NTNU
Norwegian University of
Science and Technology

Path Planning for Marine Vehicles using Bézier curves

Simen Vike Lande

Marine Technology
Submission date: June 2018
Supervisor: Vahid Hassani, IMT

Norwegian University of Science and Technology
Department of Marine Technology

Problem description

Background:
Over the past few years the maritime sector has witnessed an increasing interest in the use of autonomous ships, and in particular Autonomous Surface Vehicles (ASV) in complex applications with high associated risks. The development of autonomous ships provides more cost-effective and environmentally friendly vessel types. One important aspect in the design and operation of ASVs, is guidance and navigation in congested waterways.

The objective of this thesis is to contribute to the development of a new generation of path planning, that incorporates in its formulation the dynamics of the vehicle, and extra data made available by on board sensors about obstacle and other vehicles in its vicinity. The Bézier curve will be used as the basis in the path generation and, differential flatness will be used to incorporate the dynamics of the vehicle.

Work description:

1. Perform background research and provide relevant information on:
   (a) The Bézier curve and its formulation.
   (b) Modeling of surface vessels.
   (c) The differential flatness property.
2. Develop a path planning algorithm for an underactuated surface vessel that:
   (a) Enables the generation of a path between two points.
   (b) Utilizes the differential flatness property of the vehicle to assign a cost to each path.
   (c) Generates a path with continuous unit tangent vector and curvature.
   (d) Incorporates information about obstacles and maximum curvature.
3. Implement the proposed path planning algorithm in MATLAB.
4. Explore the behavior of the suggested path planning algorithm through numerical simulations. Present and discuss the results.

Supervisor: Vahid Hassani

Summary

The maritime sector have over the last few years witnessed a growing demand for the development of Autonomous Surface Vehicles (ASV), capable of performing complex tasks with high associated risk. Central in this development is the need for a new generation of advanced guidance, navigation and control (GNC) systems, enabling the ASV to work in any unstructured environment without human supervision. To that end, the availability of efficient and intelligent path planning algorithms are of paramount importance. The present thesis aims to contribute to this development, and proposes a path planning algorithm incorporating in its formulation, the dynamics of the vehicle and information about obstacles and other vehicles in its vicinity.

The proposed path generation algorithm is formulated within the framework of optimization, exploiting Bézier curves as the basis for the generation of a rich set of paths. Further, by the use of the differential flatness property of the vehicle, a cost is assigned to each path, reflecting the dynamic capabilities of the vehicle on that path. The proposed algorithm is capable of generating geometric and parametric continuous paths, while accounting for environmental constraints such as obstacles. Furthermore by constraining the curvature of the path, the algorithm ensures that the turning radius never exceeds the physical limitations of the the vehicle. The present thesis includes a description of the implementation of the proposed algorithm in MATLAB®, commenting on the some of the decisions made in this process.

Finally, a series of numerical simulations are presented, illustrating the efficacy and capabilities of the proposed algorithm. The simulation results presented, differ in boundary conditions and environments, thus highlighting some of the versatility of the proposed algorithm. In the obtainment of results, an issue related to the reliability of the algorithm were reveled, thus further revealing a direction for future work. On this remark, albeit with some persisting challenges, it is concluded that the Bézier curve is a worthy candidate for future considerations, offering a vast amount of possibilities and versatility.

Sammendrag

Den maritime sektoren har i løpet av de siste årene vært vitne til en økende interesse for utvikling av autonome overflate fartøy (ASV), som er i stand til å utføre komplekse operasjoner med en høy tilknyttet risiko. Sentralt i denne utviklingen er behovet for en ny generasjon av avanserte veilednings-, navigasjons- og kontrollsystemer (GNC), som muliggjør for ASVen å arbeide i et ustrukturert miljø uten menneskelig tilsyn. I denne sammenheng er det et pressende behov for utvikling av effektive og intelligente baneplanleggingsalgoritmer. Denne avhandling har som hensikt å bidra til denne utviklingen, og foreslår en baneplanleggingsalgoritme som i sin formulering inkorporerer fartøyets dynamikk, og informasjon om hindringer og andre fartøy i nærheten.

Den foreslåtte baneplanleggingsalgoritmen er formulert innenfor rammen av optimalisering, og tar i bruk Bézier kurver i genereringen av et rikt sett av baner. Videre ved bruk av fartøyets differensiell flathetsegenskap tildeles en kostnad til hver bane, som reflekterer de dynamiske egenskapene til fartøyet på denne banen. Den foreslåtte algoritmen er i stand til å generere geometriske og parametriske kontinuerlige baner, samtidig som miljømessige begrensninger som hindringer taes i betraktning. Videre sikrer algoritmen at svingradiusen til banen aldri overskrider fartøyets fysiske begrensninger, ved å begrense krumningen til banen. Denne avhandling inneholder også en beskrivelse av implementeringen av den foreslåtte algoritmen i MATLAB®, inkludert noen kommentarer til beslutninger som ble fattet.

Til slutt presenteres en rekke numeriske simuleringer som illustrerer effektiviteten og nytteverdien til den foreslåtte algoritmen. Resultatene som fremlegges er utformet slik at problembeskrivelsen varierer for simuleringene, både i forhold og miljø, og dermed fremheves noe av allsidigheten til den foreslåtte algoritmen. Under innhenting av resultatene ble et problem relatert til påliteligheten til algoritmen tydeliggjort, og dermed ble også en fremtidig arbeids retning funnet. På denne bemerkningen, til tross for de vedvarende utfordringene, konkluderes det med at Bézier kurven er en verdig kandidat for bruk i fremtidens baneplanleggingsalgoritmer.

Preface

This master thesis has been written during the spring semester of 2018, at the Department of Marine Technology at the Norwegian University of Science and Technology (NTNU).

The writing of this thesis would have been much harder without the support of my family and friends, whose advice and support have been truly treasured in this endeavor. Among them who deserve special acknowledgment, is my supervisor Vahid Hassani, with whom I have had several profitable and interesting discussions throughout this last year. I would also like to thank my good friend Aslak Seines, for helping me proofread this thesis.

The work which has been conducted during the last few months have been both interesting and challenging, and it is with mixed feelings I finish off the last adjustments. The challenges which has arisen has led to several hours being spent in frustration, but the joy in overcoming these challenges has however made it all worth it. I am able to say with full sincerity that the work has been rewarding, and I am left with a lot of gratitude for the opportunity I have been given.

Simen Vike Lande

Trondheim, June 9, 2018

Table of Contents

Summary i

Sammendrag iii

Preface v

Table of Contents viii

List of Tables ix

List of Figures xii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Scope of work . . . 3
- 1.3 Limitations . . . 3
- 1.4 List of publications . . . 3
- 1.5 Thesis outline . . . 4

2 Preliminaries 5
- 2.1 Mathematical notations and definitions . . . 5
- 2.2 Abbreviations . . . 5

3 Path parameterization 7
- 3.1 General path parameterization . . . 7
  - 3.1.1 Parametric curves . . . 7
  - 3.1.2 Piecewise parametric curve . . . 7
  - 3.1.3 Continuity . . . 8
  - 3.1.4 Arc length . . . 10
  - 3.1.5 Path curvature . . . 10
- 3.2 Bézier curves . . . 12
  - 3.2.1 History and background . . . 12

3.2.2 General definition . . . 12
3.2.3 Extension to arbitrary interval . . . 15
3.2.4 Subdivision of Bézier curves . . . 15
3.2.5 Degree elevation . . . 15
3.2.6 Derivatives . . . 16

4 System modeling and properties 17
4.1 Vehicle modelling . . . 17
4.1.1 Kinematics . . . 17
4.1.2 Vehicle dynamics . . . 20
4.1.3 Vehicle models . . . 20
4.2 Differential flatness . . . 22
4.2.1 Historical remark . . . 22
4.2.2 Flat systems . . . 23
4.2.3 Flatness and surface vessels . . . 24

5 Path planning and optimization 29
5.1 Related work . . . 30
5.2 Path description . . . 31
5.2.1 Desirable properties . . . 31
5.2.2 Analysis of the Bézier spline . . . 32
5.3 Path optimization . . . 35
5.3.1 Decision variables . . . 35
5.3.2 Objective function . . . 36
5.3.3 Constraints . . . 37
5.4 Implementation . . . 40
5.4.1 Simulation environment . . . 40
5.4.2 Initialization . . . 40
5.4.3 Path representation . . . 41
5.4.4 Constraints and objective . . . 41
5.4.5 Solver and options . . . 42

6 Numerical simulations and results 43
6.1 Preliminary comments . . . 43
6.2 First simulation scenario . . . 44
6.3 Second simulation scenario . . . 46
6.4 Third simulation scenario . . . 48
6.5 Fourth simulation scenario . . . 50

7 Discussion 55

8 Conclusion and further work 59
8.1 Further work . . . 60

Bibliography 61

Appendix 65

List of Tables

4.1 Notation for motion variables. 18

5.1 Properties of Bézier splines of different degrees. 34
5.2 Parameters for initialization. 40
5.3 Relevant path information for a single curve segment. 41

6.1 Parameters for initialization in the first scenario. 44
6.2 Parameters for initialization in the second scenario. 46
6.3 Parameters for initialization in the third scenario. 48
6.4 Parameters for initialization in the fourth scenario. 50

List of Figures

1.1 Interaction between the modules of the GNC system. 2

3.1 Two parametric curves with G^1 continuity. 9
3.2 Osculating circle at $\mathbf{p}_p(\varpi)$. 11
3.3 The Bernstein polynomials. 14
3.4 A fourth order Bézier curve with the control polygon. 14

4.1 Relationship between course χ, heading ψ and sideslip β. 19

5.1 A fifth order Bézier curve with the control polygon. 34

6.1 Graphical representation of the generated path and obstacles in the first scenario. 44
6.2 Graphical representation of the calculated course angle along the path for the first scenario. 45
6.3 Graphical representation of the calculated curvature along the path for the first scenario. 45
6.4 Graphical representation of the generated path and obstacles in the second scenario. 46
6.5 Graphical representation of the calculated course angle along the path for the second scenario. 47
6.6 Graphical representation of the calculated curvature along the path for the second scenario. 47
6.7 Graphical representation of the generated path and obstacles in the third scenario. 48
6.8 Graphical representation of the calculated course angle along the path for the third scenario. 49
6.9 Graphical representation of the calculated curvature along the path for the third scenario. 49
6.10 Graphical representation of the generated path and obstacles in the fourth scenario (1). 50

6.11 Graphical representation of the calculated course angle along the path for the fourth scenario (1). 51
6.12 Graphical representation of the calculated curvature along the path for the fourth scenario (1). 51
6.13 Graphical representation of the generated path and obstacles in the fourth scenario (2). 52
6.14 Graphical representation of the calculated course angle along the path for the fourth scenario (2). 52
6.15 Graphical representation of the calculated curvature along the path for the fourth scenario (2). 53
6.16 Graphical representation of the geometric and parametric paths in the fourth scenario. 53

Chapter 1

Introduction

1.1 Motivation

In today's commercial, scientific, and military communities there exists an ever growing interest for the development of Unmanned Surface Vehicles (USVs), also known as Autonomous Surface Vehicles (ASVs). These unmanned vehicles are defined through their capability of performing tasks and missions, in a variety of cluttered environments without any human intervention. The possible applications for these vehicles are many and include scientific research, environmental missions, transportation and ocean resource exploration. The further development of USVs are expected to provide several benefits, as compared to other manned vehicles, such as lower development and operational costs, improved personal safety, and extended operational range. USVs are also expect to be able to perform more hazardous missions and exhibit enhanced maneuverability in sophisticated environments. Despite these benefits, most USVs today does only exhibit semi-autonomy, owing to numerous challenges faced in the development of full-autonomy. The further development of fully-autonomous USVs depends on the development of effective, accurate, and reliable USV systems. This includes the development of advanced guidance, navigation and control (GNC) systems. (Liu et al., 2016)

The GNC system can be divided into three subsystems, which in accordance with Fossen (2011) can be described as follows. The *guidance system* is used to continuously compute the reference position, velocity and acceleration of the marine craft, to be used in the motion control system. The data computed by the guidance system will further be provided to a human operator. The *navigation system* is used to direct the craft by determining its position, course and distance traveled. In some cases the velocity and the acceleration of the craft is also determined. This is normally achieved by the use of a global navigation satellite system in combination with motion sensors, such as accelerometers and gyros. The *control system* is used to determine the necessary control forces and moments to be provided to the craft, in order to satisfy certain control objectives. These control objectives are often seen in conjunction with the guidance system.

A fundamental part of the USV guidance system, is the path planning system. This system can be viewed as module of the GNC system, as described in Lekkas (2014). The path planning system is used to design and generate a path, which if followed without deviations, will be both safe and feasible. The path that is generated must satisfy several desirable properties, which takes both physical constraints and workspace constraints into account. The system must therefore ensure that the path which is generated takes the dynamics of the vehicle into account, and that the path stays at a safe distance from obstacles at all times. Temporal assignments can also be included in the path planning system, in such cases the system will specify where on the path the craft should be at any time instant. In the world of robotics the term motion planning is used to describe such a system. Motion planning does however differ in the way that it often involves both the design of a suitable path, and the actions that should be taken in order to accomplish the mission.

The path planning system is of great significance when it comes to performance and mission accomplishment. The path that this system creates and the properties associated with this path, will have a direct influence on the guidance system, especially when it comes to underactuated vehicles. The path planning system and the guidance system is related in the way that they specify what one wants to achieve, and how one should act in order to achieve it. Figure 1.1 shows how the different modules of GNC system interacts. Due to the interconnection between these modules, it can often be difficult to determine the stability of the total system. In stability studies it is therefore often convenient to view the entire system as a cascade system, where the output of one module is the input to another.

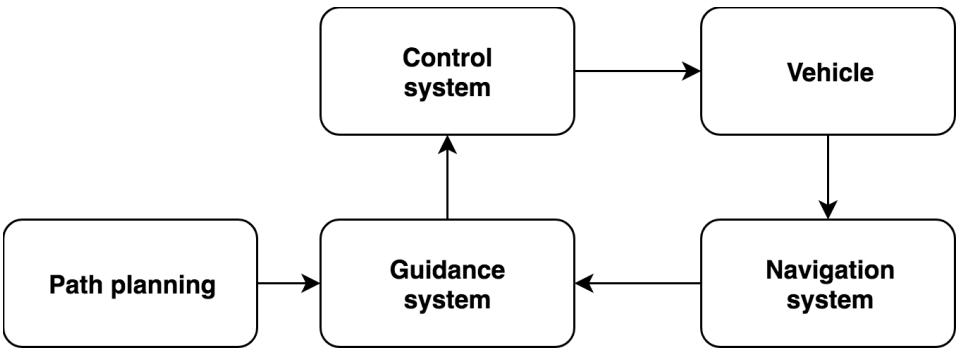

Figure 1.1: Interaction between the modules of the GNC system.

In later years a tremendous amount of effort has been dedicated to making USVs more autonomous, however there does still exist some significant challenges in this development. Some of these challenges are related to the modules of the GNC system. In relation to path planning there is a need for more computational effective global planners, global converging local planners, and more effective, accurate and reliable methodologies for obstacle avoidance.

1.2 Scope of work

This thesis aims to contribute to the development of a new generation of path planning algorithms, that incorporates in its formulation the dynamics of the vehicle, and information about obstacles and other vehicles in its vicinity. In this algorithm, the Bézier curve will be exploited in the generation of a rich set of paths. The differential flatness property of the vehicle will further be used to assign a cost to each path, reflecting the dynamic capabilities of the vehicle on that path.

To this end, the thesis begins with a presentation of relevant background information and theory. This includes the topics of parametric curves, the Bézier curve, system modeling and differential flatness. The topic of differential flatness is given special attention, and is viewed in conjunction with underactuated surface vessels. The thesis further proposes a path planning algorithm that is formulated within the framework of optimization. This algorithm is developed with basis in the theory presented in the thesis. The proposed path planning algorithm is capable of generating C^2 and G^2 continous paths, taking static obstacle constraints into consideration. The algorithm also enables the generation of paths between two points, and further constrains the curvature of the path. The thesis further describes the implementation of this algorithm, before it finish off by reviewing the proposed algorithm through a series of numerical simulations.

1.3 Limitations

The main limitation of this thesis has been the availability of a good optimization solver, that were suited for the optimization problem that has been formulated. The utilized solver exhibits a sensitivity to the initial guess provided. This has served as an issue in the retrieval of results, and it has also disallowed studies on for instance computational time.

The dynamics of the vessel is included in the path planning through the use of the differential flatness property. In order to be able to utilize this property, the mathematical model of the surface vessel had to be simplified. This implies that the developed path planning algorithm does not incorporate all the dynamics that was initial intended.

1.4 List of publications

The following conference paper has been submitted, reviewed and accepted for publication, and is the outcome of the work related to this thesis:

Vahid Hassani and Simen V. Lande. Path Planning for Marine Vehicles using Bézier Curves. In: *11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*, Opatija, Croatia, 2018.

The paper is attached in the appendix.

1.5 Thesis outline

This thesis is organized in the following way:

Chapter 2: This chapter gives an overview of some mathematical notations and definitions that are used throughout this thesis, as well as some commonly used abbreviations.

Chapter 3: This chapter introduces the concept of path parameterization, continuity, arc length and curvature. It further focuses on the Bézier curve, including a historical remark, formulations and other useful concepts.

Chapter 4: Explores the fundamentals of vehicle modeling, including kinematics and kinetics. It also provides a mathematical model for an underactuated surface vessel. The concept of differential flatness is also introduced, with an elaboration on how this property can be used in path planning.

Chapter 5: The main focus of this chapter is the development of a path generation algorithm with basis in the theory presented in the preceding chapters. An extensive analysis on the Bézier curve is presented, along with the formulation of an optimization problem that is used in order to generate paths.

Chapter 6: This chapter presents a series of numerical simulation, that is used to show the efficacy of the proposed path generation algorithm.

Chapter 7: Presents a discussion on the overall results obtained in this thesis, and some implementational challenges encountered.

Chapter 8: This chapter presents the conclusion of this thesis, and the most relevant ideas for further development on the results presented.

Chapter 2

Preliminaries

2.1 Mathematical notations and definitions

In this thesis all vectors and matrices are written in boldface, while scalars are not. Time derivatives of $\boldsymbol{x}(t)$ are given as $\dot{\boldsymbol{x}}, \ddot{\boldsymbol{x}}, \boldsymbol{x}^{(3)}, \ldots, \boldsymbol{x}^{(i)}$. Derivatives with respect to other variables are given as $\boldsymbol{x}', \boldsymbol{x}'', \boldsymbol{x}^{3'}, \ldots, \boldsymbol{x}^{i'}$. The index set $\mathcal{I}^n \subset \mathbb{N}_{>0}$ are all natural numbers from 1 to n given as: $\mathcal{I}^n = \{1, 2, ..., n\}$. The norm $|\boldsymbol{x}|$ is the Euclidean norm $\sqrt{\boldsymbol{x}^T\boldsymbol{x}}$, and $|x|$ is the absolute value of $x \in \mathbb{R}$.

2.2 Abbreviations

ASV	Autonomous Surface Vehicle
CAD	Computer Aided Design
CG	Center of Gravity
DOF	Degree Of Freedom
ECEF	Earth-Centered, Earth-Fixed
ECI	Earth-Centered Inertial
GNC	Guidance, Navigation and Control
NED	North East Down
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicle
USV	Unmanned Surface Vehicle

Chapter 3

Path parameterization

3.1 General path parameterization

3.1.1 Parametric curves

In mathematics a parametric curve in the plane is a curve described by a set of parametric equations that defines the coordinates of each point on the curve, as functions of a scalar parameter $\varpi \in \mathbb{R}$. Considering a planar path, where the position of a point belonging to the path is represented by $\mathbf{p}_p(\varpi) = [x_p(\varpi),\, y_p(\varpi)]^T \in \mathbb{R}^2$, the path will be a one-dimensional manifold that can expressed by the set (Breivik and Fossen, 2009)

$$\mathcal{P} := \left\{ \mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} = \mathbf{p}_p(\varpi)\ \forall\ \varpi \in \mathbb{R} \right\}. \tag{3.1}$$

The parameter ϖ is typically restricted to some interval $\varpi \in [\varpi_0,\, \varpi_1]$. The terms path and curve will be used interchangeably in this thesis.

3.1.2 Piecewise parametric curve

The representation of complex shapes are most often achieved by piecing together smaller curve segments with a limited validity interval. In comparison with a representation of the same shape by a single higher order curve, one achieves lower function complexity and one avoids ill-conditioned parameters. The piecewise parametric curve does however demand consideration at the transition points between the subpaths. For planar paths consisting of a number of curve segments, each single curve segment can be expressed by the set (Lekkas, 2014)

$$\mathcal{P}_i := \left\{ \mathbf{p}_i \in \mathbb{R}^2 \mid \mathbf{p}_i = \mathbf{p}_{i,p}(\varpi)\ \forall\ \varpi \in \mathcal{I}_i = [\varpi_{i,0},\, \varpi_{i,1}] \subset \mathbb{R} \right\}, \tag{3.2}$$

and the path can be written as a superset of the m curve segments

$$\mathcal{P}_s = \bigcup_{i=1}^{m} \mathcal{P}_i. \tag{3.3}$$

3.1.3 Continuity

In the context of planar curves the concept of continuity is of importance, and it is used to describe smoothness. For a parametric curve, one distinguish between *parametric* and *geometric* continuity. This section will present definitions for both types of continuity, and elaborate on the differences.

Parametric continuity

Parametric continuity is related to the derivatives of the parametric curve, and is given by the notation C^n, for which n denotes the degree of continuity. In this thesis the following definition for parametric continuity will be used (Barsky and DeRose, 1984)

Definition 3.1. C^n and regularity

- A scalar function $f(\varpi)$ belongs to the class C^n on the interval $[\varpi_0, \varpi_1]$ if it is n-times continuously differentiable on $[\varpi_0, \varpi_1]$. It is regular on $[\varpi_0, \varpi_1]$ if

$$\frac{df}{d\varpi} \neq 0 \quad \forall \varpi \in [\varpi_0, \varpi_1]. \tag{3.4}$$

- A parameterization $\mathbf{p}_p(\varpi) = [x_p(\varpi), y_p(\varpi)]^T$, $\varpi \in [\varpi_0, \varpi_1]$ is C^n if each of the coordinate functions $x_p(\varpi)$ and $y_p(\varpi)$ is C^n on $[\varpi_0, \varpi_1]$. It is regular if

$$\frac{d\mathbf{p}_p}{d\varpi} \neq \mathbf{0} \quad \forall \varpi \in [\varpi_0, \varpi_1]. \tag{3.5}$$

- A curve is regular if it can be generated by a regular parameterization.

In the case where a curve is constructed by joining several smaller curve segments together, the continuity at the joint must be considered. The following definition can be used to established the degree of parametric continuity in such cases (Barsky and DeRose, 1984)

Definition 3.2. Let $\mathbf{p}_p(\varpi)$ and $\mathbf{q}_p(\varpi)$ be regular C^n parametrizations such that $\mathbf{p}_p(\varpi_1) = \mathbf{q}_p(\varpi_0) = \mathbf{J}$. That is, the "right" endpoint of $\mathbf{p}_p$ agrees with the "left" endpoint of $\mathbf{q}_p$. They meet with n-th order parametric continuity (C^n) at $\mathbf{J}$ if

$$\left.\frac{d^k\mathbf{p}_p}{d\varpi^k}\right|_{\varpi_1} = \left.\frac{d^k\mathbf{q}_p}{d\varpi^k}\right|_{\varpi_0} \qquad k = 1, ..., n. \tag{3.6}$$

Geometric continuity

Geometric continuity is a relaxed form of parametric continuity, that place less emphasis on the particulars of the parameterization. This type of continuity is of importance, as parametric continuity often disallows many parameterizations that are visually smooth. Meaning that parametric continuity does not necessarily reflect the smoothness of the curve, but rather the smoothness of the parameterization.

Definition 3.3. Geometric continuity (Barsky and DeRose, 1984, 1989)
Let $\mathbf{p}_p(\varpi)$ and $\mathbf{q}_p(\varpi)$ be regular C^n parametrizations such that $\mathbf{p}_p(\varpi_1) = \mathbf{q}_p(\varpi_0) = \mathbf{J}$, where $\mathbf{J}$ is a simple point. They meet with n-th order geometric continuity, denoted G^n, at $\mathbf{J}$ if there exist a parameterization $\tilde{\mathbf{q}}_p(\varpi)$ to $\mathbf{q}_p(\varpi)$ such that $\mathbf{p}_p(\varpi)$ and $\tilde{\mathbf{q}}_p(\varpi)$ meet with C^n continuity at the point $\mathbf{J}$.

For a piecewise parametric curve, the following interpretation of geometric continuity up to the second degree can be adopted:

G^0: The curve is connected.

G^1: The unit tangent vector is continuous.

G^2: The curvature is continuous.

Example 3.1. Let

$$\mathbf{p}_p(\varpi) = \begin{bmatrix} \varpi \\ \varpi \end{bmatrix}, \quad \varpi \in [0, 1], \tag{3.7}$$

$$\mathbf{q}_p(\varpi) = \begin{bmatrix} \sin(\varpi - \frac{\pi}{4}) + \sin(\frac{\pi}{4}) + 1 \\ \cos(\varpi - \frac{\pi}{4}) - \cos(\frac{\pi}{4}) + 1 \end{bmatrix}, \quad \varpi \in [0, \pi], \tag{3.8}$$

be two parametrizations joined together in order to create one curve, see Figure 3.1.

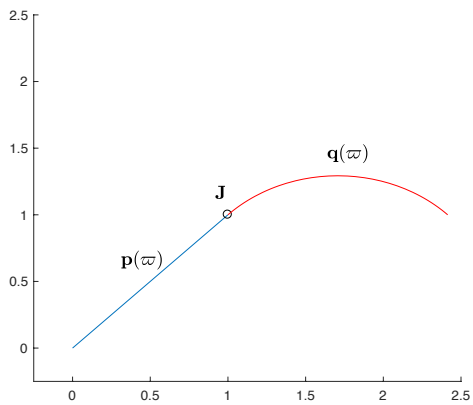

Figure 3.1: Two parametric curves with G^1 continuity.

It is obvious from Figure 3.1 that the curve is C^0 continuous, since they share a common endpoint $\mathbf{J}$. To verify if the curve is C^1 continuous we can use Definition 3.2, which demands that the first derivatives are equal at the joint endpoint. The derivatives are

$$\left.\frac{d\mathbf{p}_p}{d\varpi}\right|_{\varpi_1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \left.\frac{d\mathbf{q}_p}{d\varpi}\right|_{\varpi_0} = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}, \tag{3.9}$$

revealing that the curve is not C^1 continuous. However, if we normalize the vectors given by Eq. (3.9) we see that the unit tangent vectors are equal, which implies that the curve is G^1 continuous.

3.1.4 Arc length

The arc length of a parametric curve is described as the distance a particle has to travel along the curve from one end to the other. This positive scalar value is derived from Pythagoras' rule, and is given as

$$s = \int_{\varpi_0}^{\varpi_1} \left|\mathbf{p}_p'(\tau)\right| \, d\tau = \int_{\varpi_0}^{\varpi_1} \sqrt{x_p'(\tau)^2 + y_p'(\tau)^2} \, d\tau, \tag{3.10}$$

where τ is the integration variable. The arc length between two points on the curve can be found by changing the bounds of the integral.

Approximated arc length

The expression for the arc length given by Eq. (3.10) does not necessarily have a closed form, and as a consequence it may be necessary to find the arc length through some other means. In such cases, one could turn to numerical methods or approximations in order to estimate the arc length. The accuracy of the estimation is dependent upon the method used. In this thesis the arc length will be approximated by flattening the curve and calculating the linear distance from point to point, as:

$$s_n = \sum_{i=1}^{n} \left|\mathbf{p}_p(\varpi_i) - \mathbf{p}_p(\varpi_{i-1})\right|, \tag{3.11}$$

where n is the segment count and i is used to denote specific values of $\varpi \in [\varpi_0, \varpi_1]$. This approach does come with an error, but this error could be made arbitrarily small by increasing the segment count.

3.1.5 Path curvature

Curvature is a measure for the rate at which a curve is turning, away from its tangent line, at any point along the curve. In the case where the curve is parameterized with respect to arc length $\mathbf{p}_p(s)$, the curvature will be defined as (Adams and Essex, 2013)

$$\kappa(s) = \left|\frac{d\mathbf{T}}{ds}\right|, \tag{3.12}$$

where $\mathbf{T} = d\mathbf{p}_p/ds$ is the tangent. Further, the *radius of curvature* is the reciprocal of the curvature, given as

$$R(s) = \frac{1}{\kappa(s)}. \tag{3.13}$$

The radius of curvature is the radius of the circle through $\mathbf{p}_p(s)$ that most closely approximates the curve near that point. This circle is know as the osculating circle, see Figure 3.2.

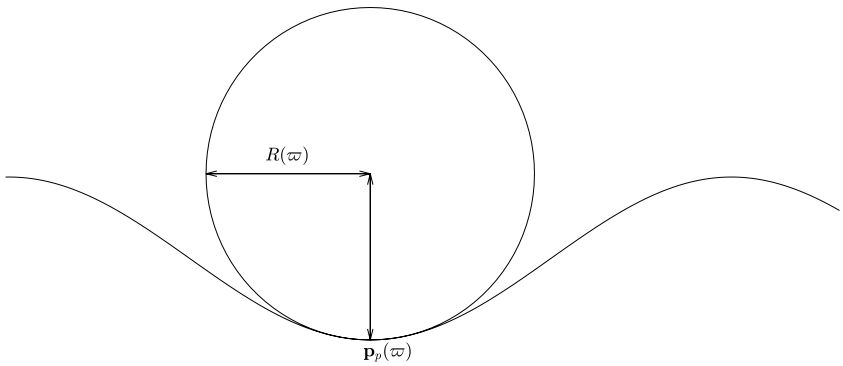

Figure 3.2: Osculating circle at $\mathbf{p}_p(\varpi)$.

The formulas given above are not so useful when the curve is not parameterized in terms of the arc length. If the curve is given by a general parameterization, the curvature can be found as

$$\kappa(\varpi) = \frac{\left|\mathbf{p}_p'(\varpi) \times \mathbf{p}_p''(\varpi)\right|}{\left|\mathbf{p}_p'(\varpi)\right|^3} = \frac{\left|x_p'(\varpi)y_p''(\varpi) - x_p''(\varpi)y_p'(\varpi)\right|}{(x_p'(\varpi)^2 + y_p'(\varpi)^2)^{\frac{3}{2}}}. \tag{3.14}$$

Signed curvature

In some cases it is convenient to know the direction for which the curve is turning, in such cases one can use the signed curvature, given as

$$\varkappa(\varpi) = \frac{\mathbf{p}_p'(\varpi) \times \mathbf{p}_p''(\varpi)}{\left|\mathbf{p}_p'(\varpi)\right|^3} = \frac{x'(\varpi)y''(\varpi) - x''(\varpi)y'(\varpi)}{(x'(\varpi)^2 + y'(\varpi)^2)^{\frac{3}{2}}}. \tag{3.15}$$

The sign of the curvature will indicate the direction for which the unit tangent vector rotates, as a function of the parameter ϖ along the curve. For $\varkappa > 0$ the unit tangent will rotate counterclockwise, and for $\varkappa < 0$ rotate clockwise.

3.2 Bézier curves

The Bézier curve is a polynomial parametric curve, which can be utilized in order to create smooth paths and shapes. They have a wide range of different applications, and are established as a common tool in computer aided design (CAD), computer graphics, animation, path planning and other related fields. A piecewise parametric curve consisting of Bézier curves, are refereed to as the Bézier spline. This section will present a short historical remark on the Bézier curve, as well as a mathematical formulations, in addition to some important function operations. The mathematical formulations in this section are based off of Bartels et al. (1986) and Sederberg (2016).

3.2.1 History and background

The mathematical basis for the Bézier curve are the Bernstein polynomials, named after the Russian mathematician Sergei Natanovich Bernstein. These polynomials were first introduced and published in 1912, as a means to constructively prove the Weierstrass theorem. In other words, as the ability of polynomials to approximate any continuous function, to any desired accuracy over a given interval. The slow convergence rate and the technological challenges in the construction of the polynomials at the time of publication, led to the Bernstein polynomial basis being seldom used for several decades to come.

The Bernstein polynomials resurfaced during the 1960s in which coincidentally two french automobile engineers of different companies, started to search for ways of representing complex shapes using digital computers. The motivation for finding a new way to represent free-form shapes, was due to the expensive process of sculpting such shapes at the time, which was done using clay.

The first engineer concerned with this matter was Paul de Faget de Casteljau working for Citroën, who did his research in 1959. His findings lead to what is known as de Casteljau's algorithm, a numerically stable method to evaluate Bézier curves. De Casteljau's work were only recorded in Citroën's internal documents, and remained unknown to the rest of the world for a long time. His findings are however today, a great tool for handling Bézier curves. The person who lends his name to the Bézier curves, and is principally responsible for making the curves so well known, is the engineer Pierre Ètienne Bézier. Bézier worked at Renault, and published his ideas extensively during the 1960s and 1970s. Both Bézier's and de Casteljau's original formulations did not explicitly invoke the Bernstein basis, however the key features are unmistakably linked to it, and today the Bernstein basis is a key part in the formulation. This section has been inspired by the work presented in Farouki (2012), the interested reader is therefore refereed to this article for further reading on the history of Bernstein polynomials and Bézier curves.

3.2.2 General definition

A Bézier curve is defined by a set of control points $\mathbf{p}_0$ to $\mathbf{p}_n$ for which n denotes the degree of the curve. The number of control points for a curve of degree n is $n+1$, and the first and last control points will always be the end points of the curve. The intermediate

points does not necessarily lay on the curve itself. The Bézier curve can be express on a general form as

$$\mathbf{p}_p(\varpi) = \sum_{i=0}^{n} B_i^n(\varpi)\mathbf{p}_i, \quad \varpi \in [0,1], \tag{3.16}$$

where ϖ defines a normalized time variable and $B_i^n(\varpi)$ denotes the blending functions of the Bézier curve, which are Bernstein polynomials defined as

$$B_i^n = \binom{n}{i}(1-\varpi)^{n-i}\varpi^i, \quad i \in \{0,1,\ldots,n\}. \tag{3.17}$$

The binomial coefficient is given as

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!}. \tag{3.18}$$

The polygon formed when connecting the control points by straight lines, from $\mathbf{p}_0$ through $\mathbf{p}_n$, creates what is know as the control polygon. The convex hull of the control polygon contains the Bézier curve.

Matrix representation

In some applications it might be more sensible to express the Bézier curve in terms of matrix operations. This can be achieved by expressing the parameters, the coefficients and the control points separately (Joy, 2000):

$$\mathbf{p}_p(\varpi) = \begin{bmatrix} 1 & \varpi & \ldots & \varpi^n \end{bmatrix} \begin{bmatrix} b_{0,0} & 0 & \ldots & 0 \\ b_{1,0} & b_{1,1} & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ b_{n,0} & b_{n,1} & \ldots & b_{n,n} \end{bmatrix} \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_n \end{bmatrix}, \tag{3.19}$$

where the coefficients can be found as

$$b_{i,j} = (-1)^{i-j}\binom{n}{i}\binom{i}{j}. \tag{3.20}$$

The advantage of using the matrix representation is that it simplifies the computation of any derivatives, as only the parameter vector needs derivation.

Example 3.2. Lets consider a fourth order Bézier curve, with the Bernstein polynomials given as

$$\begin{aligned} B_0^4(\varpi) &= (1-\varpi)^4 & B_1^4(\varpi) &= 4(1-\varpi)^3\varpi \\ B_2^4(\varpi) &= 6(1-\varpi)^2\varpi^2 & B_3^4(\varpi) &= 4(1-\varpi)\varpi^3 \\ B_4^4(\varpi) &= \varpi^4. \end{aligned}$$

These functions are shown in Figure 3.3.

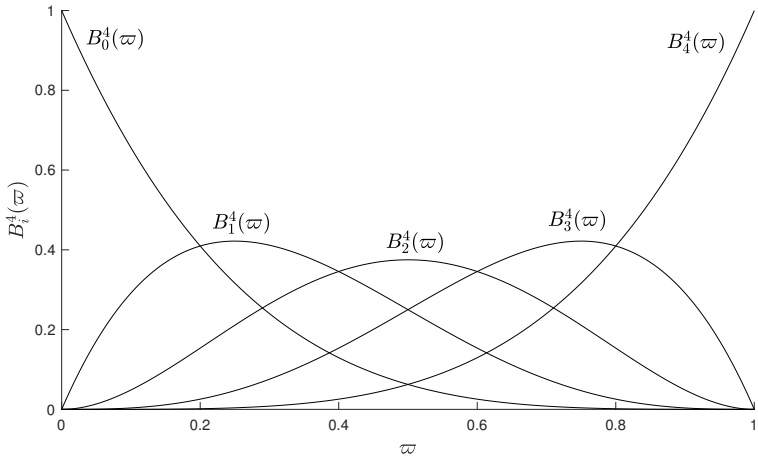

Figure 3.3: The Bernstein polynomials.

Further, lets define the following control points

$$\mathbf{p}_0 = [1, 1]^T, \quad \mathbf{p}_1 = [2, 4]^T, \quad \mathbf{p}_2 = [7, 5]^T,$$
$$\mathbf{p}_3 = [10, 3]^T, \quad \mathbf{p}_4 = [10, 1]^T.$$

The Bézier curve created by these control points, are shown in Figure 3.4.

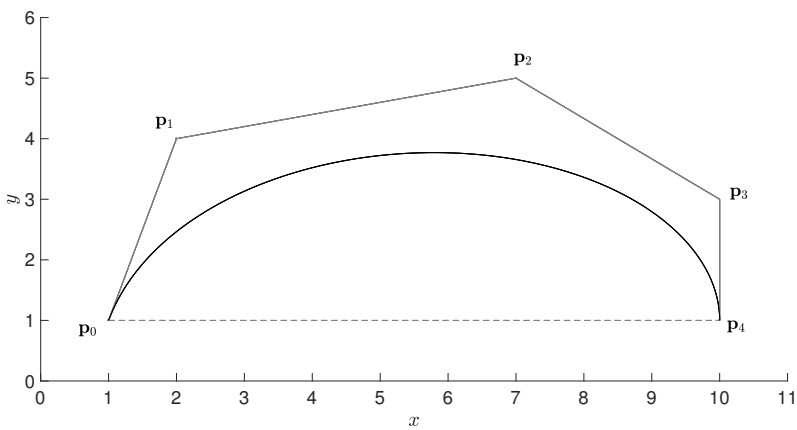

Figure 3.4: A fourth order Bézier curve with the control polygon.

In the figure above, we can see that the Bézier curve is contained within the convex hull (control polygon and dotted line).

3.2.3 Extension to arbitrary interval

In the general definition of the Bézier curve, the parameter of the curve is defined on the unit domain. While this interval is the most commonly used, it might in some cases be more useful to define the curve parameter on another interval. This could be especially useful when fitting several Bézier curves together. An arbitrary interval may be defined as

$$\varpi \in [\varpi_0, \varpi_1]. \tag{3.21}$$

This corresponds to the end points $\mathbf{p}_p(\varpi_0) = \mathbf{p}_0$ and $\mathbf{p}_p(\varpi_1) = \mathbf{p}_n$. By modification of Eq. (3.16) one then obtains the following expression

$$\mathbf{p}_p(\varpi) = \sum_{i=0}^{n} \binom{n}{i} \left(\frac{\varpi_1 - \varpi}{\varpi_1 - \varpi_0}\right)^{n-i} \left(\frac{\varpi - \varpi_0}{\varpi_1 - \varpi_0}\right)^{i} \mathbf{p}_i, \quad \varpi \in [\varpi_0,\ \varpi_1]. \tag{3.22}$$

3.2.4 Subdivision of Bézier curves

A useful algorithm when dealing with the Bézier curve is the subdivision algorithm, also know as de Casteljau algorithm. This algorithm can be used to subdivide a Bézier curve defined on the unit domain, into two new Bézier curves, the first of them defined on $[0, \tau]$ and the second on $[\tau, 1]$. This is accomplished by creating a set of new control points

$$\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau \mathbf{p}_{i+1}^{j-1}, \quad j \in \{1, \ldots, n\}, \quad i \in \{0, \ldots, n-j\}. \tag{3.23}$$

By the use of these new control points the two new Bézier curves can be represented by the following equations

$$\mathbf{p}_{p,1}(\varpi) = \sum_{i=0}^{n} B_i^n(\frac{\varpi}{\tau})\mathbf{p}_i^j, \quad \varpi \in [0, \tau], \tag{3.24}$$

$$\mathbf{p}_{p,2}(\varpi) = \sum_{i=0}^{n} B_i^n(\frac{\varpi - \tau}{1-\tau})\mathbf{p}_i^{n-j}, \quad \varpi \in [\tau, 1]. \tag{3.25}$$

The subdivision algorithm can be used for curves of any degree, and is extremely valuable when computing the coordinates and tangent vector of any point along the curve. A Bézier curve that is repeatedly subdivided will create a collection of control polygons that converge towards the curve. The algorithm therefore yields a simple way of plotting the Bézier curve.

3.2.5 Degree elevation

An interesting property of the Bézier curve is that any curve of degree n, can be exactly represented by a curve of degree $n+1$ or higher. The degree elevation of the Bézier curve is performed by giving the higher order curve specific control points. By denoting these new control points as $\mathbf{p}_i^*$, they can be computed as

$$\mathbf{p}_i^* = \alpha_i \mathbf{p}_{i-1} + (1-\alpha_i)\mathbf{p}_i, \quad \alpha_i = \frac{i}{n+1}. \tag{3.26}$$

The degree elevation works for any Bézier curve and it can be applied repeatedly to raise the degree to any desired level. It is a process that is especially useful when it is necessary to express a curve of a specific degree.

3.2.6 Derivatives

The derivative of any Bézier curve of degree n, is a Bézier curve of degree $n-1$. As the control points are constant and independent of the curve parameter ϖ, the derivative is found by computing the derivative of the Bernstein polynomials. With basis in Eq. (3.17), the derivatives of the Bernstein polynomials can be given as

$$B_i^{n\prime}(\varpi) = n\left(B_{i-1}^{n-1}(\varpi) - B_i^{n-1}(\varpi)\right). \tag{3.27}$$

The derivative of the Bézier curve then takes the following form

$$\mathbf{p}'(\varpi) = n\sum_{i=0}^{n-1} B_i^{n-1}(\varpi)(\mathbf{p}_{i+1} - \mathbf{p}_i), \quad \varpi \in [0,1]. \tag{3.28}$$

To further simplify this expression one can define the control points of the first derivative as, $\mathbf{q}_i = \mathbf{p}_{i+1} - \mathbf{p}_i$, resulting in the following expression

$$\mathbf{p}'(\varpi) = n\sum_{i=0}^{n-1} B_i^{n-1}(\varpi)\mathbf{q}_i, \quad \varpi \in [0,1]. \tag{3.29}$$

The first derivative curve is referred to as the hodograph of the original Bézier curve. To determine higher order derivatives, one could easily use Eq. (3.29), and then apply the same approach as previously presented in this section. However, it is possible to derive a general expression for the k-th derivative of any Bézier curve. As the control points of the derivative are defined by the difference between the control points of the previous step, we can derive a general expression for this as

$$\mathbf{q}_i^k = \mathbf{q}_{i+1}^{k-1} - \mathbf{q}_i^{k-1}, \quad i \in \{0,1,\ldots,n-k\}, \tag{3.30}$$

where n denotes the degree of the original curve, and k denotes the derivative. Further, $\mathbf{q}_i^k$ gives the control point for the k-th derivative and $\mathbf{q}_i^{k-1}$ are the control points of the $(k-1)$-th derivative. Note that to find the control points of the k-th derivative, it is necessary to find the control points of all the derivative before k. After the control points are determined one can express the k-th derivative of the Bézier curve as

$$\mathbf{p}^{(k')}(\varpi) = n(n-1)(n-2)...(n-k+1)\sum_{i=0}^{n-k} B_i^{n-k}(\varpi)\mathbf{q}_i^k, \quad \varpi \in [0,1]. \tag{3.31}$$

Example 3.3. Consider a fifth order Bézier curve, then the first and second derivatives evaluated at the endpoints will be given as

$$\mathbf{p}'(0) = 5(\mathbf{p}_1 - \mathbf{p}_0) \qquad \mathbf{p}'(1) = 5(\mathbf{p}_5 - \mathbf{p}_4) \tag{3.32}$$

$$\mathbf{p}''(0) = 20(\mathbf{p}_2 - 2\mathbf{p}_1 + \mathbf{p}_0) \qquad \mathbf{p}''(1) = 20(\mathbf{p}_5 - 2\mathbf{p}_4 + \mathbf{p}_3). \tag{3.33}$$

Chapter 4

System modeling and properties

4.1 Vehicle modelling

This section will cover some fundamentals in vehicle modeling, including kinematics and kinetics. A horizontal plane model for a surface vessel will also be presented.

4.1.1 Kinematics

Kinematics treats only the geometrical aspects of motion and is used to describe the position, velocity and acceleration of the body. Kinetics are used to describe the forces causing these motions.

Reference frames

In kinematics one distinguish between global and local reference frames. The global reference frames have their origin in the center of Earth, and they are used for global navigation. Examples of such frames are the Earth-centered inertial- (ECI) and the Earth-centered Earth-fixed (ECEF) reference frames. The local reference frames have their origin placed on a geographically stationary point or on the vehicle itself. Only local reference frames will be used in this thesis, and a short summary of these frames are given below (Fossen, 2011)

NED: The *North-East-Down* coordinate system $\{n\} = (x_n, y_n, z_n)$ with origin o_n is defined relative to the Earth's reference ellipsoid. For this system the x axis points towards true North, the y axis points towards East, and the z axis points downwards normal to the Earth's surface. The location of the frame's origin is determined by using the angles for longitude and latitude.

BODY: The *body-fixed* reference frame $\{b\} = (x_b, y_b, z_b)$ with origin o_b is a moving coordinate frame that is fixed to the craft. The position and orientation of the

craft are described relative to an inertial reference frame, while the linear- and angular velocities are expressed in the body-fixed coordinate system.

For marine crafts operating in a local area, the longitude and latitude can be approximated as constant. This is usually referred to as flat Earth navigation, and it allows one to assume that $\{n\}$ is an internal frame.

Position and velocity variables

In order to describe the motion of the marine craft, the same notion as in SNAME (1950) will be adopted in this thesis. The linear and angular velocities will be given in the BODY frame, while the positions and Euler angles will be given in the NED frame. The notation for all six degrees of freedom (DOF) are summarized in the table below

Table 4.1: Notation for motion variables.

DOF		Linear and angular velocities	Positions and Euler angles
1	motions in the x direction (surge)	u	x
2	motions in the y direction (sway)	v	y
3	motions in the z direction (heave)	w	z
4	rotation about the x axis (roll)	p	ϕ
5	rotation about the y axis (pitch)	q	θ
6	rotation about the z axis (yaw)	r	ψ

The quantities expressed in Table 4.1, will further be expressed in a vectorial setting as

$$\boldsymbol{\eta} = [x, y, z, \phi, \theta, \psi]^T \,, \quad \boldsymbol{\nu} = [u, v, w, p, q, r]^T \,, \tag{4.1}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\nu}$ are known as the generalized position- and generalized velocity vectors, respectively. Note that the positions in the $\{n\}$ frame often are denoted by $[N, E, D]$, this will however not be done in this thesis.

Rotation matrix

The horizontal motion of a ship is described by the motion components in surge, sway and yaw, reducing the generalized position and velocity vectors to $\boldsymbol{\eta} = [x, y, \psi]^T$ and $\boldsymbol{\nu} = [u, v, r]^T$ respectively. This implies that all dynamics associated with heave, pitch and roll are neglected. The kinematic equations of motion is in this case reduced to one principal rotation about the z axis, and can be formulated in a vectorial setting as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu}, \tag{4.2}$$

where the rotational matrix $\boldsymbol{R}(\psi)$ describes the coordinate transformation from $\{b\}$ to $\{n\}$ and is given as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{4.3}$$

Course, heading and sideslip

For marine crafts performing maneuvers in the horizontal plane the relationship between the angular quantities course, heading and sideslip, is of importance. The relationship between these variables are shown in Figure 4.1.

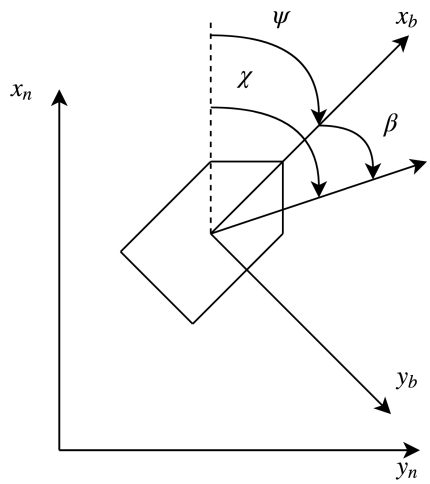

Figure 4.1: Relationship between course χ, heading ψ and sideslip β.

In this figure, the speed of the marine craft is denoted by U and is given as

$$U = \sqrt{u^2 + v^2}. \tag{4.4}$$

Further, the course denoted by χ, is the angle between the x_n axis of $\{n\}$ and the velocity vector of the craft. The sideslip denoted by β, is found as

$$\beta = \sin^{-1}\left(\frac{v}{U}\right), \tag{4.5}$$

and the relationship between the three angular variables is expressed as

$$\chi = \psi + \beta. \tag{4.6}$$

Ocean currents

In the development of mathematical models for ships one often accounts for ocean currents. The current may be assumed as irrational and constant, in which case the relative velocity of the ship $\boldsymbol{\nu}_r$, apposed to the velocity of the current $\boldsymbol{\nu}_c$ is given as

$$\boldsymbol{\nu}_r = \boldsymbol{\nu} - \boldsymbol{\nu}_c, \tag{4.7}$$

where the ocean current velocity are described through components in surge, sway and heave, for six degrees of freedom.

4.1.2 Vehicle dynamics

The equations of motion for a marine craft is derived by carefully studying the motions of rigid bodies, hydrodynamics and hydrostatics. In Part 1 of Fossen (2011) an extensive study on these topics are presented, and the equations are formalized in a vectorial setting. The 6 DOF marine craft equations of motion are here given as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu} \tag{4.8}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \boldsymbol{\tau}_{wave}, \tag{4.9}$$

where $\boldsymbol{M}$ is the mass and inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $\boldsymbol{D}(\boldsymbol{\nu})$ is the damping matrix, $\boldsymbol{g}(\boldsymbol{\eta})$ is the gravitational and buoyancy forces, and $\boldsymbol{J}(\boldsymbol{\eta})$ is a 6 DOF kinematic transformation matrix. Further the control forces and moments are denoted $\boldsymbol{\tau}$, while wind and wave forces are denoted by $\boldsymbol{\tau}_{wind}$ and $\boldsymbol{\tau}_{wave}$, respectively.

4.1.3 Vehicle models

The complexity of the vehicle model, and the number of differential equations needed for a certain representation will vary depending on the purpose of the model. In general one distinguish between three types of models; simulation model, control design model and observer design model. (Fossen, 2011) These models are of different fidelity, which indicates how accurately they describe the real system, whereas the simulation model is of the highest fidelity. For any marine craft model the rigid body kinetics are derived by applying Newtonian mechanics, while external forces and moments acting on the marine craft are usually modeled by using either:

Maneuvering theory: The study of ships moving in calm water, with a positive constant speed. It is assumed that the the hydrodynamic coefficients are frequency independent, which implies no wave excitation. The maneuvering model might be both linear and nonlinear, depending on the application. Maneuvering is associated with course keeping, course changes, turning, and stopping.

Seakeeping theory: The study of ships moving in a seaway at zero or constant speed. The hydrodynamic coefficients are computed as functions of the wave excitation frequency using the hull geometry, and mass distribution. Seakeeping models are usually derived within a linear framework.

Horizontal plane model

The horizontal plane model is described by the motions in surge, sway and yaw, and is thereby a 3 DOF model. Through the use of maneuvering theory and by assuming zero current, the equations of motion given by Eq. (4.8) and Eq. (4.9) can be reduced to

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{4.10}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}. \tag{4.11}$$

The system matrices are often derived under the assumption of homogeneous mass distribution and xz-plane symmetry, which allows for a decoupling of surge, from sway and yaw. The system inertia matrix $\boldsymbol{M}$ includes both rigid body and added mass terms, and has the following properties $\boldsymbol{M} = \boldsymbol{M}^T > 0$ and $\dot{\boldsymbol{M}} = 0$. The inertia matrix can be expressed as

$$\boldsymbol{M} = \boldsymbol{M}_{RB} + \boldsymbol{M}_A = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix}, \tag{4.12}$$

where m is the mass of the ship, I_z is the moment of inertia in yaw and x_g is the distance between the center of gravity (CG) and the origin of $\{b\}$. The remaining terms describe hydrodynamic coefficients for added mass, using SNAME (1950) notation. The Coriolis and Centripetal matrix can be parameterized as a skew symmetric matrix, that is $\boldsymbol{C}(\boldsymbol{\nu}) = -\boldsymbol{C}^T(\boldsymbol{\nu})$. The Coriolis and Centripetal will also include both rigid body and added mass terms, given as

$$\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C}_{RB}(\boldsymbol{\nu}) + \boldsymbol{C}_A(\boldsymbol{\nu}), \tag{4.13}$$

where

$$\boldsymbol{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}, \tag{4.14}$$

$$\boldsymbol{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} v + Y_{\dot{r}} r \\ 0 & 0 & -X_{\dot{u}} u \\ -Y_{\dot{v}} v - Y_{\dot{r}} r & X_{\dot{u}} u & 0 \end{bmatrix}. \tag{4.15}$$

The total hydrodynamic damping matrix accounts for both linear and nonlinear terms, and can be derived based on surge resistance and cross-flow drag, or second order modulus functions. In the case of the latter, one can express the total damping as

$$\boldsymbol{D}(\boldsymbol{\nu}) = \boldsymbol{D} + \boldsymbol{D}_n(\boldsymbol{\nu}), \tag{4.16}$$

where $\boldsymbol{D}_n(\boldsymbol{\nu})$ is the non-linear damping matrix and $\boldsymbol{D}$ denotes the linear damping matrix. The linear damping matrix, using SNAME (1950) notation, is given as

$$\boldsymbol{D} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}. \tag{4.17}$$

The linear damping will be dominant for low-speed maneuvering and stationkeeping, while non-linear damping will dominate at high speed. Due to this, one often neglects linear damping when considering higher speeds. In Fossen (2011) a slightly different formulation is used, where the added Coriolis and Centripetal matrix and the damping matrix are collected into one matrix. This is of convenience as it is difficult to distinguish between the terms of these matrices.

4.2 Differential flatness

Differential flatness is a property of some controlled dynamic systems that allows for the trivialization of the trajectory planning task, without solving differential equations, while optionally simplifying the problem of feedback controller design to that of a set of decoupled linear time invariant systems. Differential flatness or flatness in short is roughly speaking equivalent to controllability, and will thereby apply to most systems. The flatness property allows for a complete parameterization of all system variables including states, inputs and outputs, in terms of a finite set of fictitious independent variables, named *flat outputs*, and a number of their time derivatives. (Sira-Ramírez and Agrawal, 2004)

4.2.1 Historical remark

The mathematicians E. Cartan and D. Hilbert can be considered as the forefathers of differential flatness, in the context of underdetermined sets of differential equations. The problem set by D. Hilbert in 1912, were that of the second-order Monge's equation

$$\frac{d^2y}{dx^2} = F(x, y, z, \frac{dy}{dx}, \frac{dz}{dx}),$$

which describes a set of underdetermined differential equations. Hilbert sought a solution to this equation, that is not based on the computation of integrals, but rather through a group of transformations called *invertible without integral*. Cartan followed suit by a rework of the question set by Hilbert, and showed how calculations on the Pfaff systems, permits a classification of the second-order Monge's equation, that admits a solution without integrals. Cartan also suggested the notion of absolute equivalence, but did not go as far as to define it in precise terms. (Rigatos, 2015)

The precise formulation of differential flatness were introduced by Professor Michel Fliess and his coresearcheres: Jean Levine, Philippe Martin and Pierre Rouchon. One of the first fundamental articles written by this team is devoted to flatness of nonlinear systems and the associated defect, which is the lack of flatness. (Fliess et al., 1992) The setting of this article is that of differential algebra, and the idea of flatness appears as a natural outcome of the equivalence problem. Some years later, the team published a complete exposition on all developments concerning flatness and defect, including a set of physical examples. (Fliess et al., 1995)

It soon became apparent that differential flatness could be recast in a purely differential geometric setting involving infinite jet spaces, differential varieties and Cartan fields. This idea did involve space and time coordinate transformation, thus bringing the Lie-Bäcklund transformation into attention. The complete recast of differential flatness in this new setting were published by the team in Fliess et al. (1999).

Today differential flatness is considered a great tool offering power, simplicity and generality in solutions to advanced control, state estimation and trajectory problems. For further reading on the history of differential flatness the interested reader is refereed to Sira-Ramírez and Agrawal (2004).

4.2.2 Flat systems

A system is said to be differentially flat if one can find a set of outputs, equal in number to the number of inputs, such that one can express all states and inputs as functions of these outputs and their derivatives. In this thesis a somewhat informal definition of differential flatness, based off of the description presented in Van Nieuwstadt and Murray (1998), will be adopted

Definition 4.1. Consider a nonlinear system described as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \quad \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{u} \in \mathbb{R}^m \tag{4.18}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \qquad \boldsymbol{y} \in \mathbb{R}^n, \tag{4.19}$$

where $\boldsymbol{x}$ denotes the state vector, $\boldsymbol{u}$ denotes the control input vector and $\boldsymbol{y}$ denotes the tracking output vector. Such a system is said to differentially flat if there exist a vector $\boldsymbol{z} \in \mathbb{R}^m$, known as the flat output, of the form

$$\boldsymbol{z} = \boldsymbol{\zeta}(\boldsymbol{x}, \boldsymbol{u}, \dot{\boldsymbol{u}}, \ldots, \boldsymbol{u}^{(r)}), \tag{4.20}$$

such that the states and the inputs can be described as

$$\boldsymbol{x} = \boldsymbol{\phi}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ldots, \boldsymbol{z}^{(q)}), \tag{4.21}$$

$$\boldsymbol{u} = \boldsymbol{\alpha}(\boldsymbol{z}, \dot{\boldsymbol{z}}, \ldots, \boldsymbol{z}^{(q)}), \tag{4.22}$$

where $\boldsymbol{\zeta}$, $\boldsymbol{\phi}$ and $\boldsymbol{\alpha}$ are smooth functions, and r and q are finite.

There exist more formal ways of defining differential flatness, one such way is presented in Rigatos (2015). In the case were the flat outputs are only described in terms of the states of the system, the system is often refereed to as being 0-differentially flat. Determining the flat outputs for a given system is in general difficult, as there does not exist any systematic method for doing so, except for that of linear systems and affine nonlinear single input systems. In such a way the determination of the flat outputs may resemble that of Lyapunov functions, in that educated guessing and physical intuition is often necessary.

Non-differentially flat systems

There exist several mathematically challenging nonlinear systems that are know to be non-differentially flat, some examples are the ball and beam and the inverted pendulum. The lack of flatness is defined through the defect of the system, which usually is represented by the set of state variables that are not expressible through the flat outputs, associated with the largest flat subsystem. A special class of non-differentially flat systems are known as Liouvillain systems. For these systems the defect variables can be expressed in terms of quadratures of differential functions of the flat outputs. (Sira-Ramírez and Agrawal, 2004)

4.2.3 Flatness and surface vessels

In trajectory problems and path planning the differential flatness property can be exploited by assigning functions to the flat outputs, thus obtaining trajectories for all system variables. It is this specific trait of differential flatness that will be exploited in this thesis. The obtained trajectories will also reflect the dynamics of the system.

The level of actuation is in the case of surface vessel detrimental in the process of proving the system to be differentially flat. If the vessel is fully actuated then most model representations could quite easily be proven to be flat. The case is entirely different in the case of underactuated surface vessels, for which most model representations are non-differentially flat. The horizontal plane model presented earlier are among these models. Albeit most underacted surface vessel models being non-differentially flat, it is possible to prove that some models are Liouvillain, which can be done for the model presented in Pettersen and Nijmeijer (1998). The proof is presented in Sira-Ramírez and Agrawal (2004).

One of the main objectives of this thesis is to design a path planning algorithm for an underactuated surface vessel, that utilized the differential flatness property of the vehicle. The model that will be used in the path planning algorithm, is a simplified version of the horizontal plane model presented in Chapter 4.1.3. The simplification of the model must be performed in order to obtain a differentially flat model. In any process where the complexity of a model is reduced certain aspect of the dynamics are lost, this is unfortunate in our case, but at the same time necessary. Some justification can be found if one considers the fact that most path planning algorithms view the vehicle as a point mass. It is also possible to assumed that the unmodeled dynamics are accounted for by the controller.

The following assumptions and simplifications to the horizontal plane model are done in order to obtain a flat model:

- The ship is assumed to have fore/aft symmetry, which would mean that all off-diagonal entries and couplings in the inertia and damping matrices can be eliminated.
- The non-linear damping is also excluded, this assumption is valid if the speed of the vessel is assumed low.

Under these assumption one obtains the following mathematical model

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{4.23}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu} = \boldsymbol{\tau}, \tag{4.24}$$

where $\boldsymbol{\tau} = [\tau_1, 0, \tau_3]$ and the system matrices takes on the following form

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix}, \qquad \begin{aligned} \boldsymbol{M} &= \text{diag}\{m_{11}, m_{22}, m_{33}\}, \\ \boldsymbol{D} &= \text{diag}\{d_{11}, d_{22}, d_{33}\}. \end{aligned} \tag{4.25}$$

The mathematical model for the surface vessel can be written on component form as

$$\dot{x} = u\cos(\psi) - v\sin(\psi) \tag{4.26a}$$
$$\dot{y} = u\sin(\psi) + v\cos(\psi) \tag{4.26b}$$
$$\dot{\psi} = r \tag{4.26c}$$
$$\dot{u} = vr - \beta_1 u + \tau_u \tag{4.26d}$$
$$\dot{v} = -ur - \beta_2 v \tag{4.26e}$$
$$\dot{r} = -\beta_3 r + \tau_r, \tag{4.26f}$$

where the following change in notation has been applied

$$m_{11} = m_{22}, \quad \beta_1 = \frac{d_{11}}{m_{11}}, \quad \beta_2 = \frac{d_{22}}{m_{22}}, \quad \beta_3 = \frac{d_{33}}{m_{33}}, \quad \tau_u = \frac{\tau_1}{m_{11}}, \quad \tau_r = \frac{\tau_3}{m_{33}}.$$

This model is differentially flat by Definition 4.1, if the flat outputs of the system are chosen as the coordinates of the vessel in the North-East plane, that is $\boldsymbol{z} = [x,\ y]$.

Proof. In order to prove that the ship model is differentially flat, we will first express the derivatives of Eq. (4.26a) and (4.26b)

$$\ddot{x} = (\dot{u} - v\dot{\psi})\cos(\psi) - (\dot{v} + u\dot{\psi})\sin(\psi), \tag{4.27}$$
$$\ddot{y} = (\dot{v} + u\dot{\psi})\cos(\psi) + (\dot{u} - v\dot{\psi})\sin(\psi). \tag{4.28}$$

Using these equations it can easily be proven that the following relations hold

$$\ddot{x} + \beta_2\dot{x} = (\dot{u} - v\dot{\psi} + \beta_2 u)\cos(\psi) - (\dot{v} + u\dot{\psi} + \beta_2 v)\sin(\psi), \tag{4.29}$$
$$\ddot{y} + \beta_2\dot{y} = (\dot{v} + u\dot{\psi} + \beta_2 v)\cos(\psi) + (\dot{u} - v\dot{\psi} + \beta_2 u)\sin(\psi). \tag{4.30}$$

The expressions inside the parentheses can be simplified by inserting Eq. (4.26d) and (4.26e), resulting in

$$\dot{u} - v\dot{\psi} + \beta_2 u = vr - \beta_1 u + \tau_u - vr + \beta_2 u = (\beta_2 - \beta_1)u + \tau_u, \tag{4.31}$$
$$\dot{v} + u\dot{\psi} + \beta_2 v = -ur - \beta_2 v + ur + \beta_2 v = 0, \tag{4.32}$$

by defining $\beta_u := \beta_2 - \beta_1$, Eq. (4.29) and (4.30) can be expressed as

$$\ddot{x} + \beta_2\dot{x} = (\beta_u u + \tau_u)\cos(\psi), \tag{4.33}$$
$$\ddot{y} + \beta_2\dot{y} = (\beta_u u + \tau_u)\sin(\psi). \tag{4.34}$$

Using these two expressions we obtain the following relation

$$\frac{\ddot{y}+\beta_2\dot{y}}{\ddot{x}+\beta_2\dot{x}} = \frac{(\beta_u u+\tau_u)\sin(\psi)}{(\beta_u u+\tau_u)\cos(\psi)} = \tan(\psi) \iff \psi = \tan^{-1}\left(\frac{\ddot{y}+\beta_2\dot{y}}{\ddot{x}+\beta_2\dot{x}}\right). \tag{4.35}$$

Thus Eq. (4.35) proves that ψ can be written as a function of the flat outputs and their derivatives. It is further possible to prove that the following relations hold

$$\dot{x}(\ddot{x}+\beta_2\dot{x}) = (\tau_u+\beta_u u)(u\cos(\psi)-v\sin(\psi))\cos(\psi), \tag{4.36}$$
$$\dot{y}(\ddot{y}+\beta_2\dot{y}) = (\tau_u+\beta_u u)(u\sin(\psi)+v\cos(\psi))\sin(\psi). \tag{4.37}$$

If the two expressions above are added together one obtains the following

$$\dot{x}(\ddot{x}+\beta_2\dot{x})+\dot{y}(\ddot{y}+\beta_2\dot{y}) = u(\tau_u+\beta_u u). \tag{4.38}$$

Further, the following can be proven to hold

$$(\ddot{x}+\beta_2\dot{x})^2+(\ddot{y}+\beta_2\dot{y})^2 = (\tau_u+\beta_u u)^2. \tag{4.39}$$

Dividing Eq. (4.38) by the square root of Eq. (4.39) one obtains the following

$$u = \frac{\dot{x}(\ddot{x}+\beta_2\dot{x})+\dot{y}(\ddot{y}+\beta_2\dot{y})}{\sqrt{(\ddot{x}+\beta_2\dot{x})^2+(\ddot{y}+\beta_2\dot{y})^2}}. \tag{4.40}$$

Thus Eq. (4.40) proves that u can be written as a function of the flat outputs and their derivatives. Further, it can be proven that the following holds

$$\dot{x}\ddot{y} = (u\cos(\psi)-v\sin(\psi))((u\dot{\psi}+\dot{v})\cos(\psi)+(\dot{u}-v\dot{\psi})\sin(\psi)), \tag{4.41}$$
$$\dot{y}\ddot{x} = (u\sin(\psi)+v\cos(\psi))((\dot{u}-v\dot{\psi})\cos(\psi)-(u\dot{\psi}+\dot{v})\sin(\psi)). \tag{4.42}$$

Subtracting Eq. (4.41) from Eq. (4.42) and through some intermediate calculations the following expression is obtained

$$\dot{y}\ddot{x}-\dot{x}\ddot{y} = v(\tau_u+\beta_u u). \tag{4.43}$$

Now dividing this expression by the square root of Eq. (4.39), one can derive the following expression for the sway velocity

$$v = \frac{\dot{y}\ddot{x}-\dot{x}\ddot{y}}{\sqrt{(\ddot{x}+\beta_2\dot{x})^2+(\ddot{y}+\beta_2\dot{y})^2}}. \tag{4.44}$$

Thus proving that v can be expressed as a function of the flat outputs and their derivatives. The derivative of Eq. (4.35) can be expressed as

$$\frac{\dot{\psi}}{\cos^2(\psi)} = \frac{(y^{(3)}+\beta_2\ddot{y})(\ddot{x}+\beta_2\dot{x})-(x^{(3)}+\beta_2\ddot{x})(\ddot{y}+\beta_2\dot{y})}{(\ddot{x}+\beta_2\dot{x})^2}. \tag{4.45}$$

To simplify this expression one can use the following identity

$$\frac{1}{\cos^2(\psi)} = \tan^2(\psi) + 1, \tag{4.46}$$

which implies that

$$\cos^2(\psi) = \frac{(\ddot{x} + \beta_2\dot{x})^2}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}. \tag{4.47}$$

Inserting this expression into Eq. (4.45) and using the relation given in Eq. (4.26c) one obtains

$$r = \frac{(y^{(3)} + \beta_2\ddot{y})(\ddot{x} + \beta_2\dot{x}) - (x^{(3)} + \beta_2\ddot{x})(\ddot{y} + \beta_2\dot{y})}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}. \tag{4.48}$$

Thus proving that r can be expressed as a function of the flat outputs and their derivatives. As we now have proved that all states can be written as functions of the flat outputs, the task of proving the inputs to be flat becomes trivial. To prove that the τ_u is flat we will write Eq. (4.26d) as

$$\tau_u = \dot{u} - vr + \beta_1 u. \tag{4.49}$$

Now finding the derivative of Eq. (4.40) and inserting the expressions for the remaining states into this equation, yields the following

$$\tau_u = \frac{(\ddot{x} + \beta_1\dot{x})(\ddot{x} + \beta_2\dot{x}) + (\ddot{y} + \beta_1\dot{y})(\ddot{y} + \beta_2\dot{y})}{\sqrt{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}}. \tag{4.50}$$

Thus proving that τ_u is indeed flat. To further prove that τ_r is flat, we will write Eq. (4.26f) as

$$\tau_r = \dot{r} + \beta_3 r. \tag{4.51}$$

Finding the derivative of Eq. (4.48) and inserting this into the expression above one obtains

$$\begin{aligned}
\tau_r = &\frac{(y^{(4)} + \beta_2 y^{(3)})(\ddot{x} + \beta_2\dot{x}) - (x^{(4)} + \beta_2 x^{(3)})(\ddot{y} + \beta_2\dot{y})}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2} \\
&- \frac{2\left((y^{(3)} + \beta_2\ddot{y})(\ddot{x} + \beta_2\dot{x}) - (x^{(3)} + \beta_2\ddot{x})(\ddot{y} + \beta_2\dot{y})\right)}{\left((\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2\right)^2} \Big((\ddot{x} + \beta_2\dot{x})(x^{(3)} + \beta_2\ddot{x}) \\
&\qquad\qquad + (\ddot{y} + \beta_2\dot{y})(y^{(3)} + \beta_2\ddot{y})\Big) \\
&+ \beta_3 \left(\frac{(y^{(3)} + \beta_2\ddot{y})(\ddot{x} + \beta_2\dot{x}) - (x^{(3)} + \beta_2\ddot{x})(\ddot{y} + \beta_2\dot{y})}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}\right).
\end{aligned} \tag{4.52}$$

Thus proving that τ_r can be written as a function of the flat outputs and their derivatives, and that the system is flat. □

Chapter 5

Path planning and optimization

This chapter will deal with the main objective of this thesis, which is to propose a path planning algorithm that utilizes Bézier curves. The theory presented in the previous chapters will serve as the basis for the development of this algorithm.

Problem description

The path planning algorithm that has been developed and that will be presented in this chapter, can be used to generate a path for "A-to-B moves". This means that the path is generated between two predetermined waypoints, indicating the initial and final position of the craft. The path is further generated in the North-East plane.

In order to generate complex paths, while taking both physical and workspace constraints into consideration, a Bézier spline will be used to represent the path. This means that the path will be piecewise parametric, and described by m Bézier curve segments. This will further allow us to represent the path as a superset, as presented in Chapter 3.1.2. The parameter interval will be chosen equal for the m curve segments as $\varpi \in [0, 1]$. This is done to lessen the complexity of the formulation, and due to implementational aspects. An alternative approach would be to use an increasing parameter interval for the different curve segments, by using the procedure presented in Chapter 3.2.3.

The developed path planning algorithm is intended to be used for an underactuated surface vessel, but might also be used for wheeled robots and fully actuated vessels. The mathematical model used to represent the ship in the proposed algorithm, is differentially flat, and presented in Chapter 4.2.3. The path tangential will further be used as the course for the vessel, this can be expressed as

$$\chi_{p,i}(\varpi) = \text{atan2}(x'_{p,i}(\varpi),\, y'_{p,i}(\varpi)), \quad i \in \mathcal{I}^m, \tag{5.1}$$

where atan2(x, y) is the four-quadrant version of arctan(y/x).

5.1 Related work

When designing a path planning algorithm there exist a large variety of different approaches that could be considered. Utilizing Bézier curves would fall into the field of spline based path planning. The term spline refers to a mathematical function that is piecewise defined by polynomials. Splines will therefore refer to a wide class of different functions. Among some of the types of splines that exist one will find B-splines, Hermite-splines, Cubic splines and Bézier splines. This section will present some related work, for which Bézier curves has been used in path planning. The most interesting articles within this field stems from the robotics community, and will as such be presented here.

Choi et al. (2010) presents two path planning algorithms utilizing Bézier curves for autonomous vehicles, with waypoints and corridor constraints. Both approaches presented in this paper joines Bézier curve segments together in order to achieve C^2 continuity, while satisfying the corridor constraints. The paper further describes the algorithm used in order to obtain the results. This paper resembles the work done in this thesis, in the aspect of continuity. It does however differ when it comes to the implementation of constraints. In the paper by Choi et al., the path is bounded by the corridor constraint, which is a bounded area that the path is constructed within. Another difference is that they use several waypoints in order to describe the path.

Ingersoll et al. (2016) presents a path planning system for a unmanned aerial vehicle, utilizing quadratic Bézier curves to model the UAV path. In this paper, the path planning has been modeled as a single objective optimization problem, utilizing a receding horizon approach. They have further constrained the path, such that obstacle collision is avoided, and they have also accounted for flight aerodynamic constraints. The system that has been developed in this paper includes both dynamic and static obstacles, and the results shows that the system is capable of generating a near optimal solution. This paper bear some similarities to the work done in this thesis, in the way they have accounted for static obstacles and how they use several curve segments in order to generate the path.

An interesting paper by Wu and Snášel (2014), describes how Bézier curve base path planning can be used in robot soccer. The paper combines the functions of path planning, obstacle avoidance, path smoothing and posture adjustment together. This paper describes an approach were the obstacles are considered as control points for the Bézier curve, and further describes how a path can be planned in real time. They also describe the construction of a new curve, which optimizes the shape of the Bézier curve. This paper showcase some of the possibilities that exist when utilizing Bézier curves in a path planning, and how they can be used in real time planning.

In Mehdi et al. (2015) a collision prediction and avoidance algorithm for multi-vehicle time-critical cooperative missions is presented. The path replanning utilizes Bézier curves, and the algorithm presented changes the shape of the vehicles already planned trajectory, by adding an appropriated detour. The algorithm presented allows for safe operations in a wide range of different collision scenarios. This paper further describes the possibilities Bézier curves has to offer in a path planning system.

5.2 Path description

In order to fully utilize the versatility the Bézier spline has to offer, it is important to analyze the behaviour of this type of spline. In order to do so, a set of desirable properties that is to be imposed on the path will be defined. Further, with a basis in these properties an analysis on the degrees of freedom the Bézier curve has to offer will be performed.

5.2.1 Desirable properties

The desirable properties can be divided into two categories, those related to the continuity of the path, and those who facilitates the process of planning. All properties will be presented in this section.

Continuity in course

A minimum requirement when generating a path consisting of several path segments should be that all segments are connected at their endpoints, which would imply that the path is C^0 continuous. Another fundamental property in path planning is continuity in course, which is achieved by imposing either G^1 or C^1 continuity. If one were to not impose this restriction, it would imply that the vehicle is able to execute infinite rotational acceleration.

Continuity in curvature

For a marine craft following a path, a curvature discontinuity entails a discontinuity in the desired lateral acceleration of the vehicle, due to the relationship (Lekkas et al., 2013)

$$|\boldsymbol{\alpha}| = \kappa |\boldsymbol{u}|^2, \tag{5.2}$$

where $\boldsymbol{\alpha}$ is the lateral acceleration vector, $\boldsymbol{u}$ is the velocity vector and κ is the curvature. A discontinuity will affect the input to the heading controller and the vehicle performance in general. Curvature discontinuities could also imply strong and abrupt forces to the vehicle itself. This encourages the design of a path with continous curvature, which is achieved by imposing either G^2 or C^2 continuity.

Locality

Locality is a property that defines how far along the spline the effect of a small local change in the position of one control point, still can be seen.(Levien and Séquin, 2009) When dealing with a single curve segment, a perturbation in a single control point would lead to a change of that entire segment. This is unavoidable when designing a path using Bézier curves, it is therefore of greater interest to study the locality property of the spline. Normally when generating a path there will be a trade off between locality and continuity, this is due to the continuity setting certain restrictions on the neighbouring curve segments. The locality property can be viewed as a measure for how far a change in a single control point will ripple throughout the entire spline. So in order to effectively plan a path, it is preferable that a change in one curve segment will only induce changes in a closed

neighbourhood of that segment. When designing a path with C^2 continuity, it is only possible to restrict the ripple effect to one of the adjoining segments.

Boundary conditions

A property that is further going to facilitate effective planning is related to the boundary conditions of the path. These conditions would normally include the position, heading angle, steering angle and speed for the vehicle in the endpoints of the path. Through incorporation of these conditions in the path generation, the vehicle should be able to follow the path without major deviations. Including these conditions in the path planning would also facilitate updates of paths that are already being carried out. In this thesis only conditions for the position and heading angle in the endpoints are included.

5.2.2 Analysis of the Bézier spline

The desirable properties presented in the previous section defines a set of constraints that affect the Bézier spline. In order to fulfill all these requirements, it is necessary to perform an analysis on the degrees of freedom available for each Bézier curve segment, and the spline as a whole. This is done in order to determine the degree of the Bézier curve segments that will grant all the desirable properties. It is also of interest to keep the degree as low as possible, such that numerical instabilities are avoided. (Skrjanc and Klancar, 2007)

A single Bézier curve of the n-th degree will by Definition 3.1 be C^n continuous, and as such a discontinuity in the spline will be in one of the joints. The focus of the following discussion will therefore be on the continuity in the joints, and it will be assumed that parametric continuity is to be imposed. The Bézier curve is defined by two polynomial parametric equations, and as such for simplicity reasons, a general polynomial will be used in the following discussion. If the coordinate equations are of the n-th degree, they will yield $n+1$ degrees of freedom, and they can be expressed as

$$f_i(\varpi) = a_1\varpi^n + a_2\varpi^{n-1} + ... + a_n\varpi + a_{n+1}, \quad \varpi \in [0,\, 1], \tag{5.3}$$

where $i \in \mathcal{I}^m$ is used to denote the curve segment. If the path consists of m segment there will be a total of $(n+1)m$ degrees of freedom available. To ensure that the endpoints of the different curve segments coincide, the following condition must be met

$$f_i(1) = f_{i+1}(0), \quad i \in \mathcal{I}^{m-1}. \tag{5.4}$$

Further by using Definition 3.2, the following conditions can be formulated to ensure that the path is C^2 continous

$$f_i'(1) = f_{i+1}'(0), \quad i \in \mathcal{I}^{m-1}, \tag{5.5}$$

$$f_i''(1) = f_{i+1}''(0), \quad i \in \mathcal{I}^{m-1}. \tag{5.6}$$

Imposing these continuity conditions will reduce the available degrees of freedom by $3(m-1)$, resulting in a total of $(n-2)m+3$ degrees of freedom available. To be able to specify the position of the endpoints the following constraint can be formulated

$$f_1(0) = \mathrm{WP}_0, \quad f_m(1) = \mathrm{WP}_f, \tag{5.7}$$

where WP_0 and WP_f denotes the initial and final position of the vehicle. Imposing these constraints means that the degrees of freedom are reduced by two. The heading in the endpoints can be restricted by imposing the following constraint on the first derivatives

$$f_1'(0) = t_0, \quad f_m'(1) = t_f, \tag{5.8}$$

where t_0 and t_f denotes vectors with the same direction as the heading in the initial and final position respectively. Imposing the constraints related to the boundary conditions will result in $(n-2)m-1$ degrees of freedom available for the spline. Assuming that the spline is constructed by using two or more curve segments, the lowest order curve that would yield any available degrees of freedom is a curve of the third order.

The degrees of freedom in this discussion gives the number of parameters one can independently place for the polynomial. The remaining parameters will then be given directly by the constraints previously presented, in order to achieve the continuity requirements. For a Bézier spline one can view the degrees of freedom as the number of control points one can individually place in the plane. If one were to regard the x any y position of the control points independently, one would get twice the amount of degrees of freedom. As an example lets regard a Bézier spline consisting of two cubic curves. In this case one would have one degree of freedom available. This means that after fulfilling the constraints set by the boundary conditions, one would be able to place one control point freely in order to achieve C^2 continuity.

Cubic and Quartic Bézier splines

If the path is constructed by a single cubic Bézier curve, one would be able to fulfill the requirements for C^2 continuity, and boundary conditions. However, using a single lower degree curve would subsequently mean that one is not be able to represent complex shapes. In the case where the spline is construed using several cubic Bézier curves, one would have $m-1$ degrees of freedom available. Simple deduction then dictates that by raising the number of curve segments, one would be able to create a complex path, however at the cost of having to use many segments. Cubic Bézier curves also introduces a drawback related to the locality property, in that a small perturbation in a single curve segment might propagate throughout the entire spline.

The quartic Bézier spline offers all the same properties as the cubic Bézier spline, while also introducing the possibility of freely setting the second derivative at the endpoints of the spline, with additional degrees of freedom available. Being able to freely set the second derivative at the endpoints of the spline, would allow one to specify the curvature in these points. The quartic Bézier spline introduces $2m-1$ degrees of freedom, meaning that for a spline consisting of $m \geq 2$ segments one would introduce more degrees of freedom per parameter, which also is beneficial. The quartic Bézier spline also lack the same locality property as the cubic Bézier spline.

Quintic Bézier splines

The quintic Bézier spline offers the same properties as the quartic Bézier spline, while also introducing the much desired locality property. By using Bézier curves of the fifth degree each segment will be C^5 continous. One will also achieve C^2 continuity in the joints, while avoiding a small change in one segment from propagating further then to one of the adjoining segments. Since the quintic Bézier spline offers all the desirable properties, this type of spline will be used in the path planning algorithm. The quintic Bézier spline will also introduce more degrees of freedom per parameter, than the corresponding quartic spline.

The properties that each of the different spline types has to offer is summarized in Table 5.1. In this table the best combination of locality and continuity is presented, other combinations are also possible.

Table 5.1: Properties of Bézier splines of different degrees.

	Cubic	Quartic	Quintic
C^1 continuous	✓	✓	✓
C^2 continuous	✗	✓	✓
Locality	✓	✗	✓
Freely set first derivative at endpoints	✓	✓	✓
Freely set second derivative at endpoints	✗	✓	✓

In addition to the desired properties presented in the previous section, Bézier curves also provides another set of properties. One of them is the convex hull property, which has been mentioned previously. There is also a strong correlation between the parameters of the Bézier curve and its shape. This property is especially useful as it facilitates effective planning. If there were to be a weak correlation between the shape and the parameters, inverse math would often be involved.

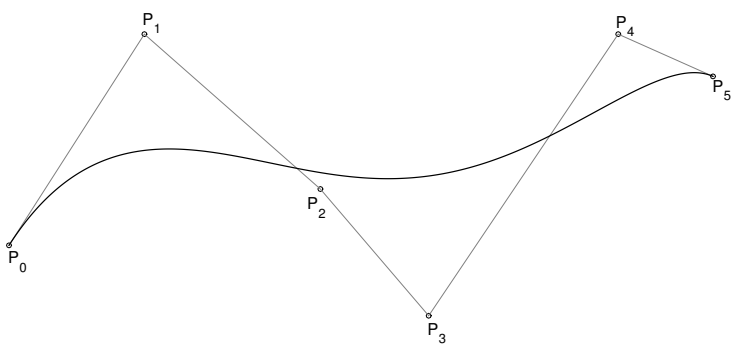

Figure 5.1: A fifth order Bézier curve with the control polygon.

5.3 Path optimization

There exist several approaches to path generation, some are given by a set of parameters, others use optimization. In this thesis an optimization problem will be formulated in order to generate an optimal path. This section will present the approach used in order to do so.

Optimization is an important tool in decision science and in the analysis of physical systems, and it is widely used in sciences, engineering, management and economics. This tool takes use of an *objective*, which is a quantitative measure of the performance of the system under study. The objective will further depend upon certain characteristic of the system, called *variables* or unknowns. The goal of the optimization is then to optimize the objective, with respect to the variables. These variables are often restricted, or *constrained* in some way. The process of identifying the objective, variables and constraints for a given problem, is know as *modeling*. The process of modeling is often viewed as the most important step in the optimization process, as the complexity of the model will affect solution time and the insight the results may offer. Once the model has been formulated, the next step is to choose an optimization algorithm, such that the problem may be solved. During this process special care must be taken. (Nocedal and Wright, 2006) A general formulation for an optimization problem is as follows

$$\min_x J \qquad \text{subject to} \quad \begin{array}{l} c_i(x) = 0,\ i \in \mathcal{E}, \\ c_i(x) \geq 0,\ i \in \mathcal{I}, \end{array}$$

where J is the objective function, and $\mathcal{E}$ and $\mathcal{I}$ are sets of indices for equality and inequality constraints, respectively. The optimization problem in this thesis can be classified as non-linear, due to the nature of the objective function and the constraints.

5.3.1 Decision variables

The shape of a Bézier curve or a Bézier spline is dependent upon the placement of the control points, and as such it is natural to choose these control points as decision variables for the optimization problem. These decisions variables can be gathered in a vector as

$$\boldsymbol{x}_1 = \{\mathbf{p}_{0,1}, \ldots, \mathbf{p}_{n,1}, \mathbf{p}_{0,2}, \ldots, \mathbf{p}_{n,2}, \ldots, \mathbf{p}_{0,m}, \ldots, \mathbf{p}_{n,m}\}, \tag{5.9}$$

where n is the degree of the curve and m is the number of curve segments. In addition to these decision variables, as later will show, it becomes necessary to include another set of decision variables. These variables will be presented as they are introduced in conjunction with the constraint. For the optimization problem all decision variables can be gathered in a single vector as

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\}, \tag{5.10}$$

where $\boldsymbol{x}_2$ is given by Eq. (5.21) and $\boldsymbol{x}_3$ is given by Eq. (5.25). Note that $\boldsymbol{x}_2$ is only included if geometric continuity constraints are used.

5.3.2 Objective function

The objective function is in a minimization problem such as this often known as the cost function. The goal of the optimization is to find the lowest value of the cost function, while satisfying the constraints. In trajectory or path optimization the goal is often to create a path where one of the following is minimized; fuel consumption, energy usage, time consumption or length. Sophisticated features such as collision avoidance, formation control and synchronization can also be included. The cost function can also be designed with several of these goals in mind.

The cost function in this thesis takes use of the differential flatness property of the vehicle, and assigns a cost to each path reflecting the dynamic capabilities of the vehicle on that path. In this process the model and calculated states presented in Chapter 4.2.3 are used. The flat outputs are as previously presented the coordinates of the vehicle in the North-East plane. Thus by creating a path in the same plane, it is possible to use the x and y coordinates of the Bézier curves as functions for the flat outputs. The derivatives of the Bézier curve can easily be obtained by using the approach presented in Chapter 3.2.6.

The cost function is further designed to optimize the path with respect to the associated energy. A fundamental assumption in the derivation of this cost function is that the time dependent flat outputs, can be directly substituted with the corresponding Bézier curve equivalents. It is further assumed that there is no sideslip along the path, which implies that surge is the only nonzero velocity component. This also implies that the heading, course and speed vectors are parallel, and that they coincide with the unit tangent of the path. Under these assumptions Newton's second law of motion, and the force and work relation can be describe only in surge, as

$$\sum F = m\dot{u}, \quad W_{ab} = \int_a^b F(s)\, ds, \tag{5.11}$$

where W_{ab} denotes the work performed between the points a and b, and s denotes length. With basis in these equations it can be stated that the integral of a well defined function of acceleration, can be related to energy. This allows the formulation of the following cost function

$$J = \int_a^b g(\dot{u})\, ds, \quad g(\dot{u}) = \sqrt{|\dot{u}|}. \tag{5.12}$$

The absolute value in this function is included to avoid positive and negative terms from canceling each other out. In the further derivation of the cost, it is assumed that $ds = d\varpi$. Now by substituting $\dot{u}(t)$ with $u'(\varpi)$, and by accounting for all curve segments, the following cost function can be derived

$$J = \sum_{i=1}^{m} \left[\int_0^1 \sqrt{|u_i'(\varpi)|} d\varpi \right], \tag{5.13}$$

where i denotes the curve segment number, and $u'(\varpi)$ is found by differentiating Eq. (4.40) and inserting the Bézier curve equivalents of x and y.

5.3.3 Constraints

The constraints are included in the optimization problem in order to create a path, for which the vehicle is able to follow. These constraints must be formulated in such a way that they are dependent upon the decision variables. Constraints can in general be divided into two categorize; soft constraints and hard constraint. Soft constraint are penalized in the objective function, while hard constraints sets conditions for the variables, which are required to be satisfied. This section will present all constraints that are included in the optimization problem.

Parametric continuity

To ensure that the path is continues, or have C^0 continuity, it is required that the last control point of any curve segment is equal to the first control point of the next segment. For a fifth order Bézier spline these constraints can be formulated as

$$\mathbf{p}_{5,i} = \mathbf{p}_{0,i+1}, \quad i \in \mathcal{I}^{m-1}, \tag{5.14}$$

where i denotes the curve segment number and the numerals denote the control point of that respective curve segment. Further, by using Definition 3.2 and the derivatives given by Eq. (3.32), the C^1 continuity constraints can be formulated as follows

$$\mathbf{p}_{5,i} - \mathbf{p}_{4,i} = \mathbf{p}_{1,i+1} - \mathbf{p}_{0,i+1}, \quad i \in \mathcal{I}^{m-1}. \tag{5.15}$$

This expression can further be simplified by using Eq. (5.14), which yields

$$\mathbf{p}_{1,i+1} + \mathbf{p}_{4,i} = 2\mathbf{p}_{5,i}, \quad i \in \mathcal{I}^{m-1}. \tag{5.16}$$

Imposing these constraint will ensure that the tangent of the path remains continues. Further, to ensure that the path will have C^2 continuity and thus have a continues curvature, the following constraints will be imposed

$$\mathbf{p}_{2,i+1} - 2\mathbf{p}_{1,i+1} + \mathbf{p}_{0,i+1} = \mathbf{p}_{5,i} - 2\mathbf{p}_{4,i} + \mathbf{p}_{3,i}, \quad i \in \mathcal{I}^{m-1}, \tag{5.17}$$

where Definition 3.2 and the second derivatives given by Eq. (3.33) have been used. This expression can also be simplified by using the relation described in Eq. (5.14), which yields

$$\mathbf{p}_{2,i+1} - 2\mathbf{p}_{1,i+1} = -2\mathbf{p}_{4,i} + \mathbf{p}_{3,i}, \quad i \in \mathcal{I}^{m-1}. \tag{5.18}$$

The continuity constraints given above can be classified as linear equality constraints, and will yield a total of $6(m-1)$ constraints if one views the x and y positions of the control points independently. All these constraint can easily be extended to any Bézier spline of degree $n \geq 2$, by noting that the constraints only include the three first and three last control points of any curve.

Geometric continuity

To ensure that the course and curvature of the path remains continues, it is also possible to formulate a set of constraints using geometric continuity. These constraints will be more relaxed, and put less emphasis on the particulars of the parameterization. To ensure that the path is G^2 continuous the following constraints must be imposed (Barsky and DeRose, 1989)

$$\mathbf{p}_i'(1) = a_i \mathbf{p}_{i+1}'(0), \qquad i \in \mathcal{I}^{m-1}, \tag{5.19}$$

$$\mathbf{p}_i''(1) = a_i^2 \mathbf{p}_{i+1}''(0) + b_i \mathbf{p}_{i+1}'(0), \quad i \in \mathcal{I}^{m-1}, \tag{5.20}$$

where the derivatives for a fifth order spline are given by Eq. (3.32) and (3.33). Further, a_i is some strictly positive constant and b_i is some constant of arbitrary value. To fully utilize the additional freedoms the geometric continuity entails, it is necessary to include a_i and b_i as decision variables, this can be formulated as

$$\boldsymbol{x}_2 = \{a_1, \ldots, a_{m-1}, b_1, \ldots, b_{m-1}\}, \tag{5.21}$$

$$a_1, \ldots, a_{m-1} > 0. \tag{5.22}$$

Note that the bounds for a_i must be included in the optimization.

Boundary conditions

The initial and final conditions can be formulated as a set of linear equality constraints. To ensure that the path begins and ends in certain points, it is possible to use Eq. (5.7), which will give the following constraints

$$\mathbf{p}_{0,1} = \mathrm{WP}_0, \quad \mathbf{p}_{5,m} = \mathrm{WP}_f. \tag{5.23}$$

Further, to ensure that the heading in the endpoints of the path, corresponds with some predetermined desired value. A set of constraints related to the tangent of the path in these points must be formulated. These constraints will be given as

$$l_0 \begin{bmatrix} \sin(\psi_0) \\ \cos(\psi_0) \end{bmatrix} = 5(\mathbf{p}_{1,1} - \mathbf{p}_{0,1}), \quad l_f \begin{bmatrix} \sin(\psi_f) \\ \cos(\psi_f) \end{bmatrix} = 5(\mathbf{p}_{5,m} - \mathbf{p}_{4,m}), \tag{5.24}$$

where ψ_0 and ψ_f denotes the heading angle in the first and last waypoint, respectively. Further, l_0 and l_f are introduced as strictly positive decision variables, representing the length of the tangents in their respective endpoints. These constraints closely resembles that of Eq. (5.19), as they only restrict the direction of the tangents and not the magnitude. The additional decision variables and bounds are given as

$$\boldsymbol{x}_3 = \{l_0, l_f\}, \tag{5.25}$$

$$l_0, l_f > 0. \tag{5.26}$$

Turning radius

One of the maneuvering characteristics of a surface vessel is its turning circle, which describes the ship's steady turning radius and how well the steering machine performs under course-changing maneuvers. (Fossen, 2011) When a surface vessel is moving with a constant speed and a maximum rudder deflection, it will eventually begin to move in a circle with a constant radius, representing the smallest turn the vessel is able to perform. This encourages the design of a path with no turns smaller than this minimum turning radius. By taking note of the inverse proportionality of Eq. (3.13), the minimum turning radius of the vessel can be related to the curvature as

$$\kappa_{max} = \frac{1}{R_{min}}, \tag{5.27}$$

where R_{min} is the minimum turning radius and κ_{max} is the corresponding maximum curvature. Using this relation, it is possible to formulate the following constraint for the curvature of the path

$$\kappa_i(\varpi) < \kappa_{max}, \quad i \in \mathcal{I}^m, \tag{5.28}$$

where the curvature of the path is found by using Eq. (3.14). This constitutes a non-linear inequality constraint.

Static obstacles

Environmental constraints will be included in the optimization as static obstacles, representing forbidden zones that the ship should not sail through. Each obstacle will be represented by a circle with radius r_j, and center in $\boldsymbol{c}_j = (x_j, y_j)$ in the North-East plane. If one considers an obstacle field consisting of l elements, the following constraint can be formulated

$$r_j \leq |\mathbf{p}_i(\varpi) - \boldsymbol{c}_j|, \quad i, j \in \mathcal{I}^m \times \mathcal{I}^l. \tag{5.29}$$

This constraint should hold for any pair of i and j in order to deem the path feasible.

While implementing these constraints enables the generation of paths circumventing a large set of different obstacles, it requires that it is verifiable for all $\varpi \in [0, 1]$. An alternative approach exist where one takes advantage of the convex hull property of the Bézier curve. Using this property one could independently of the internal parameter of the Bézier curve segments, verify whether the path is feasible or not, by for instance verifying whether the area of an obstacle and the convex hull overlaps. A drawback with this approach does however exist, in that while the areas of the convex hull and the obstacles overlap, the path in itself might not cross the boundaries of the obstacles. This drawback can be worked around by using de Casteljau algorithm presented in Chapter 3.2.4. Using this algorithm a single segment can be divided into smaller parts, with a new set of convex hulls. Thus allowing for a more accurate description of the area containing the curve. This approach is only presented as a concept, and has not been implemented.

5.4 Implementation

This section focuses on the implementation of the proposed path generation algorithm, and is intended to give some insight in the decisions made in this process.

5.4.1 Simulation environment

The proposed optimization problem has been implemented in the MathWorks™ environment MATLAB®. The optimization solver utilized in this implementation is called `fmincon` and is provided by the Optimization Toolbox™. This solver provides several different algorithms, where the SQP (Sequential Quadratic Programming) algorithm is used.

5.4.2 Initialization

In the implementation of the proposed path generation algorithm, effort was put into making the program capable of running a large variety of different scenarios, without having to change the formulation of the problem. In the initialization of the program the user is able to specify the degree of the Bézier curve segments, the number of curve segments and the type of continuity. All constraints are then constructed based on this information. This gives the implemented program some versatility, in that one is able to construct paths at a fast pace for a wide range of different environments. The parameters that the user is able to specify in the initialization is summarized in the table below:

Table 5.2: Parameters for initialization.

Initial conditions	(x_0, y_0, ψ_0)	Min. Turning radius	R_{min}
Final conditions	(x_f, y_f, ψ_f)	Number of Obstacles	l
Number and Degree of Curve segments	(n, m)	Min. and Max radius of Obstacles	(r_{min}, r_{max})

The decision variable vector for the optimization is generated based on the degree of the Bézier curve segments, the number of segments, and the choice of continuity. In order to reduce the size of this vector, the first control point for all segments where $m \geq 2$ is omitted. This simplification reduces the complexity of the problem, and is based on the constraint give by Eq. (5.14). This does however demand some consideration in the implementation of the objective and constraints, and in post processing. The optimization solver also requires that an initial guess to the values of the decision variables are made. In the current implementation a generic initial guess is provided, taking the boundary conditions into consideration, this is done such that all constraints have a numerical value.

A simple script that generates random obstacles has also been developed. The user is able to specify the number of obstacles, and the minimum and maximum radius of these obstacles, an obstacle field is then generated within some bounded area specified by the position of the endpoints of the path. This script has been used a lot in the testing phase of the algorithm, as it allows for a fast generation of different environments.

5.4.3 Path representation

In the current implementation the path and all relevant information about the path is described through the use of vectors. In this chosen approach the curve parameter is represented by a vector containing values within the unit domain, as

$$\boldsymbol{\varpi} = [\varpi_0, \varpi_1, \ldots, \varpi_{i-1}, \varpi_{n_\varpi}], \quad i \in [0, n_\varpi], \tag{5.30}$$

where i is an index. The values of this vector is calculated as follows

$$\varpi_i = i\delta\varpi, \quad \delta\varpi = \frac{1}{n_\varpi}, \tag{5.31}$$

where $\delta\varpi$ describes the incremental change in the parameter when dividing the interval into n_ϖ different parts. Using this approach the most accurate representation of the path is achieved when choosing n_ϖ sufficiently small. This approach allows the representation of all curve segments, and the associated information in terms of a database. The information associated with the path used in the optimization and in the post processing is presented in the table below

Table 5.3: Relevant path information for a single curve segment.

$\boldsymbol{\varpi}$	$\boldsymbol{x}_p(\boldsymbol{\varpi})$	$\boldsymbol{y}_p(\boldsymbol{\varpi})$	$\ldots$	$\boldsymbol{x}_p^{i'}(\boldsymbol{\varpi})$	$\boldsymbol{y}_p^{i'}(\boldsymbol{\varpi})$	$\boldsymbol{s}(\boldsymbol{\varpi})$	$\boldsymbol{\kappa}(\boldsymbol{\varpi})$	$\boldsymbol{\chi}(\boldsymbol{\varpi})$
ϖ_0	$x_p(\varpi_0)$	$y_p(\varpi_0)$	$\ldots$	$x^{i'}(\varpi_0)$	$x^{i'}(\varpi_0)$	$s(\varpi_0)$	$\kappa(\varpi_0)$	$\chi(\varpi_0)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
ϖ_f	$x_p(\varpi_f)$	$y_p(\varpi_f)$	$\ldots$	$x^{i'}(\varpi_f)$	$x^{i'}(\varpi_f)$	$s(\varpi_f)$	$\kappa(\varpi_f)$	$\chi(\varpi_f)$

The information given by the database includes position, derivatives, arc length, curvature and course angle for the different values of the curve parameter. The arc length is found through approximation using Eq. (3.11), and describes the length of the path from the initial position up till the given value of the curve parameter. Further, the curvature used in the optimization is given by Eq. (3.14), while in the post processing the signed curvature is used.

5.4.4 Constraints and objective

The continuity constraints are generated dependent upon the choice of parametric or geometric continuity set by the user, and only one set of constraints are included in the optimization. The parametric continuity constraints are represented in terms of matrix operations, where the size of the matrices are determined by the number and degree of the curve segments. The boundary conditions are also included in the same way.

Since all the information about the path is represented in terms of a database, the curvature constraint can not be implemented as earlier presented. The curvature is in the implementation represented as a vector of values, and the constraint must therefore be reformulated for this purpose. The curvature constraint is therefore implemented by demanding that all

values of the curvature vector, is below the value set by the maximum curvature.

The obstacle constraint most also be reformulated in the same manner as the curvature constraint. However, since the obstacle constraint demands that one verifies that it holds for all pairs of curve segments and obstacles, this will create a large set of constraints. In order to reduce the amount of constraints that this would constitute, a different approach has been implemented. In this approach only the minimum distance between the obstacles and the curve segments are considered. This effectively reduces the amount of constraints, and it proved to be more effective then verifying whether all points along the curve where within the allowed region. This can be formulated as

$$r_j \leq \min\{|\mathbf{p}_i(\varpi) - \boldsymbol{c}_j|\}, \quad i, j \in \mathcal{I}^m \times \mathcal{I}^l \tag{5.32}$$

The integrals of the cost function is in the implementation approximated using the trapezoidal rule. This is done since it drastically lessens the complexity of the implementation, and since there might not exist any explicit solution to these integrals. Using the trapezoidal rule the values of the integrals will be given as

$$\int_0^1 u'(\varpi)\, d\varpi \approx \frac{\delta\varpi}{2}(u'(\varpi_0) + 2u'(\varpi_1) + \cdots + 2u'(\varpi_{n_\varpi - 1}) + u'(\varpi_{n_\varpi})). \tag{5.33}$$

5.4.5 Solver and options

The optimization solver used in the implementation were chosen after investigation of which solvers were capable of solving the given problem. During this process an attempt on using the genetic algorithm were performed, but this algorithm did not succeed in finding any solutions within reasonable time. The chosen solver also provides other options to which algorithms that could be used, among which the SQP algorithm proved to be the most reliable and consistent among them. The SQP algorithm is a gradient based optimization method and the gradients are, if not provided, found by the use of finite difference. The solver has an option that allows the user to specify the finite difference step size (FDSS), this option has been exploited in the current implementation. This is due to the current implementation being somewhat lacking in consistency, and as such it is not always straight forward to find a solution. This will be further discussed in Chapter 7. The process for which the results have been obtained can be summarized in three steps as follows:

Step 1: The problem is solved with default options. If a solution is found one proceeds directly to Step 3.
Step 2: The problem is solved with a higher value for the FDSS or without an objective function. When the objective function is discarded the solver puts larger emphasis on the constraints.
Step 3: The problem is solved with an increasing lower FDSS, until a satisfactory result is obtained. For each time the problem is solved anew, the decision variables from the previous iteration are used as the initial guess.

Chapter 6

Numerical simulations and results

This chapter presents a series of numerical simulations, that is used to evaluate the efficacy of the proposed algorithm. The different scenarios are designed to show how the Bézier spline can be effectively used in a wide range of different environments. Effort has therefore been put into using different boundary conditions and parameters.

6.1 Preliminary comments

In what follows a total of four simulation scenarios will be presented. The problem statement for each scenario is summarized in a table, with the same outline as that of Table 5.2. The first figure of all scenarios depicts the path in the North-East plane, for these figures different colors are used to indicate the different path segments. Figures depicting the course angle and curvature along the path are also included, for these figures the transition point between the different segments are indicated by a red circle.

It should further be mentioned that the obstacle constraint has been implemented as previously presented, without enforcing any additional clearing between the obstacles and the path. This means that any path that is not in contact with the obstacles are deemed feasible, if all other constraints are satisfied. This combined with the scale used in the figures, explain why the path is so close to the obstacles. So to clarify, the paths that are presented constitutes feasible solutions to the given problems.

The cost function has not been explicitly given, it will however be described through the derivatives of the flat outputs and a coefficient. This coefficient where in Chapter 4.2.3 denoted β_2, and is further dependent upon the inertia and linear damping in sway. The numerical value for this coefficient, used in the simulations is 0.0238. This value has been derived with basis in the modelling of Cybership II presented in Skjetne et al. (2004).

6.2 First simulation scenario

The environment in this scenario has been constructed such that it might resemble that of a passage near a port. The intention of this scenario is to show how the Bézier spline can be effectively used in path planning, for a realistic environment. The passage has further been made large enough for the generation of a path between the initial and final positions, going through the passage without exceeding the curvature constraint. The path is constructed with parametric continuity.

The problem specification is summarized in the table below:

Table 6.1: Parameters for initialization in the first scenario.

Initial conditions	$(0, 0, 90°)$	Min. turning radius	100
Final conditions	$(1400, 1200, 45°)$	Number of obstacles	12
Degree and number of curve segments	$(5, 4)$	Min. and Max. radius of obstacles	$(75, 75)$

The following figures shows the result obtained for this simulation scenario:

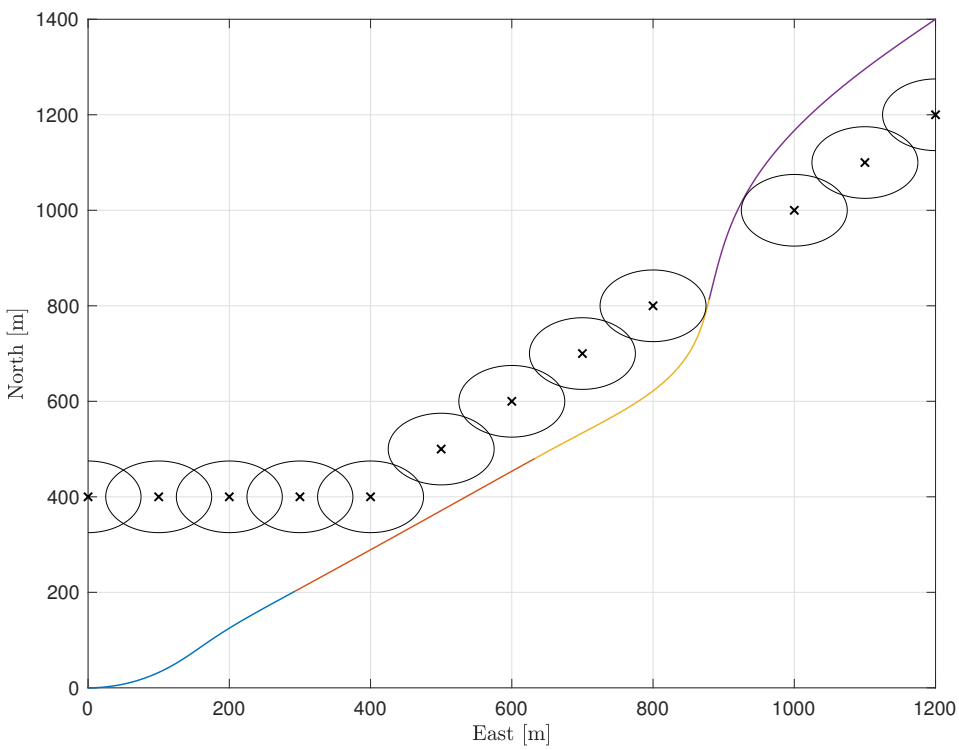

Figure 6.1: Graphical representation of the generated path and obstacles in the first scenario.

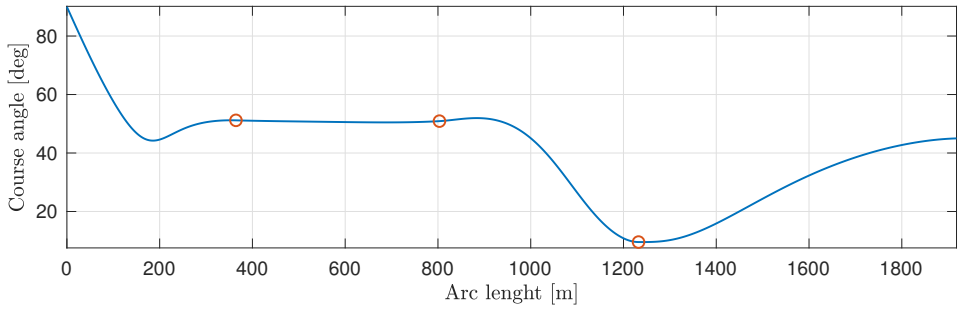

Figure 6.2: Graphical representation of the calculated course angle along the path for the first scenario.

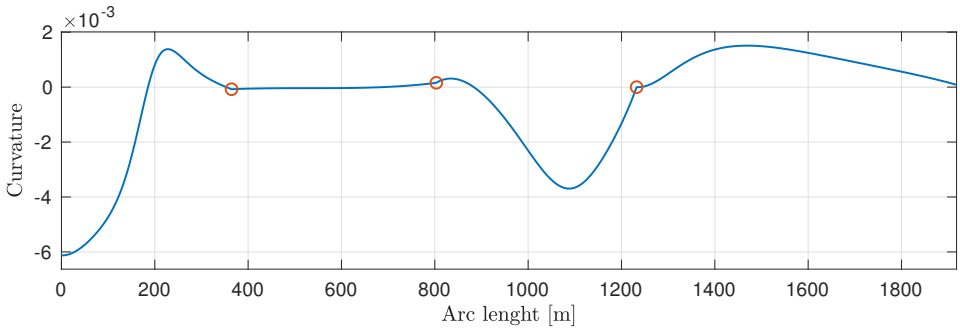

Figure 6.3: Graphical representation of the calculated curvature along the path for the first scenario.

The results shown in this scenario has been obtained through three iterations, using the approach presented in Chapter 5.4.5. The path is depicted in Figure 6.1, and is seemingly straight with two major course changes. Figure 6.2 is more descriptive in terms of the course changes, and shows how the course change from $90°$ to approximately $50°$ during the first segment. The path goes through the passage during the later part of the third segment and the fourth segment. During this turn, the course changes from approximately $50°$ to $5°$ and then back to $45°$. The curvature shown in Figure 6.3, can be used to describe how straight path is, and through inspection of this figure it can be verified that the second segment of the path is straight. The figure further shows how the curvature is quite low during the second course changing maneuver, indicating that the turning radius is low during this maneuver.

6.3 Second simulation scenario

The environment in this scenario, as with the previous, has been constructed to resemble a realistic environment. The obstacles has been placed with the intention of creating an environment resembling that of a ship passage. To further make the construction of the path a bit more difficult, some minor obstacles has been included in the passage. The path is constructed with parametric continuity.

The problem specification is summarized in the table below:

Table 6.2: Parameters for initialization in the second scenario.

Initial conditions	$(0, 0, 10°)$	Min. turning radius	200
Final conditions	$(2000, 2000, 70°)$	Number of obstacles	9
Degree and number of curve segments	$(5, 5)$	Min. and Max. radius of obstacles	$(100, 1000)$

In this scenario the obstacles in the middle have a radius of 100 meters, while the obstacles centered in the corner of the figure have a radius of 1000 meters. The following figures shows the result obtained for this simulation scenario:

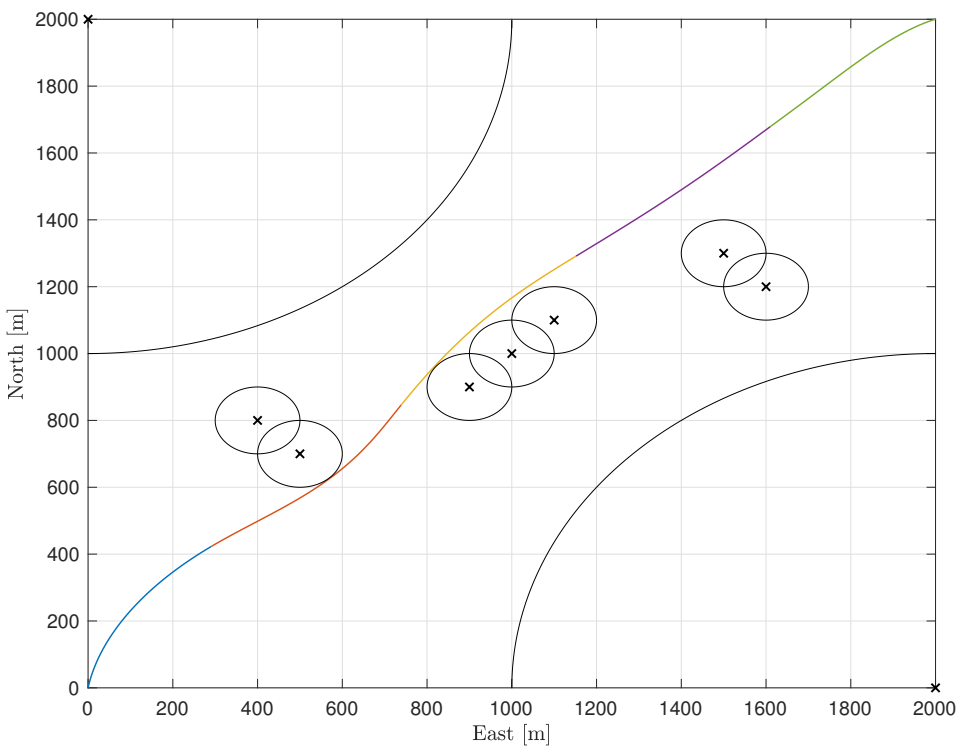

Figure 6.4: Graphical representation of the generated path and obstacles in the second scenario.

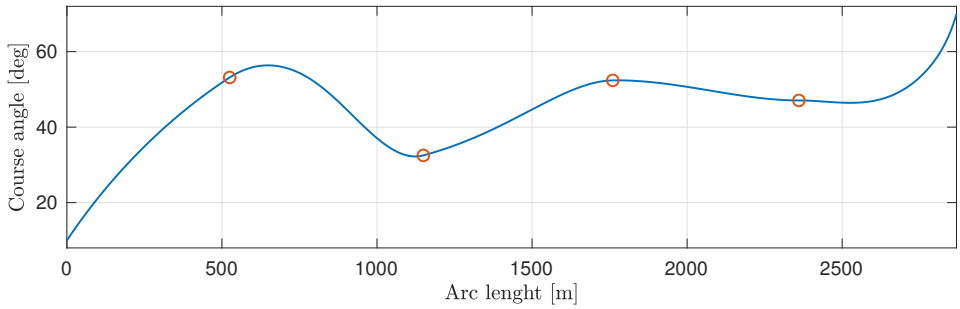

Figure 6.5: Graphical representation of the calculated course angle along the path for the second scenario.

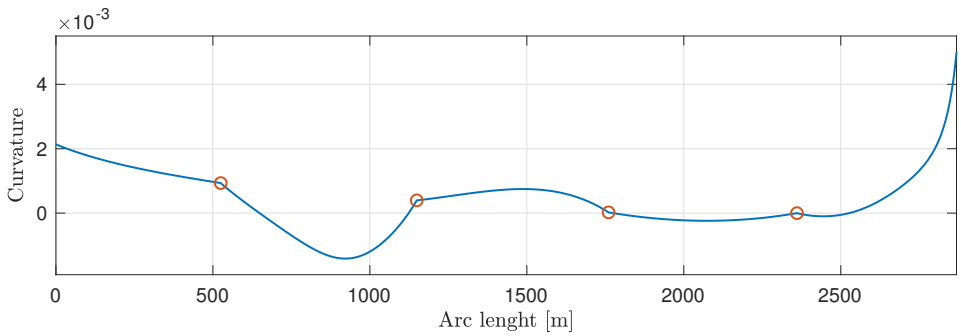

Figure 6.6: Graphical representation of the calculated curvature along the path for the second scenario.

The results shown in this scenario has been obtained through two iterations. Through inspection of Figure 6.5, it can be confirmed that the path has one major course change, and two minor. The major course change is during the first segment, and the first part of the second segment. This change corresponds to a turn, which can be observed in Figure 6.4. The three last segments are seemingly straight, as can be confirmed by inspection of the curvature given in Figure 6.6. The curvature reaches its highest value in the endpoint of the path, for which it is equal to the bound set by the curvature constraint.

6.4 Third simulation scenario

The environment in this scenario has been construed at random and contains a large number of obstacles. The intention of this scenario is to show the efficiency of the proposed algorithm, and the versatility the Bézier curve has to offer in path planning. The environment could be stated to be highly congested, and does not closely resemble a realistic environment. The path is constructed with parametric continuity.

The problem specification is summarized in the table below:

Table 6.3: Parameters for initialization in the third scenario.

Initial conditions	$(0, 0, 45°)$	Min. turning radius	75
Final conditions	$(2000, 2600, 45°)$	Number of obstacles	100
Degree and number of curve segments	$(5, 8)$	Min. and Max. radius of obstacles	$(50, 80)$

The following figures shows the result obtained for this simulation scenario:

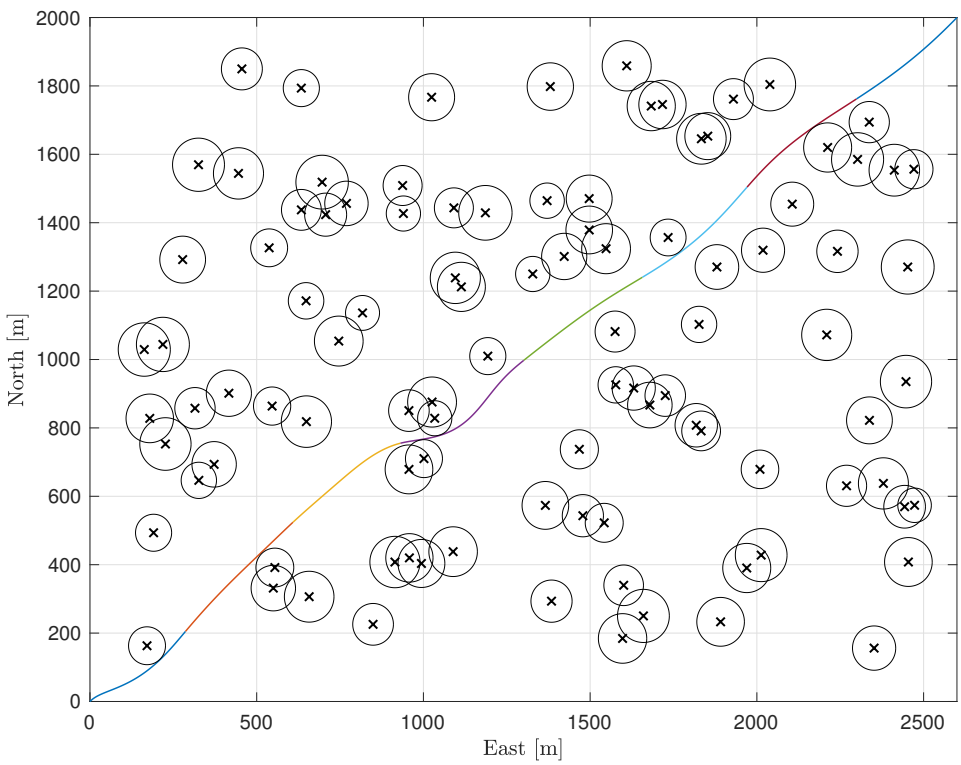

Figure 6.7: Graphical representation of the generated path and obstacles in the third scenario.

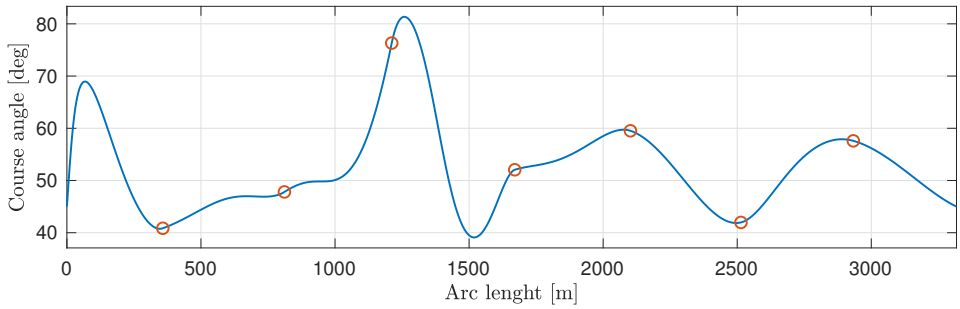

Figure 6.8: Graphical representation of the calculated course angle along the path for the third scenario.

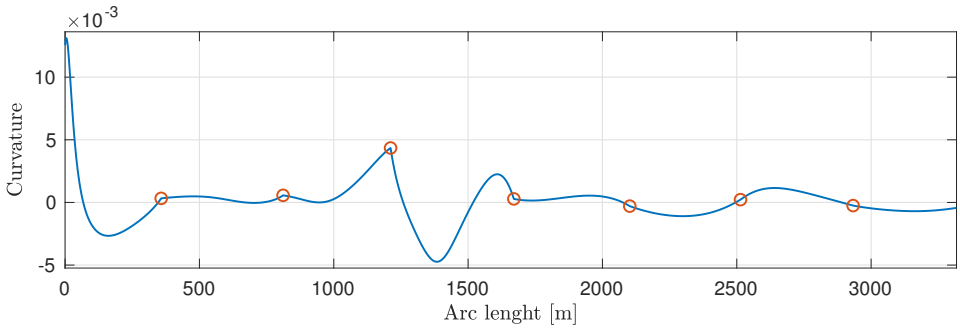

Figure 6.9: Graphical representation of the calculated curvature along the path for the third scenario.

The results obtained in this scenario has been found through three iterations. The path in this scenario have two major course changes, one during the first segment, and the second during the third and fourth segment, see Figure 6.8. The second major course change corresponds to the part of the path, going through a cluster of obstacles, as can bee seen in Figure 6.7. The path never becomes completely straight in this scenario, and have some minor course changes all along the path. These changes corresponds to the path having to circumvent some obstacles, as can be verified in Figure 6.7 and Figure 6.8. The curvature depicted in Figure 6.9, verifies that the path is not completely straight, it does however show that it comes close. This figure also shows that the curvature has its largest value in the initial point, where it is almost of the same magnitude as the bound set by the constraint.

6.5 Fourth simulation scenario

This scenario is included with the intention of highlighting some of the versatility and capabilities of the implemented algorithm. To that end two simulations are shown, both with the same problem description, however different in the choice of continuity constraints.

The problem specification is summarized in the table below:

Table 6.4: Parameters for initialization in the fourth scenario.

Initial conditions	$(0, 0, 55°)$	Min. turning radius	150
Final conditions	$(1500, 1200, 20°)$	Number of obstacles	25
Degree and number of curve segments	$(-, -)$	Min. and Max. radius of obstacles	$(50, 80)$

Geometric continuity:
The path in this scenario is generated by the use of three, fifth order curve segments. The following figures shows the result obtained for this simulation scenario:

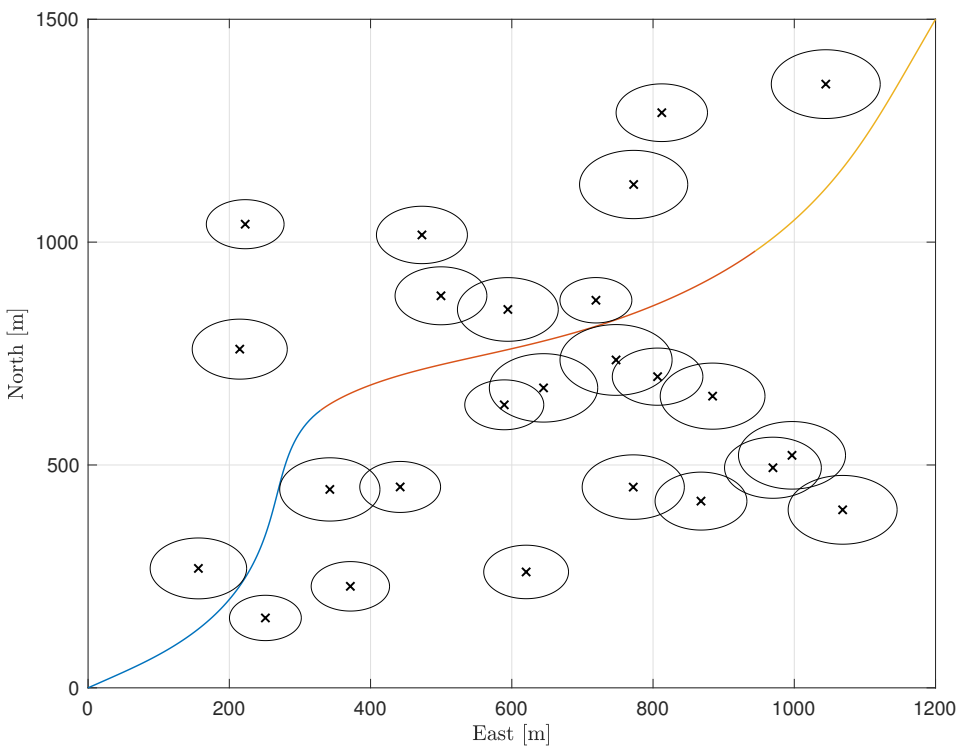

Figure 6.10: Graphical representation of the generated path and obstacles in the fourth scenario (1).

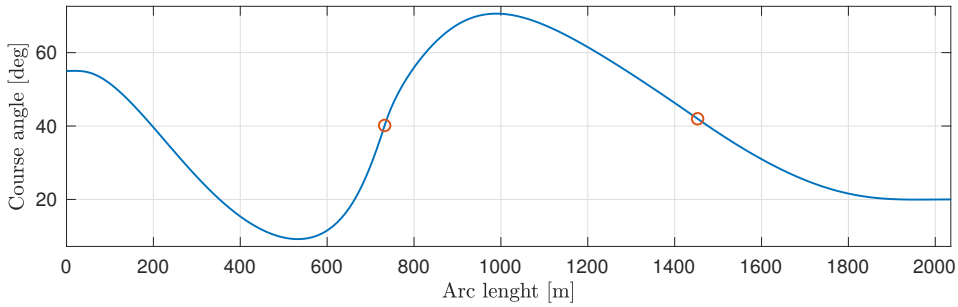

Figure 6.11: Graphical representation of the calculated course angle along the path for the fourth scenario (1).

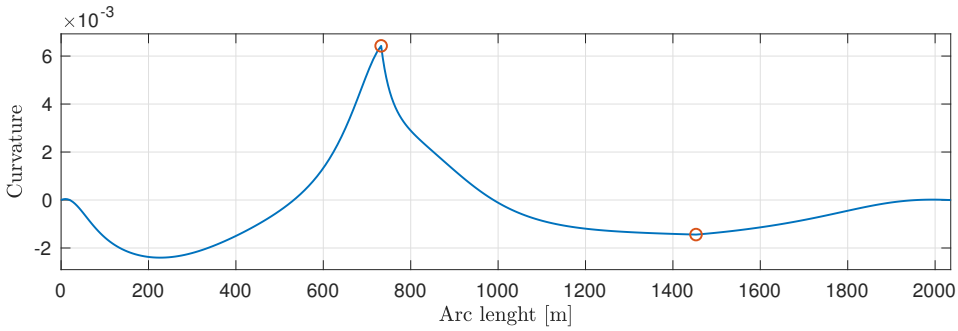

Figure 6.12: Graphical representation of the calculated curvature along the path for the fourth scenario (1).

The results obtained for parametric continuity will follow after this discussion. The results for both cases has been found through three iterations. The paths generated in these two simulations are quite different, as can be observed through inspect of Figure 6.16. The main difference between these paths, are how they pass through the first cluster of obstacles. The path obtained with the geometric continuity constraints progress more towards north, while the parametric path progress more towards east. The length of the paths are approximately equal, with only two meters in difference, in favor of the parametric path. The objective value is however drastically different, with the geometric path having an objective value of approximately 50 % of the parametric path. The course angle depicted in 6.11 and 6.14, also shows a lot of differences. The geometric path changes course more slowly and have less deflection points then the parametric path. The curvature of the geometric path also has less changes then the parametric path. Both of the paths do however have a maximum curvature close to the bound set by the constraint.

Parametric continuity:
The path in this scenario is generated by the use of four, sixth order curve segments. The following figures shows the result obtained for this simulation scenario:

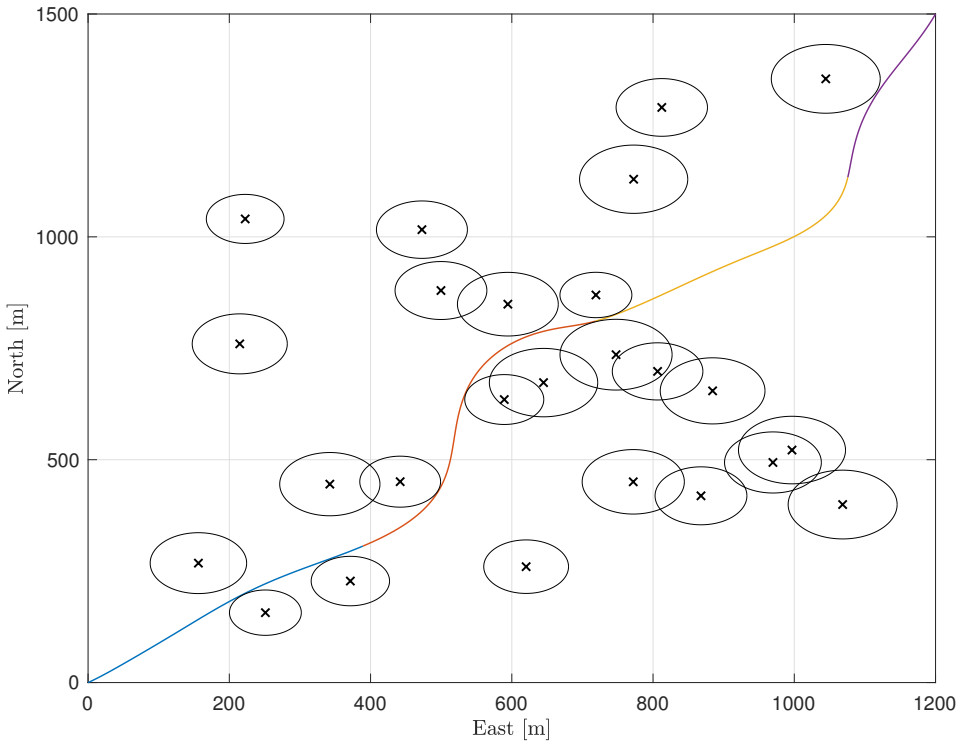

Figure 6.13: Graphical representation of the generated path and obstacles in the fourth scenario (2).

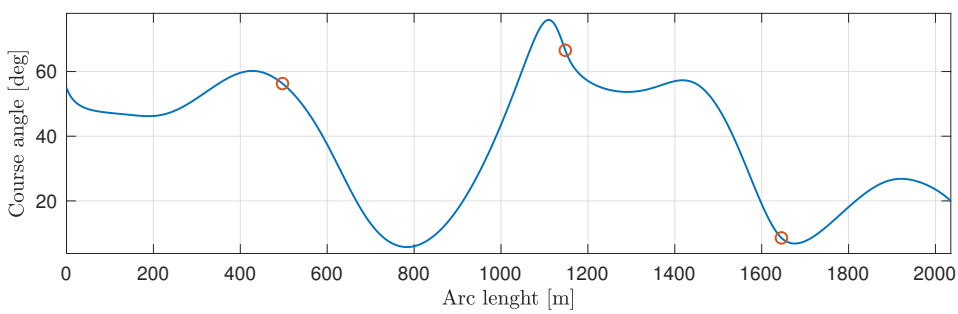

Figure 6.14: Graphical representation of the calculated course angle along the path for the fourth scenario (2).

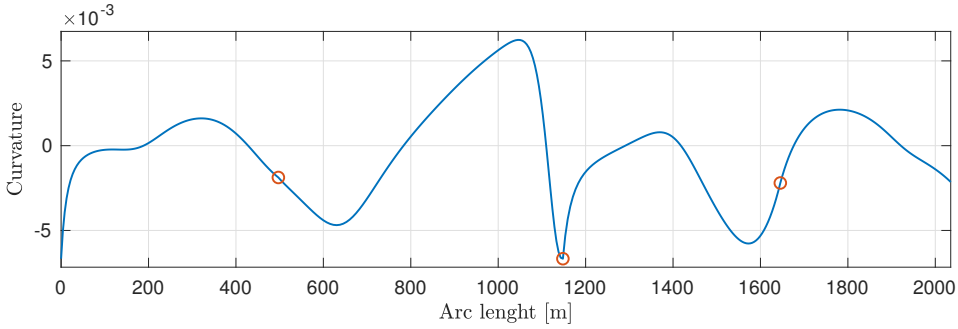

Figure 6.15: Graphical representation of the calculated curvature along the path for the fourth scenario (2).

The following figure shows the geometric and parametric paths, and can be used for comparison between the two.

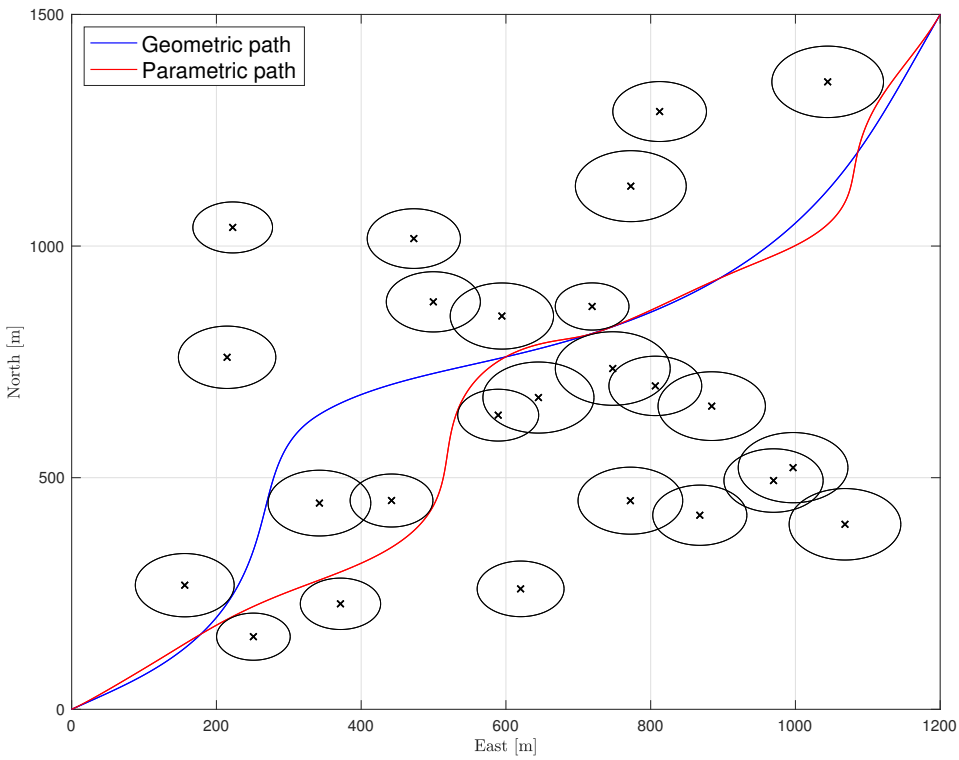

Figure 6.16: Graphical representation of the geometric and parametric paths in the fourth scenario.

Chapter 7

Discussion

The results presented highlight some of the capabilities and effectiveness of the proposed path generation algorithm. Showing that it is possible to combine Bézier curves, differential flatness and optimization in path generation, thus allowing the construction of paths incorporating the dynamics of the vehicle. The results also show that the current implementation allows the generation of paths for a large number of obstacles, this is especially apparent in Figure 6.7. The curvature and course angle are continuous for all scenarios, further implying that the generated paths should be easy to follow by the vessel. Thus it is reasonable to claim that the results are satisfactory.

The fourth simulation scenario is especially interesting, as it shows the differences between geometric and parametric continuity. It was mentioned that the geometric path had a much lower objective value, as compared to the parametric path. The reason for this is most likely due to how the paths differ in shape, and how they pass through the obstacle field. These differences results in the geometric path having less drastic changes in course and curvature along the path, thus affecting the cost. It should further be mentioned that an attempt were made to obtain some results, in which the parametric path passed by the obstacles in the same manner as the geometric path. This attempt failed, and it is therefore reasonable to assume that it would be impossible to obtain a parametric path passing by the obstacles, in the same manner as the geometric correspondence. That is at least under the condition of the degree and the number of curve segments not being increased any further then what presented. In the end, this scenario truly highlights the added freedoms the geometric continuity constraints entail, and how the parametric continuity constraints put to much emphasis on the specifics of the parameterization. This might rise the question of why parametric continuity constraints are used in this thesis, and not geometric continuity constraints. The answer to this question, is that the implemented program did perform more consistently while using parametric continuity constraints. The reason for this is most likely due to the increase in complexity, which is followed by using non-linear constraints as compared to linear constraints.

To take the discussion on parametric continuity a step further, lets consider the last segment of the path in Figure 6.13. This segment exhibits some weird behaviour in which it turns towards the nearby obstacle, and does not proceed in a straight fashion from the preceding segment. This behaviour might seem somewhat odd, and could be explained through the curvature constraints and the boundary conditions. However, if one considers the geometric path presented in Figure 6.10, this doesn't seem to explain everything. It could therefore be assumed that the locality property and the continuity constraints are the main contributors to this weird behaviour. To further explain this, it is important to note that the shape of a single curve segment is influenced by the placement of all control points, for that segment. A small perturbation in a single control point of one segment will therefore lead to a change of that entire segment. The perturbation might however not inflict major shape changes in all parts of the curve. Combining the locality property with the parametric continuity constraints, does therefore set certain restrictions to the shapes that can be represented. Thus it is possible to assume that the weird behavior is a result of this, and that the shape one observes in Figure 6.13 is optimal under these restrictions. This assertion might also be used to explain some of the behaviour exhibited in the other simulation scenarios as well. An example could be made of the first simulation scenario, in which the curvature of the first segment increases, passing zero, before it is decreased back to zero. This means that the path is turning in one direction before turning in the other direction, while this in reality might be obsolete. It might seem from this discussion that parametric continuity is inferior to geometric continuity, and this might be true for the purpose of this thesis, where the only demand is that the path is continuous in course and curvature. It might however be necessary to use parametric continuity in the future, if for instance speed assignment would require the derivatives of the parameterization to be continous.

The remaining part of this discussion will address some of the issues the current implementation faces, and to that end it is sensible to clarify a couple of things related to the behaviour of the utilized solver, and nonlinear optimization in general. For nonlinear optimization one distinguish between local and global solutions, where the local solution is a minimum in a closed neighbourhood, and the global solution describes the point with the lowest function value among all feasible points. This means that there might exist several local solution, but only one global solution. In most applications it is very difficult to recognize whether a solution is global, or even locate it. The solver utilized is only able to recognize whether a solution is local, and no effort has been put into verifying whether these solutions are global. This is due to the share complexity of the problem presented. It should further be mentioned that the utilized solver also exhibits a sensitivity to the initial guess provided. When running a simulation with the current implementation only two possible outcomes has been encountered. The first is that a solution is found constituting a local minimum, the other is that the solver has converged to an infeasible point. It is related to these two outcomes the main issue of the current implementation lays. During any first simulation for a given problem, there are no guarantee of the solver finding a solution, even though a solution might exist. The current implementation is therefore a bit inconsistent when it comes to obtaining results. The size of a given problem in terms of variables and constraints does affect this inconsistency. That is, for a smaller problem the

possibility of the solver finding a solution is greater then in the case of a larger problem. To make things easier in the following discussion, the term reliability will be used to refer to the current implementations ability to produce feasible solutions consistently, for any given problem specification.

The reliability issue essentially boils down to whether the current implementation is capable of finding a solution, to a given problem on the first attempt. In some cases it do and others not, and it has therefore been necessary to use the approach presented in Chapter 5.4.5, in order to obtain the results. The issue is not that the current implementation is incapable of producing satisfactory results, it is how these results are obtained. There might exist several reasons for why the current implementation behaves the way it does, and only speculations can be made as to why. The most reasonable explanation is that the solver used in the implementation is not the best suited for the problem. The complexity of the problem in terms of nonlinearities in constraints, and the simplifications that has been done, is also going to have an impact. The simplifications would involve how the constraints and the objective has been implemented, and how approximations are utilized. These simplifications have been justified before, and it could be further mentioned that they have been necessary in order to complete the work within the allotted time. Another reasonable explanation to the existence of the reliability issue, is the share size of the problems used in the simulations. The simulations accounts for a large number of obstacles, which could be claimed to far exceed what would be realistic. Thus the existence of the issue might be a result of pushing the simulations to the extreme. Taking this into consideration, might therefore reduced the extent of the issue, however since the current implementation never fully guarantees that a solution is found it does not completely eliminate it.

The utilized solver also exhibits a dependency on the initial guess provided, thus it is reasonable to assume that a poor initial guess is going to affect the reliability. This is at least apparent when utilizing the program, as a change in the initial guess often results in the solver finding a solution. It is especially during the first step of the approach presented in Chapter 5.4.5, the possibility of the solver converging to an infeasible solution is highest. The second step of this approach serves as way of finding a feasible initial guess, and does in almost all cases succeed. The third step use the solution found in the second step as the initial guess, and the simulations performed during this step does most often produce good results. This information should indicate that the initial guess is linked to the reliability, as when provided a feasible initial guess the solver is capable of solving problems it initial weren't capable of. It should further be mentioned that the utilized solver most often searches for a solution in the neighbourhood of the initial guess. Thus it is very unlikely that one would obtain a path drastically different from the path given by the initial guess. This does at least seem to be the case in the way the path passes by the obstacles. All of this suggests that the reliability issue could to some extent be resolved by finding a better way of assigning the initial guess. The issue could also be resolved by providing the solver with several initial guesses. This would require that more simulations are performed, however it could increase the possibility of at least finding one feasible solution.

Another factor which might affect the reliability of the current implementation are the gradients. These gradients are as previously mentioned found through finite difference, and they are essential in the determination of a search direction for the solver. The assumption of these gradients affecting the reliability, is made with a basis in the behaviour of the developed program. Through usage it has been apparent that the chosen finite different step size, has had an impact on the solutions that has been found. The solver seem to give more leeway to larger changes when the step size is chosen large, however it does not produce satisfactory results. A smaller step size would most often yield good results, however only under the condition of the initial guess being feasible. These observations does therefore substation the assumption of the gradients affecting the reliability. This is further affirmed when considering how essential the gradients are to the progress of the solver. Therefore to improve the reliability of the current implementation, a study on the accuracy of the gradients should be performed. This could include derivation of analytic expressions for the gradients, or finding a better way of approximating them.

Chapter 8

Conclusion and further work

The overall purpose of this thesis has been to develop a path generation algorithm for marine vehicles, that utilize Bézier curves in order to describe the path. The thesis has dealt with the mathematical formulation of Bézier curves, as well as the properties that are associated with these curves. Through an analysis of the properties of the Bézier curve and some desirable properties related to path planning, it were established that the fifth order Bézier curve were the most suited for the path generation algorithm. A study on the differential flatness property has also been presented, accompanied with a proof of flatness for a simplified underactuated ship model. The thesis has further described how a path generation algorithm can be formulated within the framework of optimization.

The proposed path generation algorithm is able to construct both G^2 and C^2 continuous paths, while taking both curvature and static obstacle constraints into consideration. Furthermore, it is shown how the differential flatness property can be utilized in order to assign a cost function, minimizing the energy associated with the path, while also taking the dynamics of the vehicle into consideration.

As far as results go, we have seen that Bézier curves can be effectively used in path planning, taking several environmental and physical constraints into consideration. The proposed path generation algorithm has produced satisfactory results, and it is shown capable of generating paths for a large number of obstacles. The implemented program does however not produce results consistently, and the term reliability has been used to address this issue. The reliability issue has been described, and some possible reasons for its existence has been presented. Albeit the persistence of this issue, the results have shown great promise, and the effectiveness of using Bézier curves in path planning is clear. On this basis it can be concluded that the Bézier curve is a worthy candidate for future considerations, offering a vast amount of possibilities and versatility.

8.1 Further work

The investigations performed during the work of this thesis has given rise to several ideas for future improvements and extensions. These ideas involve possible solutions to the reliability issue, and aspects that should be considered in order to improve the overall applicational value of the system. These ideas will be presented in this section.

Current implementation

The reliability issue of the current implementation should be further investigated in future work. Resolving this issue would enable studies on for instance computational time and it would improve the overall performance of the program. This issue could be resolved through a change of optimization solver, it is therefore natural to investigate different types of solvers in the future. The current implementation can also be improved upon while still using the same solver, in which case providing a better initial guess could resolve the issue. An idea could be to use Voroini diagrams in this process. This would allow the construction of a set of waypoints through the obstacle field. Strategically using these waypoints could enable the construction of several initial guesses for the solver, thus increasing the possibility of obtaining a feasible solution. Using Voroini diagrams, the endpoints of the Bézier curves could be chosen equal to the waypoints. If the initial guess also consists of straight line segments, one would ensure that the obstacle constraints would be satisfied for the initial guess.

This thesis has focused on using the differential flatness property of the vehicle. This disallowed the representation of some dynamics in the path generation algorithm. In the future it could be interesting to look into the Liouvillain nature of the surface vessel, and whether this would allow the representation of more dynamics.

Future directions

A major improvement for the applicational value of the proposed path generation algorithm is temporal assignment, or in other words time assignment. This could be achieved by assigning a speed profile to the path, thus allowing a description of the ship in both time and space. This would allow the inclusion of moving obstacles, as long as the paths of the obstacles are known, thus giving the algorithm some basic obstacle avoidance capabilities. Temporal assignment will also allow the study of path planning for multiple vehicles.

Once the reliability issue of the current implementation is resolved, a natural direction for future work would be to investigate the possibilities of online path planning. This would involve studies on the computational time required to solve a given problem, and whether this would yield sufficient safety margins to be used in real world applications. Online path planning would also allow the study of collision avoidance capabilities.

Future work should also include model tests, this would allow one to study how well the generated paths can be followed by a real ship. These tests could be performed with a simulation model or with a model in a laboratory.

Bibliography

Adams, R. A., Essex, C., 2013. Calculus: A Complete Course, 8th Edition. Pearson.

Barsky, B. A., DeRose, T. D., 1984. Geometric continuity of parametric curves. Report, University of California at Berkeley.

Barsky, B. A., DeRose, T. D., 1989. Geometric continuity of parametric curves: three equivalent characterizations. IEEE Computer Graphics and Applications 9 (6), 60–69.

Bartels, R. H., Beatty, J. C., Barsky, B. A., 1986. An Introduction to the Use of Splines in Computer Graphics. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Breivik, M., Fossen, T. I., 2009. Guidance laws for autonomous underwater vehicles. In: Inzartsev, A. (Ed.), Underwater Vehicles. IntechOpen, Rijeka, Ch. 4, pp. 51–76.

Choi, J.-w., Curry, R. E., Elkaim, G. H., 2010. Continuous curvature path generation based on bézier curves for autonomous vehicles. International Journal of Applied Mathematics 40 (2), 91–101.

Farouki, R. T., 2012. The bernstein polynomial basis: A centennial retrospective. Computer Aided Geometric Design 29 (6), 379 – 419.

Fliess, M., Lévine, J., Martin, P., Rouchon, P., 1992. Sur les systèmesnon linéaires différentiellement plats. CR Acad. Sci. Paris, 619–624.

Fliess, M., Lévine, J., Martin, P., Rouchon, P., 1995. Flatness and defect of non-linear systems: introductory theory and examples. International Journal of Control 61 (6), 1327–1361.

Fliess, M., Lévine, J., Martin, P., Rouchon, P., 1999. A lie-backlund approach to equivalence and flatness of nonlinear systems. IEEE Transactions on automatic control 44 (5), 922–937.

Fossen, T. I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons.

Ingersoll, B. T., Ingersoll, J. K., DeFranco, P., Ning, A., 2016. UAV Path-Planning using Bezier Curves and a Receding Horizon Approach. American Institute of Aeronautics and Astronautics, San Francisco, CA, USA.

Joy, K. I., 2000. Bernstein polynomials. Tech. rep., Department of Computer Science, University of California, Davis, USA.

Lekkas, A. M., 2014. Guidance and path-planning systems for autonomous vehicles. Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

Lekkas, A. M., Dahl, A. R., Breivik, M., Fossen, T. I., 2013. Continuous-curvature path generation using fermat's spiral. Modeling, Identification and Control 34 (4), 183–198.

Levien, R., Séquin, C. H., 2009. Interpolating splines: Which is the fairest of them all? Computer-Aided Design and Applications 6 (1), 91–102.

Liu, Z., Zhang, Y., Yu, X., Yuan, C., 2016. Unmanned surface vehicles: An overview of developments and challenges. Annual Reviews in Control 41, 71 – 93.

Mehdi, S. B., Choe, R., Cichella, V., Hovakimyan, N., 2015. Collision avoidance through path replanning using bézier curves. In: AIAA Guidance, Navigation, and Control Conference (GNC) 2015, AIAA 2015–0598.

Nocedal, J., Wright, S. J., 2006. Numerical Optimization. Vol. 2 of Springer Series in Operations Research and Financial Engineering. Springer-Verlag New York Inc.

Pettersen, K., Nijmeijer, H., 1998. Global practical stabilization and tracking for an underactuated ship - a combined averaging and backstepping approach. IFAC Proceedings Volumes 31 (18), 59 – 64, 5th IFAC Conference on System Structure and Control, Nantes, France.

Rigatos, G. G., 2015. Nonlinear Control and Filtering Using Differential Flatness Approaches: Applications to Electromechanical Systems. Vol. 25. Switzerland: Springer.

Sederberg, T. W., 2016. Computer aided geometric design. Computer Aided Geometric Design Course Notes, Department of Computer Science, Brigham Young University.

Sira-Ramírez, H., Agrawal, S., 2004. Differentially Flat Systems. Automation and Control Engineering. Taylor & Francis.

Skjetne, R., Øyvind Smogeli, Fossen, T. I., 2004. Modeling, identification, and adaptive maneuvering of CyberShip II: A complete design with experiments. IFAC Proceedings Volumes 37 (10), 203 – 208, IFAC Conference on Computer Applications in Marine Systems - CAMS 2004, Ancona, Italy, 7-9 July 2004.

Skrjanc, I., Klancar, G., 2007. Cooperative collision avoidance between multiple robots based on bézier curves. In: 29th International Conference on Information Technology Interfaces. pp. 451–456.

SNAME, 1950. The society of naval architects and marine engineers. nomenclature for treating the motion of a submerged body through a fluid. Technical and Research Bulletin No. 1–5.

Van Nieuwstadt, M. J., Murray, R. M., 1998. Real-time trajectory generation for differentially flat systems. International Journal of Robust and Nonlinear Control 8 (11), 995–1020.

Wu, J., Snášel, V., 2014. A bézier curve-based approach for path planning in robot soccer. Innovations in Bio-inspired Computing and Applications. Advances in Intelligent Systems and Computing. Springer, Cham 237 (4), 105–113.

Appendix

A: Conference paper

The current version of the submitted conference paper can be viewed on the following pages. The general appearance of this paper has been changed to fit the format of this thesis.

Path Planning for Marine Vehicles using Bézier Curves

Vahid Hassani [*,**,1] **Simen V. Lande** [*]

[*] *Centre for autonomous marine operations and systems (AMOS) and Dept. of Marine Technology, Norwegian Univ. of Science and Technology, Trondheim, Norway.*
[**] *SINTEF Ocean, formerly Known as Norwegian Marine Technology Research Institute (MARINTEK), Trondheim, Norway.*

Abstract: Over the past few years maritime sector has witnessed an increasing interest in use of autonomous ships and in particular Autonomous Surface Vehicles (ASV) in complex applications with high associated risks. There is an uprising interest in the development of advanced path planning algorithms for marine vehicles in congested waterways. Availability of an efficient path planning technique that considers the dynamic capabilities of the vehicle is of paramount importance in the implementation of these algorithms. This article reports an early work which aims to contribute to the development of a new generation of path planning that incorporates in its formulation the dynamics of the vehicles and extra data made available by on board sensors about obstacles and other vehicles in vicinity. To this end, Bézier Curves are exploited as the basis for generating a rich set of paths. Then, differential flatness property of the vehicle is used to assign a cost function to each path that reflects the dynamic capabilities of the vehicle on that path. The efficacy of the proposed algorithm is shown by help of numerical simulations.

Keywords: Bézier curve, path planning, differential flatness

1. INTRODUCTION

One of the earliest path planning algorithms goes back to the early 1950s, where Claude Shannon and his wife, Betty Shannon, built a three wheel magnetic mouse that could find its path through an electro-mechanical maze (MIT Museum, 1952). Theseus maze was a visual display of path planing in dial telephone systems. It showed how information would travel to find the right target telephone to ring when a phone call is made. The problem of connecting two points on the map in presence of obstacles and forbidden zones found many industrial applications. Different techniques were developed to address the path planning problem. The literature on path planning is vast and interested reader is referred to (Laumond, 1998; McLain and Beard, 2000; LaValle, 2006; Kaminer et al., 2006; Dadkhah and Mettler, 2012; Bhushan Mahajan, 2013; Lekkas, 2014) and references therein. Furthermore, assessing the efficacy of path planning algorithms for different applications is a challenging task and out of the scope of this article; see (Dadkhah and Mettler, 2012; Lekkas, 2014) for some guideline on evaluating different path planning algorithms.

(Hausler et al., 2009) gives an introductory application example where a group of autonomous marine vehicles, spread at arbitrary positions and headings, are to perform a cooperative mission at sea that requires adopting a predefined geometrical formation pattern. The vehicles should sail from their initial position and arrive at the final formation pattern at the same time. They call this as "Go-To-Formation" maneuver which, due to existence of obstacles, restricted areas, and required safety distance from other vehicles, needs an advanced path planning algorithm. The different challenges that should be addresses in course of solving Go-To-Formation problem are listed in (Hausler et al., 2009) and later in (Häusler et al., 2010).

Path planing for marine vehicles inherits an increasingly complexity and challenging requirements. Development of autonomous ships and increasing applications for multiple vehicle coordination have created a widespread interest in the development of advanced path planning algorithms for marine vehicles in congested waterways LaValle (2006); Hausler et al. (2009); Häusler et al. (2009, 2010).

Hausler et al. (2009); Ghabcheloo et al. (2009), borrowing the tools introduced by Yakimenko (2000); Kaminer et al. (2006), used a group of 5th order polynomial paths as basis for their path generation algorithm. The coefficient of the polynomials were computed such that the boundary conditions such as initial and final position and heading were met. Their methodology generates paths that completely govern spatial profile of the vehicles. A second temporal problem is solved to address the de-confliction in time to reduce the risk of collision between vehicles and speed assignment for simultaneous arrival of all the vehicles to their final formation pattern.

Motivated by the above considerations, this article reports results of an early work which aims to contribute to the

⋆ This work was supported by Centre for autonomous marine operations and systems (AMOS); the Norwegian Research Council is acknowledged as the main sponsor of AMOS.
[1] Corresponding author, (e-mail: Vahid.Hassani@ntnu.no).

development of a new generation of path planning that incorporates in its formulation the dynamics of the vehicles and extra data made available by on board sensors about obstacles and other vehicles in vicinity. In this paper, Bézier Curves are used as the basis for generating a rich set of paths that determines spatial and temporal profile of the vehicles. Using differential flatness property of the vehicle, we are able to reconstruct all the states of the vehicles during the maneuver. The calculated states are then used to assign a cost function to each path that reflects the dynamic capabilities of the vehicle on that path.

The rest of the article is organized as follows. Section 2 presents a brief introduction to Bézier curves. Section 3 describes the key idea behind the proposed path generation technique. It also provides a summary of differential flatness theory and studies how one can assign a cost to each path such that it reflects the dynamic behaviour of the vehicle. In section 4, a short description of the optimization algorithm is presented. Numerical simulation results of the proposed technique are presented in Section 5. Conclusions and suggestions for future research are summarized in Section 6.

2. BÉZIER CURVE

The mathematical basis for the Bézier curve are the Bernstein polynomials, named after the Russian mathematician Sergei Natanovich Bernstein Farin (2014). In 1912 the Bernstein polynomials were first introduced and published as a means to constructively prove the Weierstrass theorem. In other words, as the ability of polynomials to approximate any continuous function, to any desired accuracy over a given interval. The slow convergence rate and the technological challenges in the construction of the polynomials at the time of publication, led to the Bernstein polynomial basis being seldom used for several decades to come. Around the 1960s, independently, two French automobile engineers of different companies, started searching for ways of representing complex shapes, such as automobile bodies using digital computers. The motivation for finding a new way to represent free-form shapes at the time, was due to the expensive process of sculpting such shapes, which was done using clay. The first engineer concerned with this matter was Paul de Faget de Casteljau working for Citroën, who did his research in 1959. His findings lead to what is known as de Casteljau's algorithm, a numerically stable method to evaluate Bézier curves. De Casteljau's work were only recorded in Citroën's internal documents, and remained unknown to the rest of the world for a long time. His findings are however today, a great tool for handling Bézier curves Farin (2014). The person who lends his name to the Bézier curves, and is principally responsible for making the curves so well known, is the engineer Pierre Ètienne Bézier. Bézier worked at Renault, and published his ideas extensively during the 1960s and 1970s. Both Bézier's and de Casteljau's original formulations did not explicitly invoke the Bernstein basis, however the key features are unmistakably linked to it and today the Bernstein basis is a key part in the formulation Farouki (2012).

A Bézier curve is defined by a set of control points $\boldsymbol{P}_i$ $(i = 0 \dots n)$ for which n denotes the degree of the curve. The number of control points for a curve of degree n is $n+1$, and the first and last control points will always be the end points of the curve. The intermediate points does not necessarily lay on the curve itself. The Bézier curve can be express on a general form as

$$\boldsymbol{P}(t) = \sum_{i=0}^{n} B_i^n(t)\boldsymbol{P}_i \quad t \in [0,1] \tag{1}$$

here t defines a normalized time variable and $B_i^n(t)$ denotes the blending functions of the Bézier curve, which are Bernstein polynomials defined as

$$B_i^n = \binom{n}{i}(1-t)^{n-i}t^i, \quad i = 0, 1, 2..., n \tag{2}$$

2.1 Derivatives

The derivative of any Bézier curve of degree n is a Bézier curve of degree $n-1$. As the control points are constant and independent of the curve parameter t, the derivative is found by computing the derivative of the Bernstein polynomials. The first derivative for the Bernstein polynomials given by Eq.(2) are

$$\dot{B}_i^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)). \tag{3}$$

The derivative of the Bézier curve then takes the following form

$$\dot{\boldsymbol{P}}(t) = n\sum_{i=0}^{n-1} B_i^{n-1}(t)(\boldsymbol{P}_{i+1} - \boldsymbol{P}_i) \quad t \in [0,1]. \tag{4}$$

To further simplify this expression we can define the control points of the first derivative as $\boldsymbol{Q}_i = \boldsymbol{P}_{i+1} - \boldsymbol{P}_i$, the expression then takes the following form

$$\dot{\boldsymbol{P}}(t) = n\sum_{i=0}^{n-1} B_i^{n-1}(t)\boldsymbol{Q}_i \quad t \in [0,1]. \tag{5}$$

Higher order derivatives can be found by repeated use of the relation described in Eq.(3) and Eq.(5).

2.2 Curvature

The curvature of a Bézier curve, given by $\boldsymbol{P}(t) = (x(t), y(t))$, can be expressed in the following form

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{\frac{3}{2}}}. \tag{6}$$

This expression is known as the signed curvature as it takes both positive and negative values. The sign of the curvature will indicate the direction in which the unit tangent vector rotates, as a function of the parameter t along the curve.

3. DIFFERENTIAL FLATNESS

In this section, using the description of differential flatness presented in Van Nieuwstadt and Murray (1998), an informal definition of differential flatness will be presented. A system is said to be differentially flat if one can find a set of outputs, equal in number to the number of inputs, such that one can express all states and inputs as functions of these outputs and their derivatives. This can be formulated mathematically for a nonlinear system, as follows. Consider a nonlinear system

$$\dot{x} = f(x,u) \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m \tag{7}$$
$$y = h(x) \qquad y \in \mathbb{R}^m, \tag{8}$$

where x denotes the state vector, u denotes the control input vector and y denotes the tracking output vector.

Such a system is said to be differentially flat if there exist a vector $z \in \mathbb{R}^m$, known as the flat output, of the form

$$z = \zeta(x, u, \dot{u}, ..., u^{(r)}), \tag{9}$$

such that

$$x = \phi(y, \dot{y}, ..., y^{(q)}) \tag{10}$$
$$u = \alpha(y, \dot{y}, ..., y^{(q)}), \tag{11}$$

where ζ, ϕ and α are smooth functions.

3.1 Model of Surface Vessel

The mathematical model of the surface vessel motion is described by the kinematics and the dynamics as (Fossen (2011))

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{R}(\psi)\boldsymbol{\nu} \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}\boldsymbol{\nu} &= \boldsymbol{\tau}, \end{aligned} \tag{12}$$

where $\boldsymbol{\eta} = [x, y, \psi]^T$ denotes the position and orientation in the earth fixed coordinates, $\boldsymbol{\nu} = [u, v, r]^T$ denotes the generalized velocity given in the body-fixed frame and $\boldsymbol{\tau} = [\tau_1, 0, \tau_3]$ represents the control forces. Further, $\boldsymbol{R}(\psi)$ is the rotation matrix, $\boldsymbol{M}$ is constant positive-definite matrix representing the inertia of the vessel, and $\boldsymbol{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix. The term $\boldsymbol{D}$ represents the linear damping matrix. Specifically, these matrices are given as

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix}, \tag{14}$$

$$\boldsymbol{M} = \mathrm{diag}\{m_{11}, m_{22}, m_{33}\}, \tag{15}$$
$$\boldsymbol{D} = \mathrm{diag}\{d_{11}, d_{22}, d_{33}\}. \tag{16}$$

This paper will consider a simplified version of the underactuated ship model, by enforcing the following simplifications

$$m_{11} = m_{22}, \quad \beta_1 = \frac{d_{11}}{m_{11}}, \quad \beta_2 = \frac{d_{22}}{m_{22}},$$
$$\beta_3 = \frac{d_{33}}{m_{33}}, \quad \tau_u = \frac{\tau_1}{m_{11}}, \quad \tau_r = \frac{\tau_3}{m_{33}}.$$

Rearranging the vehicle dynamics in (12), the state space representation of the underactuated surface vessel follows the following form

$$\dot{x} = u\cos(\psi) - v\sin(\psi) \tag{17}$$
$$\dot{y} = u\sin(\psi) + v\cos(\psi) \tag{18}$$
$$\dot{\psi} = r \tag{19}$$
$$\dot{u} = vr - \beta_1 u + \tau_u \tag{20}$$
$$\dot{v} = -ur - \beta_2 v \tag{21}$$
$$\dot{r} = -\beta_3 r + \tau_r. \tag{22}$$

In what follows, we show that the model described above is differentially flat. we furthermore, calculate the flat outputs of the system.

Choosing the flat outputs for the system model as the coordinates of the vessel in the North-East plane, we show that all the states can be found using the selected flat outputs.

$$\boldsymbol{z} = [z_1,\ z_2] = [x,\ y]. \tag{23}$$

In order to prove flatness for the system, we will first express the derivatives of Eq. (17) and Eq. (18) as

$$\ddot{x} = (\dot{u} - v\dot{\psi})\cos(\psi) - (\dot{v} + u\dot{\psi})\sin(\psi) \tag{24}$$
$$\ddot{y} = (\dot{v} + u\dot{\psi})\cos(\psi) + (\dot{u} - v\dot{\psi})\sin(\psi). \tag{25}$$

Furthermore, by the use of Eq. (20) and Eq. (21), we can prove that the following holds

$$\ddot{x} + \beta_2\dot{x} = (\beta_u u + \tau_u)\cos(\psi) \tag{26}$$
$$\ddot{y} + \beta_2\dot{y} = (\beta_u u + \tau_u)\sin(\psi), \tag{27}$$

where $\beta_u := \beta_2 - \beta_1$. By using these two expressions, we obtain the following relation

$$\psi = \tan^{-1}\left(\frac{\ddot{y} + \beta_2\dot{y}}{\ddot{x} + \beta_2\dot{x}}\right). \tag{28}$$

Thus, Eq. (28) proves that ψ can be written as a function of the flat output and its derivatives.

From Eq. (17) and Eq. (18), we obtain

$$u = \dot{x}\cos(\psi) + \dot{y}\sin(\psi) \tag{29}$$
$$v = \dot{y}\cos(\psi) - \dot{x}\sin(\psi). \tag{30}$$

Using the above equations, and Eq. (28) we obtain

$$u = \frac{\dot{x}(\ddot{x} + \beta_2\dot{x}) + \dot{y}(\ddot{y} + \beta_2\dot{y})}{\sqrt{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}} \tag{31}$$

and

$$v = \frac{\dot{y}\ddot{x} - \dot{x}\ddot{y}}{\sqrt{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}}. \tag{32}$$

Using Eq. (22) and Eq. (28) it can be shown that the following holds

$$r = \frac{(y^{(3)} + \beta_2\ddot{y})(\ddot{x} + \beta_2\dot{x}) - (x^{(3)} + \beta_2\ddot{x})(\ddot{y} + \beta_2\dot{y})}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}. \tag{33}$$

Thus proving that all the states can be written as functions of the flat output. The task of proving that the control inputs can be written as functions of the flat output becomes trivial, as they can be expressed as functions of the states and the derivatives of the states. Through the use of Eq. (20) and Eq. (22), and the expressions for the states we obtain

$$\tau_u = \frac{(\ddot{x} + \beta_1\dot{x})(\ddot{x} + \beta_2\dot{x}) + (\ddot{y} + \beta_1\dot{y})(\ddot{y} + \beta_2\dot{y})}{\sqrt{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2}} \tag{34}$$

and

$$\tau_r = \dot{r} + \beta_3 r \tag{35}$$

where r is given by Eq.(33) and $\dot{r}$ is given as

$$\begin{aligned}\dot{r} = &\frac{(y^{(4)} + \beta_2 y^{(3)})(\ddot{x} + \beta_2\dot{x}) - (x^{(4)} + \beta_2 x^{(3)})(\ddot{y} + \beta_2\dot{y})}{(\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2} \\ &- \frac{2\left((y^{(3)} + \beta_2\ddot{y})(\ddot{x} + \beta_2\dot{x}) - (x^{(3)} + \beta_2\ddot{x})(\ddot{y} + \beta_2\dot{y})\right)}{\left((\ddot{x} + \beta_2\dot{x})^2 + (\ddot{y} + \beta_2\dot{y})^2\right)^2} \Big((\ddot{x} \\ &+ \beta_2\dot{x})(x^{(3)} + \beta_2\ddot{x}) + (\ddot{y} + \beta_2\dot{y})(y^{(3)} + \beta_2\ddot{y})\Big)\end{aligned} \tag{36}$$

Before taking the next step in formulating our path planning algorithm, let us take the discussion stage further. Showing the differentially flatness property of the vehicle, allows us by using any Bézier curve and flatness property of the system, assign a cost function to each path using the calculated states of the system along the path. Our formulation at the current stage assumes that there is no side-slip along the path.

4. OPTIMIZATION

In what follows, we formulate our path planning algorithm in an optimization framework. The proposed path planning technique, utilizes optimization in order to generate a feasible path, that accounts for both physical- and workspace constraints. The workspace constraints refers to obstacle and forbidden zones that ship should not sail through. Furthermore, the ship dynamics are accounted for by the use of differential flatness and assigning a cost function to each path based on the computed states of the system along the path.

The path planning program generates a path between two predetermined waypoints, by stitching a set of Bézier curves together such that the heading and curvature along the path remains continuous. Ultimately, this means that the path that is generated is C^2 continuous. Further, we account for workspace constraints, by including a set of static obstacles in the optimization.

In order to successfully generate a reference path for the vehicle we have used 5th-order Bézier curves. This is due to the fact that lower order curves are not able to offer all the properties that we desire such as C^2 continuity.One should note that increasing the degree of the Bézier curves, could also lead to numerical instability (Skrjanc and Klancar (2007)).

The proposed path planing technique uses the control points of the Bézier curves as design variables, and allows one to specify the number of Bézier curves segments m that is to be stitched together in order to generate the path.

4.1 Optimization Constraints

In what follows we briefly describe the set of constraints that are imposed in the optimization problem.

Continuity constraints: In order to obtain continuity in position, heading and curvature the following constraints will be imposed on the path

$$\boldsymbol{P}_{5,i} = \boldsymbol{P}_{0,i+1}, \qquad i \in [1,\, m-1] \tag{37}$$

$$\boldsymbol{P}_{1,i+1} + \boldsymbol{P}_{4,i} = 2\boldsymbol{P}_{5,i}, \qquad i \in [1,\, m-1] \tag{38}$$

$$\boldsymbol{P}_{2,i+1} - 2\boldsymbol{P}_{1,i+1} + = \boldsymbol{P}_{3,i} - 2\boldsymbol{P}_{4,i}, \; i \in [1,\, m-1] \tag{39}$$

where the numerals denote the control points and i denotes the curve segment number.

Initial and final conditions: The initial and final conditions for the position can be formulated as constraints as follows

$$\boldsymbol{P}_{0,1} = WP_0, \quad \boldsymbol{P}_{5,m} = WP_1, \tag{40}$$

where WP_0 and WP_1 denotes the position of the endpoints in the north-east plane. The constraint for the initial and final conditions for the heading in these endpoints can be formulated as follows

$$l_0 \begin{bmatrix} \sin(\psi_0) \\ \cos(\psi_0) \end{bmatrix} = 5(\boldsymbol{P}_{1,1} - \boldsymbol{P}_{0,1}), \tag{41}$$

$$l_1 \begin{bmatrix} \sin(\psi_1) \\ \cos(\psi_1) \end{bmatrix} = 5(\boldsymbol{P}_{5,m} - \boldsymbol{P}_{4,m}), \tag{42}$$

where ψ_0 and ψ_1 denotes the heading angle in the first waypoint and the second waypoint, respectively. Furthermore, $l_0, l_1 \in \mathbb{R}^+$ are some positive constants, determining the length of the vector in the two waypoints, respectively. Note that these equations will only constrain the direction of the heading vector in the endpoints, and not the magnitude of the vector. These constraint requires the introduction of l_0 and l_1 as design variables.

Turning radius: To ensure that the path has no turns smaller than the minimum turning radius of the ship, we will impose a constraint on the curvature along the path. This could be formulated as

$$|\kappa(t)| < \kappa_{max} = \frac{1}{R_{min}}, \tag{43}$$

where $\kappa(t)$ is the curvature of the path, R_{min} is the minimum turning radius and κ_{max} is the corresponding maximum curvature.

Static obstacles: Environmental constraints will be included in the optimization as static obstacles. Each obstacle will be represented by a circle with radius r and center in (x, y) in the North-East plane. These constraints will take on the following form

$$r \leq \sqrt{(x(t) - x)^2 + (y(t) - y)^2}, \tag{44}$$

where $x(t)$ and $y(t)$ denote the coordinates of the path.

4.2 Objective function:

Using the differential flatness property, we will define an objective function that minimizes the energy associated with each of the path segments. This is formulated as

$$J = \sum_{i=1}^{m} \left[\int_0^1 \dot{u}_i(t) dt \right] \tag{45}$$

where i denotes the curve segment number and $\dot{u}$ is found by differentiating Eq.(31). Since we are using the the flatness property of the system, this objective function will include the ship dynamics.

We would like to highlight that in this article the main contribution is formulating the path generation problem in an optimization framework and not solving the problem itself. Throughout this article, the overall optimization problem is solved using a general nonlinear programming solver in MATLAB®.

5. SIMULATION RESULTS

In what follows we present a series of numerical simulation to evaluate the efficacy of the proposed algorithm.

First scenario:
Fig. 1 shows the results of the generated path for the following problem:

Initial condition $(x_0, y_0, \psi_0) = (0, 0, 40)$; Final condition $(x_1, y_1, \psi_1) = (1800, 2600, 15)$; Nr. Obstacles = 65; Minimum Radius= 50 (m) and Maximum Radius 80 (m); Min turning radius= 100 (m); Nr. Bézier curves= 8.

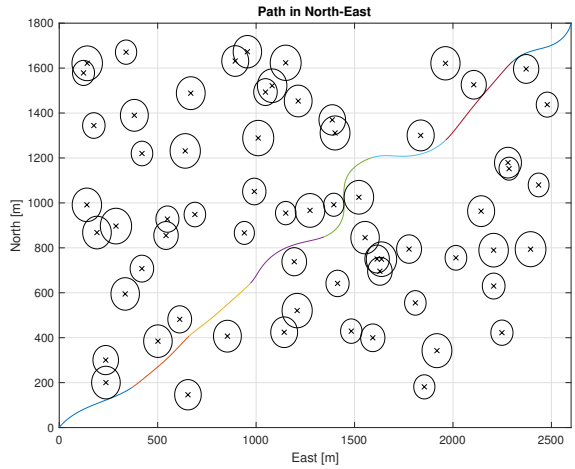

Fig. 1. Graphical representation of the generated path and obstacles in the first scenario

Second scenario:
Fig. 2 shows the results of the generated path for the following problem:

Initial condition $(x_0, y_0, \psi_0) = (0, 0, 35)$; Final condition $(x_1, y_1, \psi_1) = (2000, 2300, 55)$; Nr. Obstacles = 25; Minimum Radius= 70 (m) and Maximum Radius 100 (m); Min turning radius= 100 (m); Nr. Bézier curves= 8.

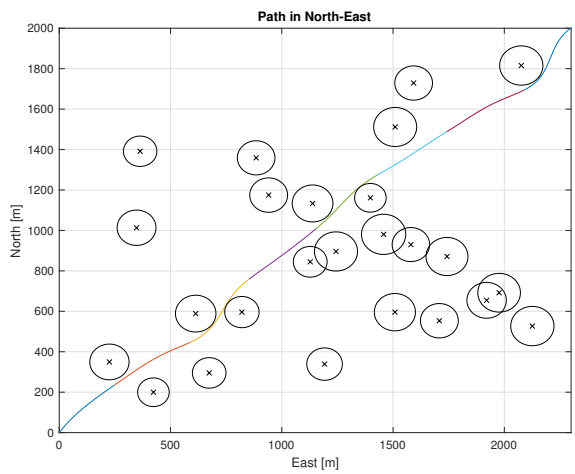

Fig. 2. Graphical representation of the generated path and obstacles in the second scenario

Third scenario:
Fig. 3 shows the results of the generated path for the following problem:

Initial condition $(x_0, y_0, \psi_0) = (0, 0, 90)$; Final condition $(x_1, y_1, \psi_1) = (2000, 2600, 15)$; Nr. Obstacles = 25; Minimum Radius= 70 (m) and Maximum Radius 100 (m); Min turning radius= 100 (m); Nr. Bézier curves= 8.

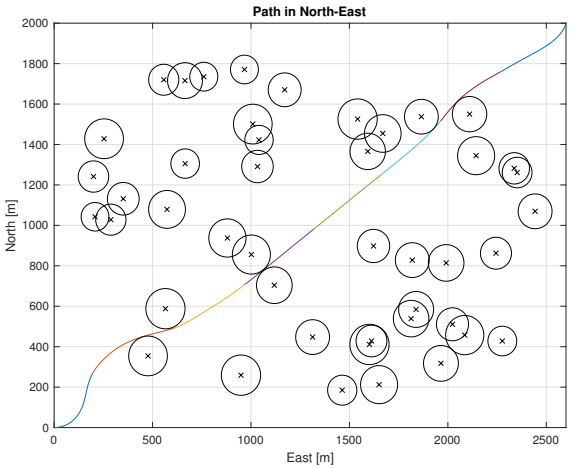

Fig. 3. Graphical representation of the generated path and obstacles in the third scenario

The numerical simulations shows effectiveness of the proposed path planning technique.

6. CONCLUSION

. The problem of path generation for a marine vehicle was addressed in a systematic way. To this end, a class of Bézier curves was used to provide a rich class of potential paths. Using the flatness property of ship, all the states and inputs of the ship along the path was computed from which a cost value was assigned to each candidate path. Finally, an optimization problem was formulated that would give birth to a path that would satisfy all the required properties. The presented work is in its early stage and far from being complete. Future work will include the application of the method developed to multiple vehicles case and development of an efficient optimization technique tailored for the above mentioned problem.

REFERENCES

Bhushan Mahajan, P.M. (2013). Literature review on path planning in dynamic environment. *International Journal of Computer Science and Network*, 2(1).

Dadkhah, N. and Mettler, B. (2012). Survey of motion planning literature in the presence of uncertainty: Considerations for uav guidance. *Journal of Intelligent & Robotic Systems*, 65(1), 233–246. doi:10.1007/s10846-011-9642-9. URL `https://doi.org/10.1007/s10846-011-9642-9`.

Farin, G. (2014). *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier.

Farouki, R.T. (2012). The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6), 379–419.

Fossen, T.I. (2011). Handbook of marine craft hydrodynamics and motion control.

Ghabcheloo, R., Kaminer, I., Aguiar, A.P., and Pascoal, A. (2009). A general framework for multiple vehicle time-coordinated path following control. In *American Control Conference, 2009. ACC'09.*, 3071–3076. IEEE.

Hausler, A.J., Ghabcheloo, R., Kaminer, I., Pascoal, A.M., and Aguiar, A.P. (2009). Path planning for multiple marine vehicles. In *OCEANS 2009-EUROPE*, 1–9. IEEE.

Häusler, A.J., Ghabcheloo, R., Pascoal, A.M., and Aguiar, A.P. (2010). Multiple marine vehicle deconflicted path planning with currents and communication constraints. *IFAC Proceedings Volumes*, 43(16), 491–496.

Häusler, A.J., Ghabcheloo, R., Pascoal, A.M., Aguiar, A.P., Kaminer, I.I., and Dobrokhodov, V.N. (2009). Temporally and spatially deconflicted path planning for multiple autonomous marine vehicles. *IFAC Proceedings Volumes*, 42(18), 376–381.

Kaminer, I., Yakimenko, O., Pascoal, A., and Ghabcheloo, R. (2006). Path generation, path following and coordinated control for timecritical missions of multiple uavs. In *American Control Conference, 2006*, 4906–4913. IEEE.

Laumond, J.P. (ed.) (1998). *Robot motion planning and control*. Springer.

LaValle, S.M. (2006). *Planning algorithms*. Cambridge university press.

Lekkas, A.M. (2014). *Guidance and Path-Planning Systems for Autonomous Vehicles*. Ph.D. thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology.

McLain, T. and Beard, R. (2000). Trajectory planning for coordinated rendezvous of unmanned air vehicles. In *AIAA Guidance, navigation, and control conference and exhibit*, 4369.

MIT Museum (1952). Theseus Maze. `http://museum.mit.edu/150/20`. Accessed: 2018-03-30.

Skrjanc, I. and Klancar, G. (2007). Cooperative collision avoidance between multiple robots based on bézier curves. In *29th International Conference on Information Technology Interfaces*, 451–456. doi: 10.1109/ITI.2007.4283813.

Van Nieuwstadt, M.J. and Murray, R.M. (1998). Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11), 995–1020. doi:10.1002/(SICI)1099-1239(199809)8:11<995::AID-RNC373>3.0.CO;2-W.

Yakimenko, O.A. (2000). Direct method for rapid prototyping of near-optimal aircraft trajectories. *Journal of Guidance, Control, and Dynamics*, 23(5), 865–875.