

CLOTHOID SPLINE BASED PATH PLANNING

CHEN YONG

2017



**NANYANG
TECHNOLOGICAL
UNIVERSITY**

CLOTHOID SPLINE BASED PATH PLANNING

**CHEN YONG
SCHOOL OF MECHANICAL AND AEROSPACE
ENGINEERING
2017**

CLOTHOID SPLINE BASED PATH PLANNING

CHEN YONG

CHEN YONG

School of Mechanical and Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2017

Acknowledgments

I would like to express my sincere thanks and appreciation to my supervisor, Prof. Yiyu Cai for his invaluable guidance and continuous support in this research. His knowledge and suggestions benefit me a lot, while his patience, encouragement, and attitude help me to overcome the difficulties encountered in my life and research. I also would like to thank my co-supervisor, Prof. Daniel Thalmann, for his suggestions and support in my research from various perspectives.

Besides my supervisors, I would like to give my special thanks to Prof. Jianmin Zheng for his insightful commentary and generous help in this research.

I want to thank Nanyang Technological University and the School of Mechanical and Aerospace Engineering for granting me this study opportunity. Besides, great thanks also goes to the Institute for Media Innovation and the Virtual Reality and Visualization Lab which provide excellent environments for my research.

I thank my current colleagues, Lihui Huang, Dr. Panpan Cai, and Dr. Yuzhe Zhang for their friendship and support. I feel extremely fortunate to work with them and former members in this lab including Dr. Yuewen Ma, Dr. Qing Li, Dr. Xiaoqun Wu, Bianyue Yang, and Nanxu Wang. I want to thank all of them for their cooperation and collegial guidance on different projects.

Finally, I want to thank and express my deepest love to my parents and sister, for their constant love and encouragement.

Contents

Acknowledgments	i
List of Figures	vi
List of Abbreviations	xiii
Abstract	1
1 Introduction	3
1.1 Motivation	3
1.2 Organization	8
2 Literature Review	11
2.1 Curvature Continuous Path Smoothing	12
2.1.1 Planar Curves for Path Smoothing	12
2.1.2 Computation and Approximation of Clothoid Curves	16
2.2 Nonholonomic Planning	24
2.2.1 Probabilistic Roadmaps	28
2.2.2 Rapidly Exploring Random Tree	32
3 Vehicle Model and Local Path Analysis	35
3.1 Generalization of Kinematic Model of Wheeled Mobile Robots	35
3.1.1 Mobility Analysis	35
3.1.2 Kinematic Models	40
3.1.3 Generic Posture Kinematic Model	46
3.1.4 Model of the Wheeled Mobile Robot	48
3.2 Local Path Configurations	51
3.2.1 General Case	52

3.2.2	Other Cases	55
3.2.3	Path Smoothing	59
3.3	Swept Path Analysis	69
3.3.1	Turning Templates	69
3.3.2	Mathematical Modeling	72
4	Approximation of Clothoid Using Bézier Curves for Path Planning	83
4.1	Elementary Clothoid and Bézier Curve	84
4.1.1	Elementary Clothoid	84
4.1.2	Quintic Bézier Curve	85
4.1.3	Approximation of the Elementary Clothoid	86
4.2	Basic Clothoid Approximation	91
4.2.1	Basic Clothoid	91
4.2.2	Arc Length Computation	92
4.2.3	Bézier Approximation and Error Analysis	95
4.3	General Clothoid Approximation	98
4.4	Results and Error Analysis	101
4.4.1	Offline Approximation Algorithms	101
4.4.2	Online Approximation	105
5	Clothoid Spline RRT Path Planner Based on Bézier Approximation	111
5.1	Local Path Primitives	112
5.1.1	Path Parameterization of CASE 1	113
5.1.2	Path Parameterization of CASE 2	115
5.2	Clothoid Spline Based RRT	116
5.2.1	Reachability Tree	116
5.2.2	Integration of Clothoid Spline and RRT	117
5.2.3	Path Planning in a Maze environment	120
5.3	Integration of Clothoid and Roadmap	123
5.3.1	Preprocessing of the Environment Data	123

5.3.2	Roadmap Representation	125
5.4	Applications and Simulations	126
6	Conclusions	129
6.1	Contributions	129
6.2	Future Work	133
Appendix A		135
A.1	G^3 Approximation	135
A.2	Elementary Bézier Clothoid	139
References		141

List of Figures

1.1	(a) Mobile crane with two axles. (b) Side view and dimensions (unit: mm).	4
1.2	Robot of different geometric shapes: (a) No solution found for a plate shape. (b) Solution is found for a rectangular shape.	7
1.3	Local path planning for a mobile crane.	8
2.1	Clothoid defined on $[-10, 10]$: (a) $\mathbf{p}_{\text{clo}}(s)$. (b) $\kappa(s)$	16
2.2	Clothoid approximation through Taylor expansion as $N = 1, 2, \dots, 5$: (a) Approximation result. (b) Approximation Error.	18
2.3	Clothoid approximation error via s-series: (a) $\sigma = 1, k = 1, 2, 3, 4, 5$. (b) $k = 2, \sigma = 0.01, 0.1, 1, 10, 100$	20
2.4	Clothoid approximation by an 11-th order RBC: (a) Approximation result. (b) Approximation Error.	22
2.5	Clothoid approximation via circular interpolation: (a) Approximation result. (b) Approximation Error.	23
3.1	Wheel types: (a) Steerable and fixed wheel model. (b) Castor wheel model.	37
3.2	(a) A FWS vehicle model with Ackermann steering system. (b) Equivalent bicycle model.	40
3.3	Multi-axle vehicle: (a) Model with only one non-steerable axle. (b) Equivalent bicycle model.	43
3.4	Multi-axle vehicle: (a) Model with multiple non-steerable axles. (b) Equivalent slip-free bicycle model.	44

3.5	4 wheel-steering vehicle: (a) Positive configuration. (b) Negative configuration.	45
3.6	Kinematic model of a four-wheeled mobile robot: (a) The bicycle model representation. (b) Velocities of front wheel P and rear wheel Q .	49
3.7	Local path (CASE 1): (a) Symmetric configuration. (b) Feasible local spline paths of the same deflection angle.	53
3.8	Local path: (a) CASE 2. (b) CASE 3. (c) CASE 4.	56
3.9	Path smoothing example: (a) B-spline smoothing. (b) Proposed path smoothing. (c) Curvature profile of B-spline path. (d) Curvature profile of proposed path.	60
3.10	Path smoothing example: (a) Proposed spline path. (b) Curvature profile. (c) Sharpness profile.	62
3.11	Collision avoidance around a corner: (a) CASE 1. (b) CASE 2.	63
3.12	$w_v = 1$: (a) upper bound of d_{SAFE} . (b) Lower bound of σ_{SAFE} .	65
3.13	Computation of r_{SAFE} when $w_v = 1$: (a) r_{\max} as $\sigma_{\max} = 0.1, 1.0$, and 10.0. (b) r_{\max} and r_{\min} as $\sigma_{\max} = 1.0$.	66
3.14	Collision avoidance around a corner: (a) CASE 3. (b) CASE 1 by modifying waypoints.	67
3.15	(a) A standard turning template for a single-unit truck. (b) A turning template based on the EU Standard.	70
3.16	Turning maneuver following a local planner: (a) FWD model. (b) RWD model.	71
3.17	Rear wheel path following a straight line.	74
3.18	Rear wheel path following a circular arc: (a) $r = 2.0, l_w = 1.0$. (b) $r = 1.0, l_w = 1.0$. (c) $r = 0.8, l_w = 1.0$. (d) $r = 0.4, l_w = 1.0$.	76
3.19	Rear wheel path following a clothoid: (a) $\mathbf{p}(t)$ starts at $(0, 0)^T$. (b) $\mathbf{p}(t)$ starts at $(\pi C_f(1), \pi S_f(1))^T$.	78
3.20	Front wheel path of an RWD model: (a) Following a circular arc of radius $r = 2$. (b) Following a clothoid of sharpness $\sigma = \frac{1}{\pi}$.	80
3.21	Computation of the swept path: (a) FWD model. (b) RWD model.	81

4.1	$\epsilon_\kappa(\beta_1, \gamma_1)$: (a) $l = 0.1, \theta = \frac{\pi}{4}$. (b) $l = 0.1, \theta = \frac{\pi}{2}$. (c) $l = 1, \theta = \frac{\pi}{4}$. (d) $l = 1, \theta = \frac{\pi}{2}$. (e) $l = 10, \theta = \frac{\pi}{4}$. (f) $l = 10, \theta = \frac{\pi}{2}$.	89
4.2	Optimal β_1 and γ_1 values in region $[0, 10] \times [-0.06, 0]$: (a) $\beta_1(l, k, \theta = \frac{\pi}{4})$. (b) $\gamma_1(l, k, \theta = \frac{\pi}{4})$. (c) $\beta_1(l, k, \theta = \frac{\pi}{2})$. (d) $\gamma_1(l, k, \theta = \frac{\pi}{2})$.	90
4.3	Divergence problem is shown by setting $\theta = \frac{\pi}{3}, l = 1$ and $k = -1.1 k_d$: (a) $\beta_2(k)$ (blue solid), $\gamma_2(k)$ (orange solid), and linear interpolations (green dashed). (b) Clothoid (red dashed) and its Bézier approximation (blue solid) using proposed method.	91
4.4	Basic clothoid approximation: (a) Individual Bézier curve $\mathbf{B}_{\varepsilon,i}$ ($i = 1, 2, \dots, 7$) are computed as primitives with setting $\theta_{\max} = \frac{\pi}{2}, N = 7$, and $\sigma_c = 1$. It shows both the original segments (dashed) and modified ones after adjusting arc length (solid) by setting $\xi_u = 0.5$. (b) The basic clothoid $\mathbf{F}_c(s)$ (dashed) and its Bézier approximation (solid) derived based on adjusted segments with the curvature magnitudes of the Bézier curves (shaded). (c) $\epsilon_\kappa(s)$ before adjusting segment length. (d) $\epsilon_\kappa(s)$ after adjusting segment length.	96
4.5	Bézier approximations (solid) of basic clothoids (red dashed) and curvature error functions: (a) $\theta_{\max} = \frac{\pi}{2}$ (blue) and $\theta_{\max} = \frac{\pi}{4}$ (orange) as $\sigma_c = 1$. (b) $\sigma_c = 0.5$ (blue) and $\sigma_c = 5$ (orange) as $\theta_{\max} = \frac{\pi}{3}$. The corresponding error function $\epsilon_\kappa(s)$ of each approximation: (c) Top: $\theta_{\max} = \frac{\pi}{2}, \sigma_c = 1$; Bottom: $\theta_{\max} = \frac{\pi}{4}, \sigma_c = 1$; (d) Top: $\sigma_c = 5, \theta_{\max} = \frac{\pi}{3}$; Bottom: $\sigma_c = 0.5, \theta_{\max} = \frac{\pi}{3}$.	97
4.6	Curvature error as a function of $\theta_{\max} \in (0, \frac{\pi}{2}], \sigma_c \in (0, 10]$ and $l_{\max,i} \in (0, l_\tau]$: all the values of $\epsilon_\kappa(\theta_{\max}, \sigma_c, l_{\max,i})$ are indicated by colors and within range $(0, 0.01)$.	98
4.7	Curvature error with regard to θ_{\max} and σ_c : (a) $\epsilon_\kappa(\theta_{\max}, \sigma_c)$ as $\theta_{\max} \in (0, \frac{\pi}{2}], \sigma_c \in (0, 10]$ and $l_{\max,i} = 0.05$. (b) Top: $\epsilon_\kappa(\theta_{\max})$ as $\sigma_c = 5.10$; Bottom: $\epsilon_\kappa(\sigma_c)$ as $\theta_{\max} = 0.77$.	98
4.8	Different approximations (blue solid) of $\mathbf{F}(s) = (C_f(s), S_f(s))^T$ (red dashed) of unit length: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.	102

4.9	$\epsilon_\kappa(s)$ of unit length clothoid approximations: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.	103
4.10	Different approximations (blue solid) of $\mathbf{F}(s)$ (red dashed) with $s \in [0, 2]$: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.	104
4.11	$\epsilon_\kappa(s)$ of non-unit length clothoid approximations: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.	105
4.12	Proposed G^3 approximations (blue solid) of $\mathbf{F}(s)$ (red dashed) with prescribed parameters: (a) $\sigma = 1.0$, $l = 6.0$, $\kappa_0 = -1.0$, $\theta_0 = 0.0$, $(x_0, y_0) = (-2.0, 0.5)$. (b) $\sigma = -2.0$, $l = 6.0$, $\kappa_0 = 5.0$, $\theta_0 = \frac{\pi}{4}$, $(x_0, y_0) = (-2.0, 0.5)$	105
4.13	Position error ϵ_p of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.	107
4.14	Orientation error ϵ_θ of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.	107
4.15	Curvature error ϵ_κ of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.	108
4.16	Position error ϵ_p with respect to clothoid scaling.	109
5.1	Parameterize path primitive: CASE 1.	114
5.2	Parameterize path primitive: CASE 2.	116
5.3	Reachability tree: (a) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{\pi}{3}]$. (b) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{2\pi}{3}]$. (c) $\kappa_{\max} = 5.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{\pi}{3}]$. (d) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 40.0$, $\delta \in [0, \frac{\pi}{3}]$	118
5.4	Clothoid spline based RRT in a free 3D space.	121
5.5	Clothoid spline based RRT in a cluttered 3D space (a) Tree expansion results. (b) Solution path.	122
5.6	Bézier spline based RRT in a cluttered 3D space (a) Tree expansion results. (b) Solution path.	122

5.7	Individual scans of the car park environment: each dataset contains up to 4.3×10^7 points.	124
5.8	Road information extraction (courtesy of Dr. Xiaoqun Wu): (a) Point cloud representation of the car park environment in grayscale. (b) Road segmented from the PCD dataset. (c) Road mesh generated from the PCD model.	125
5.9	Roadmap processing: (a) Thinning. (b) Pruning. (c) Simplification.	126
5.10	Local path planning.	127
5.11	Path planning in a cluttered space.	127

List of Abbreviations

4WS	Four-Wheel Steering
BVH	Bounding Volume Hierarchy
BVP	Boundary Value Problem
CC	Continuous-Curvature
CS	Configuration Space
DOF	Degree of Freedom
FWD	Front-Wheel Drive
FWS	Front-Wheel Steering
GCS	Generalised Cornu Spiral
GNAT	Geometric Near-neighbor Access Tree
GVM	Gross Vehicle Mass
HGV	Heavy Goods Vehicle
ICR	Instantaneous Center of Rotation
KPIECE	Kinodynamic Planning by Interior-Exterior Cell Exploration
LGV	Large Goods Vehicle
LTCCS	Large Truck Crash Causation Study
PCD	Point Cloud Data
PH	Pythagorean Hodograph
PMP	Pontryagin's Maximum Principle
PRM	Probabilistic Roadmap
RBC	Rational Bézier Curve
RDT	Rapidly Exploring Dense Tree
RRT	Rapidly Exploring Random Tree
RWD	Rear-Wheel Drive
RWS	Rear-Wheel Steering
SBP	Sampling Based Planning
WMR	Wheeled Mobile Robot

Abstract

Path planning for long vehicles plays an important role in safe and efficient operation, especially for those with very large lateral and longitudinal sizes driving in narrow spaces. To address this issue, we start with generalizing a kinematic bicycle model to represent different types of vehicles, based on which a path planning framework composed of a local path planner and a global path planner is developed. During the local planning phase, feasible path primitives are generated subject to kinematic constraints of the curvature maxima and the angular acceleration upper bound. By designing a family of path primitives with clothoid arcs involved which can be efficiently configured according to the given environment and reference points in physical space referred to as waypoints, we are able to obtain a near-optimal solution without numerically solving the two point boundary value problem to generate a feasible path connecting two states under differential constraints. During the global planning phase, the local path planner can be conveniently integrated with sampling based frameworks like the rapidly exploring random tree, which complies with the kinematic constraints when sampling the new states and expanding towards unexplored space. Meanwhile, an efficient clothoid approximation method is presented based on the minimization of curvature profile difference. Quintic Bézier curves are used to ensure higher order geometric continuity with smaller approximation error, and the data is stored in a lookup table to reduce the computing time. The developed methodology is novel and effective, which is capable of addressing a class of nonholonomic planning problems successfully subject to curvature and angular acceleration constraints.

Chapter 1

Introduction

1.1 Motivation

Vehicles are central to the high efficiency and productivity in all industries. Typically, a group of them are classified as heavy vehicles in the sense of Gross Vehicle Mass (GVM) and number of axles, such as heavy trucks, long buses, as well as some engineering vehicles like mobile cranes and dump trucks. An example of mobile crane LTM 1040-2.1 from Liebherr® and the corresponding longitudinal dimensions are illustrated in Fig. 1.1. The chassis of some industrial vehicles are specially designed with nonconventional wheelbase and axles to improve the capacity or tractive force to accommodate specific needs in various work environments. Considering the unusual size and shape of the vehicle body in addition to the complexity of the work environment, safety issues must be exclusively addressed when operating these heavy vehicles, particularly those with very large lateral and longitudinal dimensions.

To improve the maneuvering safety of heavy vehicles, various methods and technologies have been proposed from different aspects in addition to following the safety regulations, such as with improved vehicle and roadway design, effective traffic control, accident study, etc. Accidents not only threaten lives but also increase operational costs and disrupt productivity. It is reported that 34 percent of crashes are caused by road-related elements [3]. It has been suggested that crashes can be reduced if brake and steering defects are largely eliminated through

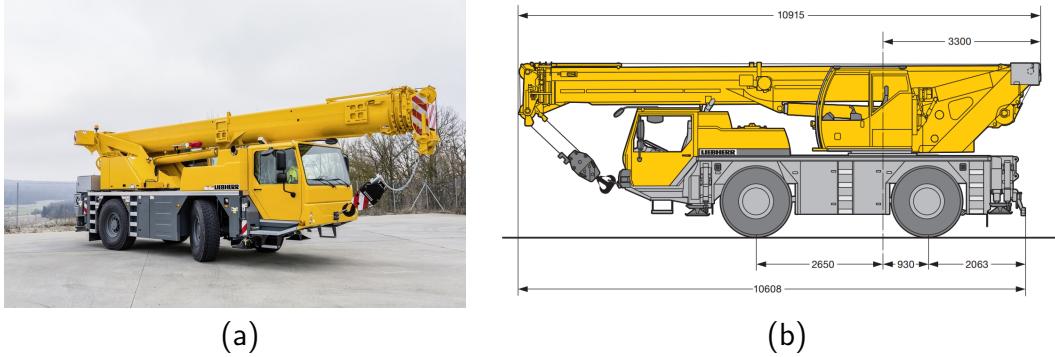


Figure 1.1: (a) Mobile crane with two axles. (b) Side view and dimensions (unit: mm).

routine maintenance by users [4]. Although injury and death rates can be reduced to some extent by improving the roadway and automobile designs, driving safety is still a major concern. Serious work-related roadway crashes are reported to be the leading cause of death from traumatic injuries in the U.S. [5]. Compared with small vehicles, long and heavy vehicles are more difficult to maneuver due to their large body size that presents challenges. For example, roadway design standards are found to be deficient when vehicles of larger size are involved [6]. Also, on reviewing the Large Truck Crash Causation Study (LTCCS) database [7], several relevant maneuvers, such as negotiating a curve, turning at an intersection, etc., have been found to be critical factors of loss of control [8]. It reveals that accident-prone maneuvers are mainly left or right swerves, which are considered to be related to the road geometry such as the radius of curvature, length of the curve, etc., when investigating the accidents involving a Heavy Goods Vehicle (HGV) [9].

For these reasons, an efficient path planning framework becomes critically important especially when the vehicles are operating within a narrow and cluttered environment. First of all, the alignment, which is the route of the road composed of a series of horizontal tangents and curves, needs to be taken into consideration and carefully evaluated before maneuvering the vehicle. This is extremely important when a vehicle is steering around a corner, which is one of the frequently encountered situations at roundabouts, intersections, and parking lots. Apart from the road issues, the characteristics of the vehicle is another important factor. To

evaluate the kinematic and dynamic properties, these vehicles are also regarded as wheeled mobile robots. Considering practical scenarios like a construction site or a petrochemical plant, vehicles are often maneuvered at very low speed especially when steering around a corner. Most heavy vehicles, including conventional four wheel vehicles, are designed and manufactured with Ackermann steering geometry [10] to avoid sideslip of tires. It has been found that no considerable lateral acceleration occurs in the process of slow cornering, which is contrary to the fast cornering process where tires have to absorb the centrifugal force generated by the lateral acceleration. In other words, the slip angle can be neglected in slow cornering mode, thus a pure kinematic model can be used without considering the dynamic properties. In such a model, a instantaneous center of rotation (ICR) is shared by all the wheels when negotiating a curve [11]. This kinematic model is also referred to as the slip-free bicycle model or single-track model, which is widely used in many works, such as [12], [13], [14], etc. There are multiple benefits of using a simplified kinematic model. Firstly, the system identification process is much easier since fewer parameters are involved compared with a higher fidelity model. Secondly, the kinematic model has a comparable accuracy against the dynamic model at lower speeds using the same controller [14]. Finally, the kinematic model is a generalized representation so that a controller or path planner designed based on this model can be quickly adapted to other types of vehicles.

With regard to the same geometric representation (Fig. 3.6 in Chapter 3), there exists two variations of kinematic models. The first type (Type I) is a car-like model constrained by equation (2.28), where the configuration of the vehicle posture is composed of position coordinates and orientation of the vehicle body, i.e., $(x, y, \theta)^T$. Depending on the allowed range of the speed and steering angle, this model can be further classified as different subtypes according to [15]: tricycle or unicycle [16], simple car [17], Reeds-Shepp car [18], and Dubins car [19]. The other type (Type II) is an extension from the Dubins car by adding the curvature as an extra configuration parameter, which is originally proposed in [20] and used in [21], [22] and many other applications. The basic idea is to replace the angular velocity with the angular acceleration as an input, and restrict the bounds of

both the angular acceleration and steering angle at the same time, which is much stricter than the bounded angular velocity required by Type I. Consequently, when addressing a path planning problem, we can only consider vehicles of Type II because the results can be conveniently applied to Type I without violating the kinematic constraints. An additional advantage of using kinematic model of Type II is that the steering angle is continuously changed, thus the obtained curve is curvature continuous, which guarantees that the vehicle does not need to stop at positions where curvatures are discontinued to reorient its wheels [22]. Thus a path with higher order geometric continuity provides more comfort to the driver and passengers.

After establishing the kinematic model, generating a feasible path remains a major challenge considering the nonholonomic constraints imposed on the vehicle and the complex environment. A classic approach for formulating and solving such problems is the configuration space method [23], which tries to simplify the robot or vehicle into a single point and map the robot from the workspace to configuration space where the obstacles are compensated according to the robot shape. However, approaches based on configuration space cannot handle nonholonomic constraints appropriately. With regard to modeling of the vehicle and environment, another issue is the geometric shape of the vehicle. Most vehicles are relatively small compared with the surrounding obstacles, thus they are commonly treated as a plate [24]. As a result, the planner only needs to handle the nonholonomic constraints while the collision checking stage can be very efficient with such simplifications. When the robot size is comparatively large or its geometric shape is irregular, this kind of simplification may lead to serious errors. Taking a long vehicle as an example, valid paths from start configuration x_s to goal configuration x_g exist obviously, as shown in Fig. 1.2b. In contrast, a path planner using a vehicle of plate shape with the same bounding sphere cannot find a valid path, which is illustrated in Fig. 1.2a. In this situation, it can be verified that the configuration space approach making use of Minkowski sum generates the same results as Fig. 1.2a. As a result, an effective collision checking process is critical to a feasible path planner, which must fully utilize the space to accommodate the vehicle maneuvering. Fig. 1.3 is a

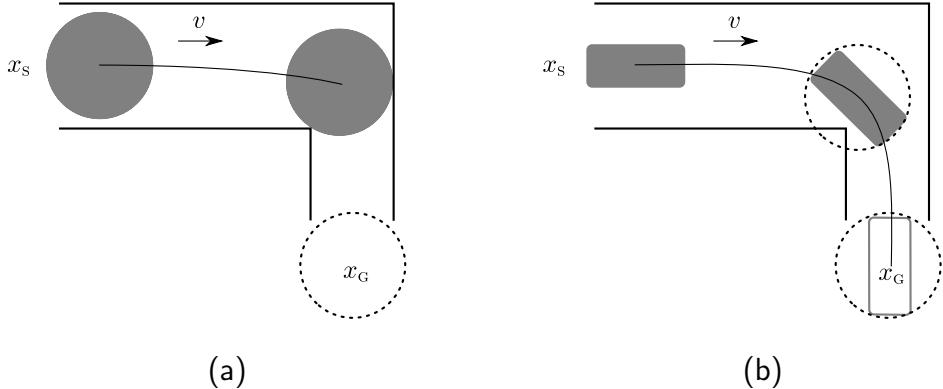


Figure 1.2: Robot of different geometric shapes: (a) No solution found for a plate shape. (b) Solution is found for a rectangular shape.

practical demonstration of a local planner for a two-axle mobile crane which takes both the front and rear wheel paths into consideration and the collision checking is performed based on the swept path analysis.

Another issue is the computation involved when generating a local smooth path connecting two discrete configurations, which requires solving a two point boundary value problem. It is usually a time consuming process because the obtained path should satisfy the nonholonomic constraints imposed on the kinematic model. Also we expect that the desired local path can be characterized by all the necessary parameters associated with the kinematic model. In such a way, the obtained curve can be reparameterized and further tuned according to the vehicle types and obstacle distributions. Also the curve should be easily computed or approximated for online planning with an acceptable error range, which demands accurate and efficient approximation algorithms.

Finally yet importantly, it is much more challenging when generate feasible paths in a cluttered environment. Generally, a hierarchical framework should be considered and the local path planner should be seamlessly integrated into the global planner. Among all the available approaches, sampling based planning (SBP) [25] is the most promising one serving as the global planner due to its efficiency and flexibility. With random computation, it is able to provide fast solutions to complicated path planning problems in high dimensions. The connectivity of

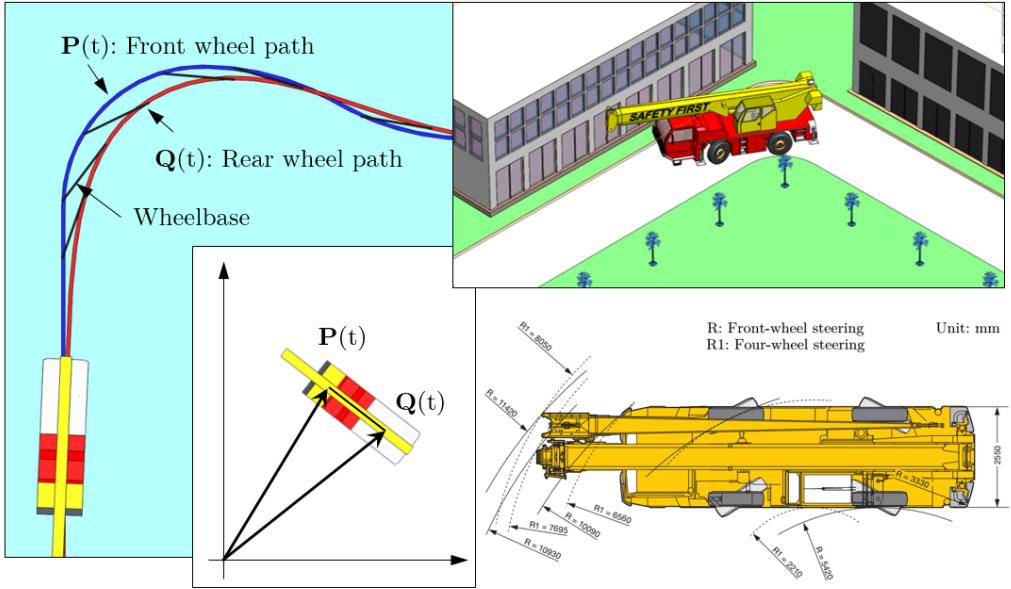


Figure 1.3: Local path planning for a mobile crane.

the search space is approximated and represented with a graph structure, and the strategy for generating new edges should reflect the kinematic constraints. Other practical issues including map preprocessing and node pruning are also needed to be taken into consideration when implementing the SBP.

1.2 Organization

The rest of the thesis is organized as follows:

- (i) Chapter 2 provides a background information on problem of curvature continuous path smoothing, which introduces state-of-the-art approximation algorithms for clothoid curves. We also review the nonholonomic path planning techniques with a focus on sampling based solutions.
- (ii) Chapter 3 starts with the kinematic analysis of different wheels and various types of vehicles, and finally generalizes a simplified kinematic bicycle model. Based on this model, local path primitives involving clothoids are derived for path smoothing and analyzed in a single turn.

- (iii) Chapter 4 designs a framework to approximate the clothoid curves based on quintic Bézier curves by minimizing the curvature error. Accuracy and robustness are demonstrated by comparing the proposed algorithm with other approaches.
- (iv) Chapter 5 examines the geometric properties of the path primitives and a clothoid spline based RRT algorithm is proposed to handle nonholonomic planning problems of vehicles subject to a maximum angular acceleration. We also show that the PRM algorithm can be easily integrated with obtained local planners.
- (v) Chapter 6 concludes the research and discusses potential future work.

Chapter 2

Literature Review

Safety and efficiency are two major issues when it comes to vehicle driving and operating, particularly those with relatively large lateral and longitudinal sizes. Such vehicles can be considered as wheeled mobile robots (WMRs). When performing path planning for vehicles with nonholonomic constraints, we have to consider not only the environment involving obstacles but also the constraints of the vehicles themselves. Particularly, heavy vehicles used in a process plant or construction site have limited maneuverability with a maximum steering angle and are usually operated at a considerably slow speed. To simplify the analysis, the vehicles are supposed to move at a constant speed without slipping thus the mechanical constraints of the vehicle impose a maximum curvature constraint on the reference path. Accordingly, a generalized kinematic model can be obtained for a single-unit vehicle with different geometric parameters and steering structures. Meanwhile, a desired path for a vehicle subject to bounded curvature and angular acceleration constraints should have G^2 continuity, i.e., the curve of the path is supposed to have continuous curvature profile to ensure a smooth steering behavior. With respect to a static environment containing narrow corridors, a feasible path should be safe in the sense of maximizing the obstacle clearance while satisfying the above constraints. In this chapter, by reviewing the existing methods a geometric local path planning framework for nonholonomic vehicles is proposed to generate a G^2 smooth path with maximum curvature constraint focusing on vehicle driving in a static environment with narrow corridors.

2.1 Curvature Continuous Path Smoothing

It is a nontrivial task to find a feasible path for wheeled robots especially those under nonholonomic constraints. A practical planner should satisfy certain conditions limited by the kinematic and dynamic properties of the robot while avoiding obstacles in the environment [15]. A common way of achieving this is by computing a series of configurations and joining them with continuous and oriented curves, among which the most frequently used are Dubins path [19] and Reeds-Shepp path [26]. Both of them are composed of circular arcs and straight lines, and the curvature is discontinuous at the joints, which means that the nonholonomic robot has to stop to reorient its steering wheels when tracking these paths [22]. To address this issue, various smooth path planning methods have been proposed which introduce different G^2 continuous curves, such as Bézier curves of different degrees in [2], [27] and [28], Fermat's spiral [29], Pythagorean-hodograph curve [30], and clothoid [31]. One of the characteristics that distinguish the clothoid from others is its simple curvature profile, which makes the generated path much easier to track [32]. On the other hand, it is proved that extremal paths of the Dubins model contain clothoidal curves if the vehicle is controlled by the angular acceleration [20]. Also, clothoids are used to obtain the shortest path following prescribed conditions of lateral accelerations and steering wheel speeds [33]. Moreover, clothoid turns out to be a natural choice when handling path planning problems with curvature related constraints [34], [35]. Therefore, clothoids are widely used for numerous path planning problems involving wheeled robots [36], UAVs [37], human locomotors [38], etc.

2.1.1 Planar Curves for Path Smoothing

Given a sequence of N waypoints $(P_i)_{i=1}^n$ on a plane, determining a smoothing path \mathcal{P} can be equivalently described as constructing a continuous curve \mathcal{C} within the range of a set of discrete sample points from a mathematical point of view. Generally, this approximation procedure can be performed using two different approaches, i.e., interpolation (along with extrapolation) and curve fitting. To avoid

significantly deviating from the specified waypoints, the desired path should be kept close enough to the waypoints. This can be quantified by the distance between the given waypoint P_i and its nearest neighbor on the curve, i.e., the orthogonal projection of the point on the curve P'_i , should be within an acceptable range. On the other hand, safety has higher priority than accuracy in most path smoothing problems. For example, to avoid collisions with obstacles and surrounding objects in a cluttered environment, the actual path cannot always follow all the waypoints precisely, especially under conditions where the waypoint is close to the obstacles or the mobile robot is moving in narrow spaces. Thus, the constraint that the curve has to go exactly through the waypoints should be relaxed in the majority of cases. On the other hand, waypoints are always connected to serve as a control polyline or polygon to adjust the shape of the continuous curve. The smoothness of the obtained curve is usually defined by two different types of continuities, i.e., parametric continuity C^n and geometric continuity G^n [39]. Therefore, to meet different requirements, a drivable planar path is commonly expressed by a subset of piecewise curves $\mathcal{P} = \bigcup_{i=1}^n \mathcal{P}_i$ [29]. Each constituent curve segment \mathcal{P}_i is defined as

$$\mathcal{P}_i := \{\mathbf{p}_i \in \mathbb{R}^2 \mid \mathbf{p}_i = \mathbf{p}_{i,p}(t), \forall t \in [t_{i,0}, t_{i,1}] \subset \mathbb{R}\}, \quad (2.1)$$

where \mathbf{p}_i is the vector form of P_i which is equivalent to $\overrightarrow{OP_i}$ (O is the origin of the coordinate system). Here $\mathbf{p}_{i,p}(t)$ is a parametric function indicating the position coordinates of the point \mathbf{p}_i moving along the i -th segment \mathcal{P}_i and t is a path parameter defined on interval $[t_{i,0}, t_{i,1}]$ which does not necessarily represent the traveling time.

Parameters such as position, orientation and curvature are often included in the state vector representing the generated trajectory [40], continuous representation of these parameters are of great importance to path planning and smoothing applications as mentioned in [36], [37] and [38]. Different orders of geometric continuity are required in various scenarios. For example, G^0 indicates the curves agree at the join point. In addition to the zeroth order of geometric continuity condition, G^1 demands a common tangent direction, i.e., the same tangential angle at the

junction while G^2 needs an extra continuity of the curvature profile. Given a planar curve $\mathbf{p}(t)$, its tangential angle $\theta(t)$, which is also referred to as turning angle in [41], is characterized by:

$$\frac{\mathbf{p}'(t)}{\|\mathbf{p}'(t)\|} = \begin{pmatrix} \cos(\theta(t)) \\ \sin(\theta(t)) \end{pmatrix}. \quad (2.2)$$

Actually, the parametric equation of curve segment (2.1) can be easily changed via reparameterization with a different parameter. Since we here only focus on the geometric property of the path without considering timing information, we express the curve equation using the arc length parameter s in order to keep the path representation unique. After reparameterization with respect to the arc length s , the curvature (more specifically, the extrinsic curvature) of the planar curve $\mathbf{p}(s)$ is defined by:

$$\kappa(s) = \|\mathbf{T}'(s)\|, \quad (2.3)$$

where $\mathbf{T}(s) = \frac{\mathbf{p}'(t)}{\|\mathbf{p}'(t)\|}$ is the unit tangent vector defined in (2.2).

When handling robotic path smoothing and planning problems, the requirements of different orders of geometric continuity are usually satisfied by adapting various types of curves to a piecewise smooth spline for path synthesis. Practically, the segments used for path smoothing are generally characterized by simple curves, and the most popular ones include straight lines [42], circular arcs [19], and clothoids [22]. Taking a two-dimensional straight line of tangential angle θ_0 passing through a waypoint $\mathbf{p}_0 = (x_0, y_0)^T$ as an example, its parametric equation with respect to s is given by:

$$\mathbf{p}_{\text{line}}(s) = \mathbf{p}_0 + \mathbf{v} s, \quad s \in [0, l], \quad (2.4)$$

where l is the total length of the linear path with $\mathbf{v} = (\cos \theta_0, \sin \theta_0)^T$. It can be easily seen that the tangential angle $\theta(s) = \theta_0$ keeps constant across the entire range while the curvature always remains 0 everywhere. Similarly, a circular arc of radius r with center $\mathbf{c}_0 = (x_0, y_0)^T$ can be represented by:

$$\mathbf{p}_{\text{arc}}(s) = \mathbf{c}_0 + \begin{pmatrix} \sin(\theta_0 + \frac{s}{r}) \\ -\cos(\theta_0 + \frac{s}{r}) \end{pmatrix} r, \quad s \in [0, l], \quad (2.5)$$

where θ_0 is the initial tangential angle and l is the total length of the circular path. The orientation of a point on the circular path segment (2.5) is given by $\theta(s) = \theta_0 + \frac{s}{r}$, while the curvature equals the reciprocal of the radius, i.e., $\kappa(s) = \frac{1}{r}$. Consequently, a spline only composed of circular arcs and straight lines can achieve up to G^1 continuity, and other types of transition curves are needed if a higher order of geometric continuity is required. For example, to obtain a G^2 path, the segment \mathcal{P}_i should share the same curvature with its adjacent curves at the joints. For this reason, a large number of smooth curves for joining a straight line and a circle have been proposed and developed as the transition curves to guarantee G^2 continuity: Pythagorean hodograph (PH) curve [43], Bézier curve [44], Fermat's spiral [29], polynomial [45], clothoid [46], just to name a few.

Among these methods, clothoid is one of the most frequently used spirals as a track transition curve in highway and railway design [47]. This is mainly due to its simplicity that the curvature of a clothoid is exactly linear in arc length [48]. Furthermore, it reveals that the optimal solution connecting two distinct configurations can be obtained by applying Pontryagin's Maximum Principle on Dubins model, and the extremal path is composed of line segments and arcs of clothoids [49]. Additionally, due to the linear curvature property, clothoid paths are easy to follow and thus widely used as transition curves for path smoothing applications [50]. As an example, a steering method based on a clothoid pair and a circular arc is presented in [22] to form a smooth path with upper-bounded curvature. A general clothoid path of length l starting at $\mathbf{p}_0 = (x_0, y_0)^T$ can be expressed in parametric form with respect to arc length s as in [51]:

$$\mathbf{p}_{\text{clo}}(s) = \mathbf{p}_0 + \left(\begin{array}{l} \int_0^s \cos \left(\theta_0 + \kappa_0 \xi + \frac{\sigma \xi^2}{2} \right) d\xi \\ \int_0^s \sin \left(\theta_0 + \kappa_0 \xi + \frac{\sigma \xi^2}{2} \right) d\xi \end{array} \right), \quad s \in [0, l], \quad (2.6)$$

where θ_0 and κ_0 are the initial tangential angle and curvature as $s = 0$, respectively. Here σ is a constant factor related to the shape of the path, which is called the sharpness of the clothoid. As a result, based on equation (2.2) and (2.3), the tangential or winding angle of the clothoid path (2.6) is given by

$$\theta(s) = \theta_0 + \kappa_0 s + \frac{\sigma s^2}{2}, \quad \theta_0, \kappa_0, \sigma \in \mathbb{R}, \quad (2.7)$$

while the curvature $\kappa(s)$ can be expressed as a linear function of its curve length s :

$$\kappa(s) = \kappa_0 + \sigma s, \quad \kappa_0, \sigma \in \mathbb{R}. \quad (2.8)$$

An example of clothoid with its curvature profile is shown in Fig. 2.1. It can

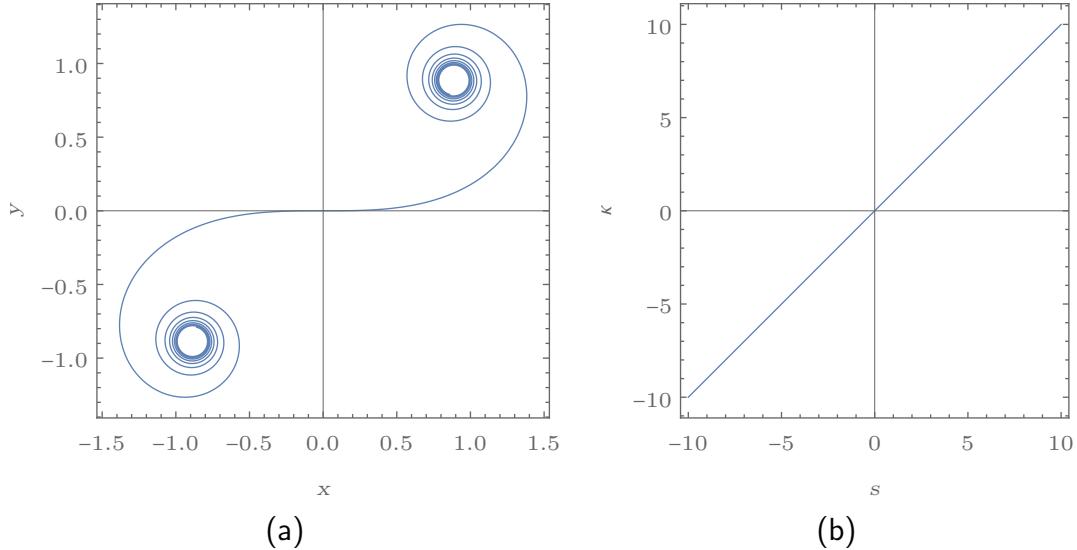


Figure 2.1: Clothoid defined on $[-10, 10]$: (a) $\mathbf{p}_{\text{clo}}(s)$. (b) $\kappa(s)$.

be seen that the curvature profile of the clothoid segment (2.6) is able to change linearly from 0 to $\frac{1}{r}$, thus a G^2 continuous path can be constructed by joining a line segment (2.4) or a circular arc (2.5) of radius r with a piece of clothoid transition curve.

2.1.2 Computation and Approximation of Clothoid Curves

Despite having such attractive properties, clothoids are difficult to implement in real time applications due to Fresnel integrals involved with no closed form expression available [52]. Here, Fresnel integrals are represented by a pair of nonnegative functions $C_f(s)$ and $S_f(t)$ [53]:

$$\begin{aligned} C_f(t) &= \int_0^t \cos \frac{\pi \xi^2}{2} d\xi, \\ S_f(t) &= \int_0^t \sin \frac{\pi \xi^2}{2} d\xi. \end{aligned} \quad (2.9)$$

Thus a number of methods have been developed to compute and approximate clothoids, which generally fall into two categories. The first attempts to evaluate the clothoid coordinates in a pointwise manner by calculating Fresnel integrals numerically as in [54] and [55]. The major drawback of these methods is the loss of geometric information, e.g., continuous representation of position, orientation and curvature profiles, which are of great importance to path planning and smoothing applications as mentioned in [36], [37] and [38].

2.1.2.1 Bézier Approximation via Taylor Expansion

Instead, another category of methods relies on smooth approximation using continuous curves with geometric information easily accessible and retrievable. A typical example is the Bézier curve approximation using Taylor expansion [56], which restricts the winding angle within $[0, \frac{\pi}{2}]$. By setting zero initial conditions to the general form (2.6), i.e., $(x_0, y_0)^T = (0, 0)^T$, $\kappa_0 = 0$, and $\theta_0 = 0$, we can obtain

$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = \begin{pmatrix} \int_0^s \cos \frac{\sigma \xi^2}{2} d\xi \\ \int_0^s \sin \frac{\sigma \xi^2}{2} d\xi \end{pmatrix}. \quad (2.10)$$

Using Taylor expansion, (2.10) can be expressed as

$$\begin{pmatrix} x(s) \\ y(s) \end{pmatrix} = \begin{pmatrix} \sum_{n=0}^{\infty} \frac{(-\frac{1}{4})^n \sigma^{2n} s^{4n+1}}{(4n+1)(2n)!} \\ \sum_{n=0}^{\infty} \frac{(-1)^n 2^{-2n-1} \sigma^{2n+1} s^{4n+3}}{(4n+3)(2n+1)!} \end{pmatrix}. \quad (2.11)$$

After truncating $x(s)$ and $y(s)$ in (2.11) to only the first N terms, and normalizing the arc length parameter s to $\hat{s} \in [0, 1]$, it can be rewritten by an equivalent Bézier form $(x_N(\hat{s}), y_N(\hat{s}))^T$ of degree $4N + 3$. For the sake of convenience, we evaluate the approximation error by $\epsilon_N = \sqrt{(x_N - x)^2 + (y_N - y)^2}$. Obviously, ϵ_N becomes smaller as N increases and $\epsilon_N = 0$ when N approaches infinity. Fig. 2.2a is a demonstration of the approximation approach ($\sigma = 1$) through Taylor expansion by setting $N = 1, 2, \dots, 5$, where the corresponding equivalent Bézier curves are of degree 7, 11, ..., 23 respectively. The approximation error is shown in Fig. 2.2b

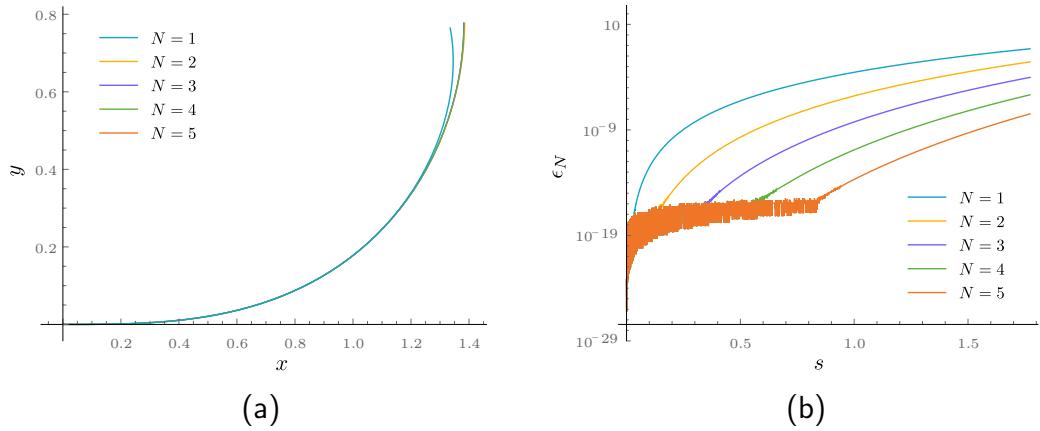


Figure 2.2: Clothoid approximation through Taylor expansion as $N = 1, 2, \dots, 5$:
(a) Approximation result. (b) Approximation Error.

and the greatest error occurs when a Bézier curve of degree 7 is used. As revealed by Fig. 2.2, the resulting Bézier curve can be of very high degree to achieve a reasonable approximation error. In [56], the author uses a Bézier curve of degree 23 to ensure an acceptable approximation error, which renders this approach unsuitable for most real time applications.

2.1.2.2 Hermite Approximation via S-Power Series

To reduce the order of the polynomials used, s-power series [57] are employed to construct an order- k Hermite interpolant, which generates a C^k continuous Hermitian spline [58]. Also, s-power series can be regarded as the two-point analog of the Taylor series, and by truncating at the k -th term, the s-power series becomes a Hermite interpolant of order k . By replacing the variable t in function $f(t)$ ($t \in [t_0, t_1]$) with $u \in [0, 1]$, it can be rewritten as $a(u) = f(t(u))$. Here, we define

$$t(u) = t_0(1-u) + t_1u. \quad (2.12)$$

Correspondingly, its power series can be expressed by

$$\begin{aligned} a(u) &= \sum_{k=0}^{\infty} a_k(u) s^k, \\ s &= (1-u)u, \end{aligned} \quad (2.13)$$

$$a_k(u) = (1-u)a_k^0 + u_k^1,$$

where the function $a(u)$ is expressed in Bernstein form and a_k^0, a_k^1 are the Bézier ordinates. The Hermite interpolant of order k is obtained from the s-power series (2.13):

$$H_k(a, u) = \sum_{i=0}^k a_i(u) s^i. \quad (2.14)$$

The coefficients can be computed according to [59]:

$$\begin{aligned} \tilde{a}(u) &= a(1-u), \\ c_i &= \frac{1}{i!} a^{(i)}(0), \\ \tilde{c}_i &= \frac{1}{i!} \tilde{a}^{(i)}(0), \\ h_{i,k}(u) &= \begin{cases} 1-u & \text{if } i = k \\ -u \binom{-i+2k-1}{-i+k-1} - (u-1) \binom{-i+2k-1}{-i+k} & \text{if } i < k \\ 0 & \text{if } i > k \end{cases} \\ \tilde{h}_{i,k}(u) &= h_{i,k}(1-u), \\ a_k(u) &= \sum_{i=0}^k \left(c_i h_{i,k}(u) + \tilde{c}_i \tilde{h}_{i,k}(u) \right). \end{aligned} \quad (2.15)$$

Finally, by substituting the results (2.15) into (2.14), we can derive the Hermite approximation. If we define the approximation error as $\epsilon = \|H_k(a, u) - a(u)\|$, the corresponding approximation errors are illustrated in Fig. 2.3. It can be seen that the maximum approximation error occurs close to the middle part of the curve, and at both ends the approximation error approaches zero. In Fig. 2.3a, we find that a very high order polynomial needs to be used to ensure an acceptable approximation error. The order of the Hermite polynomials are of order $2k + 1$ corresponding to $k = 1, 2, \dots, 5$. Also, a higher accuracy can be achieved when approximating a clothoid with greater sharpness, as shown in Fig. 2.3b.

2.1.2.3 Bézier Approximation by Minimizing the Curvature Error

Moreover, Bézier curve is also used to approximate the clothoid, which is a subset of generalized Cornu spiral (GCS), with an emphasis on geometric continuity in

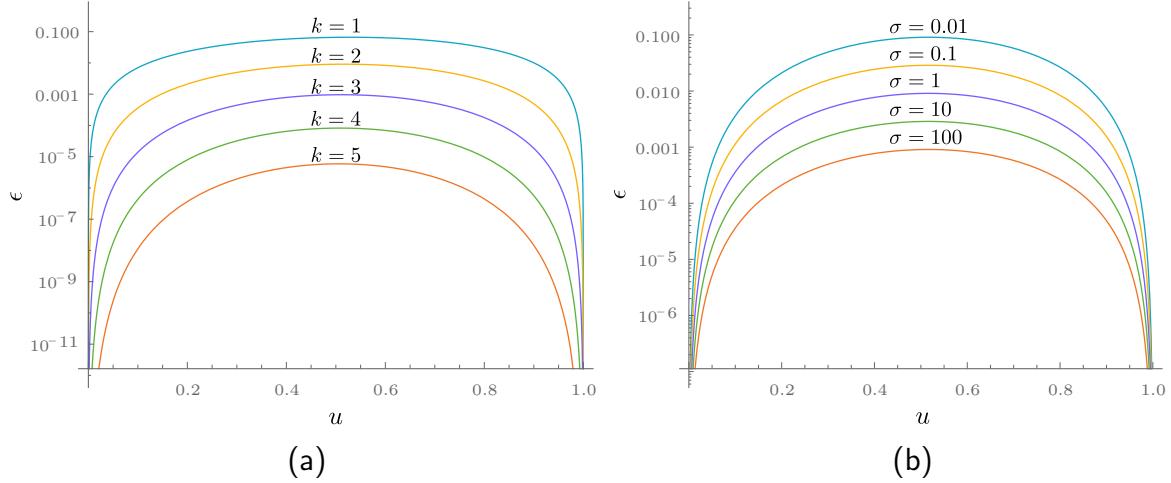


Figure 2.3: Clothoid approximation error via s-series: (a) $\sigma = 1$, $k = 1, 2, 3, 4, 5$. (b) $k = 2$, $\sigma = 0.01, 0.1, 1, 10, 100$.

[60] and [61]. A GCS curve has a rational linear monotonic curvature profile with respect to its arc length, namely,

$$\kappa(s) = \frac{\kappa_0 l + (\kappa_1 - \kappa_0 + r\kappa_1)s}{l + rs}, \quad s \in [0, l], \quad (2.16)$$

where l is the total arc length of the curve, κ_0, κ_1 are the initial and final curvature values, and $r \in (-1, \infty)$ denotes the shape factor. (2.16) can also be rewritten by

$$\kappa(s) = \frac{p + qs}{l + rs}, \quad (2.17)$$

where $p = \kappa_0 l$ and $q = \kappa_1 - \kappa_0 + r\kappa_1$. Depending on the values of p, q , and r , the GCS can be subdivided into different types of curves: straight line ($p = q = 0$), circular arc ($q = r = 0$), clothoid ($q \neq 0$ and $r = 0$), and logarithmic spiral ($q = 0$ and $r \neq 0$).

For a 2D spiral $\mathbf{r}(s)$ with $s \in [0, l]$ where l is the total length, its parametric form $\mathbf{r}(t)$ ($t \in [t_0, t_1]$) with respect to the curvature can be represented by

$$\mathbf{r}(t) = \frac{\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t) \cdot \hat{\mathbf{z}}}{\|\dot{\mathbf{r}}(t)\|^3} \quad (2.18)$$

according to (2.3), where $\hat{\mathbf{z}} = (0, 0, 1)$. Non-rational Bézier curves of degree 3 and 5 are used in [60] to approximate the GCS by matching coordinates, slopes and

curvatures at end positions. Also the author proposed to use the curvature profiles to measure the approximation error. [61] uses a similar approach with quintic Bézier curve approximation. After applying the geometric continuity constraints, two parameters remains independent. An optimization procedure is applied to search for desired values by minimizing the curvature error, which requires a proper initial starting point. One of the common disadvantages shared by [60] and [61] is that the non-deterministic search algorithms cannot guarantee satisfactory results, i.e., the performance of these approaches is unstable and not robust. Last but not least, the numerical optimization procedures are computationally intensive thus do not fulfill the requirements of online applications. Seeing these shortcomings, [62] uses a G^{2+} deterministic approximation method with linear interpolations to handle the divergent regions.

All the approximation approaches used in [60], [61] and [62] are able to generate acceptable Bézier curve approximations with sufficient accuracy in certain cases. However, these methods are only applicable for clothoids of unit arc length ($l = 1$), and they suffer from the same drawback that the winding angle of targeted clothoid has to be restricted to $|\theta| \leq \frac{\pi}{2}$. Under these restrictions, a large portion of clothoids cannot be approximated even with scaling transformation. Consequently, these approaches are not scalable and the application of the above results are limited in practical use where generic clothoids with arbitrary parameters are involved.

2.1.2.4 RBC Approximation

Different from the methods used above, computation efficiency is addressed by another category of approaches. A decent attempt is to introduce rational Bézier curves (RBCs) for online clothoidal paths generation in [63] and [64]:

$$\mathbf{B}(t) = \frac{\sum_{i=0}^n b_{i,n}(t) w_i \mathbf{P}_i}{\sum_{i=0}^n b_{i,n}(t) w_i}, \quad (2.19)$$

where $b_{i,n}(t)$ is the Bernstein basis, \mathbf{P}_i is the control point, and w_i is the non-negative weight. Using the transformation invariance property, a scaling factor K

can be included in the RBC to approximate a piece of scaled clothoid:

$$\mathbf{B}(t) = \frac{\sum_{i=0}^n b_{i,n}(t) w_i K \mathbf{P}_i}{\sum_{i=0}^n b_{i,n}(t) w_i}. \quad (2.20)$$

The Fresnel integrals are continuously approximated with a rational Bézier curve

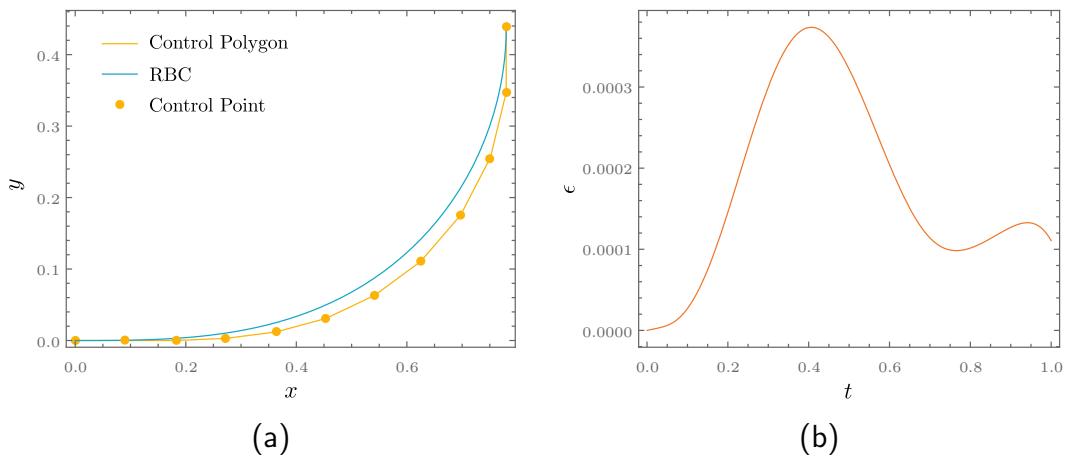


Figure 2.4: Clothoid approximation by an 11-th order RBC: (a) Approximation result. (b) Approximation Error.

whose control points are obtained based on the least squares criterion and reallocated accordingly. The order of the curve will be increased and this procedure is repeated until specified accuracy is achieved. Generally, the obtained rational Bézier curve has a relatively high order. This approach also requires the winding angle limited to $[0, \frac{\pi}{2}]$. As illustrated in Fig. 2.4a, an RBC of order 11 is used to approximate $(C_f(s), S_f(s))^T$ and its approximation error is shown in Fig. 2.4b. The approximation accuracy of RBC method will be further discussed later in Chapter 4.

2.1.2.5 Circular Interpolation Approach

Furthermore, a circular interpolation based approach is proposed in [1] for path planning applications. It takes discrete samples from the clothoid and interpolates

these points with circular arcs. For the standard clothoid curve in the lookup table, its sampling step is $\Delta s_{\mathcal{L}}$ from j -th to $(j + 1)$ -th point. The required parameters are computed as

$$\begin{aligned} r_j &= (\sigma_{\mathcal{L}} (j + 0.5) \Delta s_{\mathcal{L}})^{-1}, \\ \theta_{\mathcal{L}}^j &= \frac{1}{2} \sigma_{\mathcal{L}} (j \Delta s_{\mathcal{L}})^2, \\ \begin{pmatrix} x_c^j \\ y_c^j \end{pmatrix} &= \begin{pmatrix} x_{\mathcal{L}}^j \\ y_{\mathcal{L}}^j \end{pmatrix} + r_j \begin{pmatrix} \cos(\theta_{\mathcal{L}}^j + \frac{\pi}{2}) \\ \sin(\theta_{\mathcal{L}}^j + \frac{\pi}{2}) \end{pmatrix}, \end{aligned} \quad (2.21)$$

where r_j is the radius, $\theta_{\mathcal{L}}^j$ is the tangent angle, $(x_c^j, y_c^j)^T$ is the center of the circular arc. Thus any coordinates $(x_{\mathcal{L}i}, y_{\mathcal{L}i})^T$ at length s are computed as:

$$\begin{pmatrix} x_{\mathcal{L}i} \\ y_{\mathcal{L}i} \end{pmatrix} = \begin{pmatrix} x_{\mathcal{L}}^j \\ y_{\mathcal{L}}^j \end{pmatrix} + 2r_j \sin \frac{\Delta_s}{2r_j} \begin{pmatrix} \cos(\theta_{\mathcal{L}}^j + \frac{\Delta_s}{2r_j}) \\ \sin(\theta_{\mathcal{L}}^j + \frac{\Delta_s}{2r_j}) \end{pmatrix}, \quad (2.22)$$

where $\Delta s = s - j \Delta s_{\mathcal{L}}$.

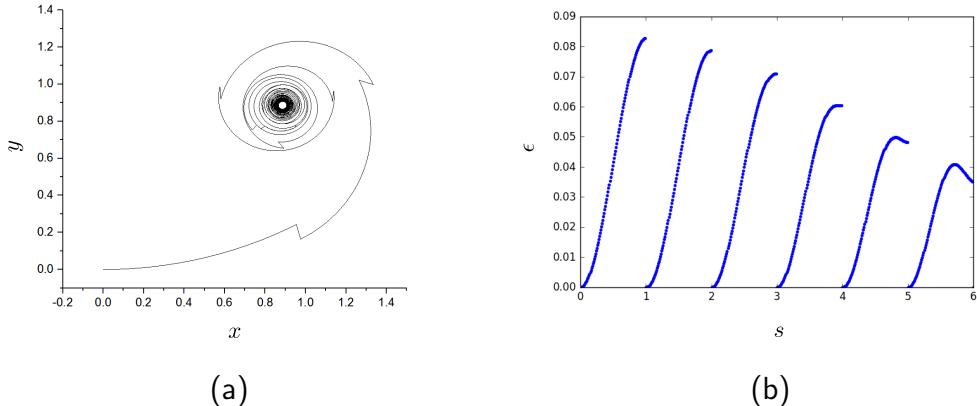


Figure 2.5: Clothoid approximation via circular interpolation: (a) Approximation result. (b) Approximation Error.

One of the benefits of circular interpolation approach is the computation efficiency. Nevertheless, because the resulting curve is not continuous, gaps occur at the end of each interpolation interval. The gaps are illustrated in Fig. 2.5a, which adopts a large sampling interval $\Delta s_{\mathcal{L}} = 1.0$ to address the discontinuity problem, and the remaining parameters are $\sigma = 1$, $\kappa_0 = 0$, $\theta_0 = 0$ and $(x_0, y_0)^T = (0, 0)^T$.

These gaps are also the positions where the maximum approximation error happens (Fig. 2.5b). As a result, the approximation error heavily relies on the sampling interval Δs_c . Choosing a smaller sampling interval means higher accuracy but this will increase the number of samples as well as computational costs. In practical applications, the accuracy and efficiency must be carefully balanced. Based on the precomputed values of a basic clothoid, any point on a general clothoid can be computed by querying a lookup table and applying a proper transformation.

2.2 Nonholonomic Planning

A time-invariant nonlinear control system in \mathbb{R}^n can be expressed by a differential model [65] as

$$\dot{x}(t) = f(x(t), u(t)), \quad (2.23)$$

where $x(t) \in \mathbb{R}^n$ is a state in the state space \mathcal{X} and f is called the state transition function. If \mathcal{X} only consists of all reachable positions of an object, it is also referred to as Configuration Space (CS) denoted by \mathcal{C} . Considering the case of a rigid body in two-dimensional space, the result of its configuration space is a topological space named special Euclidean group $SE(2)$, which is homeomorphic to $\mathbb{R}^2 \times SO(1)$. Here, $SO(n)$ is a special orthogonal group representing the group of n -dimensional rotation matrices. $u(t) \in U(x(t))$ is a state in the action space, which is defined as the union of all possible actions: $U = \bigcup_{x(t) \in \mathcal{X}} U(x(t))$. For a specific planning problem, a state $x_s \in \mathcal{X}$ and a subset $\mathcal{X}_G \subset \mathcal{X}$ are defined as the initial or start state and goal set, respectively.

Sometimes, a system model may not be explicitly expressed as (2.23), an implicit system model can be specified alternatively:

$$g(q, \dot{q}) = 0, \quad (2.24)$$

where $q \in \mathbb{R}^n$ is a configuration, and $q = (x, y)^T$ as $n = 2$. In certain cases, the implicit model (2.24) can be transformed to the parametric form (2.23).

Nevertheless, we may encounter higher order systems when handling practical problems, such as a second order system involving acceleration information in

parametric form $\ddot{x}(t) = f(u(t), x(t), \dot{x}(t))$. In 2-dimensional scenario of $\mathcal{C} = \mathbb{R}^2$, this system can be expressed in a state space model alternatively [66]:

$$\begin{pmatrix} \ddot{x} \\ \ddot{y} \end{pmatrix} = f\left(\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}, \begin{pmatrix} x \\ y \end{pmatrix}, u\right). \quad (2.25)$$

It seems different from (2.23), however, any higher order system with expression similar to (2.25) can always be transformed to an equivalent representation only involving first order deviations after a progressive series of order reduction procedures. By assigning $\xi \doteq \dot{x}$ and $\chi \doteq \dot{y}$, the model (2.25) can be extended and reduced to

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\xi} \\ \dot{\chi} \end{pmatrix} = f'\left(\begin{pmatrix} x \\ y \\ \xi \\ \chi \end{pmatrix}, u\right), \quad (2.26)$$

where $(x, y, \xi, \chi)^T \in \mathcal{X}$ and $(x, y)^T \in \mathcal{C}$.

Constraints have to be taken into consideration when tackling planning problems. Generally, the constraints imposed on a rigid body can be classified into two categories: kinematic constraints and dynamic constraints, which together are also known as kinodynamic constraints [67]. As pointed in [68], kinematic constraints are generally defined on the configuration space \mathcal{C} and can be expressed as

$$F(q, \dot{q}) = 0. \quad (2.27)$$

Kinematic constraints can be further categorized into holonomic and nonholonomic constraints based on the integrability. In an expression of a holonomic constraint, only parameters directly defined on the configuration space are involved. That is, by eliminating the velocity variables in (2.27), its holonomic form becomes $F(q) = 0$. It is always possible to transform a holonomic system to an unconstrained one by choosing an appropriate parameterization of \mathcal{C} [15]. On the other hand, Nonholonomic constraints are not integrable, which means that the velocity component \dot{q} in (2.27) cannot be eliminated completely. Many practical planning problems involve this kind of constraints, especially for wheeled mobile robots. For example, a simple car model defined on $\mathcal{C} = \mathbb{R}^2 \times \mathbb{S}^1$ with a configuration $q = (x, y, \theta)^T$ can be written as a Pfaffian constraint [69]:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0, \quad (2.28)$$

which, obviously, cannot be integrated to an expression involving neither \dot{x} nor \ddot{y} .

Dynamic constraints, conceivably, are those involving not only the configuration and the velocity, but also the acceleration imposed by momentums, inertias, forces, etc. Usually, up to second order configuration variables are concerned while higher order may be incorporated in some problems such as vibration analysis. These constraints are usually obtained from Euler-Lagrange equation or Hamilton's equations [15], which are in the following form:

$$G(q, \dot{q}, \ddot{q}) = 0. \quad (2.29)$$

In general, dynamic constraints are always in nonholonomic forms.

Originally, the term nonholonomic planning was proposed in [70] particularly targeting for motion planning problem of wheeled mobile robots. In most cases, nonholonomic planning is applied to problems involving kinematic constraints only [15]. Another related concept is kinodynamic planning, which was first introduced in [67] focusing on motion planning problems subject to velocity and acceleration bounds. After being used in more general scenarios, kinodynamic planning can be regarded as planning problems involving at least second-order differential constraints. From the aspect of control theory, developing a feasible nonholonomic planner without considering collisions is directly associated with a steering problem [17].

A valid planner provides a feasible path as the results that can be achieved by a set of allowed control inputs. In nonholonomic planning problems, the kinematic and/or dynamic constraints need to be taken into consideration, and possible obstacles involved should also be evaluated, which is handled by the collision checking and avoiding subsystem of the planner. Most of the nonholonomic planners are developed with a hierarchical design including a global planner and a local planner, respectively. The local planner only needs to provide a solution to the local boundary value problem (BVP) subject to possible local constraints, which does not necessarily depend on the global map information. Since local planning algorithms are repeatedly evaluated for each step, most of them are implemented with straightforward approaches that can save computational resources. On the other

hand, the final path connecting the initial and the goal configurations is obtained at global planning stage, which may require higher computational costs, especially for global path planning algorithms aiming to derive optimal or suboptimal results. In most cases, the kinematic and/or dynamic constraints of the robot are ignored in the global planning stage, where the environmental constraints are the main concern.

The local planning algorithm for nonholonomic problems plays a role of the BVP solver. Many efforts have been made to address this problem, and a common method is to discretize the action space and the result is a reduced set of possible action trajectories which drive the nonholonomic robot to a finite set of configurations. The reduced set of action trajectories comprises the motion primitives, while the resulting paths are the path primitives. The discretization based method greatly simplifies the nonholonomic planning problem, and the motion and path primitives obtained can be adapted to integrate with local planners. For example, the local motion of the robot can be optimized subject to the potential function or optimal control by choosing the best action among a given subset of motion primitives. These obtained local planners are flexible enough to integrate with global planning framework such as sampling based planner [71] or lattice based search algorithm [72]. On the other hand, the corresponding path primitives can also be directly applied to appropriate local planners. The path primitives can also be derived using a geometric approach. For instance, path primitives represented by cubic Bézier curves can be integrated with rapidly exploring random tree (RRT) to efficiently compute curvature bounded paths [73].

Compared with those less efficient methods such as potential fields [74] and variational methods [75], which can be easily trapped by a local minima, search based and sampling based planning algorithms are widely used due to the flexibility and effectiveness when handling nonholonomic planning problems. Search based methods requires a discretization stage of the configuration space in advance and each vertex of the graph represents a configuration of the vehicle. There are two major drawbacks of this kind of approaches. The most important one is the graph size, because the lattice or grid used by the planner has many more vertices compared with sampling based planners, it will certainly consume a significant amount

of computing resources. Even after preprocessing [76], a greater storage space is still demanded than that of a randomized roadmap data structure. On the other hand, the ends of all the path primitives should land in the center of the cells or coincide with the vertices. This adjustment procedure is error-prone and may involve position or orientation errors. Thus we are interested in sampling based planners, such as probabilistic roadmap (PRM) and rapidly exploring random tree (RRT). With respect to nonholonomic path planning, there is a large variety of sampling based methods exhibiting higher performance, ensuring asymptotic optimality, or offering advanced features, such as RRT-connect [77], RRT* (also referred to as optimal RRT) [78], KPIECE (Kinodynamic Planning by Interior-Exterior Cell Exploration) [79], etc. Most of these planners and other sampling based planners we have not mentioned are developed based on RRT, PRM, or designed with similar philosophy.

2.2.1 Probabilistic Roadmaps

Sampling based roadmap approach, often referred to as probabilistic roadmap (PRM), is a general path planning framework proposed in [80]. It serves as the global planner using a graph structure with multiple nodes precomputed before the query phase. Compared with the common RRT framework focusing on single-query problems, PRM is a very efficient algorithm for multiple-query purposes, which means that a number of pairs of (q_s, q_g) are given, while both the robot model and the environment are kept unchanged.

A typical PRM planning framework is divided into two phases [15]. Let a graph $\mathcal{G}(V, E)$ be the topological representation of the roadmap, where V is a set of vertices or nodes and E is the set of edges or paths in the configuration space \mathcal{C} . The preliminary requirement of these nodes and edges is that they should be free of collisions. During the construction or preprocessing phase, a graph is built from an empty set, and the basic process of PRM construction is illustrated by function `BUILDPRM`, which takes the total number of sampling nodes N as the input, as described in Algorithm 1.

Algorithm 1 Probabilistic Roadmap Construction

```
function BUILDPRM ( $N$ )
1:  $i \leftarrow 0$ 
2:  $\mathcal{G}.\text{Init}()$ 
3: while  $i < N$  do ▷  $N$  sampling nodes
4:   repeat ▷ find a collision-free node
5:      $q_i \leftarrow \text{RANDOMSAMPLE}(\mathcal{C})$ 
6:   until  $\text{VALIDVERTEX}(q_i) = \text{true}$ 
7:    $\mathcal{G}.\text{AddVertex}(q_i)$ 
8:    $i \leftarrow i + 1$ 
9:   for all  $q \in \text{NEIGHBORHOOD}(q_i, \mathcal{G})$  do ▷
10:    connect nodes with neighbors
11:    if  $\mathcal{G}.\text{SameComponent}(q_i, q) = \text{false}$  and  $\text{CONNECT}(q_i, q)$  then
12:       $\mathcal{G}.\text{AddEdge}(q_i, q)$ 
13:    end if
14:   end for
15: end while
16: return  $\mathcal{G}$ 
```

At the beginning of each iteration, a new configuration is sampled randomly in \mathcal{C} in function RANDOMSAMPLE. By default, a uniform sampling strategy is adopted, and the efficiency can be improved under specific situations with proper sampling strategies, such as obstacle-sensitive strategy [81], cost-sensitive adaptive strategy [82], and bridge-test strategy [83]. Once the sampled node q_i has been confirmed to be collision-free by the collision validation function VALIDVERTEX, it can be added to the graph \mathcal{G} , and effort will be made to connect the new configuration as a node with the existed neighboring nodes in the graph. It will be checked against its nearest neighbors in sequence, which are chosen or determined by NEIGHBORHOOD with respect to certain distance metric. For example, the most important category of metrics in motion planning over \mathbb{R}^n is referred to as

L_p metrics [84]:

$$\rho(q, q') = \left(\sum_{i=1}^n |q_i - q'_i|^p \right)^{\frac{1}{p}}. \quad (2.30)$$

The most popular ones are Manhattan metric ($p = 1$), Euclidean metric ($p = 2$), and L_∞ metric defined by

$$L_\infty(q, q') = \max_{1 \leq i \leq n} \{|q_i - q'_i|\}. \quad (2.31)$$

Basically, a valid node q_i and its neighbor q should not be already in the same components of the graph and the generated edge (q_i, q) should be collision free. Note that these edges are not necessarily straight lines, and in certain cases the condition that edges are required to be collision free against the obstacles can be released when the robot does not strictly follow the edge. After successful validation of the collision condition, the edge (q_i, q) is formed and added to the graph \mathcal{G} . After the specified number of nodes N is reached, the graph is constructed successfully and \mathcal{G} is returned. Such a PRM construction procedure is computationally intensive, but luckily, in most cases this phase can be completed offline, which is the reason why it is also referred to as preprocessing phase or learning phase. Once the graph \mathcal{G} is generated, it does not need to be recomputed or modified, as long as the kinematic model of the vehicle and the environment are not changed. Thus for applications involving multiple queries, PRM based approaches can be very efficient.

Different from the preprocessing phase, the query phase takes less computational resources. Based on the existed graph structure \mathcal{G} , a discrete search is performed given a query from q_s to q_g . A typical query procedure from [69] is illustrated in Algorithm 2.

Algorithm 2 Probabilistic Roadmap Query

function QUERYPRM(q_s, q_g, K, \mathcal{G})

1: $\mathcal{G}.\text{AddVertex}(q_s)$

2: $\mathcal{G}.\text{AddVertex}(q_g)$

3: $N_{q_s}, N_{q_g} \leftarrow \text{NEARESTK}(q_s, \mathcal{G}), \text{NEARESTK}(q_g, \mathcal{G})$

4: **repeat** \triangleright find the nearest neighbor of q_s in N_{q_s}

5: $q' \leftarrow \text{NEARESTNEIGHBOR}(q_s, N_{q_s})$

```
6: if VALIDEDGE( $q_s, q'$ ) then
7:    $\mathcal{G}$ .AddEdge( $q_s, q'$ )
8:   break
9: else
10:   $N_{q_s}$ .RemoveVertex( $q'$ )
11: end if
12: until  $q_s = \emptyset$ 
13: repeat                                 $\triangleright$  find the nearest neighbor of  $q_G$  in  $N_{q_G}$ 
14:   $q'' \leftarrow$  NEARESTNEIGHBOR( $q_G, N_{q_G}$ )
15:  if VALIDEDGE( $q_G, q''$ ) then
16:     $\mathcal{G}$ .AddEdge( $q_G, q''$ )
17:    break
18:  else
19:     $N_{q_G}$ .RemoveVertex( $q''$ )
20:  end if
21: until  $q_G = \emptyset$ 
22:  $\mathcal{P} \leftarrow \mathcal{G}$ .FindPath( $q_s, q_G$ )           $\triangleright$  find the shortest path
23: if  $\mathcal{P} \neq \emptyset$  then
24:   return  $\mathcal{P}$ 
25: end if
26: return failure
```

The first step is to connect the start and goal configurations, namely, q_s and q_G , to the graph \mathcal{G} . In Algorithm 2, a function NEARESTK is used to obtain the neighboring nodes, which searches for K closest points in the graph to a given vertex. Also other approaches, such as selection according to a specified radius or visibility property, can be used to select neighboring samples. Both q_s and q_G are checked against its neighbors before the corresponding edges are connected to the graph \mathcal{G} . With a graph containing q_s and q_G , a graph search algorithm is used to find a valid path, which may be subject to certain objective functions.

2.2.2 Rapidly Exploring Random Tree

Besides, PRM itself can also be used in single-query applications [85], which takes advantage of lazy evaluation collision checking. In most cases, a feasible path is found before a complete roadmap is constructed, which means that the planner does not need to fully explore the space. Actually, there exists a family of incremental sampling and search algorithms specifically designed for single-query applications. Among them, rapidly exploring random tree is the most basic and important one. In addition, at the beginning it was developed aiming to solve non-holonomic and kinodynamic planning problems [86], and has been widely used in autonomous robotic motion planning. In specific contexts, it is also referred to as rapidly exploring dense tree (RDT) to address the dense space covering capability. A basic procedure of building an RRT is illustrated in Algorithm 3.

Algorithm 3 Rapidly Exploring Random Tree Construction

function BUILD RRT(q_0, n)

- 1: $\mathcal{G}.\text{Init}(q_0)$
 - 2: **for** $i = 1$ **to** n **do**
 - 3: $q_{\text{rand}} \leftarrow \text{RANDOMSAMPLE}(\mathcal{C})$
 - 4: EXTENDRRT($\mathcal{G}, q_{\text{rand}}, \Delta q$)
 - 5: **end for**
 - 6: **return** \mathcal{G}
-

The scheme of the RRT algorithm is very concise and straightforward. Firstly, a tree structure \mathcal{G} starts with an initial configuration q_0 . Then during each iteration, a configuration q_{rand} is randomly sampled in configuration space \mathcal{C} . Then an EXTENDRRT function handles all the remaining procedures to expand this tree according to the information provided by q_{rand} , and n is the number of vertices in this RRT. In the basic RRT framework, a distance Δq is specified in addition to the randomly sampled vertex q_{rand} , as illustrated in Algorithm 4.

Algorithm 4 Rapidly Exploring Random Tree Extension

```
function EXTENDRRT ( $\mathcal{G}$ ,  $q_{\text{rand}}$ ,  $\Delta q$ )
1:  $q_{\text{near}} \leftarrow \text{NEARESTNEIGHBOR}(q_{\text{rand}}, \mathcal{G})$ 
2:  $q_{\text{new}} \leftarrow \text{NEWCONF}(q_{\text{near}}, q_{\text{rand}}, \Delta q)$ 
3: if VALIDVERTEX( $q_{\text{new}}$ ) then
4:    $\mathcal{G}.\text{AddVertex}(q_{\text{new}})$ 
5:    $\mathcal{G}.\text{AddEdge}(q_{\text{near}}, q_{\text{new}})$ 
6: return  $\mathcal{G}$ 
7: end if
8: return failure
```

q_{near} is the nearest neighbor of q_{rand} in the tree \mathcal{G} . The new node to be expanded q_{new} is obtained based on the distance between q_{near} and q_{rand} : $q_{\text{new}} = q_{\text{rand}}$ if $\rho(q_{\text{near}} - q_{\text{rand}}) \leq \Delta q$; Otherwise, based on (2.30), q_{new} is an interpolation point on the edge $(q_{\text{near}}, q_{\text{rand}})$ with $\rho(q_{\text{near}} - q_{\text{new}}) = \Delta q$. This process is contained in function NEWCONF. Finally, the obtained new vertex q_{new} along with the new edge $(q_{\text{near}}, q_{\text{new}})$ is added to the tree after validating collision conditions.

Chapter 3

Vehicle Model and Local Path Analysis

3.1 Generalization of Kinematic Model of Wheeled Mobile Robots

Wheeled mobile robots (WMRs) are widely used for their higher efficiency in view of energy cost and mobility compared with other types of locomotion mechanisms. Performing a comprehensive analysis of a vehicle can be a very complex task when considering the detailed mechanisms of a specific system. As a result, it is a common practice to simplify the vehicle system to an equivalent kinematic bicycle model, which is also referred to as single-track model. This chapter begins with a comprehensive kinematic analysis of different types of wheels, and then derives equivalent simplified vehicle models for common single-unit vehicles with Ackermann steering geometry, before finally obtaining a generalized kinematic model for the bicycle model.

3.1.1 Mobility Analysis

Wheels used by most conventional WMRs can be classified into three types: Orientable or steerable wheels, fixed wheels, and castor or off-centered steerable wheels. To clarify the difference between them, we assume that $Vx_Vy_Vz_V$ is the coordinate system attached to a point V fixed to the robot frame, where the motion of the wheel B_i is in the direction of v_i . At the same time, the distance from

the reference point V to the center of the wheel is defined as l_i while the orientation of the axle is denoted by α_i , as shown in Fig. 3.1. In such a way, we can locate the position of the wheel B_i in polar coordinates conveniently.

- (i) Orientable or steerable wheel: As illustrated in Fig. 3.1a, the orientation of a steerable wheel B_i is determined by the rotation of its orientation axis, which intersects with the axle as projected on the wheel's plane. The wheel orientation is calculated by $\alpha_i + \beta_i$, where β_i is the relative orientation of the wheel with respect to the spin axis or axle and is a function of time.
- (ii) Fixed wheel: The axle of such a wheel is fixed to the body frame of the mobile robot and the fixed wheel has only one degree of freedom (DOF) with respect to the axle. Its position and orientation are completely dependent on the robot movement, thus the relative angle γ_i is a constant (also refer to Fig. 3.1a).
- (iii) Castor or off-centered steerable wheel: The steering axis A_i of a castor wheel has an offset d_i from the wheel center B_i . The wheel movement is in the same direction with $\overrightarrow{A_iB_i}$, as illustrated in Fig. 3.1b. In fact, the orientable wheel can be treated as a special case of the castor wheel whose orientation axis has zero offset from the wheel center.

To obtain the velocity \vec{v}_{c_i} of any contact point C_i of the wheel B_i , we begin with analyzing a general wheel model with all the parameters included. Let $B_i x_i y_i z_i$ represent the local coordinate system with the wheel itself as the reference frame, and \vec{x}_i be the same direction with \vec{v}_i shown in Fig. 3.1b while \vec{z}_i aligns with \vec{z}_v . θ is the robot orientation or heading with respect to the horizontal axis in the global coordinate system, \vec{v}_v is the velocity of the robot frame, and φ_i denotes the rotation angle of the wheel with respect to its axle. Consequently, we have

$$\vec{v}_{c_i} = \vec{v}_v + \dot{\theta} \vec{z}_v \wedge \overrightarrow{VC_i} + \dot{\beta}_i \vec{z}_v \wedge \overrightarrow{A_iC_i} + \dot{\varphi}_i \vec{y}_i \wedge \overrightarrow{B_iC_i}. \quad (3.1)$$

A concrete representation of posture or pose of the vehicle includes the position and orientation information [10], i.e., $\boldsymbol{\xi} = (x, y, \theta)^T$. Accordingly, by letting $\psi_i =$

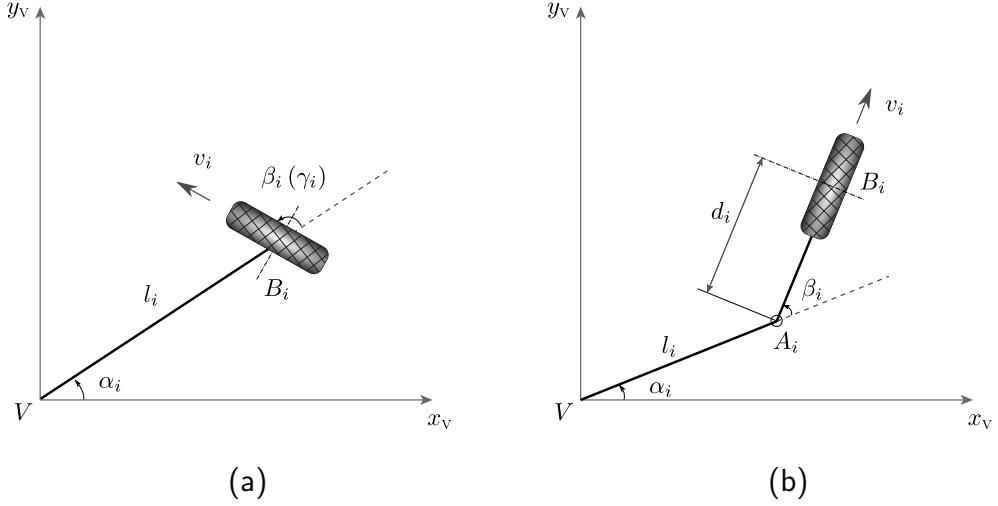


Figure 3.1: Wheel types: (a) Steerable and fixed wheel model. (b) Castor wheel model.

$\alpha_i + \beta_i + \gamma_i$, (3.1) can be represented by its tangential and normal components separately:

$$\begin{pmatrix} v_t \\ v_n \end{pmatrix} = \begin{pmatrix} \cos \psi_i & \sin \psi_i & d_i \sin \gamma_i + l_i \sin (\beta_i + \gamma_i) \\ -\sin \psi_i & \cos \psi_i & d_i \cos \gamma_i + l_i \cos (\beta_i + \gamma_i) \end{pmatrix} \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} + \begin{pmatrix} \sin \gamma_i \\ \cos \gamma_i \end{pmatrix} d_i \dot{\beta}_i - \begin{pmatrix} r_i \\ 0 \end{pmatrix} \dot{\varphi}_i, \quad (3.2)$$

where r_i is the wheel radius and

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

represents a rotation transformation about the internal frame.

The pure rolling condition requires $v_t = 0$ and the nonslip condition requires $v_n = 0$. According to (3.2), by setting $d_i=0$ and $\gamma_i=0$, we can obtain the kinematic constraints of pure rolling without slipping for steerable wheels as:

$$\begin{pmatrix} \cos \psi_i & \sin \psi_i & l_i \sin \beta_i \\ -\sin \psi_i & \cos \psi_i & l_i \cos \beta_i \end{pmatrix} \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} - \begin{pmatrix} r_i \\ 0 \end{pmatrix} \dot{\varphi}_i = \mathbf{0}. \quad (3.4)$$

For fixed wheels, by setting $d_i = 0$ and $\beta_i = 0$, the constraints reduce to

$$\begin{pmatrix} \cos \psi_i & \sin \psi_i & l_i \sin \gamma_i \\ -\sin \psi_i & \cos \psi_i & l_i \cos \gamma_i \end{pmatrix} \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} - \begin{pmatrix} r_i \\ 0 \end{pmatrix} \dot{\varphi}_i = \mathbf{0}. \quad (3.5)$$

With regard to the castor wheel, by setting $\gamma_i = 0$ in (3.2) the constraints are expressed as:

$$\begin{pmatrix} \cos \psi_i & \sin \psi_i & l_i \sin \beta_i \\ -\sin \psi_i & \cos \psi_i & d_i + l_i \cos \beta_i \end{pmatrix} \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} d_i \dot{\beta}_i - \begin{pmatrix} r_i \\ 0 \end{pmatrix} \dot{\varphi}_i = \mathbf{0}. \quad (3.6)$$

Let the total number of wheels be $N_w = N_s + N_f + N_c$, where the subscripts s , f , and c denote three types of wheels, i.e., steerable wheels, fixed wheels, and castor wheels as mentioned above. According to [87], the configuration of a WMR can be fully described by the following vectors:

- (i) Pose vector of dimension 3: $\boldsymbol{\xi}(t) = (x(t), y(t), \theta(t))^T$
- (ii) Orientation vector of dimension $N_s + N_c$: $\boldsymbol{\beta}(t) = (\boldsymbol{\beta}_s(t), \boldsymbol{\beta}_c(t))^T$, where $\boldsymbol{\beta}_s(t) = (\beta_{s,1}(t), \beta_{s,2}(t), \dots, \beta_{s,N_s}(t))^T$ and $\boldsymbol{\beta}_c(t) = (\beta_{c,1}(t), \beta_{c,2}(t), \dots, \beta_{c,N_c}(t))^T$
- (iii) Rotation vector of dimension N_w : $\boldsymbol{\varphi}(t) = (\boldsymbol{\varphi}_s(t), \boldsymbol{\varphi}_f(t), \boldsymbol{\varphi}_c(t))^T$, where $\boldsymbol{\varphi}_s(t) = (\varphi_{s,1}(t), \varphi_{s,2}(t), \dots, \varphi_{s,N_s}(t))^T$, $\boldsymbol{\varphi}_f(t) = (\varphi_{f,1}(t), \varphi_{f,2}(t), \dots, \varphi_{f,N_f}(t))^T$, and $\boldsymbol{\varphi}_c(t) = (\varphi_{c,1}(t), \varphi_{c,2}(t), \dots, \varphi_{c,N_c}(t))^T$

Thus, the configuration of the robot body together with all its wheels are specified as $(\boldsymbol{\xi}(t), \boldsymbol{\beta}(t), \boldsymbol{\varphi}(t))^T$, which is a vector of dimension $N_f + 2N_s + 2N_c + 3$. The pure rolling condition for the whole system enforces that all the wheels satisfy their corresponding pure rolling constraints at each contact point by setting $v_t = 0$ in (3.4), (3.5) and (3.6):

$$\mathbf{J}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} + \mathbf{J}_2 \dot{\boldsymbol{\varphi}} = \mathbf{0}, \quad (3.7)$$

where

$$\mathbf{J}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) = \begin{pmatrix} \mathbf{J}_{1,f} \\ \mathbf{J}_{1,s}(\boldsymbol{\beta}_s) \\ \mathbf{J}_{1,c}(\boldsymbol{\beta}_c) \end{pmatrix}, \quad \mathbf{J}_2 = -r_i \mathbf{I}_{N_w \times N_w}. \quad (3.8)$$

Here, $\mathbf{J}_{1,f}$, $\mathbf{J}_{1,s}(\boldsymbol{\beta}_s)$, and $\mathbf{J}_{1,c}(\boldsymbol{\beta}_c)$ are matrices of dimensions $N_f \times 3$, $N_s \times 3$, and $N_c \times 3$ respectively. \mathbf{J}_2 is a diagonal matrix of dimension $N_w \times N_w$ and all the wheels are assumed to have the same radius r_i . Moreover, $\dot{\boldsymbol{\varphi}}$ is the N_w -dimensional angular velocity vector of all the wheels while $\mathbf{R}(\theta)$ is expressed in (3.3).

Similarly, the matrix representation of the nonslip condition for the whole system is:

$$\mathbf{C}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} + \mathbf{C}_2 \dot{\boldsymbol{\beta}}_c = \mathbf{0}, \quad (3.9)$$

where

$$\mathbf{C}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) = \begin{pmatrix} \mathbf{C}_{1,f} \\ \mathbf{C}_{1,s}(\boldsymbol{\beta}_s) \\ \mathbf{C}_{1,c}(\boldsymbol{\beta}_c) \end{pmatrix}, \quad \mathbf{C}_2 = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{C}_{2,c} \end{pmatrix}. \quad (3.10)$$

Analogous to (3.8), $\mathbf{C}_{1,f}$, $\mathbf{C}_{1,s}(\boldsymbol{\beta}_s)$, and $\mathbf{C}_{1,c}(\boldsymbol{\beta}_c)$ are matrices of dimensions $N_f \times 3$, $N_s \times 3$, and $N_c \times 3$ respectively, while \mathbf{C}_2 is a matrix of dimension $N_w \times N_c$ and $\mathbf{C}_{2,c} = \text{diag}(d_1, d_2, \dots, d_{N_c})$.

With (3.7), we are able to satisfy the pure rolling constraint for all types of wheels by directly specifying

$$\dot{\boldsymbol{\varphi}} = \frac{1}{r_i} \mathbf{J}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) \mathbf{R}(\theta) \dot{\boldsymbol{\xi}}. \quad (3.11)$$

Based on (3.9), the nonslip constraints for both the steerable and fixed wheels are simplified to

$$\mathbf{C}_1^*(\boldsymbol{\beta}_s) \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} = \mathbf{0}, \quad (3.12)$$

i.e.,

$$\mathbf{R}(\theta) \dot{\boldsymbol{\xi}} \in \ker(\mathbf{C}_1^*(\boldsymbol{\beta}_s)), \quad \mathbf{C}_1^*(\boldsymbol{\beta}_s) = \begin{pmatrix} \mathbf{C}_{1,f} \\ \mathbf{C}_{1,s}(\boldsymbol{\beta}_s) \end{pmatrix}. \quad (3.13)$$

On the other hand, the nonslip constraint for castor wheels can be satisfied by setting

$$\dot{\boldsymbol{\beta}}_c = -\mathbf{C}_{2,c}^{-1} \mathbf{C}_{1,c}(\boldsymbol{\beta}_c) \mathbf{R}(\theta) \dot{\boldsymbol{\xi}} \quad (3.14)$$

in (3.9), where $\mathbf{C}_{2,c}^{-1} = \text{diag}(d_1^{-1}, d_2^{-1}, \dots, d_{N_c}^{-1})$.

The mobility of a WMR can be characterized by the degree of mobility δ_m and the degree of steerability δ_s introduced in [88], which are defined by

$$\begin{aligned} \delta_m &= 3 - \text{rank}(\mathbf{C}_1^*(\boldsymbol{\beta}_s)), \\ \delta_s &= \text{rank}(\mathbf{C}_{1,s}(\boldsymbol{\beta}_s)). \end{aligned} \quad (3.15)$$

From (3.13), we know that $\delta_m < 3$ if the mobile robot has wheels of fixed or steerable types, while $\delta_s \leq 2$ if the instantaneous center of rotation (ICR) exists.

3.1.2 Kinematic Models

3.1.2.1 Front-Wheel Steering Vehicle

Though there are many configurations of wheeled mobile robots, the most commonly encountered type is the vehicle with four wheels, and the robots with similar structures are known to be car-like robots. Considering a practical front-wheel steering (FWS) vehicle shown in Fig. 3.2a with Ackermann steering mechanism named after Rudolph Ackermann [10], which is the most exclusively used steering system by modern vehicles [11]. The main benefit of adopting the Ackermann

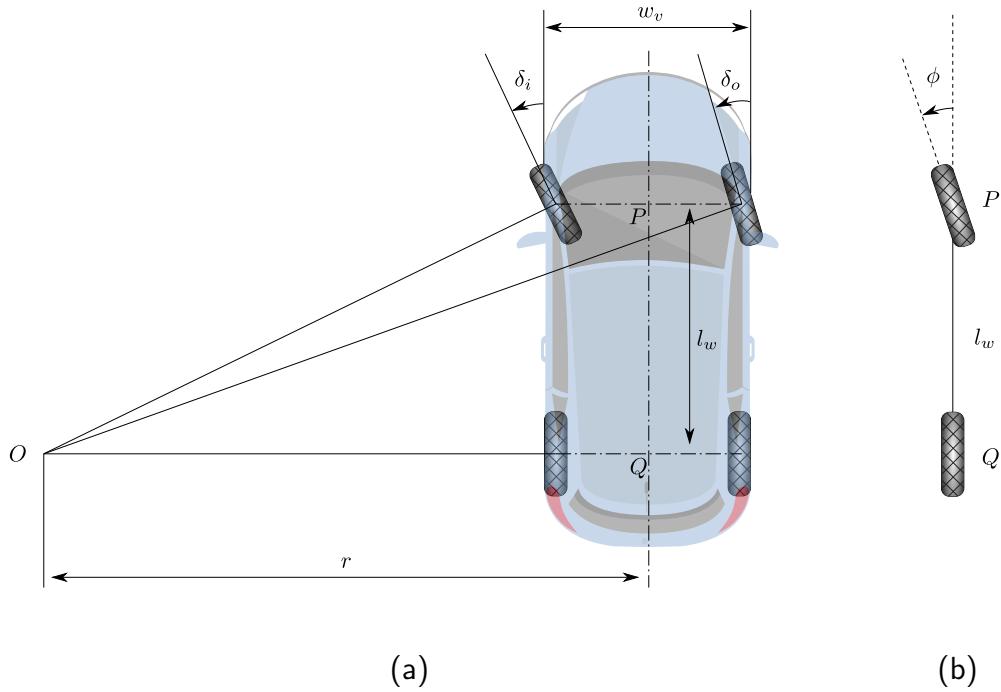


Figure 3.2: (a) A FWS vehicle model with Ackermann steering system. (b) Equivalent bicycle model.

geometry is the ability to avoid the slippage when following a curved path. Specifically, the rear wheels are fixed while two front wheels are steered with different angles [89]. At any moment, the inner wheel and outer wheel are on two concentric circles and the common center is the instantaneous center of rotation O . Here, to guarantee slip-free steering, the inner wheel turns further than the outer wheel,

thus, the steering angles should satisfy $\delta_i > \delta_o$. More precisely, the relationship between these two angles is

$$\cot \delta_o - \cot \delta_i = \frac{w_v}{l_w}, \quad (3.16)$$

where w_v is the vehicle width and l_w is the wheelbase of the vehicle. Such a model can be simplified into a kinematically equivalent bicycle model (Fig. 3.2b) and the two wheels on the same axle are reduced to a single one located at the midpoint of the axle, namely, P and Q . It can be verified that the virtual wheel P is also on a concentric circle with center O . With this model, the turning radius r and equivalent steer angle ϕ are computed easily as

$$\begin{aligned} r &= l_w \cot \phi, \\ \phi &= \arctan \frac{2}{\cot \delta_o + \cot \delta_i}. \end{aligned} \quad (3.17)$$

Moreover, it indicates that (3.16) and (3.17) also hold for a rear-wheel steering (RWS) vehicle model [90].

Alternatively, if the angular velocities of the drive wheels are specified instead of steering angles, we can still obtain the kinematic relationships. For a rear-wheel drive (RWD) vehicle with ω_i , ω_o as the angular velocities of the inner and outer rear wheels, we have

$$\frac{\omega_i r_i}{r - \frac{w_v}{2}} = \frac{\omega_o r_i}{r + \frac{w_v}{2}} = \frac{\omega_Q r_i}{r} \quad (3.18)$$

based on the geometric constraints, where r_i is the wheel radius and ω_Q is the equivalent angular velocity at point Q . Thus the kinematic relationships can be expressed alternatively as:

$$\begin{aligned} \delta_i &= \arctan \left(\frac{l_w}{w_v} \left(\frac{\omega_o}{\omega_i} - 1 \right) \right), \\ \delta_o &= \arctan \left(\frac{l_w}{w_v} \left(1 - \frac{\omega_i}{\omega_o} \right) \right), \\ r &= \frac{w_v (\omega_o + \omega_i)}{2 (\omega_o - \omega_i)}, \end{aligned} \quad (3.19)$$

and the equivalent velocity of Q is

$$v_Q = \frac{r_i}{2} (\omega_i + \omega_o). \quad (3.20)$$

With regard to a front-wheel drive (FWD) vehicle with ω_i , ω_o as the angular velocities of the inner and outer front wheels, we can obtain the following constraints:

$$\frac{\omega_i r_i}{\frac{l_w}{\sin \delta_i}} = \frac{\omega_o r_i}{\frac{l_w}{\sin \delta_o}} = \frac{\omega_P r_i}{\frac{l_w}{\sin \phi}}, \quad (3.21)$$

where ω_P is the equivalent angular velocity at point P . Accordingly, δ_i , δ_o , and r can be derived from the following equations:

$$\begin{aligned} \omega_P \sin \phi &= \omega_i \sin \delta_i = \omega_o \sin \delta_o, \\ w_v &= l_w (\cot \delta_o - \cot \delta_i), \\ r &= \frac{l_w}{2} (\cot \delta_i + \cot \delta_o), \\ v_P &= \omega_P r_i, \end{aligned} \quad (3.22)$$

where v_P is the equivalent velocity at point P in FWD mode.

3.1.2.2 Multi-axle Vehicle

Some Ackermann steering systems are designed to have multiple axles for legal weight restriction or specific design reasons. As an example, a three-axle vehicle with two steerable axles is shown in Fig. 3.3. This kind of multi-axle vehicle can only have one non-steerable axle to guarantee steerability without slipping, and all the wheels are required to share the same center of rotation at any moment as stated in [91] and [92]. The three-axle vehicle of width w_v in Fig. 3.3a can be regarded as a combination of two Ackermann steering structures, which are dependent on each other [90]. Point O is the ICR and r is the turning radius. We use $l_{w,1}$ and $l_{w,2}$ to express the equivalent wheelbases in these two steering subsystems, while the steering angles of inner and outer steerable wheels are denoted as $\delta_{i,1}$, $\delta_{i,2}$ and $\delta_{o,1}$, $\delta_{o,2}$ respectively. So, we have

$$\begin{aligned} \cot \delta_{o,1} - \cot \delta_{i,1} &= \frac{w_v}{l_{w,1}}, \\ \cot \delta_{o,2} - \cot \delta_{i,2} &= \frac{w_v}{l_{w,2}}. \end{aligned} \quad (3.23)$$

Since the two steering subsystems are restricted by a common ICR, this kinematic model can be simplified to a bicycle model PQ by choosing either one of the

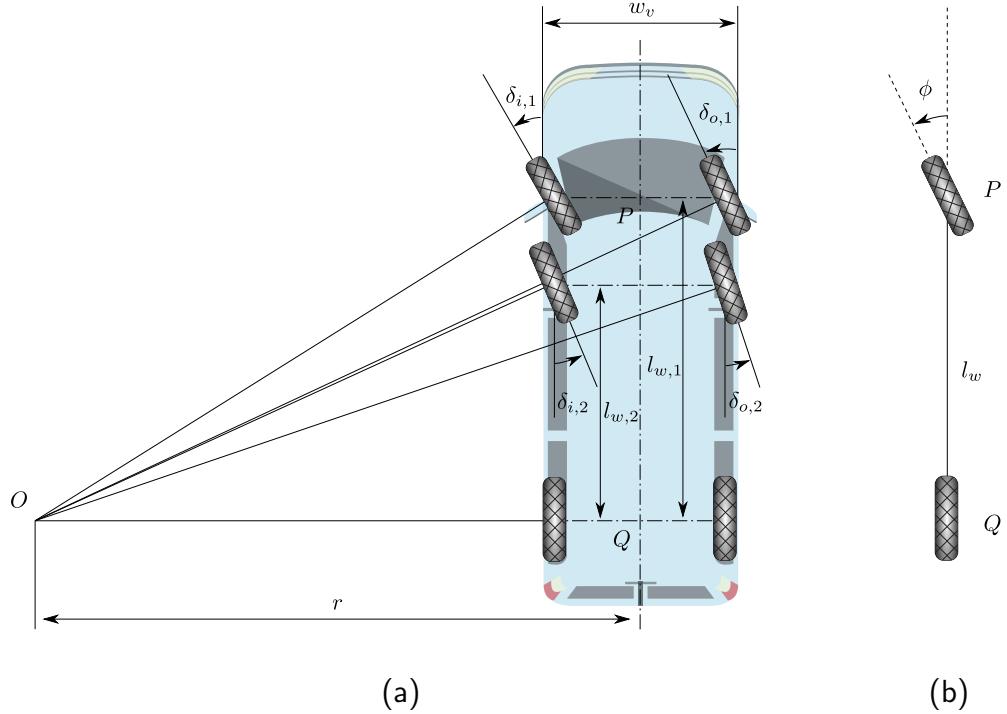


Figure 3.3: Multi-axle vehicle: (a) Model with only one non-steerable axle. (b) Equivalent bicycle model.

Ackermann steering subsystems, e.g., the first and the rear axles ($l_w = l_{w,1}$), as the reference framework. As such, the remaining process is the same as that of the FWS model with four wheels (Fig. 3.2) analyzed before.

On the other hand, in practice there exist other configurations of multi-axle vehicles, such as the three-axle vehicle model shown in Fig. 3.4. It is difficult to clearly define its kinematic steering condition for such a model with multiple non-steerable axles because the nonslip assumption is no longer applicable. For simplification, the steering mechanism can be intentionally designed to make the ICR on a lateral line between the rear axles. The kinematic analysis is the same with model in Fig. 3.2 by using the geometric wheelbase l_w , which is the distance from the steering axle to the tandem center referred to as midline, i.e., the equivalent rear axle [90]. In [93], a more precise model is developed based on the steady-state handling equation proposed by [94], and the equivalent wheelbase l'_w

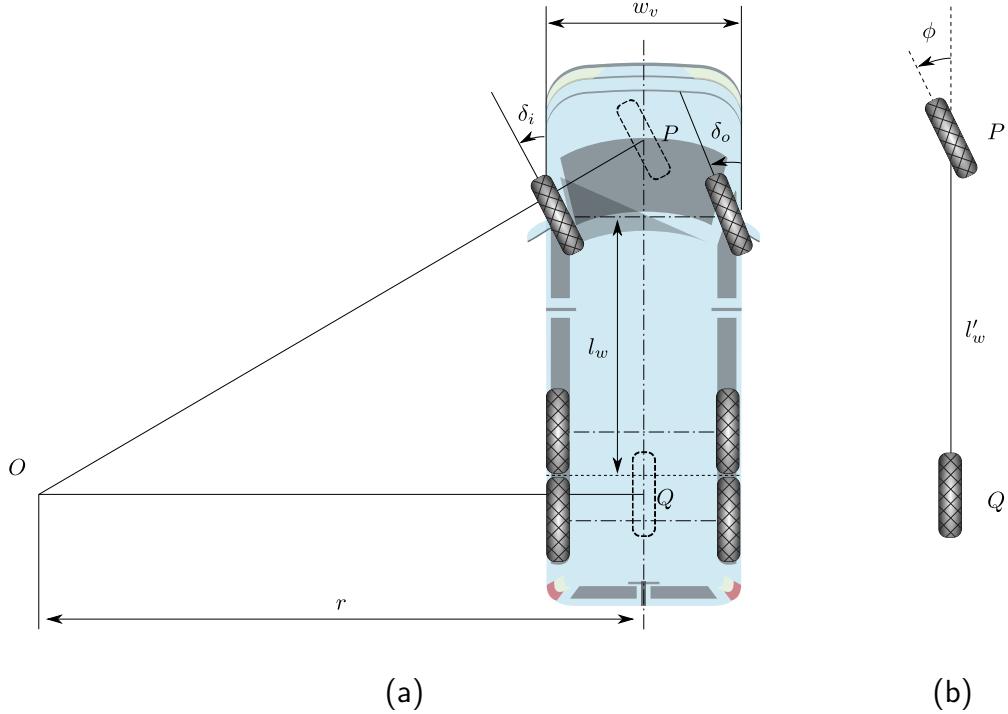


Figure 3.4: Multi-axle vehicle: (a) Model with multiple non-steerable axles. (b) Equivalent slip-free bicycle model.

is computed by

$$l'_w = l_w \left(1 + \frac{T}{l_w^2} \left(1 + \frac{C_{\alpha r}}{C_{\alpha f}} \right) \right), \text{ where } T = \frac{\sum_{i=1}^N \Delta_i^2}{N}, \quad (3.24)$$

and $C_{\alpha f}$ and $C_{\alpha r}$ are the sums of the cornering stiffness of all front and rear tires, N is the total number of non-steerable axles, and Δ_i is the distance from the i -th non-steerable axle to the midline. The equivalent bicycle model (Fig. 3.4) has the same front axle steering angle ϕ and turning radius r with the actual vehicle, which are calculated by (3.17).

3.1.2.3 Four-Wheel Steering Vehicle

Last but not least, to improve steering response, four-wheel steering (4WS) is also one of the systems employed by some vehicles. Such kind of system also has the advantages of gaining increased vehicle stability at high speed as well as allowing small turning radius at low speed. In a 4WS system, all the wheels turn

simultaneously as the driver steers, and the kinematic steering condition that a common ICR has to be shared should be satisfied at low speed. The 4WS system has two different states: The front and rear wheels are steered in the same direction in a positive 4WS configuration (Fig. 3.5a), while the front and rear wheels are steered in opposite directions in a negative 4WS configuration (Fig. 3.5b) [90]. For both configurations, each of them can be regarded as a combination of two

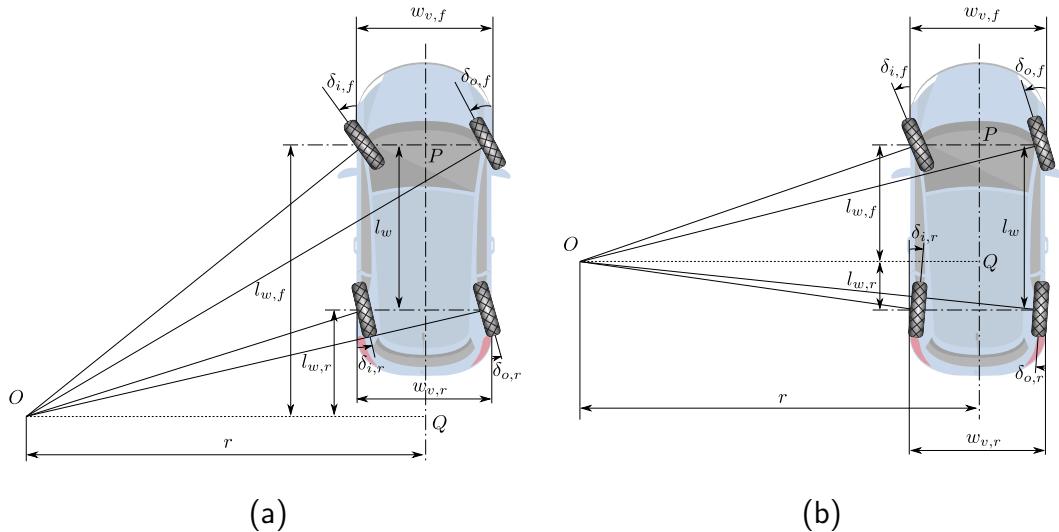


Figure 3.5: 4 wheel-steering vehicle: (a) Positive configuration. (b) Negative configuration.

interrelated Ackermann steering subsystems analogous to the three-axle vehicle model with only one non-steerable axle in Fig. 3.3. Without loss of generality, let the widths of the front and rear axles be $w_{v,f}$ and $w_{v,r}$ respectively. $l_{w,f}$ and $l_{w,r}$ are the distances from the position of the equivalent rear axle to the front and rear axles, while l_w represents the wheelbase of the vehicle, i.e., the actual distance between two axles. $\delta_{i,f}$, $\delta_{o,f}$ are the steering angles of front wheels and $\delta_{i,r}$, $\delta_{o,r}$ denote the steering angles of rear wheels. Based on the geometric relationships in Fig. 3.5, we have

$$\begin{aligned} \tan \delta_{i,f} &= \frac{l_{w,f}}{r - \frac{w_{v,f}}{2}}, & \tan \delta_{o,f} &= \frac{l_{w,f}}{r + \frac{w_{v,f}}{2}}, \\ \tan \delta_{i,r} &= \frac{l_{w,r}}{r - \frac{w_{v,r}}{2}}, & \tan \delta_{o,r} &= \frac{l_{w,r}}{r + \frac{w_{v,r}}{2}}, \end{aligned} \quad (3.25)$$

where r is the turning radius. The wheelbases of positive and negative configurations are calculated as $l_w = l_{w,f} - l_{w,r}$ and $l_w = l_{w,f} + l_{w,r}$, respectively. Let l'_w be the equivalent wheelbase and we can choose $l'_w = l_{w,f}$ for both configurations. Thus similar to (3.17), r and the equivalent steering angle ϕ can be computed as.

$$\begin{aligned} r &= \frac{w_{v,f} \sin(\delta_{i,f} + \delta_{o,f})}{2 \sin(\delta_{i,f} - \delta_{o,f})}, \\ \phi &= \arctan\left(\frac{2 \sin \delta_{i,f} \sin \delta_{o,f}}{\sin(\delta_{i,f} + \delta_{o,f})}\right), \end{aligned} \quad (3.26)$$

and the equivalent wheelbase l'_w , which equals the distance from P to Q in the corresponding bicycle model, is obtained as

$$l'_w = \frac{w_{v,f} \sin \delta_{i,f} \sin \delta_{o,f}}{\sin(\delta_{i,f} - \delta_{o,f})}. \quad (3.27)$$

In a passive 4WS system, the 4WS factor $c_s = \frac{l_{w,2}}{l_{w,1}}$ keeps unchanged [90], thus the equivalent wheelbase l'_w is regarded as a constant that the steering angles of rear wheels are adjusted by the control system to coordinate the steering angles of the front wheels. Instead, if we set $l'_w = l_{w,r}$, these parameters can be computed as

$$\begin{aligned} r &= \frac{w_{v,r} \sin(\delta_{i,r} + \delta_{o,r})}{2 \sin(\delta_{i,r} - \delta_{o,r})}, \\ \phi &= \arctan\left(\frac{2 \sin \delta_{i,r} \sin \delta_{o,r}}{\sin(\delta_{i,r} + \delta_{o,r})}\right), \\ l'_w &= \frac{w_{v,r} \sin \delta_{i,r} \sin \delta_{o,r}}{\sin(\delta_{i,r} - \delta_{o,r})}. \end{aligned} \quad (3.28)$$

3.1.3 Generic Posture Kinematic Model

With appropriate transformations and simplifications proposed in Section 3.1.2, we are able to convert a common single-unit vehicle to a unified slip-free bicycle model with equivalent wheelbase as well as the steering angle. As Section 3.1.1 pointed out that the mobility of a vehicle can be characterized by a combination of degree of mobility and degree of steerability, namely, (δ_m, δ_s) , based on the number and type of the wheels possessed by this vehicle, we can derive corresponding generic posture kinematic representations for the simplified vehicles and bicycle models

illustrated in Section 3.1.2. According to [88], a WMR which can be simplified to a bicycle model with one conventional steerable wheel and one non-steerable wheel. As a result, all the vehicles mentioned above are of type (1, 1). Notice that a general 4WS vehicle belongs to type (1, 2) because the orientation of the remaining wheel provides one more degree of steerability, however, the passive 4WS vehicle we are interested in is still of type (1, 1) due to the restriction of a constant 4WS factor c_s .

The relationship (3.13) can be expressed explicitly as:

$$\mathbf{R}(\theta) \dot{\boldsymbol{\xi}} = \boldsymbol{\Omega}(\boldsymbol{\beta}_s) \mathbf{u}_m, \quad (3.29)$$

where the $\boldsymbol{\Omega}(\boldsymbol{\beta}_s)$ is a matrix of δ_m columns which forms a basis of $\ker(\mathbf{C}_1^*(\boldsymbol{\beta}_s))$, and \mathbf{u}_m is a vector of δ_m dimension. The differential equation of the posture kinematic model is:

$$\begin{pmatrix} \dot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{\beta}}_s \end{pmatrix} = \begin{pmatrix} \mathbf{R}(\theta) \boldsymbol{\Omega}(\boldsymbol{\beta}_s) & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{u}_m \\ \mathbf{u}_s \end{pmatrix}. \quad (3.30)$$

For a general vehicle model with only one fixed wheel and one steerable wheel, we have $N_f = N_s = 1$, $N_c = 0$, $\mathbf{C}_1(\boldsymbol{\beta}_s, \boldsymbol{\beta}_c) = \mathbf{C}_1^*(\boldsymbol{\beta}_s) = \begin{pmatrix} -\sin \psi_f \cos \psi_f & l_f \cos \gamma_f \\ -\sin \psi_s \cos \psi_s & l_s \cos \beta_s \end{pmatrix}$, and

$$\boldsymbol{\Omega}(\boldsymbol{\beta}_s) = \begin{pmatrix} l_f \cos \gamma_f \cos \psi_s - l_s \cos \beta_s \cos \psi_f \\ l_f \cos \gamma_f \sin \psi_s - l_s \cos \beta_s \sin \psi_f \\ \sin(\psi_f - \psi_s) \end{pmatrix} \quad (3.31)$$

based on (3.9) and (3.12), where $\psi_f = \alpha_f + \gamma_f$ and $\psi_s = \alpha_s + \beta_s$.

Considering the bicycle model generalized for all these vehicles, we assume that the orientation of the vehicle is θ and two wheels are attached to the body frame so that $N_f = N_s = 1$ and $N_c = 0$. Also we have $\gamma_f = \frac{\pi}{2}$, $\psi_f = 0$, $l_f = 0$, $\beta_s = \phi$, $\psi_s = \phi + \pi$, and $l_s = l_w$, where ϕ and l_w are the steering angle and wheelbase of the bicycle model respectively. Based on (3.31), the kinematic model can be computed by substituting these conditions into (3.30) after assigning an appropriate coefficient to $\boldsymbol{\Omega}(\boldsymbol{\beta}_s)$:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} -\cos \theta & 0 \\ \sin \theta & 0 \\ \frac{\tan \phi}{l_w} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_m \\ u_s \end{pmatrix}. \quad (3.32)$$

We will show that by replacing θ with $\pi - \theta$, the result (3.32) is transformed to the bicycle model (3.37) developed in Section 3.1.4. By taking the angular acceleration instead of angular velocity as the input, another model (3.40) is derived, which is identical to the system proposed in [20]. These models are formulated based on the assumption that the rear wheel of the bicycle model is attached to the reference coordinate system. Alternatively, by applying other reference frames to (3.30), different kinematic models can be obtained, such as (3.38) and (3.41) in Section 3.1.4.

3.1.4 Model of the Wheeled Mobile Robot

Starting from analyzing the kinematic property of different types of wheels, we have reviewed the feasibility and rationality of the kinematic simplification approach with regard to different steering configurations. Consequently, for the mobile robot considered here, we directly adopt the kinematically equivalent bicycle model [89], which is shown in Fig. 3.6. Different from the Dubins model [19], here the speed v and angular acceleration σ are regarded as control variables, which was initially proposed in [20]. The bicycle model is a skeleton representation of the four-wheeled robot whose front and rear wheels are simplified by two distinct wheels in the middle of the axles. In Fig. 3.6a, P and Q are the positions of the front and rear wheels while their velocities can be expressed in two dimensions as shown in Fig. 3.6b. O is called the instantaneous center of rotation (ICR) which is the point of intersection of the normal components v_P^n and v_Q^n . The distance from the ICR to each wheel is the turning radius denoted as r_P and r_Q respectively. With the bicycle model, the pose of the mobile robot can be specified by the generalized coordinates $\mathbf{q} = (x, y, \theta, \phi)^T$, where (x, y) are the Cartesian coordinates of a specific wheel P or Q , θ is the orientation (angle between the tangential velocity v_Q^t and the x axis), and ϕ is the steering angle of the front wheel P (angle between v_Q^t and v_P^t). Considering the rolling without slipping constraint, the normal component of the velocities v_P^n and v_Q^n should be zero:

$$\begin{cases} v_P^n = \dot{x}_P \sin(\theta + \phi) - \dot{y}_P \cos(\theta + \phi) = 0, \\ v_Q^n = \dot{x}_Q \sin \theta - \dot{y}_Q \cos \theta = 0, \end{cases} \quad (3.33)$$

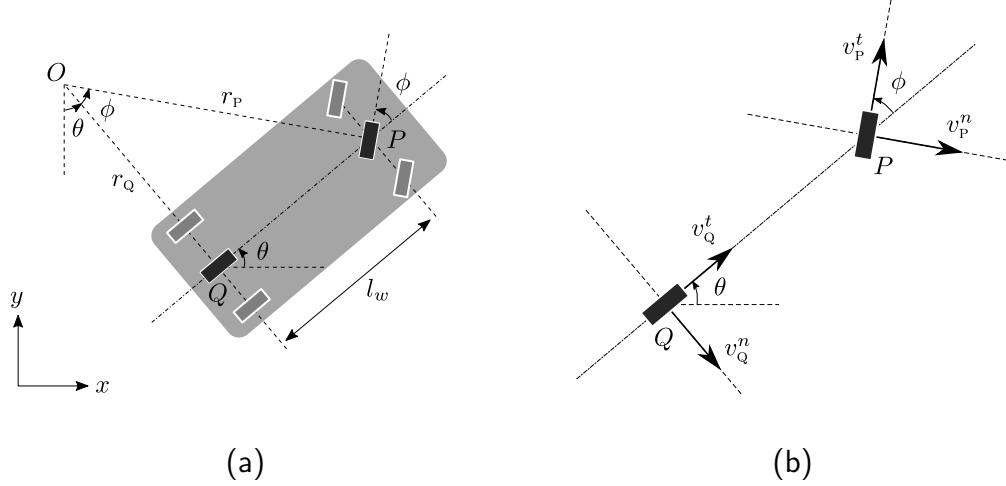


Figure 3.6: Kinematic model of a four-wheeled mobile robot: (a) The bicycle model representation. (b) Velocities of front wheel P and rear wheel Q .

where (x_P, y_P) and (x_Q, y_Q) are the coordinates of point P and Q respectively and have the following relationship with respect to the wheelbase l_w :

$$\begin{cases} x_P - x_Q = l_w \cos \theta, \\ y_P - y_Q = l_w \sin \theta. \end{cases} \quad (3.34)$$

Thus based on (3.33) and (3.34), the Pfaffian constraints [69] of the bicycle model can be obtained in matrix form $\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0}$ as

$$\begin{pmatrix} \sin(\theta + \phi) & -\cos(\theta + \phi) & -l_w \cos \phi & 0 \\ \sin \theta & -\cos \theta & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \mathbf{0} \quad (3.35)$$

for rear-wheel drive (RWD) layout and

$$\begin{pmatrix} \sin(\theta + \phi) & -\cos(\theta + \phi) & 0 & 0 \\ \sin \theta & -\cos \theta & l_w & 0 \end{pmatrix} \begin{pmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \mathbf{0} \quad (3.36)$$

for front-wheel drive (FWD). Then we can obtain the corresponding basis of the null space of matrix $\mathbf{A}^T(\mathbf{q})$ denoted as $\mathcal{N}(\mathbf{A}^T(\mathbf{q}))$ and use it to represent $\dot{\mathbf{q}}$ as

$$\begin{pmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \frac{\tan \phi}{l_w} \\ 0 \end{pmatrix} \xi_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \xi_2 \quad (3.37)$$

in rear-wheel drive mode and

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\theta} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} \cos(\theta + \phi) \\ \sin(\theta + \phi) \\ \frac{\sin \phi}{l_w} \\ 0 \end{pmatrix} \chi_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \chi_2 \quad (3.38)$$

for front-wheel drive layout respectively, where ξ_i and χ_i ($i = 1, 2$) are the inputs. From Fig. 3.6a, we also notice the geometric relationships

$$\begin{cases} \kappa_Q = \frac{\tan \phi}{l_w}, \\ \kappa_P = \frac{\sin \phi}{l_w}, \end{cases} \quad (3.39)$$

where $\kappa_Q = r_Q^{-1}$, $\kappa_P = r_P^{-1}$ are the instantaneous curvature values of the rear and front wheels respectively. Furthermore, considering that the mobile robot is controlled by its velocity and angular acceleration, (3.37) and (3.38) can be rewritten using (3.39) as:

$$\begin{pmatrix} \dot{x}_Q \\ \dot{y}_Q \\ \dot{\theta} \\ \dot{\kappa}_Q \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \kappa_Q \\ 0 \end{pmatrix} v_Q + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \sigma_Q \quad (3.40)$$

and

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{\psi} \\ \dot{\kappa}_P \end{pmatrix} = \begin{pmatrix} \cos \psi \\ \sin \psi \\ \kappa_P \\ 0 \end{pmatrix} v_P + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \sigma_P, \quad (3.41)$$

where $\psi = \theta + \phi$ is the orientation of the front wheel, v_P, v_Q are the velocities of the front and rear wheels, and $\sigma_P = \dot{\kappa}_P, \sigma_Q = \dot{\kappa}_Q$ are the angular accelerations. Thus, both the front-wheel drive and rear-wheel drive have similar kinematic models with

several distinctions. For example, model in (3.41) is close to the kinematic model of a unicycle with an additional trailer. Thus it is able to have a maximum steering angle ϕ_{\max} greater than $\frac{\pi}{2}$ though rarely encountered while the model in (3.40) has to avoid the singularity. In most cases, both models are operated within limited steering angle $|\phi| \leq \phi_{\max} \in [0, \frac{\pi}{2}]$ and angular acceleration $|\dot{\sigma}| \leq \sigma_{\max}$, therefore, in these two driving modes the values of curvature and acceleration are derived as

$$\begin{cases} \kappa_Q = \frac{\tan \phi}{l_w}, \\ \sigma_Q = \frac{\dot{\phi}}{l_w \cos^2 \phi}, \end{cases} \quad \text{and} \quad \begin{cases} \kappa_P = \frac{\sin \phi}{l_w}, \\ \sigma_P = \frac{\dot{\phi} \cos \phi}{l_w}. \end{cases} \quad (3.42)$$

When the driving speed is slow or the length of the path is the major concern, we can assume the velocity to be constant and let v_P and v_Q be the unit velocity for simplicity. For convenience, both models of FWD and RWD are referred to as continuous-curvature (CC) vehicles.

3.2 Local Path Configurations

Considering a static environment with corridors and obstacles, to guarantee safe and feasible driving along the resulting path, we use a hierarchical planning approach to obtain desired paths. First, for the global path planning stage, we compute a roadmap based on medial axis of the environment to achieve maximum clearance. Second, use splines as local planner to find a smooth drivable path on the roadmap. Finally, obstacle avoidance is considered to adjust the resulting path.

After applying Pontryagin's maximum principle (PMP) to (3.40) and (3.41) subject to the corresponding constraints (3.42), necessary conditions for an optimal solution can be obtained [49]. However, the desired paths are expected to contain an infinite number of concatenated clothoid arcs, which are too complex to implement in practical applications. Instead, a local near-optimal path with continuous curvature is formulated using straight lines, circular arcs, and clothoid arcs as primitives with the constant driving velocity assumption as in [22]. Since the spline based local planner can efficiently connect two configurations without

resorting to a boundary value problem solver [73], the clothoid primitives can potentially be used as an effective alternative to the path smoothing and planning problems.

3.2.1 General Case

To begin with, let us consider the most straightforward scenario that three successive waypoints P_{n-1} , P_n and P_{n+1} form a corner with the angle from $\overrightarrow{P_{n-1}P_n}$ to $\overrightarrow{P_nP_{n+1}}$ named deflection angle δ , as illustrated in Fig. 3.7. Different from [22] which aims to solve for a spline path exactly passing the start and goal configurations, we are focusing on the path smoothing approach dealing with sharp corner steering, thus the start and goal positions are assumed changeable in most cases. Depending on the value of δ , the local path may have different variations. One of the most common and efficient formulations is using a pair of symmetric clothoid arcs to steering around the corner with G^2 continuity and then concatenating all the clothoid pairs with line segments, thus the obtained composite curve is still a CC path. We denote this type of configuration by CASE 1, as shown in Fig. 3.7a. The local spline path is composed of a pair of symmetric clothoids $\widehat{C_bC_m}$ and $\widehat{C_mC_e}$ of the same sharpness σ , possibly along with one or two pieces of straight line segments $\overline{P_{n-1}C_b}$ and $\overline{C_eP_{n+1}}$ to connect to the waypoints.

The computation of such a natural configuration with two symmetric clothoid arcs is also quite straightforward. We define the length of a segment on the bisector \mathcal{L}_0 connecting the junction C_m and corner as the offset distance between the local path $\widehat{C_bC_e}$ and the n -th waypoint P_n , which is denoted by d_n and used for evaluating the quality of the path. A smaller value of d_n indicates that the obtained path is closer to the reference polyline formed by the specified waypoints, which indicates better trackability and safety if the reference waypoints fall on the centerline of the road. Let θ_m and y_m be the winding angle and vertical distance of clothoid $\widehat{C_bC_m}$ at point C_m respectively. Considering $\delta = 2\theta_m$, it follows easily that

$$\begin{cases} \kappa_m = \sigma s, \\ \delta = \sigma s^2, \\ d_n \cos\left(\frac{\delta}{2}\right) = \sqrt{\frac{\pi}{\sigma}} S_f\left(\sqrt{\frac{\sigma}{\pi}} s\right), \end{cases} \quad (3.43)$$

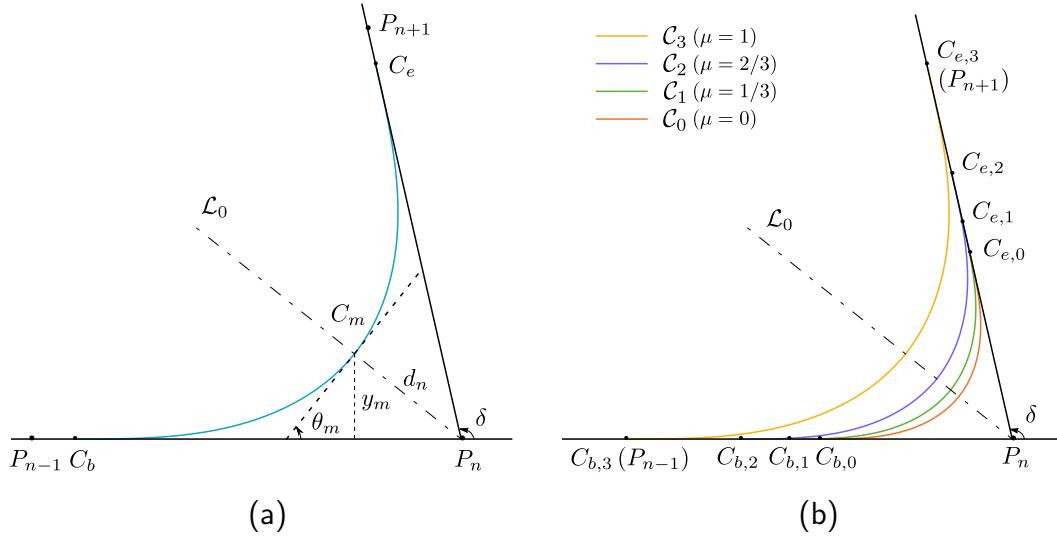


Figure 3.7: Local path (CASE 1): (a) Symmetric configuration. (b) Feasible local spline paths of the same deflection angle.

where κ_m is the curvature at position C_m . Consequently, d_n in CASE 1 can be regarded as a function of δ and σ :

$$d_n^I(\sigma, \delta) = \sqrt{\frac{\pi}{\sigma}} S_f \left(\sqrt{\frac{\delta}{\pi}} \right) \sec \left(\frac{\delta}{2} \right). \quad (3.44)$$

It can be easily checked that $\frac{\partial d_n}{\partial \sigma} < 0$, $\frac{\partial d_n}{\partial \delta} > 0$, therefore generating a path close enough to the corner that requires either a great sharpness σ or a small deflection angle δ . In fact, in the extreme case of $\delta = 0$ or $\sigma = +\infty$ without considering other restrictions, the local path degenerates to a point which coincides with P_n . Once the deflection angle δ is specified, the valid smoothing path is not unique, because all the value pairs (σ, s) satisfying (3.43) are supposed to be feasible solutions. As shown in Fig. 3.7b, given a deflection angle δ , the curves \mathcal{C}_i ($i = 0, 1, 2, 3$) are all valid paths with G^2 continuity. From (3.43) we know $\sigma = \kappa_m^2 / \delta$, hence a greater sharpness should produce a CC curve with higher curvature like \mathcal{C}_0 . Although steering with a high curvature curve has the advantage of keeping the path close to the corner, however, the generated path is longer than the others, which can be analyzed quantitatively. According to Fig. 3.7a, l_n is defined as the total length of the local path which includes the linear parts $\overline{P_{n-1}C_b}$ and $\overline{C_e P_{n+1}}$ along with the

clothoid pair $\widehat{C_b C_e}$:

$$l_n^I(\sigma, \delta) = l_{n,P} - \frac{2}{\sqrt{\sigma}} \left(\sqrt{\pi} C_f \left(\sqrt{\frac{\delta}{\pi}} \right) + \sqrt{\pi} \tan \left(\frac{\delta}{2} \right) S_f \left(\sqrt{\frac{\delta}{\pi}} \right) - \sqrt{\delta} \right), \quad (3.45)$$

where $l_{n,P} = \|P_{n-1}P_n\| + \|P_nP_{n+1}\|$. It can be verified that $\frac{\partial l_n}{\partial \delta} < 0$, $\frac{\partial l_n}{\partial \sigma} > 0$ for all $\delta \in (0, \pi)$ and $\sigma > 0$, while the partial derivatives are equal to 0 only when $\delta = 0$.

In practice, the deflection angle δ cannot be easily changed once the waypoints have been specified. As a result, both d_n and l_n are dependent on the sharpness of the clothoid pair in most cases, which in turn determines the shape of the local path. Now the value of δ is considered as a constant, and we have to figure out the bounds of σ . Based on (3.43), we notice that the curvature at C_m and the sharpness should satisfy the constraint of $\kappa_m \leq \kappa_{\max}$ and $\sigma \leq \sigma_{\max}$ respectively, where κ_{\max} and σ_{\max} are the maximum curvature and sharpness restricted by the kinematics of the vehicle model according to (3.42). Consequently, the upper bound of sharpness in an obstacle-free environment in CASE 1 is determined by

$$\sigma_{\max, \text{FREE}}^I = \min \left\{ \frac{\kappa_{\max}^2}{\delta}, \sigma_{\max} \right\}. \quad (3.46)$$

Meanwhile, the lower bound σ_{\min}^I is directly related to the maximum length of the local path, which is denoted by $l_{n,\min}^I$ and mainly depends on the distribution of the waypoints and surrounding objects. For convenience, here we first consider the restriction of the waypoints in an obstacle-free environment. Since both the start and goal positions, i.e., C_b and C_e , should fall on the edges formed by the waypoints, we have $\|C_b P_n\| \leq \min \{\|P_{n-1}P_n\|, \|P_nP_{n+1}\|\}$ and

$$\sigma_{\min, \text{FREE}}^I = \frac{\pi \left(C_f \left(\sqrt{\frac{\delta}{\pi}} \right) + \tan \left(\frac{\delta}{2} \right) S_f \left(\sqrt{\frac{\delta}{\pi}} \right) \right)^2}{(\min \{\|P_{n-1}P_n\|, \|P_nP_{n+1}\|\})^2}. \quad (3.47)$$

It is also worth to mention that given a pair of (σ, κ_m) , the winding angle constraint $\theta_m = \frac{\kappa_m^2}{2\sigma} = \frac{\delta}{2} \leq \frac{\pi}{2}$ should be satisfied as well. For example, the lower bound of the sharpness is above $\frac{\kappa_m^2}{\pi}$ if the curvature κ_m is specified.

In addition, if we take the existence of surrounding objects into consideration like the road boundary or obstacles in the environment, the offset distance is further

restricted to $d_{n,\text{LOWER}}^{\text{I}} \leq d_n^{\text{I}} \leq d_{n,\text{UPPER}}^{\text{I}}$. Accordingly, from (3.44) the bounds of sharpness only subject to the environment can be derived respectively:

$$\begin{aligned}\sigma_{\min,\text{ENV}}^{\text{I}} &= \frac{\pi S_f \left(\sqrt{\frac{\delta}{\pi}} \right)^2}{\left(d_{n,\text{UPPER}}^{\text{I}} \right)^2 \cos^2 \left(\frac{\delta}{2} \right)}, \\ \sigma_{\max,\text{ENV}}^{\text{I}} &= \frac{\pi S_f \left(\sqrt{\frac{\delta}{\pi}} \right)^2}{\left(d_{n,\text{LOWER}}^{\text{I}} \right)^2 \cos^2 \left(\frac{\delta}{2} \right)}.\end{aligned}\quad (3.48)$$

Combining (3.46), (3.47) and (3.48), when a deflection angle is specified, we can obtain the bounds of the sharpness in CASE 1 as:

$$\begin{aligned}\sigma_{\min}^{\text{I}} &= \max \left\{ \sigma_{\min,\text{FREE}}^{\text{I}}, \sigma_{\min,\text{ENV}}^{\text{I}} \right\}, \\ \sigma_{\max}^{\text{I}} &= \min \left\{ \sigma_{\max,\text{FREE}}^{\text{I}}, \sigma_{\max,\text{ENV}}^{\text{I}} \right\}.\end{aligned}\quad (3.49)$$

As a consequence, the minimum values of d_n^{I} and l_n^{I} can be computed by substituting corresponding extrema of sharpness in (3.49) into (3.44) and (3.45), both of which are monotonic functions with respect to σ . In a typical path smoothing problem, it is not possible to minimize the offset distance and path length at the same time. Considering the fact that both d_n^{I} and l_n^{I} are functions of σ if δ is treated as a constant, we are able to adjust the shape and position of the local path by introducing a shape factor $\mu \in [0, 1]$ so that all the feasible sharpness values can be expressed as $\sigma = \mu \sigma_{\min}^{\text{I}} + (1 - \mu) \sigma_{\max}^{\text{I}}$. An example of four curves with different shape factors is shown in Fig. 3.7b.

3.2.2 Other Cases

Although the configuration in CASE 1 can ensure a G^2 smooth path, it is not an optimal solution, which requires that the clothoid arcs involved should have the extreme value of sharpness according to [21]. In other words, the vehicle has to be maneuvered at its maximum angular acceleration σ_{\max} . However, from (3.46) we known that $\sigma = \sigma_{\max}$ is an invalid input in an obstacle-free environment for CASE 1 with $\delta > \kappa_{\max}^2 / \sigma_{\max}$. As a consequence, we introduce a different type of path configuration which involves a piece of circular arc for joining the clothoids, denoted by CASE 2. By combining this kind of spline path with CASE 1, the mobile robot

is able to drive at maximum angular acceleration when steering around a corner of arbitrary deflection angle. The basic configuration is illustrated in Fig. 3.8a, which includes a piece of intermediate circular arc $\widehat{A_b A_e}$ of radius r in addition to the clothoids and straight lines in CASE 1. Still, it is expected that the obtained local path is G^2 continuous, whose curvature increases from 0 to κ_{\max} linearly on $\widehat{C_b A_b}$, remains constant on $\widehat{A_b A_e}$, and decreases from κ_{\max} to 0 symmetrically on $\widehat{A_e C_e}$. Similarly, we define the length from C_m , the intersection point of bisector \mathcal{L}_0 and $\widehat{A_b A_e}$, to P_n as the offset distance d_n . In the case of $\sigma = \sigma_{\max}$, the curvature at A_b

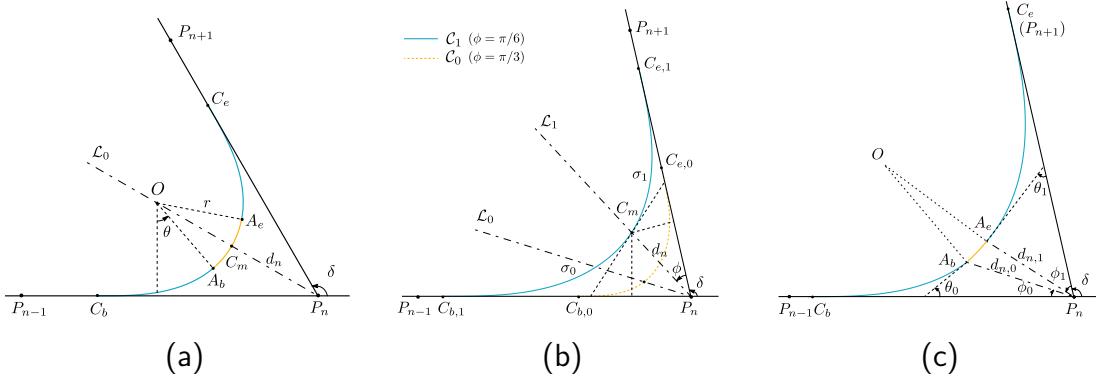


Figure 3.8: Local path: (a) CASE 2. (b) CASE 3. (c) CASE 4.

happens to be the reciprocal of the minimum turning radius $r = r_{\min}$ of the vehicle, while the winding angle and the deflection angle should satisfy the constraint of $\delta > 2\theta$ and

$$\left\{ \begin{array}{l} \kappa_m = \frac{1}{r} = \sigma s, \\ \theta = \frac{1}{2}\sigma s^2, \\ d_n \cos\left(\frac{\delta}{2}\right) = \sqrt{\frac{\pi}{\sigma}} S_f \left(s \sqrt{\frac{\sigma}{\pi}}\right) + r \left(\cos \theta - \cos\left(\frac{\delta}{2}\right)\right), \end{array} \right. \quad (3.50)$$

where κ_m is the curvature at position C_m . Similar to CASE 1, d_n in CASE 2 can be regarded as a function of δ , σ , and r :

$$d_n^{II}(\sigma, \delta, r) = \sqrt{\frac{\pi}{\sigma}} S_f \left(\frac{1}{r\sqrt{\pi\sigma}}\right) \sec\left(\frac{\delta}{2}\right) + r \left(\cos\left(\frac{1}{2r^2\sigma}\right) \sec\left(\frac{\delta}{2}\right) - 1\right). \quad (3.51)$$

It is straightforward to verify that $d_n^{II}(\sigma, \delta, r)$ is a monotonic function with respect to both $\sigma > 0$ and $0 < \delta < \pi$ since $\frac{\partial d_n}{\partial \sigma} < 0$ and $\frac{\partial d_n}{\partial \delta} > 0$. For the radius parameter r , $\frac{\partial d_n}{\partial r} > 0$ holds when $r > \frac{1}{\sqrt{\sigma \delta}}$ based on the condition $\delta > 2\theta$ and (3.50). The result is consistent with the conclusion that d_n can be significantly reduced if we always set $\sigma = \sigma_{\max}$ and $r = \kappa_{\max}^{-1}$ in CASE 2.

Occasionally, due to the environmental constraints, an asymmetric variation of CASE 1 may be used, which is referred to as CASE 3 and illustrated in Fig. 3.8b, which includes two different groups of local paths. Take \mathcal{C}_1 as an example, the path is composed of a pair of asymmetric clothoid arcs ($\widehat{C_{b,1}C_m}$ and $\widehat{C_mC_{e,1}}$) and straight line segments ($\overrightarrow{P_{n-1}C_{b,1}}$ and $\overrightarrow{C_{e,1}P_{n+1}}$). We represent the angle from $\overrightarrow{P_nP_{n+1}}$ to $\overrightarrow{P_nC_m}$ as ϕ , and define the offset distance $d_n = \|P_nC_m\|$. If the values of sharpness and arc length of each clothoid arc are denoted as σ_i and s_i ($i = 0, 1$) respectively, they can be obtained by solving the nonlinear system:

$$\begin{cases} \sigma_0 s_0 = \sigma_1 s_1, \\ \delta = \frac{1}{2}\sigma_0 s_0^2 + \frac{1}{2}\sigma_1 s_1^2, \\ d_n \sin(\delta + \phi) = \sqrt{\frac{\pi}{\sigma_0}} S_f \left(s_0 \sqrt{\frac{\sigma_0}{\pi}} \right), \\ d_n \sin \phi = \sqrt{\frac{\pi}{\sigma_1}} S_f \left(s_1 \sqrt{\frac{\sigma_1}{\pi}} \right). \end{cases} \quad (3.52)$$

The curvature at point C_m is $\kappa_m = \sigma_0 s_0 = \sigma_1 s_1 \leq \kappa_{\max}$ and both σ_0 and σ_1 should be smaller than the maximum angular acceleration allowed σ_{\max} . From (3.52) we know the constraints of σ_0 and σ_1 as

$$\begin{cases} \sigma_{\max} \geq \max \{ \sigma_0, \sigma_1 \}, \\ \frac{2\delta}{\kappa_m^2} = \frac{1}{\sigma_1} + \frac{1}{\sigma_0}. \end{cases} \quad (3.53)$$

Without loss of generality, we assume $\theta_1 < \frac{\delta}{2} < \theta_0$, it is obvious from (3.52) and (3.53) that $\sigma_0 < \frac{\kappa_m^2}{\delta} < \sigma_1$ and vice versa, where θ_0 and θ_1 are the winding angles of corresponding clothoid arc. In fact, CASE 1 can be treated as a specific instance of CASE 3 when $\theta_0 = \theta_1 = \frac{\delta}{2}$. Again, based on (3.52) the offset distance in CASE 3 can be expressed by

$$d_n^{III}(\sigma_1, \kappa_m, \phi) = \sqrt{\frac{\pi}{\sigma_1}} S_f \left(\frac{\kappa_m}{\sqrt{\pi \sigma_1}} \right) \csc \phi. \quad (3.54)$$

As discussed before, the bounds of the sharpness in CASE 1 can be determined by (3.46) and (3.47) in a free space. Subsequently, no valid solution exists in CASE 1 if $\sigma_{\max,\text{FREE}}^{\text{I}} < \sigma_{\min,\text{FREE}}^{\text{I}}$ holds. This kind of situation happens because one edge is too short for path smoothing, i.e.,

$$\min \{ \|P_{n-1}P_n\|, \|P_nP_{n+1}\| \} < \frac{\sqrt{\pi} \left(C_f \left(\sqrt{\frac{\delta}{\pi}} \right) + \tan \left(\frac{\delta}{2} \right) S_f \left(\sqrt{\frac{\delta}{\pi}} \right) \right)}{\sqrt{\sigma_{\max,\text{FREE}}^{\text{I}}}}. \quad (3.55)$$

Take the path \mathcal{C}_1 in Fig. 3.8b as an example. If $\overline{P_nP_{n+1}}$ satisfies (3.55) while $\overline{P_{n-1}P_n}$ is long enough, we can adopt the configuration in CASE 3. Assuming $\min \{ \|P_{n-1}P_n\|, \|P_nP_{n+1}\| \} = \|P_nP_{n+1}\|$, from (3.55) and (3.52) the length of $\overline{P_nP_{n+1}}$ is bounded by $\|P_nP_{n+1}\| \geq \|P_nC_{e,1}\|$, where $\|P_nC_{e,1}\|$ can be computed by substituting $\sigma = \sigma_1$ and $\kappa = \kappa_m$ into

$$p_n^{\text{III}}(\sigma, \kappa, \phi) = \sqrt{\frac{\pi}{\sigma}} \left(C_f \left(\frac{\kappa}{\sqrt{\pi\sigma}} \right) + S_f \left(\frac{\kappa}{\sqrt{\pi\sigma}} \right) \cot \phi \right). \quad (3.56)$$

Accordingly, in order to ensure the existence of a valid solution, the other edge is constrained by $\|P_{n-1}P_n\| \geq \|C_{b,1}P_n\| = p_n^{\text{III}}(-\frac{\kappa_m^2 \sigma_1}{\kappa_m^2 - 2\delta\sigma_1}, \kappa_m, \pi - \delta - \phi)$.

In the same way, there should be CASE 4 as shown in Fig. 3.8c, which, mathematically, is an asymmetric extension of CASE 2. However, CASE 4 is rarely used since it can be replaced by CASE 3 in most cases. Here, we just give a brief description of CASE 4 for the sake of completeness. Offset distances $d_{n,i}$ and angles ϕ_i ($i = 0, 1$) are needed to solve for σ_i and s_i in the following system:

$$\left\{ \begin{array}{l} \kappa_m = \frac{1}{r} = \sigma_0 s_0 = \sigma_1 s_1, \\ \theta_0 = \frac{1}{2} \sigma_0 s_0^2, \\ \theta_1 = \frac{1}{2} \sigma_1 s_1^2, \\ d_{n,0} \sin \phi_0 = \sqrt{\frac{\pi}{\sigma_0}} S_f \left(s_0 \sqrt{\frac{\sigma_0}{\pi}} \right), \\ d_{n,1} \sin \phi_1 = \sqrt{\frac{\pi}{\sigma_1}} S_f \left(s_1 \sqrt{\frac{\sigma_1}{\pi}} \right), \\ \left(2r \sin \left(\frac{\delta - \theta_0 - \theta_1}{2} \right) \right)^2 = d_{n,0}^2 + d_{n,1}^2 + 2d_{n,0}d_{n,1} \cos (\delta + \phi_0 + \phi_1). \end{array} \right. \quad (3.57)$$

3.2.3 Path Smoothing

3.2.3.1 Path Smoothing in Free Space

To demonstrate the advantages of clothoid based smoothing, we compare our method with the B-spline based algorithm presented in [95] with a typical example extracted from the same paper, which is originally from [96]. Without adding or removing waypoints, we use the symmetric configuration of CASE 1 and follow the same rule to pick the start positions of the steering paths for consistency. It is illustrated that the obtained paths is G^2 continuous, which, if necessary, can be reparameterized accordingly to satisfy C^2 continuity as the B-spline path. A comparison of two approaches is shown in Fig. 3.9 and the path curvature profiles are obtained with regard to the normalized curve length s . Considering the fact that any curve in \mathbb{R}^2 has no rational arc length parameterization except straight lines [97], we use the iterative sampling algorithm presented in [98] to approximate the arc length reparameterization with a tolerance of 10^{-8} . After that, we compute corresponding curvature profiles numerically for comparison with respect to normalized arc length values of both spline paths. Thanks to the inherent properties of clothoid, it can be seen from Fig. 3.9d that the smoothed path has upper and lower bounds of signed curvatures $\kappa_{\max}^* = 1.15$ and $\kappa_{\min}^* = -0.63$, which are much smaller than $\kappa_{\max}^* = 3.07$ and $\kappa_{\min}^* = -2.00$ with B-spline smoothing approach shown in Fig. 3.9c. Also it can be observed that the B-spline path is significantly sharper around each corner, thus the it places a more strict demand on the kinematics and maneuverability of the mobile robot with respect to the maximum steering angle and acceleration, which makes the generated path more difficult to follow. Additionally, the total length of the path based on clothoid smoothing is $s_{\text{TOTAL}} = 134.39$, which is shorter than that using B-spline smoothing method ($s_{\text{TOTAL}} = 135.62$).

Since we have totally four different configurations of local paths for various purposes. Among them, the configuration of CASE 1 featured with symmetric clothoids pair is the general solution for most scenarios. As a supplement, by introducing a piece of circular arc, CASE 2 is used when a maximum angular

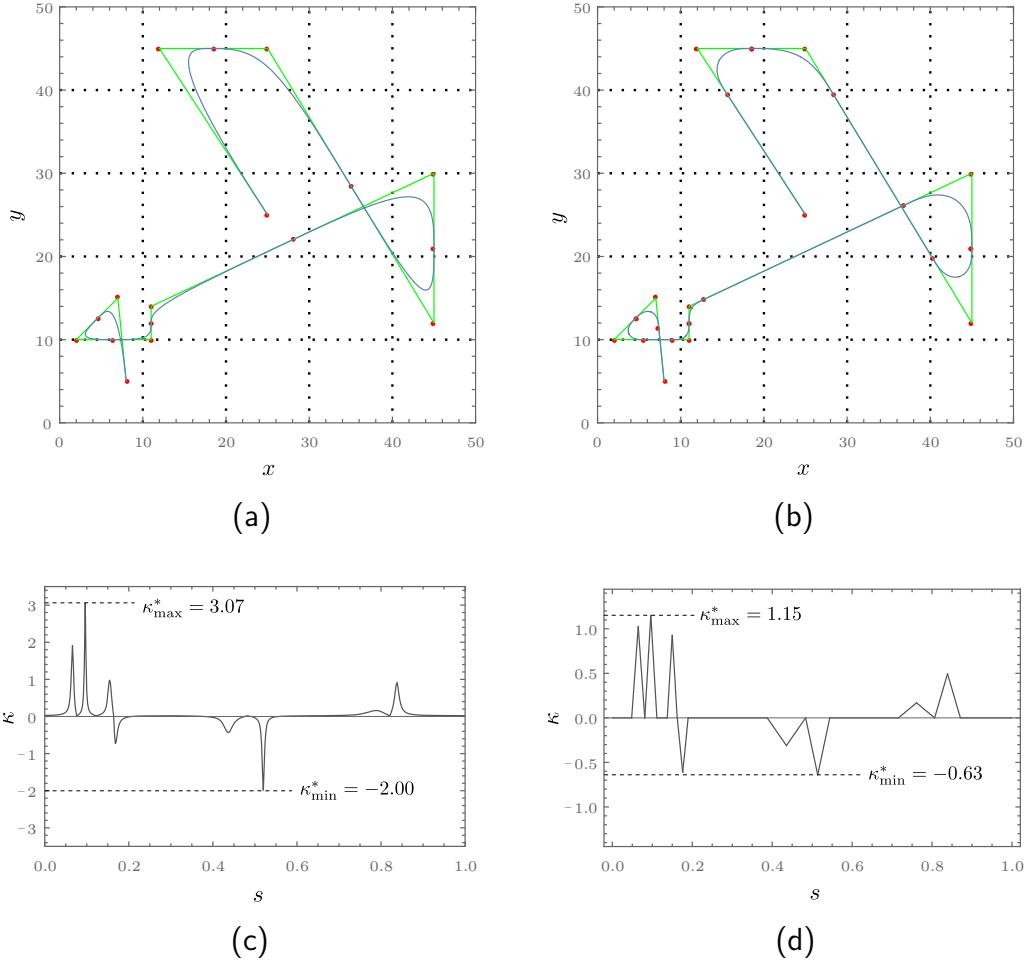


Figure 3.9: Path smoothing example: (a) B-spline smoothing. (b) Proposed path smoothing. (c) Curvature profile of B-spline path. (d) Curvature profile of proposed path.

acceleration σ_{\max} is always required. In addition, CASE 3 and 4 with asymmetric configurations have the advantage of being able to smooth corners composed of short edges, which typically connect waypoints with the start and goal poses. Fig. 3.10 is a contrived example of making use of different types of configurations. Given waypoints P_i ($i = 1, 2, \dots, 6$) on the plane and a G^2 composite path is formulated. In this example, the prerequisites of the mobile robot are that the path curvature and angular acceleration should not exceed $\kappa_{\max} = 0.5$ and $\sigma_{\max} = 0.1$ respectively. The curvature of the desired path should increase when approaching each corner and decrease when departing from it. As a result, a main configuration

involving two pieces of curves is considered, which includes an ascending path and descending path with respect to their curvature profiles.

Here, in order to satisfy the requirements, different strategies are used to steer around each corner. We first try to apply the general configuration in CASE 1 at each corner. However, according to (3.46), $\sigma_{\max} = 0.1$ is an invalid input at corner P_2 . If we stick to CASE 1 in this situation, σ should be reduced and consequently the obtained path becomes inevitably longer. Hence, we adopt the configuration of CASE 2 instead, which allows the maximum sharpness $\sigma_{\max} = 0.1$ and curvature $\kappa_{\max} = 0.5$ at the same time by inserting a short piece of circular arc of radius $r = \kappa_{\max}^{-1} = 2$. For corner P_3 and P_4 , the symmetric local path in CASE 1 is used by setting the sharpness to be the extrema. Observing that the goal P_6 is close to the waypoint P_5 , the solution in CASE 1 and 2 does exist according to (3.55) and CASE 3 or 4 can be resorted to at this moment. For simplicity, we apply the asymmetric clothoid path from CASE 3 and the sharpness of the descending part is set to be σ_{\max} . To determine the path shape, (3.52) must be solved, and an additional parameter such as an appropriate arc length or the angle ϕ needs to be specified. Finally, by connecting all these local paths with straight lines, we are able to create a complete smooth path with G^2 continuity. Not only does the maximum curvature comply with the conditions as shown in Fig. 3.10b, but the sharpness values also fall within the specified range, which is a distinguishing characteristic of clothoid and demonstrated in Fig. 3.10c.

3.2.3.2 Collision Avoidance and Continuous Collision Checking

Collision avoidance is a major concern during the maneuvering of the vehicle and is an indispensable module of a practical path planner. In a corridor-like environment, all the collisions can be classified into two different scenarios. The first kind of collision occurs between the generated path and the surrounding environment like the walls and obstacles on the road. Although the waypoints and reference path are collision free in most cases, the corresponding smoothing path may probably intersect with other objects. These paths are dangerous to follow thus are invalid results, though they satisfy the kinematic constraints or even are optimal solutions in a free space.

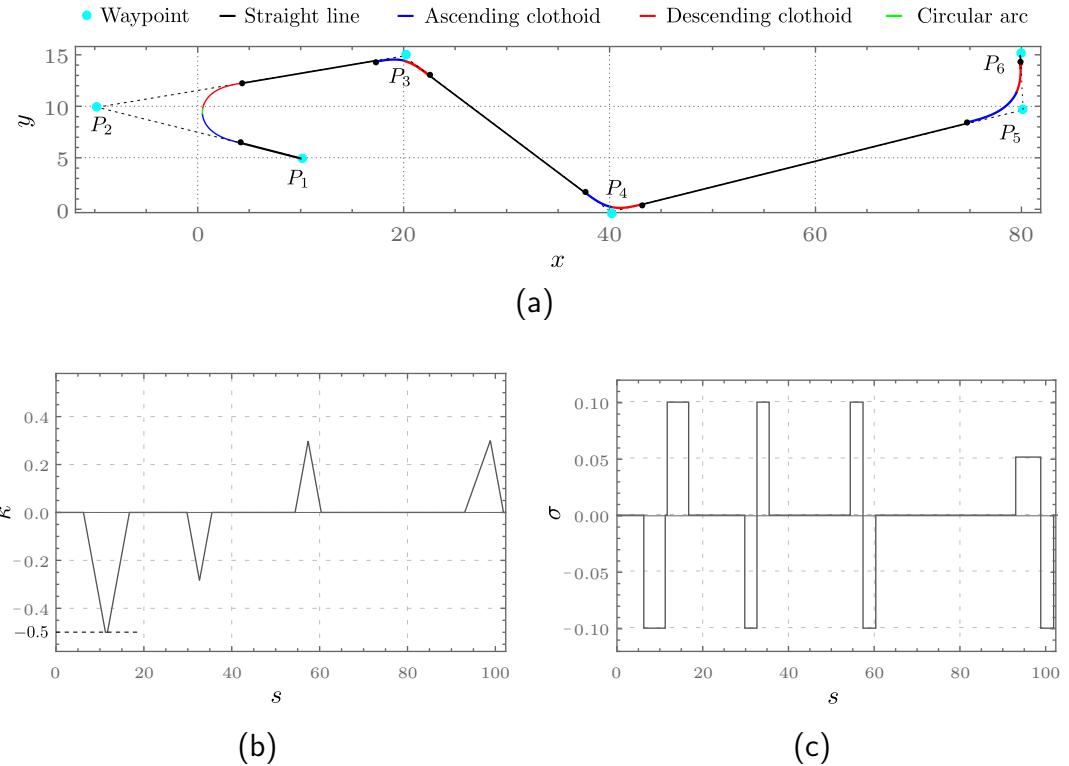


Figure 3.10: Path smoothing example: (a) Proposed spline path. (b) Curvature profile. (c) Sharpness profile.

The second kind of collision is caused by the robot body, which includes the collisions between the robot and the environment as well as the self-collisions between the robot components. Usually the vehicle is regarded as a point robot and only the intersections between the generated paths and the environment are considered as collisions [99]. In such a way, the planning procedure is simplified and can be highly efficient for agile robots of small body size due to the fact that collision checking is usually the most time-consuming aspect of motion planning especially in a complex environment. Under the circumstances where the robot size cannot be neglected, an alternative method is using the Minkowski set difference [100] if the obstacles are represented by polygons as stated in [101]. Nevertheless, this kind of approach may be less effective when the shape of the vehicle cannot be treated as a plate. For vehicles in non-plate shape, a general purpose collision detection algorithm may be necessary which mainly depends on the bounding volume

hierarchies [102].

Here, both types of collisions should be considered in the proposed planner. Since the generated path is represented with a spline explicitly, the collision checking can be performed using an effective computational geometry approach after extracting the geometric information of the environment. First we consider a basic scenario that a vehicle in a rectangular shape with width w_v is steering around a narrow passage, as illustrated in Fig. 3.11. Waypoints P_i ($i = 1, 2, 3$) are generated by the global planner with straight skeleton approach and polyline is formed as our reference path, which in fact coincide with the road centerline in this case. The vehicle is moving from P_{n-1} to P_{n+1} . \mathcal{W}_i and \mathcal{W}_o represent inner and outer road boundaries or walls and not allowed to pass through. Similar to the local planner generated in a free space, we just apply the configuration in CASE 1 with different pairs (σ, κ_m) . As can be seen in Fig. 3.11a, 6 separate local paths

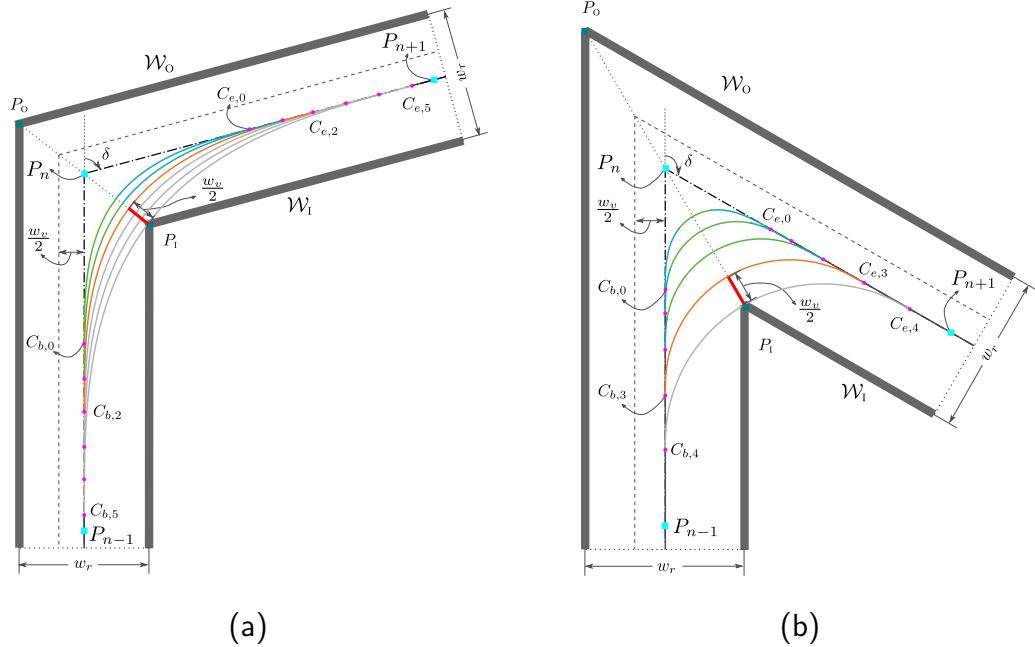


Figure 3.11: Collision avoidance around a corner: (a) CASE 1. (b) CASE 2.

$C_{b,i}C_{e,i}$ ($i = 0, 1, \dots, 5$) with different sharpness values are created by smoothing the reference path and all of these curves can be regarded as qualified paths for a point robot. However, after the vehicle size is considered, these paths should

be further examined based on the collision conditions. To begin with, we compute curve $C_{b,0}C_{e,0}$ with the extrema $\sigma = \sigma_{\max,\text{FREE}}^{\text{I}}$ in (3.46) and offset distance $d_n^{\text{I}}(\sigma_{\max,\text{FREE}}^{\text{I}}, \delta)$ based on (3.44). Assuming that $C_{b,5}, C_{e,5}$ is the curve that passing through the critical point P_i located on the inner boundary \mathcal{W}_i and $C_{b,2}C_{e,2}$ is the path with an exact distance of $\frac{w_v}{2}$ from P_i , all the path between them will collide with the inner boundary \mathcal{W}_i . Due to the symmetric property, P_i should coincide with the joint of two clothoids. In addition, a safe path should stay clear from the outer boundary \mathcal{W}_o with a distance greater than $\frac{w_v}{2}$. Therefore, to maintain a safe clearance from the boundaries, the generated paths should fall into the region between $C_{b,0}C_{e,0}$ (included) and $C_{b,2}C_{e,2}$. According to (3.44) we know that a safe offset distance should satisfy $d_{\text{SAFE}} \geq d_n^{\text{I}}(\sigma_{\max,\text{FREE}}^{\text{I}}, \delta)$ while its upper bound can be obtained according to the geometric relationships shown in Fig. 3.11a:

$$d_{\text{SAFE}} < \frac{1}{2} \left(\sec \left(\frac{\delta}{2} \right) w_r - w_v \right). \quad (3.58)$$

Accordingly, the allowed sharpness range is determined by substituting (3.58) into (3.44):

$$\frac{4\pi S_f^2 \left(\sqrt{\frac{\delta}{\pi}} \right)}{(w_r - \cos \left(\frac{\delta}{2} \right) w_v)^2} < \sigma_{\text{SAFE}} \leq \sigma_{\max,\text{FREE}}^{\text{I}}. \quad (3.59)$$

Hence, the maximum curvature of the curve is derived as $\kappa_{m,\text{SAFE}} = \sqrt{\delta \sigma_{\text{SAFE}}}$ based on (3.43). Now considering a vehicle of unit width $w_v = 1$, we are able to find out the upper bound of d_{SAFE} as well as the lower bound of σ_{SAFE} with respect to the road width $w_r \geq w_v$ and deflection angle $\delta \in [0, \pi]$, as shown in Fig. 3.12. For the sake of simplicity, the limitation of κ_m is not taken into consideration here. It can be easily seen that the deflection angle has more influence on d_{SAFE} and the latter becomes extremely large (these unreasonable values are clipped in the figure) when δ approaches π . Meanwhile, σ_{SAFE} is greatly affected by the road width w_r . As a result, to ensure the existence of an acceptable solution with appropriate offset distance and angular acceleration, a smaller δ and a greater w_r are preferred.

As demonstrated in Fig. 3.11a, for a specified δ , a greater sharpness implies a shorter steering curve considering that the arc length of one piece of the clothoid pair is $s_{\text{clo}} = \sqrt{\frac{\delta}{\sigma_{\text{SAFE}}}}$. On the other hand, by setting $\sigma = \sigma_{\max}$, we are able to

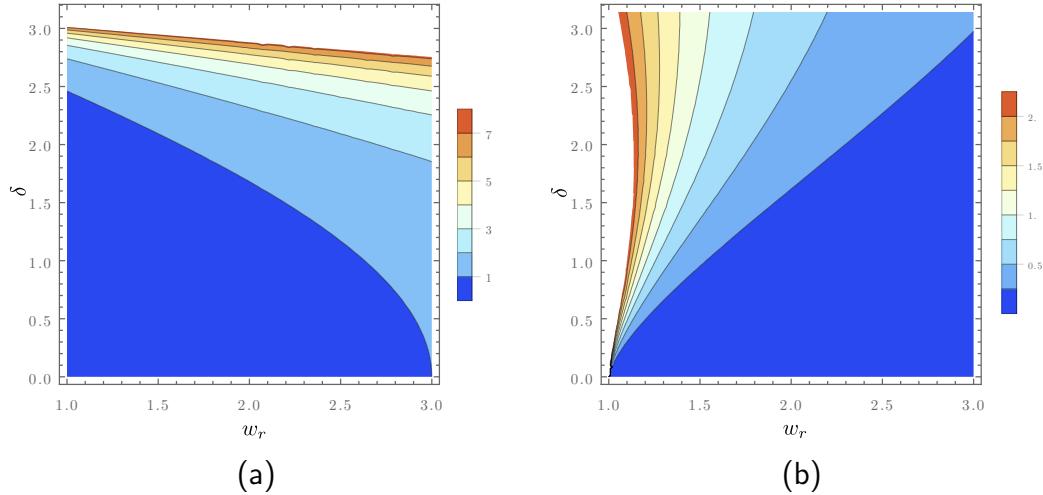


Figure 3.12: $w_v = 1$: (a) upper bound of d_{SAFE} . (b) Lower bound of σ_{SAFE} .

guarantee the minimum offset distance according to (3.44), which characterizes the safest path with the maximum clearance from the inner boundary (see $C_{b,0}C_{e,0}$ in Fig. 3.11a). We already know that for any given deflection angle δ , a CC path can be generated based on CASE 1. However, according to (3.46), the maximum angular acceleration cannot reach σ_{\max} in the case of $\sigma_{\max} > \frac{\kappa_{\max}^2}{\delta}$. In such a situation, CASE 2 is adopted as shown in Fig. 3.11b, where 5 curves $C_{b,i}C_{e,i}$ ($i = 0, 1, \dots, 5$) are illustrated. Since the sharpness is fixed to σ_{\max} given a specific vehicle model, the only factor r is used to modify the shape of the path based on (3.50). Here, $C_{b,0}C_{e,0}$ denotes the extreme path with $r_{\min} = \kappa_{\max}^{-1}$ while $C_{b,3}C_{e,3}$ is the path characterized by the critical point P_i in the middle of the circular arc. Therefore, all the curves in the region bounded by $C_{b,0}C_{e,0}$ and $C_{b,3}C_{e,3}$ are feasible paths free of collisions. Accordingly, the upper bound of the offset distance σ_{SAFE} is determined by (3.58) and presented in Fig. 3.12a where the condition $w_v = 1$ is assumed. Meanwhile, its lower bound is restricted to $d_n^{\text{II}}(\sigma_{\max}, \delta, \kappa_{\max}^{-1})$. Similarly, by substituting them into (3.51) respectively and solving the equation $d_{\text{SAFE}} = d_n^{\text{II}}(\sigma_{\max}, \delta, r)$ numerically, the safe range of the radius r_{SAFE} is obtained due to its monotonicity with respect to r . In such a way, the radius of the circular arc is up-bounded by r_{\max} which is expressed as an implicit function of w_r , δ and σ_{\max} , while its lower bound is simultaneously defined by a surface $r_{\min} = \frac{1}{\sqrt{\delta\sigma_{\max}}}$.

Fig. 3.13a depicts the shape of the upper bound r_{\max} in the case of 3 different maximum sharpness values while Fig. 3.13b shows r_{SAFE} bounded by two surfaces r_{\max} and r_{\min} when $\sigma_{\max} = 1.0$.

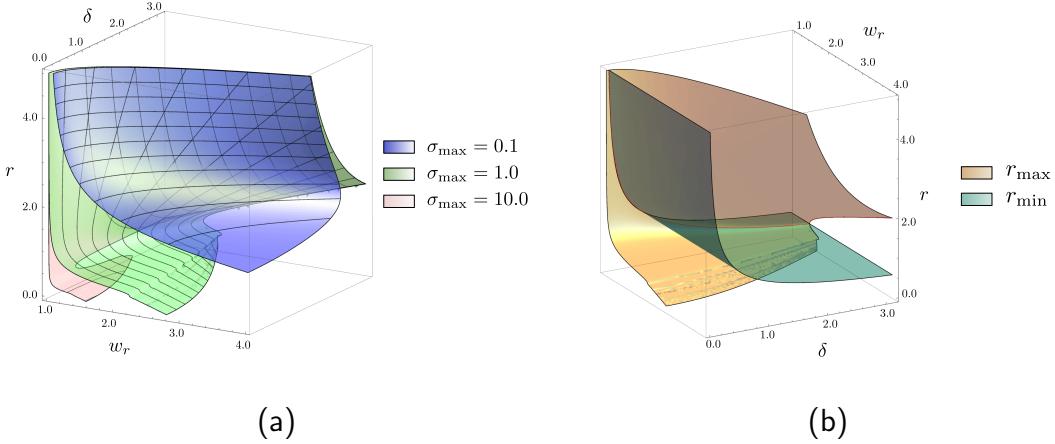


Figure 3.13: Computation of r_{SAFE} when $w_v = 1$: (a) r_{\max} as $\sigma_{\max} = 0.1, 1.0$, and 10.0 . (b) r_{\max} and r_{\min} as $\sigma_{\max} = 1.0$.

So far we have discussed the most common scenario of a corner with the same road width at both ends. When it comes to a more general case with different road widths (w_{r1} and w_{r2}) with a relationship of $\frac{w_{r1}}{w_{r2}} = \frac{\sin(\delta+\phi)}{\sin\phi}$ shown in Fig. 3.14, the space cannot be effectively utilized using the symmetric configurations in CASE 1 and CASE 2. The vehicle is moving from P_{n-1} to P_{n+1} and the deflection angle and other parameters are defined analogously to the symmetric instance discussed before. One of the major differences here is the angle ϕ formed by $\overrightarrow{P_n P_{n+1}}$ and $\overrightarrow{P_o P_i}$, which is not the bisector of $\angle P_{n-1} P_n P_{n+1}$ anymore. Naturally, the asymmetric configuration of CASE 3 can be very helpful considering this kind of path smoothing problem and 6 different CC paths $C_{b,i}C_{e,i}$ ($i = 0, 1, \dots, 5$) are illustrated with the same deflection angle in Fig. 3.14a. It should be noted that the critical point C_d , namely the projection of P_i onto the path $C_{b,2}C_{e,2}$, does not fall on the straight line P_oP_i . To figure out the parameters of the corresponding

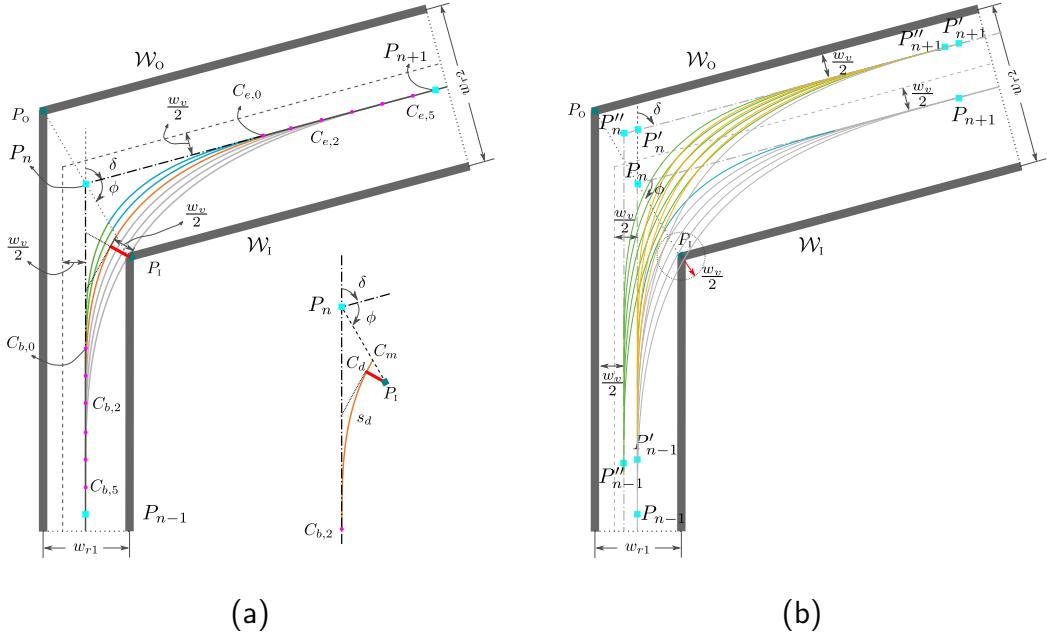


Figure 3.14: Collision avoidance around a corner: (a) CASE 3. (b) CASE 1 by modifying waypoints.

curve, we derive the following relationships

$$\left\{ \begin{array}{l} \sqrt{\frac{\pi}{\sigma_0}} \left(C_f \left(s_d \sqrt{\frac{\sigma_0}{\pi}} \right) - C_f \left(s_0 \sqrt{\frac{\sigma_0}{\pi}} \right) \right) = (w_p - d_n) \cos(\delta + \phi) \\ \quad + \frac{w_v}{2} \sin \left(\frac{\sigma_0 s_d^2}{2} \right), \\ \sqrt{\frac{\pi}{\sigma_0}} S_f \left(s_d \sqrt{\frac{\sigma_0}{\pi}} \right) = w_p \sin(\delta + \phi) - \frac{w_v}{2} \cos \left(\frac{\sigma_0 s_d^2}{2} \right), \end{array} \right. \quad (3.60)$$

where $w_p = \|P_n P_1\| = \frac{w_{r1}}{2 \sin(\delta + \phi)} = \frac{w_{r2}}{2 \sin \phi}$ and s_d is the arc length of the path $C_{b,2}C_d$. By combining (3.52) and (3.60), we are able to solve for the required parameters σ_0 , σ_1 , s_0 , s_1 , s_d , and d_n by providing a certain combination of δ , ϕ , w_p , and w_v .

However, finding an asymmetric solution for every corner depends on a time-consuming process of solving a nonlinear system. Also, we notice that even in symmetric scenarios, the space is not fully utilized. Take Fig. 3.11a as an example, the region near the inner boundary is heavily exploited by the local planer while the other side is almost left unused except for considering a minimum clearance of

$\frac{w_v}{2}$. After noticing that a lot of wasted space is caused by sticking to the waypoints falling on the road centerline, in some instances we can make changes to the relevant waypoints according to the practical requirements as shown in Fig. 3.14b. First we use the configuration of CASE 1 to smooth the corner with waypoints P_{n-1} , P_n and P_{n+1} without considering the boundaries. Totally 5 paths are generated correspondingly, among which only one candidate has no collisions against the inner boundary \mathcal{W}_l . Observing that the road has a much greater width at P_nP_{n+1} part, we translate the waypoint P_i to P'_i respectively ($i = n-1, n, n+1$) and then smooth the new polyline formed by P'_i using the previously generated curves by applying the same translation. $P'_nP'_{n+1}$ must maintain a minimum clearance from the outer boundary \mathcal{W}_o and it can be seen that with the new arrangement, 4 out of 5 paths are free of collisions. Furthermore, in order to elicit the free space of $P_{n-1}P_n$ part, we can apply an additional transformation to waypoints P_{n-1} and P_n so that the new polyline is defined by P''_{n-1} , P''_n and P''_{n+1} . Again, the original 5 paths continue to be used and all of them are now feasible solutions since no collisions are detected.

Compared with the asymmetric smoothing approach, the transformation method described here is more effective and easier to implement without solving for a non-linear system, though the configurations of CASE 3 and CASE 4 can still be integrated into this approach to further improve the space utilization. In addition, this method is not limited to the asymmetric scenario and can be applied to corners with same road widths at both in and out ends, too. In practice, a translation from P_i to P'_i is sufficient enough, because the modified arrangement only affect the arrangements at the current and next corners. Although the configuration of $P''_{n-1}P''_nP''_{n+1}$ seems more advantageous than $P'_{n-1}P'_nP'_{n+1}$ in the name of more accessible space and allowing greater clearances from the inner boundary, it requires a corresponding modification of the previous local planner at corner $P_{n-2}P_{n-1}P_n$. Thus normally, we apply the transformation from P_i to P'_i first if a feasible solution does not exist following the original arrangement. If we still cannot obtain a desired path, afterwards we make a further attempt by translating the waypoints to P''_i .

3.3 Swept Path Analysis

3.3.1 Turning Templates

Until now we have discussed the local planner in the situations of free space and collision avoidance with vehicle width considered. When a large vehicle with great longitudinal length is involved, the issues of offtracking [103] and swing-out [104] cannot be neglected. Both of them are caused by the lateral excursion of the rear wheels when following the front trace in a turn and the capability of turning within a given area can be evaluated with given geometric features of road facilities [105]. Different from the posteriori and discrete methods used by most planners, in this section we are aiming to find a collision-free path with a priori and continuous collision detection to make full use of the space. Under such conditions, a swept path analysis of the vehicle undertaking a turning maneuver becomes particularly helpful for the identification of those extreme paths.

Considering a single-unit truck (all vehicles on a single frame according to [106]) of rectangular shape in 2D space, a standard turning template [107] is developed to represent the vehicle maneuverability based on a single turn, as shown in Fig. 3.15a. The turning template is composed of 6 discrete maneuvers based on the minimum turning curb-to-curb radius r_{c-c} , which can be regarded as half the minimum road width demanded for such a vehicle to successfully complete a U-turn. Apart from the standard style, there are other standards for turning templates as well. For example, the EU Directive 96/53/EC [108] requires that the vehicle makes a 270-degree turn within the gap between two circles of radii 5.3 and 12.5, as shown in Fig. 3.15b. It has been found that several essential characteristics of the vehicle, which include the minimum curb-to-curb turning radius, the wheelbase and the inner rear wheel, have direct effects on the roadway design [109]. Specifically, the swept region of each turning template illustrated in Fig. 3.15 is bounded by the paths of the outer (left) front corner and the right (inner) rear wheel of the vehicle body during a single turn. In the case of 3D space, the swept volume can be treated as a collection of all the planar swept paths formed by each layer of the geometric shape at different heights.

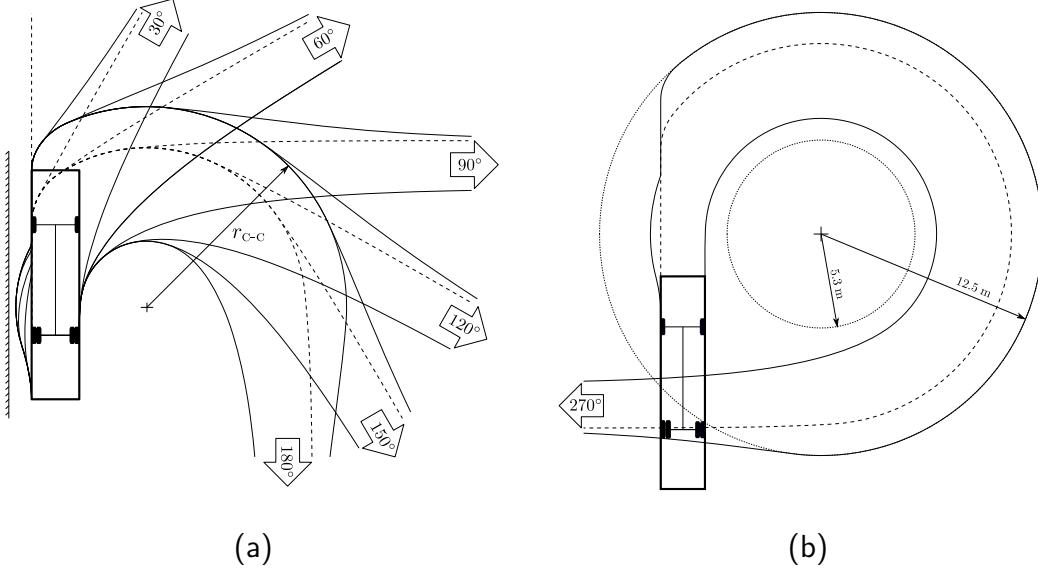


Figure 3.15: (a) A standard turning template for a single-unit truck. (b) A turning template based on the EU Standard.

Taking a vehicle steering around a corner with the kinematic model depicted in Fig. 3.6 as an example, the motion of vehicle roughly consists of three stages when following a spline path S_bS_e generated by a local planner, as illustrated in Fig. 3.16. As before, all the wheels are assumed to be pure rolling with no slipping nor skidding. The initial position of vehicle body is represented as PQ and $P'Q'$ denotes its final position. As illustrated in Fig. 3.16a, we consider a vehicle with its midpoint in its front axle as the reference point (it coincides with P at the start position), and the three stages are defined as follows.

- (i) \mathfrak{S}_1 : The start position PQ is initially aligned with a straight line PC_b . The vehicle performs a linear motion when its front wheels follow the straight line PC_b .
- (ii) \mathfrak{S}_2 : The front wheels travel along a nonlinear curve from S_b to S_e . In our case, the curve is composed of clothoids and possibly includes a piece of circular arc. The motion of the vehicle body is much more complicated and the traces of the rear wheels may not have analytical representations. It

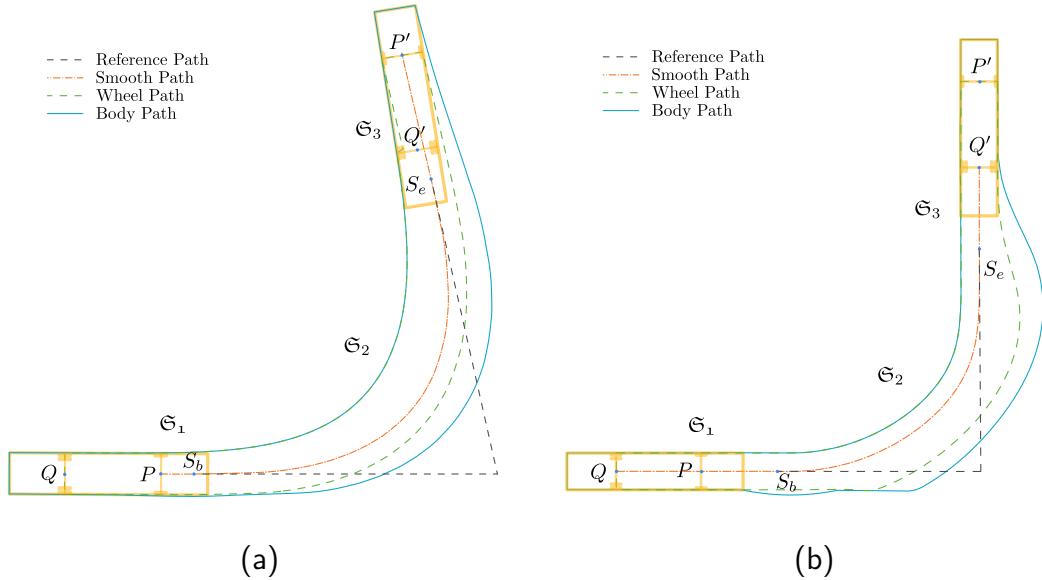


Figure 3.16: Turning maneuver following a local planner: (a) FWD model. (b) RWD model.

mainly depends on the position of the vehicle after finishing stage \mathfrak{S}_1 and the expression of the curve S_bS_e , which will be discussed later on.

- (iii) \mathfrak{S}_3 : In this stage, the vehicle begins to exit the nonlinear curve. Though the trace of the reference point is a straight line S_eP' , the motion of the vehicle is not linear because its initial pose right after stage \mathfrak{S}_2 does not align with S_eP' , and we will demonstrate that it can be described by a closed-form expression.

These stages apply to a vehicle with reference point P at the front axle, i.e., the FWD layout described in (3.41). However, for an RWD vehicle (3.40) with the reference point Q at the rear axle, the turning maneuver is analogous to a FWD vehicle except that the reference point Q has to follow the planned path, as shown in Fig. 3.16b. The equation of the vehicle motion is also different from that of the FWD case.

3.3.2 Mathematical Modeling

We are interested in the profile of the entire swept path, which in theory requires computing the trace of each point on the vehicle body. Considering that all these points belong to the same fixed frame for a single-unit vehicle, we are able to generate the profile once one path of any point except the reference point is obtained. Here we choose to focus on the path of the midpoint of the rear axle Q because it has not only clear kinematic properties but also an explicit geometric relationship with the reference point P . In the following text, we may refer to the midpoints of front and rear axles simply as front wheel and rear wheel for convenience. Based on the bicycle model described in Fig. 3.6 and the fact that the rear axle is fixed on the vehicle body and each wheel rotating around the rear axle cannot change the direction on its own, a pair of curves as parametric functions, i.e., $\mathbf{p}(t) = (x_p(t), y_p(t))^T$ and $\mathbf{q}(t) = (x_q(t), y_q(t))^T$ respectively, are used to represent the traces traveled by the front wheel P and the rear wheel Q . Here t stands for the elapsed time. Let $\hat{\tau}(t)$ be the unit vector from point Q to P , and from Fig. 3.6, it can be represented by

$$\hat{\tau}(t) = (\cos(\theta(t)), \sin(\theta(t)))^T, \quad (3.61)$$

where $\theta(t)$ is the orientation of the vehicle body at time t . Based on the analysis, the vehicle motion has two obvious characteristics:

- (i) The distance between P and Q keeps constant and equal to the wheelbase l_w :

$$\mathbf{p}(t) - \mathbf{q}(t) = l_w \hat{\tau}(t); \quad (3.62)$$

- (ii) The straight line PQ is tangent to the path $\mathbf{q}(t)$ at point Q ; in fact, the velocity at point Q is in the direction of $\hat{\tau}(t)$:

$$\frac{\dot{\mathbf{q}}(t)}{\|\dot{\mathbf{q}}(t)\|} = \hat{\tau}(t). \quad (3.63)$$

Since the wheelbase l_w cannot be changed, PQ can be treated as a rigid rod, which constrains the velocity component in the rod direction $\hat{\tau}(t)$ at both ends P and Q to be equal, namely,

$$\dot{\mathbf{q}}(t) \cdot \hat{\tau}(t) = \dot{\mathbf{p}}(t) \cdot \hat{\tau}(t). \quad (3.64)$$

Noticing that $\dot{\mathbf{q}}(t) \cdot \hat{\boldsymbol{\tau}}(t) = \|\dot{\mathbf{q}}(t)\|$, and using the relationships from (3.61) to (3.64), we are able to compute the path of the rear wheel Q as

$$\begin{pmatrix} x'_Q \\ y'_Q \end{pmatrix} = \frac{1}{l_w^2} \begin{pmatrix} x'_P (x_P - x_Q)^2 + y'_P (x_P - x_Q)(y_P - y_Q) \\ x'_P (x_P - x_Q)(y_P - y_Q) + y'_P (y_P - y_Q)^2 \end{pmatrix}, \quad (3.65)$$

or in vector form:

$$\dot{\mathbf{q}} = \frac{1}{l_w^2} \dot{\mathbf{p}} \cdot (\mathbf{p} - \mathbf{q}) \cdot (\mathbf{p} - \mathbf{q}). \quad (3.66)$$

In the above procedure we did not assume the velocity to be constant, thus the obtained equation system (3.65) or (3.66) has no restrictions on the vehicle velocity.

3.3.2.1 Following a Straight Line

Given a path traveled by the front wheel $\mathbf{p}(t)$, the rear wheel path $\mathbf{q}(t)$ can be obtained by solving (3.65). By looking into this equation system, we can study the paths formed by the rear wheel in previously mentioned stages \mathfrak{S}_1 - \mathfrak{S}_3 . Considering that our local planners in CASE 1-4, which mainly consist of three types of curves: straight lines (2.4), circular arcs (2.5), and clothoids (2.6), we focus on studying the rear wheel path following these paths.

The most common scenario is that the vehicle moves forward with steering angle being zero. Apparently, the path of Q is a straight line following the front wheel P just like the motion on a straight highway. However, it is not that easy to determine the rear wheel path considering different initial poses (positions and orientations) of the vehicle. We assume P moves following a straight line $\mathbf{p}(t) = (t, 0)^T$ ($t \geq 0$). Then (3.65) becomes

$$\begin{pmatrix} x'_Q(t) \\ y'_Q(t) \end{pmatrix} = \frac{1}{l_w^2} \begin{pmatrix} (t - x_Q(t))^2 \\ (x_Q(t) - t) y_Q(t) \end{pmatrix}. \quad (3.67)$$

Its solution can be analytically represented as

$$\begin{pmatrix} x_Q(t) \\ y_Q(t) \end{pmatrix} = \begin{pmatrix} l_w \frac{l_w - c_1 e^{\frac{2t}{l_w}}}{e^{\frac{2t}{l_w}}} + t \\ c_2 e^{\frac{1}{l_w^2} \int_0^t (x_Q(\xi) - \xi) d\xi} \end{pmatrix}, \quad (3.68)$$

where c_1 and c_2 are arbitrary constants. Without loss of generality, we set the initial condition $\mathbf{q}(0) = -l_w (\cos \theta, \sin \theta)^T$, where the vehicle orientation satisfies

$\theta \in [-\pi, \pi]$. Accordingly, the solution to (3.67) becomes

$$\begin{pmatrix} x_Q(t) \\ y_Q(t) \end{pmatrix} = \begin{pmatrix} \frac{2l_w(1-\cos\theta)}{1-\cos\theta+(\cos\theta+1)e^{\frac{2t}{l_w}}} + t - l_w \\ -\frac{l_w \sin\theta}{\cos\theta \sinh(\frac{t}{l_w}) + \cosh(\frac{t}{l_w})} \end{pmatrix}. \quad (3.69)$$

Although the solution holds for any value of θ , however, it does not mean that the initial pose can be chosen arbitrarily. As illustrated in Fig. 3.17, by assuming $l_w = 1$, totally 12 separate paths of $\mathbf{q}(t)$ are demonstrated by placing the rear wheel at different positions Q_i ($i = 0, 1, \dots, 11$), and the corresponding orientations are $\theta_i = (\frac{i}{6} - 1)\pi$, where $i = 0, 1, \dots, 11$. The paths with cusp points in the figure are regarded as invalid solutions because they involve reverse motions, which cannot happen with steering angle limited in range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ as stated in Section 3.1.4. On

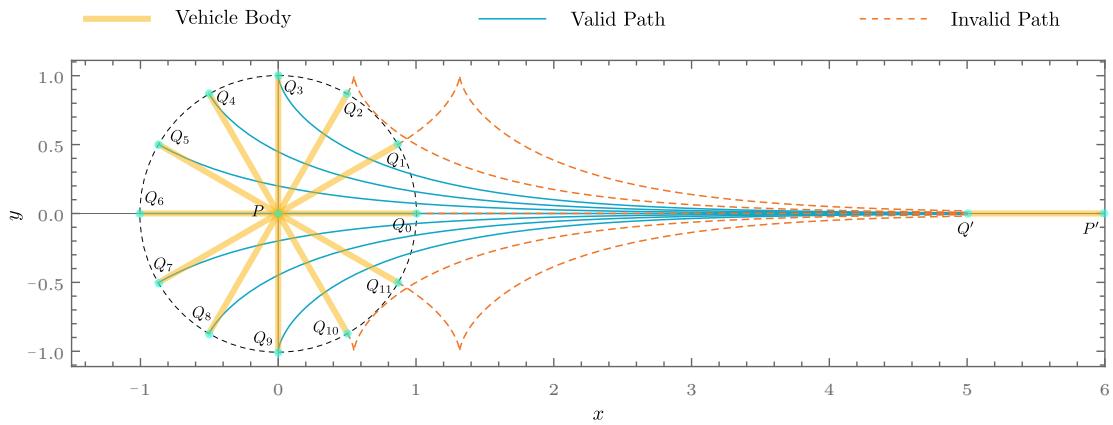


Figure 3.17: Rear wheel path following a straight line.

the other hand, the solutions in (3.69) correspond to the situations in stage \mathfrak{S}_3 of the local steering maneuver, where the rear wheel of the vehicle is leaving the nonlinear curve to approaching the straight line. Thus, we are quite interested in the time or distanced required to travel before Q begins a linear motion. In other words, when the rear wheel indeed follows a straight line, the vertical distance should be zero, i.e., $y_Q(t) = 0$. However, by observing (3.69), we notice that if $\theta \neq 0$, the vertical distance requires infinite time to be zero theoretically. So, we just set a threshold $\Delta_v > 0$ and the vehicle motion is treated to be linear if

$|y_Q(t)| \leq \Delta_v$ holds, and in practice we can set $\Delta_v = \frac{w_v}{2}$, where w_v is the width of the vehicle body. As a result, the elapsed time t is

$$t_v = \begin{cases} 0 & \text{if } \Delta_v \geq l_w \sin(|\theta|) \\ l_w \ln \left(\frac{1}{\Delta_v} \tan \left(\frac{|\theta|}{2} \right) \left(\sqrt{l_w^2 - \Delta_v^2} + l_w \right) \right) & \text{if } \Delta_v < l_w \sin(|\theta|) \end{cases} \quad (3.70)$$

while the corresponding orientation θ is

$$\theta_v = 2 \tan^{-1} \left(\frac{\Delta_v e^{\frac{t}{l_w}}}{\sqrt{l_w^2 - \Delta_v^2} + l_w} \right). \quad (3.71)$$

Thus, the rear wheel path may not be straight lines in stage \mathfrak{S}_1 and \mathfrak{S}_3 if the vehicle orientation satisfies $|\theta| > \theta_v$. If a linear motion is intended, the minimum road length between two adjacent waypoints should be $x_P(t_v) = t_v$.

3.3.2.2 Following a Circular Arc

Another frequently encountered situation is following a circular arc involved in CASE 2 discussed in Section 3.2.2. To analyze the shape of the rear wheel path when the front wheel P transfers from a clothoid to a circular arc, we represent the front wheel path as $\mathbf{p}(t) = r (\sin(\frac{t}{r}), -\cos(\frac{t}{r}))^T$, which is a circular arc of radius r with center located at $(0, 0)^T$ and zero initial tangential angle based on (2.5). The rear wheel path $\mathbf{q}(t)$ is determined by solving

$$\begin{pmatrix} x'_Q(t) \\ y'_Q(t) \end{pmatrix} = \frac{x_Q(t) \cos(\frac{t}{r}) + y_Q(t) \sin(\frac{t}{r})}{l_w^2} \begin{pmatrix} x_Q(t) - r \sin(\frac{t}{r}) \\ y_Q(t) + r \cos(\frac{t}{r}) \end{pmatrix}. \quad (3.72)$$

The solution cannot be explicitly expressed in closed-form, thus we use numerical results to illustrate the relationships with respect to the initial conditions. As before, we place the rear wheel at $\mathbf{q}(0) = -(l_w \cos \theta, r + l_w \sin \theta)^T$, where $\theta \in [-\pi, \pi]$ and the wheelbase l_w can be set to unit length for convenience. We compute the paths of the rear wheel Q according to the ratio of turning radius to wheelbase, as illustrated in Fig.3.18.

- (i) $\frac{r}{l_w} > 1$: It is the most common scenario that the radius of a curved section of the road r is larger than the vehicle wheelbase l_w . In this case, the effects of initial vehicle position PQ is our concern. Fig. 3.18a gives a demonstration

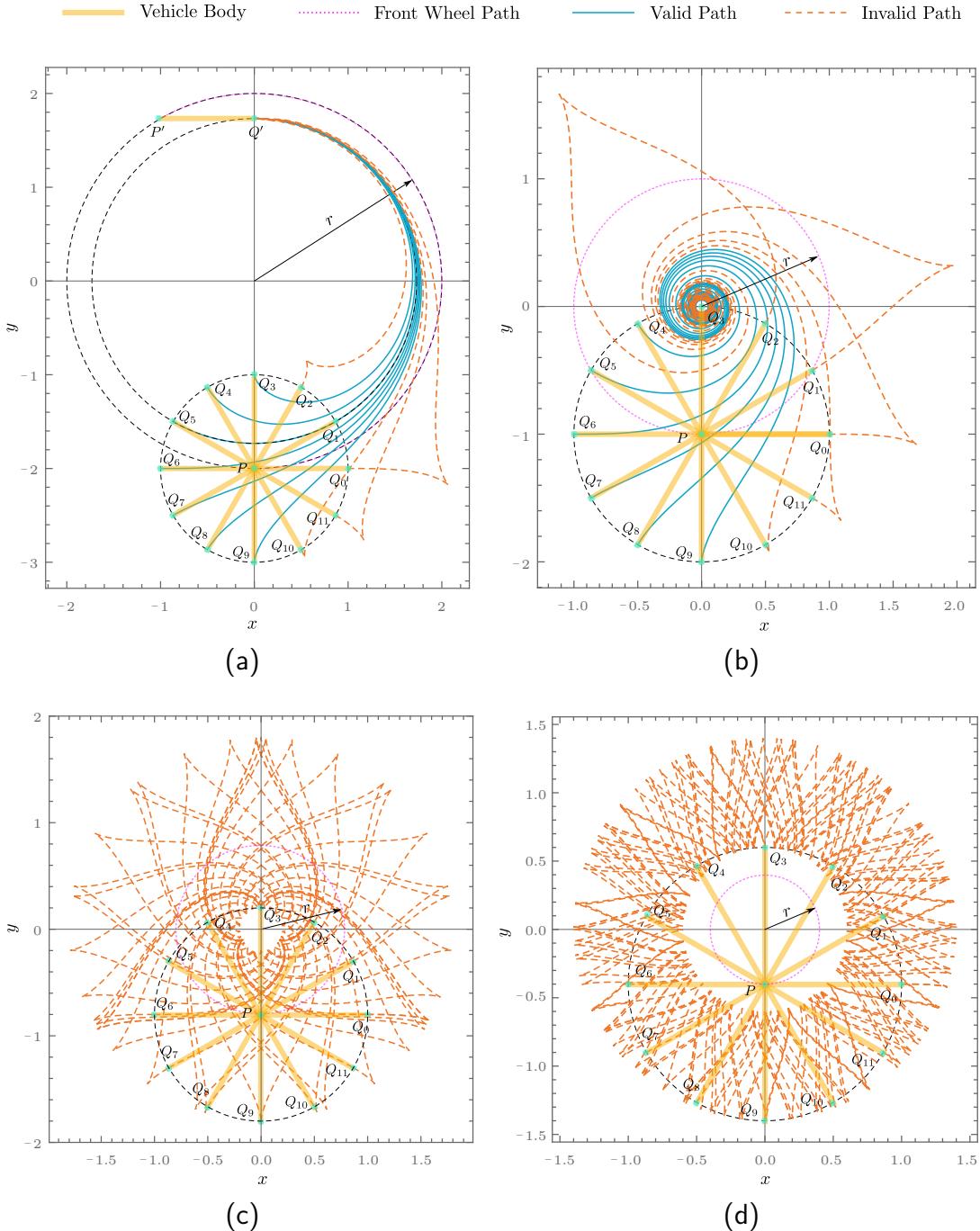


Figure 3.18: Rear wheel path following a circular arc: (a) $r = 2.0$, $l_w = 1.0$. (b) $r = 1.0$, $l_w = 1.0$. (c) $r = 0.8$, $l_w = 1.0$. (d) $r = 0.4$, $l_w = 1.0$.

of a group of curves $\mathbf{q}(t)$ with different initial values of angle $\theta_i = (\frac{i}{6} - 1)\pi$, where $i = 0, 1, \dots, 11$. Again, those curves with initial angle $\theta_i > \frac{\pi}{2}$ are

considered as invalid paths and discarded as a consequence. It reveals that all the rear wheel paths evolve into a circular shape if the elapsed time is long enough. The radius of this inner circle can be found based on the fact that the line segment PQ is always tangent to the circle at point Q :

$$r_Q = \sqrt{r^2 - l_w^2}, \quad (3.73)$$

which apparently holds only when $\frac{r}{l_w} > 1$ to ensure the existence of the circle, i.e., $r_Q > 0$. From (3.73) we know that if the initial orientation $\theta = -\arcsin \frac{l_w}{r}$, the rear wheel path $\mathbf{q}(t)$ is exactly a circle of radius $\sqrt{r^2 - l_w^2}$. In Fig. 3.18a with $r = 2$, it corresponds to the pose with $\theta_5 = -\frac{\pi}{6}$.

- (ii) $\frac{r}{l_w} = 1$: Vehicles satisfying our constraint that the steering angle falling in $[-\frac{\pi}{2}, \frac{\pi}{2}]$ can theoretically follow a circular arc with its front wheels and keep the position of Q unchanged, just as the pose PQ_3 shown in Fig. 3.18b which limits the steering angle to $-\frac{\pi}{2}$ all the time. For other initial poses with $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2}]$, the rear wheel paths degenerate to a point coinciding with Q_3 after a long period of time.
- (iii) $\frac{r}{l_w} < 1$: Because here we focus on vehicles with P as the reference point, which corresponds to the FWD model (3.41) and its maximum turning curvature is defined as $\kappa_P = \frac{\sin \phi}{l_w}$ in (3.42). From this we know that the minimum turning radius of this kind of vehicle is $r_{\min} = \frac{l_w}{\sin |\phi|} \geq l_w$ where the steering angle satisfies $|\phi| \leq \phi_{\max} \in [0, \frac{\pi}{2}]$. Thus the vehicle is not capable to follow a circular arc of radius $r < l_w \leq r_{\min}$. Thought the paths are not applicable in this situation, the results are shown in Fig. 3.18c and Fig. 3.18d respectively for the sake of completeness. It is noticed that the result has a periodic behavior, and can be classified into two different categories according to the pattern of the obtained curves. When $\frac{1}{2} \leq \frac{r}{l_w} < 1$, the rear wheel path pass through the circle of radius r , while no intersections appear between the rear wheel trace and the circular path of the front wheel when $\frac{r}{l_w} < 0.5$.

3.3.2.3 Following a Clothoid

Apart from the linear segment and circular arc as previously mentioned, another significant type of paths encountered is clothoid (2.6), which is extensively involved in CASE 1-4. As before, we demonstrate the computational results of the rear wheel path $\mathbf{q}(t)$ given different initial positions Q_i , where $i = 0, 1, \dots, 11$, as shown in Fig. 3.19. The clothoid curve as the front wheel path in Fig. 3.19a is

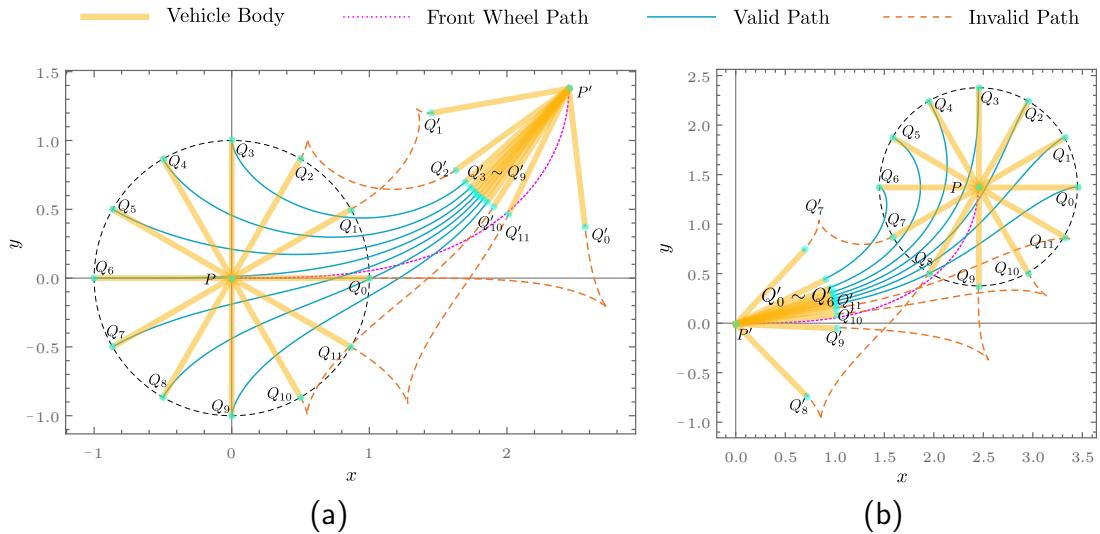


Figure 3.19: Rear wheel path following a clothoid: (a) $\mathbf{p}(t)$ starts at $(0, 0)^T$. (b) $\mathbf{p}(t)$ starts at $(\pi C_f(1), \pi S_f(1))^T$.

$$\mathbf{p}(t) = \pi \begin{pmatrix} C_f\left(\frac{t}{\pi}\right) \\ S_f\left(\frac{t}{\pi}\right) \end{pmatrix}, \quad (3.74)$$

where $t \in [0, \pi]$. It starts at $(0, 0)^T$ and has a sharpness of $\sigma = \frac{1}{\pi}$ and the winding angle reaches $\frac{\pi}{2}$ when $t = \pi$. The vehicle of wheelbase $l_w = 1$ is initially placed at 12 different locations on a circle $(-l_w \cos \theta, -l_w \sin \theta)^T$ and the corresponding final positions are computed as $P'Q'_i$ ($i = 0, 1, \dots, 11$) after solving the equation system numerically which is obtained by substituting (3.74) into (3.65). It can be seen that $PQ_3 \sim PQ_9$ are valid positions which have steering angles within $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Since all the clothoids involved in CASE 1-4 are in the form of symmetric or asymmetric pairs, we also compute the rear wheel path following its counterpart given the same initial positions. The clothoid $(\pi C_f(1 - \frac{t}{\pi}), \pi S_f(1 - \frac{t}{\pi}))^T$ ($t \in [0, \pi]$) as the front

wheel path illustrated in Fig. 3.19b is not the actual symmetric reflection of the curve (3.74) but a simplified form after proper transformations. The start positions are all placed on a circle $(\pi C_f(1) - l \cos(\theta), \pi S_f(1) - l \sin(\theta))^T$ and $PQ_0 \sim PQ_6$ out of the computed results are recognized as valid candidates, because each of them has an absolute value of the steering angle no greater than $\frac{\pi}{2}$.

3.3.2.4 Front Wheel Path of an RWD Vehicle

Until now we have computed the rear wheel paths for all the primitive paths including straight lines, circular arcs and clothoids involved in CASE 1-4. However, only FWD vehicle model with front wheel P as the reference point is considered and analyzed. With regard to the RWD model (3.40) whose reference point is the rear wheel Q , its front path $\mathbf{p}(t)$ is undetermined. Similarly, only the special cases are taken into consideration when the rear wheel Q follows a straight line, a circular arc and a clothoid. We can obtain the path $\mathbf{p}(t)$ directly by solving the equation system (3.65) or (3.66), which is straightforward and does not need any additional requirements. However, it is preferable to avoid the complex computations of solving differential equations. Given the rear wheel path $\mathbf{q}(t)$, by substituting (3.63) to (3.62), the front wheel path is obtained as:

$$\mathbf{p}(t) = l_w \frac{\dot{\mathbf{q}}(t)}{\|\dot{\mathbf{q}}(t)\|} + \mathbf{q}(t), \quad (3.75)$$

where l_w is the wheelbase of the vehicle (3.40). Consequently, the results of following three basic types of path primitives are as follows.

- (i) Straight line $\mathbf{q}(t) = (t, 0)^T$: The front wheel is also following a straight line expressed by

$$\mathbf{p}(t) = \begin{pmatrix} t + l_w \\ 0 \end{pmatrix}. \quad (3.76)$$

- (ii) Circular arc $\mathbf{q}(t) = r (\sin(\frac{t}{r}), -\cos(\frac{t}{r}))^T$ of radius r : The front wheel path is also a circular arc

$$\mathbf{p}(t) = \sqrt{l_w^2 + r^2} \begin{pmatrix} \cos\left(\frac{t}{r} - \psi\right) \\ \sin\left(\frac{t}{r} - \psi\right) \end{pmatrix}, \quad (3.77)$$

where $\psi = \arctan\left(\frac{r}{l_w}\right)$. Fig. 3.20a is an example of an RWD vehicle with a wheelbase of unit length following a piece of circular arc of radius $r = 2$. The initial and final poses of the vehicle are denoted as PQ and $P'Q'$ respectively.

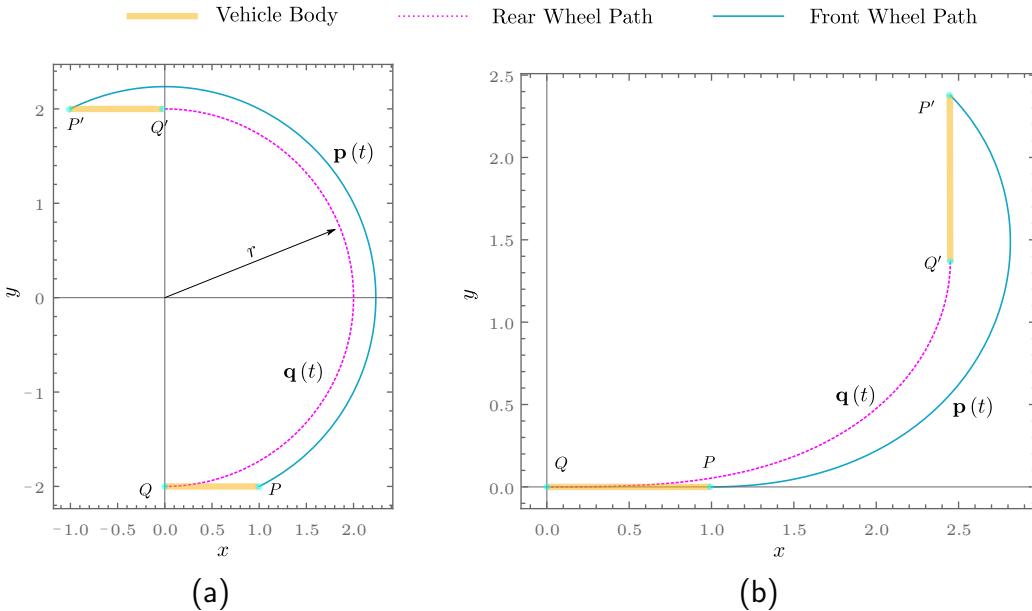


Figure 3.20: Front wheel path of an RWD model: (a) Following a circular arc of radius $r = 2$. (b) Following a clothoid of sharpness $\sigma = \frac{1}{\pi}$.

- (iii) Clothoid $\mathbf{q}(t) = \sqrt{\frac{\pi}{\sigma}} (C_f(\sqrt{\frac{\sigma}{\pi}}t), S_f(\sqrt{\frac{\sigma}{\pi}}t))^T$ of sharpness σ : The front wheel path is obtained as

$$\mathbf{p}(t) = \begin{pmatrix} \sqrt{\frac{\pi}{\sigma}} C_f(\sqrt{\frac{\sigma}{\pi}}t) + l_w \cos\left(\frac{\sigma t^2}{2}\right) \\ \sqrt{\frac{\pi}{\sigma}} S_f(\sqrt{\frac{\sigma}{\pi}}t) + l_w \sin\left(\frac{\sigma t^2}{2}\right) \end{pmatrix}, \quad (3.78)$$

which actually can be regarded as a superposition of the original clothoid curve and a circle $(l_w \cos\left(\frac{\sigma t^2}{2}\right), l_w \sin\left(\frac{\sigma t^2}{2}\right))^T$, as illustrated in Fig. 3.20b.

Now we are able to obtain the path of P or Q when following a reference path no matter it is a FWD or RWD vehicle. Back to the topic of collision checking, the final goal is to compute the swept path of the vehicle. Given any point M on

the vehicle body, the path $\mathbf{m}(t)$ traveled by M can be computed if both $\mathbf{p}(t)$ and $\mathbf{q}(t)$ are known:

$$\mathbf{m}(t) = \mathbf{q}(t) + \frac{l_{QM}}{l_w} \mathbf{R}(\beta) (\mathbf{p}(t) - \mathbf{q}(t)) , \quad (3.79)$$

where $l_{QM} = \|QM\|$, l_w is the vehicle wheelbase, β is the signed angle from \overrightarrow{QP} to \overrightarrow{QM} , and $\mathbf{R}(\beta) = \begin{pmatrix} \cos \beta & -\sin \beta \\ \sin \beta & \cos \beta \end{pmatrix}$ represents the rotation matrix with respect to angle β .

Fig. 3.21 shows the computational results of the paths traveled by the reference points as well as each characteristic point on the vehicle body for both the FWD and RWD models. According to (3.41), we pick the position representing the front wheel P as the reference point for a FWD model to follow a path $\mathbf{p}(t)$ generated by a local planner, as illustrated in Fig. 3.21a. Other paths $\mathbf{p}_A(t), \mathbf{p}_B(t), \dots, \mathbf{p}_H(t)$

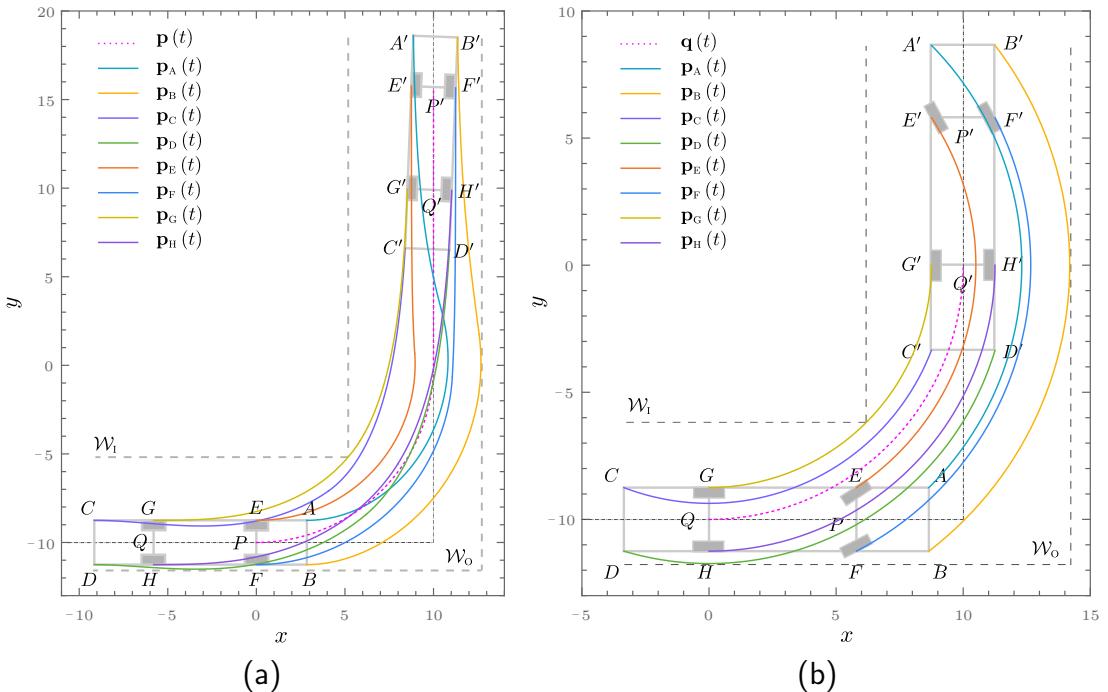


Figure 3.21: Computation of the swept path: (a) FWD model. (b) RWD model.

are computed based on the relationship with respect to the rear wheel path (3.79). Taking a vehicle of rectangular shape in a single turn as an instance, it is obvious that the swept profile is determined by three important points: the outer front

corner B , the outer rear corner D , and the inner rear wheel G . Correspondingly, the collision-free condition can be satisfied if and only if the inner boundary of the road \mathcal{W}_I and the track $\mathbf{p}_G(t)$ have no intersections, and simultaneously no crossing occurs between the outer boundary \mathcal{W}_O and $\mathbf{p}_B(t), \mathbf{p}_D(t)$. Similarly, these conclusions still apply for a RWD model, which uses its rear wheel Q as the reference point (Fig. 3.21b).

Chapter 4

Approximation of Clothoid Using Bézier Curves for Path Planning

In this chapter, we propose a generic Bézier curve approximation method which can deal with any types of clothoids. Since commonly used approaches only focus on unit-length clothoid, procedures such as reparameterizing, scaling and rescaling are required in most scenarios. We are aiming to deal with any clothoid not necessarily of unit arc length, thus avoiding these irrelevant normalization procedures. Accordingly, we apply G^3 condition to the quintic Bézier curve and the elementary clothoid to obtain a Bézier representation with only two free shape parameters. To determine the optimal values for these parameters, a reasonable approximation with regard to the curve length is applied and the assumption is validated through numerical optimizations and error analysis. Also, we resolve the divergence problem by restricting the arc length of the elementary clothoid within a safe region which will be discussed later. The algorithm proposed is thus highly robust being able to produce reasonable results in a deterministic way. A lookup table is constructed to efficiently compute the so-called basic clothoid composed of any number of elementary clothoids segments. Due to the fact that a general clothoid can be represented with a piece of basic clothoid through appropriate geometric transformation, its corresponding Bézier approximation can be obtained using the existing lookup table. The error function is defined based on curvature profiles and a numerical analysis shows that the accuracy can be adjusted by choosing different winding angle parameters. Both unit-length and non-unit length approximation

examples are demonstrated using proposed approach and comparisons are made against several other algorithms. We apply our method to a practical path smoothing problem and demonstrate that the resulting path achieves better quality from perspective of path length and curvature maxima. We also compared the accuracy and performance of the proposed method with the circular arc based algorithm recently reported in [1] and the RBC approximation [64]. The related work [110] has been accepted for publication.

The rest of this chapter is organized as follows. Section 4.1 describes the approximation procedures of any elementary clothoids with G^3 continuity. The results are used to establish a basic clothoid approximation for the purpose of creating a lookup table with error analysis conducted in Section 4.2. In Section 4.3, approximation of a general clothoid with arbitrary parameters is presented. The effectiveness of the proposed method is tested and compared with other algorithms in Section 4.4.

4.1 Elementary Clothoid and Bézier Curve

4.1.1 Elementary Clothoid

Clothoid (2.6) is also referred to as Euler spiral or Cornu spiral, which can be expressed in parametric form as in [51]:

$$\mathbf{F}(s) = \begin{pmatrix} x_0 + \int_0^s \cos \left(\theta_0 + \kappa_0 \xi + \frac{\sigma \xi^2}{2} \right) d\xi \\ y_0 + \int_0^s \sin \left(\theta_0 + \kappa_0 \xi + \frac{\sigma \xi^2}{2} \right) d\xi \end{pmatrix}, \quad (4.1)$$

where σ is the sharpness, and $(x_0, y_0)^T$, κ_0 , θ_0 are the initial position, curvature, and winding angle, respectively. Based on (4.1) we can define a piece of elemental clothoid denoted by $\mathbf{F}_\varepsilon(s)$ with a positive sharpness $\sigma_\varepsilon > 0$. Furthermore, it is restricted to the first quadrant, thus $s \in [0, \infty)$ and initial conditions are specified as $(x_0, y_0) = (0, 0)$, $\theta_0 = 0$ and $\kappa_0 \geq 0$:

$$\mathbf{F}_\varepsilon(s) = \sqrt{\frac{\pi}{\sigma_\varepsilon}} \mathbf{R} \left(-\frac{\kappa_0^2}{2\sigma_\varepsilon} \right) \begin{pmatrix} \delta_c(s) \\ \delta_s(s) \end{pmatrix}, \quad (4.2)$$

where $\delta_c(s) = C_f(\frac{\sigma\varepsilon s + \kappa_0}{\sqrt{\pi}\sigma\varepsilon}) - C_f(\frac{\kappa_0}{\sqrt{\pi}\sigma\varepsilon})$, $\delta_s(s) = S_f(\frac{\sigma\varepsilon s + \kappa_0}{\sqrt{\pi}\sigma\varepsilon}) - S_f(\frac{\kappa_0}{\sqrt{\pi}\sigma\varepsilon})$. Here, $\mathbf{R}(\theta)$ represents the planar rotation matrix

$$\mathbf{R}(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}. \quad (4.3)$$

4.1.2 Quintic Bézier Curve

Assuming that the quintic Bézier curve

$$\mathbf{B}_\varepsilon(t) = \sum_{i=0}^5 \binom{5}{i} (1-t)^{5-i} t^i \mathbf{V}_i, \quad t \in [0, 1] \quad (4.4)$$

is a feasible approximation to the elementary clothoid curve (4.2), let S_b and S_c denote the total length of $\mathbf{B}_\varepsilon(t)$ and $\mathbf{F}_\varepsilon(s)$ respectively. Without loss of generality, in addition to the G^2 condition mentioned in [60] we need to apply the following constraints to $\mathbf{B}_\varepsilon(t)$ at two ends in order to have G^3 continuity:

$$\begin{aligned} \kappa'_b(s)|_{s=0} &= \kappa'_c(s)|_{s=0}, \\ \kappa'_b(s)|_{s=S_b} &= \kappa'_c(s)|_{s=S_c}, \end{aligned} \quad (4.5)$$

where κ_b and κ_c represent the curvature profiles of $\mathbf{B}_\varepsilon(t)$ and $\mathbf{F}_\varepsilon(s)$, respectively. Considering the inconvenience of representing the curvature profile of a quintic Bézier curve with regard to arc length analytically as required by (4.5), we choose an equivalent condition using Beta-constraints with shape parameters [39] instead. Accordingly, the control points \mathbf{V}_i ($i = 0, 1, 2, \dots, 5$) are expressed as

$$\begin{pmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \mathbf{V}_4 \\ \mathbf{V}_5 \end{pmatrix} = \begin{pmatrix} \mathbf{F}_\varepsilon(0) \\ \frac{\beta_1}{5} \mathbf{F}'_\varepsilon(0) + \mathbf{F}_\varepsilon(0) \\ \frac{\beta_1^2}{20} \mathbf{F}''_\varepsilon(0) + \frac{8\beta_1 + \beta_2}{20} \mathbf{F}'_\varepsilon(0) + \mathbf{F}_\varepsilon(0) \\ \frac{\gamma_1^2}{20} \mathbf{F}''_\varepsilon(l) + \frac{\gamma_2 - 8\gamma_1}{20} \mathbf{F}'_\varepsilon(l) + \mathbf{F}_\varepsilon(l) \\ -\frac{\gamma_1}{5} \mathbf{F}'_\varepsilon(l) + \mathbf{F}_\varepsilon(l) \\ \mathbf{F}_\varepsilon(l) \end{pmatrix}, \quad (4.6)$$

where $l = S_c$ and shape parameters β_2 and γ_2 are indeed dependent on the other two free parameters $\beta_1 > 0$ and $\gamma_1 > 0$, and likewise can be calculated as in [62] (also refer to Section A.1 in Appendix A):

$$\beta_2(\beta_1, \gamma_1) = \frac{B(\beta_1, \gamma_1)}{D(\beta_1, \gamma_1)}, \quad \gamma_2(\beta_1, \gamma_1) = \frac{G(\beta_1, \gamma_1)}{D(\beta_1, \gamma_1)}. \quad (4.7)$$

The common denominator function $D(\beta_1, \gamma_1)$ is expressed as

$$D(\beta_1, \gamma_1) = \beta_1 \gamma_1 \kappa_0 \kappa_1 - \sin^2 \theta, \quad (4.8)$$

where $\kappa_1 = \kappa_0 + \sigma_\varepsilon l$ is the final curvature at endpoint ($s = l$).

4.1.3 Approximation of the Elementary Clothoid

To guarantee that the approximation error is within acceptable tolerance, the winding angle θ is restricted to

$$0 < \theta \leq \frac{\pi}{2} \quad (4.9)$$

as suggested in [56]. Taking a piece of $\mathbf{F}_\varepsilon(s)$ with $s \in [0, l]$ and $\kappa_0 = 0$ as an example, its sharpness σ_ε is limited to $(0, \frac{\pi}{l^2}]$ by applying (4.9). Although it seems to be a limitation here, later on we will show that a general clothoid can be approximated free of such constraints.

4.1.3.1 Solve the Optimization Problem

The approximation accuracy involves three types of errors with respect to position, orientation, and curvature. Given that higher order continuity is more difficult to guarantee, we focus on minimizing the curvature error first and investigate the other two at a later time. The curvature error $\epsilon_\kappa(s)$ is defined as a function of arc length parameter s by combining the expressions of absolute and relative curvature difference [61]. To compute $\epsilon_\kappa(s)$, the curvatures of both the clothoid and corresponding Bézier curve have to be represented with arc length parameter as $\kappa_c(s)$ and $\kappa_b(s)$ respectively. Reparameterization of the Bézier curve with regard to arc length s is achieved using the iterative sampling algorithm in [98]. Therefore, the maxima of $\epsilon_\kappa(s)$ over the entire domain $[0, l]$

$$\epsilon_\kappa = \max_{s \in [0, l]} \epsilon_\kappa(s) = \max_{s \in [0, l]} \frac{|\kappa_b(s) - \kappa_c(s)|}{\max \{ |\kappa_c(s)|, 1 \}} \quad (4.10)$$

is used to quantify the approximation error. Let μ represent the tolerance, and the approximated curve is acceptable if $\epsilon_\kappa \leq \mu$. $\mu = 0.05$ can be adopted [111] in most cases.

As in [62], by introducing a new parameter $k = \kappa_0 - \kappa_1$ and assuming β_1 and γ_1 are relatively stable as k varies, we can represent sharpness and curvature values with regard to winding angle θ and parameter k :

$$\begin{aligned}\sigma_\varepsilon &= -\frac{k}{l}, \\ \kappa_0 &= \frac{k}{2} + \frac{\theta}{l}, \\ \kappa_1 &= -\frac{k}{2} + \frac{\theta}{l},\end{aligned}\tag{4.11}$$

where $l > 0$ is the total arc length. Substituting (4.6) and (4.7) into (4.4) successively, the desired quintic Bézier curve $\mathbf{B}_\varepsilon(t)$ can be expressed as a function of the shape parameters β_1 and γ_1 . Then by applying the relationship (4.11), the obtained Bézier curve and elementary clothoid (4.2) are reformulated as $\tilde{\mathbf{B}}_\varepsilon(s, l, k, \theta, \beta_1, \gamma_1)$ and $\mathbf{F}_\varepsilon(s, l, k, \theta)$ separately after reparameterization with regard to arc length. In such a way the maximum error of curvature (4.10) actually can be written as $\epsilon_\kappa(\beta_1, \gamma_1)$. The desirable values of the two free variables, i.e., β_1 and γ_1 , can be determined by solving for the constrained optimization problem below

$$\begin{aligned}\min \quad & \epsilon_\kappa(\beta_1, \gamma_1) \\ \text{s.t.} \quad & \beta_1 > 0, \gamma_1 > 0\end{aligned}\tag{4.12}$$

Accordingly, the optimal quintic Bézier curve among all the candidates satisfying G^3 continuity condition can be computed.

With values of $\kappa_b(s)$ and $\kappa_c(s)$ computed subsequently, the minima of $\epsilon_\kappa(\beta_1, \gamma_1)$ can be acquired. In [62] an analysis of a set of generalized Cornu spirals with unit arc length ($l = 1$), suggests that $(\beta_1, \gamma_1) = (1, 1)$ is an effective approximation of the solution to problem (4.12) for clothoid curves. From (4.6) we can see that β_1 and γ_1 are closely related to the edge lengths of the control polygon: $\|\mathbf{V}_1 - \mathbf{V}_0\| = \frac{\beta_1}{5}$ and $\|\mathbf{V}_5 - \mathbf{V}_4\| = \frac{\gamma_1}{5}$, which increase linearly with the total arc length l of the clothoid. Consequently, it is reasonable to assume that $(\beta_1, \gamma_1) = (l, l)$ is the optimal solution for clothoids of non-unit arc length.

A numerical optimization procedure to solve for (4.12) is implemented to verify our assumption by performing a search from the initial position $(\beta_1, \gamma_1) = (l, l)$.

The objective function $\epsilon_\kappa(\beta_1, \gamma_1)$ is evaluated by calculating the differences of corresponding values at uniformly sampled positions of two parameterized curves, and due to lack of derivative information, its gradient can be estimated using finite differences, or adopting derivative free optimization algorithms like Nelder-Mead method [112] instead. Fig. 4.1 shows different $\epsilon_\kappa(\beta_1, \gamma_1)$ values computed over region $[0, 10l] \times [0, 10l]$ and totally 10 points are sampled uniformly on each curve. The arc length l is set to 0.1, 1.0, and 10.0 respectively. In Fig. 4.1 the region in a darker color has smaller values, and the black points around (l, l) in the figure represent coordinates evaluated during the search process with a ring of minima $|\epsilon_\kappa - \min \epsilon_\kappa(\beta_1, \gamma_1)| \leq 0.05$. Other numerical evaluations for different combinations of l and θ values yield consistent results that the distributions of $\epsilon_\kappa(\beta_1, \gamma_1)$ have similar patterns while the minima always appears around (l, l) . This confirms that the optimal values of β_1 and γ_1 can be regarded as independent of winding angle. To further elaborate the relationship among shape parameters (β_1, γ_1) , arc length l and k , a numerical search across different combinations is performed. The optimal values of β_1 and γ_1 are represented in the form of $\beta_1(l, k, \theta)$ and $\gamma_1(l, k, \theta)$ respectively as shown in Fig. 4.2. It can be clearly seen that the obtained β_1 and γ_1 are approximately proportional to arc length l and the effects from both winding angle θ and k can be neglected. This is consistent with our assumption that

$$\beta_1(l, k, \theta) \doteq l, \quad \gamma_1(l, k, \theta) \doteq l \quad (4.13)$$

can be considered as the optimal position.

4.1.3.2 Divergence Problem

However, another problem occurs when we adhere to G^3 continuity since such values of β_2 and γ_2 do not always exist by taking a close look at (4.7). More specifically, by substituting (4.11) into (4.8), divergences will occur when $D(\beta_1, \gamma_1)$ approaches zero if

$$k = \pm k_d = \pm 2 \sqrt{\frac{\theta^2}{l^2} - \frac{\sin^2(\theta)}{\beta_1 \gamma_1}}. \quad (4.14)$$

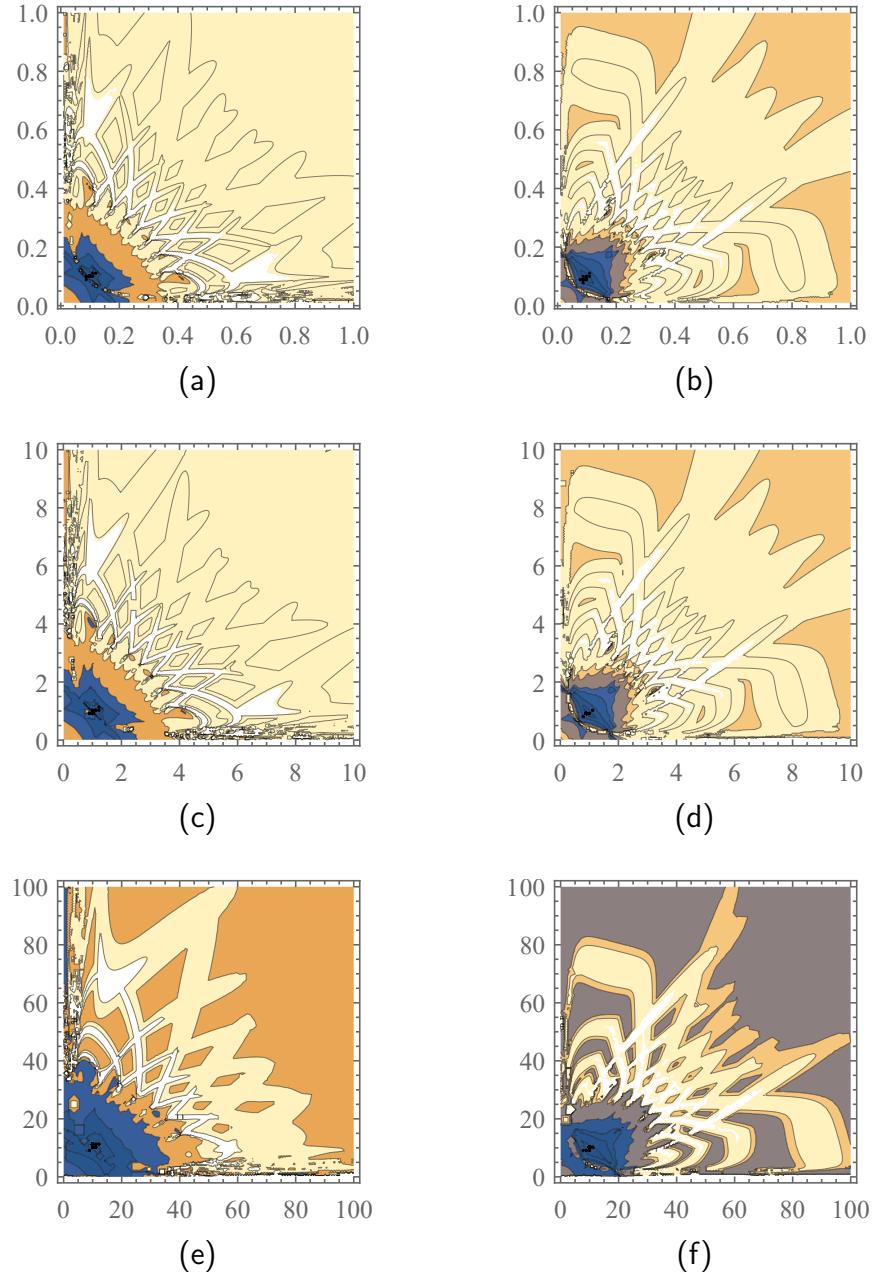


Figure 4.1: $\epsilon_k(\beta_1, \gamma_1)$: (a) $l = 0.1, \theta = \frac{\pi}{4}$. (b) $l = 0.1, \theta = \frac{\pi}{2}$. (c) $l = 1, \theta = \frac{\pi}{4}$. (d) $l = 1, \theta = \frac{\pi}{2}$. (e) $l = 10, \theta = \frac{\pi}{4}$. (f) $l = 10, \theta = \frac{\pi}{2}$.

The divergence problem is illustrated in Fig. 4.3a, where functions of $\beta_2(k)$ and $\gamma_2(k)$ have infinite values at $k = \pm k_d$ ($k_d = 1.1775$) with linear interpolations applied as $k \in (-2 k_d, 2 k_d)$. In this case, the values of $\beta_2(k)$ and $\gamma_2(k)$ (see Section A.2

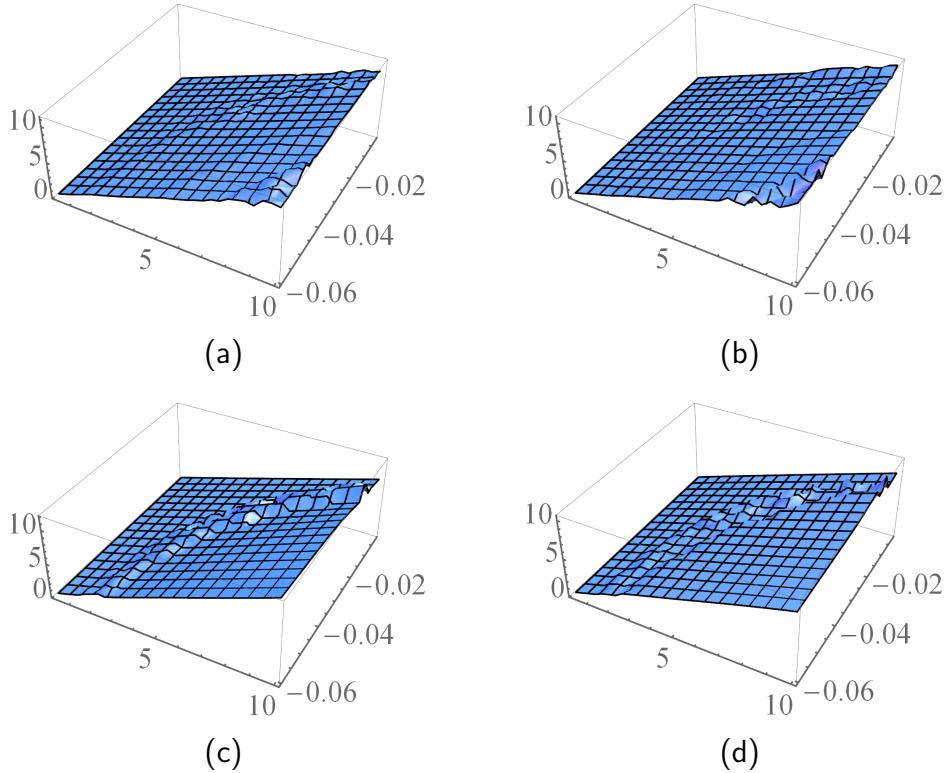


Figure 4.2: Optimal β_1 and γ_1 values in region $[0, 10] \times [-0.06, 0]$: (a) $\beta_1(l, k, \theta = \frac{\pi}{4})$. (b) $\gamma_1(l, k, \theta = \frac{\pi}{4})$. (c) $\beta_1(l, k, \theta = \frac{\pi}{2})$. (d) $\gamma_1(l, k, \theta = \frac{\pi}{2})$.

in Appendix A) have infinite discontinuities at $k = \pm k_d$ and extra compromised procedures are required to obtain a continuous and smooth curve. In [62], linear approximations of β_2 and γ_2 are made in region $(-2 k_d, 2 k_d)$ to smoothen the irregular part. Alternatively, a similar search process is performed to locate the optimal position of $(\beta_1, \beta_2, \gamma_1, \gamma_2)$ by just considering the G^2 condition as described in [113]. Although smooth approximations can be always found with these methods, the property of G^3 continuity cannot be retained unfortunately. Here, we propose a new method to avoid the divergent positions as well as preserving the G^3 property by carefully adjusting the value of l . From (4.11) and $l > 0, \sigma > 0$, it can be found that $k < 0$. Thus only the negative divergent point $k = -k_d$ needs to be taken into consideration. Let $\xi_{0,k}$ and $\xi_{1,k}$ be adjustable positive variables, and the divergent point and its neighborhood can be represented as $(-\xi_{0,k} k_d, -\xi_{1,k} k_d)$. So by specifying $k \in (-\infty, -\xi_{0,k} k_d] \cup [-\xi_{1,k} k_d, 0)$, or equivalently, $l \in (0, \frac{\xi_{1,k} k_d}{\sigma}] \cup [\frac{\xi_{0,k} k_d}{\sigma}, +\infty)$, we

are able to guarantee G^3 continuity free of divergence problem. It is fine to have the approximated Bézier curve with a different arc length (typically shorter) from the original clothoid. In next sections we will discuss how to make final approximation with the exactly the same arc length.

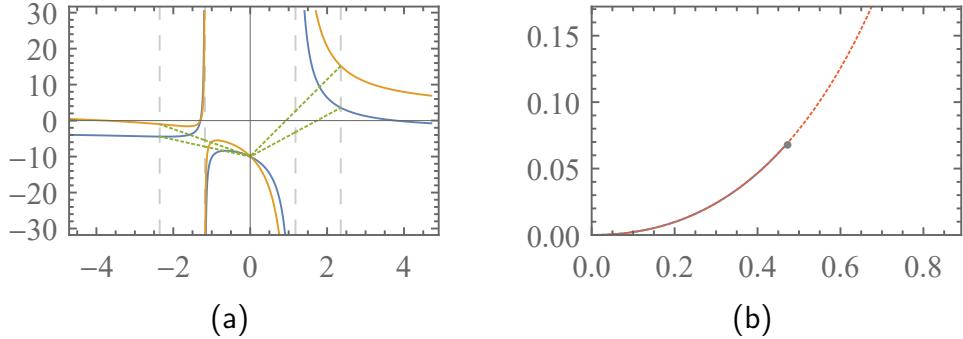


Figure 4.3: Divergence problem is shown by setting $\theta = \frac{\pi}{3}, l = 1$ and $k = -1.1 k_d$.
(a) $\beta_2(k)$ (blue solid), $\gamma_2(k)$ (orange solid), and linear interpolations (green dashed).
(b) Clothoid (red dashed) and its Bézier approximation (blue solid) using proposed method.

4.2 Basic Clothoid Approximation

4.2.1 Basic Clothoid

Once all the shape parameters β_i and γ_i ($i = 1, 2$) are determined according to (4.7) and (4.13), any elementary clothoid $\mathbf{F}_\varepsilon(s)$ defined on $[0, l]$ with constraint (4.9) can be approximated with G^3 continuity. However, for those curves out of bounds, the above approximation procedure cannot be applied directly and appropriate preprocessing is needed. Taking a piece of clothoid with a positive initial curvature as an example, it may have a large sharpness which makes θ out of range easily as s increases according to equation (2.7). Elementary segments can only guarantee reasonable results when condition (4.9) is satisfied. Nevertheless, they are flexible to approximate a clothoid with arbitrary starting and ending curvatures κ_0, κ_1 . By concatenating these elementary segments successively, theoretically a complete clothoid curve can be formed with curvature κ changing across the entire domain $(-\infty, +\infty)$.

In practice, we only need to focus on the half of this composed curve located in the first quadrant referred to as basic clothoid and denoted by $\mathbf{F}_c(s)$. We only consider the finite range case consisting of N elementary segments in total with $\kappa \geq 0$ and $\sigma_c = \sigma_\varepsilon > 0$. Also, the initial status is specified as $\kappa_{0,c} = 0$ and $\theta_{0,c} = 0$. Now considering the i -th piece among the first N elementary segments, $\mathbf{F}_{\varepsilon,i}(s)$ ($1 < i \leq N$), its starting curvature $\kappa_{0,i}$ and winding angle $\theta_{0,i}$ should be equal to the ending values of the previous segment $\kappa_{1,i-1}$, $\theta_{1,i-1}$ respectively. Because σ_ε is set to be positive, $\kappa_{0,i}$ and $\theta_{0,i}$ become larger as i increases. On the other hand, the winding angle of an elementary segment has to be within the bound as well according to (4.9):

$$\theta = \kappa_0 l + \frac{\sigma_\varepsilon}{2} l^2 \leq \frac{\pi}{2}. \quad (4.15)$$

4.2.2 Arc Length Computation

Considering the constraint (4.15), the maximum allowable arc length of each elementary piece is ($i = 1, 2, \dots, N$):

$$l_{\max,i} = \frac{1}{\sigma_\varepsilon} \left(\sqrt{\kappa_0^2 + 2\theta_{\max}\sigma_\varepsilon} - \kappa_0 \right), \quad (4.16)$$

where $\theta_{\max} \leq \frac{\pi}{2}$ is the maximum winding angle allowed, which is used to control the upper bound of the curvature error. For simplicity, θ_{\max} can be set to $\frac{\pi}{2}$, however, under certain circumstances it can be reduced to a smaller value if ϵ_κ is beyond an acceptable tolerance. In order to compute $\mathbf{F}_c(s)$, arc lengths of all the elementary segment l_i ($i = 1, 2, \dots$) should be determined with $l_i \in (0, l_{\max,i}]$. Typically, to approximate the basic clothoid of constant length, the maximum allowable value can be chosen as the arc length of each segment $l_i = l_{\max,i}$ to reduce the number of elementary pieces. On the other hand, splitting a curve into shorter segments often produces better approximation accuracy, thus in practice l_i should be selected balancing computation efficiency with accuracy. In the case of $l_i = l_{\max,i}$, by combining (4.13), (4.14) and (4.16), the threshold of l_i is calculated as

$$l_\tau = \sqrt{\frac{2}{\sigma_\varepsilon}} (\theta_{\max}^2 - \sin^2(\theta_{\max}))^{\frac{1}{4}}. \quad (4.17)$$

As a result, analysis of the divergence problem caused by k is equivalent to evaluating the relationship between l_i and l_τ : if they are close to each other, the value of l_i needs to be changed accordingly to satisfy G^3 continuity. Without loss of generality, by setting $l_i \in [0, l_{\max,i}]$, we obtain $\theta_{0,i} = \frac{1}{2}\sigma_\varepsilon(l_i^\Sigma)^2$ and $\kappa_{0,i} = \sigma_\varepsilon l_i^\Sigma$, where

$$l_i^\Sigma = \begin{cases} 0, & \text{if } i = 1, \\ \sum_{j=1}^{i-1} l_j, & \text{otherwise.} \end{cases} \quad (4.18)$$

Consequently, the i -th elementary segment $\mathbf{F}_{\varepsilon,i}(s)$ starting at point $(x_{0,i}, y_{0,i})^T$ can be derived by applying $\kappa_0 = \kappa_{0,i}$ in (4.2) and then transforming the acquired $\mathbf{F}_\varepsilon(s)$:

$$\mathbf{F}_{\varepsilon,i}(s) = \begin{pmatrix} x_{0,i} \\ y_{0,i} \end{pmatrix} + \mathbf{R}(\theta_{0,i}) \mathbf{F}_\varepsilon(s), \quad s \in [0, l_i], \quad (4.19)$$

where

$$\begin{pmatrix} x_{0,i} \\ y_{0,i} \end{pmatrix} = \begin{cases} (0, 0)^T, & \text{if } i = 1, \\ \mathbf{F}_{\varepsilon,i-1}(l_{i-1}), & \text{otherwise.} \end{cases} \quad (4.20)$$

The basic curve $\mathbf{F}_\varepsilon(s)$ can be therefore represented with a series of elementary pieces $\mathbf{F}_{\varepsilon,i}(s)$ ($i = 1, 2, \dots, N$), and each individual segment can be approximated with corresponding quintic Bézier curve $\mathbf{B}_{\varepsilon,i}(t) = (x_{0,i}, y_{0,i})^T + \mathbf{R}(\theta_{0,i})\mathbf{B}_\varepsilon(t)$ ($i = 1, 2, \dots, N$) easily. More specifically, the curve $\mathbf{F}_\varepsilon(s)$ can be expressed as

$$\mathbf{F}_\varepsilon(s) = \sqrt{\frac{\pi}{\sigma_\varepsilon}} \begin{pmatrix} C\left(\sqrt{\frac{\sigma_\varepsilon}{\pi}}s\right) \\ S\left(\sqrt{\frac{\sigma_\varepsilon}{\pi}}s\right) \end{pmatrix}, \quad s \in [0, l_N^\Sigma]. \quad (4.21)$$

On these grounds, the approximation of a basic clothoid can be computed offline with any number of segments, although only a small number of segments are needed for practical use, especially when handling path design or smoothing problems. Algorithm 5 describes the procedure of approximating a basic clothoid with a successive generation of elementary curves by providing the sharpness σ_ε , number of segments N , and maximum winding angle $\theta_{\max} \in (0, \frac{\pi}{2}]$. The computed results can be stored in a lookup table for faster inquiries. The function $\text{CLOTHOID}(s, \kappa_0, \sigma, \theta_0, x_0, y_0)$ is computed according to (4.1), while the definition of $\text{BÉZIER}(t, l, \kappa_0, \sigma, \beta_1, \gamma_1, x_e, y_e)$ represents the final form of $\mathbf{B}_\varepsilon(t)$ (see Section A.2 in Appendix A). In each loop, calculating coordinates of two endpoints of $\mathbf{F}_{\varepsilon,i}(s)$

$(i = 1, 2, \dots, N)$ is necessary to formulate a quintic Bézier curve based on evaluations of Fresnel integrals. This process can be accomplished using numerical integration or any software with built-in functions supporting Fresnel integrals. From (4.16) and (4.17) we know that the obtained $(l_{\max,i})_{i=1}^N$ is a non-increasing sequence, thus l_τ is always less than $l_{\max,1}$ and greater than a certain value when N is great enough seeing $\lim_{n \rightarrow \infty} l_{\max,n} = 0$, i.e., $\min_{i \in \mathbb{Z}^+} l_{\max,i} < l_\tau < \max_{i \in \mathbb{Z}^+} l_{\max,i}$ for any $\theta_{\max} \in (0, \frac{\pi}{2}]$. Therefore the divergence position is certainly included in the domain where both β_2 and γ_2 have abrupt changes. Previously we have already obtained a proper range of l in the form of $(0, \frac{\xi_{1,k} k_d}{\sigma}] \cup [\frac{\xi_{0,k} k_d}{\sigma}, +\infty)$ for l by analyzing the characteristic of k . Again, in consideration of the divergence problem, a ratio test of $l_{\max,i}$ to l_τ is used to quantify the closeness between $l_{\max,i}$ and l_τ for convenience. Values of $l_{\max,i}$ are restricted to $(0, \xi_u l_\tau]$, which corresponds to the left part $(0, \frac{\xi_{1,k} k_d}{\sigma}]$. By this means l_τ is always larger than $l_{\max,i}$ and G^3 continuity can be achieved for any valid inputs.

Algorithm 5 Approximate Basic Clothoid $\mathbf{F}_c(s)$

```

function APPROXBASICCLO ( $\sigma_c, N, \theta_{\max}$ )
1:  $l_\tau \leftarrow \sqrt{2/\sigma_c}(\theta_{\max}^2 - \sin^2(\theta_{\max}))^{\frac{1}{4}}$      $\triangleright$  threshold of  $l_i$ 
2: for  $i = 1$  to  $N$  do                                 $\triangleright N$  elementary clothoid segments
3:   if  $i = 1$  then
4:      $l_i^\Sigma \leftarrow 0$ 
5:   else
6:      $l_i^\Sigma \leftarrow \sum_{j=1}^{i-1} l_{\max,j}$ 
7:   end if
8:    $\kappa_{0,i} \leftarrow \sigma_c l_i^\Sigma$ 
9:    $l_{\max,i} \leftarrow (\sqrt{\kappa_{0,i}^2 + 2\theta_{\max}\sigma_c} - \kappa_{0,i})/\sigma_c$ 
10:  if  $l_{\max,i} > \xi_u l_\tau$  then           $\triangleright$  avoid divergence
11:     $l_{\max,i} \leftarrow \xi_u l_\tau$ 
12:  end if
13:   $(x_{0,i}, y_{0,i})^T \leftarrow \text{CLOTHOID}(l_i^\Sigma, 0, \sigma_c, 0, 0, 0)$ 
14:   $(x_{b,i}, y_{b,i})^T \leftarrow \text{CLOTHOID}(l_{\max,i}, \kappa_{0,i}, \sigma_c, 0, 0, 0)$ 

```

```

15:    $\mathbf{B}_\varepsilon(t) \leftarrow \text{BÉZIER}(t, l_{\max,i}, \kappa_{0,i}, \sigma_\mathcal{L},$ 
            $l_{\max,i}, l_{\max,i}, x_{b,i}, y_{b,i})$ 
16:    $\mathbf{B}_{\varepsilon,i}(t) \leftarrow (x_{0,i}, y_{0,i})^T + \mathbf{R}(\sigma_\mathcal{L}(l_i^\Sigma)^2/2) \mathbf{B}_\varepsilon(t)$  ▷
        geometric transformation
17: end for
18: return  $(l_{\max,i})_{i=1}^N, (l_i^\Sigma)_{i=1}^N, (\mathbf{B}_{\varepsilon,i}(t))_{i=1}^N$ 

```

4.2.3 Bézier Approximation and Error Analysis

As an example, an approximation of basic clothoid curve composed of $N = 7$ elementary segments is shown in Fig. 4.4. A series of Bézier curves $\mathbf{B}_{\varepsilon,i}(t)$ are computed as approximations of elementary clothoid segments $\mathbf{F}_{\varepsilon,i}(s)$. After proper linear transformations, they are joined up to form the approximated basic clothoid following Algorithm 5. The curvature magnitude of the Bézier approximation is also displayed with a shaded area, and it can be seen that both the curvature and its derivative with regard to arc length are continuous, which confirms the G^3 property (4.5) between adjacent segments. By adjusting the arc length $l_{\max,i}$ of each segment, the errors are significantly reduced.

In fact, $\epsilon_\kappa(s)$ is a piecewise function defined by multiple sub-functions $\epsilon_{\kappa,i}(s)$ ($i \in \mathbb{Z}^+$), and each sub-function represents the curvature error between the elementary clothoid and its corresponding Bézier curve. For this reason, to analyze the upper bound of the error, we only need to evaluate the curvature error with regard to each pair of curves, i.e., $\mathbf{F}_{\varepsilon,i}(s)$ and $\mathbf{B}_{\varepsilon,i}(t)$. By applying G^3 conditions, both $\mathbf{F}_{\varepsilon,i}(s)$ and $\mathbf{B}_{\varepsilon,i}(t)$ can be alternatively represented with functions defined by a triplet $(\theta_{\max}, \sigma_\mathcal{L}, l_{\max,i})$. After arc length reparameterizations with regard to s , the obtained error function $\epsilon_\kappa(s)$ is only dependent on this triplet. Thus, there exists an infinite number of possible unique combinations since all these three parameters change continuously. Specifically, let us take the basic clothoid approximation with segment number $N = 100$ as a more practical example shown in Fig. 4.5. We start with considering the role of the first parameter θ_{\max} by setting the sharpness unchanged. As expected, longer Bézier curve can be formed with the growth of winding angle; however, the curvature error also becomes more significant. At the

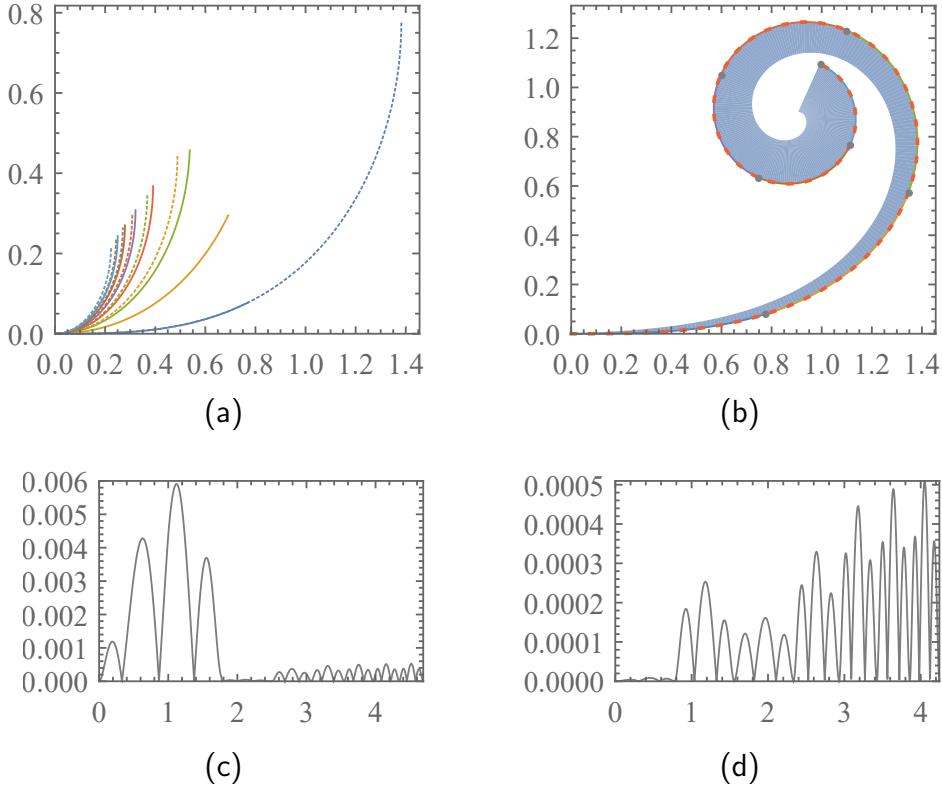


Figure 4.4: Basic clothoid approximation: (a) Individual Bézier curve $\mathbf{B}_{\varepsilon,i}$ ($i = 1, 2, \dots, 7$) are computed as primitives with setting $\theta_{\max} = \frac{\pi}{2}$, $N = 7$, and $\sigma_c = 1$. It shows both the original segments (dashed) and modified ones after adjusting arc length (solid) by setting $\xi_u = 0.5$. (b) The basic clothoid $\mathbf{F}_c(s)$ (dashed) and its Bézier approximation (solid) derived based on adjusted segments with the curvature magnitudes of the Bézier curves (shaded). (c) $\epsilon_\kappa(s)$ before adjusting segment length. (d) $\epsilon_\kappa(s)$ after adjusting segment length.

same time, it can be observed that the error function $\epsilon_\kappa(s)$ turns to be comparatively stable as the total arc length increases in all cases. More specifically, a relatively stable error range can be obtained by supplying a small enough value of $l_{\max,i}$. Similarly, to investigate the effect of sharpness σ_c , we fix the other two parameters in the triplet. Though the approximated curves have significantly different arc lengths and shapes, the error functions are roughly within the same range as shown in Fig. 4.5d.

In consideration of the general case, Fig. 4.6 gives the numerical results of curvature errors evaluated at each position $(\theta_{\max}, \sigma_c, l_{\max,i})$ within space $(0, \frac{\pi}{2}] \times$

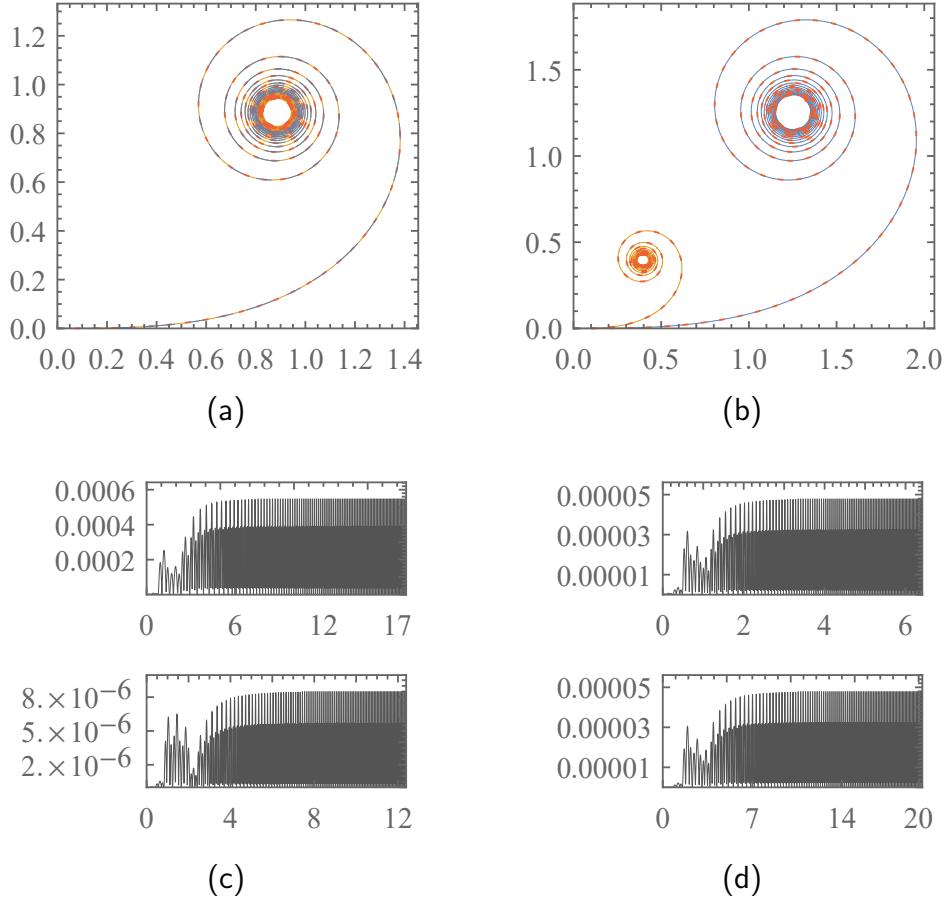


Figure 4.5: Bézier approximations (solid) of basic clothoids (red dashed) and curvature error functions: (a) $\theta_{\max} = \frac{\pi}{2}$ (blue) and $\theta_{\max} = \frac{\pi}{4}$ (orange) as $\sigma_c = 1$. (b) $\sigma_c = 0.5$ (blue) and $\sigma_c = 5$ (orange) as $\theta_{\max} = \frac{\pi}{3}$. The corresponding error function $\epsilon_\kappa(s)$ of each approximation: (c) Top: $\theta_{\max} = \frac{\pi}{2}, \sigma_c = 1$; Bottom: $\theta_{\max} = \frac{\pi}{4}, \sigma_c = 1$; (d) Top: $\sigma_c = 5, \theta_{\max} = \frac{\pi}{3}$; Bottom: $\sigma_c = 0.5, \theta_{\max} = \frac{\pi}{3}$.

$(0, 10] \times (0, l_\tau)$, where the upper bound of $l_{\max,i}$ is determined by (4.17). It can be seen that θ_{\max} plays the most important role and larger error values appear where θ_{\max} is close to $\frac{\pi}{2}$. From Fig. 4.5 we already know that error variations are much more stable as $l_{\max,i}$ decreases. Thus we can ignore the effect of arc length change and focus on θ_{\max} and σ_c by choosing a small value of $l_{\max,i} = 0.05$, as shown in Fig. 4.7 with 2250 groups of values evaluated within region $(0, \frac{\pi}{2}] \times (0, 10]$. The impact of one parameter can be examined by setting the other fixed as shown in Fig. 4.7b. It can be seen that ϵ_κ is almost only related to θ_{\max} and a smaller

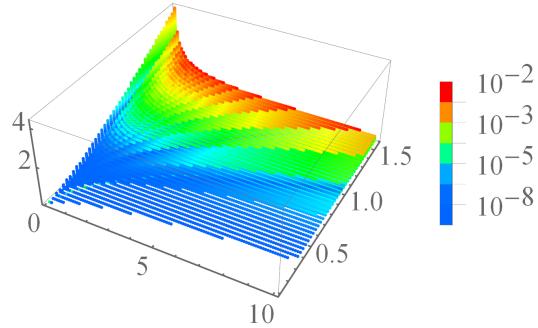


Figure 4.6: Curvature error as a function of $\theta_{\max} \in (0, \frac{\pi}{2}]$, $\sigma_c \in (0, 10]$ and $l_{\max,i} \in (0, l_\tau)$: all the values of $\epsilon_\kappa(\theta_{\max}, \sigma_c, l_{\max,i})$ are indicated by colors and within range $(0, 0.01)$.

winding angle provides better accuracy as expected. Also it demonstrates that the values of curvature error do not vary too much as σ_c changes. During creating a lookup table for a basic clothoid, this serves as a guidance that a smaller wingding angle θ_{\max} should be preferred when higher accuracy is required.

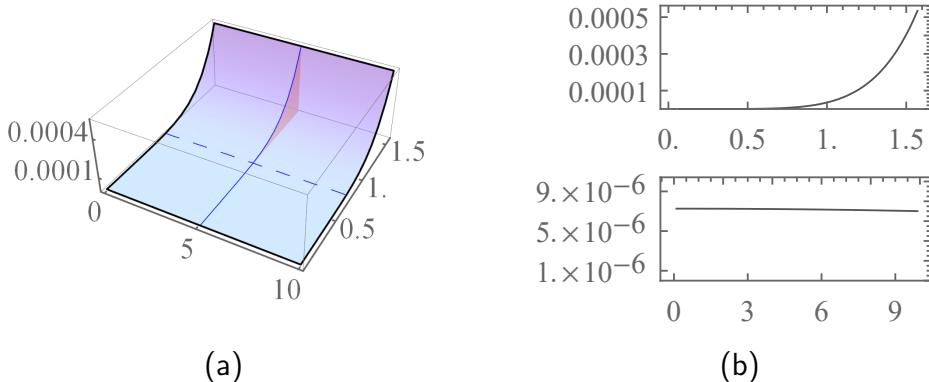


Figure 4.7: Curvature error with regard to θ_{\max} and σ_c : (a) $\epsilon_\kappa(\theta_{\max}, \sigma_c)$ as $\theta_{\max} \in (0, \frac{\pi}{2}]$, $\sigma_c \in (0, 10]$ and $l_{\max,i} = 0.05$. (b) Top: $\epsilon_\kappa(\theta_{\max})$ as $\sigma_c = 5.10$; Bottom: $\epsilon_\kappa(\sigma_c)$ as $\theta_{\max} = 0.77$.

4.3 General Clothoid Approximation

Revisiting the clothoid defined in (4.1), if both its sharpness and initial curvature are positive, the G^3 approximation can be made by reusing values of basic clothoid

$\mathbf{F}_\mathcal{L}(s)$ in (4.21) due to the relationship:

$$\mathbf{F}(l) = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \mathbf{R} \left(\theta_0 - \frac{\kappa_0^2}{2\sigma} \right) \sqrt{\frac{\sigma_\mathcal{L}}{\sigma}} (\mathbf{F}_\mathcal{L}(s_u) - \mathbf{F}_\mathcal{L}(s_0)), \quad (4.22)$$

where $s_0 = \frac{\kappa_0}{\sqrt{\sigma\sigma_\mathcal{L}}}$ and $s_u = \frac{\kappa_0 + \sigma l}{\sqrt{\sigma\sigma_\mathcal{L}}}$. In fact, a general clothoid curve with arbitrary sharpness and initial curvature can be approximated based on the basic clothoid curve approximation $(\mathbf{B}_{\varepsilon,i}(t))_{i=1}^N$ defined in the first quadrant, whose values are available in the existing lookup table. Algorithm 6 is implemented to locate the second endpoint of a basic clothoid with arc length l in the lookup table, and the returned index h and parameter t facilitate the process of approximating a general clothoid. Function ARCLENPAR is an implementation of the Hermite interpolation based algorithm mentioned earlier which takes a normalized arc length and a derivative of the given function as its input arguments. It generates a parameter $t \in [0, 1]$ at corresponding reference point with prescribed and normalized arc length, i.e., $l_n \in [0, 1]$.

Algorithm 6 Locate Endpoint in the Lookup Table

```

function LOCATELUT ( $\sigma_\mathcal{L}, N, \theta_{\max}$ )
    1:  $(l_{\max,i})_{i=1}^N, (l_i^\Sigma)_{i=1}^N, (\mathbf{B}_{\varepsilon,i}(t))_{i=1}^N \leftarrow \text{APPROXBASICCLO}(\sigma_\mathcal{L}, N)$ 
    2:  $i \leftarrow 1$ 
    3: while  $l_i^\Sigma \leq l$  do determine the index h in the lookup table
        4:  $i \leftarrow i + 1$ 
    5: end while
    6:  $h \leftarrow i - 1$ 
    7:  $l_n \leftarrow (l - l_h^\Sigma)/l_{\max,h}$ 
    8:  $t \leftarrow \text{ARCLENPAR}(l_n, \mathbf{B}'_{\varepsilon,h}(t))$  parameter of the last Bézier segment
    9: return  $h, t$ 

```

Let $\mathbf{B}(t)$ be the Bézier based approximation to the general clothoid (4.1), and the detailed process of constructing $\mathbf{B}(t)$ is illustrated in Algorithm 7. The strategy is first dividing the original curve into three different parts according to s_0, s_u , which are mapped to the existing lookup table subsequently. Then corresponding

approximations are computed under different conditions: the starting and ending segments are approximated by a fragment of transformed quintic Bézier curve $\mathbf{B}_{\varepsilon,i}(t)$ respectively while the intermediate part is acquired by joining the remaining transformed pieces. Depending on the values of sharpness σ , initial curvature κ_0 , and curve length l , parameters s_0 and s_u may appear in different quadrants. Thus corresponding transformation matrices are applied to the basic clothoid approximation to place resulted curves in correct quadrant. Finally, a group of polynomial functions established on the basis of standard quintic Bézier curves through linear transformations are obtained with their individual domains.

Algorithm 7 Approximate General Clothoid $\mathbf{F}(s)$

```

function APPROXGENCLO ( $l, \sigma, \kappa_0, \theta_0, (x_0, y_0)^T$ )
1:  $s_0, s_u \leftarrow \sqrt{|\sigma|/\sigma_L} \kappa_0 / \sigma, \sqrt{|\sigma|/\sigma_L} (\kappa_0 + \sigma l) / \sigma$ 
2:  $l_s, l_e \leftarrow |s_0|, |s_u|$ 
3:  $h_s, t_s \leftarrow \text{LOCATELUT} (l_s)$             $\triangleright$  locate the start point in LUT
4:  $h_e, t_e \leftarrow \text{LOCATELUT} (l_e)$             $\triangleright$  locate the end point in LUT
5:  $(l_{\max,i})_{i=1}^N, (l_i^\Sigma)_{i=1}^N, (\mathbf{B}_{\varepsilon,i}(t))_{i=1}^N \leftarrow \text{APPROXBASICCLO} (\sigma_L, N)$ 
6:  $\mathbf{B}_0 \leftarrow \mathbf{B}_{\varepsilon,l_s}(t_s)$ 
7:  $\mathbf{T} \leftarrow \mathbf{R}(\theta_0 - \kappa_0^2/(2\sigma)) \sqrt{\sigma_L/|\sigma|} \begin{pmatrix} 1 & 0 \\ 0 & \text{sgn}(\sigma) \end{pmatrix}$ 
8: if  $s_0 \geq 0, s_u \geq 0$  then            $\triangleright$  both in the 1st quadrant
9:   if  $h_e - h_s = 0$  then            $\triangleright$  two points are in the same segment
10:     $I = ([t_s, t_e])$ 
11:   else            $\triangleright$  composed of multiple segments
12:     $I = ([t_s, 1], [0, 1]_{\times(h_e-h_s-1)}, [0, t_e])$ 
13:   end if
14:   for  $h = h_s$  to  $h_e$  do
15:      $\mathbf{B}(t) \leftarrow (x_0, y_0)^T + \mathbf{T}(\mathbf{B}_{\varepsilon,h}(t) - \mathbf{B}_0)$ 
16:   end for
17: else if  $s_0 < 0, s_u < 0$  then            $\triangleright$  both in the 3rd quadrant
18:   if  $h_s - h_e = 0$  then
19:      $I = ([t_e, t_s])$ 
20:   else
```

```
21:    $I = ([0, t_s], [0, 1]_{\times(h_s-h_e-1)}, [t_e, 1])$ 
22:   end if
23:   for  $h = h_s$  to  $h_e$  do
24:      $\mathbf{B}(t) \leftarrow (x_0, y_0)^T - \mathbf{T}(\mathbf{B}_{\varepsilon,h}(t) - \mathbf{B}_0)$ 
25:   end for
26:   else                                     > located in two different quadrants
27:      $I = ([0, t_s], [0, 1]_{\times(h_e+h_s-2)}, [0, t_e])$ 
28:     for  $h = h_s$  to 1 do
29:        $\mathbf{B}(t) \leftarrow (x_0, y_0)^T - \mathbf{T}(\mathbf{B}_{\varepsilon,h}(t) - \mathbf{B}_0)$ 
30:     end for
31:     for  $h = 1$  to  $h_e$  do
32:        $\mathbf{B}(t) \leftarrow (x_0, y_0)^T + \mathbf{T}(\mathbf{B}_{\varepsilon,h}(t) + \mathbf{B}_0)$ 
33:     end for
34:   end if
35:   return  $I, \mathbf{B}(t)$ 
```

4.4 Results and Error Analysis

4.4.1 Offline Approximation Algorithms

We compare our newly proposed piecewise G^3 approximation method with existing algorithms including C^2 Hermite approximation via s-power series [58], G^3 Bézier approximation with numerical search in [61] and [113], and its improved variant “ G^{2+} ” deterministic approximation using linear interpolations [62]. For the sake of fairness, the functions used by all of these methods are set to be 5th order polynomials. The first comparison is made on a basic pair of Fresnel integrals $\mathbf{F}(s) = (C_f(s), S_f(s))^T$, which is equivalent to (4.1) by setting $\sigma = \pi, \kappa_0 = 0, \theta_0 = 0$ and $(x_0, y_0) = (0, 0)$. The approximation results using different approaches are shown in Fig. 4.8. The curvature error functions of corresponding approximations are demonstrated in Fig. 4.9. The C^2 method gives a mediocre result and has an acceptable error bound in this case. Benefiting from the numerical optimization procedure, it can be seen that the result of G^3 approximation is quite close to

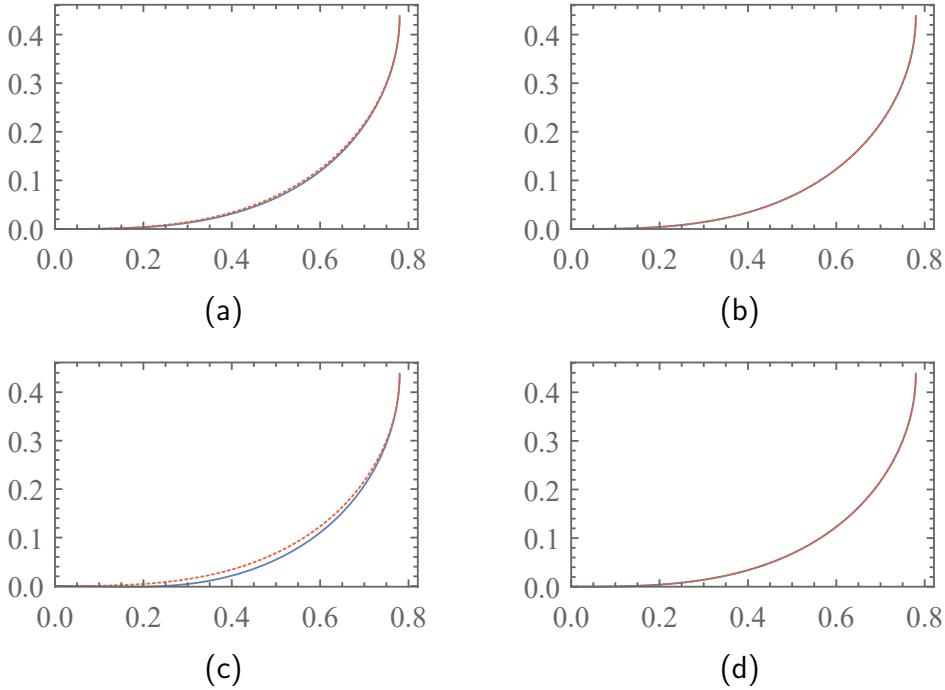


Figure 4.8: Different approximations (blue solid) of $\mathbf{F}(s) = (C_f(s), S_f(s))^T$ (red dashed) of unit length: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.

the original curve and the curvature error is satisfactory. For the “ G^{2+} ” method, since the corresponding k value falls within range $(-2 k_d, 2 k_d)$, adoption of linear interpolations free of complicated numerical computations is able to avoid the divergence problem. It however potentially increases curvature errors as a trade-off and gives the worst approximation with only G^2 continuity ensured. Among all the results, the result based on the proposed method has the smallest curvature error and additionally, no numerical search procedure is involved. Moreover, it guarantees G^3 continuity over the entire domain. Here the basic clothoid used has specifications $\sigma_c = 1$ and $\theta_{\max} = \frac{\pi}{2}$, and the curvature error can be reduced even further by specifying a smaller θ_{\max} as mentioned before.

Meanwhile, most available methods such as C^2 , G^3 , and G^{2+} approaches heavily rely on the normalization procedure during which the arc length of the clothoid is scaled. With the approximation computed based on the normalized curve, the final result is then obtained by rescaling the approximation curve back according

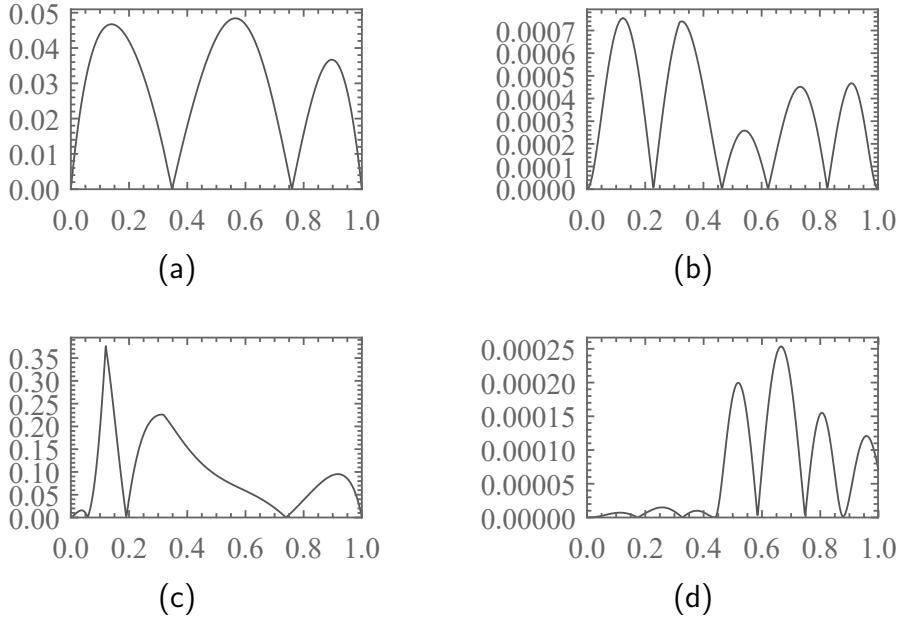


Figure 4.9: $\epsilon_\kappa(s)$ of unit length clothoid approximations: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.

to the same scale factor. Instead of scaling the clothoid, the proposed method is able to compute the approximation curve directly based on the lookup table. As an example of non-unit length clothoid approximation, Fig. 4.10 compares the approximation results of these methods with the proposed approach by setting $(x_0, y_0) = (0, 0)$, $\kappa_0 = 1$, $\theta_0 = 0$ and $\sigma = 1.8$. It can be observed that both C^2 and “ G^{2+} ” approximations have large deflections from the target clothoid curve though continuity constraints are satisfied at two ends. G^3 approximation curve has a better behavior but deflections still can be identified easily. Only the proposed method gives close approximation under this situation. In the sense of preserving curvatures, the proposed method also has the best performance as shown in Fig. 4.11 while other approximations lead to unacceptable results, i.e., $\epsilon_\kappa > 0.05$. In fact, the abnormal behaviors of the unsatisfactory results stem from the violating constraints that the winding angle of the scaled curve has to be limited within $[0, \frac{\pi}{2}]$ in those algorithms. As a consequence, only a small fraction of clothoid curves in (4.1) satisfy this criteria after scaling, thus these methods are not always ap-

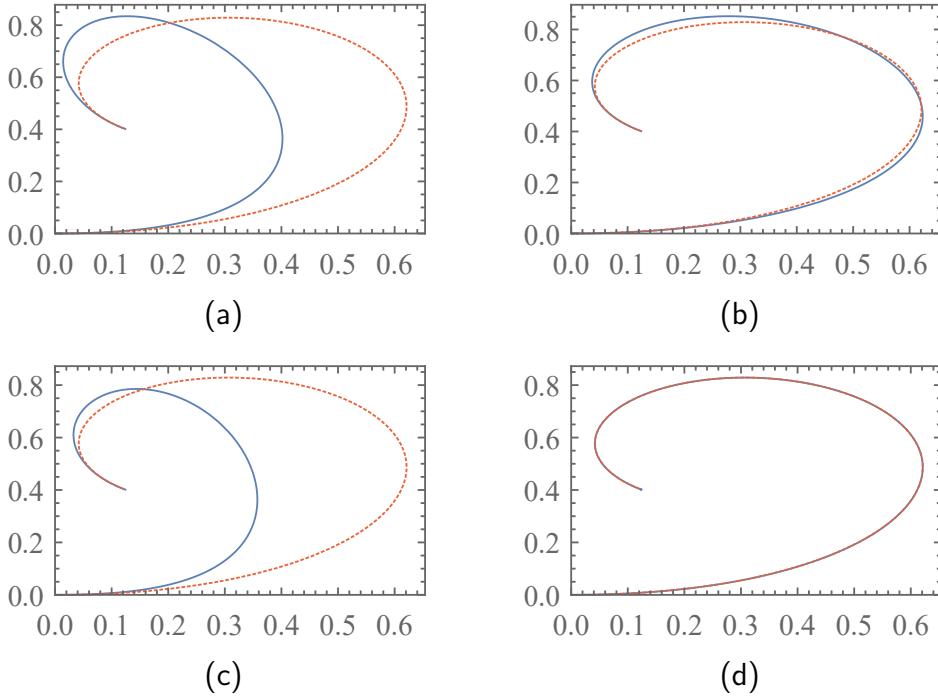


Figure 4.10: Different approximations (blue solid) of $\mathbf{F}(s)$ (red dashed) with $s \in [0, 2]$: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.

plicable. It can be verified that the results of C^2 , “ G^{2+} ”, and G^3 approximations becomes even worse as the growth of arc length of approximated curve.

Apart from the previously mentioned problems, some methods have inherent disadvantages. For example, the G^3 method cannot handle the divergence problem properly thus the approximation curve does not exist if the clothoid curve meets certain criteria, e.g., $k = -k_d$. For longer curves with large curvature values, their winding angles are not able to meet the requirements after normalization. Thus these methods become impractical and cannot handle arbitrary case. Fig. 4.12 further demonstrates the flexibility and generality of our proposed method by giving two nontrivial examples of non-unit length clothoid curves with positive and negative sharpness values respectively, which are difficult to approximate within satisfactory error bounds using other methods.

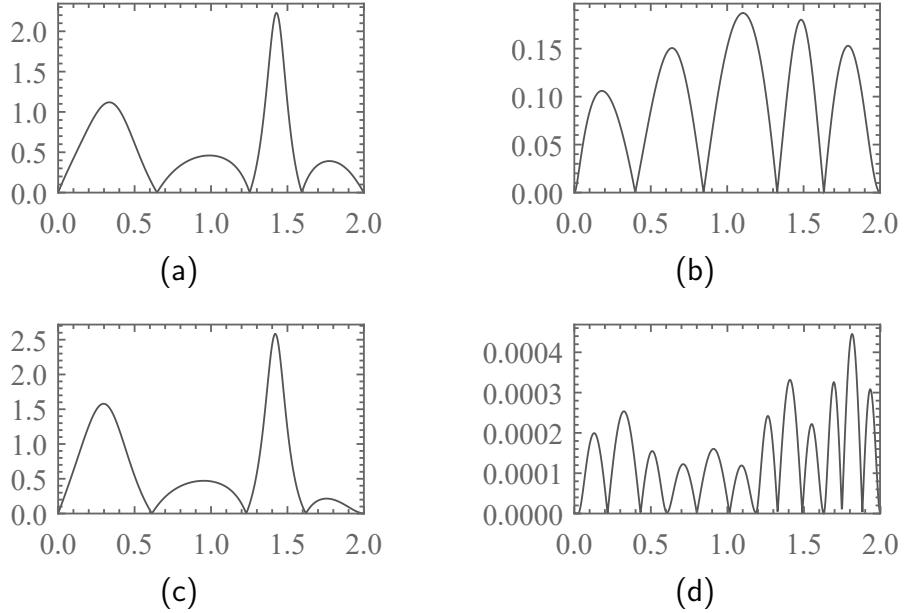


Figure 4.11: $\epsilon_\kappa(s)$ of non-unit length clothoid approximations: (a) C^2 Hermite approximation. (b) G^3 approximation with numerical search. (c) “ G^{2+} ” approximation. (d) Proposed G^3 approach.

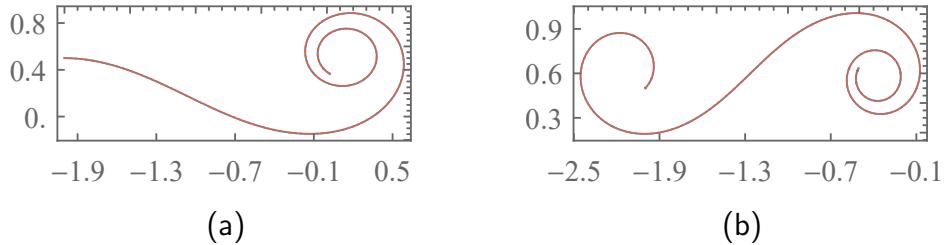


Figure 4.12: Proposed G^3 approximations (blue solid) of $\mathbf{F}(s)$ (red dashed) with prescribed parameters: (a) $\sigma = 1.0$, $l = 6.0$, $\kappa_0 = -1.0$, $\theta_0 = 0.0$, $(x_0, y_0) = (-2.0, 0.5)$. (b) $\sigma = -2.0$, $l = 6.0$, $\kappa_0 = 5.0$, $\theta_0 = \frac{\pi}{4}$, $(x_0, y_0) = (-2.0, 0.5)$.

4.4.2 Online Approximation

In addition to the curvature error (4.10), we evaluate the approximation accuracy with the position and orientation errors, both of which are defined based on Euclidean metric as in [1]:

$$\begin{aligned}\epsilon_p &= \|\mathbf{B}(s) - \mathbf{F}_c(s)\|, \\ \epsilon_\theta &= \|\theta_b(s) - \theta_c(s)\|.\end{aligned}\tag{4.23}$$

Here $\mathbf{B}(s)$ is the reparameterization of $\mathbf{B}(u)$ with respect to arc length while $\theta_b(s)$ and $\theta_c(s)$ represent the tangent angle of the approximated curve and the basic clothoid respectively.

We compared our algorithm with the circular arc based approximation [1] and the RBC algorithm [64], which are used for path computation of mobile robots. The accuracy of these approaches are evaluated with respect to the errors of position, orientation and curvature. The source curve is a basic clothoid with unit sharpness and arc length $s = 6.0$, which is taken from [1]. The sampling interval of the circular arc interpolation is $\Delta s_c = 0.2$ while an 11-th order rational Bézier curve is used for the RBC method as in [1]. Because the RBC algorithm is particularly sensitive to the winding angle, we limit the RBC approximation to be within $[0, \frac{\pi}{2}]$ in the comparisons. Since our Bézier approximation based approach supports uniform and nonuniform sampling, and in order to be consistent, both the uniform sampling implementation and the circular interpolation have $N = 30$ intervals of the same step length $\Delta s_c = 0.2$. The nonuniform sampling version allows a maximum winding angle $\theta_{\max} = 0.35\pi$ with only $N = 19$ segments stored in the lookup table for the same curve approximation. The position error ϵ_p computed with different approaches is shown in Fig. 4.13. It shows that an equivalent or even higher accuracy can be achieved with a smaller lookup table when nonuniform sampling is used. In practice, the lookup table can be built with an appropriate winding angle, given the fact that accuracy can be further improved by adopting a smaller value of θ_{\max} but more computation time and larger is required. In addition, to meet the specific needs of some path planning problems, the orientation error ϵ_θ and the curvature error ϵ_κ are evaluated and illustrated in Figs. 4.14 and 4.15 respectively. From the basic clothoid approximation we can clearly see that the proposed approach has significant advantage over the circular interpolation and the RBC algorithm with respect to approximation accuracy. It can also be found that at the end of each interval, the gaps generated by circular interpolation contribute to the discontinuity of coordinates, which is also reflected in its position error profile. It can also be found that the sudden changes in the coordinates deteriorate the approximation accuracy of orientation and curvature as well. Meanwhile, the

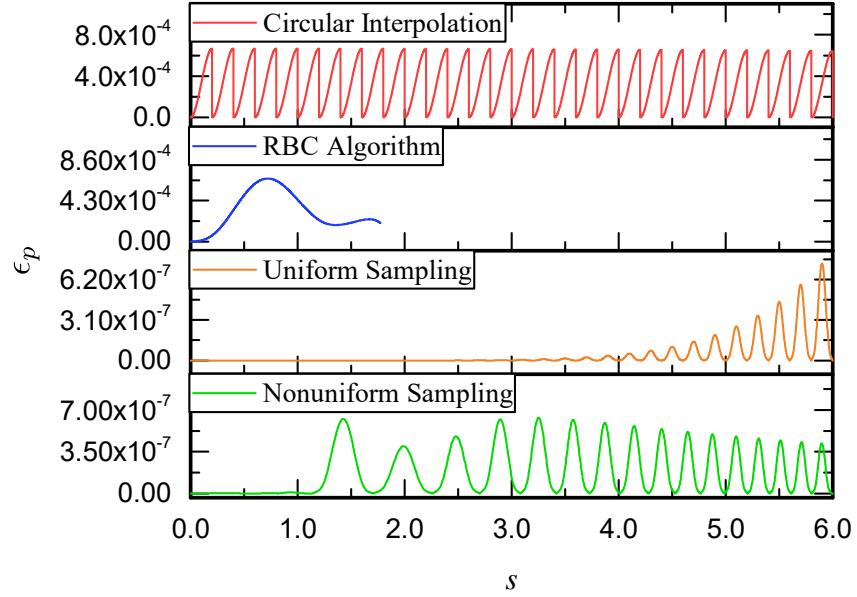


Figure 4.13: Position error ϵ_p of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.

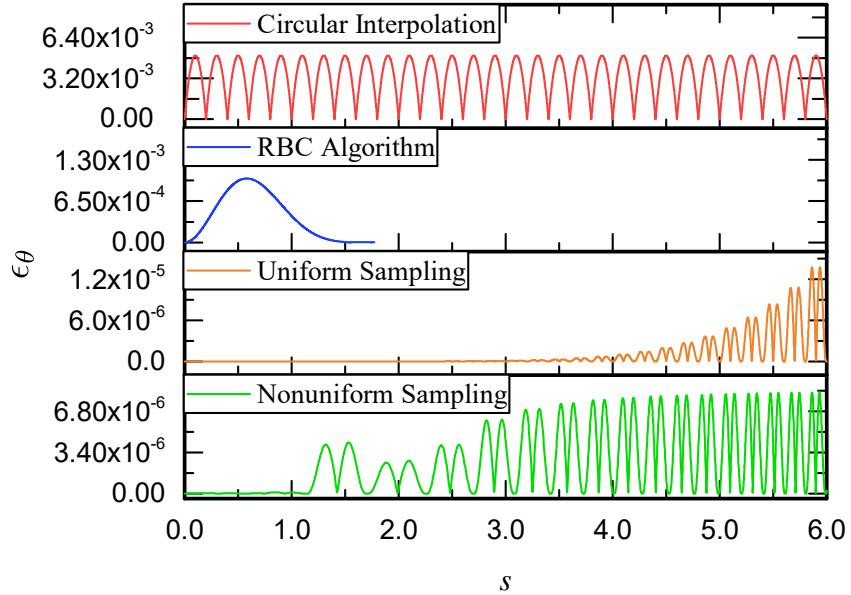


Figure 4.14: Orientation error ϵ_θ of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.

accuracy of the RBC algorithm decreases noticeably once the winding angle exceeds the limit, i.e., the arc length of the basic clothoid cannot be greater than $\sqrt{\pi}$.

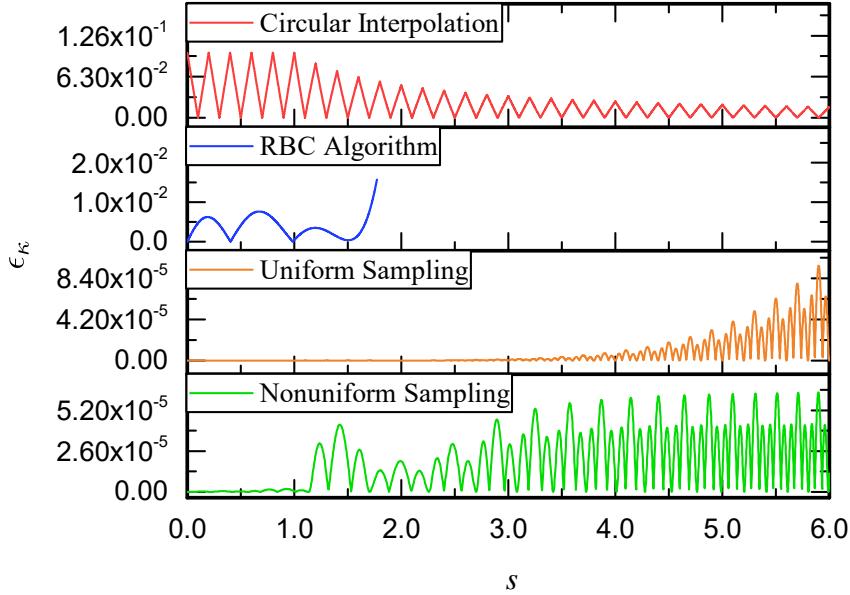


Figure 4.15: Curvature error ϵ_κ of different algorithms in [1], [2], and proposed approach with uniform and nonuniform sampling.

Also, we evaluate the feasibility of above approaches by considering the clothoids used in the robot planning applications with a typical constraint of the maximum orientation change $\Delta\Theta_{\max} = \frac{\pi}{2}$ and curve length $L_{\max} = 5$ m, as illustrated in [1]. We compared the position errors with respect to the scaling factor $C = |\sigma|^{-\frac{1}{2}}$ which changes from 10^{-5} to 10^5 , as shown in Fig. 4.16. Nonuniform sampling is used and two different values (0.05π and 0.002π) of the maximum winding angle θ_{\max} are tested for each segment $\mathbf{F}_{\mathcal{L},i}(s)$. The position error of the proposed approach can be much smaller than other methods with a relatively small sampling interval, e.g., $\theta_{\max} = 0.002\pi$. As expected, the circular interpolation with a lookup table of totally 626 points provides greater accuracy than the RBC algorithm as well as the simple linear interpolation in most scenarios. When the scaling factor becomes infinity, the clothoid degenerates to a straight line. Therefore, for considerably large values of scaling factor, the linear interpolation approach has obvious advantages.

Let $\epsilon_p(s)$ be the position error of $\mathbf{F}(s)$ and $\mathbf{B}(u)$ while $\epsilon_{p,\mathcal{L}}(s_{\mathcal{L}})$ be the position error of $\mathbf{F}_{\mathcal{L}}(s)$ and its Bézier approximation $\mathbf{B}_{\mathcal{L}}(u)$. Seeing the relationship (4.22)

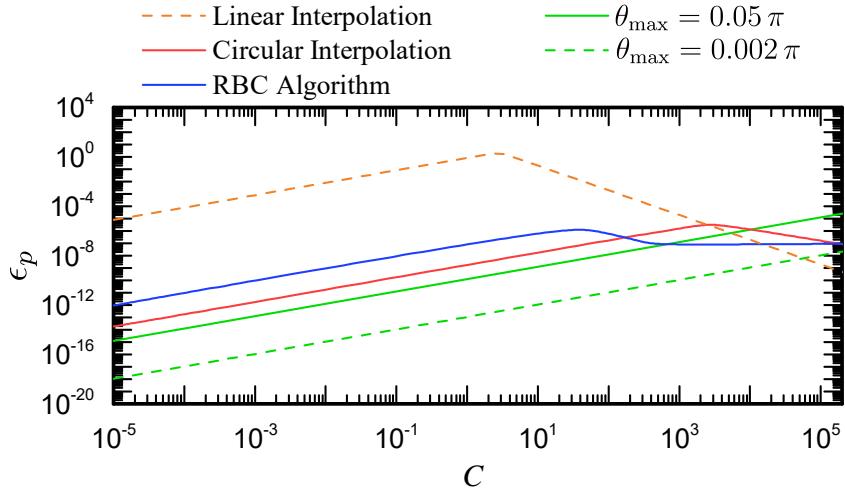


Figure 4.16: Position error ϵ_p with respect to clothoid scaling.

between a general and basic clothoid, the following relation should hold:

$$\epsilon_p(s) = \sqrt{\frac{|\sigma|}{\sigma_c}} \epsilon_{p,c}(s_c), \quad (4.24)$$

where $s = \sqrt{\frac{\sigma_c}{|\sigma|}} s_c$. Thus when $\sigma_c = 1$, we can estimate the position error of a general clothoid approximation by referring to the error $\epsilon_{p,c}(s_c)$ of the corresponding basic clothoid approximation and then multiplying it by the scaling factor C . This is also indicated in Fig 4.16, which shows a linear relationship with an approximate slope of 1 between logarithmic values of ϵ_p and C across the entire range.

Regarding the computation efficiency, we take the same example to test the three algorithms, all of which are implemented in Julia programming language [114] and run on a 3.60 GHz PC. The circular interpolation is implemented with a lookup table of totally 626 points while 11-th order RBC is used for comparison. The same setting $\theta_{\max} = 0.05\pi$ with nonuniform sampling is used for the proposed algorithm. To retrieve the clothoid path, a bunch of coordinates has to be evaluated. We compute 100 points on different source curves with distinct scaling factor values. A mean execution time is obtained to indicate the performance of each algorithm. The circular interpolation method is the most efficient one which takes an average of $199.5\ \mu s$, while the execution time of our algorithm is $305\ \mu s$, which is slightly slower than the circular interpolation. This can be acceptable considering the

benefit of higher accuracy as shown in Fig. 4.16. The RBC algorithm is much more time-consuming compared with the other two with an average execution time of $1477.5 \mu s$, which, however, is still sufficient for most real time applications.

Chapter 5

Clothoid Spline RRT Path Planner Based on Bézier Approximation

In Section 3.1 and 3.1.4 in Chapter 3, we are able to develop the kinematic models (3.40) and (3.41) corresponding to RWD and FWD systems respectively based on simplified bicycle models. Though these two kinematic models are derived based on different reference points on the vehicle body, they have mathematically identical expressions and similar constraints of maximum curvature and angular acceleration:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{pmatrix} = \begin{pmatrix} \cos \theta \\ \sin \theta \\ \kappa \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \sigma, \quad (5.1)$$

which is proposed in [20] and can be regarded as an extension of Dubins car model [19]. Here (x, y) are the coordinates of the reference point, and θ is the orientation. For a RWD model (3.40), this point corresponds to the rear wheel Q , and for a FWD model (3.41) it corresponds to the front wheel P , respectively, as illustrated in Fig. 3.6. For each case, the corresponding kinematic model must satisfy the constraints (3.42).

Considering a vehicle with upper-bounded steering angle $|\phi| \leq \phi_{\max} \in (0, \frac{\pi}{2})$ and steering rate $|\dot{\phi}| \leq \dot{\phi}_{\max}$, the corresponding parameters of a feasible path are restricted to be $|\kappa| \leq \kappa_{\max}$ and $|\sigma| \leq \sigma_{\max}$. Mathematically, by keeping the constant velocity assumption in [20] and maintaining a maximum angular acceleration

σ_{\max} , the curve of path traveled by the reference point is a piece of clothoid. As pointed out in [20] and [49], without considering the external restrictions such as the surrounding obstacles or space limitation, a locally optimal path satisfying condition (5.1) should have extreme value of the sharpness σ , namely, the maximum angular acceleration while the curvature is bounded by the steering capability of the vehicle:

$$\begin{cases} \sigma = \pm\sigma_{\max}, \\ |\kappa| \leq \kappa_{\max}, \end{cases} \quad (5.2)$$

where σ_{\max} and κ_{\max} can be computed according to (3.42). Thus in [22], a local composite path with continuous curvature property is formulated using straight lines, circular arcs, and clothoid arcs as primitives in [22].

Meanwhile, as a category of the most popular and general planning framework under nonholonomic constraints, sampling based planning methods, for example, PRM, RRT and their variations, prove to be very effective in various applications, especially in complicated scenarios where deterministic planning algorithms will probably fail or take an exceedingly long time [25]. For applications involving multiple queries, PRMs are always used as the global path planner while RRTs mainly apply to single-query problems. Despite such attractiveness of these global planners, the boundary value problem remains unsolved. Considering the kinematic characteristics in our problem, we can resort to geometric approaches to obtain a set of path primitives. Since spline based local planner can efficiently connect two configurations without resorting to a boundary value problem solver [73], the clothoid primitives can potentially be used as an effective alternative to the extend function in Algorithm 4. Different from Bézier splines, computation of clothoid curves involves Fresnel integrals, which is time consuming and thus may render it unsuitable in real time applications. To address this issue, the quintic Bézier curve based approach discussed in Chapter 4 is integrated into the framework to approximate the clothoids with high accuracy.

5.1 Local Path Primitives

Considering a basic scenario involving three successive waypoints W_1 , W_2 , and W_3 , the problem is to find a smooth path with (1) curvature continuity (2) upper-

bounded curvature and sharpness. Based on [22], here we use an efficient steering method instead of seeking for the optimal path exactly for the sake of simplicity and practical feasibility. Let $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2$ be the unit vectors along $\overrightarrow{W_1W_2}$ and $\overrightarrow{W_2W_3}$ respectively. Also, to accommodate the sampling based planning framework, we put our focus on the deflection δ , i.e., the angle between $\hat{\mathbf{e}}_1$ and $\hat{\mathbf{e}}_2$, by assuming that the distances between these waypoints can be flexibly adjusted. A local path configuration has two variations according to the value of δ . It mainly consists of a pair of clothoids, and a circular arc is to be included as $|\delta| > \delta_{\min} = \kappa_{\max}^2 / \sigma_{\max}$, while no circular arc is involved as $|\delta| \leq \delta_{\min}$. In both cases line segments may be introduced as transitions to comply with different lengths of adjacent edges W_1W_2 and W_2W_3 . These two cases correspond to the local path configurations in CASE 2 in Section 3.2.2 and CASE 1 in Section 3.2.1 discussed before, but now the geometric shape of these local planners will be parameterized for use by a sampling based planner. To take full advantage of the steering capability of the robot as well as reducing the steering time, the sharpness of the clothoid pair is set to be $\sigma = \sigma_{\max}$ as indicated in (5.2). Let $s \geq 0$ be the arc length parameter, and a piece of clothoid of sharpness σ_{\max} starting at $(0, 0)$, with other initial parameters being $\theta_0 = 0, \kappa_0 = 0$, can be expressed as:

$$\mathbf{F}(s) = \begin{pmatrix} x_c(s) \\ y_c(s) \end{pmatrix} = \sqrt{\frac{\pi}{\sigma_{\max}}} \begin{pmatrix} C_f\left(\sqrt{\frac{\sigma_{\max}}{\pi}} s\right) \\ S_f\left(\sqrt{\frac{\sigma_{\max}}{\pi}} s\right) \end{pmatrix}, \quad (5.3)$$

where $C_f(s)$ and $S_f(s)$ are Fresnel integrals defined by (2.9). Based on the standard position and orientation of (5.3), all the clothoid components within each path primitive subject to constraints (5.2) can be obtained through appropriate geometric transformations. In the next section, we will derive the parametric representations of each component in the coordinate system defined by given waypoints W_i ($i = 1, 2, 3$).

5.1.1 Path Parameterization of CASE 1

As $|\delta| = \delta_{\min}$, the circular arc degenerates to a point and the clothoid pair meet at C_m where the maximum curvature occurs. To ensure curvature continuity of the

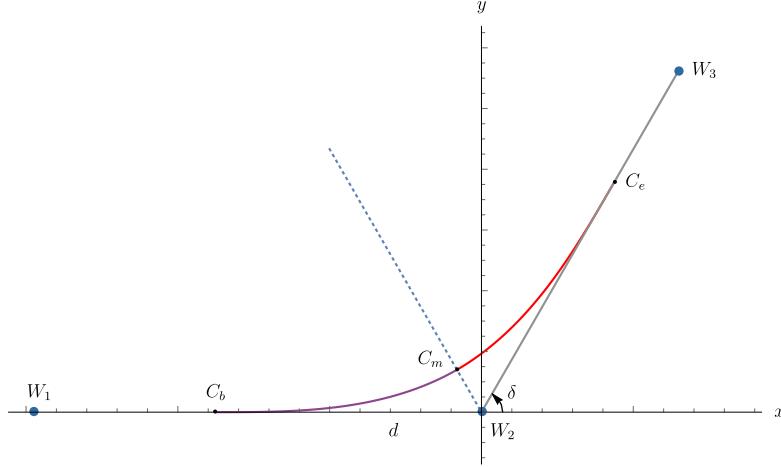


Figure 5.1: Parameterize path primitive: CASE 1.

local path subject to a smaller deflection, we join the clothoid pair of sharpness σ_{\max} before they reach the maximum curvature, as shown in Figure 5.1.

The local smoothing path formed by waypoints W_1 , W_2 and W_3 are composed of a pair of clothoids C_bC_m , C_eC_m and straight line segments W_1C_b , W_3C_e . For clothoid C_bC_m , its winding angle equals $|\delta|/2$ with an arc length of $\sqrt{|\delta|/\sigma_{\max}}$. Similarly, the distance from the known waypoint W_2 to the starting point C_b , denoted as d , can be obtained:

$$d = x_c \left(\sqrt{\frac{|\delta|}{\sigma_{\max}}} \right) + y_c \left(\sqrt{\frac{|\delta|}{\sigma_{\max}}} \right) \tan \left(\frac{|\delta|}{2} \right), \quad (5.4)$$

where x_c , y_c are the coordinates of a clothoid at standard posture given by (5.3). With all such information, it is straightforward to express clothoid C_bC_m as $(x_{c1}(s), y_{c1}(s))^T$:

$$\begin{pmatrix} x_{c1}(s) \\ y_{c1}(s) \end{pmatrix} = \mathbf{R}(\delta_0) \begin{pmatrix} x_c(s) \\ \text{sgn}(\delta)y_c(s) \end{pmatrix} - d \hat{\mathbf{e}}_1 + W_2, \quad (5.5)$$

where δ_0 is the angle from x -axis to $\hat{\mathbf{e}}_1$ since generally $\hat{\mathbf{e}}_1$ does not coincide with x -axis, and $\mathbf{R}(\delta_0)$ is the planar rotation matrix for a counterclockwise rotation by δ_0 .

Consequently, its counterpart C_eC_m , denoted as $(x_{c2}(s), y_{c2}(s))^T$, can be derived as a reflection across axis W_2C_m :

$$\begin{pmatrix} x_{c2}(s) \\ y_{c2}(s) \end{pmatrix} = \mathbf{R}(\delta_0 + \delta) \begin{pmatrix} -x_c(s) \\ \text{sgn}(\delta)y_c(s) \end{pmatrix} + d\mathbf{R}(\delta)\hat{\mathbf{e}}_1 + W_2. \quad (5.6)$$

The obtained local path is G^2 continuous since each piece has its tangent perpendicular to the bisector W_2C_m with the same curvature value $\kappa = \sqrt{|\delta|\sigma_{\max}}$ at the intersection point C_m . Also it can be easily verified that the curvature value does not exceed κ_{\max} considering $|\delta| \leq \delta_{\min}$.

5.1.2 Path Parameterization of CASE 2

From constraints (5.2), the clothoid pair C_bA_b , C_eA_e are expected to end at $s = \kappa_{\max}/\sigma_{\max}$. Considering the first piece C_bA_b with the winding angle θ increasing from 0 to $\kappa_{\max}^2/(2\sigma_{\max})$ at the ending point A_b , it is connected to its mirror counterpart C_eA_e through a piece of circular arc A_bA_e , the center of which is denoted as O . The distance from O to either edge (W_1W_2 or W_2W_3) can be obtained as

$$h = y_c \left(\frac{\kappa_{\max}}{\sigma_{\max}} \right) + \frac{1}{\kappa_{\max}} \cos \left(\frac{\kappa_{\max}^2}{2\sigma_{\max}} \right). \quad (5.7)$$

Thus the configuration of the desired path, shown in Figure 5.2, is composed of a pair of clothoids, a circular arc and line segments. To explicitly locate the local path with regard to given waypoints, the distance between C_b and W_2 is required, which is denoted as d and can be obtained according to the geometrical relationships:

$$\tilde{d} = x_c \left(\frac{\kappa_{\max}}{\sigma_{\max}} \right) + h \tan \left(\frac{|\delta|}{2} \right) - \frac{1}{\kappa_{\max}} \sin \left(\frac{\kappa_{\max}^2}{2\sigma_{\max}} \right). \quad (5.8)$$

Based on (5.5) and (5.6), we are able to get the expressions for both C_bA_b and C_eA_e by replacing d with \tilde{d} in (5.8).

Apart from the clothoid pair, the circular arc is also to be located, which requires the coordinates of O :

$$\begin{pmatrix} x_O \\ y_O \end{pmatrix} = \mathbf{R}(\delta_0) \begin{pmatrix} \tilde{d} - h \tan \left(\frac{|\delta|}{2} \right) \\ \text{sgn}(\delta)h \end{pmatrix} - \tilde{d}\hat{\mathbf{e}}_1 + W_2. \quad (5.9)$$

And the range of its central angle subtended by A_bA_e can be identified:

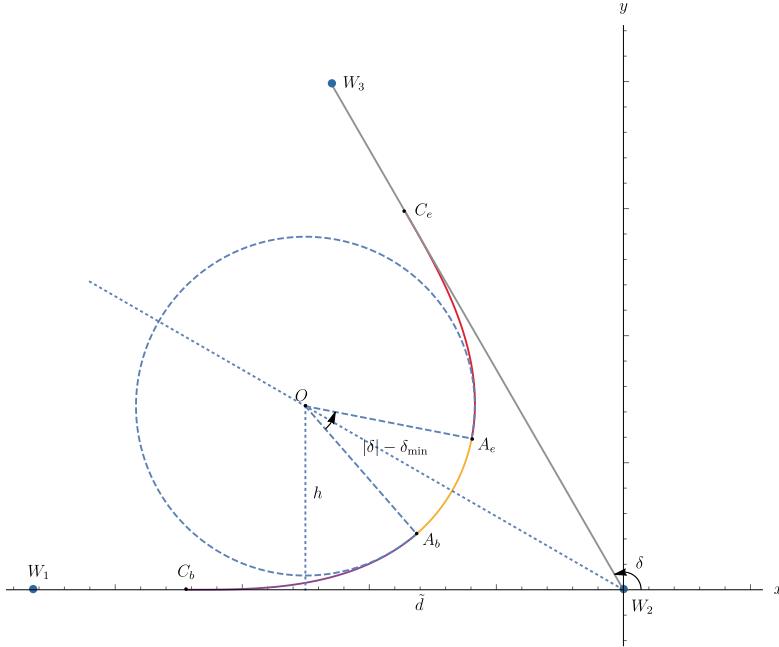


Figure 5.2: Parameterize path primitive: CASE 2.

$$\begin{cases} \left[\delta_0 + \frac{1}{2}(\delta_{\min} - \pi), \delta_0 + |\delta| - \frac{1}{2}(\delta_{\min} + \pi) \right], & \text{if } \delta > \delta_{\min} \\ \left[\delta_0 - |\delta| + \frac{1}{2}(\delta_{\min} + \pi), \delta_0 - \frac{1}{2}(\delta_{\min} - \pi) \right], & \text{if } \delta < -\delta_{\min} \end{cases} \quad (5.10)$$

Also, the G^2 continuity of the local path can be verified and the maximum curvature value κ_{\max} is reached, which is equivalent to the reciprocal of the minimum turning radius.

5.2 Clothoid Spline Based RRT

5.2.1 Reachability Tree

Apart from these two commonly used cases, other situations covered in Section 3.2.2 can also be parameterized accordingly and further tuned for integrating with sampling based planners. Considering that the path primitives in other cases have more complicated representations, the computational efficiency may be affected. Meanwhile, path primitives in both CASE 1 and CASE 2 are symmetrically arranged which facilitates the design of a feasible local planner. Moreover, a valid local planner should be dense that the space can be fully explored given a large enough

set of path primitives. In a native RRT framework Algorithm 3, the EXPANDRRT function uses a fixed length Δq to extend the tree structure. Analogously, here we can also set a fix step size for the local planner, and for the sake of convenience, we can restrict the edge length W_1W_2 and W_2W_3 to a constant Δs , which can also be regarded as the control polygon of the composite clothoid path.

Considering the kinematic constraints which restrict both the curvature and the curvature derivative, namely, sharpness, the capability of expanding of the robot is further impaired. Without considering obstacles, we define $R(q_0, \mathcal{U}) \subseteq \mathcal{X}$ as the reachable set from q_0 , where \mathcal{U} denotes the set of all permissible action trajectories given length of the control polygon $s \in [0, \infty]$. All the states in \mathcal{X} that can be visited by the robot is expressed by

$$R(q_0, \mathcal{U}) := \{q_1 \in \mathcal{X} \mid \exists \tilde{u} \in \mathcal{U} \wedge \exists s \in [0, \infty] \text{ such that } q(s) = q_1\}, \quad (5.11)$$

where $q(s)$ represents the states reached with respect to the edge length s of the control polygon and $q(0) = q_0$. Given a restricted edge length Δs , the reachable set can be presented with a reachability tree shown in Fig. 5.3.

5.2.2 Integration of Clothoid Spline and RRT

Since Δs is fixed, all the possible states q_1 can reach are on the same circular arc of radius Δs . The green lines are composed of clothoid curves while the blue ones are straight line segments. The minimum value of ΔS should be greater than a specific value ΔS_{\min} such that the curved path in the motion primitive should be complete. The length of straight lines can be shortened or extended flexibly according to prescribed parameters, however, no compromise can be made with respect to the curved components including the clothoid pair and circular arcs. Thus a minimally required length of the control polygon can cover all the curved components in a primitive, and consequently ΔS_{\min} should be twice the minimum length considering two discrete nodes possibly connected may share the same edge during tree evolution. From Fig. 5.3 we know that the edge length corresponding to the curved components increases monotonically with respect to the deflection

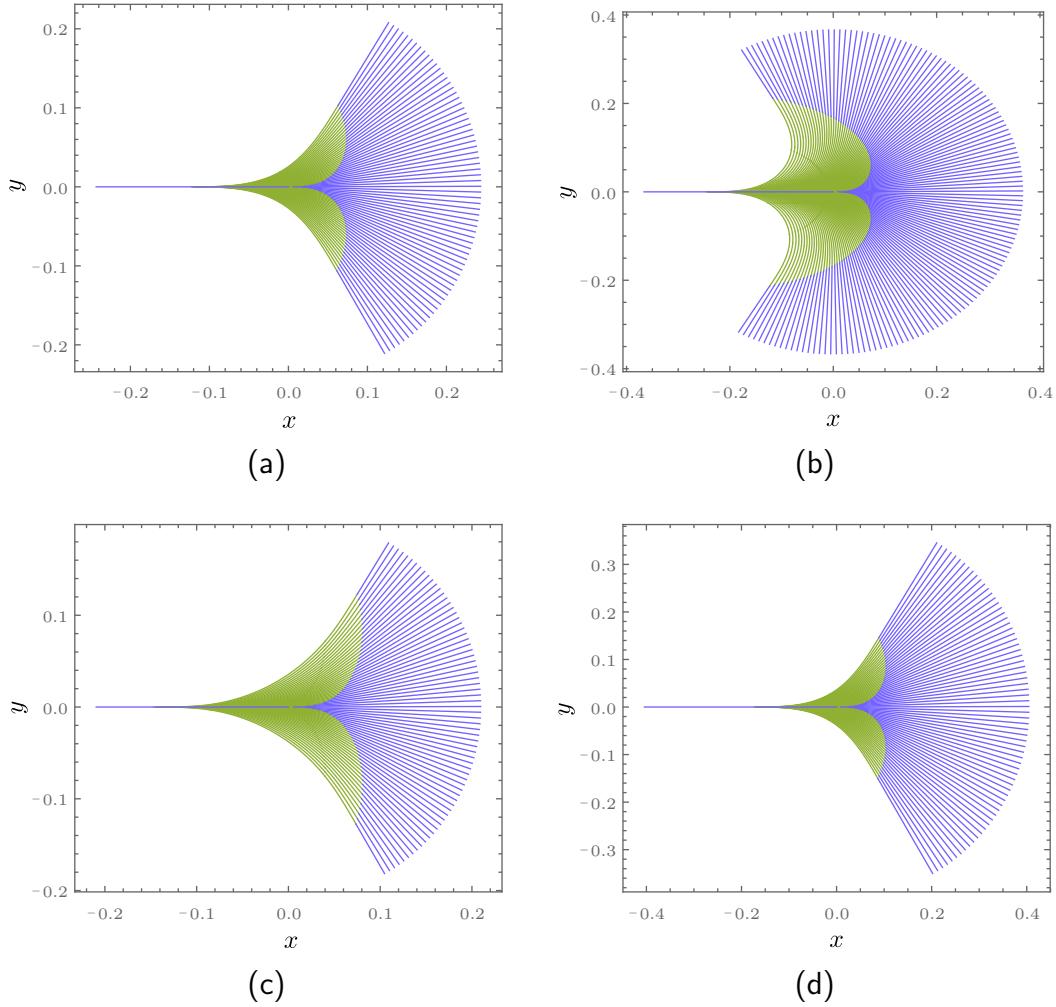


Figure 5.3: Reachability tree: (a) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{\pi}{3}]$. (b) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{2\pi}{3}]$. (c) $\kappa_{\max} = 5.0$, $\sigma_{\max} = 80.0$, $\delta \in [0, \frac{\pi}{3}]$. (d) $\kappa_{\max} = 10.0$, $\sigma_{\max} = 40.0$, $\delta \in [0, \frac{\pi}{3}]$.

angle. Considering the worst case occurs in CASE 2 where a turn with deflection angle being $\delta = \delta_{\max}$, we have

$$\Delta s_{\min} = 2x_c \left(\frac{\kappa_{\max}}{\sigma_{\max}} \right) + 2h \tan \left(\frac{|\delta_{\max}|}{2} \right) - \frac{2}{\kappa_{\max}} \sin \left(\frac{\kappa_{\max}^2}{2\sigma_{\max}} \right), \quad (5.12)$$

where δ_{\max} is specified by the user to control the shape of the reachability tree. Meanwhile, the step size cannot be too large, which will decrease the tree density in a given space. Analogous to the RRT algorithm (Algorithm 3), this problem can be solved by specifying a maximum edge length according to the dimension of

the environment, i.e., Δs_{\max} , which is a constant. The path planning procedure using clothoid based RRT approach is described in Algorithm 8.

Algorithm 8 Clothoid Spline Based RRT

```

function CLOTHOIDRRT ( $q_s, q_G, \epsilon$ )
    1:  $\mathcal{G}.\text{Init}(q_s)$ 
    2: while  $\rho(q_{\text{new}}, q_G) > \epsilon$  do
    3:    $q_{\text{rand}} \leftarrow \text{STATESAMPLE}()$ 
    4:    $q_{\text{near}} \leftarrow \text{SELECTNODE}(q_{\text{rand}}, \mathcal{G})$ 
    5:    $q_{\text{new}} \leftarrow \text{EXPANDNODE}(q_{\text{near}}, q_{\text{rand}}, \Delta q)$ 
    6: end while
    7: return  $\mathcal{G}$ 
```

Given the start and goal states q_s , q_G and a distance tolerance ϵ , the tree \mathcal{G} keeps expanding until a state q_{new} is close enough to the goal q_G . Based on known constraints κ_{\max} and σ_{\max} (5.2), a critical deflection angle denoted by δ_c is defined as $\delta_c = \frac{\kappa_{\max}^2}{\sigma_{\max}}$. According to the robot size and the environment, the start and goal states q_s , q_G are checked to ensure the planner is in a working state. Generally, the remaining procedure can be divided into three phases after initialization:

- (i) STATESAMPLE: A state q_{rand} is generated by uniformly sampling the space covered by the reachable set (Fig. 5.3), which is restricted by δ_{\max} . Note that we are using a lazy evaluation, which is different from the basic RRT algorithm. Here we do not check the collision condition of q_{rand} because we know that such points only serve as the vertices of the control polygon where the final resulting path does not pass through them. Only the collision information of the actual path is checked. The lazy collision checking is one of the advantages owned by spline based RRTs, which not only saves computational resource, but also avoids the false negative error. Also, to improve the convergence speed, a state in the goal region can be sampled as q_{rand} occasionally to bias the search towards the goal.
- (ii) SELECTNODE: A nearest q_{near} node in \mathcal{G} is selected to connect with q_{rand} , which uses a k -nearest neighbors algorithm with a GNAT (Geometric Near-neighbor Access Tree) data structure [115].

(iii) EXPANDNODE: Then k nearest neighbors are returned in sorted order, and the distance $\rho(q_{\text{near}}, q_{\text{RAND}})$ is computed and checked if it falls on the range $[\Delta s_{\min}, \Delta s_{\max}]$. If the edge is outside this interval, it is clamped by this space and q_{RAND} is replaced with a state obtained by interpolation ($\rho > \Delta s_{\max}$) or extrapolation ($\rho < \Delta s_{\min}$). Otherwise, q_{RAND} is kept unchanged. Then the new state q_{new} is connected and an edge is formed correspondingly, which is associated with a clothoid local planner. Such a planner can be classified according to the value of δ_c : The local planner is of CASE 1 if the deflection angle is smaller than δ_c . Otherwise it belongs to CASE 2. The new edge is validated accordingly. Again, lazy evaluation is applied, which only checks the curved components and only the straight line motion connected to the last state is examined. These straight line primitives associated with outgoing edges are ignored in current iteration. Once the motions are validated, q_{new} is added to the tree.

5.2.3 Path Planning in a Maze environment

Finding a feasible path in such an environment is very difficult because we need to consider not only the kinematic constraints of the vehicle that both the curvature and sharpness are up-bounded, but also the obstacles distributed in the environment. Taking use of the simplified bicycle model and we adopt the clothoid spline based RRT to incrementally search for a feasible geometric solution.

First, we test our algorithm in a 3D space without obstacles (Fig. 5.4). The states are sampled within the region bounded by the rectangular walls. In total, 20 states are randomly sampled in $SE(2)$ where 11 among them along with the start state consist of the solution path. In Fig. 5.5, the states sampled are in red, while the edges, spline curves, and the solution path are shown in green, yellow and blue, respectively. It can be seen that during each iteration, the sampling process only picks the states in an area constrained by the kinematic model of the vehicle. Actually, the valid sampling region is in the form of an annulus bounded by Δs_{\min} and Δs_{\max} , and the samples only fall on an area in front of the vehicle, which is constrained by the specified deflection angle.

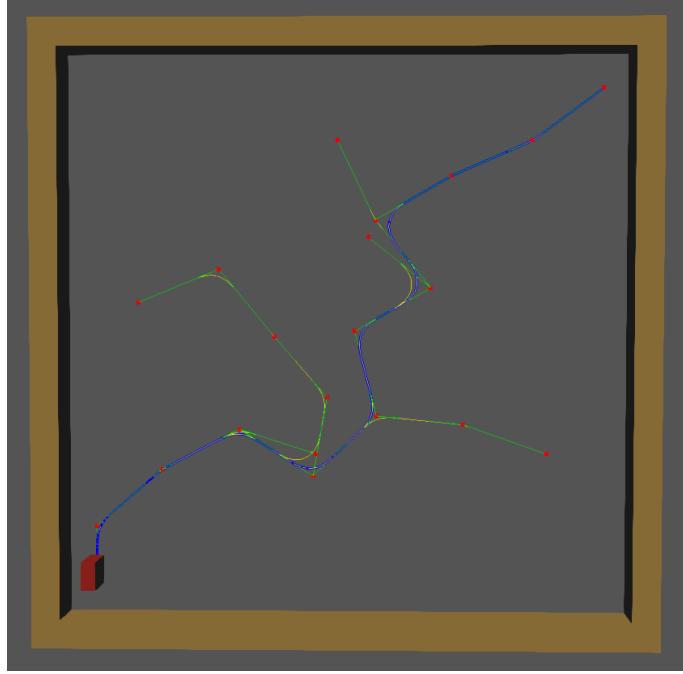


Figure 5.4: Clothoid spline based RRT in a free 3D space.

Fig. 5.5 is another test of our clothoid spline based RRT (Algorithm 8), which is constrained in a 3D environment in the form of a maze. The environmental model has the same boundary walls as in Fig. 5.4 except that more obstacles are involved. Similar to the free space example, the vehicle starts at the lower left corner with specified orientation, while the goal is at the upper right corner. Totally 1775 states are created by the RRT and it takes 0.577 s to obtain a feasible solution, which is composed of 23 states.

The solution path and its control polygon are shown together in Fig. 5.5b. It can be noticed that some edges in the tree collide with the walls and obstacles, which, however, does not affect the final results as shown in Fig. 5.5. These green edges are not actual paths which are in yellow, thus they are handled by the lazy evaluation mechanism mentioned before. The collision detection system represents the vehicle as a bounding volume instead of a single point robot, thus more computational resources are used. Lastly, it is worth mentioning that the clothoid paths are all approximated and displayed using the quintic Bézier curves, which are computed by querying a precomputed lookup table as described in Algorithm 7.

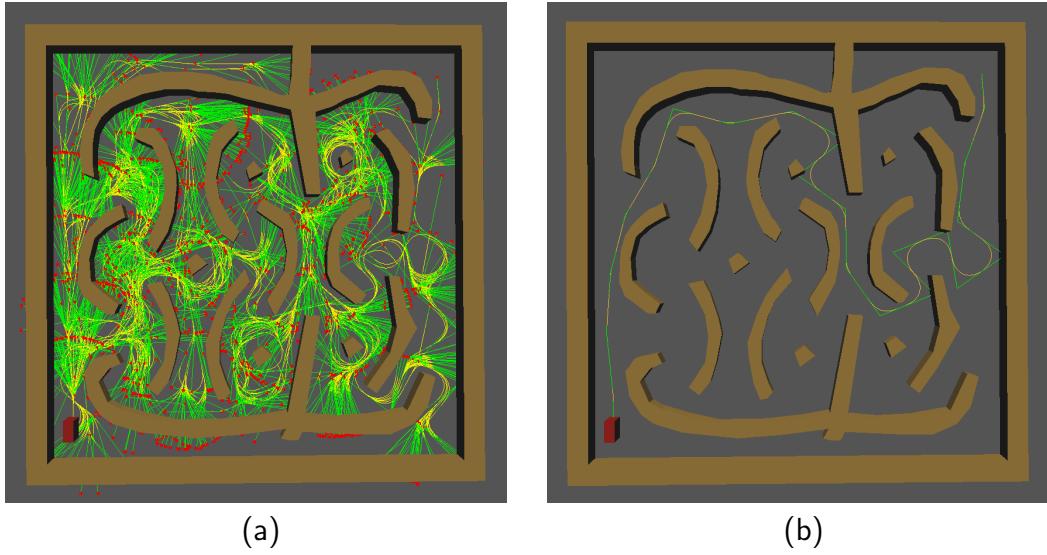


Figure 5.5: Clothoid spline based RRT in a cluttered 3D space (a) Tree expansion results. (b) Solution path.

As a comparison, we implemented a spline based RRT algorithm using cubic Bézier curves [73] is presented in Fig. 5.6. The Bézier spline based RRT can only

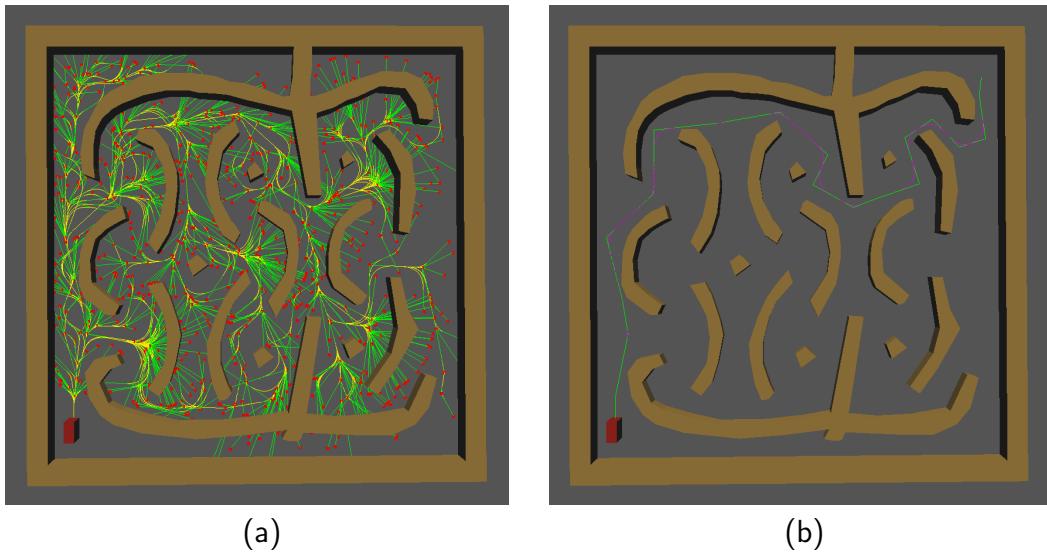


Figure 5.6: Bézier spline based RRT in a cluttered 3D space (a) Tree expansion results. (b) Solution path.

handle curvature-bounded vehicle models, and the obtained path can guarantee

curvature continuous. The start and goal settings are the same with those in clothoid spline based RRT example (Fig. 5.5a). The vehicle models used by these algorithms have the same minimum turning radius, i.e., the same κ_{\max} . Also the environment and vehicle size are not changed. In this demonstration, the lazy collision checking function is also implemented. The RRT is composed of 983 nodes and the solution path has 20 states in total. It takes 0.638 s to generate a feasible path (Fig. 5.6b) on the same computer. All examples tested here are executed on a 3.60 GHz PC and the programming language used is C++. After a series of comprehensive comparisons, we found that the Bézier spline based RRT has a comparable performance with the clothoid spline version. In this example, the clothoid version has more states than the Bézier planner does. However, most computational time is used for collision detection, where the clothoid based local planner contains circular arcs in CASE 2 and the clothoid curves approximated with Bézier curves. This helps to improve its performance due to the fact that collision checking on circular arcs are more efficient than Bézier curves.

5.3 Integration of Clothoid and Roadmap

We have mentioned the basic roadmap motion planning framework in Section 2.2.1, which is a very efficient approach for wheeled mobile robots planning subject to nonholonomic constraints. Apart from the multi-query benefit, roadmap can be used to address planning problems in complicated scenarios including corridor-like environments. However, the roads generated by a probabilistic roadmap are of irregular shape due to its randomized property. Thus they are not cost-effective and unsuitable for industrial vehicles to follow.

Here, we propose a new path planning framework based on roadmap targeting at path planning problems in a corridor-like environment, such as petrochemical plants and parking lots.

5.3.1 Preprocessing of the Environment Data

Nowadays, many actual environments are represented by point cloud data (PCD), which are acquired from a 3D laser scanner. Usually the point cloud data con-

tains a large number of points defined by x , y , and z coordinates and no other topological information included. In general, to acquire a PCD representation of an environment, one or more scans are needed, which generates a group of PCD models with internal relationships. Let us consider a practical environment, which is a parking lot with narrow passages of irregular geometric shapes. The raw data are composed of 8 individual scans shown in Fig. 5.7.

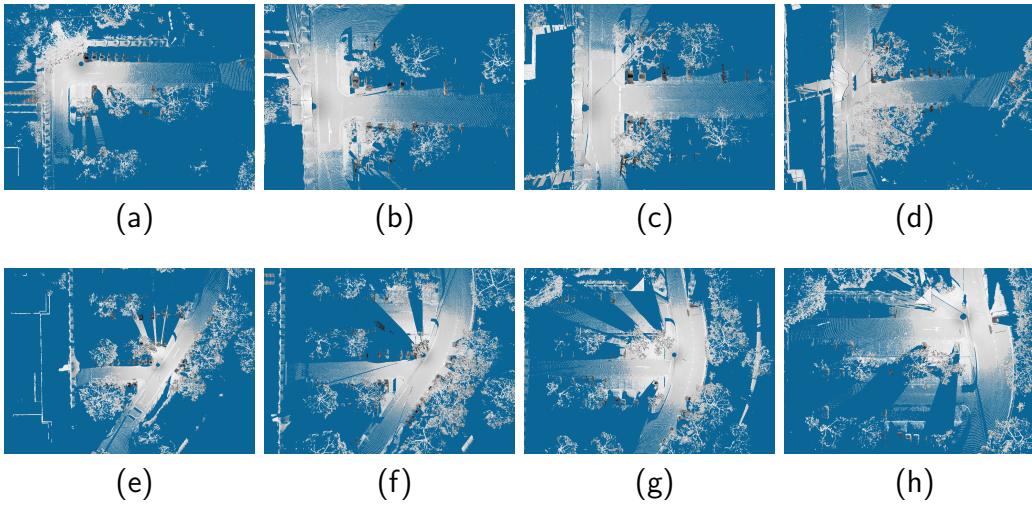


Figure 5.7: Individual scans of the car park environment: each dataset contains up to 4.3×10^7 points.

As can be noticed, the raw dataset usually contains noisy measurements, and some appear as speckles or ghost points. They can be filtered and trimmed based on the mean distance from each point to all its neighbors, i.e., all points whose mean distances do not obey the Gaussian distribution are automatically considered as noises or outliers with higher probability. This process can be done with the sparse outlier removal of the Point Cloud Library [116]. After noise reduction, the individual scans are aligned relatively to each other and merged into a whole scene based on PCD representations, as shown in Fig. 5.8a. Then the general road information is extracted based on plane segmentation approach which uses a random sample consensus (RANSAC) algorithm [117]. After manually cleaning the redundant objects, the point cloud of the road is obtained and illustrated in Fig. 5.8b. In the end, based the PCD model, we are able to generate the triangular

mesh representation of the road with Poisson surface reconstruction algorithm [118], which is shown in Fig. 5.8c. Consequently, a binary image representation can also be easily obtained.

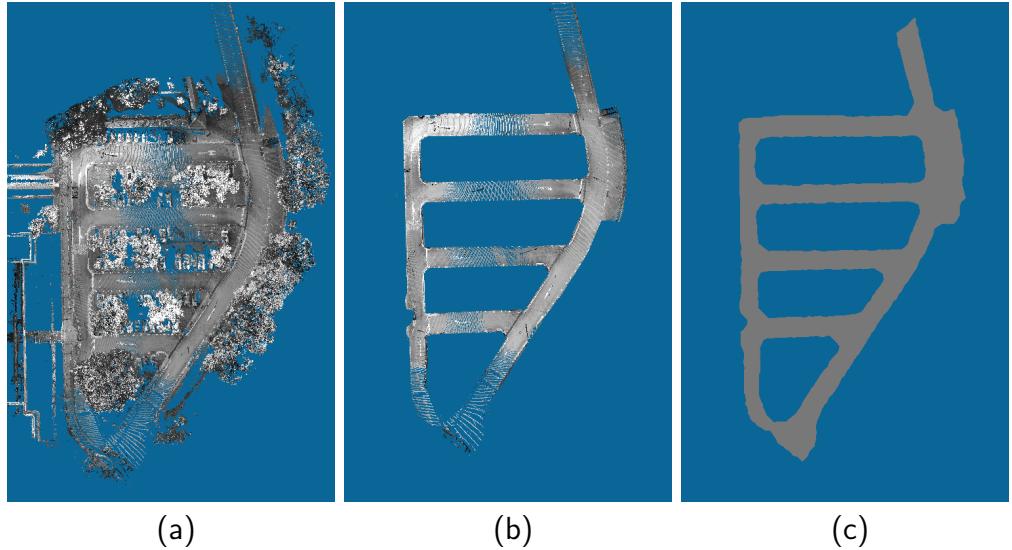


Figure 5.8: Road information extraction (courtesy of Dr. Xiaoqun Wu): (a) Point cloud representation of the car park environment in grayscale. (b) Road segmented from the PCD dataset. (c) Road mesh generated from the PCD model.

5.3.2 Roadmap Representation

Then we can use various thinning methods, like [119] or [120], to get the straight skeleton of the road based on the mesh or image representation. Here, Fig. 5.9a is obtained using a morphological thinning method [121], and after pruning outer branches, a roadmap with curved edges is derived in Fig. 5.9b. Then we can further simplify the resulting roadmap via RamerDouglasPeucker algorithm [122] or other shape simplification approaches, as shown in Fig. 5.9c. This roadmap can be used for path planning directly according to different kinematic models. At each intersection point, any two edges form a possible turn with a fixed deflection angle. With such geometric information, local planners can be derived based on the requirements of different cases, which have been fully discussed in Chapter 3 and Section 5.1. Alternatively, the roadmap Fig. 5.9b can also be regarded as a Voronoi diagram, and a PRM can be built upon it directly, as proposed in [123].

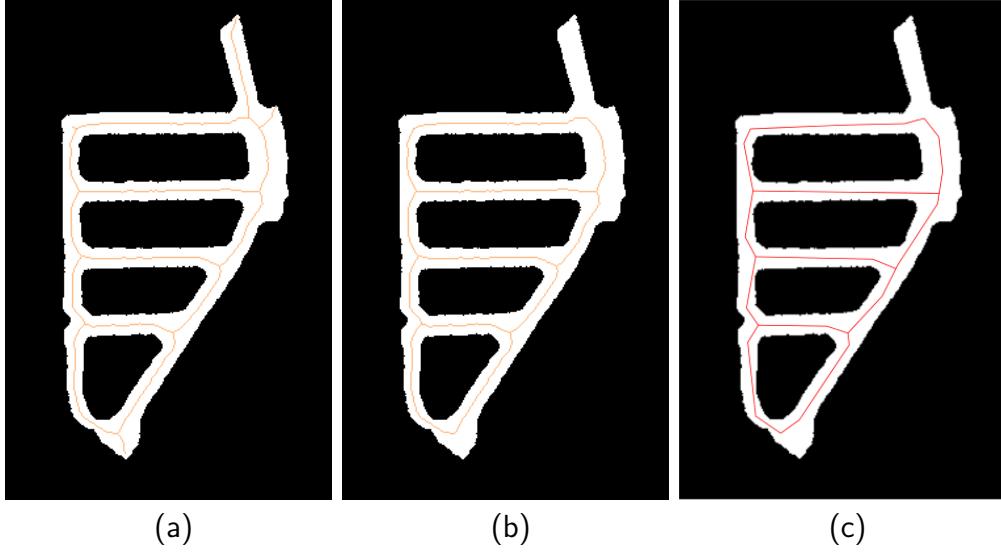


Figure 5.9: Roadmap processing: (a) Thinning. (b) Pruning. (c) Simplification.

5.4 Applications and Simulations

A simulation of local path planning is shown in Fig. 5.10, which gives a 3D perspective of a maneuvering process following a computed path. The mobile crane is driving at a relatively low velocity subject to the kinematic constraints of the vehicle as well as the environmental constraints. Among all the kinematic factors, the maximum steering angle ϕ_{\max} and the upper bound of the angular acceleration σ_{\max} are the most important two parameters. The former determines the turning radius while the latter characterizes the shape of the local path. The environmental constraints, on the other hand, are from the walls \mathcal{W}_i and \mathcal{W}_o , which we have analyzed in Chapter 3.

From a 3D perspective, the importance of these two walls are different. The inner wall \mathcal{W}_i is defined by the road boundary, thus collision checking is performed against the wheels instead of the vehicle body. At certain postures, the vehicle body may have crossed the wall \mathcal{W}_i but no actual collision occurs. On the other hand, the outer boundary composed of surrounding buildings is a hard constraint that a safe clearance must be maintained during the steering. Fig. 5.11 shows another path planning example, which is a cluttered space planning problem that

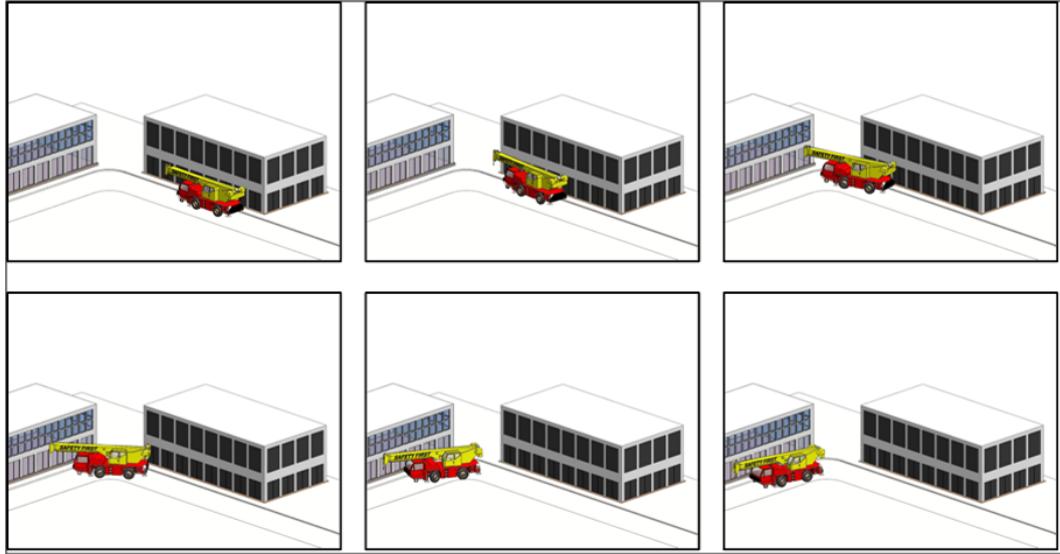


Figure 5.10: Local path planning.

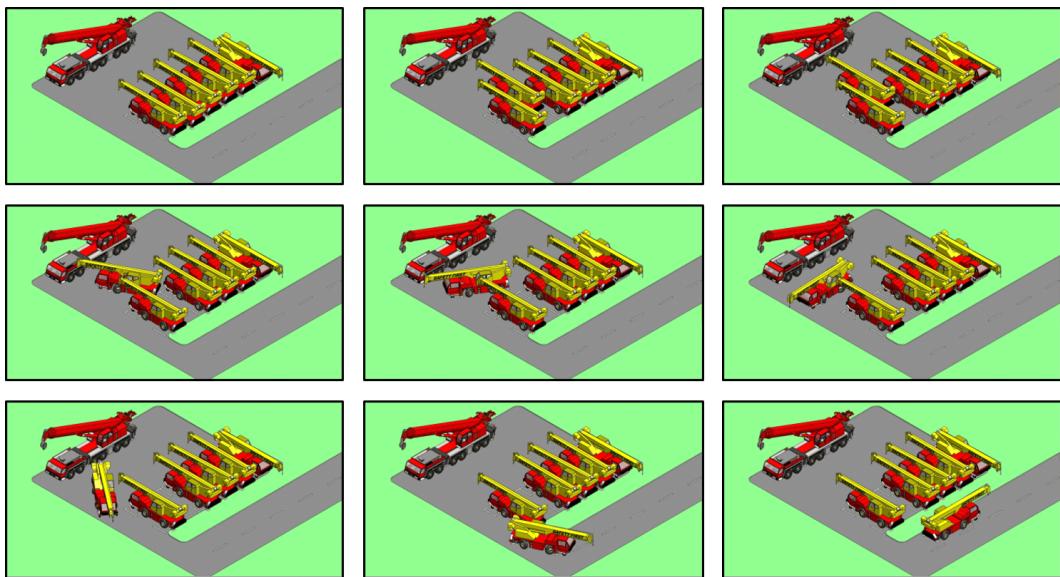


Figure 5.11: Path planning in a cluttered space.

the same mobile crane moves out of a parking lot by following the path computed by clothoid spline based RRT. Collision checking also plays an important role during the computation, where a bounding volume hierarchy (BVH) representation can be used for the moving vehicle. To improve the convergence rate, the maximum step size Δs_{\max} should not be too large for a comparably small sized environment

like this.

Chapter 6

Conclusions

In this work, we developed methodologies targeting at nonholonomic path planning problems of heavy vehicles. This category of vehicles usually drive at relatively low speeds to ensure safety. In fact, the path planning framework we proposed here applies to not only heavy vehicles, but also general wheeled mobile robots, as long as they satisfy certain kinematic constraints. To cover most types of vehicles, we started with generalizing a mathematical model to represent different types of vehicles. Then we developed a path planning framework composed of two phases: local path planning and global path planning. Subject to angular acceleration constraint, the local optimal path of the vehicle involves clothoid curves. By designing a family of path primitives based on clothoids, we are able to obtain a near-optimal solution without numerically solving the boundary value problem. Thus, it is convenient and efficient to handle this class of nonholonomic planning problems with parametrization of the kinematic models as well as the waypoints. To facilitate the computation, an approximation approach based on quintic Bézier curves is proposed. This local planner can be conveniently integrated with existing global planners like PRM, RRT and their variants.

6.1 Contributions

There are many existing path planning approaches and algorithms, and most of them cannot be directly applied to industrial vehicles. Although these vehicles belong to the family of wheeled mobile robots, they have several distinguishing

characteristics. The most significant one is the body size and the steering geometry. In the path planning stage especially for collision detection, the significant size cannot be simplified by a point robot as used by most planning frameworks. Furthermore, vehicles of different types have distinct steering structures and dynamic systems. Thus it is extremely difficult to represent the vehicles precisely using a universal dynamic model that may contain all possible related parameters to characterize the dynamic properties of the tires, drivelines including gearboxes and engines, steering and suspension mechanisms, etc. Even if such a complicated dynamic model is built, it cannot be easily applied to another vehicle, and a large amount of work is needed to fine tune the model, which may involve substantial changes and modifications. Still, it is very challenging to establish an effective planning framework based on such an accurate model, which requires a complicated controller specifically designed to tackle certain environmental conditions. Furthermore, this kind of planning framework may not be able to achieve real time performance for online applications. Consequently, we focus on a general kinematic model, which is referred to as a bicycle or single-track model. This model was proposed to describe the kinematic constraints for simple cars like Dubins car and Reeds-Shepp car. However, after examining vehicles with different steering structures and driving modes, it is concluded that the bicycle model can be used as a universal kinematic description such that all the conventional single-unit vehicles including many mobile robots can be represented through a few minor changes. The assumption is that at any moment all the wheels are in pure rolling motion subject to nonslip conditions, and during a simple turn all these wheels share a common ICR. Based on the analysis of the steering and fixed wheels, we show that most vehicle systems can be simplified such as two-axle FWS and RWS vehicles, multi-axle vehicles as well as 4WS vehicles. When they are controlled by angular acceleration instead of angular velocity, a general model is obtained for these vehicles, where the reference point is determined by the vehicle driving mode. The kinematic constraints include a maximum curvature and an upper-bounded angular acceleration. With the assumption of the velocity being constant, the former corresponds to the minimum turning radius subject to the maximum steering

angle allowed, while the latter is dependent on the design of the steering system of the vehicle.

Based on the simplified kinematic model, we design a family of local path primitives involving clothoid curves to smooth a simple turn formed by specified waypoints. Depending on the value of the deflection angle as well as the edge lengths, the local path primitives can be classified into four categories. The general case is composed of a pair of symmetric clothoid curves with opposite sharpness values, which meet at a junction where they share the same values of tangential angle and curvature. Thus the obtained path is a composite curve made of clothoids and straight lines. Since the curvature of each clothoid arc is designed to start from zero to the maximum linearly, the curvature values are continuous at each intersection formed by clothoids and straight lines. Consequently, the resulting path is curvature continuous, which does not require the vehicle to stop at each junction to reorient the wheels. Also we computed the characteristic parameters such as the distance from the corner as well as the path length for swept path analysis, both of which can be used as the distance metric in a global path planner. Based on specified waypoints, the allowed range of sharpness, which corresponds to the angular acceleration of the vehicle, is determined. By discretizing this range, a group of feasible primitives is obtained, which can be used for collision avoidance and integrated with sampling based planners. Apart from the general case, the other three types of path primitives are also discussed to adapt to different situations. They may involve circular arcs to deal with smaller deflection angles, or using unsymmetrical configurations to cater for extreme edge lengths or to avoid possible collisions with obstacles. Combining these primitives, we are able to smooth a path defined by waypoints subject to specific kinematic constraints. By comparing with other curvature continuous smoothing methods, we show that the resulting path is a G^2 continuous curve with a smaller curvature maxima and simultaneously ensures an upper-bounded curvature derivative. As a feasible local planner, the path primitive must have the capability to handle collisions by adjusting related parameters. Thus we can compute the swept volume of a vehicle based on a mathematical front-rear wheel model. The swept paths in case

of following a straight line, circular arc, and a clothoid are analyzed, providing steering templates for an actual vehicle body with specific dimensions.

Clothoid curve is involved in both the local planner and path smoothing; however, it does not have a closed form representation. Thus to reduce the computing time without loss of geometric properties, we present a smooth approximation method for clothoid using quintic Bézier curves. The approximation framework is divided into three stages. First, a category of clothoids defined as elementary clothoids is approximated by a piece of quintic Bézier curve to minimize the curvature error subject to G^3 continuity at both ends. Then we define another group of clothoids as basic clothoids, and mathematically, they can be obtained by concatenating a sequence of elementary clothoids. Next, a lookup table is computed offline to store the data of the basic clothoids approximation. Finally, a piece of general clothoid with arbitrary parameter can be approximated efficiently by querying the lookup table via appropriate geometric transformations. During the construction of the lookup table, the error can be bounded to an acceptable tolerance by restricting the winding angle of each elementary piece within a limited range. We show that this method is robust and can always guarantee G^3 continuity even under divergent conditions. The effectiveness of the approximation method is explained through comparing with existing approaches. In practice, to provide more flexibility, the approximation supports uniform and nonuniform sampling and the accuracy can be improved by increasing the number of basic segments.

Since the proposed primitives are able to reflect all the possible trajectories of the kinematic system with a given input in a limited time, they can be integrated with incremental search algorithms. First, we represent the path primitives of CASE 1 and CASE 2 with respect to the initial node of the expansion tree. The reachability of a local tree is evaluated by using the edge length as the step size. To work with the RRT framework, the step size is restricted to a range, and the lower and upper bounds are determined by the deflection angle and the environment, respectively. A clothoid spline based RRT algorithm is proposed and it takes advantage of the efficiency of the randomized sampling strategy as well as complying with the kinematic constraints of the vehicle. Thus, the role of the

local planner which is responsible for solving the boundary value problem is replaced with an efficient geometric planner. To further improve the performance, lazy collision checking is implemented. Only the actual curved paths are validated during an iteration while the outgoing edges are left unchecked unless they are connected by descendant nodes during the subsequent iterations. The effectiveness and efficiency are illustrated with a 3D space planning example, which requires a box shaped vehicle moving between two discrete configurations in a complex maze environment. Assuming the local planner has been established, the most difficult part to integrating with the PRM is the map preprocessing. An example of point cloud data representation is processed to extract the road structure, and the final roadmap is obtained to be integrated with the existing local path planner.

6.2 Future Work

Although the proposed geometric based path planning framework and the clothoid approximation approach are able to solve a class of fundamental problems, many areas remains unexplored.

One of the most important tasks in the future is the validation of the proposed algorithm on practical vehicles and roadways. The path planning framework has been implemented for generating standard trajectory report for evaluation, and also it has been integrated with a simulator used for training purposes by a local company. However, we have not validated the algorithm on a real vehicle platform. In the future, the effectiveness of this approach will be tested on actual mobile cranes mounted with sensors. The data will be collected to analyze the overall performance, and the algorithm may need further improvements with regard to the accuracy and efficiency.

Furthermore, the clothoid approximation method proposed here can also be used in road design and other applications where clothoids are involved. Though only approximation of clothoid curve is addressed in this research, it is promising to extend this approach to approximate other spiral curves such as logarithmic spirals. The application of this method will be further explored such as computer-aided design, graphics representation. So far we have only studied the planar

motion, while in the future we will apply the algorithm to space curves for 3D path generation. Also, Bézier surfaces are widely used in computer graphics as well as computer-aided geometric design, and the quintic Bézier curves we proposed here belong to a special class of common Bézier curves. Thus we can generate a group of Bézier surfaces with advanced properties based on the approximation of clothoid splines we obtained.

Finally, another aspect of future work is to improve the completeness of the planning framework. Now we have only considered single-unit vehicles with steerable and fixed wheels. However, articulated vehicles are very common in industrial applications, such as multi-trailer trucks. Though the bicycle model can be further improved and extended to a chained form, more complications can arise such as the jackknife problem. Also, if the vehicle is allowed to stop and reverse, the node selection and tree expansion phases of the clothoid spline based RRT would have to be redesigned. On the other hand, developing an asymptotically optimal version of the spline based planner would be an interesting direction. Meanwhile, the existing algorithm can be further improved with more efficient data structures and distance metrics, which is still an underdeveloped research area for nonholonomic planning problems.

Appendix A

A.1 G^3 Approximation

Apply the Beta-constraints to the start and end points to obtain

$$\begin{pmatrix} \mathbf{B}_\varepsilon(0) \\ \mathbf{B}'_\varepsilon(0) \\ \mathbf{B}''_\varepsilon(0) \\ \mathbf{B}_\varepsilon^{(3)}(0) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \beta_1 & 0 & 0 \\ 0 & \beta_2 & \beta_1^2 & 0 \\ 0 & \beta_3 & 3\beta_1\beta_2 & \beta_1^3 \end{pmatrix} \begin{pmatrix} \mathbf{F}_\varepsilon(0) \\ \mathbf{F}'_\varepsilon(0) \\ \mathbf{F}''_\varepsilon(0) \\ \mathbf{F}_\varepsilon^{(3)}(0) \end{pmatrix} \quad (\text{A.1})$$

and

$$\begin{pmatrix} \mathbf{B}_\varepsilon(1) \\ \mathbf{B}'_\varepsilon(1) \\ \mathbf{B}''_\varepsilon(1) \\ \mathbf{B}_\varepsilon^{(3)}(1) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \gamma_1 & 0 & 0 \\ 0 & \gamma_2 & \gamma_1^2 & 0 \\ 0 & \gamma_3 & 3\gamma_1\gamma_2 & \gamma_1^3 \end{pmatrix} \begin{pmatrix} \mathbf{F}_\varepsilon(l) \\ \mathbf{F}'_\varepsilon(l) \\ \mathbf{F}''_\varepsilon(l) \\ \mathbf{F}_\varepsilon^{(3)}(l) \end{pmatrix} \quad (\text{A.2})$$

respectively, where β_i, γ_i ($i = 1, 2, 3$) are shape parameters and $\beta_1 > 0, \gamma_1 > 0$.

The derivatives of the quintic Bézeir curve can be easily computed from equation (4.4) as

$$\begin{pmatrix} \mathbf{B}_\varepsilon(0) \\ \mathbf{B}'_\varepsilon(0) \\ \mathbf{B}''_\varepsilon(0) \\ \mathbf{B}_\varepsilon^{(3)}(0) \end{pmatrix} = \begin{pmatrix} \mathbf{V}_0 \\ -5\mathbf{V}_0 + 5\mathbf{V}_1 \\ 20\mathbf{V}_0 - 40\mathbf{V}_1 + 20\mathbf{V}_2 \\ -60\mathbf{V}_0 + 180\mathbf{V}_1 - 180\mathbf{V}_2 + 60\mathbf{V}_3 \end{pmatrix} \quad (\text{A.3})$$

and

$$\begin{pmatrix} \mathbf{B}_\varepsilon(1) \\ \mathbf{B}'_\varepsilon(1) \\ \mathbf{B}''_\varepsilon(1) \\ \mathbf{B}_\varepsilon^{(3)}(1) \end{pmatrix} = \begin{pmatrix} \mathbf{V}_5 \\ -5\mathbf{V}_4 + 5\mathbf{V}_5 \\ 20\mathbf{V}_3 - 40\mathbf{V}_4 + 20\mathbf{V}_5 \\ -60\mathbf{V}_2 + 180\mathbf{V}_3 - 180\mathbf{V}_4 + 60\mathbf{V}_5 \end{pmatrix}. \quad (\text{A.4})$$

Similarly, the derivatives of the corresponding elementary clothoid (4.2) at the start and end points are

$$\begin{aligned} \mathbf{F}_\varepsilon(0) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \mathbf{F}'_\varepsilon(0) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \mathbf{F}''_\varepsilon(0) &= \begin{pmatrix} 0 \\ \kappa_0 \end{pmatrix} & \mathbf{F}_\varepsilon^{(3)}(0) &= \begin{pmatrix} -\kappa_0^2 \\ \sigma \end{pmatrix} \end{aligned} \quad (\text{A.5})$$

and

$$\begin{aligned}\mathbf{F}_\varepsilon(l) &= \begin{pmatrix} x_l \\ y_l \end{pmatrix} \\ \mathbf{F}'_\varepsilon(l) &= \begin{pmatrix} \cos\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right) \\ \sin\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right) \end{pmatrix} \\ \mathbf{F}''_\varepsilon(l) &= \begin{pmatrix} -\sin\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right)(l\sigma + \kappa_0) \\ \cos\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right)(l\sigma + \kappa_0) \end{pmatrix} \\ \mathbf{F}_\varepsilon^{(3)}(l) &= \begin{pmatrix} -\sigma \sin\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right) - \cos\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right)(l\sigma + \kappa_0)^2 \\ \sigma \cos\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right) - \sin\left(\frac{1}{2}l(l\sigma + 2\kappa_0)\right)(l\sigma + \kappa_0)^2 \end{pmatrix}\end{aligned}\tag{A.6}$$

respectively, where (x_l, y_l) is the end point.

Thus, we have

$$\begin{pmatrix} \mathbf{V}_0 \\ \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \end{pmatrix} = \begin{pmatrix} \mathbf{F}_\varepsilon(0) \\ \mathbf{F}_\varepsilon(0) + \frac{\beta_1}{5}\mathbf{F}'_\varepsilon(0) \\ \mathbf{F}_\varepsilon(0) + \frac{8\beta_1+\beta_2}{20}\mathbf{F}'_\varepsilon(0) + \frac{\beta_1^2}{20}\mathbf{F}''_\varepsilon(0) \\ \mathbf{F}_\varepsilon(0) + \frac{36\beta_1+9\beta_2+\beta_3}{60}\mathbf{F}'_\varepsilon(0) + \frac{9\beta_1^2+3\beta_1\beta_2}{60}\mathbf{F}''_\varepsilon(0) + \frac{\beta_1^3}{60}\mathbf{F}_\varepsilon^{(3)}(0) \end{pmatrix} \quad (\text{A.7})$$

and

$$\begin{pmatrix} \mathbf{V}_2 \\ \mathbf{V}_3 \\ \mathbf{V}_4 \\ \mathbf{V}_5 \end{pmatrix} = \begin{pmatrix} \mathbf{F}_\varepsilon(l) + \frac{-36\gamma_1 + 9\gamma_2 - \gamma_3}{60} \mathbf{F}'_\varepsilon(l) + \frac{9\gamma_1^2 - 3\gamma_1\gamma_2}{60} \mathbf{F}''_\varepsilon(l) - \frac{\gamma_1^3}{60} \mathbf{F}_\varepsilon^{(3)}(l) \\ \mathbf{F}_\varepsilon(l) + \frac{-8\gamma_1 + \gamma_2}{20} \mathbf{F}'_\varepsilon(l) + \frac{\gamma_1^2}{20} \mathbf{F}''_\varepsilon(l) \\ \mathbf{F}_\varepsilon(l) - \frac{\gamma_1}{5} \mathbf{F}'_\varepsilon(l) \\ \mathbf{F}_\varepsilon(l) \end{pmatrix}. \quad (\text{A.8})$$

Expression (4.6) can be obtained from these two equations. From (A.7) and (A.8), we know that

$$\left\{ \begin{array}{l} \mathbf{F}_\varepsilon(0) + \frac{8\beta_1 + \beta_2}{20} \mathbf{F}'_\varepsilon(0) + \frac{\beta_1^2}{20} \mathbf{F}''_\varepsilon(0) = \mathbf{F}_\varepsilon(l) + \frac{-36\gamma_1 + 9\gamma_2 - \gamma_3}{60} \mathbf{F}'_\varepsilon(l) \\ \qquad \qquad \qquad + \frac{9\gamma_1^2 - 3\gamma_1\gamma_2}{60} \mathbf{F}''_\varepsilon(l) - \frac{\gamma_1^3}{60} \mathbf{F}_\varepsilon^{(3)}(l), \\ \mathbf{F}_\varepsilon(l) + \frac{-8\gamma_1 + \gamma_2}{20} \mathbf{F}'_\varepsilon(l) + \frac{\gamma_1^2}{20} \mathbf{F}''_\varepsilon(l) = \mathbf{F}_\varepsilon(0) + \frac{36\beta_1 + 9\beta_2 + \beta_3}{60} \mathbf{F}'_\varepsilon(0) \\ \qquad \qquad \qquad + \frac{9\beta_1^2 + 3\beta_1\beta_2}{60} \mathbf{F}''_\varepsilon(0) + \frac{\beta_1^3}{60} \mathbf{F}_\varepsilon^{(3)}(0). \end{array} \right. \quad (\text{A.9})$$

By denoting the x , y components of the i -th derivative ($i = 0, 1, 2, 3$) of the clothoid function $\mathbf{F}_\varepsilon(s)$ as $d_{s,x}^i$, $d_{s,y}^i$ respectively, shape parameters β_2 , β_3 , γ_2 , γ_3 can be expressed by solving for (A.9):

$$\begin{aligned}\beta_2(\beta_1, \gamma_1) &= (-3(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)(20(d_{0,y}^0 d_{l,x}^1 - d_{0,x}^0 d_{l,y}^1 + d_{l,y}^1 d_{l,x}^0 - d_{l,x}^1 d_{l,y}^0) \\ &\quad + \beta_1(8d_{0,y}^1 d_{l,x}^1 - 8d_{0,x}^1 d_{l,y}^1 + (d_{0,y}^2 d_{l,x}^1 - d_{0,x}^2 d_{l,y}^1) \beta_1)) + (d_{l,y}^2 d_{l,x}^1 \\ &\quad - d_{l,x}^2 d_{l,y}^1)(60(d_{0,y}^1(-d_{0,x}^0 + d_{l,x}^0) + d_{0,x}^1(d_{0,y}^0 - d_{l,y}^0)) + \beta_1^2(9d_{0,y}^2 d_{0,x}^1 \\ &\quad - 9d_{0,x}^2 d_{0,y}^1 + (d_{0,y}^3 d_{0,x}^1 - d_{0,x}^3 d_{0,y}^1) \beta_1)) \gamma_1 - 15(d_{0,y}^1 d_{l,x}^1 \\ &\quad - d_{0,x}^1 d_{l,y}^1)(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1) \gamma_1^2 + (d_{0,x}^1(-3(d_{l,y}^2)^2 d_{l,x}^1 + 3d_{l,x}^2 d_{l,y}^2 d_{l,y}^1 \\ &\quad + d_{l,y}^1(d_{l,y}^3 d_{l,x}^1 - d_{l,x}^3 d_{l,y}^1)) + d_{0,y}^1(3d_{l,x}^2 d_{l,y}^2 d_{l,x}^1 - 3(d_{l,x}^2)^2 d_{l,y}^1 \\ &\quad + d_{l,x}^1(-d_{l,y}^3 d_{l,x}^1 + d_{l,x}^3 d_{l,y}^1))) \gamma_1^3)/(3(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)^2 \\ &\quad - 3(d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1)(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1) \beta_1 \gamma_1), \\ \gamma_2(\beta_1, \gamma_1) &= (-15(d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1)(-d_{0,y}^1 d_{l,x}^1 + d_{0,x}^1 d_{l,y}^1) \beta_1^2 + (3(d_{0,y}^2)^2 d_{0,x}^1 d_{l,x}^1 \\ &\quad + d_{0,y}^3 d_{0,x}^1(-d_{0,y}^1 d_{l,x}^1 + d_{0,x}^1 d_{l,y}^1) - 3d_{0,x}^2 d_{0,y}^2(d_{0,y}^1 d_{l,x}^1 + d_{0,x}^1 d_{l,y}^1) \\ &\quad + d_{0,y}^1(d_{0,x}^3 d_{0,y}^1 d_{l,x}^1 + 3(d_{0,x}^2)^2 d_{l,y}^1 - d_{0,x}^3 d_{0,y}^1 d_{l,y}^1) \beta_1^3 + 3(d_{0,y}^1 d_{l,x}^1 \\ &\quad - d_{0,x}^1 d_{l,y}^1)(20(d_{0,y}^1(d_{0,x}^0 - d_{l,x}^0) + d_{0,x}^1(-d_{0,y}^0 + d_{l,y}^0)) + \gamma_1(8d_{0,y}^1 d_{l,x}^1 \\ &\quad - 8d_{0,x}^1 d_{l,y}^1 + (-d_{0,y}^1 d_{l,x}^2 + d_{0,x}^1 d_{l,y}^2) \gamma_1)) + (d_{0,y}^2 d_{0,x}^1 \\ &\quad - d_{0,x}^2 d_{0,y}^1) \beta_1(60(d_{0,y}^0 d_{l,x}^1 - d_{0,x}^0 d_{l,y}^1 + d_{l,y}^1 d_{l,x}^0 - d_{l,x}^1 d_{l,y}^0) \\ &\quad + \gamma_1^2(-9d_{l,y}^2 d_{l,x}^1 + 9d_{l,x}^2 d_{l,y}^1 + (d_{l,y}^3 d_{l,x}^1 - d_{l,x}^3 d_{l,y}^1) \gamma_1))/(3(d_{0,y}^1 d_{l,x}^1 \\ &\quad - d_{0,x}^1 d_{l,y}^1)^2 - 3(d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1)(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1) \beta_1 \gamma_1), \\ \beta_3(\beta_1, \gamma_1) &= (120(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)(d_{0,y}^0 d_{l,x}^1 - d_{0,x}^0 d_{l,y}^1 + d_{l,y}^1 d_{l,x}^0 - d_{l,x}^1 d_{l,y}^0) \\ &\quad + 12(3(d_{0,y}^1)^2(d_{l,x}^1)^2 - 6d_{0,x}^1 d_{0,y}^1 d_{l,x}^1 d_{l,y}^1 + 5d_{0,y}^2 d_{l,x}^1(d_{0,y}^0 d_{l,x}^1 - d_{0,x}^0 d_{l,y}^1) \\ &\quad + d_{l,y}^1 d_{l,x}^0 - d_{l,x}^1 d_{l,y}^0) + d_{l,y}^1(3(d_{0,x}^1)^2 d_{l,y}^1 + 5d_{0,x}^2(-d_{0,y}^0 d_{l,x}^1 + d_{0,x}^0 d_{l,y}^1) \\ &\quad - d_{l,y}^1 d_{l,x}^0 + d_{l,x}^1 d_{l,y}^0))) \beta_1 + 24(d_{0,y}^2 d_{l,x}^1 - d_{0,x}^2 d_{l,y}^1)(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1) \beta_1^2 \\ &\quad + 3(d_{0,y}^2)^2(d_{l,x}^1)^2 \beta_1^3 - d_{0,y}^3 d_{0,x}^1(d_{l,x}^1)^2 \beta_1^3 - 6d_{0,x}^2 d_{0,y}^2 d_{l,x}^1 d_{l,y}^1 \beta_1^3 \\ &\quad + d_{0,y}^3 d_{0,x}^1 d_{l,x}^1 d_{l,y}^1 \beta_1^3 + d_{0,x}^3 d_{0,y}^1 d_{l,x}^1 d_{l,y}^1 \beta_1^3 + 3(d_{0,x}^2)^2(d_{l,y}^1)^2 \beta_1^3 \\ &\quad - d_{0,x}^3 d_{0,y}^1(d_{l,y}^1)^2 \beta_1^3 - (d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1)(180(d_{0,y}^1(-d_{0,x}^0 + d_{l,x}^0) \\ &\quad + d_{0,x}^1(d_{0,y}^0 - d_{l,y}^0)) + \beta_1(60(d_{0,y}^2(-d_{0,x}^0 + d_{l,x}^0) + d_{0,x}^2(d_{0,y}^0 - d_{l,y}^0)))\end{aligned}$$

$$\begin{aligned}
& + \beta_1(-9d_{0,y}^2 d_{0,x}^1 + 9d_{0,x}^2 d_{0,y}^1 + \beta_1(3d_{0,y}^3 d_{0,x}^1 - 3d_{0,x}^3 d_{0,y}^1 + (d_{0,y}^3 d_{0,x}^2 \\
& - d_{0,x}^3 d_{0,y}^2) \beta_1))) \gamma_1 + 3(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1)(16(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1) \\
& + 5(d_{0,y}^2 d_{l,x}^1 - d_{0,x}^2 d_{l,y}^1) \beta_1) \gamma_1^2 + (d_{l,x}^3 (d_{l,y}^1)^2 (3d_{0,x}^1 + d_{0,x}^2 \beta_1) \\
& + d_{l,x}^1 (-9d_{0,y}^1 d_{l,x}^2 d_{l,y}^2 + 9d_{0,x}^1 (d_{l,y}^2)^2 + 3d_{0,y}^1 d_{l,y}^3 d_{l,x}^1 + (-3d_{0,y}^2 d_{l,x}^2 d_{l,y}^2 \\
& + 3d_{0,x}^2 (d_{l,y}^2)^2 + d_{0,y}^2 d_{l,y}^3 d_{l,x}^1) \beta_1) + d_{l,y}^1 (-d_{l,x}^1 (3(d_{0,y}^1 d_{l,x}^3 + d_{0,x}^1 d_{l,y}^3) \\
& + (d_{0,y}^2 d_{l,x}^3 + d_{0,x}^2 d_{l,y}^3) \beta_1) + 3d_{l,x}^2 (3d_{0,y}^1 d_{l,x}^2 - 3d_{0,x}^1 d_{l,y}^2 + (d_{0,y}^2 d_{l,x}^2 \\
& - d_{0,x}^2 d_{l,y}^2) \beta_1)) \gamma_1^3) / ((d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)^2 \\
& - (d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1)(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1) \beta_1 \gamma_1), \\
\gamma_3(\beta_1, \gamma_1) = & (120(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)(d_{0,y}^1 (d_{0,x}^0 - d_{l,x}^0) + d_{0,x}^1 (-d_{0,y}^0 + d_{l,y}^0)) \\
& + 12((d_{0,y}^1)^2 (-5d_{0,x}^0 d_{l,x}^2 + 3(d_{l,x}^1)^2 + 5d_{l,x}^2 d_{0,x}^0) + d_{0,x}^1 d_{0,y}^1 (5d_{0,y}^0 d_{l,x}^2 \\
& + 5d_{0,x}^0 d_{l,y}^2 - 6d_{l,x}^1 d_{l,y}^1 - 5d_{l,y}^2 d_{l,x}^0 - 5d_{l,x}^2 d_{l,y}^0) + (d_{0,x}^1)^2 (-5d_{0,y}^0 d_{l,y}^2 \\
& + 3(d_{l,y}^1)^2 + 5d_{l,y}^2 d_{l,y}^0) \gamma_1 - 24(d_{0,y}^1 d_{l,x}^2 - d_{0,x}^1 d_{l,y}^2)(d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1) \gamma_1^2 \\
& + (d_{0,y}^1)^2 (3(d_{l,x}^2)^2 - d_{l,x}^3 d_{l,x}^1) + d_{0,x}^1 d_{0,y}^1 (-6d_{l,x}^2 d_{l,y}^2 + d_{l,y}^3 d_{l,x}^1 + d_{l,x}^3 d_{l,y}^1) \\
& + (d_{0,x}^1)^2 (3(d_{l,y}^2)^2 - d_{l,y}^3 d_{l,y}^1) \gamma_1^3 - 3(d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1) \beta_1^2 (16(-d_{0,y}^1 d_{l,x}^1 \\
& + d_{0,x}^1 d_{l,y}^1) + 5(d_{0,y}^1 d_{l,x}^2 - d_{0,x}^1 d_{l,y}^2) \gamma_1) + \beta_1^3 (3(3(d_{0,y}^2)^2 d_{0,x}^1 d_{l,x}^1 \\
& + d_{0,y}^3 d_{0,x}^1 (-d_{0,y}^1 d_{l,x}^1 + d_{0,x}^1 d_{l,y}^1) - 3d_{0,x}^2 d_{0,y}^2 (d_{0,y}^1 d_{l,x}^1 + d_{0,x}^1 d_{l,y}^1) \\
& + d_{0,y}^1 (d_{0,x}^3 d_{0,y}^1 d_{l,x}^1 + 3(d_{0,x}^2)^2 d_{l,y}^1 - d_{0,x}^3 d_{0,y}^1 d_{l,y}^1)) + (-3(d_{0,y}^2)^2 d_{0,x}^1 d_{l,x}^2 \\
& + d_{0,y}^3 d_{0,x}^1 (d_{0,y}^1 d_{l,x}^2 - d_{0,x}^1 d_{l,y}^2) + 3d_{0,x}^2 d_{0,y}^2 (d_{0,y}^1 d_{l,x}^2 \\
& + d_{0,x}^1 d_{l,y}^2) + d_{0,y}^1 (-d_{0,x}^3 d_{0,y}^1 d_{l,x}^2 - 3(d_{0,x}^2)^2 d_{l,y}^2 + d_{0,x}^3 d_{0,y}^1 d_{l,y}^2)) \gamma_1) \\
& + (d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1) \beta_1 (180(d_{0,y}^0 d_{l,x}^1 - d_{0,x}^0 d_{l,y}^1 + d_{l,y}^1 d_{l,x}^0 - d_{l,x}^1 d_{l,y}^0) \\
& + \gamma_1 (60(-d_{0,y}^0 d_{l,x}^2 + d_{0,x}^0 d_{l,y}^2 - d_{l,y}^2 d_{l,x}^0 + d_{l,x}^2 d_{l,y}^0) + \gamma_1 (9d_{l,y}^2 d_{l,x}^1 \\
& - 9d_{l,x}^2 d_{l,y}^1 + \gamma_1 (3d_{l,y}^3 d_{l,x}^1 - 3d_{l,x}^3 d_{l,y}^1 + (-d_{l,y}^3 d_{l,x}^2 + d_{l,x}^3 d_{l,y}^2) \gamma_1)))) \\
& / ((d_{0,y}^1 d_{l,x}^1 - d_{0,x}^1 d_{l,y}^1)^2 - (d_{0,y}^2 d_{0,x}^1 - d_{0,x}^2 d_{0,y}^1)(d_{l,y}^2 d_{l,x}^1 - d_{l,x}^2 d_{l,y}^1) \beta_1 \gamma_1).
\end{aligned}$$

By substituting $d_{s,x}^i$, $d_{s,y}^i$ with the corresponding values in (A.5) and (A.6), the above expressions can be further simplified. Thus, the common denominator shared by β_2 and γ_2 is computed as

$$D(\beta_1, \gamma_1) = \beta_1 \gamma_1 \kappa_0 (\kappa_0 + l\sigma) + \frac{1}{2} (\cos(l(2\kappa_0 + l\sigma)) - 1), \quad (\text{A.10})$$

which can be further simplified to equation (4.8). In addition, the numerators in (4.7) are

$$\begin{aligned}
B(\beta_1, \gamma_1) &= (-24\beta_1(\cos(l(2\kappa_0 + l\sigma)) - 1) - 2\gamma_1^2(\gamma_1\sigma + 15(\kappa_0 \\
&\quad + l\sigma))\sin(l(2\kappa_0 + l\sigma)/2) + 6\gamma_1^3(\kappa_0 + l\sigma)^2\cos(l(2\kappa_0 + l\sigma)/2) \\
&\quad + 60x_l(\cos(l(2\kappa_0 + l\sigma)) - 1) - 2\gamma_1(\kappa_0 + l\sigma)(\beta_1^2(\beta_1\sigma + 9\kappa_0) \\
&\quad - 60y_l) + 3\sin(l(2\kappa_0 + l\sigma))(20y_l - \beta_1^2\kappa_0))/6, \\
G(\beta_1, \gamma_1) &= (2\beta_1\gamma_1^2\kappa_0(9(\kappa_0 + l\sigma) - \gamma_1\sigma) + 3\gamma_1^2(\kappa_0 + l\sigma)\sin(l(2\kappa_0 + l\sigma)) \\
&\quad + 24\gamma_1(\cos(l(2\kappa_0 + l\sigma)) - 1) + 2\sin(l(2\kappa_0 + l\sigma)/2)(-\beta_1^3\sigma \\
&\quad + 15\beta_1\kappa_0(\beta_1 - 4x_l) + 60y_l) \\
&\quad - 6\beta_1\kappa_0\cos(l(2\kappa_0 + l\sigma)/2)(\beta_1^2\kappa_0 - 20y_l))/6.
\end{aligned} \tag{A.11}$$

A.2 Elementary Bézier Clothoid

Let (x_b, y_b) be the coordinates of the second endpoint of the clothoid curve of arc length u , the obtained β_2 and γ_2 can be rewritten as:

$$\begin{aligned}
\beta_2(k) &= \frac{B_k}{D_k}, \\
\gamma_2(k) &= \frac{G_k}{D_k},
\end{aligned} \tag{A.12}$$

according to (4.7) and (4.11), where

$$D_k = \beta_1\gamma_1 \left(\frac{\theta^2}{u^2} - \frac{k^2}{4} \right) - \sin^2(\theta), \tag{A.13}$$

$$\begin{aligned}
B_k &= \frac{1}{12u^2}(-6u\sin(\theta)(40ux_b\sin(\theta) - 40uy_b\cos(\theta) \\
&\quad + \beta_1(\beta_1\cos(\theta)(2\theta + ku) - 16u\sin(\theta))) \\
&\quad + \gamma_1(ku - 2\theta)(\beta_1^2(18\theta - 2\beta_1k + 9ku) - 120uy_b) \\
&\quad + 30\gamma_1^2u\sin(\theta)(ku - 2\theta) + \gamma_1^3(4ku\sin(\theta) \\
&\quad + 3\cos(\theta)(ku - 2\theta)^2)),
\end{aligned} \tag{A.14}$$

$$\begin{aligned}
G_k = & -\frac{1}{12u^2}(120\beta_1 ux_b \sin(\theta)(2\theta + ku) \\
& - 120uy_b(\beta_1 \cos(\theta)(2\theta + ku) + 2u \sin(\theta)) \\
& - 2\beta_1 \gamma_1^3 k(2\theta + ku) + 3\gamma_1^2(ku - 2\theta)(3\beta_1(2\theta + ku) \\
& + u \sin(2\theta)) + \beta_1^2(3\beta_1 \cos(\theta)(2\theta + ku)^2 \\
& - 2u \sin(\theta)(30\theta + 2\beta_1 k + 15ku)) + 96\gamma_1 u^2 \sin^2(\theta)).
\end{aligned} \tag{A.15}$$

Bézier curve BÉZIER ($t, u, \kappa_0, \sigma, \beta_1, \gamma_1, x_e, y_e$) satisfying G^3 continuity can be obtained by substituting relationship (4.7) into the following function:

$$\mathbf{Q}(t, u, \kappa_0, \sigma, \beta_1, \gamma_1, \beta_2, \gamma_2, x_b, y_b) = \begin{pmatrix} Q_x \\ Q_y \end{pmatrix}, \tag{A.16}$$

where

$$\begin{aligned}
Q_x = & \frac{1}{2}t(2t^4x_b - 2t^3(t-1)(5x_b - \gamma_1 \cos(\frac{1}{2}u(2\kappa_0 + \sigma u))) \\
& + t^2(t-1)^2(20x_b + \gamma_1^2(\kappa_0 + \sigma u)(-\sin(\frac{1}{2}u(2\kappa_0 \\
& + \sigma u)) + (\gamma_2 - 8\gamma_1) \cos(\frac{1}{2}u(2\kappa_0 + \sigma u))) \\
& + 2\beta_1(t-1)^4 - (8\beta_1 + \beta_2)t(t-1)^3),
\end{aligned} \tag{A.17}$$

$$\begin{aligned}
Q_y = & \frac{1}{2}t^2(2t^3y_b - 2t^2(t-1)(5y_b - \gamma_1 \sin(\frac{1}{2}u(2\kappa_0 + \sigma u))) \\
& + t(t-1)^2(20y_b + (\gamma_2 - 8\gamma_1) \sin(\frac{1}{2}u(2\kappa_0 + \sigma u)) \\
& + \gamma_1^2(\kappa_0 + \sigma u) \cos(\frac{1}{2}u(2\kappa_0 + \sigma u))) - \beta_1^2\kappa_0(t-1)^3).
\end{aligned} \tag{A.18}$$

References

- [1] M. Brezak and I. Petrovi, “Real-time approximation of clothoids with bounded error for path planning applications,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 507–515, 2014.
- [2] J.-w. Choi, R. Curry, and G. Elkaim, “Path planning based on Bézier curve for autonomous ground vehicles,” in *World Congress on Engineering and Computer Science 2008, WCECS’08. Advances in Electrical and Electronics Engineering-IAENG Special Edition of the.* IEEE, 2008, pp. 158–166.
- [3] H. Lum and J. A. Reagan, “Interactive highway safety design model: accident predictive module,” *Public Roads*, vol. 58, no. 3, pp. 14–17, 1995.
- [4] G. Robb, S. Sultana, S. Ameratunga, and R. Jackson, “A systematic review of epidemiological studies investigating risk factors for work-related road traffic crashes and injuries,” *Injury Prevention*, vol. 14, no. 1, pp. 51–58, 2008.
- [5] S. G. Pratt, “Work-related roadway crashes challenges and opportunities for prevention,” National Institute for Occupational Safety and Health (NIOSH), Tech. Rep., 2003.
- [6] R. Cox, “A review of the effect of increases in vehicle size on Australian geometric road design standards,” in *International Symposium on Heavy Vehicle Weights and Dimensions, 5th,,* vol. 3, 1998.
- [7] U.S. Department of Transportation, “The Large Truck Crash Causation Study (LTCCS),” <https://ai.fmcsa.dot.gov/LTCCS/>.

REFERENCES

- [8] S. Kharrazi, “Steering based lateral performance control of long heavy vehicle combinations,” Ph.D. dissertation, Chalmers University of Technology, 2012.
- [9] M. Gothié, “Heavy vehicle accident factors,” in *9th International Symposium on Heavy Vehicle Weights and Dimensions, Pennsylvania State University, PA, USA*, 2006.
- [10] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer-Verlag Berlin Heidelberg, 2011, vol. 73.
- [11] M. Harrer and P. Pfeffer, *Steering Handbook*. Springer International Publishing Switzerland, 2017.
- [12] G. Indiveri, “Kinematic time-invariant control of a 2D nonholonomic vehicle,” in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3. IEEE, 1999, pp. 2112–2117.
- [13] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed. Springer Science+Business Media, 2012.
- [14] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design,” in *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 1094–1099.
- [15] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [16] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, “Closed loop steering of unicycle like vehicles via Lyapunov techniques,” *IEEE Robotics & Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.
- [17] J.-P. Laumond, S. Sekhavat, and F. Lamiraux, “Guidelines in nonholonomic motion planning for mobile robots,” in *Robot Motion Planning and Control*. Springer-Verlag London Limited, 1998, pp. 1–53.
- [18] H. J. Sussmann and G. Tang, “Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control,” Department of Mathematics, Rutgers University, Tech. Rep., 1991.

REFERENCES

- [19] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.
- [20] J.-D. Boissonnat, A. Cerezo, and J. Leblond, “A note on shortest paths in the plane subject to a constraint on the derivative of the curvature,” INRIA, Research Report RR-2160, 1994.
- [21] A. Scheuer and C. Laugier, “Planning sub-optimal and continuous-curvature paths for car-like robots,” in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 1, Oct 1998, pp. 25–31 vol.1.
- [22] T. Fraichard and A. Scheuer, “From Reeds and Shepp’s to continuous-curvature paths,” *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.
- [23] T. Lozano-Pérez and M. A. Wesley, “An algorithm for planning collision-free paths among polyhedral obstacles,” *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.
- [24] C. Chen, M. Rickert, and A. Knoll, “Combining space exploration and heuristic search in online motion planning for nonholonomic vehicles,” in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 1307–1312.
- [25] M. Elbanhawi and M. Simic, “Sampling-based robot motion planning: A review,” *IEEE Access*, vol. 2, pp. 56–77, 2014.
- [26] J. Reeds and L. Shepp, “Optimal paths for a car that goes both forwards and backwards,” *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.
- [27] C. Chen, Y. He, C. Bu, J. Han, and X. Zhang, “Quartic Bézier curve based trajectory generation for autonomous vehicles with curvature and velocity constraints,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6108–6113.

REFERENCES

- [28] A. Neto, D. G. Macharet, M. F. Campos *et al.*, “Feasible RRT-based path planning using seventh order Bézier curves,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 1445–1450.
- [29] A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen, “Continuous-curvature path generation using Fermat’s spiral,” *Modeling, Identification and Control*, vol. 34, no. 4, p. 183, 2013.
- [30] H. Bruyninckx and D. Reynaerts, “Path planning for mobile and hyper-redundant robots using Pythagorean hodograph curves,” in *Advanced Robotics, 1997. ICAR’97. Proceedings., 8th International Conference on*. IEEE, 1997, pp. 595–600.
- [31] A. Scheuer and T. Fraichard, “Continuous-curvature path planning for car-like vehicles,” in *Intelligent Robots and Systems, 1997. IROS’97., Proceedings of the 1997 IEEE/RSJ International Conference on*, vol. 2. IEEE, 1997, pp. 997–1003.
- [32] H. Delingette, M. Hebert, and K. Ikeuchi, “Trajectory generation with curvature constraint based on energy minimization,” in *Intelligent Robots and Systems’ 91. ’Intelligence for Mechanical Systems, Proceedings IROS’91. IEEE/RSJ International Workshop on*. IEEE, Nov 1991, pp. 206–211 vol.1.
- [33] J. Pérez, J. Godoy, J. Villagra, and E. Onieva, “Trajectory generator for autonomous vehicles in urban environments,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 409–414.
- [34] A. Scheuer and T. Fraichard, “Collision-free and continuous-curvature path planning for car-like robots,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 867–873.

REFERENCES

- [35] J. Villagra, V. Milanés, J. Pérez, and J. Godoy, “Smooth path and speed planning for an automated public transport vehicle,” *Robotics and Autonomous Systems*, vol. 60, no. 2, pp. 252–265, 2012.
- [36] M. Shimizu, K. Kobayashi, and K. Watanabe, “Clothoidal curve-based path generation for an autonomous mobile robot,” in *2006 SICE-ICASE International Joint Conference*. IEEE, 2006, pp. 478–481.
- [37] M. Shanmugavel, A. Tsourdos, B. White, and R. Źbikowski, “Co-operative path planning of multiple UAVs using Dubins paths with clothoid arcs,” *Control Engineering Practice*, vol. 18, no. 9, pp. 1084–1092, 2010.
- [38] G. Arechavaleta, J. P. Laumond, H. Hicheur, and A. Berthoz, “An optimality principle governing human walking,” *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 5–14, Feb 2008.
- [39] B. A. Barsky and T. D. DeRose, “Geometric continuity of parametric curves: Three equivalent characterizations,” *IEEE Computer Graphics and Applications*, no. 6, pp. 60–68, 1989.
- [40] A. Kelly, *Mobile Robotics: Mathematics, Models, and Methods*. Cambridge University Press, 2013.
- [41] A. Gray, *Modern Differential Geometry of Curves and Surfaces with Mathematica*. CRC Press, 1996.
- [42] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, “Practical search techniques in path planning for autonomous driving,” *Ann Arbor*, vol. 1001, p. 48105, 2008.
- [43] Z. Habib and M. Sakai, “ G^2 PH quintic spiral transition curves and their applications,” *Scientiae Mathematicae Japonicae*, vol. 61, no. 2, pp. 207–218, 2005.
- [44] D. Walton, D. Meek, and J. Ali, “Planar G^2 transition curves composed of cubic Bézier spiral segments,” *Journal of Computational and Applied Mathematics*, vol. 157, no. 2, pp. 453–476, 2003.

REFERENCES

- [45] W. Nelson, “Continuous-curvature paths for autonomous vehicles,” in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on.* IEEE, 1989, pp. 1260–1264.
- [46] D. Meek and D. Walton, “Clothoid spline transition spirals,” *Mathematics of Computation*, vol. 59, no. 199, pp. 117–133, 1992.
- [47] R. Levien, “The Euler spiral: a mathematical history,” EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-111, 2008.
- [48] D. Meek and D. Walton, “A note on finding clothoids,” *Journal of Computational and Applied Mathematics*, vol. 170, no. 2, pp. 433–453, 2004.
- [49] P. Souères and J.-D. Boissonnat, “Optimal trajectories for nonholonomic mobile robots,” in *Robot Motion Planning and Control.* Springer-Verlag London Limited, 1998, pp. 93–170.
- [50] D. Walton and D. Meek, “Computer-aided design for horizontal alignment,” *Journal of Transportation Engineering*, vol. 115, no. 4, pp. 411–424, 1989.
- [51] A. Nutbourne, P. McLellan, and R. Kensit, “Curvature profiles for plane curves,” *Computer-Aided Design*, vol. 4, no. 4, pp. 176–184, 1972.
- [52] K. Yang and S. Sukkarieh, “3D smooth path planning for a UAV in cluttered natural environments,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008, pp. 794–800.
- [53] M. Abramowitz, I. A. Stegun *et al.*, *Handbook of Mathematical Functions.* Dover New York, 1965, vol. 1046.
- [54] R. Bulirsch, “Numerical calculation of the sine, cosine and Fresnel integrals,” *Numerische Mathematik*, vol. 9, no. 5, pp. 380–385, 1967.
- [55] K. D. Mielenz, “Computation of Fresnel integrals,” *Journal of Research of the National Institute of Standards and Technology*, vol. 102, no. 3, p. 363, 1997.

REFERENCES

- [56] L. Z. Wang, K. T. Miura, E. Nakamae, T. Yamamoto, and T. J. Wang, “An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves,” *Computer-Aided Design*, vol. 33, no. 14, pp. 1049–1058, 2001.
- [57] J. Sánchez-Reyes, “Applications of the polynomial s-power basis in geometry processing,” *ACM Transactions on Graphics*, vol. 19, no. 1, pp. 27–55, 2000.
- [58] J. Sánchez-Reyes and J. M. Chacón, “Polynomial approximation to clothoids via s-power series,” *Computer-Aided Design*, vol. 35, no. 14, pp. 1305–1313, 2003.
- [59] ——, “S-power series: an alternative to poisson expansions for representing analytic functions,” *Computer Aided Geometric Design*, vol. 22, no. 2, pp. 103 – 119, 2005.
- [60] R. Cripps, M. Hussain, and S. Zhu, “Smooth polynomial approximation of spiral arcs,” *Journal of Computational and Applied Mathematics*, vol. 233, no. 9, pp. 2227–2234, 2010.
- [61] B. Cross and R. J. Cripps, “ G^3 quintic polynomial approximation for generalised Cornu spiral segments,” *Journal of Computational and Applied Mathematics*, vol. 236, no. 13, pp. 3111–3122, 2012.
- [62] ——, “Efficient robust approximation of the generalised Cornu spiral,” *Journal of Computational and Applied Mathematics*, vol. 273, pp. 1–12, 2015.
- [63] N. Montes, M. C. Mora, and J. Tornero, “Trajectory generation based on rational Bezier curves as clothoids,” in *2007 IEEE Intelligent Vehicles Symposium*. IEEE, 2007, pp. 505–510.
- [64] N. Montés, A. Herraez, L. Armesto, and J. Tornero, “Real-time clothoid approximation by rational Bezier curves,” in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 2246–2251.

REFERENCES

- [65] F. Jean, *Control of Nonholonomic Systems: from Sub-Riemannian Geometry to Motion Planning.* Springer International Publishing, 2014.
- [66] L. Krammer, “Motion planning for car-like robots,” Master’s thesis, Vienna University of Technology, 2010.
- [67] B. Donald, P. Xavier, J. Canny, and J. Reif, “Kinodynamic motion planning,” *Journal of the ACM (JACM)*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [68] S. S. Ge and F. L. Lewis, *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications.* CRC Press, 2006, vol. 22.
- [69] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT press, 2005.
- [70] J.-P. Laumond, “Feasible trajectories for mobile robots with kinematic and environment constraints,” in *Intelligent Autonomous Systems, An International Conference.* North-Holland Publishing Co., 1986, pp. 346–354.
- [71] L. Palmieri and K. O. Arras, “A novel RRT extend function for efficient and smooth mobile robot motion planning,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2014, pp. 205–211.
- [72] J. Butzke, K. Sapkota, K. Prasad, B. MacAllister, and M. Likhachev, “State lattice with controllers: Augmenting lattice-based path planning with controller-based motion primitives,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2014, pp. 258–265.
- [73] K. Yang, S. Moon, S. Yoo, J. Kang, N. L. Doh, H. B. Kim, and S. Joo, “Spline-based RRT path planner for non-holonomic robots,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1-4, pp. 763–782, 2014.
- [74] S. S. Ge and Y. J. Cui, “Dynamic motion planning for mobile robots using potential field method,” *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, 2002.

REFERENCES

- [75] I. Duleba and J. Z. Sasiadek, “Nonholonomic motion planning based on newton algorithm with energy optimization,” *IEEE transactions on control systems technology*, vol. 11, no. 3, pp. 355–363, 2003.
- [76] R. A. Knepper and A. Kelly, “High performance state lattice planning using heuristic look-up tables.” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3375–3380.
- [77] J. J. Kuffner and S. M. LaValle, “RRT-connect: An efficient approach to single-query path planning,” in *Robotics and Automation, 2000. Proceedings. ICRA ’00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.
- [78] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [79] I. A. Sucan and L. E. Kavraki, “Kinodynamic motion planning by interior-exterior cell exploration,” in *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 449–464.
- [80] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [81] N. M. Amato and Y. Wu, “A randomized roadmap method for path and manipulation planning,” in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 113–120.
- [82] D. Hsu, G. Sánchez-Ante, and Z. Sun, “Hybrid PRM sampling with a cost-sensitive adaptive strategy,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 3874–3880.
- [83] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, “Narrow passage sampling for probabilistic roadmap planning,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1105–1115, 2005.

REFERENCES

- [84] S. Rolewicz, *Functional Analysis and Control Theory: Linear Systems*. Springer Science+Business Media Dordrecht, 1987, vol. 29.
- [85] R. Bohlin and L. E. Kavraki, “Path planning using lazy PRM,” in *Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on*, vol. 1. IEEE, 2000, pp. 521–528.
- [86] S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” in *Proceedings IEEE International Conference on Robotics and Automation*, vol. 1, 1999, pp. 473–479.
- [87] W. Chung and K. Iagnemma, “Wheeled robots,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer-Verlag Berlin Heidelberg, 2016, pp. 575–594.
- [88] G. Campion, G. Bastin, and B. Dandrea-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, Feb 1996.
- [89] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer-Verlag London Limited, 2010.
- [90] R. N. Jazar, *Vehicle Dynamics: Theory and Application*, 2nd ed. Springer Science+Business Media, 2014.
- [91] G. Qin, Y. Sun, Y. Zhang, and L. Chen, “Analysis and optimization of the double-axle steering mechanism with dynamic loads,” *The Open Mechanical Engineering Journal*, vol. 6, pp. 26–39, 2012.
- [92] Information Resources Management Association, *Robotics: Concepts, Methodologies, Tools, and Applications (3 Volumes)*. IGI Global, 2014.
- [93] P. Fancher and C. Winkler, “Directional performance issues in evaluation and design of articulated heavy vehicles,” *Vehicle System Dynamics*, vol. 45, no. 7-8, pp. 607–647, 2007.

REFERENCES

- [94] C. B. Winkler, “Simplified analysis of the steady-state turning of complex vehicles,” *Vehicle System Dynamics*, vol. 29, no. 3, pp. 141–180, 1998.
- [95] M. Elbanhawi, M. Simic, and R. N. Jazar, “Continuous path smoothing for car-like robots using B-spline curves,” *Journal of Intelligent & Robotic Systems*, vol. 80, no. 1, pp. 23–56, 2015.
- [96] U.-Y. Huh and S.-R. Chang, “A G^2 continuous path-smoothing algorithm using modified quadratic polynomial interpolation,” *International Journal of Advanced Robotic Systems*, vol. 11, no. 25, 2014.
- [97] T. Sakkalis, R. T. Farouki, and L. Vaserstein, “Non-existence of rational arc length parameterizations for curves in,” *Journal of Computational and Applied Mathematics*, vol. 228, no. 1, pp. 494 – 497, 2009.
- [98] V. Hernndez-Mederos and J. Estrada-Sarlabous, “Sampling points on regular parametric curves with control of their distribution,” *Computer Aided Geometric Design*, vol. 20, no. 6, pp. 363–382, 2003.
- [99] J. Bruce and M. Veloso, “Real-time randomized path planning for robot navigation,” in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2383–2388 vol.3.
- [100] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, 2008.
- [101] A. F. van der Stappen and M. H. Overmars, “Motion planning amidst fat obstacles,” in *Proceedings of the Tenth Annual Symposium on Computational Geometry*, ser. SCG ’94. New York, NY, USA: ACM, 1994, pp. 31–40.
- [102] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan, “Efficient collision detection using bounding volume hierarchies of k-DOPs,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 21–36, Jan 1998.

REFERENCES

- [103] M. Sayers, “Exact equations for tractrix curves associated with vehicle off-tracking,” *Vehicle System Dynamics*, vol. 20, no. 5, pp. 297–308, 1991.
- [104] K. H. Moon, S. H. Lee, S. Chang, J. K. Mok, and T. W. Park, “Method for control of steering angles for articulated vehicles using virtual rigid axles,” *International Journal of Automotive Technology*, vol. 10, no. 4, pp. 441–449, 2009.
- [105] P. Sweatman, “A study of heavy vehicle swept path performance,” *Special Report Australian Road Research Board*, 48, no. 48, 1991.
- [106] W. vdM Steyn and L. du Plessis, “Freight-truck-pavement interaction, logistics, and economics: Final phase 1 report (tasks 9–11),” Caltrans Division of Transportation Planning, Tech. Rep., 2014.
- [107] R. Garcia, “Roadway design manual,” *TxDOT Online Manuals, Texas Department of Transportation, Austin, Texas, USA*, 2014.
- [108] Council Directive 96/53/EC, “Laying down for certain road vehicles circulating within the Community the maximum authorized dimensions in national and international traffic and the maximum authorized weights in international traffic,” *Official Journal of the European Communities*, vol. 235, pp. 59–75, 1996.
- [109] J. Gattis and M. D. Howard, “Large school bus design vehicle dimensions,” University of Arkansas, Mack-Blackwell National Rural Transportation Study Center, Tech. Rep., 1998.
- [110] Y. Chen, Y. Cai, J. Zheng, and D. Thalmann, “Accurate and efficient approximation of clothoids using bzier curves for path planning,” *IEEE Transactions on Robotics*, in press.
- [111] J. Ali, R. Tookey, J. Ball, and A. Ball, “The generalised Cornu spiral and its application to span generation,” *Journal of Computational and Applied Mathematics*, vol. 102, no. 1, pp. 37 – 47, 1999.

REFERENCES

- [112] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.
- [113] L. Lu, “A note on quintic polynomial approximation of generalized Cornu spiral segments,” *Journal of Computational and Applied Mathematics*, vol. 253, pp. 123–130, 2013.
- [114] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, “Julia: A fresh approach to numerical computing,” *SIAM Review*, vol. 59, no. 1, pp. 65–98, 2017.
- [115] S. Brin, “Near neighbor search in large metric spaces,” in *Proceedings of the 21th International Conference on Very Large Data Bases*, ser. VLDB ’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 574–584.
- [116] R. B. Rusu and S. Cousins, “3D is here: Point Cloud Library (PCL),” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.
- [117] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [118] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, ser. SGP ’06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70.
- [119] I. Bitter, A. E. Kaufman, and M. Sato, “Penalized-distance volumetric skeleton algorithm,” *IEEE Transactions on Visualization and computer Graphics*, vol. 7, no. 3, pp. 195–206, 2001.
- [120] K. Saeed, M. Tabedzki, M. Rybnik, and M. Adamski, “K3M: A universal algorithm for image skeletonization and a review of thinning techniques,”

REFERENCES

International Journal of Applied Mathematics and Computer Science, vol. 20, no. 2, pp. 317–335, 2010.

- [121] L. Lam, S. W. Lee, and C. Y. Suen, “Thinning methodologies-a comprehensive survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 869–885, Sep 1992.
- [122] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.
- [123] D. Nieuwenhuisen, A. Kamphuis, M. Mooijekind, and M. H. Overmars, “Automatic construction of roadmaps for path planning in games,” in *International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, pp. 285–292.