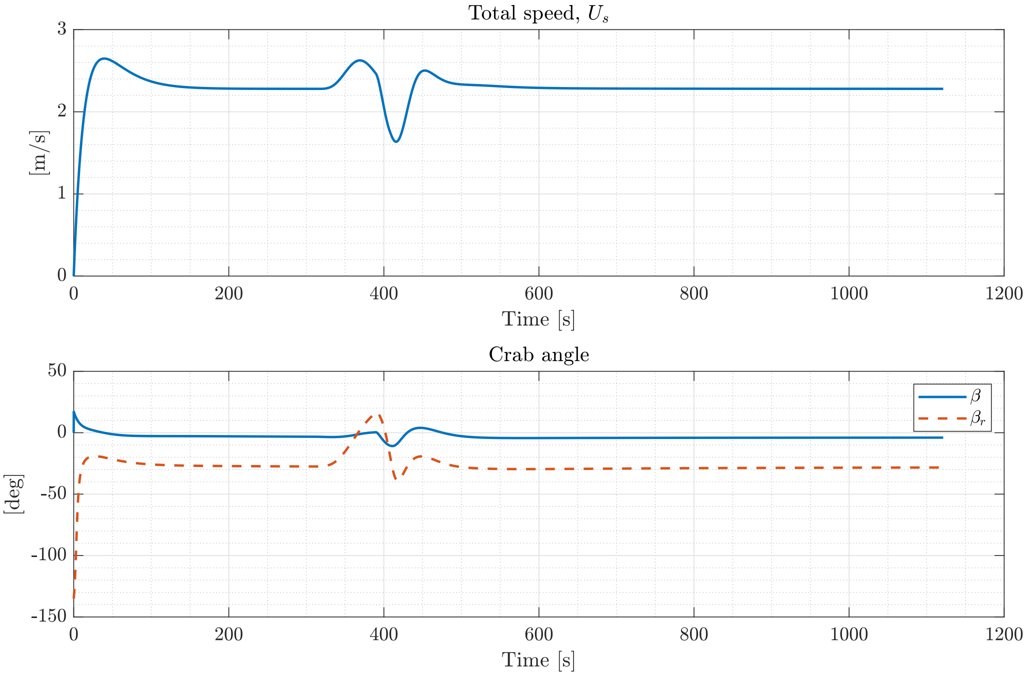

Cross-track position relative to nominal reference path
y [m]
150
100
50
0
y
y_d
0
200
400
600
800
1000
1200
Time [s]
Along-track error with respect to reference path
x_e^p [m]
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
5
0
-5
-10
0
200
400
600
800
1000
1200
Time [s]

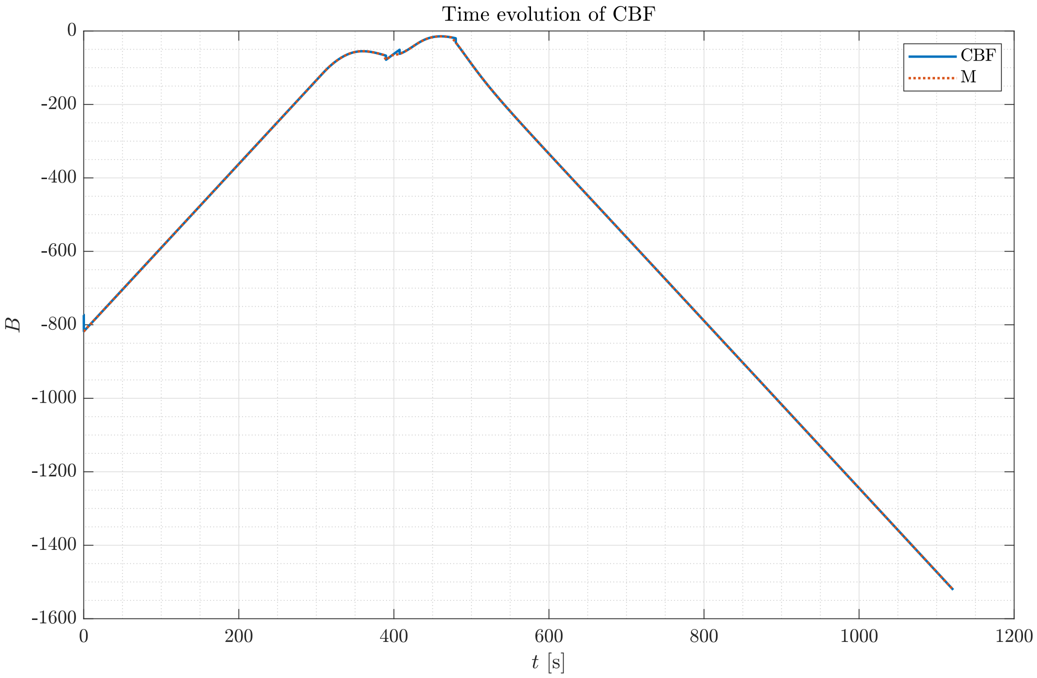

Surge velocity
[m/s]
3
2
1
0
-1
u
u_r
u_d
0
200
400
600
800
1000
1200
Time [s]
Sway velocity
[m/s]
0.5
0
-0.5
-1
-1.5
v
v_r
v_d
0
200
400
600
800
1000
1200
Time [s]

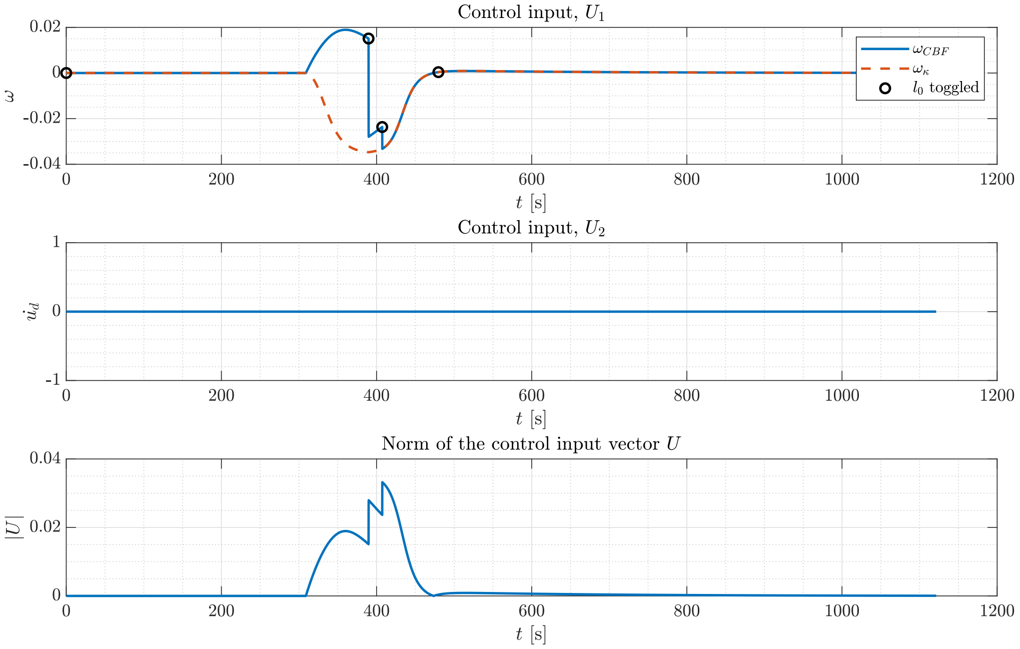

Yaw angle
[deg]
80
60
40
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
ψ
ψ_d
Yaw velocity
[deg/s]
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
Time [s]
r
r_d

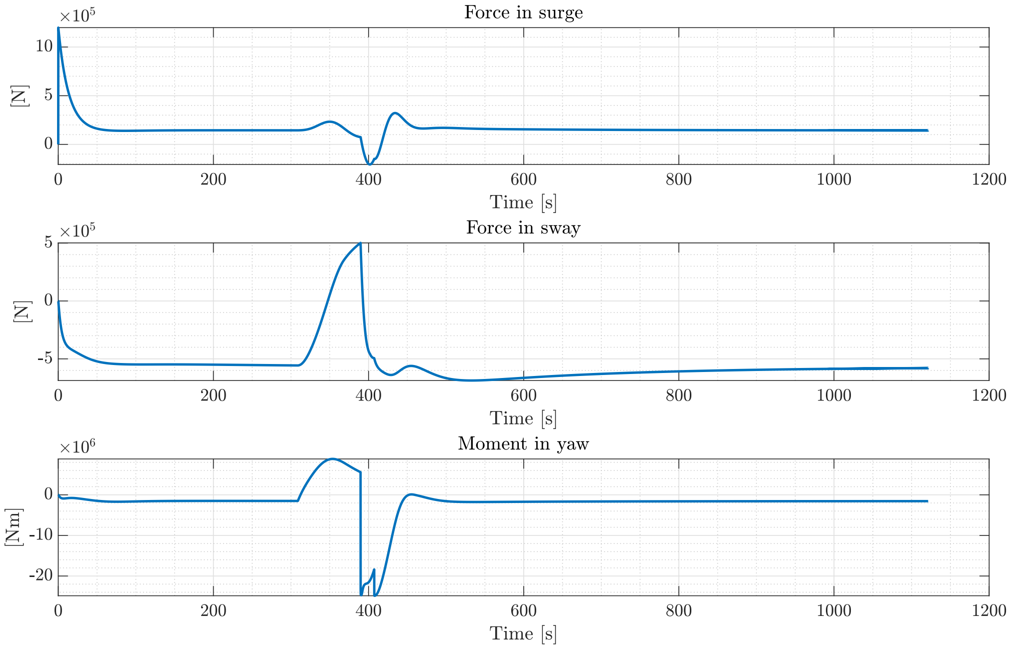

Pivot point
[m]
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
350
400
450
500
Time [s]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
Yaw velocity
[deg/s]
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
Time [s]


Course angle
[deg]
80
60
40
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
χ
χ_d
Course velocity
[deg/s]
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
Time [s]
$\dot{\chi}$
$\dot{\chi}_d$

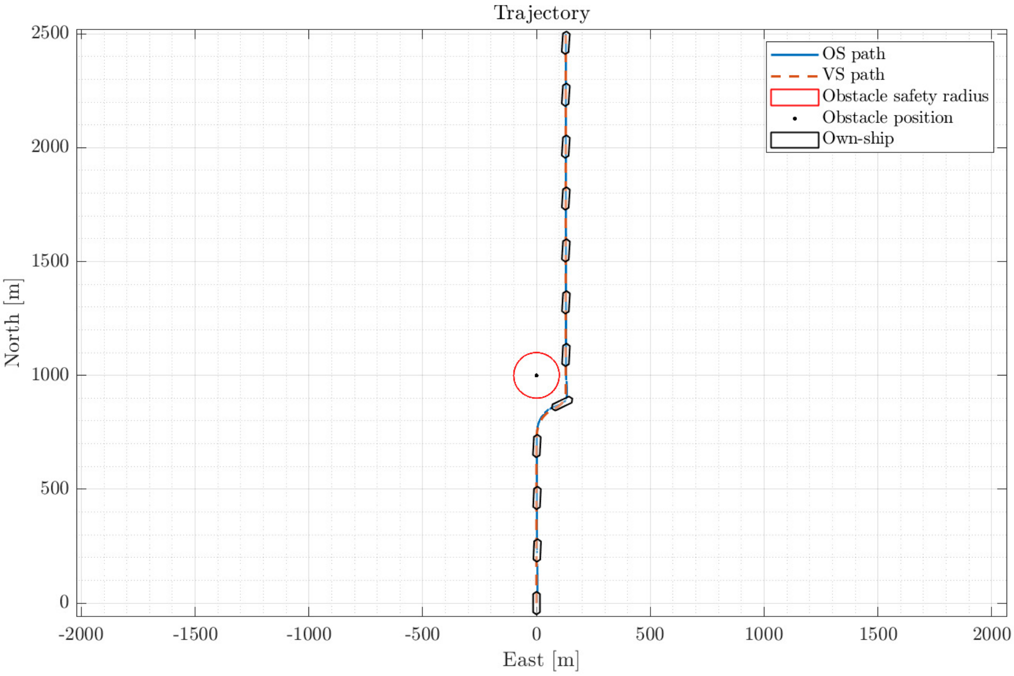

Total speed, U_s
[m/s]
3
2
1
0
0
200
400
600
800
1000
1200
Time [s]
Crab angle
[deg]
50
0
-50
-100
-150
0
200
400
600
800
1000
1200
Time [s]
β
β_r

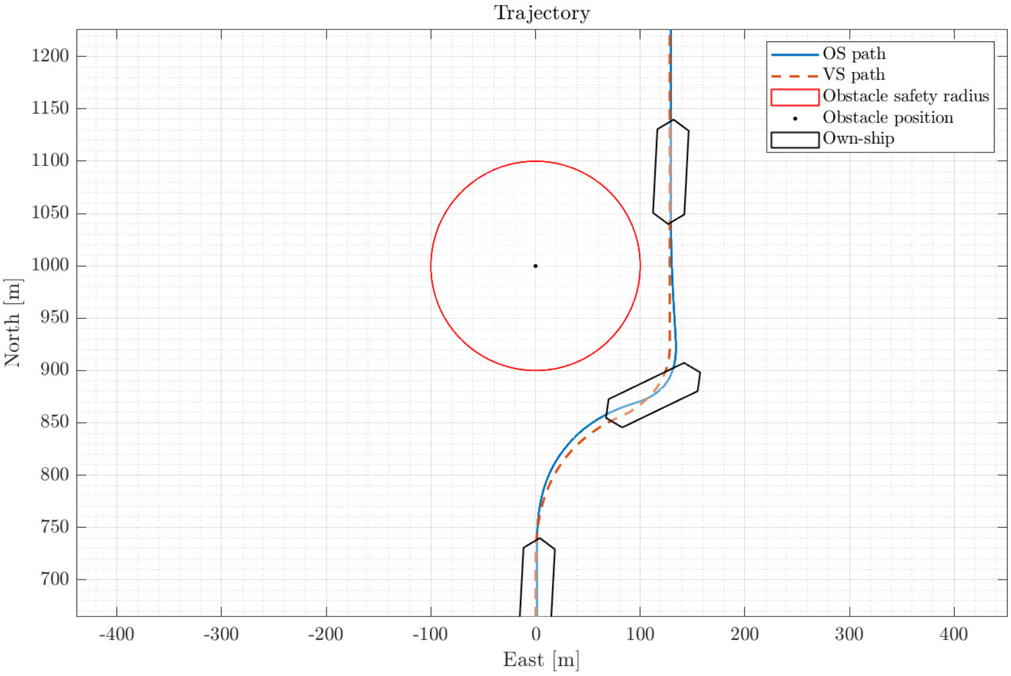

Time evolution of CBF
CBF
M
B
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
200
400
600
800
1000
1200
t [s]

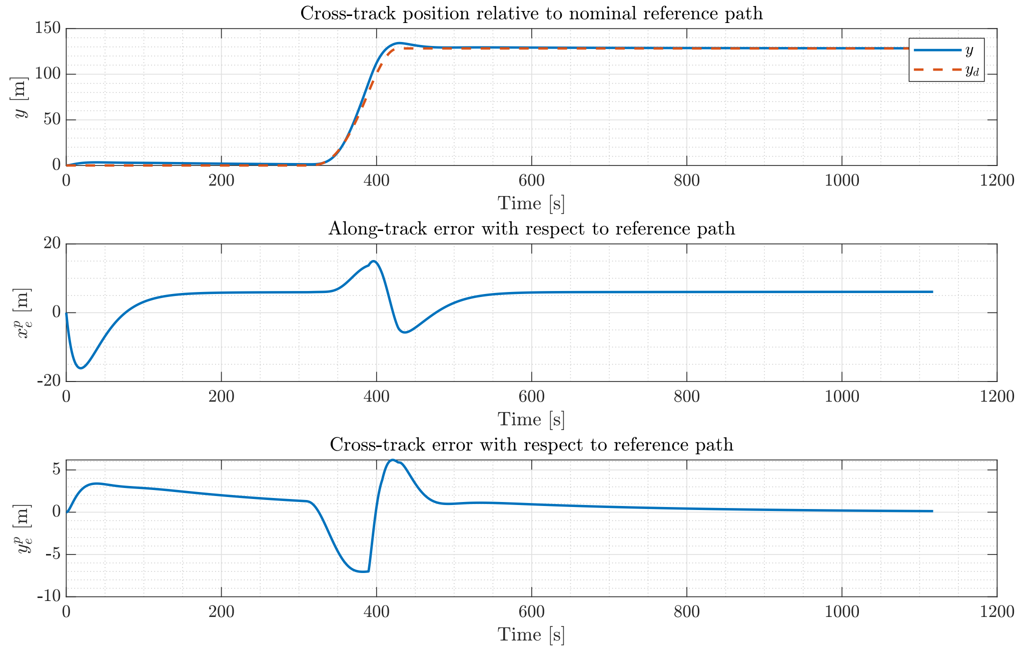

Control input, U_1
ω_{CBF}
ω_{κ}
l_0 toggled
ω
0.02
0
-0.02
-0.04
t [s]
Control input, U_2
u_d
1
0
-1
t [s]
Norm of the control input vector U
$|U|$
0.04
0.02
0
0
200
400
600
800
1000
1200
t [s]

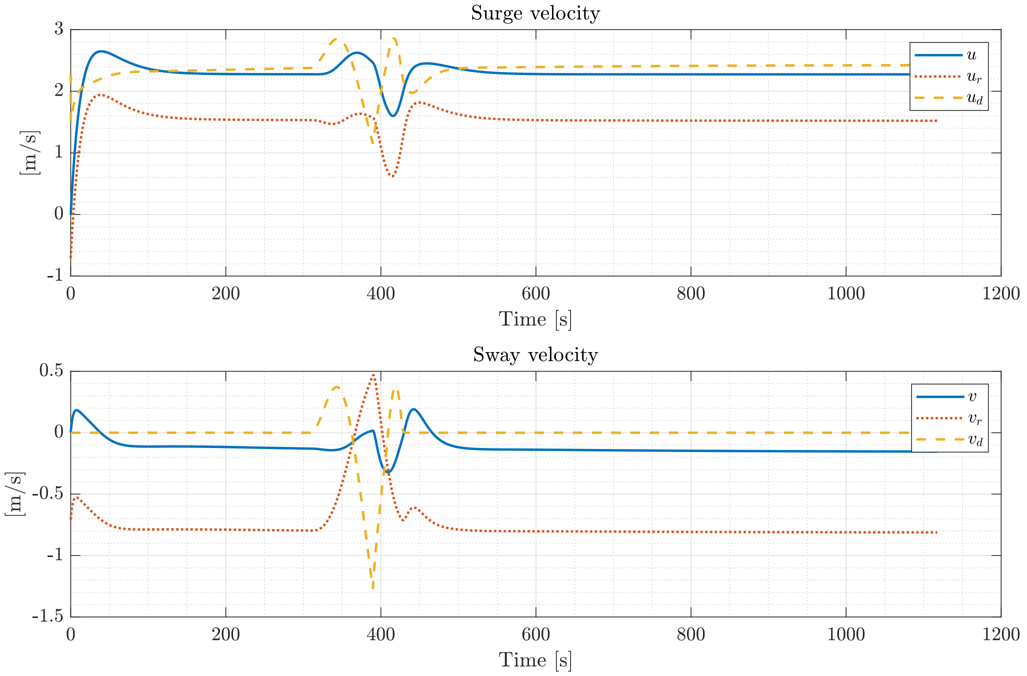

Force in surge
×10^5
10
5
0
[N]
0
200
400
600
800
1000
1200
Time [s]
Force in sway
×10^5
5
0
-5
[N]
0
200
400
600
800
1000
1200
Time [s]
Moment in yaw
×10^6
0
-10
-20
[Nm]
0
200
400
600
800
1000
1200
Time [s]

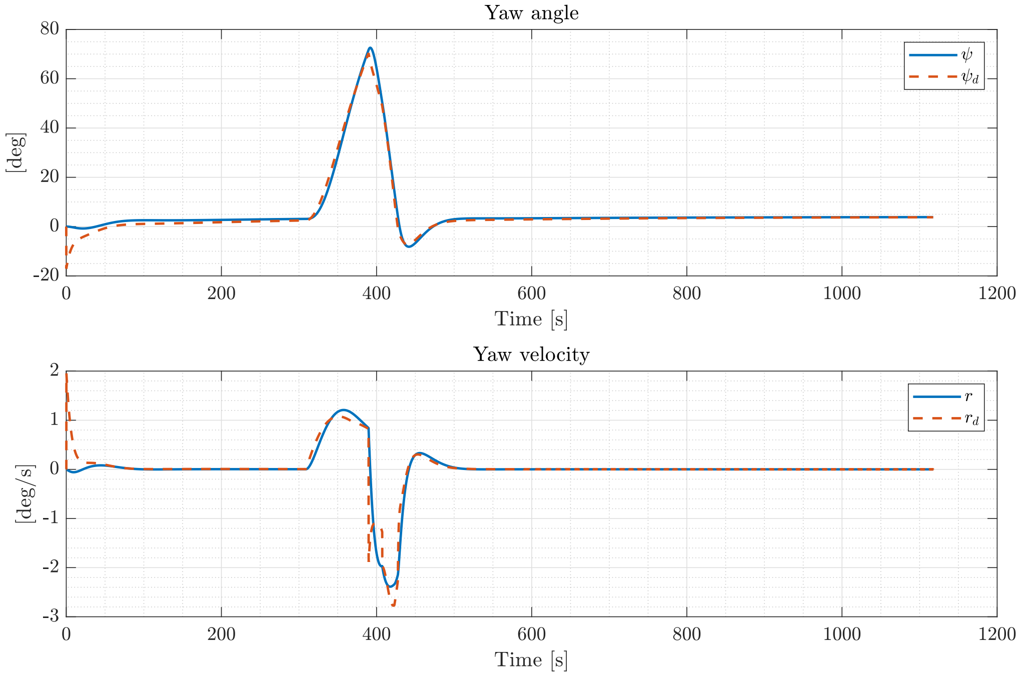
E Figures from Section 7.5.2

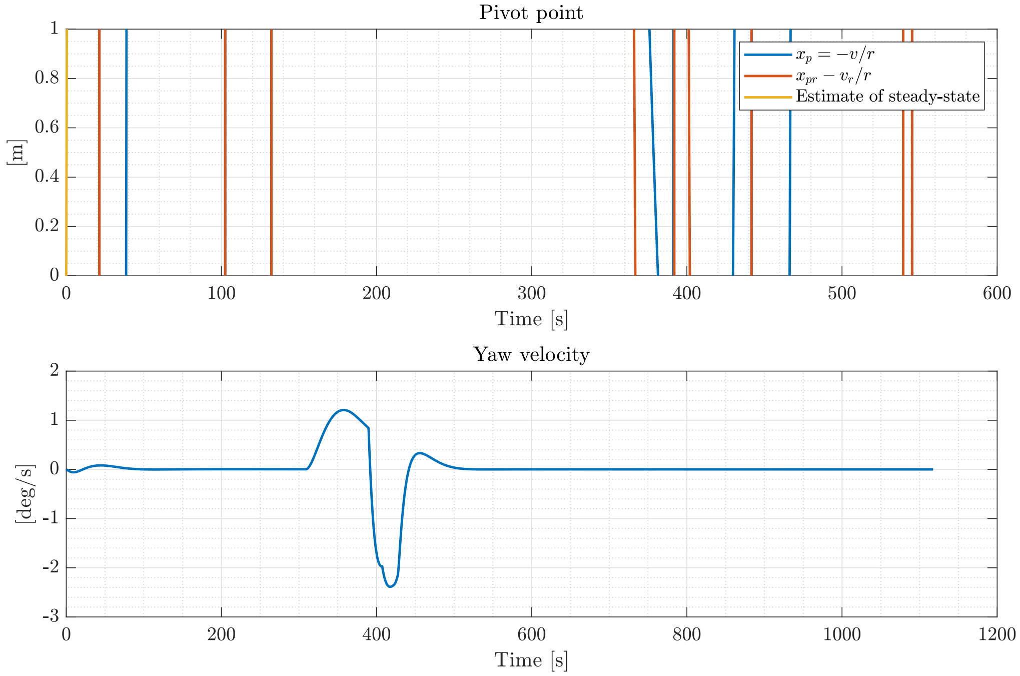


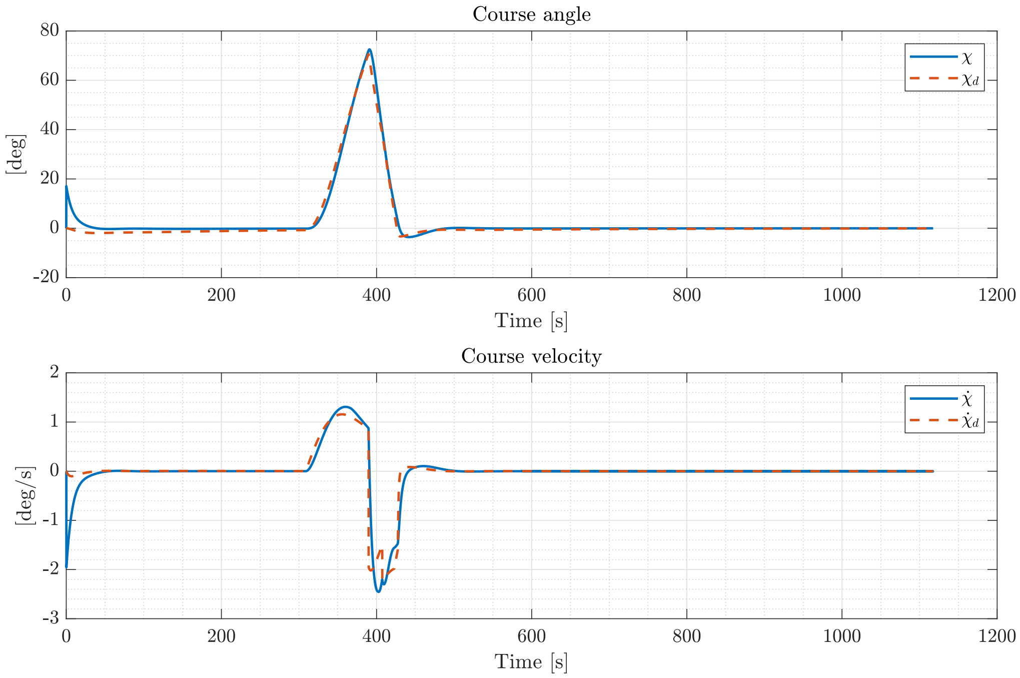


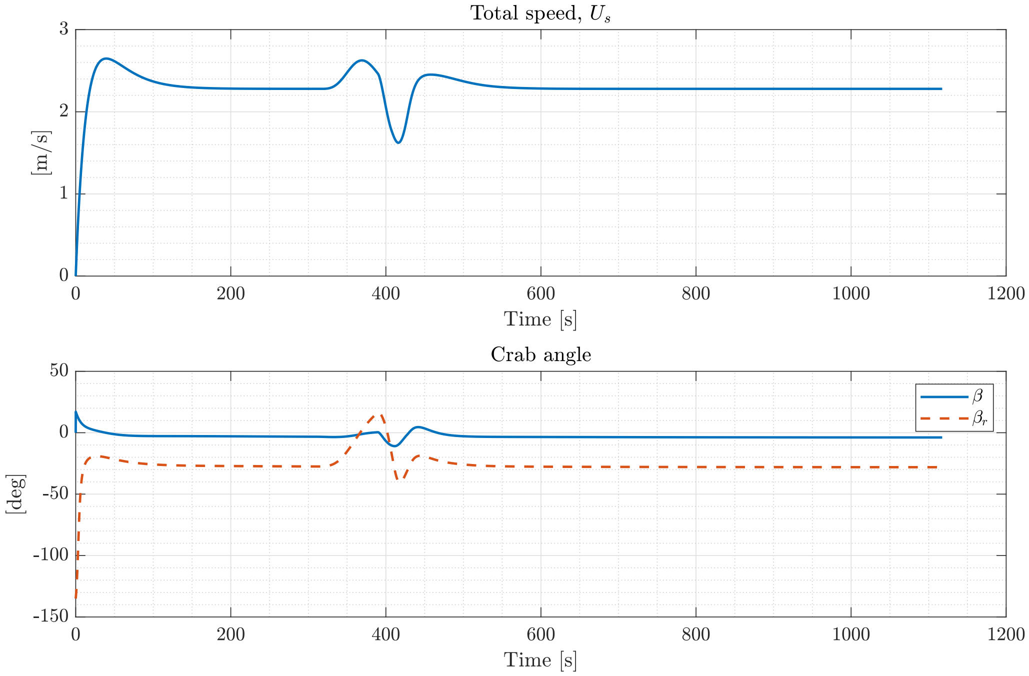

Cross-track position relative to nominal reference path
y [m]
150
100
50
0
0
200
400
600
800
1000
1200
Time [s]
y
y_d
Along-track error with respect to reference path
x_e^p [m]
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
5
0
-5
-10
0
200
400
600
800
1000
1200
Time [s]

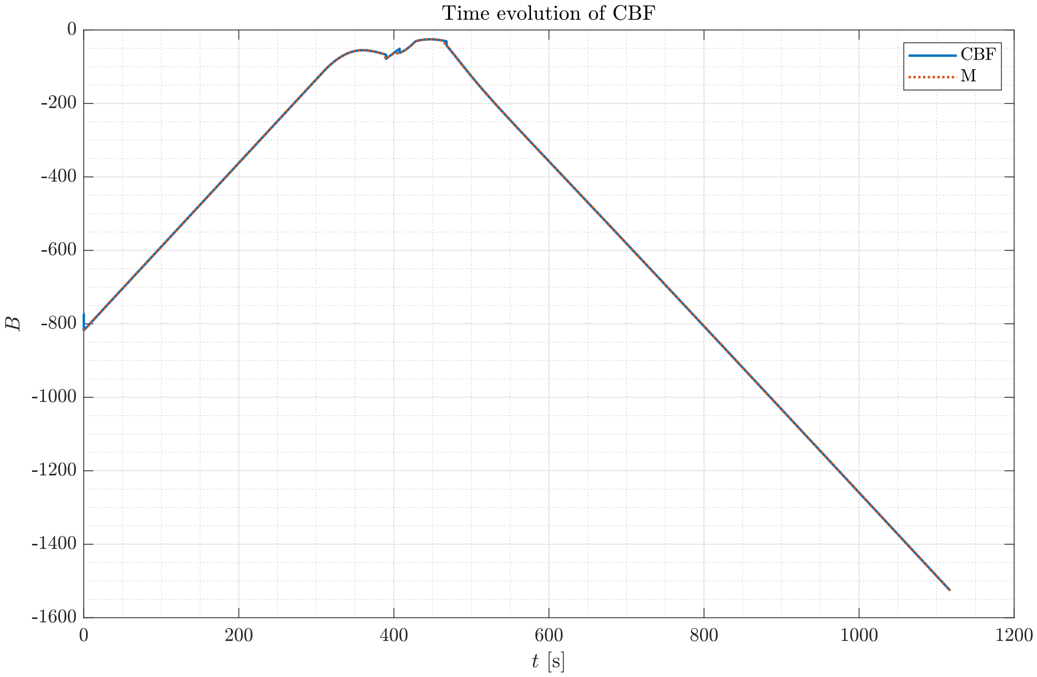

Surge velocity
[m/s]
3
2
1
0
-1
0
200
400
600
800
1000
1200
Time [s]
u
u_r
u_d
Sway velocity
[m/s]
0.5
0
-0.5
-1
-1.5
0
200
400
600
800
1000
1200
Time [s]
v
v_r
v_d

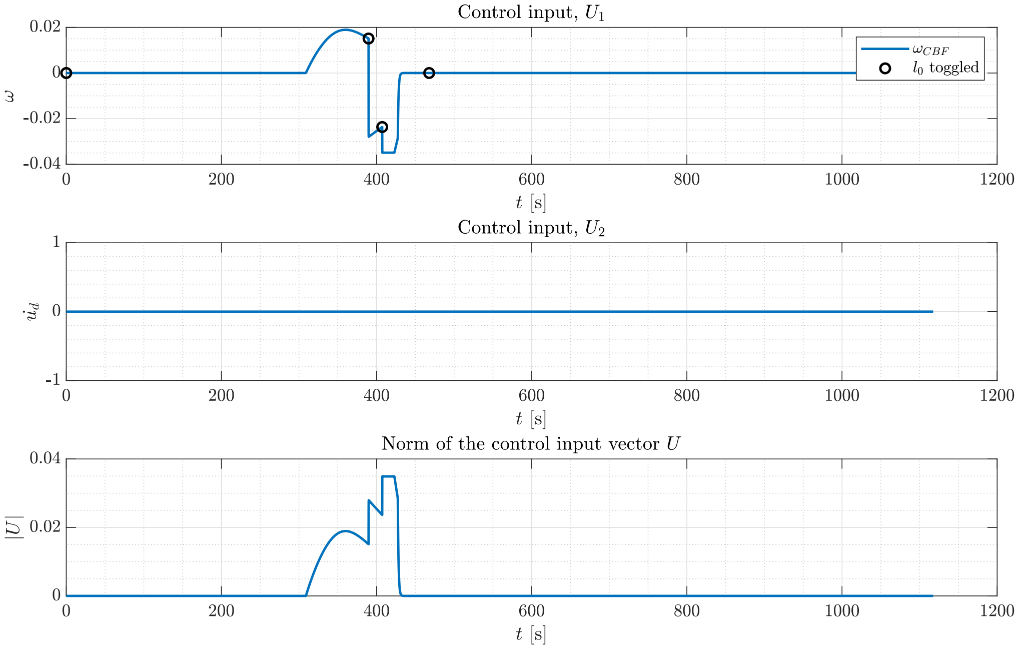

Yaw angle
80
60
40
20
0
-20
[deg]
0
200
400
600
800
1000
1200
Time [s]
ψ
ψ_d
Yaw velocity
2
1
0
-1
-2
-3
[deg/s]
0
200
400
600
800
1000
1200
Time [s]
r
r_d

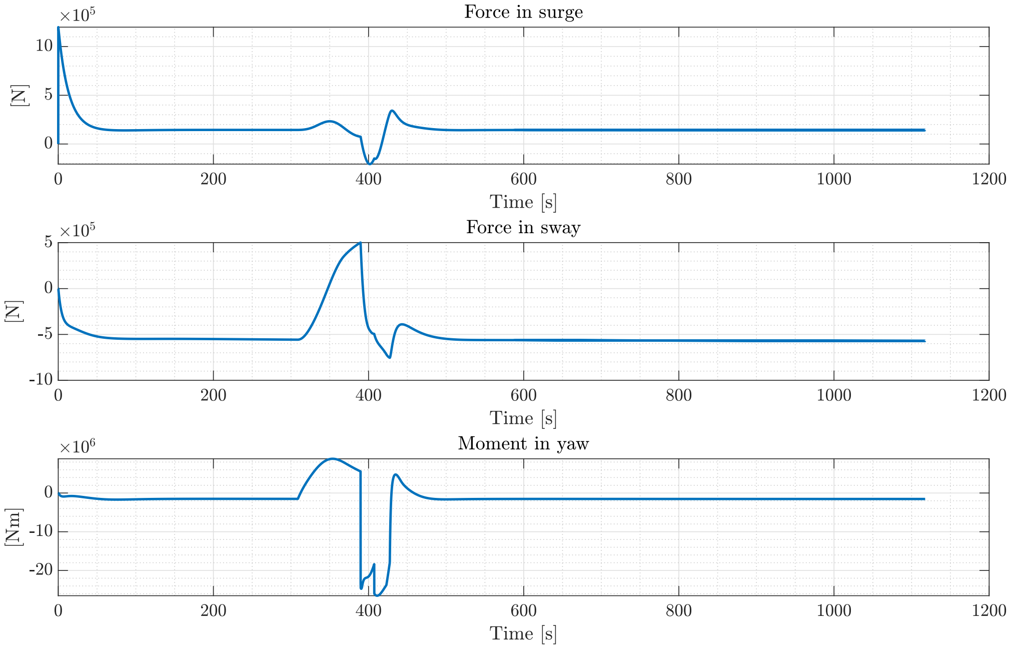

Pivot point
1
0.8
0.6
0.4
0.2
0
[m]
0
100
200
300
400
500
600
Time [s]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
Yaw velocity
2
1
0
-1
-2
-3
[deg/s]
0
200
400
600
800
1000
1200
Time [s]


Course angle
80
60
40
20
0
-20
[deg]
0
200
400
600
800
1000
1200
Time [s]
χ
χ_d
Course velocity
2
1
0
-1
-2
-3
[deg/s]
0
200
400
600
800
1000
1200
Time [s]
$\dot{\chi}$
$\dot{\chi}_d$

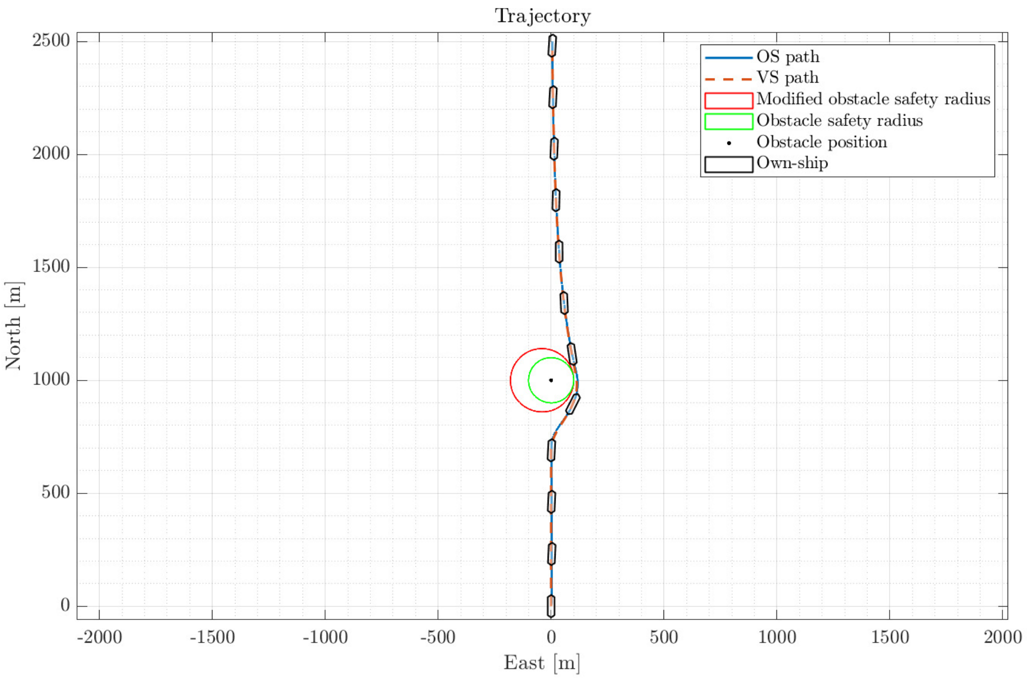

Total speed, U_s
3
2
1
0
[m/s]
0
200
400
600
800
1000
1200
Time [s]
Crab angle
50
0
-50
-100
-150
[deg]
0
200
400
600
800
1000
1200
Time [s]
β
β_r

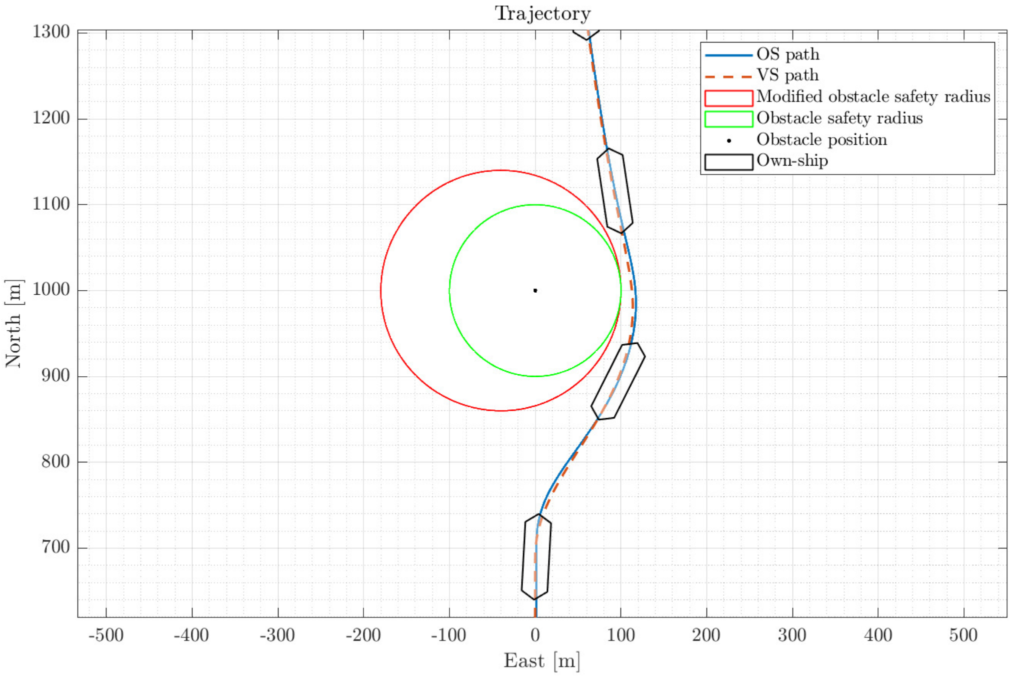

Time evolution of CBF
B
t [s]
CBF
M
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
200
400
600
800
1000
1200

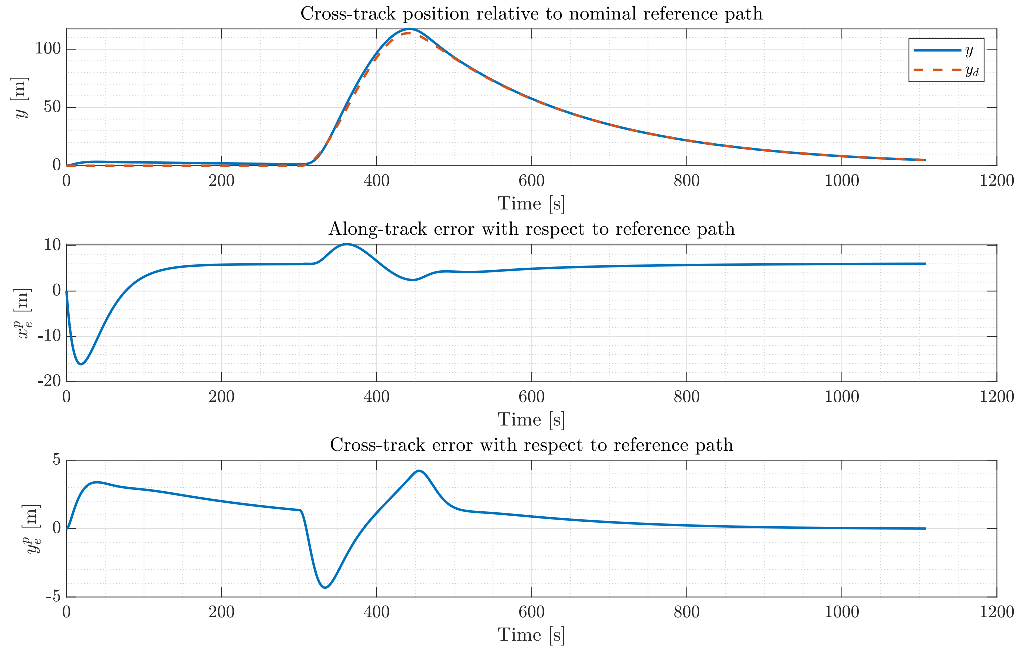

Control input, U_1
ω
0.02
0
-0.02
-0.04
ω_{CBF}
l_0 toggled
t [s]
Control input, U_2
u_d
1
0
-1
t [s]
Norm of the control input vector U
$|U|$
0.04
0.02
0
t [s]
0
200
400
600
800
1000
1200

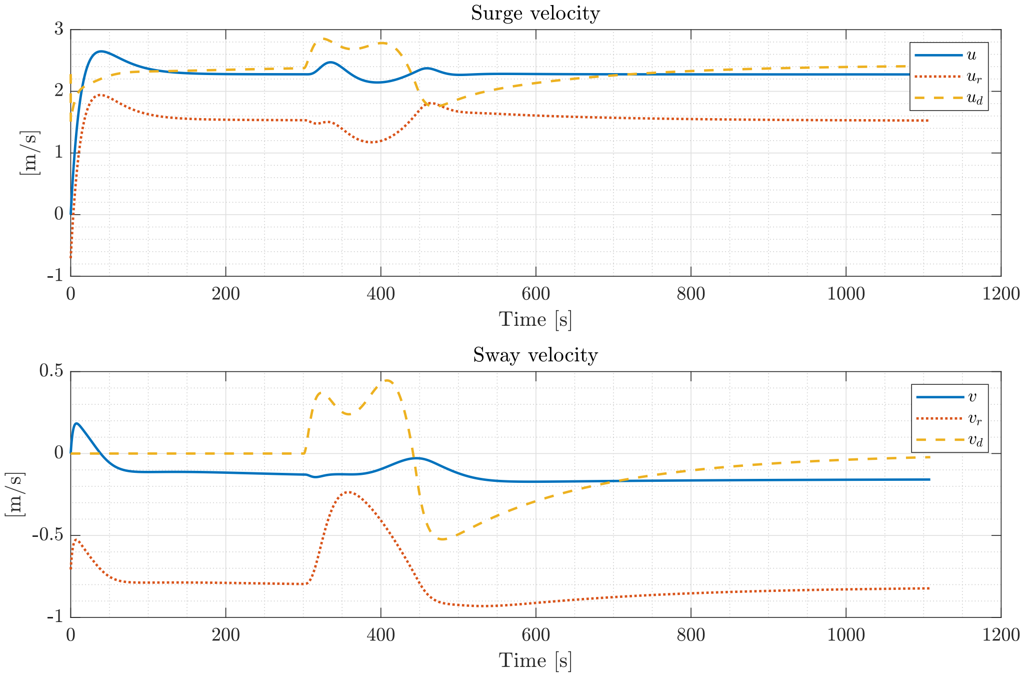

Force in surge
×10^5
10
5
0
[N]
0
200
400
600
800
1000
1200
Time [s]
Force in sway
×10^5
5
0
-5
-10
[N]
0
200
400
600
800
1000
1200
Time [s]
Moment in yaw
×10^6
0
-10
-20
[Nm]
0
200
400
600
800
1000
1200
Time [s]

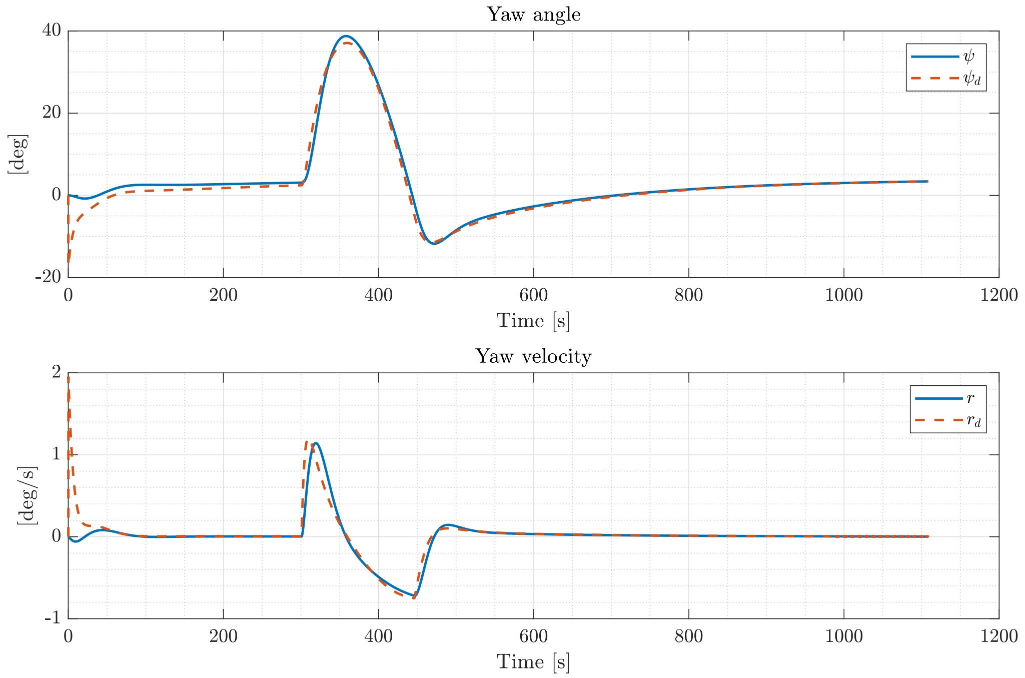
F Figures from Section 7.6.1

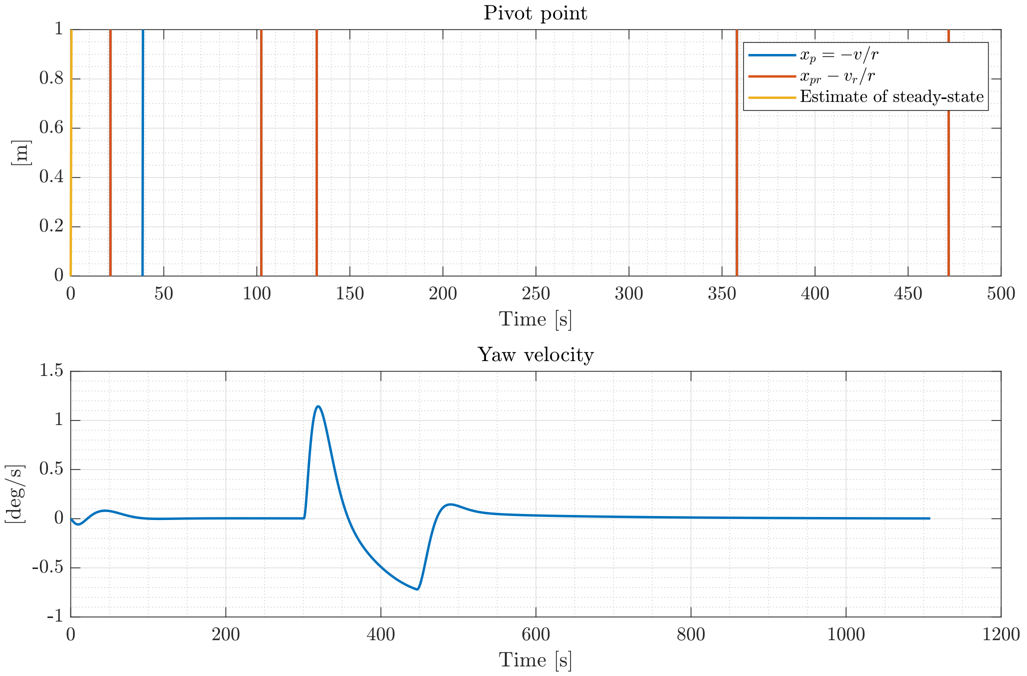


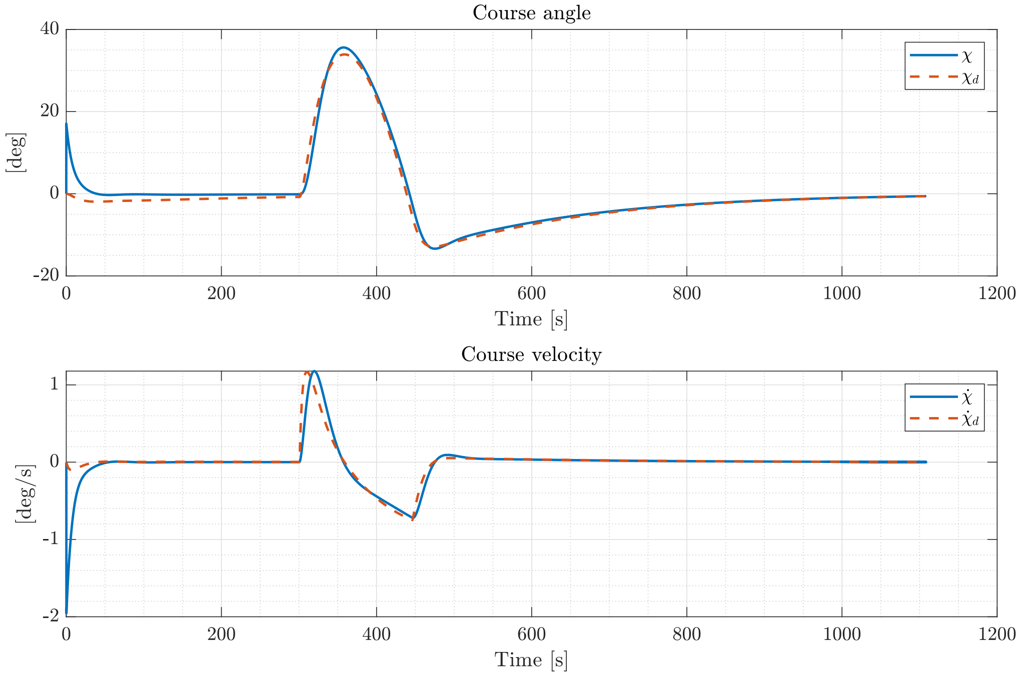


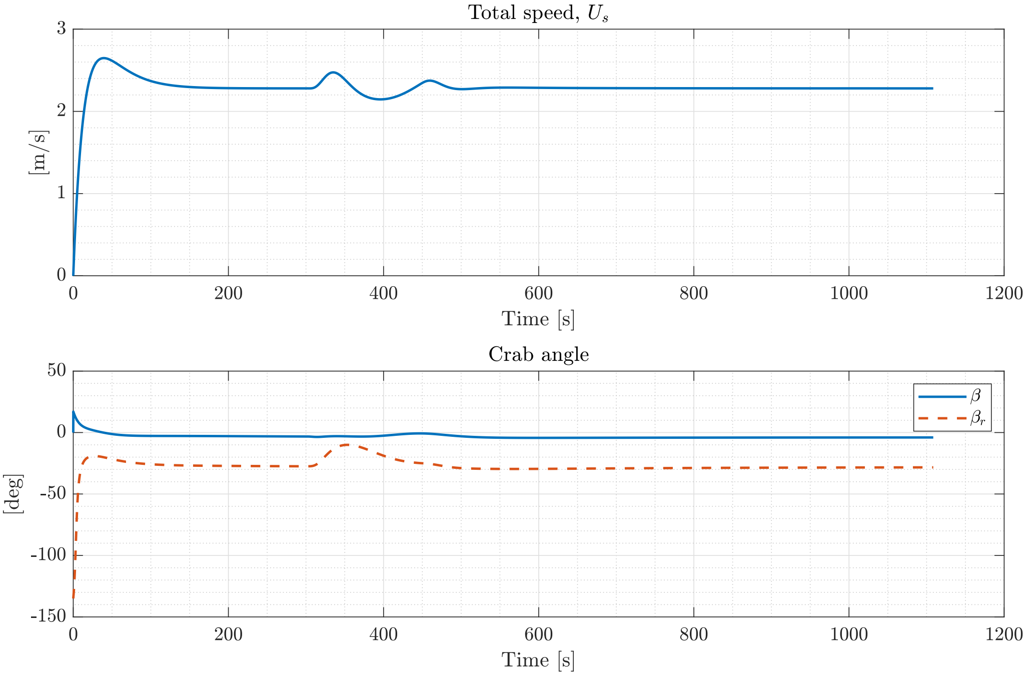

Cross-track position relative to nominal reference path
y [m]
100
50
0
y
y_d
Time [s]
Along-track error with respect to reference path
x_e^p [m]
10
0
-10
-20
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
5
0
-5
0
200
400
600
800
1000
1200
Time [s]

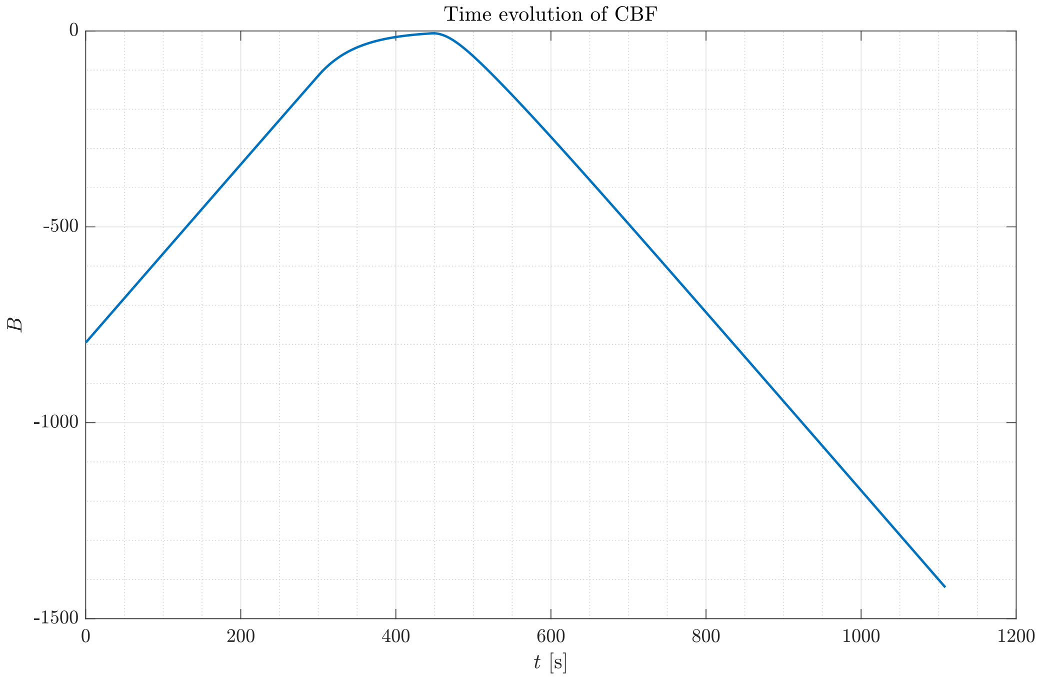

Surge velocity
[m/s]
3
2
1
0
-1
u
u_r
u_d
Time [s]
Sway velocity
[m/s]
0.5
0
-0.5
-1
v
v_r
v_d
0
200
400
600
800
1000
1200
Time [s]

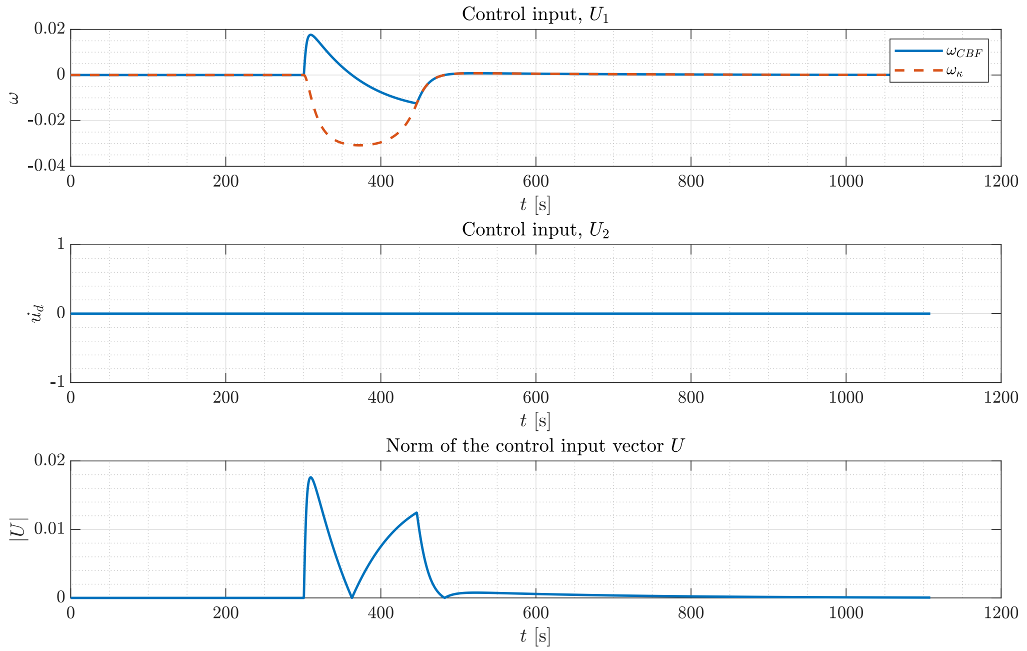

Yaw angle
[deg]
40
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
ψ
ψ_d
Yaw velocity
[deg/s]
2
1
0
-1
0
200
400
600
800
1000
1200
Time [s]
r
r_d

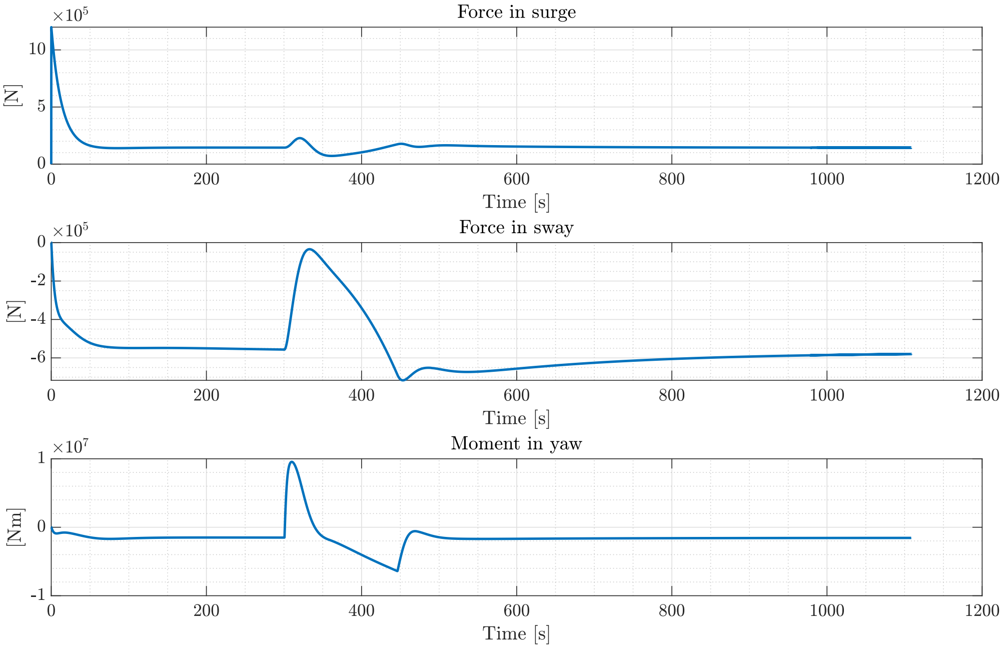

Pivot point
[m]
1
0.8
0.6
0.4
0.2
0
0
50
100
150
200
250
300
350
400
450
500
Time [s]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
Yaw velocity
[deg/s]
1.5
1
0.5
0
-0.5
-1
0
200
400
600
800
1000
1200
Time [s]


Course angle
40
20
0
-20
[deg]
χ
χ_d
0
200
400
600
800
1000
1200
Time [s]
Course velocity
1
0
-1
-2
[deg/s]
$\dot{\chi}$
$\dot{\chi}_d$
0
200
400
600
800
1000
1200
Time [s]

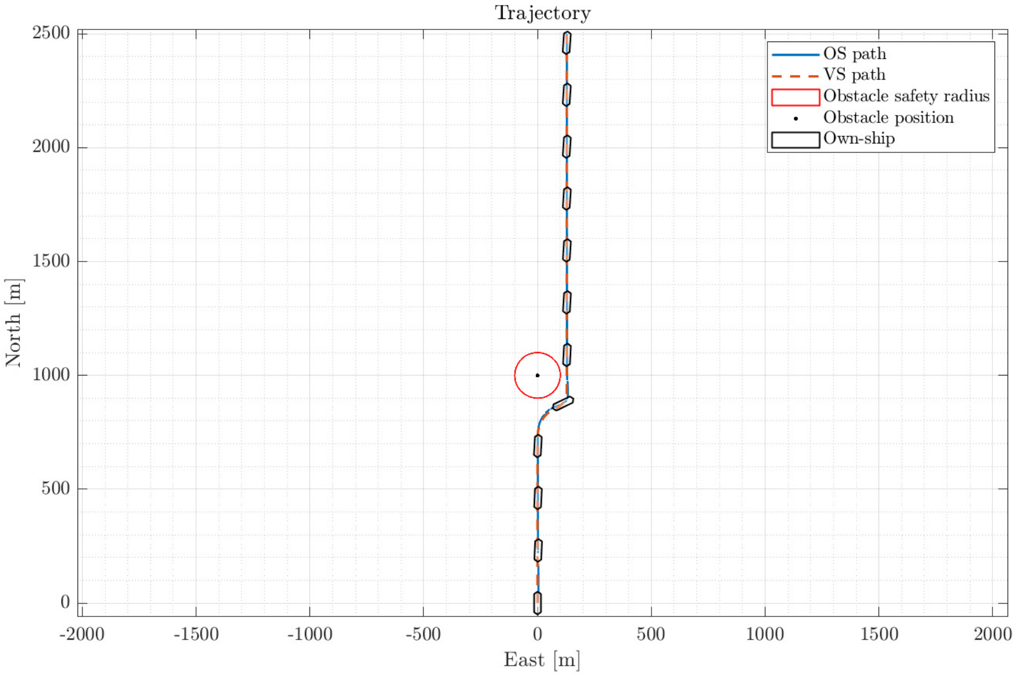

Total speed, U_s
3
2
1
0
[m/s]
0
200
400
600
800
1000
1200
Time [s]
Crab angle
50
0
-50
-100
-150
[deg]
β
β_r
0
200
400
600
800
1000
1200
Time [s]

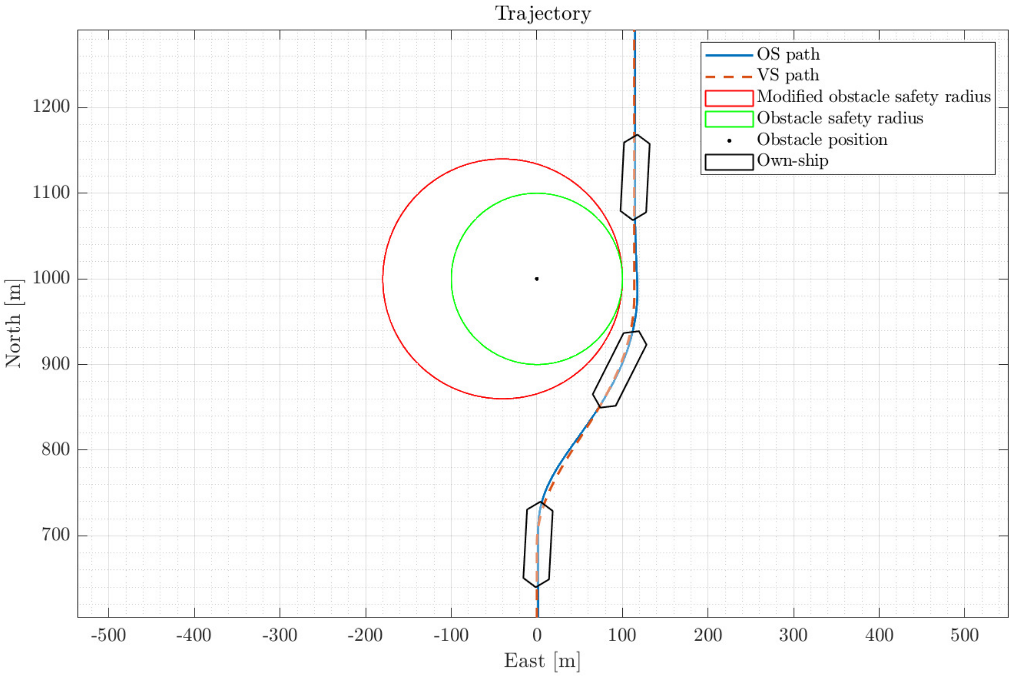

Time evolution of CBF
B
0
-500
-1000
-1500
0
200
400
600
800
1000
1200
t [s]

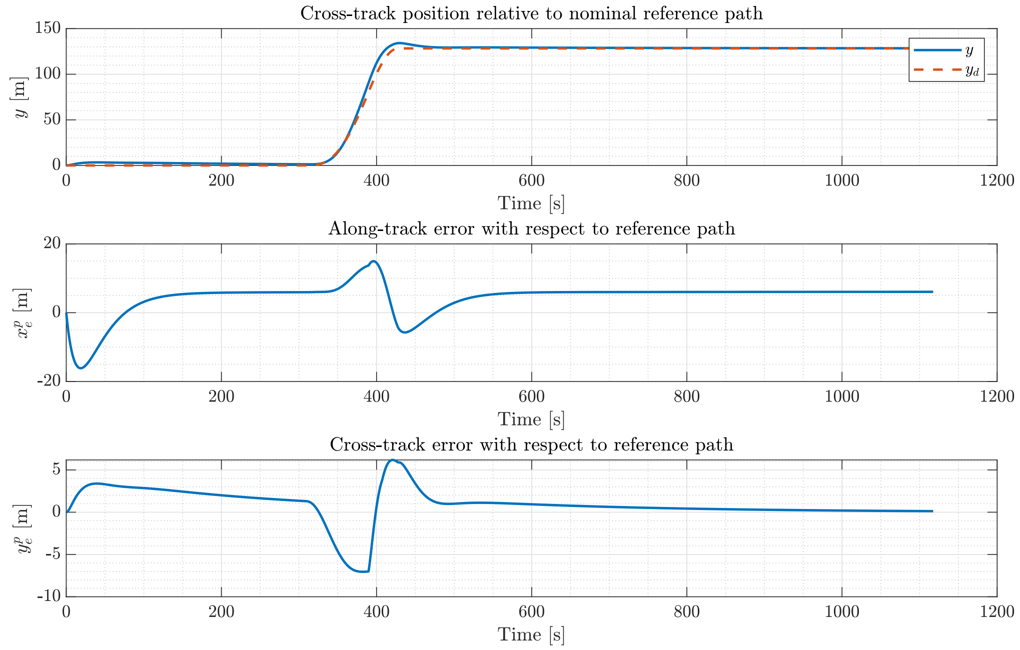

Control input, U_1
ω
0.02
0
-0.02
-0.04
ω_{CBF}
ω_κ
0
200
400
600
800
1000
1200
t [s]
Control input, U_2
u_d
1
0
-1
0
200
400
600
800
1000
1200
t [s]
Norm of the control input vector U
|U|
0.02
0.01
0
0
200
400
600
800
1000
1200
t [s]

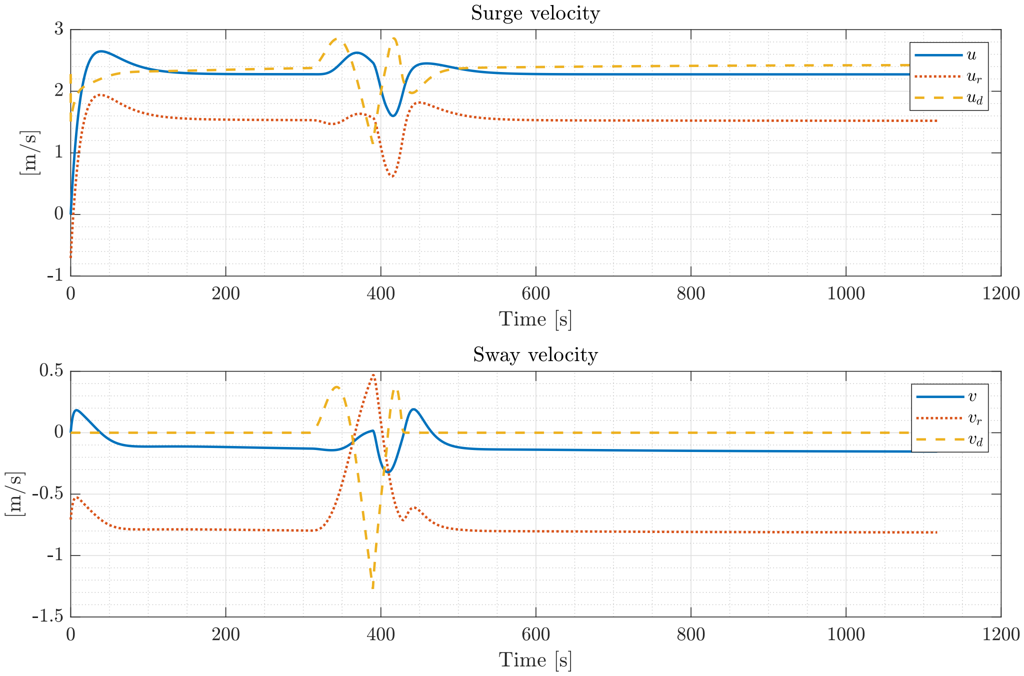

Force in surge
×10^5
10
5
0
[N]
0
200
400
600
800
1000
1200
Time [s]
Force in sway
×10^5
0
-2
-4
-6
[N]
0
200
400
600
800
1000
1200
Time [s]
Moment in yaw
×10^7
1
0
-1
[Nm]
0
200
400
600
800
1000
1200
Time [s]

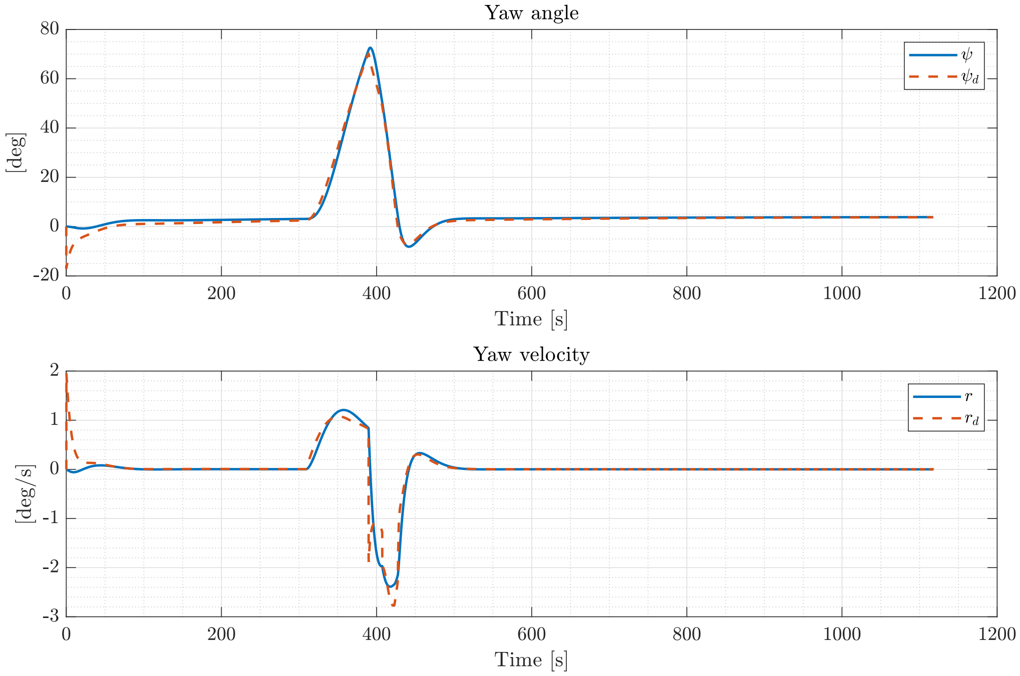
G Figures from Section 7.6.2

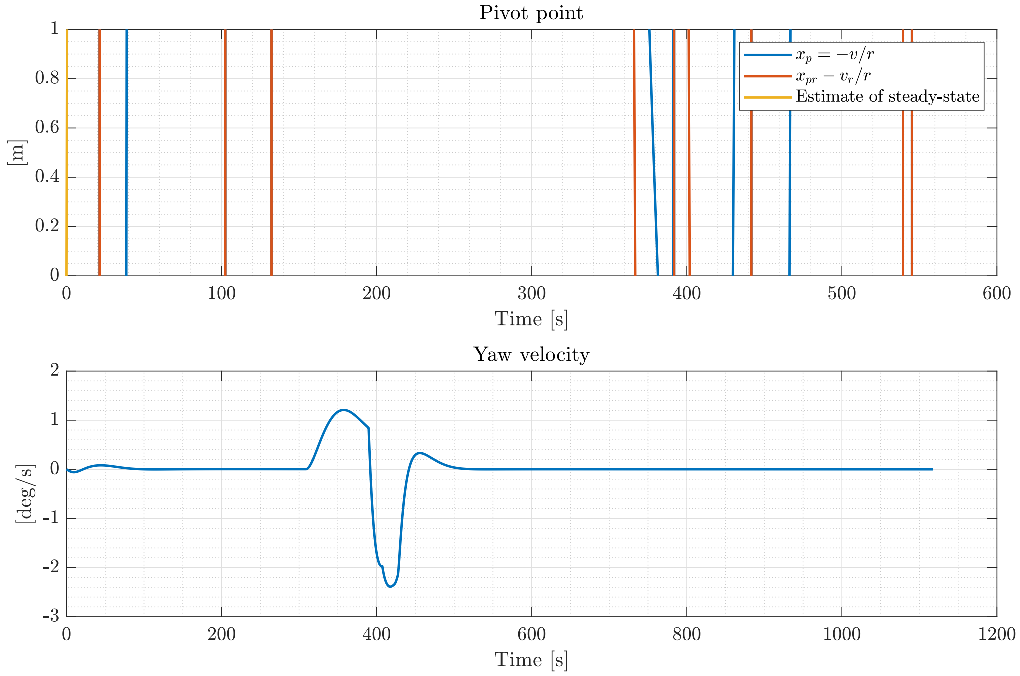


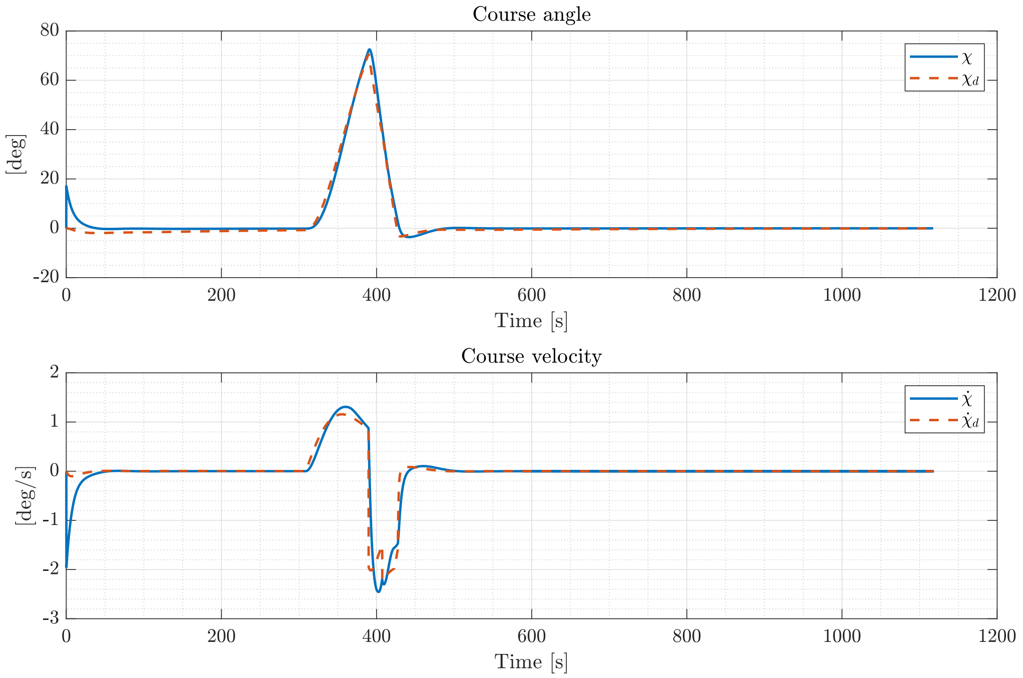


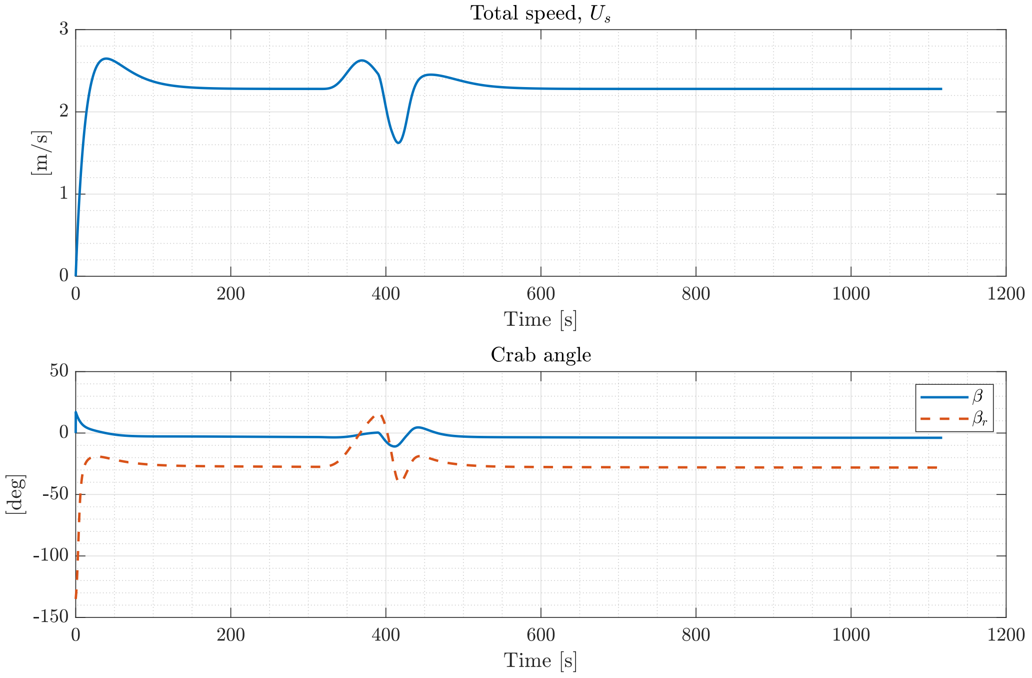

Cross-track position relative to nominal reference path
y [m]
150
100
50
0
y
y_d
Time [s]
Along-track error with respect to reference path
x_e^p [m]
20
0
-20
Time [s]
Cross-track error with respect to reference path
y_e^p [m]
5
0
-5
-10
0
200
400
600
800
1000
1200
Time [s]

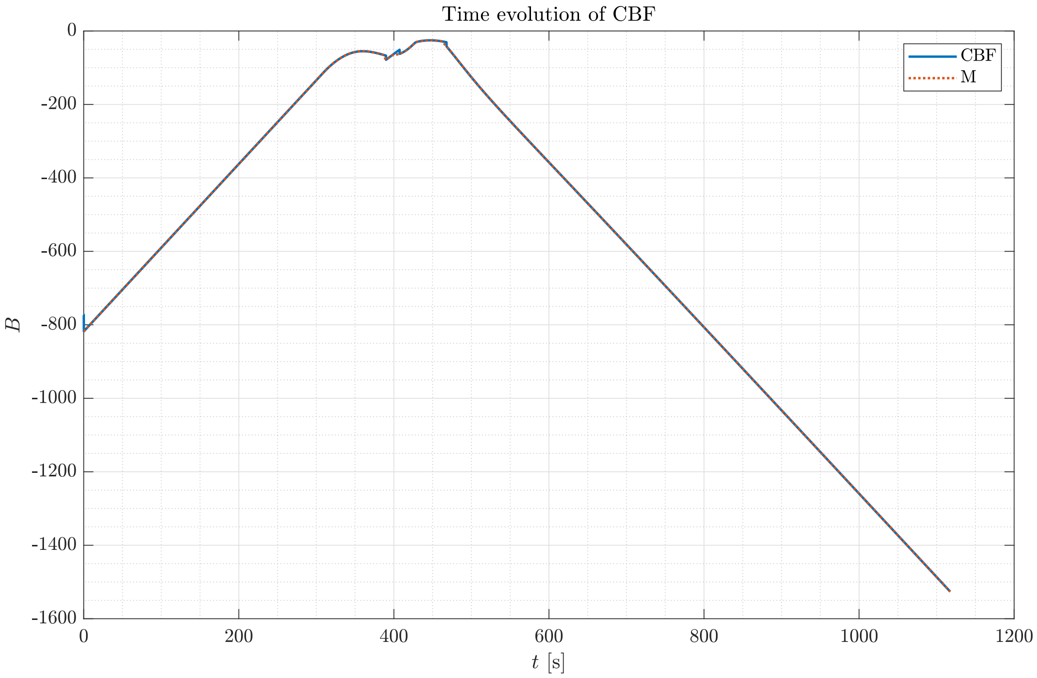

Surge velocity
[m/s]
3
2
1
0
-1
u
u_r
u_d
Time [s]
Sway velocity
[m/s]
0.5
0
-0.5
-1
-1.5
v
v_r
v_d
0
200
400
600
800
1000
1200
Time [s]

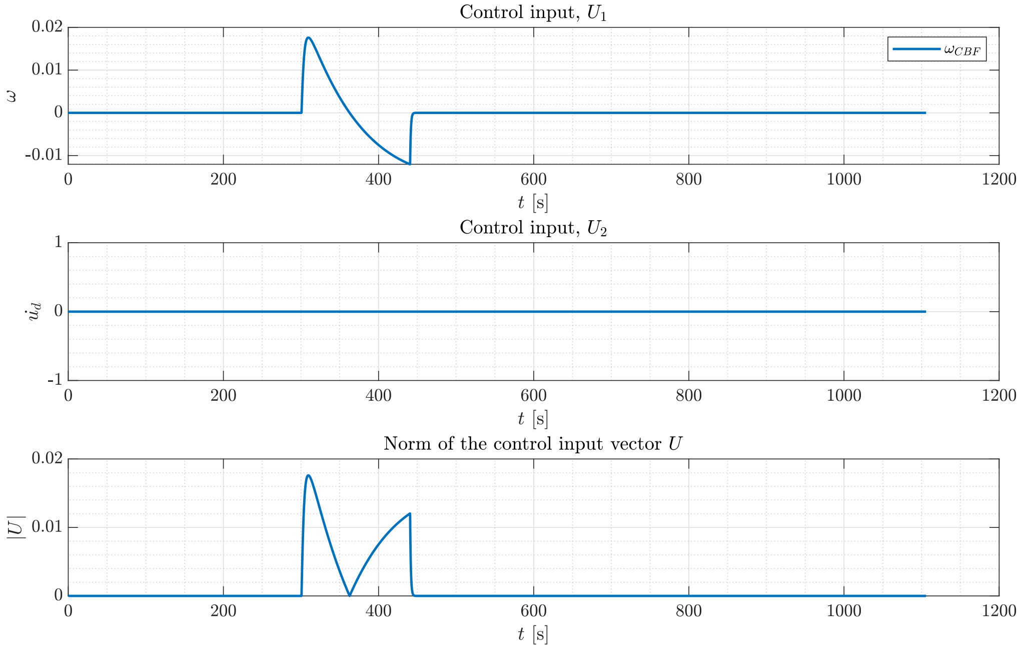

Yaw angle
80
60
40
20
0
-20
[deg]
0
200
400
600
800
1000
1200
Time [s]
ψ
ψ_d
Yaw velocity
2
1
0
-1
-2
-3
[deg/s]
0
200
400
600
800
1000
1200
Time [s]
r
r_d

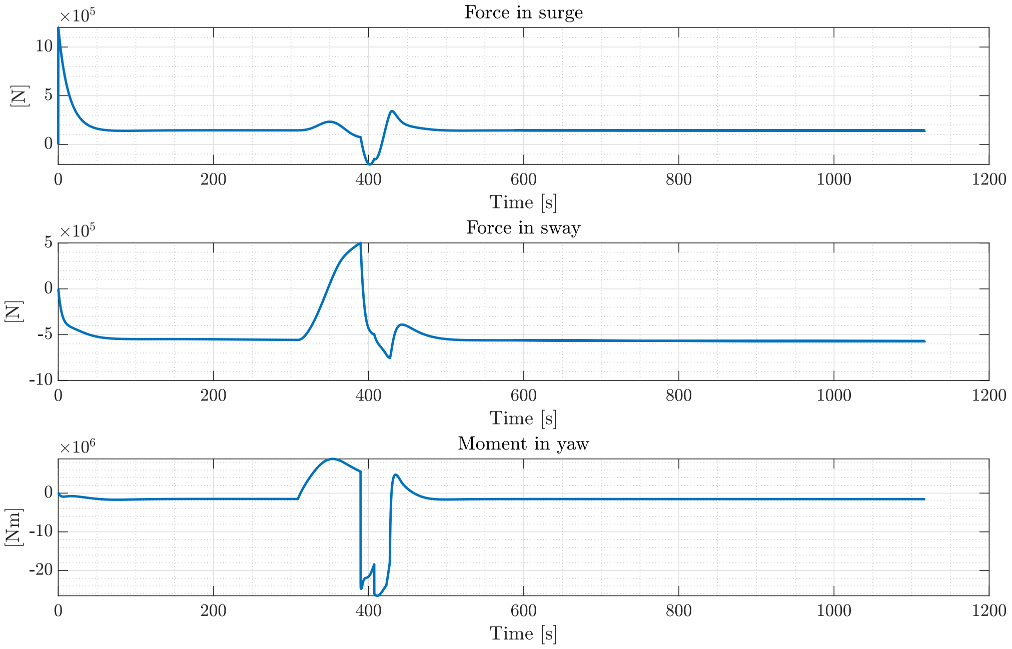

Pivot point
1
0.8
0.6
0.4
0.2
0
[m]
0
100
200
300
400
500
600
Time [s]
$x_p = -v/r$
$x_{pr} - v_r/r$
Estimate of steady-state
Yaw velocity
2
1
0
-1
-2
-3
[deg/s]
0
200
400
600
800
1000
1200
Time [s]


Course angle
χ
χ_d
[deg]
80
60
40
20
0
-20
0
200
400
600
800
1000
1200
Time [s]
Course velocity
$\dot{\chi}$
$\dot{\chi}_d$
[deg/s]
2
1
0
-1
-2
-3
0
200
400
600
800
1000
1200
Time [s]


Total speed, U_s
[m/s]
3
2
1
0
0
200
400
600
800
1000
1200
Time [s]
Crab angle
β
β_r
[deg]
50
0
-50
-100
-150
0
200
400
600
800
1000
1200
Time [s]


Time evolution of CBF
CBF
M
B
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
200
400
600
800
1000
1200
t [s]


Control input, U_1
ω_{CBF}
ω
0.02
0.01
0
-0.01
t [s]
Control input, U_2
$\dot{u}_d$
1
0
-1
t [s]
Norm of the control input vector U
$|U|$
0.02
0.01
0
0
200
400
600
800
1000
1200
t [s]


Force in surge
$\times 10^5$
[N]
10
5
0
0
200
400
600
800
1000
1200
Time [s]
Force in sway
$\times 10^5$
[N]
5
0
-5
-10
0
200
400
600
800
1000
1200
Time [s]
Moment in yaw
$\times 10^6$
[Nm]
0
-10
-20
0
200
400
600
800
1000
1200
Time [s]

Nora Åsheim

Autonomous ship maneuvering with guaranteed safety

NTNU
Norwegian University of
Science and Technology