Yuhan Chen

Temporally Deconflicted Path Planning for Multiple Marine Vehicles

Master's thesis in Marine Technology
Supervisor: Vahid Hassani
June 2019

Master's thesis

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

NTNU
Kunnskap for en bedre verden

Yuhan Chen

Temporally Deconflicted Path Planning for Multiple Marine Vehicles

Master's thesis in Marine Technology
Supervisor: Vahid Hassani
June 2019

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

NTNU
Norwegian University of
Science and Technology

Summary

Over the past few years, growing interests and demands have been witnessed in the development of Automatic Surface Vehicle(ASV) operation, especially in controlling fleet of vehicles to perform complex missions. Basing on the path planning algorithm which generates single path between two points, guiding multiple vehicles to their own ends while avoiding possible inter-vehicle collisions brings more challenges and thus requires more versatility. These kind of algorithms need to first consider goal related influencing factors like environmental constraints and vehicle dynamics. In multiple vessels scenario there would also be some specific requirements like path deconfliction in time or space, final arrival synchronizing or avoiding inter-vehicle communication constraints. Further, measurements and optimized solutions will also be needed, such as energy and length minimized path.

The multiple path planning algorithm is developed basing on single path generation optimization algorithm using Bézier curves. For arbitrary number of vehicles, path planning algorithm capable of leading each vehicle to its assigned target is proposed. For vehicle motion at the intersection points, temporal deconfliction is an important issue to be taken care of. To avoid inter-vehicle collisions, first the motion along the whole path and its corresponding speed need to be defined. After assigning the speed, time profile of each point could be obtained, as well as the distance between two vehicles at a certain time instant.

Further, for the crossing point which would be shown geometrically on the path, temporal deconfliction are formulated as constraints in the framework of optimization problem, as well as vehicle dynamic like velocity and acceleration restrictions, in order to satisfy the physical limitations and avoid problems in path following. Cost function is chosen to penalize the exceeding operation time, thus a feasible solution with shortest time and smooth speeding process could be obtained. And despite of the different length of path, they would finally arrive simultaneously.

Finally, a series of simulation results are presented in the last chapter, showing the feasibility and efficiency of the algorithm in multiple path planning. By presenting simulation scenarios with different conditions like path number, random obstacles and velocity limitations, the versatility of the proposed algorithm could be shown as well. By this end, the conclusion could be made that paths for arbitrary number of vessels are temporally deconflicted with simultaneous arriving time. Some challenges so far still existing would be revealed through discussion, and indicate some crucial points for the future work.

Preface

This thesis was carried out during the spring semester of 2019, at the Norwegian University of Science and Technology(NTNU) as a conclusion for my study in the Department of Marine Technology.

My years at NTNU have been an unforgettable experience, and would be tough if without the support from my family and friends. Marine cybernetic is a new field for me, and it truly brings me challenges and joy. It is never easy to start from somewhere new, but the sense of achievement would make every hard and frustrating time worth it.

Thanks to my supervisor Vahid Hassani, for all the profitable discussions and encouragement throughout the year. And specially thanks to my parents, who would always be there for me with selflessly support. I would never be where I am if it were not their understanding. Also thanks to my friends who always invite me to dinner, since they know how bad I am at cooking. I have learnt a lot during these two years, and hopefully this final work would be useful for someone who works with similar problems.

Yuhan Chen

Trondheim, June 10, 2019

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

While we are paying much attention to the outer space, there is an enormous part of the world which is still hiding in the ocean. The ocean covers 71% of the planet surface and contains the 99% live space. Most of it is unreachable and dangerous to explore, requiring the development of the Autonomous Surface Vehicles(ASV). The information from ocean exploration is vital for nearly all the fields: biology, chemistry, physic, geology, and even archaeology, etc, and its great potential commercial benefits attracts many companies and governments as well. Moreover, ocean is playing a crucial and significant part in military. These all make ocean becoming a focus of researchers and scientists, promoting the development of automatic technology.

Path planning and trajectory generation has long been a crucial problem for automation, and has been studied for years, especially in the field of Unmanned Surface Vehicle(USV). The current trend for developing USV is to perform tasks without human intervention, achieving higher level of autonomy while being able to deal with unpredictable accidents. This gives big challenges on the motion control system of the USVs, of which path generation and planning algorithm is only a part.

The motion control system is usually constructed as guidance, navigation and control(GNC) systems, which usually constitutes by three interconnected subsystems respectively, according to the definition of Fossen (2011) theory. These systems corporate with each other by exchanging data and signals like flows, and their main structure could be illustrated as follows in Figure 1.1. The main tasks for the three subsystem is: **Guidance** continuously collects data from external sensors or computers to perform calculation about model dynamics like velocity, acceleration or heading, etc, and then transmits these data to the human operator. **Navigation** refers to the action that determines the reference position, course and distance to lead the vehicle. Sometimes even the acceleration and velocity are determined at the same time. To do this, a global navigation satellite system combined

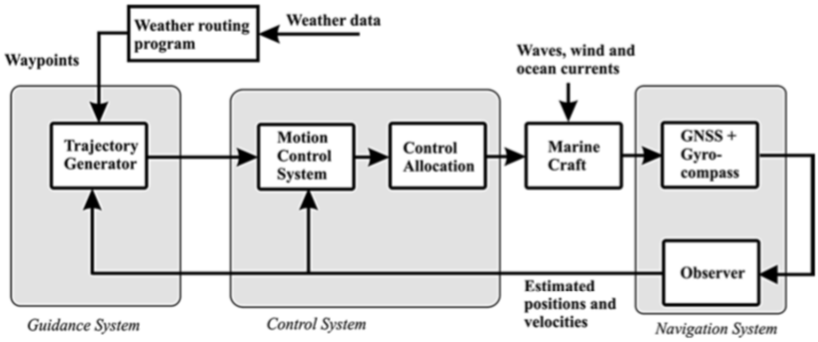


Figure 1.1: GNC signal flow from Fossen (2011)

with motion sensors like accelerometers and gyros is basically needed. **Control**, to be more specific, should be called motion control. This is to determine the desired forces and moments needed by the craft to achieve control objectives, such as minimum energy, path following, trajectory tracking and maneuvering control. These objectives are usually seen in conjunction with the guidance system.

In the path planning algorithm, most of the time plenty of feasible solutions that satisfy all the constraints could be proposed in the end. In order to make the final choice a criteria should be set for the optimization purpose. Usually the whole operating duration of the craft is expected to be as much short as possible, to save energy or increase the efficiency, which however may lead the robots to operating under high speed and steep path following. This would expose the craft to the risk of damaging, such as vibration of machine structures, wear and tear of actuators, and inaccuracy and unstable of the system as well. Such a trade-off process leads to the need of proper path planning and trajectory generation algorithm, paving a smooth path avoiding excessive accelerations and sharp curvature, while minimize the duration cost. For such reasons, path planning algorithm has been proposed, generating a geometric path from the initial point to the destination, and bypassing all the obstacles along the way. The crucial tasks further asks for the cooperation between multiple robots operating in the same scenario and avoid collisions between each other. To deal with this kind of problem from the aspect of temporal deconfliction, time profile for the whole process is required additionally, compared to path planning algorithms. And because of this time coordinate augment, the path could then be related to motion dynamics like velocity, therefore could be analyzed numerically. Besides, it could be resolved in another different way, that all the vehicles need to keep a fixed minimum safe distance between each other, and their routes would never cross or get close in the two dimensional space, despite of the operating time. This method is called spatially deconflicted path planning algorithm. These two ideas for deconfliction would be further introduced in later chapter.

Although lots of attention have been paid in developing more autonomous unmanned marine vehicles, there are still many significant problems remain unsolved. Some of them lie in the accuracy of a good local and global planners, some may be the computational limitations during the process of planning, where more intelligent and efficient planning or better structured algorithm are needed. Greater effort should be kept making to achieve higher level automation for the future development in Unmanned Surface marine vehicles.

1.2 Previous work

In Lande (2018), a new path planning algorithm for marine vehicles using Bézier curves is proposed. It explores the capability of high order Bézier curves in generating smooth and continuous curves in two dimensional space, and take the advantages of its properties and relationships between control points and points on the curves. This algorithm is capable of leading a path that could avoid randomly generated static obstacles, with other vessels nearby and system dynamic limitations taken into considerations as well, in a local 2D space within certain boundaries. It specifically generates a feasible path between two fixed points which is G^2 and C^2 continuous, while constraining the curvature along the path to ensure the result would never exceed the physical limits. Also it includes the vessel dynamic capabilities by using properties of differential flatness, to define a cost to the path. And further, different scenarios of simulations for developed algorithm are implemented in MATLAB, and results and existing challenges are presented and discussed by the end of this thesis work. His work for single path generation algorithm provides a basis for this thesis in path generation part.

1.3 Objective

The overall objective of this thesis is aiming at proposing a multiple vehicles path planning algorithm which avoids inter-vehicle collision and performs simultaneous arrival, for underactuated USVs and ASVs. To achieve this goal, a set of sub-objectives has been set as following to guarantee the main success:

- Perform background literature survey on multiple path planning algorithms, Bézier curves formulations and optimization algorithms.
- Research on the general framework of temporally and spatial deconfliction for multiple vehicle scenario, and its formulation and methods.
- Develop an arbitrary number of path planning algorithm, and mark the intersection points which would geometrically presented in the route map.
- Assign the speed for the whole path, and formulate temporal deconfliction constraints and cost functions within the framework of optimization problems.
- Include the vehicle dynamics like acceleration and velocity, to constrain the velocity and ensure the continuity of the motion.
- Simulate in Matlab and improve the functions basing on the result. Discuss the final result and analyze.

1.4 Contribution

This thesis is mainly contributing a method in multiple marine vehicle path planning with temporal deconfliction. Firstly, related background theories is introduced, including the topic of Bezier curve, vehicle mathematical modeling, temporal and spatial deconfliction. Basing on the path planning algorithm for one path, arbitrary number of paths are generated and speed assignment is proposed to obtain time profile along the path. The crossing points which consequently presented would be taken special care of. Specifically, constraints are formulated within the vicinity of every intersection point in the frame of optimization problem, to get staggered time of passing crossing point and avoid inter-vehicle collisions. Further, final target arriving time for each vessel is constrained, as well as vehicle dynamics like velocity and acceleration allowed in a certain range, in order to obtain smooth speeding process. The total operating time is minimized as well, at the final stage.

Thus the proposed multi-path planning algorithm could lead arbitrary number of vehicles to arrive at assigned destinations respectively, with inter-vehicle collisions avoidance. It would provide solutions in time to form smooth and continuous velocity profiles, with reasonable velocity and acceleration constraints taken into account. Finally, despite the different length of every path, each vehicle would arrive at target simultaneously with same shortest operation time. The behavior is discussed at last by presenting sets of numerical simulation results, and to present the efficiency of the algorithm.

1.5 Structure and Limitations

1.5.1 Outline

Main outline of this thesis is constructed as followings:

Chapter 2: This chapter mainly presents introductions to some background theories and concepts, such as path planning algorithm, Bézier curves and trajectory optimization, along with some mathematical definitions and properties being used, serving as the theoretical support for this thesis. It also provides a mathematical model for an surface vessel.

Chapter 3: This chapter presents some important algorithms related to the topic of multiple vehicle path planning and deconfliction methods, as the main reference literature. It also introduces a range of related work as the result of the literature survey.

Chapter 4: The main concern is to develop the mathematical formulations of the proposed algorithm, basing on the theories introduced in the preceding chapters, and further describe the implementation in the numerical simulation environment MATLAB within the framework of optimization problems.

Chapter 5: This chapter presents the simulation results for four different scenarios, to show the efficiency of the proposed algorithm in arbitrary number of path temporal deconfliction planning problem.

Chapter 6: This chapter discusses the existing problems and challenges basing on the simulation result shown in Chapter 5 and during the process of simulation.

Chapter 7: This chapter presents the conclusions for the overall problem, and discusses some possible improving directions for the future work basing on what this thesis obtained so far.

1.5.2 Limitation

The main limitation in this thesis should be the optimization formulation and a good optimizer. The resulting velocity curves obtained so far still exist oscillation and could not be smoothed perfectly, although the speed changing rate meets the requirement. And the current optimization solver shows a dependency on the value of finite different step size, which is an issue in the obtainment of satisfactory result.

Also the computational time and efficiency is another problem to be considered, especially when dealing with a complicated situation with a large number of multiple vehicles. This would be important when the vehicle needs online modification and path replanning, and thus disallowing the deviation occurs as well.

1.6 Preliminaries

1.6.1 List of Abbreviations

GNC	Guidance, Navigation and Control System
ASV	Autonomous Surface Vehicle
USV	Unmaned Surface Vehicle
2D	Two dimension
3D	Three dimension
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAGD	Computer Aided Geometric Design
SQP	Sequential Quadratic Programming
ECEF	Earth-centered Earth fixed

Chapter 2

Theoretical Background

2.1 Path planning

2.1.1 Motion control system

Motion control typically refers to control the motion of moving objects like position, velocity or accelerations, by motion controllers and actuators. It is a sub-field of automation, and usually carried out in the following structure:

- Motion controllers, to calculate and output the set points or desired motion as control signals to the next stage.
- Amplifiers who receives the control signals from controllers and transform them into the number of forces or torques that actuators needs to produce, in order to reach the desired set points.
- Actuators or motors are what produce thrusts or forces for vehicles to execute desired motion, for example hydraulic pump or engine. For marine vessels, actuators are usually propellers or azimuth thrusters.

Motion control systems could either be in open loops or closed loops. In open loop systems the controller send commends out and never know whether the motion has been carried out successfully in the end, thus is less precise and unguaranteed but easy in implementation. For more precised ones an observing system could be added to measure the output and transform it back to the controller as signal, thus the controller could be aware of the deviation and knows how much should be compensated.

This control theory could further be combined with path planning algorithm, in the cooperation with GNC systems which are introduced in Chapter 1.1 above. The result a path planner outputs is input into the Guidance sub-system, and thus being computed as reference position, velocity and acceleration to guide the vehicle to the destination. Navigation sub-system locates the position of the vehicles in the coordinates and determines the vehicle dynamics by using Global Navigation Satellite System(GNSS) or motion sensors. And motion control finally collects these data and determines the force or torque needed to achieve control objectives, like following the desired path.

2.1.2 Path planning algorithm

Path planning problems is about how to generate one or more feasible routes from initial to the final point, and basically a geometric problem. In the most simplified situation, it is normally working within a static 2D or 3D plane with fixed coordinate system despite the change of time. Here the whole environment within regional boundaries, and static constraints are all known. In a more complex and autonomous situation, one might need to deal with a dynamic environment where the environment is only partially known, with kinematic and dynamic constraints. In comparison to this, the definition of trajectory planning includes assigning a proper time law to the static path, besides performing feasible path planning. The definition of path planning from Carbone and Gomez-Bravo (2015) is straightforward:

Definition 2.1.1. *Path Planning: Find a collision-free motion between an initial (start) and a final configuration (goal) within a specified environment.*

Operation Space

Some general definitions about the operating space should be introduced about the path planning problem, as:

- The configuration space, $C - space$;
- The space of free configurations, $C - free$;
- The obstacle's representation in configuration space, $C - obs$;

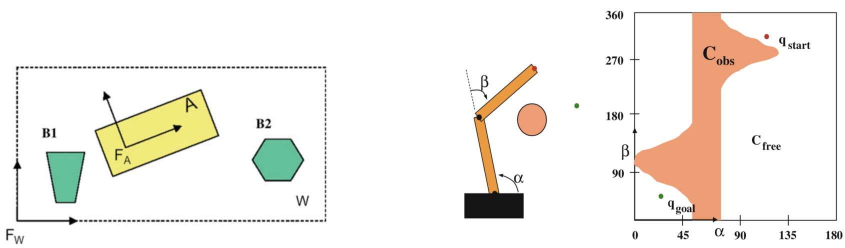
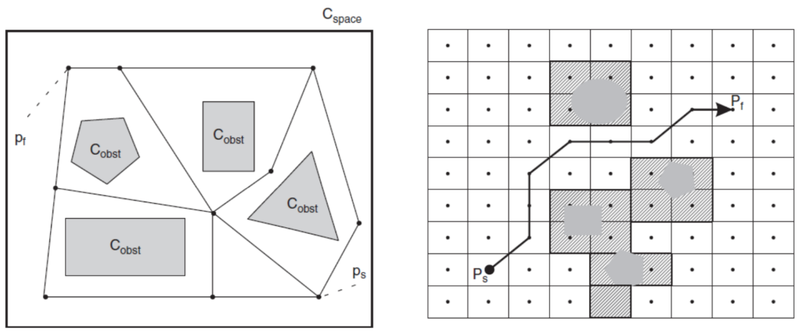


Figure 2.1: Robot with obstacles and in a joint space from Carbone and Gomez-Bravo (2015)

The term configuration refers to the location of the robot or vehicle, and its orientation in the space with respect to the reference frame, which is shown in the left of Figure 2.1, F_w. Thus the configuration space is the set of all possible configurations. For another robot in the right of Figure 2.1, the $C - space$ needs to be presented in a joint space, thus the

$C - obs$ could be described as following, and $C - free$ could be represented as $\{C - space$ - $C - free\}$.

In the process of producing the feasible route, the handling with constraints is a crucial problem. Most of the time the purpose of setting constraints is to ensure the operation safety, or to make sure it is achievable for the vehicle to follow the path. Dynamic and kinematic constraints includes the curvature of the path or turning radius, for example. And ensuring the safety refers to keep a safe distance with other vehicles and obstacles, to avoid crushing. Once all the dynamic and environment constraints are satisfied, the route would be considered as a feasible one, and most of the time the set of feasible solutions could be rich. Under this circumstance, a cost function, based on different objectives would be needed to evaluate each path, such as minimum execution time or length, and minimum energy.



(a) Road Map Method

(b) Cell Decomposition Method

Figure 2.2: Two example methods in path planning techniques in Roald (2015)

Path planning has long been a significant topic when it comes to the field of automation, and has been in development ever since 1970s. Now after such a long period, there are plenty of technology which are capable of generating path and perform path planning. Some based on the graphical techniques (like Figure2.2), exploring the feasible way in dimensional space, and other might based on artificial potential methods. Some alternative approaches also take advantage of probabilistic algorithms, such as random sampling in building the roadmap, and has done excellent work in complex path planning situation. Every methodology has its own potential and field for implementation, and here in this thesis, mathematical modeling and optimization is utilized, and would be presented in following chapters.

2.1.3 Formulation and Parameterization

General Parameterization

Generally in path planning, a path, or route, could be constituted by one or several segments of parametric curves. In mathematics, parametric curve is a set of equations $x =$

$f(t)$ and $y = g(t)$, that trace a curve C when parameter t varies in its domain. In most cases the parameter t is used to symbolize "time" for a curve.

A parametric curve could either be defined separately by using two function $f(t)$ and $g(t)$, representing x-coordinate and y-coordinate respectively with a third parameter t, or could just be described using a scalar parameter ϖ, as following set (Breivik and FossenBreivik and Fossen (2009), 2009):

$$\mathcal{P} := \{\mathbf{p} \in \mathbb{R}^2 \mid \mathbf{p} = \mathbf{p}_p(\varpi) \quad \forall \varpi \in \mathbb{R}\} \tag{2.1}$$

Here the path is considered as a planar path, and all the points on the curve is described in one dimension as $\mathbf{p}_p(\varpi) = [x_p(\varpi), y_p(\varpi)]^T \in \mathbb{R}^2$. The parameter ϖ takes domain in the interval $\varpi \in [\varpi_0, \varpi_1]$.

In Hausler and Ghabcheloo (2009)Häusler et al. (2009), the path is parameterized by using an algebraic polynomial of degree N, such as, $x(\tau)$ is of the form:

$$x(\tau) = \sum_{k=0}^{N} a_{x^k} \tau^k \quad i = 1, 2, 3, \tag{2.2}$$

The number of degree N of the polynomial $x(\tau)$, $y(\tau)$, $z(\tau)$ is determined by the number of boundary conditions needed to be satisfied. The parameter τ could be further be parameterized by time t as $\tau_i(t)$, which gives the potential for assigning a proper time laws that allocate the nominal speed for vehicles along the path.

Piecewise curves

If the path calculation is quite complicated and requires for more versatility, only one segment of parametric curve might not suffice to meet the requirements. Thus piecewise curves with different length of segments connected together are a good choice to express complex shape. Compared with single curves, this might increase more possibilities in extending in space, but also place extra limitations on the connection points, for example continuity in curvature and turning radius. If the path planner consists of allocating a time law, the velocity and accelerations on transition points should be constrained additionally as well. Apart from this, complex shape could also be represented by high order single curve, which may lead to unstable performance and higher level function complexity and computational loads indeed. According to Lekkas (2014)Lekkas (2014), for path that composed by a number of curve segments, every segment could be described as the set:

$$\mathcal{P} := \{\mathbf{p}_i \in \mathbb{R}^2 \mid \mathbf{p}_i = \mathbf{p}_{i,p}(\varpi) \quad \forall \varpi \in \mathcal{I}_i = [\varpi_{i,0}, \varpi_{i,1}] \subset \mathbb{R}\} \tag{2.3}$$

And the path could be expressed as:

$$\mathcal{P}_s = \bigcup_{i=1}^{m} \mathcal{P}_i. \tag{2.4}$$

In this thesis, piecewise curve is also utilized in the path planning algorithm in generating feasible route using Bézier curve, since high order Bézier could be easily numerically unstable. For this reason, it is desirable to join low-degree Bézier curves together in a smooth way, and the least degree of the Bézier curve that can satisfy this requirement is three (Choi, Ji-wung and Curry, 2008Choi et al. (2008)), which is namely cubic Bézier curve.

2.1.4 Global and Local Methods

Methodologies for path planning could be divided by the way of generating geometric path, but would also depends on the knowledge about the surrounding environment, thus global and local methods are defined.

Global planning often means that the information about the surrounding environment are all well known and could be utilized directly in planning process. Since it is aware of the whole map during finding the route, the final optimal decision is also determined at once and would not be changed ever since obtained. The goal could either be a fixed given target or series of points with fixed sequence of going through, and the operating space might be assumed static or dynamic, global planner does not need any information from sensors or observations about current state of the vehicles, since everything has already been prestored in the memory. But the main drawbacks of global planner is thus caused by such big amount of calculation, compared with local planners, in particular. This would lead to a quite long computation time and heavy computational loads, the efficiency of reacting to the world outside is thus comparatively low, if dealing with a dynamic situation outside. However, the result are more likely to be a feasible solution, and further to be the optimal one among all possible ones exist.

Local planners, on the other hand, is focusing more on partial place nearby, or only have limited knowledge towards the operating space. Under this circumstances, it could only choose to be sensitive to the changing outside by applying sensors or monitoring devices to collect real-time data. This makes it suitable in executing tasks in dynamic environment, as it emphasizes more on the reacting efficiency in this way, and would not have strict requirements towards the information processing speed. It works mainly dealing with the current state of the vehicle, finding a locally feasible or optimal solution, and proceeding to the next stage. It has no information about the place the sensor not covered, even the place it just went through since the environment is changing and no memory has left. By making progress step by step, it should finally be able to reach the end, but the success could never be guaranteed, and the solution it obtained is less likely to be the globally optimal one, although with much lower cost.

2.2 Bézier Curve

2.2.1 Background

Bézier curve is a mathematical parametric curve, which is often utilized in generating two dimensional parametric curve in the field of Computer Aided Design(CAD) and Computer Aided Manufacturing(CAM). It could draw precise and smooth line by using vector graphic software, and therefore is very common in the industrial application, especially in designing smooth and curved surface in three dimension. Line segments and control points are the main components of typical Bézier curves, that coordinates of points could be assigned arbitrarily, and line segments are just like retractable rubber bands. Bézier curve has been explored and implemented in many mature image processing software as curve drawing tool, and is of great importance and takes a significant place in Computer Graphics.

A pieccewise parametric curve constituted by Bézier curve is used in this thesis, which is namely Bézier spline. In the following section, some basic introduction about Bézier curve including historical background, mathematical formulations, important function properties and applications would be presented.

2.2.2 History

Bézier Curve

In industrial computer design, there is always a demand in how to represent and produce a smooth curve that must be agreeable to the observer aesthetically and not too mathematically. With the development of Computer Graphic, the technology of representation complex curvilinear has developed rapidly, and great progress in software that defining curved free surface has been made during 1950's to 1960's in the field of automobile and aircraft industries, by mechanical engineers. In solving this problem, French engineer Pierre Bézier, working for a automobile company named Renault, has made a major and remarkable contribution.

At that time in 1950's in industry, producing curves other than basic lines, circles and parabolas was inaccurate and low efficient. Although design engineers could produce great and satisfied drawings in drawing board, transferring it to digital graphics and guide the manufacture in factory were hard and would cause many additional problems like severe graph distortion. In a word, they were lacking of an efficient tool for transformation and communication between drawing and pattern manufacturing, not to mention complex surface shape was needed to be in three dimensional space. The development of hardware was fast which had already enabled computation and procession for complex surface, therefore all was ready except for this software. Pierre Bézier, at this time, made an impressive progress in curvilinear parameterization. He successfully defined surface with polynomials to expressing spatial properties, which is now considered as the beginning of Computer Aided Geometric Design (CAGD). He represented a curve between two given points by formulations, introducing a new concept named ”control point”. Based on the control points, the tangency and coordinates of the points on the curve could all be expressed explicitly. Additionally, the shape of the curve could be change freely by moving the control point, while still keep smooth during the process.

Actually, the original polynomial which gives Bézier's work solid mathematical basis is Bernstein polynomials, which named after a Russian mathematician Sergei Natanovich Bernstein and became well-known to the world on 1912. Briefly explaining it, Bernstein polynomials could be utilized in proving that every continuous function on a certain interval $[a, b]$ could be approximated by polynomials with strong convergence. In another words, that is for any continuous function, there would always exist an alternative expression in the form of summation of several Bernstein polynomials which would converges to the original function along with the degree n approaching infinity. However, the poor technology in constructing accurate polynomials and slow convergence rate at that time limited the development of Bernstein polynomials and made it to be used after publication for decades.

However, Bézier's work is not unique at that time, even not exactly the first that came up with. There was an another excellent automobile engineer working for Citroën just finished a similar work at almost the same time, but only published his result later than Bézier. This engineer's name is Paul De Casteljau, and his work was kept by his company Citroën as internal secret documents for a long time, named as De Casteljau Algorithm. But Bézier published his findings extensively and used it in aiding car body industrial design which makes the curves so well-known, thus his name is linked directly to this kind of curve since than, and being used until nowadays.

De Casteljau Algorithm

Paul De Casteljau was a French physicist working for Citroën in Paris in 1958. In that same year he finished his work on developing the algorithm, a numerically stable method to deal with Bézier curve, which made excellent progress in modeling car body surface. His first theoretical result had been sealed as soon as he complete that, later for decades the rest of the world kept knowing nothing about his impressive ideas. On De Casteljau's mid-sixties, Pierre Bézier published his own finding at Renault and on De Casteljau's mid-seventies, his fundamental work was finally be able to became well-known by public. De Casteljau's central idea was recognized as De Casteljau Algorithm and playing an important role ever since then. As a recursive analysis method, his algorithm is numerically stable in evaluating Bernstein polynomials, as well as in splitting a single Bézier into piecewise curve at an arbitrary parameter value. Even nowadays his algorithm remains a good tool, although slower than other direct method, but still more stable numerically.

2.2.3 General Formulations

For this section some basic formulations, properties and important math operation of Bézier curve is presented, and the main reference for this, especially the mathematical part is from Sederberg (2012), Lande (2018) and Kamermans (2011).

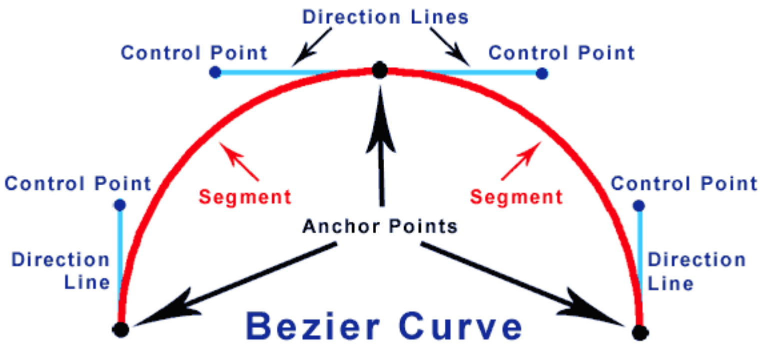


Figure 2.3: Bezier Curves in Novin (2016)

As previously introduced above, an important composition of Bézier curves is control points. Basically, Bézier curves could be considered as interpolation functions, which means they take a set of control points and generate values between these points. This, in consequences, leads an significant properties that pretty useful, which is called *convex hull property*. It guarantees that no point generated by the control points would escape from the outline for the control points, which means all of them would stay in the small region space with certain boundaries. In fact, not every point are contributing to the curve uniformly: there always exists someone more important, and someone less, and the parameter that evaluates this is conventionally called weight. To change the curve it is also never necessary to change the place of control points, changing the weight could also suffice instead. Assume the Bézier curve is defined by a series of control points $\mathbf{P}_0$ to $\mathbf{P}_n$, the general form of Bézier curve function is:

$$\mathbf{p}_p(\varpi) = \sum_{i=0}^{n} B_i^n(\varpi)\mathbf{p}_i, \quad \varpi \in [0,1] \tag{2.5}$$

where the variable ϖ represents time and the term $B_i^n(\varpi)$ is the Bernstein polynomials, defined as Lande (2018):

$$B_i^n = \binom{n}{i}(1-\varpi)^{n-i}\varpi^i, \quad i \in 0, 1, ..., n. \tag{2.6}$$

By adding weight term the Bézier function is then straight forward as Kamermans (2011):

$$Bezier(n,t) = \sum_{i=0}^{n} \underbrace{\binom{n}{i}}_{binomial\ term} \cdot \underbrace{(1-t)^{n-i} \cdot t^i}_{polynomial\ term} \cdot \underbrace{w_i}_{weight} \tag{2.7}$$

And the binomial term is defined as factorial:

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!} \tag{2.8}$$

In fact, it is never complicated to decide what the weight value should be for Bézier curves, because it is just the coordinates one places the control points. Like mentioned above, for a curve with n^{th} order, the control point is $\mathbf{P}_0$ to $\mathbf{P}_n$, which the first one is exactly the starting point, and the last one is the end point of the curve.

Example 2.2.1. *Assume a cubic Bézier curve which starts at (100,100), ends at (400,210) and is control by (200,200) and (300,110).*

By applying the formula (2.7):

$$\begin{cases} x = 100 \cdot (1-t)^3 + 200 \cdot 3 \cdot (1-t)^2 \cdot t + 300 \cdot 3 \cdot (1-t) \cdot t^2 + 400 \cdot t^3 \\ y = 100 \cdot (1-t)^3 + 200 \cdot 3 \cdot (1-t)^2 \cdot t + 110 \cdot 3 \cdot (1-t) \cdot t^2 + 210 \cdot t^3 \end{cases}$$

The x-coordinates and y-coordinates of all points could be obtained, which further gives the example curve shown in Figure 2.4:

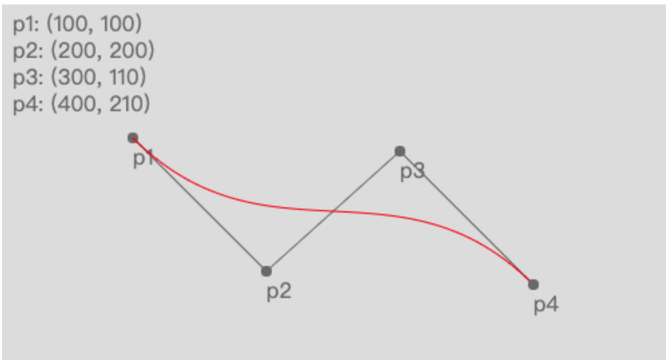


Figure 2.4: Example Bézier Curve

2.2.4 Matrix Form

Bézier formula could also be represented in the matrix form, by expressing it as a polynomial and coefficent matrices. Assume a cubic Bézier that:

$$B(t) = P_1 \cdot (1-t)^3 + P_2 \cdot 3 \cdot (1-t)^2 \cdot t + P_3 \cdot 3 \cdot (1-t) \cdot t^2 + P_4 \cdot t^3 \tag{2.9}$$

First considering the part without control points coordinates, it can be expressed in the following form:

$$[t^3 \;\; t^2 \;\; t \;\; 1] \cdot \begin{bmatrix} -1 \\ 3 \\ -3 \\ 1 \end{bmatrix} + [t^3 \;\; t^2 \;\; t \;\; 1] \cdot \begin{bmatrix} 3 \\ -6 \\ 3 \\ 0 \end{bmatrix} + [t^3 \;\; t^2 \;\; t \;\; 1] \cdot \begin{bmatrix} -3 \\ 3 \\ 0 \\ 0 \end{bmatrix} + [t^3 \;\; t^2 \;\; t \;\; 1] \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.10}$$

Furthermore, the complete matrix form for cubic Bézier curve according to Kamermans (2011) would be:

$$B(t) = [1 \;\; t \;\; t^2 \;\; t^3] \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \end{bmatrix} \tag{2.11}$$

Similarly, quadratic Bézier curve matrix form would be:

$$B(t) = [1 \, t \, t^2] \cdot \begin{bmatrix} 1 & 0 & 0 \\ -2 & 2 & 0 \\ 1 & -2 & 1 \end{bmatrix} \cdot \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{2.12}$$

Matrix form representation is usually helpful in discovering properties that hard to find in polynomials, and sometimes the matrix approach can be quite fast.

2.2.5 De Casteljau's algorithm

De Casteljau's algorithm is pretty useful in splitting Bézier curves into several segments, thus it is also named as subdivision algorithm. On the other hand, it also implies that if a complex shape of Bézier curve is desired, de Casteljau's algorithm could be utilized to connect sets of Bézier curve together as one, since modeling complex shape by using single high order curve can be quite computationally expensive and numerically unstable. This is accomplished by finding all the new control points needed by smaller segments:

$$\mathbf{p}_i^j = (1-\tau)\mathbf{p}_i^{j-1} + \tau \mathbf{p}_{i+1}^{j-1}, \quad j \in \{1, ..., n\}, \quad i \in \{0, ..., n-j\}. \tag{2.13}$$

Based on this two new control points, new Bézier curve can be presented by:

$$\mathbf{p}_{p,1}(t) = \sum_{i=0}^{n} B_i^n(\frac{t}{\tau})\mathbf{p}_i^j, \quad t \in [0, \tau], \tag{2.14}$$

$$\mathbf{p}_{p,2}(t) = \sum_{i=0}^{n} B_i^n(\frac{t-\tau}{1-\tau})\mathbf{p}_i^(n-j), \quad t \in [\tau, 1]. \tag{2.15}$$

This subdivision algorithm is suitable for any degree of curves, especially useful when computing the coordinates and tangent vector of any points.

2.2.6 Lowering and elevating curve order

One interesting property of Bézier curve is that, any n^{th} degree curve could be modeled perfectly by using an higher order curve, if giving it specific control points. Denoting as $\mathbf{p}_i^*$, the new control points can be obtained by (assuming that the start and the end points do not change):

$$\mathbf{p}_i^* = \alpha_i \mathbf{p}_{i-1} + (1-\alpha_i)\mathbf{p}_i, \quad \alpha_i = \frac{i}{n+1} \tag{2.16}$$

By using it recursively, a Bézier curve can be raised to any higher degree desired, or on the other hand, lower. If involving the concept of weight mentioned before, this function can be reformulated as:

$$Bezier(n+1,t) = \sum_{i=0}^{n+1} \underbrace{\binom{n+1}{i}}_{binomial\ term} \cdot \underbrace{(1-t)^{n+1-i} \cdot t^i}_{polynomial\ term} \cdot \underbrace{(\frac{(n+1-i) \cdot w_i + i \cdot w_{i-1}}{n+1})}_{new\ weights},$$

where $w_{i-1} = 0$ when i = 0. However this one is not as perfect as the former function (2.16), since it cannot generate safely back to lower degree curve from n^{th} to $n-1^{th}$ degree. There are possibilities that the resulting curve would not be exactly the same, even completely different if worse.

2.2.7 Derivatives

Another interesting observation for Bézier curve is that, their derivatives are also Bézier curves, which is one degree lower. The differentiation of a Bézier curve is kind of straight forward, based on the general formulas (2.6), it can be written as:

$$B_i^{n'}(\varpi) = n(B_{i-1}^{n-1}(\varpi) - B_i^{n-1}(\varpi)). \tag{2.17}$$

Then the derivative becomes:

$$\mathbf{p}'(\varpi) = n \sum_{i=0}^{n-1} B_i^{n-1}(\varpi)(\mathbf{p}_{i+1} - \mathbf{p}_i), \quad \varpi \in [0,1]. \tag{2.18}$$

Thus it is possible to define the $k-th$ derivative of any Bézier by using the similar approach. Same for the formula (2.7), to express the derivative including the term weight, it could be reformulated as:

$$Bezier'(n,t) = \sum_{i=0}^{n-1} Bezier(n-1,t)_i \cdot n \cdot (w_{i+1} - w_i) \tag{2.19}$$

2.3 Trajectory optimization

Path planning algorithm, as discussed before, is mainly about to lead one or more feasible path from initial to the final position or pose, and therefore is somewhat a geometric problem. Compared with this, for problems which need to take the time change into consideration, an extra parameter t is required to be augmented in planner, and this kind of problem is named trajectory planning accordingly. Trajectory planning and optimization is the main concern for this thesis, since to achieve time deconfliction the assignment of speed is a vital part and time augment is taken as optimization variable.

Trajectory optimization problem is the crucial problem for the trajectory planning algorithm to achieve some certain objectives while satisfying constrains to stay feasible. Usually these objectives are related to some measurement of performance, for example minimizing total length of path, operation time or energy. It takes the form of general optimization control problem framework according to Roald (2015), and could be commonly formulated as following:

$$
\begin{aligned}
&Minimize \quad \mathbf{J}[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f] = \mathbf{E}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathbf{F}(\mathbf{x}(t), \mathbf{u}(t), t)dt \\
&Subject\ to: \\
&\qquad\qquad\qquad \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \\
&\qquad\qquad\qquad \mathbf{g}_i(\mathbf{x}, \mathbf{u}, t) \leq 0, \quad i = 1, ..., m \\
&\qquad\qquad\qquad \mathbf{h}_i(\mathbf{x}, \mathbf{u}, t) = 0, \quad i = 1, ..., m
\end{aligned}
\tag{2.20}
$$

Here $\mathbf{J}[\mathbf{x}(\cdot), \mathbf{u}(\cdot), t_f]$ is the objective function to be minimized, $\dot{\mathbf{x}}$ is equations of the system, $\mathbf{g}_i(\mathbf{x}, \mathbf{u}, t)$ represents the inequality constraint for the system and $\mathbf{h}_i(\mathbf{x}, \mathbf{u}, t) = 0$ represents the equality constraints.

For a constrained nonlinear quadratic programming problem in this thesis, according to Nocedal and Wright (2006), there are categorized optimization algorithm can be used to solve problems. For example Sequential Quadratic Programming(SQP) methods and certain interior-point methods for nonlinear programming. The general principle for sequential quadratic programming is to model a quadratic programming subproblem at each iterate subject to a linearization of constraints. If the problem is unconstrained it would reduce to Newton's method and find a search direction for this subproblem where the gradient of the objective vanishes. If the problem is only linear constrained, then it becomes equivalent to apply Newton's method to the first order optimality conditions. SQP algorithm is also utilized in this thesis to solve optimization problem, and would be presented again in later chapter. For interior-point method, it can be considered as extension of the primal-dual interiot-point methods for linear programming.

Also, if combining the objective function and constraint into a penalty function, the constrained problem can be reformulated as unconstrained problems. The penalty function can be defined like following:

$$f(x) + \frac{\mu}{2} \sum_{i \in \mathcal{E}} c_i^2(x), \tag{2.21}$$

Minimizing this unconstrained function for a series value of μ, where $\mu > 0$ is called penalty parameter, and the accuracy of the solution is rapidly growing sufficient. This algorithm is called penalty and augmented Lagrangian methods.

Actually, although there are lots of optimization techniques besides introduced above available to solve optimizations, none of them could guarantee that problem could finally be solved analytically optimal. Sometimes it can be really difficult in computation for an analytical optimal solution and most of the time the solution found are solved by numerical approximation and try to stay close to optimal one. And how close it is to optimal solution depends on the transformations and approximations being used.

2.4 Vehicle Modeling

This section would introduce some fundamentals in vehicle modeling, mainly about a surface vessel in two-dimensional space, including kinematics and kinetics aspects.

2.4.1 Kinematics

Kinematics only includes description of the position, velocity and acceleration of the object, while kinetics mainly analysis forces or moments causing the motion.

Reference frames

In kinematic problems there is a basic aspect needed to be demonstrated clearly at the very beginning, the reference frame. There are usually two kinds of frames: global reference frame and local reference frames. Global one is based on the Earth and place its origin on the center of Earth. It is used more in globally navigation such as Earth-centered Earth fixed (ECEF) reference frames, and Earth centered inertial reference frame. The local reference frame is generally placing the origin on some stationary point on vehicle itself. Each category contains several different reference frames, and in this thesis only two kinds of local reference would be used. Short introduction and definition basing on Fossen (2011) about these two frames is given.

NED frame is the abbreviation of North-East-Down coordination system $\{n\} = (x_n, y_n, z_n)$, with origin o_n being defined in Earth's reference ellipsoid. It is also a system that we use everyday in our daily life. Basically it is defined on the tangent plane on the Earth surface, with axis pointing differently than body-fixed axis of the vehicle. In this system, as its name indicates, the x axis is pointing towards the geographical North, the y axis is pointing towards geographical East, and the z axis points directly down, normal to the Earth's surface. *Body* fixed frame is a moving coordinate frame with reference to the Earth. Its origin is fixed on the vehicle itself with $\{b\} = (x_b, y_b, z_b)$, and the position and orientation of the vehicle are represented relative to the inertial reference frame, while the linear and angular velocities of the craft is described in the body fixed coordinate system. The origin o_b is often chosen as a point mid ship in the water line and would be referred as CO. For marine crafts, the three axes x_b,y_b and z_b are usually defined as below, like Figure 2.5 indicated:

- x_b - longitudinal axis (directed from aft to fore);
- y_b - transversal axis (directed to starboard);
- z_b - normal axis (directed from top to bottom)

2.4.2 Variable Notations

For marine crafts moving in six degree of freedoms(DOFs), in order to describe the position and orientation of the craft, six following independent coordinates are often used. The first three coordinates represent for the position along x,y, and z axes, while their

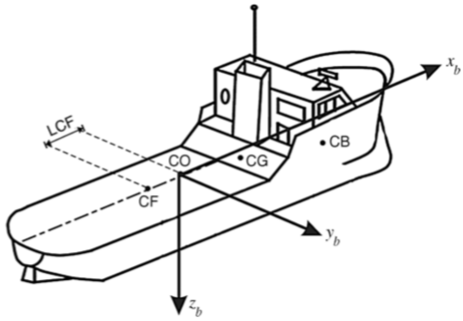


Figure 2.5: Body-fixed reference for marine craft from Fossen (2011)

derivatives of time correspond to accelerations. The last three coordinates and their time derivatives describe orientation and rotational motion. These six important motion is defined as surge, sway, heave, roll, pitch and yaw (see Table 2.1). The linear and angular velocity is defined in the BODY frame, and the position and Euler angles is given in NED frame.

Table 2.1: The notation of SNAME (1950) for marine vessels

DOF		Forces and moments	Linear and angular velocities	Position and Euler angles
1	motions in the x direction(surge)	X	u	x
2	motions in the y direction(sway)	Y	v	y
3	motions in the z direction(heave)	Z	w	z
4	rotation about the x axis (roll, heel)	K	p	ϕ
5	rotation about the y axis (pitch, trim)	M	q	θ
6	rotation about the z axis (yaw)	N	r	ψ

2.4.3 Transformation between BODY and NED

The transformation work between two frames can be done through rotation matrix R, which is usually denoted as R_b^a, representing transforming from coordinate a to b, and it is useful when performing the kinematic equation calculations of the motion. Usually it can be denoted as given below when transforming a vector from one frame to another:

$$\nu^{\mathbf{to}} = \mathbf{R}_{\mathbf{from}}^{\mathbf{to}} \nu^{\mathbf{from}} \tag{2.22}$$

And the rotation matrix R_b^n is a frequently used rotation matrix in guidance, navigation and control. To deriving the expression of R_b^n, first describing the horizontal motion by generalized position and velocity in reduced vectors, $\eta = [x, y, \psi]^T$ and $\nu = [u, v, r]^T$. The heave, pitch and roll motion in this stage is currently neglected, and rotational motion is only left with rotating about the z axis. Kinematic equation is thus can be formulated as:

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \tag{2.23}$$

And the rotational matrix $R_b^n(\psi)$ indicating the coordinate transformation from $\{\mathrm{b}\}$ to $\{\mathrm{n}\}$ is derived as below:

$$R_b^n(\psi) = \begin{bmatrix} cos(\psi) & -sin(\psi) & 0 \\ sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.24}$$

Chapter 3

Multiple Path Planning and Deconfliction

Multiple path planning is an important aspect for multiple vehicle motion control basing on the single path planning, and the key problem for this work is to well organize multiple paths and ensure every vehicle's operation could reach the goal. In this section, some important concepts and categories of algorithm that commonly used in multiple path planning are introduced, along with some related work from reference literature.

3.1 Multiple path planning algorithm

Consider an important example that usually happens in marine vehicle maneuvering applications: Several autonomous vehicles are launched at one place or scattered at more different positions at the same time, and expected to cooperate and execute a mission together, maneuvering from initial place to prescribed target neighborhood respectively at a same time. Then the further mission like underwater detection or exploration can be performed. Since they might be operating in a restricted area and finally come quite close to each other, this mission probably takes the risk of colliding with support ships, static or moving obstacles (like floating buoy, coastline) or other vehicles, and thus for multiple vehicle path planners, deconfliction task is the main challenge that must be solved to guarantee the operation safety. It is particularly crucial in multiple vehicle control, requiring the algorithm to not only generate a feasible path leading to the goal point, but also never allow any other objects to come to close vicinity of each other in space. This property is the main focus of the concept *deconfliction*. Furthermore, vehicles might be required to arrive at approximately same time and speed, and this problem can be detailed with more constraints, like dynamic limitations of vehicle velocity and accelerations, total energy restrictions for the whole maneuvering process, and environmental constraints including external disturbances, caused by ocean currents and sea waves. Solving this kind of task in path planning problem also can yield a number of objectives to be optimized in the process, such as proposing an energy-related cost function, or minimizing executing time or path length.

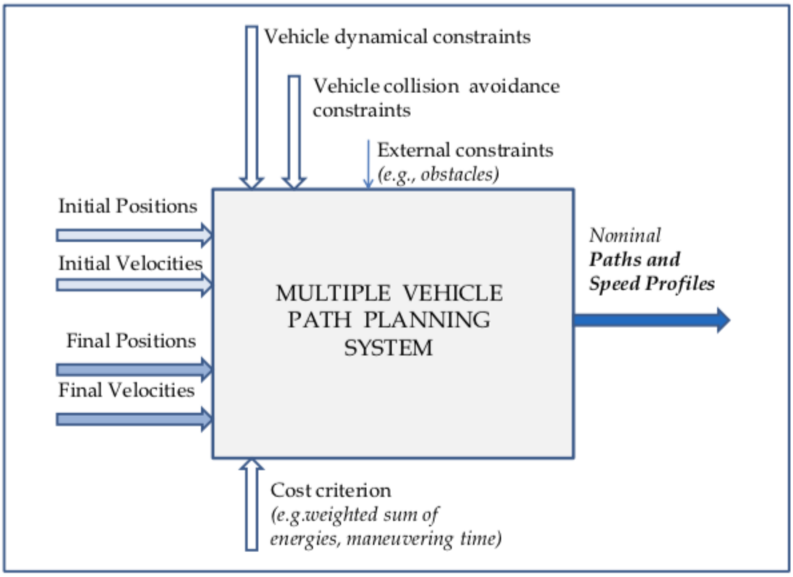


Figure 3.1: Multiple path planning system from Häusler et al. (2009)

In this section, two kinds of algorithms about multiple vehicle path deconfliction will be introduced, which are based on time and space assignment respectively. In order to differentiate vehicles, let $V := \{V_i;\ i = 1, ..., n\}$ denotes the set of vehicles involved, where $n \geq 2$. A general framework of multiple path planning system is constructed according to Häusler et al. (2009), as illustrated in Figure 3.1 above.

3.2 Temporal deconfliction

Recalling the difference in the concept of path and trajectory, spatial deconfliction would not require an extra degree of freedom - time, and thus could still remain in the field of path planning, if further constraints regarding time and speed are not required. Basically, the main difference between these two algorithm is, temporal deconfliction allows for each vehicle to plan path separately, taking only obstacles and constraints into consideration and its route can intersect with others in some places. The main point is assigning proper speed to stagger the time for different vehicles to pass the same point, and guarantee that at no time they come close to the vicinity of each other.

In comparison, for spatial deconfliction, time coordinate is not a main concern in planning and is usually being considered in further constraints regarding arriving speed or later stage like path following. The major focus is therefore to take each vehicle's running route into account together at the same time, and restrict the closest distance between them to be larger than a prescribed minimum safety distance, say E. The intersection interval or crossing point would at no time appear spatially in this situation, like illustrated in Figure 3.3. Consequently, time deconfliction algorithm is capable of providing solutions of a larger class of problems than spatial deconfliction algorithms can deal with, due to the augment of time coordinate as an extra freedom. In spatial deconfliction, time coordinate is not a necessary part to be included. However what in common is, spatial coordinates and time parameter are both decoupled and dealt with at different stage in this two algorithms, which gives more potential in further improving and allowing more versatility.

Addressing the problem of temporal deconfliction, since it allows the path to come to close vicinity and intersect spatially, the path planning for different vehicles can be dealt with separately, to its assigned target which can be formulated respectively. The most crucial part, in achieving the objective is the speed assignment optimization, which staggers the time in crossing the intersection point and ensures a collision free operation in multiple vehicle motion control. As a result of this, intersection points and its vicinity where collision might possibly happen should be detected and marked after finishing path planning process, like the area indicated by a red circle in Figure 3.2. It is the place which would be referred as danger zone in later chapter, and needed to be treated carefully for this problem.

Actually for temporal deconfliction path planning problems, it seems if once the path and velocity has been assigned, it is strictly required that the vehicle should follow the path precisely as planned. This sets a high standard for trajectory tracking execution, and would no doubt meet with considerable problems. In this way, synchronized arriving would also be a tough manner. In fact, this is the main drawback for the time deconflicted path planning, and in real applications, deviation spatially and temporally could always appear due to environment disturbance or mechanical failures. Thus the use of absolute time, which on-board systems on the vehicle are using, is not practical and necessary to be avoided. Also, it is vital to propose an efficient path following control, replanning from the current position as reaction to the deviation from original plan and controlling the vehicle try to maintain the path. This kind of problem is called Time Coordinated Path Following(TC-

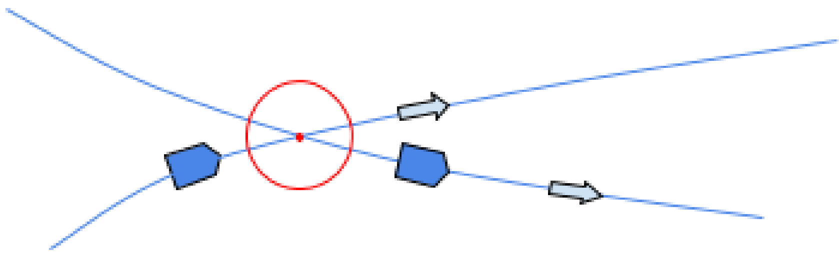


Figure 3.2: Temporal deconfliction for multiple path planning

PF) and is significant in the later stage for temporal deconfliction path planning. Methodologies relating to this can be referred to Ghabcheloo et al. (2009a), and some relevant introductions would be presented in later chapter as literature survey for related work.

3.3 Spatial deconfliction

Spatial planners focus on the path planning in the operation space, and with no present of time coordinate during the planning process. It requires that each vehicle should follow a feasible route to the target while always keep a minimum prescribed safety distance E with other vehicle, in other words, their route at no where would intersect with each other. This could be quite demanding under some certain situations, especially when there are many objects existing around and the operation space is relatively limited. It takes other vehicles as moving obstacles and avoids to pass all the places others will or already passed, and this obviously narrows the set of the feasible solutions, thus the set of problems it could tackle with is of smaller size compared with time deconfliction. But the main superiority it may have over temporal deconflicciton is that, it does not need to worry about the high precision path following problem, both in space and time.

Generally, for an open and wide space it would be easier to choose spatial deconfliction algorithm obtaining a feasible path with collision avoidance, and reduce the computation loads both in planning and replanning. If extra constraints are required in motion speed or arriving time, time coordinate could be further involved, otherwise time could never be necessarily considered in this situation.

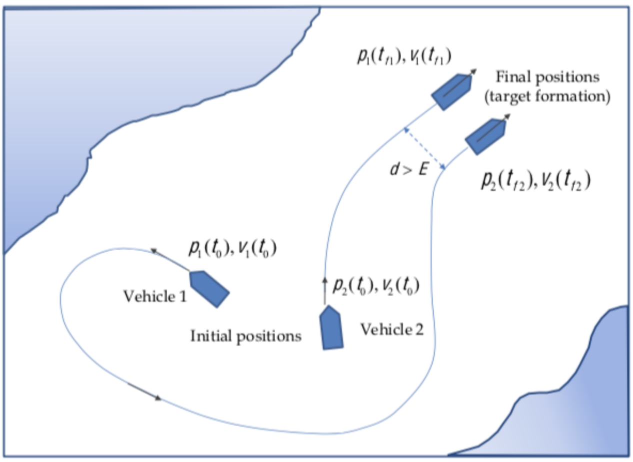


Figure 3.3: Spatial deconfliction for multiple path planning in Häusler et al. (2009)

In this case, the general form of optimization problem could be used as following, which would be presented with more detailed method in Häusler et al. (2009):

$$
\begin{aligned}
&Minimize \sum_{i=1}^{n} \quad w_i J_i \\
&Subject\ to\ geometric\ conditions\ or\ dynamic\ constraints \\
&for\ any\ i\ \in\ [1, n], \\
&Minimize\ ||p_j - p_k||^2\ \geq\ E^2 \\
&for\ any\ j, k = 1, ..., n, j \neq k.
\end{aligned}
\tag{3.1}
$$

Spatial deconfliction here in this thesis would only be presented as alternative methods regarding deconfliction task, and only discussed with basic formulation as additional information. In this thesis, temporal deconfliction is the main task and concern.

3.4 Related work

Multiple path planning problem is a crucial topic for automation and has been studied by researchers in various fields during the past ten years. For this topic many different approaches have been already proposed. Some algorithms stem from the robotics field and computer science, some mainly focus on the marine vehicle and being used in marine control systems and vessel maneuvering.

3.4.1 Path generation algorithm

For path generating algorithms, Lande (2018) takes the advantages of Bézier curve to generate a smooth curve which leads a feasible path starting from the initial point to the goal point, and avoiding all the random generated static obstacles within a bounded 2D space. He limits the curve shape by utilizing characteristics like convex hull property, to ensure that the curve would never exceed the boundary and cross the section where static obstacles lie in. And further, he uses Bézier derivatives to add constraints on the curvature, and guarantee the path that is smooth and continuous in both C^2 and G^2. To formulate the objective function he utilizes differential flatness to assign cost, minimizing the energy associated with the path and taking the vehicle dynamic into consideration. Then he takes control points as optimization variables to solve this nonlinear constraint problem and obtain the optimal control points as solution. His work presents the efficiency and potential for Bézier curves to generate feasible path, capable of avoiding large number of obstacles while stay continuous and curvature constrained.

3.4.2 Multiple path planning

Recalling the concept configuration space introduced in Chapter 2, basing on this some path planning algorithms have developed different approaches in multiple vehicle path planning, such as coordinated path planning for multiple robots proposed by Svestka and Overmars (1996). They utilized a series of identical simple robots and composite them together as one robot with many degree of freedom, and this robot is the so-called composite robot. The path feasible for the composite robot with mutual collision avoided is referred to as coordinated path. By first constructing a simple roadmap for single robot, n of such roadmaps are combined into a roadmap as a super-graph. Then they adopt two structure of super-graph for retrieval of coordinated paths. By doing this rather than adopting usual decoupled planning, they achieve a centralized approach which allows for complete planners that could always find a solution as long as it exists. Further they prove that their proper construction of the simple roadmaps guarantees probabilistic completeness of the resulting planners.

Specifically for multiple marine vehicles path planning, in Hausler et al. (2009), they utilized direct optimization methods in path generation, parameterize the path and use an algebraic polynomials in N degree to represent the coordinates of each point on the path. The degree N is determined by the number of boundary conditions, and the value of N could be set a bit larger to allow for additional degree of freedom. However, the main focus and contribution of this paper is the deconfliction work for multiple vehicle path planning, which is also an important reference for this thesis. In order to achieve inter-vehicle collision avoidance, they consider the deconfliction work from two aspects, spatial and temporal. To consider the practical issue that using absolute time would not allow any deviation from the original plan, and therefore place high requirements on the accuracy of path following which is not practical for real-time operation, they adopt a different solution which dispense with the absolute time but ensure the simultaneous arrival at the target point. By doing this the optimization process includes the timing laws that indicate how the speed of each vehicle evolve along the path instead of absolute time constraints, and allows for replanning and online modifications if any deviation appears during the whole process. Furthermore, spatial and temporal constraints could be decoupled and treated respectively.

In fact, ideas in Hausler et al. (2009) which is proposed by Ghabcheloo et al. (2009b) at first has led to an integrated and complete strategy in multiple path planning and control. For temporal deconfliction work specifically, they mainly extend the method which could be referred to a paper from Kaminer et al. (2007) with more details. In that paper from Kaminer et al. (2007) they re-parametrized the trajectory by a variable which they call virtual time. And basing on this variable, temporally deconflicted trajectories for multiple vehicles could be conveniently obtained, leading to a series of spatial paths as well as the speed profiles all along the way. Then, next step they design a path following algorithm, controlling each vehicle to follow its path as planned and track the trajectory with the corresponding speed profile. In this step, just as introduced before absolute time is not being used. Finally, the work is needed on coordinating the relative motion of the vehicles along their paths, to make sure the trajectory is deconflicited and desired constraints are met in operation. The step two and three are actually the following work after the temporal deconfliction has been done, and all three steps are significant in order to implement this series of strategy into practice. For last two steps, more detailed theoretical support and introduction could be found in Ghabcheloo et al. (2009b), and is referred to as Time-Coordinated Path Following (TC-PF) in that paper.

Another approach in spatial deconfliction is made by Lizarraga and Elkaim (2008), basing on the path generation algorithm using Bézier curve for multiple UAVs. They first explore the characteristics of Bézier curves, for example convex hull property and de Casteljau algorithm, and take the control points as the optimization variables. Further in order to achieve spatial deconfliction, they set the constraint as limiting the minimum distance between each two path to be larger than E, the so-called minimum separation to ensure safety. The key points for this paper is to ensure the generated paths lie within a prescribed 3D airspace, that is never exceeding the boundary while space deconflicted. They obtain this mainly basing on the convex hull property, limiting Bézier curves by using control polygons to continually approximate the desired goal space. Furthermore, the cost function is set to penalize excessive lengths to obtain shortest path for each UAV. This paper shows an approach for Bézier curves implemented in multiple path planning deconfliction problems, and demonstrate its potential and efficiency in achieving spatial deconfliction within 3D space for UAVs.

Extending these multiple path planning algorithms and utilizing different cost functions, one could further tackle with issues with more practical objectives and solve problems that related closely to real-time operation. For example, in Bitar et al. (2018), they formulate an optimization control problem with an energy-based objective function, which is defined as the work done by the actuators. To increase the efficiency they further make an approximation on this objective function and solve the optimization problem by using pseudospectral algorithm. Another example is still from Häusler et al. (2010), where they present a multiple vehicle path planning problem with currents and communication constraints. In this case the vehicle is operating with the existence of ocean current, thus the direction of the vehicle's speed and the thrust would change accordingly. The vehicle's speed needs to be oriented such that the resulting velocity is following the velocity profile as planned, and this is also included in the construction of cost function in the stage of path planning besides path following, to make the solution to be the most energy efficient one even facing a current. And for communication constraints, it indicates that if two vehicle's distance exceeds the maximum magnitude of C, the loss of communications would occur and this situation needs to be avoided as well. They consequently add an restriction, as the environmental constraint to include this, which has been proved to has an obvious effect on the result.

3.4.3 Optimization method

Actually for multiple path planning, it has long been studied and taken as an multidisciplinary problem, since this would involve many aspects of knowledge in application for example cybernetic engineering, computer science and mathematics. Methods for solving this kind of problem are various as well, other than optimization there are still many other approaches could be used such as `A*` algorithm from computer science. For this thesis that only concentrates on the optimization algorithms, there are also lots of different approaches could be referred to. In Forsmo (2012), he performs optimal path planning for unmanned aerial systems with the goal of finding a path from start point to the end target, passing several desired waypoint and avoiding stationary obstacles. He chooses Mixed Integer Linear Programming(MILP) to solve this since this method could efficiently solve non-convex problems, provided with many powerful solvers like Gurobi and CPLEX optimizer. And in Roald (2015), he formulates a simple path planning problem within a 2D space, starting from a point with an initial pose and find an optimal way to reach the end point with desired orientation. Then he implements and tests numerical optimization methods to solve this problem, mainly semi-Lagrangian approximation scheme and pseudospectral numerical optimization method. In a word, to solve optimization problems, there exists various methods and each has its own features and benefits. Basing on the methods' characteristics and the way of formulating the problem, one could choose which method is most suitable and how the problem can be better formulated. Different methods could behave and give results quite different in the end.

Chapter 4

Deconfliction and optimization

This chapter will introduces the method proposed to solve the main objectives of this thesis, that is the temporal deconfliction path planning for multiple vehicles, as well as its formulations and implementations. The background theories and related algorithms presented in the previous chapters have provided the theoretical basis and support for this method.

4.1 Preliminary description

The multiple path planning algorithm that presented in this chapter, is proposed to lead arbitrary number of marine vehicles to the assigned end points respectively, with collision avoidance for static environmental obstacles. These paths would be generated in the North-East plane.

4.1.1 Method

The main focus is to guarantee that the inter-vehicle collision is also avoided, meaning a fleet of vehicles would be able to move safely and execute missions cooperatively. This deconfliction work could be done temporally or spatially. For spatial deconfliction algorithm, its basic idea is to never allow any vehicles to come to other vehicle's close vicinity or has intersection points on path with other's in space, thus every vehicle would operate kind of separated and at no time appear in the same place or neighborhood. For temporal deconfliction algorithms, however, with time t as one more degree of freedom than spatial, it allows intersection points on the path and two vehicles could pass a same point, if only these occurrences are separated in time. This leads to more possibility in find feasible solutions, and therefore make temporal deconfliction able to deal with a larger number of problems in complex situations. Thus in this thesis, problems are formulated with time coordinate and solved by temporal deconfliction.

During the planning for feasible paths, using absolute time which trajectory tracking system on board uses will lead to a main drawback. This would make the requirements for the accuracy of the path following very strict, and not allow any deviations from original plan. Thus to avoid this problem and make the algorithm more practical, dispense with the absolute time and adopting proper timing law to indicate how speed should be involved would be a better way. Consequently, simultaneous arrival of the vehicle is formulated in the optimization problem, which provides the possibility of replanning and modification if the deviation occurs.

In this thesis, the arriving time of each vehicle is constrained to be strictly simultaneous, and the final time of arrival is not specified with a certain number (but has been constrained with a maximum time of arriving). This could be achieved with this algorithm, but in the thesis the operating time of vehicle is minimized such that the total time would be the shortest.

4.1.2 Objectives

There are four main objectives and desired properties for the proposed algorithm, which will be analyzed and formulated in later section into optimization problem. These objectives are listed below.

Time deconflicted

In this stage a proper way of assigning speed is needed to be proposed and thus produce the speed profile along with the path. To staggering the instant each vehicle passes the intersection point, constraints could either be based on time or speed, but the velocity profile is what needed to pay attention to, in case replanning in path following is needed. Additionally, considering the actual size of the vehicle, taking the intersection point as the center, a zone with a certain range is needed to be marked, rather than a small interval.

Simultaneous Arrival

This requires that all the vehicles should arrive at its assigned end point approximately at the same time, despite the different length of the paths. This could be done by restricting the total time of vehicle operation and be formulated as equality constraints. In this thesis the exact time instant of arriving is not specified, but by limiting the minimum velocity of the vehicle, the maximum time of arrival has already been constrained.

Minimum Time

For the purpose of guarantee the mission execution efficiency, total time is included in the cost function to be minimized as the shortest. In fact, the construction of the cost function could be viewed in two different ways. Except achieving some certain objectives, cost function could also be used to penalize something which is not wanted. In this thesis only the total execution time is used. But further, terms such as energy or vehicle dynamic could also be taken, restricting the vehicle's behavior. Additionaly, the starting and ending speed has been penalized in case the vehicle is running too fast at both end, however it is only an implementation in simulation in this thesis, and can be changed later according to the real need.

Dynamic constraints

Considering the physical limitations of the vehicle's actuators, they apparently cannot produce as many thrust as needed. And there might exist some environmental disturbances for marine vehicles, for example ocean current which will influence the magnitude or direction of the velocity as well. Thus the vehicles would have a certain range of the speed which they are able to operate with. Similarly, the acceleration of the vehicle has to be restricted, in order to avoid the wear and tear, and make sure the vehicle could run in a smooth process. This also will contributes to a good performance in the vehicle path following.

4.2 Optimization formulations

There exists several approaches in multiple path planning. Some might base on the search algorithm like A*, some are using mathematical modeling and optimization. In this thesis the method is based on the optimization algorithm, and in this section formulations and corresponding analysis will be presented.

Optimization is a widely used theory in many fields like engineering, science and economics, and especially useful in analysis and searching. It mainly serves as a mathematical tool for the searching of feasible solutions, and for different kinds of problems it could provide many powerful algorithms as solver. The optimization problem usually could take the form as given below:

$$min \quad f(x) \tag{4.1a}$$

subject to:

$$\begin{aligned}
&A_{eq}x = B_{eq} && (4.2a)\\
&Ax \leq B && (4.2b)\\
&c_i(x) = 0, \quad i \in \mathcal{E}, && (4.2c)\\
&c_i(x) \leq 0, \quad i \in \mathcal{I}, && (4.2d)\\
& && (4.2e)
\end{aligned}$$

This form is also suitable for this thesis's formulations. The first part $f(x)$ called objective function, is taken to measure the performance of the system being studied in the problem. It is formulated as the quantitative measurement of the objective and minimized with respect to the optimization variables. However in another way of thinking, objective function could not necessarily be constructed only related to the goals, it can also be used to penalize something that is not wanted in this problem, since it would finally be minimized as much as possible. Thus it could also be called as cost function.

In the second part, formulations and indices sets $\mathcal{E}$ and $\mathcal{I}$ represent the restrictions or the equality and inequality constraints that the optimization variables are limited by, respectively. Only by satisfying these constraints could a feasible solution be obtained. The process of defining variables, formulating objective function and constraints for a physical problem in this form is called modeling. It however might be the most crucial part in the optimization process, since the way it formulated relates closely to the choice of proper algorithm to solve the problem and the complexity of the model would have big influences on how the result will be. The way of modeling could be differ, and various of algorithms have been proposed so far available for solving optimization problem. A detailed presentation of optimization algorithms and theories could be referred to Nocedal and Wright (2006). In this thesis, the optimization proposed belongs to the nonlinear constrained problem, due to the formulation of constraints.

4.2.1 Optimization variables

In this thesis path planning for multiple vehicles process could generally be separated into two stages, first planning paths for multiple vehicles, and then assigning speed to satisfy constraints such as time deconfliction. This is actually two optimization problems with different optimization variables. First path planning basing on Bézier curves takes control points as decision variables. As for finding the optimal speed profile, time coordinate is taken as the optimization variables, since using speed would lead to more nonlinear constraints than time and is not simple to formulate. The output from the path planning is sets of x-coordinates and y-coordinates expressing the spatial information. It is actually many small line segments connecting together to form a curve, and not completely continuous everywhere. In another words it indicates the coordinate of an arbitrary point on the path is not obtainable. Basing on this, assume for segment i on the path for vehicle j, the time spent on traveling in it is denoted as t_i^j. By this setting the optimization variable for the problem can be expressed in a time vector as given below:

$$\begin{aligned} \mathbf{t} = \{\, t_1^1, ..., t_n^1\,, t_1^2, ..., t_n^2, \quad ... \quad , t_1^j, ..., t_i^j, ..., t_n^j, \quad ... \quad , t_1^m, ..., t_n^m\}, \\ i \in \{1, ..., n\},\ j \in \{1, ..., m\} \end{aligned} \tag{4.3}$$

where n is the number of the line segments for one path, and m represents the number of paths, or vehicles being planned. In a shorter form, this variable vector could also be express as:

$$\mathbf{t} = \{\mathbf{t^1}, ..., \mathbf{t^j}, ..., \mathbf{t^m}\}, \quad j \in \{1, ..., m\} \tag{4.4}$$

4.2.2 Objective function

Generally objective function could be considered as a tool to evaluate the solution and select the optimal one, with respect to the optimization goal. In path or trajectory planning problems, the goal is usually to be the energy minimized, path length shortest or additionally combined with some other features. One objective for this thesis is to obtain the time shortest solution, thus the objective function needs to include the summation of every small time interval for the line segments, for one single path j. That is:

$$f(x) = \sum_{i=1}^{n} t_i^j, \quad i \in \{1, ..., n\} \tag{4.5}$$

This is because the arriving synchronization will be guaranteed in the equality constraints, and the total time would be almost the same in the end. However, the choice for the path j is not random. Thinking of these multiple paths starting and ending both at different points, their lengths will differ and the longest path would normally take the longest running time, according to the velocity limitations. Thus there exists a minimum magnitude for the time that all the vehicles would need, and that is basically decided by how the speed of longest path is assigned. Vehicles on shorter paths need to wait for the slow one on the longest, and it is not hard for them to run faster and arrive earlier. So in order to ensure the shortest time consumption, choosing the longest path as the minimization object is more effective than others. This is similar to the Bucket effect, which is also called Liebig's law of the minimum. The lengths of every path are obtainable after planning is finished, and the longest path could be selected and marked to be used later.

The objective function here can further be augmented with terms related to features other than time, such as vehicle dynamics and related energy. For example including accelerations would force the vehicles to run under a speed as constant as possible. These term can be added along with a coefficient to distinguish the strength of penalty, such as:

$$f(x) = \alpha f_1(x) + \beta f_2(x); \tag{4.6}$$

And this can be improved in the future work. Here in this thesis, dynamics are taken care of in formulating constraints. Cost function uses a linear summation term of time, in order to avoid more nonlinear terms. And additionally, through simulation an observation found is that the velocity sometimes starts and ends with a quite high value, which is not a desired phenomenon since the vehicle can not directly run under its maximum speed at the very beginning, and it is supposed to stop at the end. Thus a penalization term regarding the starting and ending speed has been added in the cost function as well.

4.2.3 Constraints

Similar to objective function, constraints can also be viewed as a selecting tool, to determine if a plan is a feasible solution for this problem. In the path planning problems constraints represent all the requirements for the path such that a vehicle is able to follow. These constraints are also needed to be constructed with the optimization variables either in equality or inequality form, and in this part formulations for all important constraints will be presented.

Simultaneous arrival

As discussed before by restricting the simultaneous arrival has its own practical meaning in real-time operation. The main point is that paying attention more on the speed profile instead of time instant. By this setting the time coordinate used in the path planning is no longer the absolute time used in the on-board system of vehicle, but only a relative virtual concept. This can guarantee that during the operation the plan of motion can be adjusted any time in order to achieve the desired objective, and therefore allows for the occurrence of deviation. In this thesis the final arriving time for all vehicles are constrained to be the same and minimized to be the shortest. However it is also achievable for this algorithm to specify the total time with a certain number, if is required.

Here basing on the construction of the decision variables, simultaneous arrival can be formulated as a linear equality constraint, or linear inequality constraint to allow for a little difference but within an acceptable range. The decision variable is composed of many small time intervals that correspond to the curve segments on the path in space, and is defined as the time vehicle needed to spend traveling on the line segments. Thus, the summation of time intervals for different path should all be equal, or the absolute value of their difference is not allowed to exceed a small number, say 0.5 second, for example. Then the formulation for equality constraints is:

$$\sum_{i=0}^{n} t_i^1 = \sum_{i=0}^{n} t_i^2 = \quad ... \quad = \sum_{i=0}^{n} t_i^j = \quad ... \quad = \sum_{i=0}^{n} t_i^m, \qquad i \in \{1, ..., n\}, \quad j \in \{1, ..., m\} \tag{4.7}$$

For inequality constraints however, it seems that it can not be directly formulated into a linear form since it includes the absolute value. But thinking of the longer paths are more possible to take a longer time, the absolute value of difference can be changed to the longer path minus the shorter one. And it is not hard and complex, since the length of every path is obtainable after the paths are planned, and can be put in a order like from shortest to the longest when constructing the optimization variables. This would be useful for objective function as well since it also needs to use the profile of the longest path.

Temporal deconfliction

Specifically, by including the time coordinate, the spatial path planning for multiple vehicle and speed or time assignment work is separated in two different stages. Basing on the path well generated first, the intersection points that indicates where the collision could possibly happen will be clearly presented. Considering the actual size of vehicles, it is more safer to mark a zone with a certain range as the danger zone, with the intersection point serves as the center. Vehicle motions within this zone would all be necessarily taken care of. To formulate this, first the danger zone is needed to be marked, and this could be done by finding the intersection point and then extending the range backward and forward, like illustrated in the Figure 4.1.

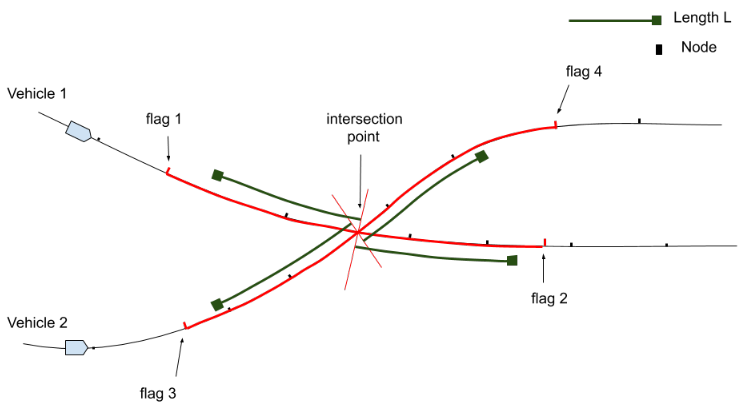


Figure 4.1: Searching for the boundaries of the danger zone

Four boundary points of the danger zone could be fixed consequently, denoted as flag 1 to flag 4 respectively. The range of the zone is denoted as L, which in this thesis taken as L=15 meters, for example. One important problem needed to be clear is that, like mentioned before, the output data from the path planning describing the path are sets of x-coordinates and y-coordinates, and the path is in fact formed by many small curve segments connecting together. This would lead to a problem that the intersection point would possibly lie within some curve segments but not on the node, and its exact coordinate is thus unknown. To mark the intersection point, only the last nodes before it could be found, that is the black dot in the Figure 4.2. Similarly when extending the distance of L towards two different direction, only the nearest node could be confirmed and marked as the boundary. Then the searching criteria is that whether the length is larger than L, if it is, mark the next node and record it as a flag. The district between four flag points which indicated by the red line is the so-called danger zone in Figure 4.1 and 4.2.

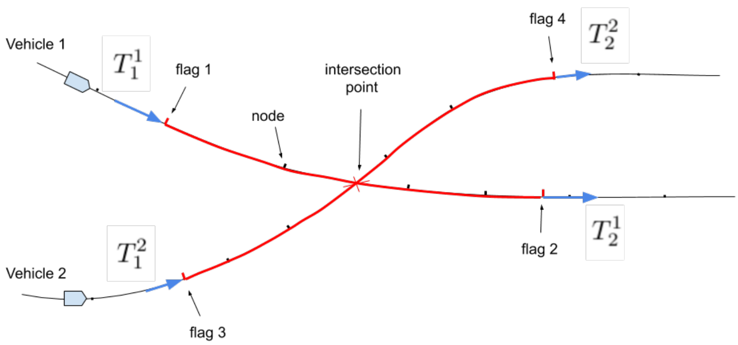


Figure 4.2: Time deconfliction for two vehicles

Further to restrict the time and achieve time deconfliction, the constraints could be set as given below, assuming there are vehicle 1 and 2 might collide here:

$$T_1^1 > T_2^2 \quad or \quad T_1^2 > T_2^1; \tag{4.8}$$

Note here the capital letter T means that this is the total time the vehicle needs to spend to reach here, in other words, the summation of the small time intervals before this point. The index 1 represents the vehicle entering time instant, index 2 represents the vehicle exiting time instant, and the superscript 1 indicates that it is the time for the vehicle 1, the superscript 2 indicates it is the time for the vehicle 2. This setting is illustrated in the Figure 4.2. And like what used in previous sections, the lowercase letter t represents the small time interval corresponding to one certain curve segment. The following formulations can be constructed:

$$\begin{aligned} T_1^1 &= \sum_{i=1}^{flag1} t_i^1, \quad T_2^1 = \sum_{i=1}^{flag2} t_i^1 \\ T_1^2 &= \sum_{i=1}^{flag3} t_i^2, \quad T_2^2 = \sum_{i=1}^{flag4} t_i^2 \end{aligned} \tag{4.9}$$

These could be explained as, if satisfying the constraint $T_1^1 > T_2^2$, then the vehicle 1 would enter the danger zone after vehicle 2 has left; if otherwise satisfying the constraint

$T_1^2 > T_2^1$, then the vehicle 2 would instead enter the danger zone after vehicle 1 has left. Either of these two conditions suffices the requirement of time deconfliction, and it guarantees that these two vehicles would at no time appear in the danger zone at the same time.

Dynamic constraints

Vehicle dynamics in this thesis can be analyzed from two aspects. First to constrain the velocity, a minimum value is needed to be set, ensuring that the operation would not take too long and gives the total time a maximum restriction. Maximum value could be set basing on the actuator or the environment, or to meet the real need. As for the acceleration, it is also necessary and crucial, determining whether the path could be perfectly followed. Unreasonable assignment of acceleration not only causes serious deviations in path following, but also leads to wear and tear to the machine and damaging the system.

Velocity constraint formulation relates closely to the way of speed assignment. Based on the path description in path planning stage, spatial profile about paths obtained are sets of x- and y-coordinates. Also considering the number of curve segments are usually several hundreds for one path and their length are pretty small, velocity could be defined by length for every curve segment divided by time, which is t_i^j from previous section. Assume length of curve segments is denoted by s_i^j, thus velocity will be:

$$v_i^j = \frac{s_i^j}{t_i^j}, \quad i \in \{1, ..., n\}, \quad j \in \{1, ..., m\} \tag{4.10}$$

Although it is nonlinear, the numerical value of s_i^j is already known, thus turning this formula into the range of t_i^j and making it serve as the lower and upper bounds for the optimization variables. In fact these hundreds of segments are all divided basing on the De Casteljau's algorithm, and number of segments can be set manually. An important note here is, the length s_i^j is not uniformly distributed everywhere. This is due to the nature of Bézier curve when it generates curve. And because of this, the velocity v_i^j is actually an average value of the true velocity in one segment, with different length of time. If plotting it, the figure will be like what illustrated in Figure 4.3.

Therefore, ignoring the different duration of each velocity, taking it as a point and use the time difference directly as $T_{i+1}^j - T_i^j$, or just t_{i+1}^j is not precise. To calculate accelerations a way of approximation will be picking the mid point of every interval and connecting them together, changing the bar chart to a line chart, like denoted by the red line. Then the acceleration turns into the slope of each small line, and formulates as:

$$a_{i+1}^j = \frac{v_{i+1} - v_i}{(t_i^j + t_{i+1}^j)/2} \tag{4.11}$$

In this thesis, the absolute value of accelerations are restricted to be no more than $2m/s^2$, and the velocity is ranging from $v_{min} = 10m/s$ to $v_{max} = 30m/s$ at most.

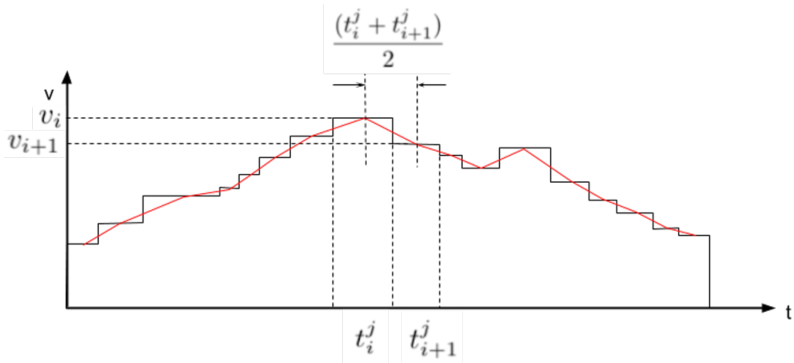


Figure 4.3: Example velocity figure

4.3 Implementation

This section describes the implementation process and gives details about the proposed path planning algorithm, thus to present how the algorithm has been built for numerical simulations.

4.3.1 Implementation environment

This problem has been formulated and implemented in the MATLAB developed by the MathWorks™. The main optimization algorithm used to solve this problem is SQP (Sequential Quadratic Programming), and the nonlinear programming solver is chosen as `fmincon` in the Optimization Toolbox of MATLAB. All constraints and objective function are further modified to fit this solver.

4.3.2 Parameters

In the initial settings, there are some parameters that can be modified by the user, making this algorithm fit for various situations and meet the real need. The constraints would therefore changed along with this without modifying the formulation inside the solver.

Table 4.1: Parameters for path planning

Starting points	$[x_0^1, y_0^1, \psi_0^1; ....; x_0^m, y_0^m, \psi_0^m]$
Destinations	$[x_n^1, y_n^1, \psi_0^1; ....; x_n^m, y_n^m, \psi_n^m]$
Number and Degree of curve segments	(n_c, d_c)
Minimum turning radius	R_{min}
Number of obstacles	n_o
Size range of obstacles	(r_{min}, r_{max})

Firstly the number of the paths to be planned can be arbitrarily assigned. Specifying all the starting points, destinations and the heading angles in these two points with three matrices respectively, the multiple paths in a 2D space can be planned. Also there are some other parameters can be changed during path planning, and is summarized above in the Table 4.1. Further inputting the velocity range, danger zone range and acceleration limitations, the constraints in the time deconfliction can be constructed accordingly, and these parameters are summarized in the Table 4.2. The parameter n in the Table 4.2 indicates the number of intervals that a path would be divided into. If setting this number large the result would be more accurate theoretically, however in implementation this number would not only influence the computation loads, but also the path generated. Thus a better value has to be set basing on simulation results where the trade-off might be needed during the process.

Table 4.2: Parameters for temporal deconfliction path planning

number of path	m
velocity range	(v_{min}, v_{max})
Maximum acceleration	a_{max}
danger zone range	l

The optimization solver `fmincon` requires an initial guess for the optimization variable to generate numerical value for all constraints, and here in the implementation the minimum and maximum value of velocity have been taken.

4.3.3 Motion representation

The resulting plan for the paths are along with the speed profiles of each vehicle, and after the simulation, the velocity and time, position and length, all these relative information about this process would be presented in the vector form. When constructing the optimization variables all the paths will be put in one vector in sequence. This setting produces all the data in the same form and order, as illustrated below:

$$\begin{aligned}
\mathbf{x} &= [\, \mathbf{x^1};\ \mathbf{x^2};\quad \cdots \quad;\ \mathbf{x^m} \,]; \\
\mathbf{y} &= [\, \mathbf{y^1};\ \mathbf{y^2};\quad \cdots \quad;\ \mathbf{y^m} \,]; \\
\mathbf{s} &= [\, \mathbf{s^1};\ \mathbf{s^2};\quad \cdots \quad;\ \mathbf{s^m} \,]; \\
\mathbf{t} &= [\, \mathbf{t^1};\ \mathbf{t^2};\quad \cdots \quad;\ \mathbf{t^m} \,]; \\
\mathbf{v} &= [\, \mathbf{v^1};\ \mathbf{v^2};\quad \cdots \quad;\ \mathbf{v^m} \,];
\end{aligned} \tag{4.12}$$

And each element in the vector is another vector itself, for example $\mathbf{x}$:

$$\begin{aligned}
\mathbf{x^1} &= [\, x_1^1;\ x_2^1;\ \cdots\ ;\ x_n^1 \,]; \\
\mathbf{x^2} &= [\, x_1^2;\ x_2^2;\ \cdots\ ;\ x_n^2 \,]; \\
&\ \ \vdots \\
\mathbf{x^m} &= [\, x_1^m;\ x_2^m;\ \cdots\ ;\ x_n^m \,];
\end{aligned} \tag{4.13}$$

Thus all the data are stored in three-dimensional matrices during simulations. Total running time would be calculated by summation of each element from t^1 to t^n, showing the vehicle is arriving simultaneously. And since the boundary marking flags have been recorded when fixing the danger zone, by retrieving the data and doing summations, the vehicle passing time instant can be obtained as well showing that their paths are all time deconflicted.

Chapter 5

Numerical simulation

This chapter presents several simulation scenarios to evaluate the performance of the proposed algorithm. There are four different scenarios shown in total, mainly with the change of the vehicle number, showing the efficiency of this algorithm on arbitrary number of paths time deconfliction planning. And their detailed parameter specifications are also listed in tables respectively.

5.1 Preliminary comments

In the following four situations, different number of paths has been planned and time deconfilicted. Each problem will be specified with different parameters which are summarized in tables. During the simulation an important observation is that the change of the path number would have influences on other parameters that can not be ignored during the optimization process. This will be discussed in details in later chapter.

In all figures, paths are generated in the North-East plane, and their corresponding speed assignments would be shown followed the path plan. Different color in the figure indicates the different path, and the same path will be in same color both in path planning figure and velocity plot. For important positions like initial and goal points, they are all indicated by a black circle, and related data will also be shown nearby.

It should further be mentioned here the main focus of this thesis is to show the capability of the algorithm in arbitrary number of path planning with inter-vehicle avoidance, thus the effort has been put into changing the number of path being planned, and try to model the complex realistic environment with more conflicted points on path. Obstacles here are only kept in a reasonable number and generated randomly in space, but not changed trying to resemble some realistic environment or showing the ability of paths in avoiding obstacles. Number of segments and degree of Bézier curves are also chosen to stay at reasonable level and they will be listed before presenting the simulation result.

5.2 First scenario

This scenario shows the simplified and basic situation when two vehicles are about to collide somewhere on their path. The starting points and end points are marked with black circles, and are set close to make vehicles operate in a relative narrow space, to create conflicted path. The numerical value of parameters used are listed in the table, and the result obtained from the simulation are shown as following:

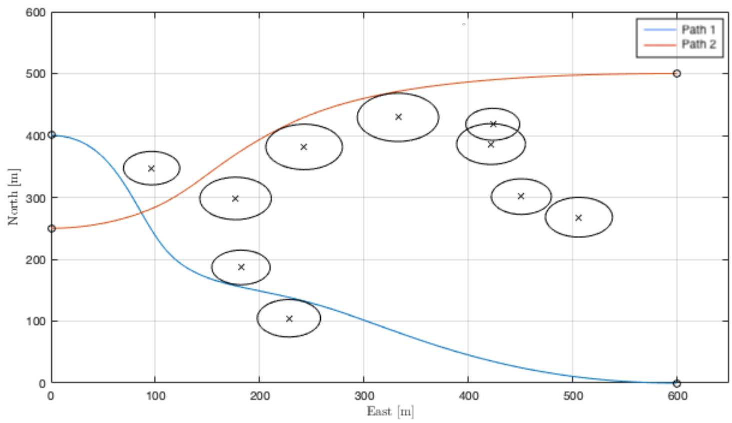


Figure 5.1: Generated paths and obstacles in the first scenario

The paths generated for two vehicle operation are presented in Figure 5.1, and related parameter specifications are in Table 5.1.

Table 5.1: Parameters for path planning in first scenario

Starting points	[400, 0, 0°; 250, 0, 0°]
Destinations	[0, 600, 0°; 500, 600, 0°]
Number and Degree of curve segments	(3, 3)
Minimum turning radius	150
Number of obstacles	10
Size range of obstacles	(25,40)

The velocity profile has been presented in the Figure 5.2. It can been seen that, at the first stage before 10s, the two vehicles are trying to avoid the collisions so one of them is slowing down and the other is speeding up. This also corresponds with the time interval shown in the Table 5.3. After vehicle 1 has passed the danger zone at 10s, and because of the length of the path 1 is longer than the path 2 which can be figured out from the route map in Figure 5.1, vehicle 1 needs to slow down to wait for the vehicle 2, so that they will arrive at the same time. At last, vehicle 1 speeds up to arrive earlier, and vehicle 2 arrives

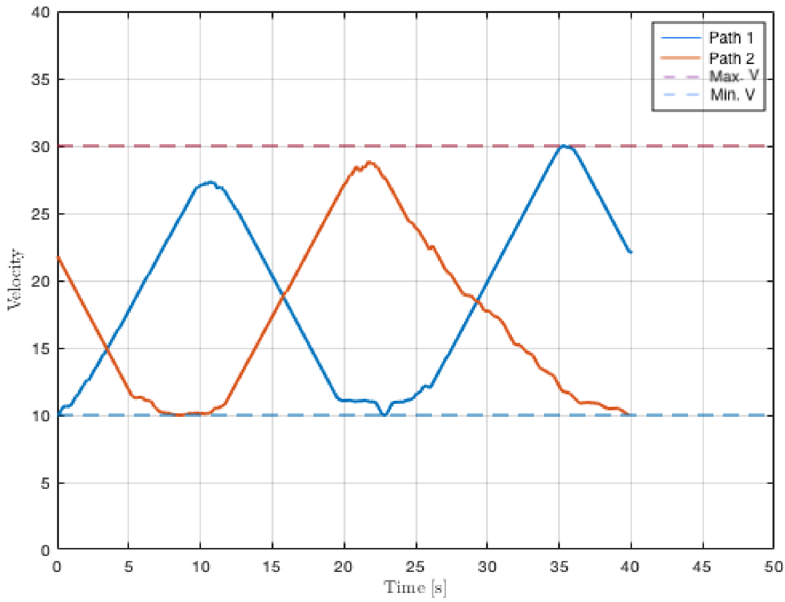


Figure 5.2: Corresponding velocity for each path in the first scenario

at the destination with its lowest speed, due to the penalization term in the cost function. They are arriving simultaneously finally, and temporally deconflicted.

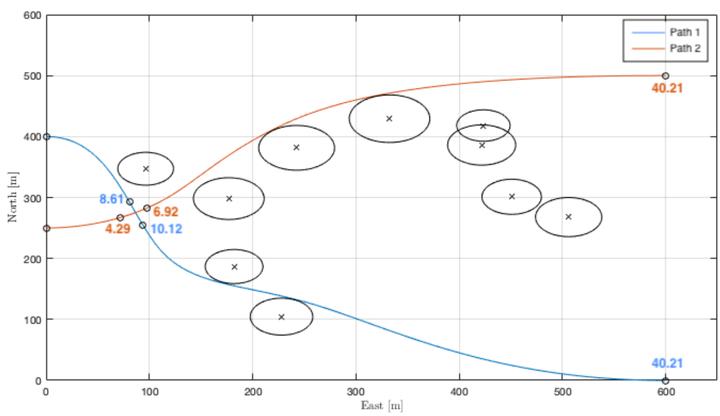


Figure 5.3: Generated temporal deconfilicted paths for the first scenario

In Figure 5.2, two dash lines indicate the maximum and minimum velocity constraints respectively. Purple one represents the maximum value of 30 m/s, and blue one represents the minimum value 10 m/s. The result of temporal deconfliction is presented in Figure 5.3, and related parameters in deconfliction path planning are listed in the Table 5.2:

Table 5.2: Parameters for temporal deconfliction path planning in the first scenario

Number of path	2
Velocity range	(10,30)
Maximum acceleration	2
Danger zone range	30

Table 5.3: Time intervals for each vehicle passing danger zone in the first scenario

	Path 1	Path 2
Intersection point 1	(8.61 , 10.12)	(4.29 , 6.92)

From the time interval of each vehicle passing the collision zone, it could be seen that they are staggered, and vehicle 2 will enter the zone after vehicle 1 has left. The final arriving time for them is 40.21s.

5.3 Second scenario

The situation in this scenario shows the operation executed by three vehicles at the same time. The starting points and end points have been set to create more intersection points. And the intention of this scenario is also to resemble a realistic environment for multiple vehicles operation, and prove the stability and efficiency of this algorithm. The main parameter specifications is shown in the Table 5.4 below:

Table 5.4: Parameters for path planning in second scenario

Starting points	[700, 0, 0°; 450, 0, 0°; 0, 0, 0°]
Destinations	[0, 600, 0°; 500, 600, 0°; 700, 600, 0°]
Number and Degree of curve segments	(3, 3)
Minimum turning radius	150
Number of obstacles	12
Size range of obstacles	(25,40)

The multiple paths generated are shown in Figure 5.4:

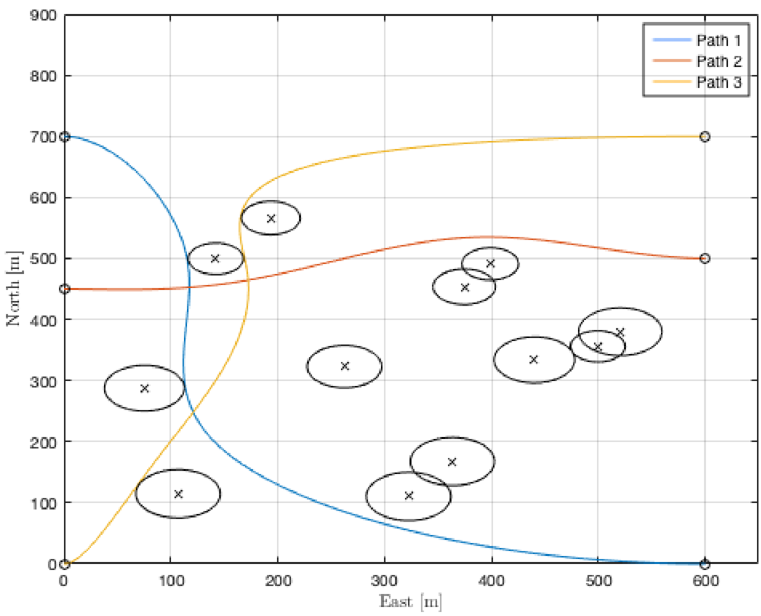


Figure 5.4: Generated paths and obstacles in the second scenario

Further the result of temporal deconfliction, velocity profiles for these three paths are shown in Figure 5.5:

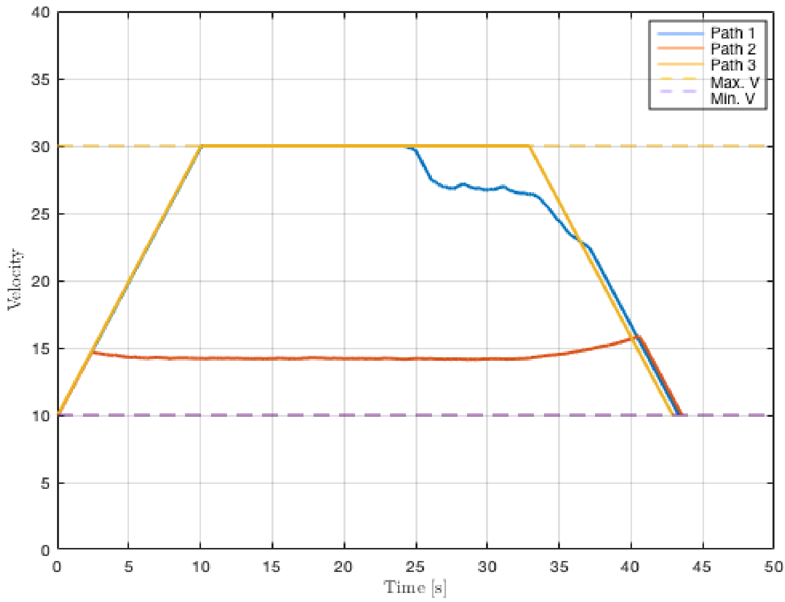


Figure 5.5: Corresponding velocity for each path in the second scenario

Path 2 is apparently the shortest one among these three, thus it keeps running under a relatively low speed, as well as Path 1, starting to slow down before Path 3. And finally they end up with the lowest allowed speed. Note at the end point in Figure 5.5, it seems that the velocity of three vehicles are not stopping at the same point, which might indicate that they are not in fact arriving at the same time. However the reason causing this might be the way velocity is approximated and plotted, as introduced in Chapter 4, section 4.2.3. Corresponding parameter specifications relating temporal deconfliction are shown in Table 5.5:

Table 5.5: Parameters for temporal deconfliction path planning in the second scenario

Number of path	3
Velocity range	(10,30)
Maximum acceleration	2
Danger zone range	30

The three danger zones centered with intersection points are marked in the following Figure 5.6. And the time intervals every vehicle passes the danger zone are listed in the Table 5.6. The result shows that the motion of three vehicles are perfectly deconflicted in the

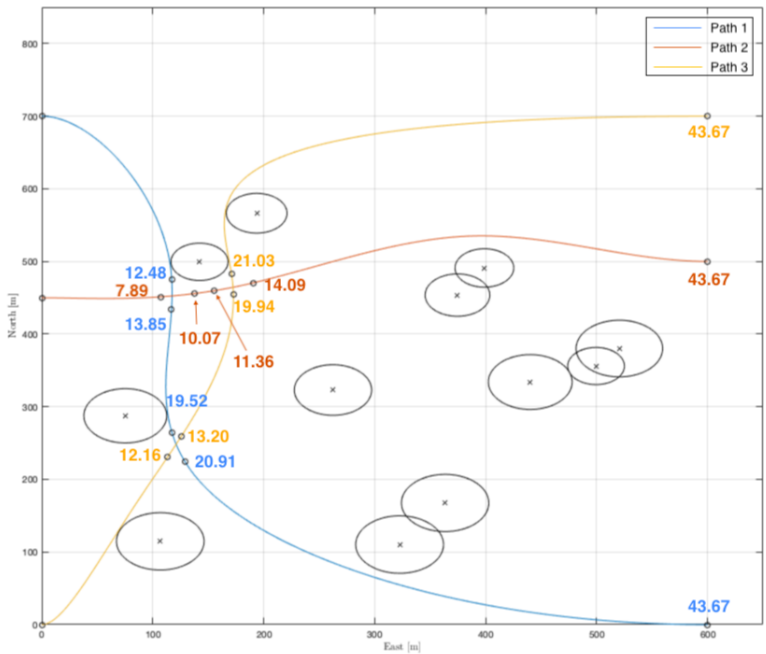


Figure 5.6: Generated temporal deconfilicted paths for the second scenario

crossing section in this scenario, and their arrival at the destination are also synchronized.

Table 5.6: Time intervals for each vehicle passing danger zone in the second scenario

	Path 1	Path 2
Intersection point 1	(12.48 , 13.85)	(7.89 , 10.07)
	Path 1	Path 3
Intersection point 2	(19.52 , 20.91)	(12.16 , 13.20)
	Path 2	Path 3
Intersection point 3	(11.36 , 14.09)	(19.94 , 21.03)

5.4 Third scenario

This scenario simulates four vehicles operate together at the same time, and situation in this simulation just becomes more complicated. Obstacles here are only generated randomly in space resembling static environmental obstacles, and the parameter specifications in path planning are summarized below in Table 5.7.

Table 5.7: Parameters for path planning in third scenario

Starting points	[700, 0, 0°; 550, 0, 0°]
	[300, 0, 0°; 100, 0, 0°]
Destinations	[50, 600, 0°; 200, 600, 0°]
	[450, 600, 0°; 600, 600, 0°]
Number and Degree of curve segments	(3, 3)
Minimum turning radius	150
Number of obstacles	8
Size range of obstacles	(25,40)

The generated paths for four vehicles are presented in the Figure 5.7, and the corresponding speed assignment is in Figure 5.8, along with parameter specifications in Table 5.8.

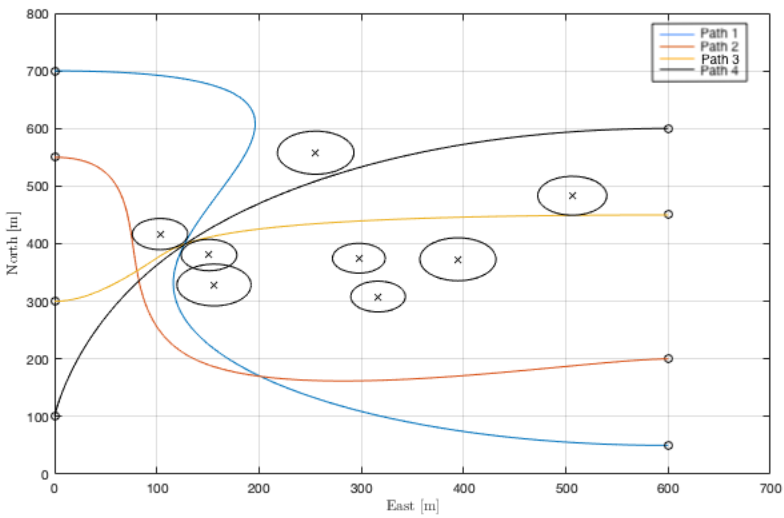


Figure 5.7: Generated paths and obstacles in the third scenario

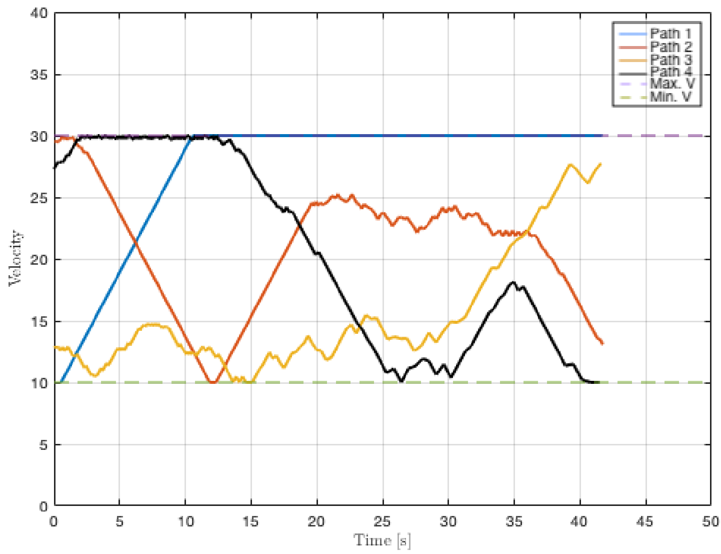


Figure 5.8: Corresponding velocity for each path in the third scenario

Table 5.8: Parameters for temporal deconfliction path planning in the third scenario

Number of path	4
Velocity range	(10,30)
Maximum acceleration	2
Danger zone range	30

Path 3 is the shortest one from the route map, thus it is kept to run under the lowest speed shown in Figure 5.8, while Path 1 as the longest are running full speed ahead to reach target earlier. The result of temporal deconfliction are summarized in the Table 5.9 in detail. Figure 5.9 is the general route map with some points marked with time, and the two area where paths intersect intensively would be presented in Figure 5.10 and Figure 5.11 respectively.

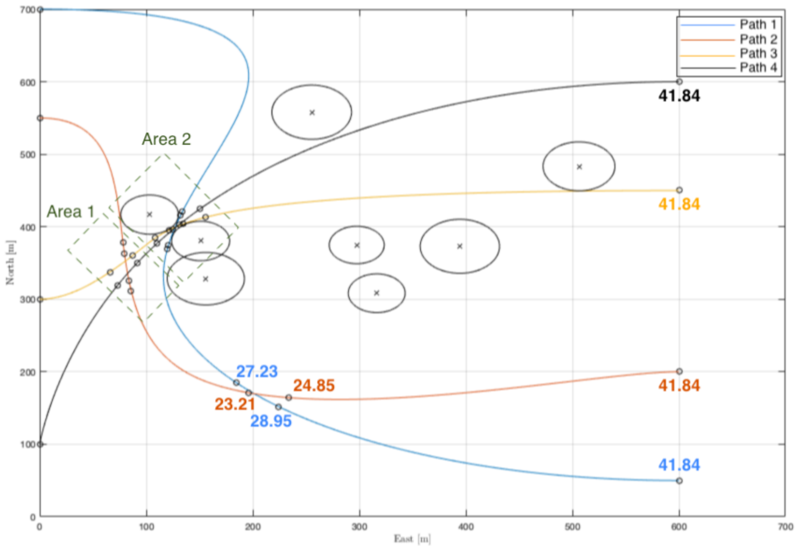


Figure 5.9: Generated temporal deconfilicted paths for the third scenario

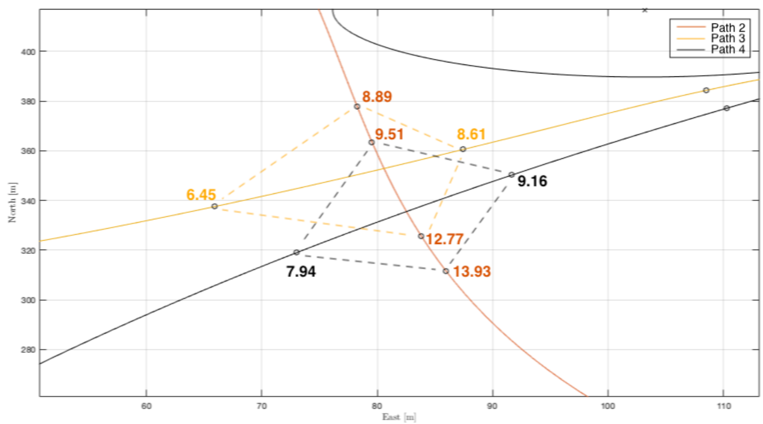


Figure 5.10: Generated temporal deconfilicted paths for Area 1

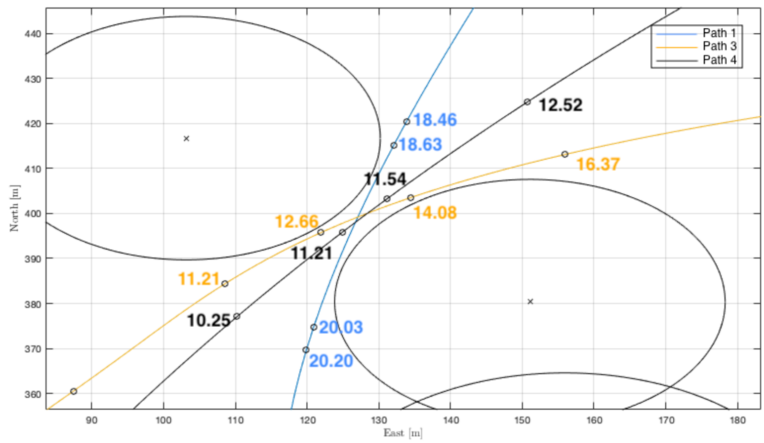


Figure 5.11: Generated temporal deconfilicted paths for Area 2

Table 5.9: Time intervals for each vehicle passing danger zone in the third scenario

	Path 1	Path 2
Intersection point 1	(27.23 , 28.95)	(23.21 , 24.85)
	Path 1	Path 3
Intersection point 2	(18.46 , 20.03)	(11.21 , 14.08)
	Path 1	Path 4
Intersection point 3	(18.63 , 20.20)	(10.25 , 11.54)
	Path 2	Path 3
Intersection point 4	(8.89 , 12.77)	(6.45 , 8.61)
	Path 2	Path 4
Intersection point 5	(9.51 , 13.93)	(7.94 , 9.16)
	Path 3	Path 4
Intersection point 6	(12.66 , 16.37)	(11.21 , 12.52)

Note that in Figure 5.11, Path 1, 3 and 4 almost intersect at the same place and the three intersection points are in fact quite close to each other, making the danger zones overlap greatly. But there are four points on each path, by only looking into the middle two it can be seen that these three vehicle are deconflicted in the place. Also the intersection points shown in Figure 5.11 are point 2, 3 and 6 in Table 5.9, which still could show that they are all successfully staggered. Total arriving time is 41.84s in this situation, while the velocity never exceeding the limitations.

5.5 Fourth scenario

This scenario presents the situation that five vehicles operate in a bounded two dimensional space, resembling a small fleet of the surface vessels executing mission cooperatively. The intention of this situation is also to show the efficiency of the proposed algorithm in path planning for a more complicated situation. The specification for parameters in path planning is summarized at first in Table 5.10, and generated five paths are presented in Figure 5.12.

Table 5.10: Parameters for path planning in fourth scenario

Starting points	[900, 0, 0°; 700, 0, 0°] [550, 0, 0°; 300, 0, 0°] [100, 0, 0°]
Destinations	[50, 600, 0°; 200, 600, 0°] [450, 600, 0°; 600, 600, 0°] [800, 600, 0°]
Number and Degree of curve segments	(3, 3)
Minimum turning radius	150
Number of obstacles	10
Size range of obstacles	(25,40)

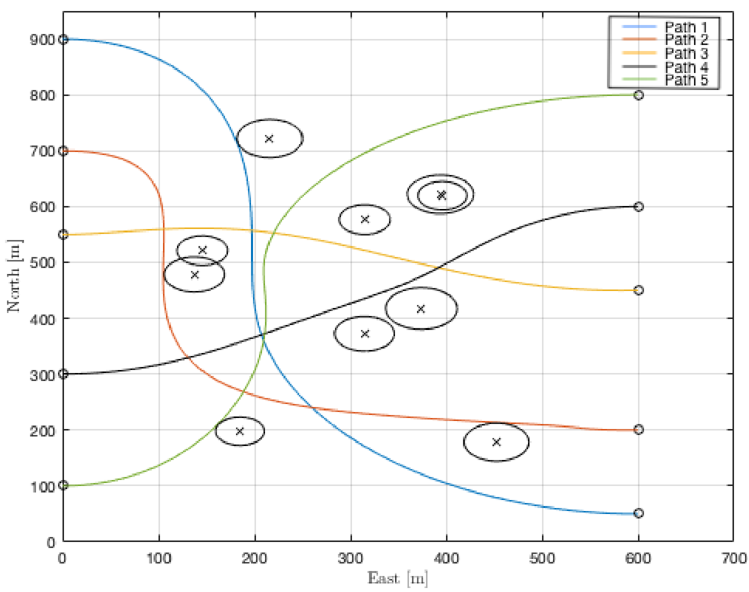


Figure 5.12: Generated paths and obstacles in the fourth scenario

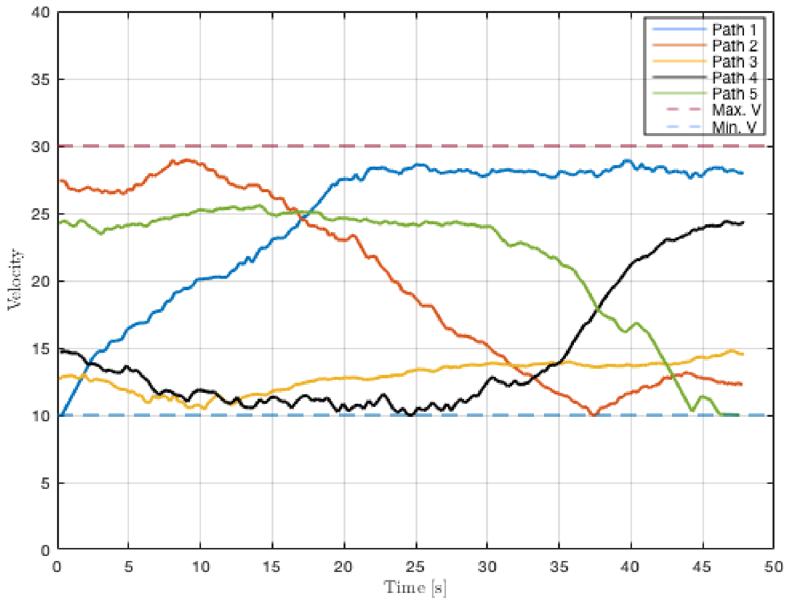


Figure 5.13: Corresponding velocity for each path in the fourth scenario

In the temporal deconfliction, parameters except path number remain unchanged compared to preceding scenarios. And since the paths intersect intensively in some places, they are marked and presented respectively in three figures. The general route map for this scenario is in Figure 5.14.

Table 5.11: Parameters for temporal deconfliction path planning in the fourth scenario

Number of path	5
Velocity range	(10,30)
Maximum acceleration	2
Danger zone range	30

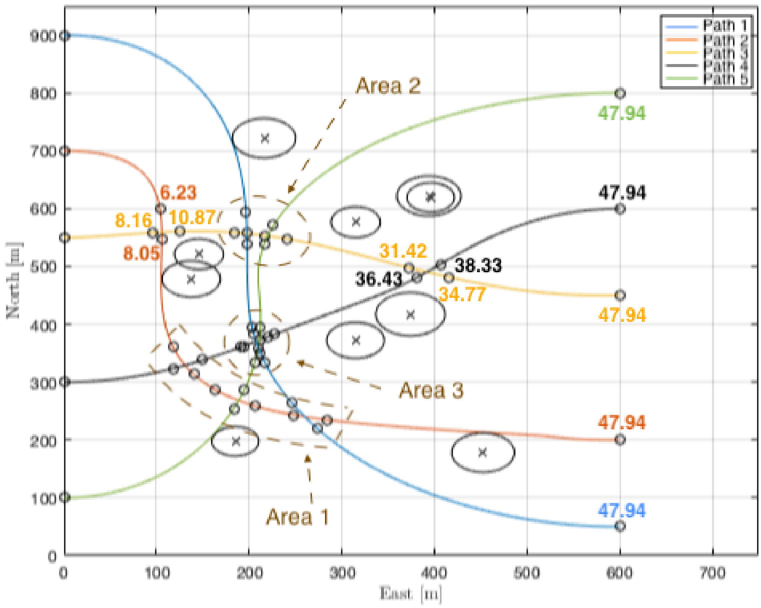


Figure 5.14: Generated temporal deconfilicted paths for the fourth scenario

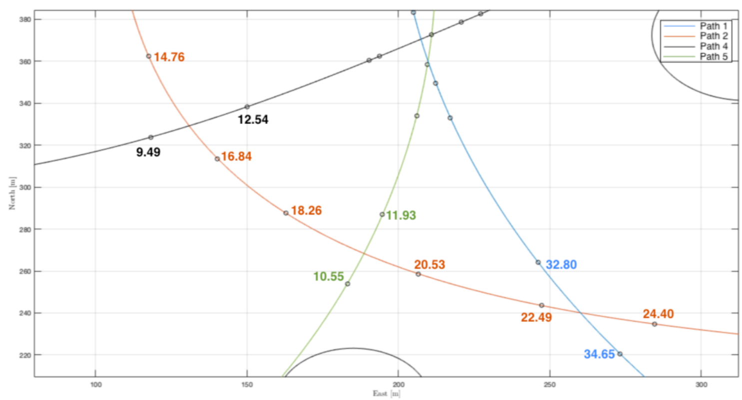


Figure 5.15: Generated temporal deconfilicted paths for Area 1

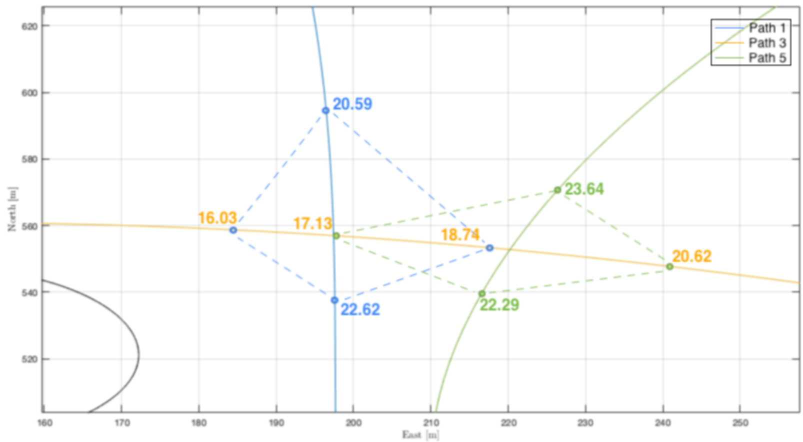


Figure 5.16: Generated temporal deconfilicted paths for Area 2

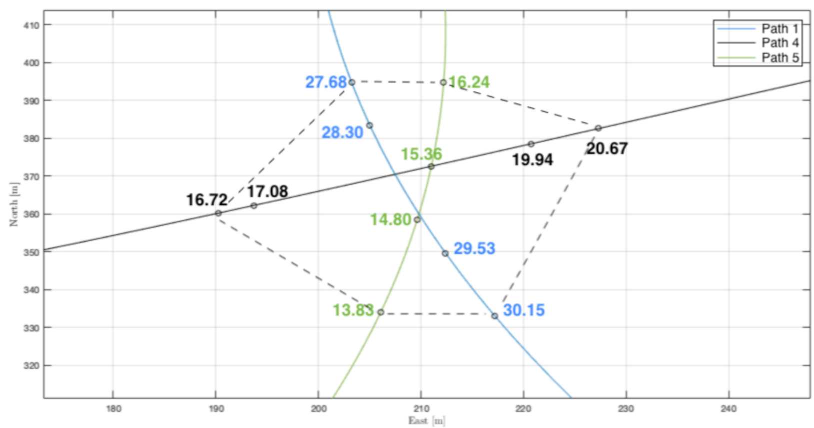


Figure 5.17: Generated temporal deconfilicted paths for Area 3

Table 5.12: Time intervals for each vehicle passing danger zone in the fourth scenario

	Path 1	Path 2
Intersection point 1	(32.80 , 34.65)	(22.49 , 24.40)
	Path 1	Path 3
Intersection point 2	(20.59 , 22.62)	(16.03 , 18.74)
	Path 1	Path 4
Intersection point 3	(27.68 , 29.53)	(17.08 , 19.94)
	Path 1	Path 5
Intersection point 4	(28.30 , 30.15)	(13.83 , 15.36)
	Path 2	Path 3
Intersection point 5	(6.23 , 8.05)	(8.16 , 10.87)
	Path 2	Path 4
Intersection point 6	(14.76 , 16.84)	(9.49 , 12.54)
	Path 2	Path 5
Intersection point 7	(18.26 , 20.53)	(10.55 , 11.93)
	Path 3	Path 4
Intersection point 8	(31.42 , 34.77)	(36.43 , 38.33)
	Path 3	Path 5
Intersection point 9	(17.13 , 20.62)	(22.29 , 23.64)
	Path 4	Path 5
Intersection point 10	(16.72 , 20.67)	(14.80 , 16.24)

In Figure 5.17, the similar situation appears as before that three paths intersect quite close to each other and danger zone overlap together forming a polygon. However by only looking into the points on the edge of the polygon, it could be seen that three vehicles are deconflicted already, as well as shown in Table 5.12 as intersection point 3, 4 and 10. According to the figure above, these five path are successfully temporal deconflicted at last, and achieve simultaneous arrival. The total arriving time is 47.94s, with dynamic constraints satisfied as well.

Chapter 6

Discussion

The results obtained have shown the capability of the propose algorithm in multiple path generation and temporal deconfliction planning. It not only demonstrates that Bézier curve is an efficient tool in multiple path construction satisfying desired constraints, but also shows that it can further serve as the basis for temporal deconfliction work and be combined with constraints like vehicle dynamics, to meet the real need in the application. The four simulation scenarios together implies the algorithm is efficient for arbitrary number of path planning, and the result is reliable. The generated velocity profile could be further implemented in the path following, and if deviation occurs only to change the starting points and some other initial conditions, a new velocity plan can be generated again to modify and compensate for the deviation. Since the main concern for this thesis is to show the ability of the proposed algorithm in temporal deconfliction path planning with inter-vehicle collision avoidance, but not the path generation avoiding static obstacles, thus obstacles here in simulation are just generated randomly in space with a reasonable number, and are not used to resemble some realistic environments like port or coastline. Attention has also not been put into changing other parameters in path generation stage, thus they just stay almost unchanged.

However, during the simulation an important observation is that, although the path generation and speed optimization has been decoupled and separated into two stages, they are not completely independent and the parameter specifications for the path generation is in fact significant for the result of the temporal deconfliction work. It is easy to understand since the path generation serves as the basis for the temporal deconfliction after all, and its result data provides a foundation for the later work. When determining the degree of the Bézier curve and the number of segments, theoretically the choice should be making them larger, thus the curve will be divided into more intervals such that the velocity should be more smooth. In fact when running the simulation increasing the segment number blindly not only causes the heavy computation loads, the solution might fail to converge more easier in the end. And due to the fact that the higher order Bézier curves are easier to be numerically unstable in the result, it will affect the path generation greatly as well. It is the priority that insuring the multiple paths are properly constructed since it provides the basic for the further deconfliction step. To guarantee the result of the temporal deconfliction, there is also another important factor to be considered, the finite step size of the optimization solver. To explain this we might need to take a step back, to the choice of the optimization algorithm, Sequential Quadratic Programming.

This optimization method is chosen basing on the way constraints are formulated, and also investigating the implementation of other provided solvers. In this thesis, the constraints on temporal deconfliction includes a logic relationship "or" but not "and", although linear but it makes this formula can not be represented in the linear inequality matrices since `fmincon` requires a data form input. Also constraints for accelerations are nonlinear, making this problem a nonlinear constrained one. Optimization solver `fmincon` in fact provides five algorithm options in total: Interior-point(default), Trust-region-reflective, SQP, SQP-legacy, and active-set algorithm. Trust region and active set are not suitable due to the way problem constructed, and interior point fails every time to obtain the minimized objective function, although very fast. Sequential quadratic programming is a powerful and reliable method among them, however it does show an tendency to rely on the specification of finite different step size. It is a gradient based algorithm after all and the gradients are obtained by the use of finite difference. The value of the finite difference step size can be changed manually by user, and it needs to be tuned for different problems to produce satisfactory result and it is very essential. Basing on the performance of the numerical simulation a larger value of step size usually adapts to more situations, but not always produce the best result, and a smaller value is more likely to have a better one if only feasible solution could be found, and it is relatively sensitive to the change of situations. This actually is kind of similar to choosing the number of Bézier curve's segment. And this phenomenon is apparent when performing the simulation for different number of paths. For complicated situations with more paths, like the fourth simulation scenario presented, a larger step size is easier to be feasible and usually faster in computation, but sometimes ends up with a unsatisfactory result, and in this situation trying on a smaller value is necessary. If a smaller step size could yield a feasible solution it will more likely be better. This however might be a bit rough to be as a conclusion, and in fact the relationship inside between step size and finding a feasible solution and their influence should address more attention and needs more study further.

Another factor affecting the result is the choice of initial guess. When running the simulation changing the initial guess would possibly result in finding a different solution, and sometimes it could even be better, although both of them are optimal solutions provided by the optimization solver `fmincon`. And in different situations this phenomenon often can be observed. It is not hard to understand the reason since the purpose to make the initial guess is to provide a starting point to SQP algorithm for the later solution searching. Step size affects the searching gradient, and initial guess decide where it should start first. These two factors combined together could produce a good result if set properly. These observations are significant and unignorable during the process of obtaining good result, and this process of tuning parameters leads to the assumption that the result it obtained might only be a local optimal solution. And it is hard to verify whether it is global, or which one is in fact global.

However, the ability of yielding satisfactory result has been proved through the numerical simulations, demonstrating the algorithm can be capable of doing that only through determining a proper value of step size and initial guess. This actually indicates that this algorithm might only not be the best suitable one for this problem, or effort should be further put into finding a new optimization solver that is better than the current one. This disallows the replanning and reaction to the environment if deviation occurs during the operation. And for SQP algorithms, every time before running the simulation, whether a feasible solution exists is unknown beforehand and can not be located if exists. It depends on the initial guess as well, as mentioned before, thus the result obtained might not be a global solution, only the local one. This may be the main cause that the finding of feasible solution is not always successful, and tuning the value is necessary. However to discuss this topic, we again need to go back to the choice of algorithms. Currently the way of formulating the constraints mainly determines this problem to be a nonlinear constrained one. But recalling the optimization algorithms categories, the decision can also be made basing on how the approximation of the problem can be done. Linearization and simplification of the nonlinear problem is an important and necessary procedure to lower the computation loads and improve the efficiency. Some simplifications has been mention before, and the way of achieving it can be further improved and worked on, even without changing the solver this is going to have an influence on the result.

Like discussed before, the efficiency of the algorithm obtaining satisfactory result determines whether it can achieve online modification and replanning in real application. However, current implementation might disallow the use like this. A better suitable algorithm needs to be explored in the future work, as well as the approximation and formulation this problem constructed. In fact the computational loads currently needs also to be further relieved, since the computation of the result usually takes more times than producing a fast reaction to the change of the environment. All of these discussed are the work that further improvement should be involved.

Chapter 7

Conclusion and future work

7.1 Conclusion

In this thesis, a temporal deconflicted path planning algorithm for multiple vehicles has been developed, using Bézier curve as the basic path generation tool. It mainly deals with the problem of inter-vehicle collision avoidance when multiple vehicles are operating in the same space cooperatively and closely. Basing on the single path generation algorithm, spatial multiple path planning and deconfliction work have been decoupled and formulated separately. Basing on the concept of temporal deconfliction and augmenting the time coordinate, the mathematical formulation of the deconfliction property is proposed as well as the vehicle dynamic restrictions and simultaneous arrival requirement. Further constructing the constraints and objectives within the optimization framework and utilizing proper optimization algorithm, the problem is solved and a temporal deconflicted solution with arrival synchronized and operation time minimized is obtained.

This algorithm is able to plan path for arbitrary number of vehicles, and the physical dynamic restrictions are taken into consideration during the whole process. The vehicle motion near the crossing points on the path are specially taken care of and staggered, ensuring that two vehicles would never appear at the intersection point and its close vicinity at the same time. Moreover the operation time is minimized and final arriving time is constrained as the same and quantitatively presented in the numerical simulation.

The result has shown that the temporal deconfliction work can be implemented basing on the Bézier curves and it is efficient for arbitrary number of vehicles, while taking the vehicle dynamic constraints into consideration. The proposed algorithm is able to solve a multiple path planning problem, and restrict the length of the operation duration by constraining it as simultaneous. This can also be changed such as specifying each vehicle's operating time length respectively, and thus provides more versatility for real practice. Although there are some problems existing in the obtainment of the satisfactory result, for which the reason and improving methods have been discussed before, it can be concluded that this temporal deconfliction method for multiple path planning is efficient and useful. The result of the numerical simulations presented could demonstrate this, and also indicate the efficiency and potential of this method to be improved in the further work.

7.2 Future work

Current problems

In the current implementations, there are some problems existing in the persistently obtainment of the satisfactory result for the proposed method. Right now through changing the step size and the initial guess a good result can be obtained, and in fact the step size seems to be a bit more influential on the result than the initial guess, but this still needs further research to be confirmed and precisely defined. And the algorithm may probably need a better suitable optimization solver, or improving the simplification and formulations for the nonlinear terms. More different optimization solvers or algorithms need to be evaluated and compared in the future, and also the efficiency and computation time. It is an important aspect since it is the main factor determining whether the algorithm is useful in real time application and suitable for the online modification and path replanning. Thus the problem needs to be solved not only satisfying but as efficient as possible, and that requires the construction of the constraints to be more simple.

Further the algorithm can include more vehicle dynamic constraints in the future. So far the surface vehicle is only taken as a moving particle and only the acceleration and velocity are taken into consideration, but for real ship model, for example moving in the North-east plane, angular velocity and heading are of equivalent importance compared to these, and need to be taken care of as well. This thesis is presenting a basic ship mathematical model, which however in future in order to achieve real application, more detailed aspects is needed to be completed.

Future work

The main focus for this thesis is the temporal deconfliction for multiple vehicle, however in the future it can be expanded to a three dimensional space path planning problem. It can be achieved by augmenting another z-coordinate, thus allowing the description of the motion in the depth. This would make the algorithm change from surface vehicle to underwater robotics path planning, and become more versatile. However the requirement on the efficiency and reliability of optimization solver is going to be higher as well.

Also, the spatial deconfliction is an important issue worthy to be dealt with. Although the time deconfliction can tackle with more problems, in some open and wide areas especially in three dimensional space, spatial deconfliction will provide more possibilities and it is also available to be combined with time coordinate and assign speed, achieving simultaneous arrival for example. In real world application spatial deconfliction is easier to be achieved and since each vehicle is operating independently, it would have lower accuracy requirements for the path following task.

Finally if the computational time issue is solved, and the obtainment of satisfactory result can be achieved in only one attempt, vehicle path following and control can also be considered as a further direction of work. The algorithm will allow online modification and compensation for deviations, as well as path replanning problem if the goal point is not a single destination but series of waypoints to be passed in sequence. A real ship model can be included in this topic, and used in simulations or in a laboratory as real model test.

Bibliography

Barnier, N., Allignol, C., 2012. Trajectory deconfliction with constraint programming. The Knowledge Engineering Review 27 (3), 291–307.

Bitar, G., Breivik, M., Lekkas, A. M., 2018. Energy-optimized path planning for autonomous ferries. IFAC-PapersOnLine 51 (29), 389–394.

Breivik, M., Fossen, T. I., 2009. Guidance laws for autonomous underwater vehicles. In: Underwater vehicles. IntechOpen.

Carbone, G., Gomez-Bravo, F., 2015. Motion and operation planning of robotic systems. Springer International Publishing. Switzerland. DOI 10, 978–3.

Choi, J.-w., Curry, R., Elkaim, G., 2008. Path planning based on bézier curve for autonomous ground vehicles. In: Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008. IEEE, pp. 158–166.

Forsmo, E. J., 2012. Optimal path planning for unmanned aerial systems. Master's thesis, Institutt for teknisk kybernetikk.

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

Ghabcheloo, R., Aguiar, A. P., Pascoal, A., Silvestre, C., Kaminer, I., Hespanha, J., 2009a. Coordinated path-following in the presence of communication losses and time delays. SIAM journal on control and optimization 48 (1), 234–265.

Ghabcheloo, R., Kaminer, I., Aguiar, A. P., Pascoal, A., 2009b. A general framework for multiple vehicle time-coordinated path following control. In: 2009 American Control Conference. IEEE, pp. 3071–3076.

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

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

Hooke, R., Jeeves, T. A., 1961. "direct search"solution of numerical and statistical problems. Journal of the ACM (JACM) 8 (2), 212–229.

Ingersoll, B. T., Ingersoll, J. K., DeFranco, P., Ning, A., 2016. Uav path-planning using bezier curves and a receding horizon approach. In: AIAA Modeling and Simulation Technologies Conference. p. 3675.

Kamermans, M. P., 2011. A Primer on Bézier curves. `https://pomax.github.io/bezierinfo/en-GB/`, [A free, online book].

Kaminer, I., Yakimenko, O., Dobrokhodov, V., Pascoal, A., Hovakimyan, N., Patel, V., Cao, C., Young, A., 2007. Coordinated path following for time-critical missions of multiple uavs via l1 adaptive output feedback controllers. In: AIAA Guidance, Navigation and Control Conference and Exhibit. p. 6409.

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

Lande, S. V., 2018. Path planning for marine vehicles using bézier curves. Master's thesis, NTNU.

Lekkas, A. M., 2014. Guidance and path-planning systems for autonomous vehicles.

Lizarraga, M., Elkaim, G., 2008. Spatially deconflicted path generation for multiple uavs in a bounded airspace. In: 2008 IEEE/ION Position, Location and Navigation Symposium. IEEE, pp. 1213–1218.

Nocedal, J., Wright, S., 2006. Numerical optimization. Springer Science & Business Media.

Novin, G., 2016. A History of Graphic Design. `http://guity-novin.blogspot.com/2013/04/chapter-66-bezier-curves-for-digital.html`, [Online Textbook].

Pannequin, J., Bayen, A., Mitchell, I., Chung, H., Sastry, S., 2007. Multiple aircraft deconflicted path planning with weather avoidance constraints. In: AIAA Guidance, Navigation and Control Conference and Exhibit. p. 6588.

Roald, A. L., 2015. Path planning for vehicle motion control using numerical optimization methods. Master's thesis, NTNU.

Saccon, A., Aguiar, A. P., Häusler, A. J., Hauser, J., Pascoal, A. M., 2012. Constrained motion planning for multiple vehicles on se (3). In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC). IEEE, pp. 5637–5642.

Scerri, P., Owens, S., Yu, B., Sycara, K., 2007. A decentralized approach to space deconfliction. In: 2007 10th international conference on information fusion. IEEE, pp. 1–8.

Sederberg, T. W., 2012. Computer aided geometric design.

Svestka, P., Overmars, M. H., 1996. Coordinated path planning for multiple robots. Vol. 1996. Utrecht University: Information and Computing Sciences.

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

NTNU
Kunnskap for en bedre verden